Mpowerr

Empowering the Next-Gen Ecosystems

Thesis REPORT

FOR

An Artificial Intelligence-based Human Elephant Conflict Mitigation System (HECMS)

Author: Gunarakulan Gunaretnam

Affiliation: Mpowerr – Open-Source AI for Conservation

Date of Publication: 12 December 2022

Abstract

This project aims to build a computerized system to solve the topic "Human Elephant Conflicts.” As a sustainable solution for this problem, this research proposes the following early-warning system to be developed "An Artificial Intelligence-based Human Elephant Conflict Mitigation System." Sri Lanka is famous for wild animals and their natural beauty. Elephants are one of the critical components of Sri Lankan tourism. However, human-wild elephant conflicts are the most frequently reported issue in rural areas of our country. Several lives and properties have been lost due to past conflicts between humans and elephants. This system is going to perform the following methodology to prevent human-elephant conflicts: At first, this system will process the video(frame by frame) from CCTV / wildlife camera traps devices and try to discover elephants in video frames, and if it is found, then it sends an early warning message to the corresponding village with the location (Which area the elephant is identified) to alert the village people to take actions to prevent damage. This system uses external methods to block (scare) elephants, such as the artificial sounds of buzzing bees, firecrackers, and loud noises; A study shows that elephants are a bit afraid of buzzing bees and firecracker sounds in Sri Lanka. It sends an early warning SMS to the corresponding village authority's phone and an email to the elephant conflict management authority. It updates the conflict information in the database to be analyzed by the management team. Currently, the government is using the electric fence methodology to block elephants from entering villages. However, elephants can break the fence and enter villages. There is no systematic approach to solving this problem in Sri Lanka; there are no electric fences in some places. Typically, this task is assigned to real humans; their job is to monitor/watch for elephants in the middle of paddy fields or sitting in jungles during the nighttime like a watchman: if they find elephants, they will alert the village people and let them take actions, this research is aimed to build an Artificial Intelligence-based human-elephant conflicts mitigation system. This system should be able to monitor elephants during the day/nighttime to provide efficient results. This research tries to build cutting-edge technology to provide an end-to-end solution that includes features of elephant detection, early warning SMS/email, and a management system dashboard. The research outcome would be in the form of software; it doesn’t involve any

hardware-building process. This research tries to lead Sri Lankan communities in technology to be more powerful and productive to have beautiful lives for everyone.

Declaration

I, Gunarakulan Gunaretnam (2208408), declare that the research thesis report titled "An Artificial Intelligence-based Human Elephant Conflict Mitigation System" submitted by me for the fulfillment of the requirements of BSc (Hons) Computer Science & Software Engineering at the University of Bedfordshire UK is entirely my original work. I now attest that I have duly acknowledged and referenced all sources of information utilized in my research thesis paper, encompassing but not limited to journals, websites, and various scholarly and non-scholarly sources, in strict conformity with the citation style stipulated by my esteemed university.

I confirm that this research thesis paper represents my ideas, thoughts, and findings and has not been submitted for any other degree, diploma, or academic qualification. I have not plagiarized any content or data from any other source, and I have provided proper credit and attribution to all the works of others used in this thesis.

I also declare that I have followed all the ethical guidelines and principles in conducting my research, including obtaining necessary approvals and permissions, maintaining confidentiality, and adhering to academic integrity.

I accept that any violation of academic integrity or ethical guidelines in this research thesis paper may result in severe consequences, including but not limited to the rejection of my thesis or other disciplinary actions as per my university’s policies.

I take full responsibility for the accuracy, validity, and originality of the content presented in this research thesis paper. I accept that my thesis may be scrutinized, including by plagiarism detection programs, and I am prepared to give additional evidence as references to prove the validity of my work if necessary.

I hereby declare that this research thesis paper is my original work and meets the standards of academic integrity and ethical conduct. I have accepted and referenced all the sources

used, and I have complied with all the rules and regulations of the University of Bedfordshire - UK.

Acknowledgment

I want to express my heartfelt appreciation and gratitude to all those who have supported and contributed to completing my final project and thesis report; their encouragement, guidance, and assistance have been invaluable throughout this journey.

First and foremost, I am deeply grateful to my supervisor, Ms. Nideshika Ellepola, for her unwavering guidance, mentorship, and expertise. Her continuous support and feedback have shaped my research and pushed me to achieve my best.

I extend my heartfelt thanks to Ms. Gayana Fernando, my instructor, who lectured on Research Methodologies & Emerging Technologies and Undergraduate Project subjects. Her insightful lectures and guidance have been pivotal in helping me understand the intricacies of conducting research and developing my thesis.

I want to convey my deepest gratitude to the online evaluators who evaluated my research works from the University of Bedfordshire - UK. Their constructive feedback and assessments have been invaluable in refining my research and enhancing the quality of my thesis.

I am grateful to the SLIIT Academy and the University of Bedfordshire - UK for allowing me to pursue my BSc (Hons) in Computer Science & Software Engineering, enabling me to engage in this remarkable research.

I want to express my gratitude to the farmers and residents of Oddamavadi village, who generously provided me with on-ground information on human-elephant conflicts. Their first-hand experiences and insights have been crucial in understanding the real-world implications of my research.

I thank my colleagues, friends, and the administration staff at SLIIT Academy for their proper responsible support throughout this research and academic journey.

I express my deepest gratitude to all mentioned and unknown supporters for their support and contribution to my research thesis. I am genuinely grateful for that.

Table of Contents

List of Tables

List of Figures

Chapter 01: Introduction

1.1 Introduction

This chapter provides an overview of the research project addressing the critical issue of human-elephant conflict in Sri Lanka; it begins with background information on the project and discusses the fundamental concepts of the research in a story-telling manner to briefly introduce the research problem.

This chapter discussed the research problem and the project background, highlighting current approaches' limitations in effectively mitigating the conflict; it covers the aim and the research objectives to propose an Artificial Intelligence (AI)-based Human Elephant Conflict Mitigation System as a sustainable solution to reduce such conflicts.

The study's scope was covered precisely to give brief information on defining the research's boundaries and limitations. The significance of the study is emphasized, highlighting the potential impact of the proposed AI-based system in mitigating human-elephant conflicts in Sri Lanka. This chapter also briefly describes the report's structure, providing an overview of the chapters in subsequent sections. Additionally, it discusses the artifact's description and the prototype, explaining how the proposed project differs from existing methods that rely on the human workforce for elephant detection.

1.2 Research Background

The conflict between humans and elephants is one of the biggest problems in elephant-range countries. There are many reported issues and serious conservation concerns worldwide in elephant-range countries. Several management strategies have been developed and employed at different scales to reduce conflicts between people and elephants worldwide.

Human-elephant conflict is one the most popular and much-debated topics in Sri Lanka. There were 14,516 human–elephant conflict cases reported during 2010–2019. Approximately three hundred elephants were intentionally killed by humans each year; there were 807 total human deaths and 10,532 property damages caused by human-elephant conflicts in Sri Lanka, according to a 2020's research paper.

Moreover, Sri Lanka has endangered subspecies of elephants (Elephas, Maximus); therefore, protecting these species of elephants is one of the enormous responsibilities of the Sri Lankan communities to maintain the natural beauty and tourism sector.

Elephant tourism in Sri Lanka has become a prominent element of the country's tourism industry. Sri Lanka is known for its large population of wild elephants, and many tourists are drawn to the opportunity to observe these majestic creatures in their natural habitat. Elephant safaris, elephant orphanages, and elephant conservation projects are popular attractions for tourists visiting Sri Lanka.

Still, elephants are the golden gifts for the country's natural beauty; it has another side: human-elephant conflicts, as there is no solid solution to this problem yet. Currently, many innovators and social workers are motivated to build solutions that are all on prototype levels, such as IoT sensor-based systems for detecting elephants. The only implemented system for this problem by the government is an electric fence, which needs to be fixed. In contrast, elephants can break the fence; sometimes, the electric fence methodology does not work efficiently as expected.

However, This research is trying to implement an AI approach that uses a camera and computer vision technology to process video; the system plays artificial sounds effects such as buzzing bees and firecracker sounds to block elephants from entering the village, and at the same time the system can send early warning messages to responsible villagers' mobile phones and send warning emails to the system authority and the system also records elephants intrusion incidents on the database, this will give a glimpse to the village people to be alert. Usually, this method is done by manpower. In some villages, some humans sit and watch for elephants during nighttime. If the elephant is found, they forward the message to the nearest village; this research aims to automate this manual to machines with artificial intelligence technology.

In Sri Lanka, many youngsters and innovators are naturally motivated by this problem, especially people from the country's rural areas; this problem provides many reasons for innovators to be invented a solution for this problem.

This project aims to propose a computerized system to solve the topic "Human Elephant Conflicts." As a sustainable solution for this problem, this research suggests the following early-warning approach to be developed "An Artificial Intelligence-based Human Elephant Conflict Mitigation System."

The idea of the system is complex. But, it is very straightforward to understand that this system is going to perform the following methodology to prevent human-elephant conflicts: At first, this system will process the video(frame by frame) from CCTV / wildlife camera traps devices and try to discover elephants in video frames, and if it is found, then it sends an early warning message to the corresponding villages to alert the village people to take actions to prevent damage. This system uses external methods to block (scare) elephants, such as the artificial sounds of buzzing bees and firecrackers sound effects. This system stores this information, such as the number of elephants and incidents; all the information will be sent to the database and accessed by the elephant conflict team authority through a management panel to make future decisions with the collected data.

1.3 Research Problem

The human-elephant conflict issue in Sri Lanka is a significant challenge threatening human and endangered elephant populations. Sri Lanka is known for its rich biodiversity and wildlife, including its wild elephant population, which is a critical component of its natural beauty and a major tourist attraction. However, despite conservation efforts and tourism opportunities, human-elephant conflicts remain a pressing issue in the country.

Human-elephant conflicts in Sri Lanka are mainly triggered by elephants entering villages for food or water, damaging crops, homes, and property, and sometimes causing harm or casualties to humans. Elephants are large, powerful animals that can pose a significant risk to human populations, and property damage caused by elephants can result in economic losses for local communities. Moreover, as elephants are protected animals in Sri Lanka, harming or killing them is illegal, further complicating the situation.

This research is aimed to provide a technological solution for the human-elephant conflict issue in Sri Lanka. Sri Lanka is famous for tourism, and the elephant is one of the keys components of its natural beauty; despite the significant conservation efforts and tourism opportunities associated with Sri Lanka's wild elephant population, human-elephant conflicts remain a pressing issue, with thousands of incidents reported annually, resulting in human casualties and property damage.

The existing methods to mitigate human-elephant conflicts, such as electric fences, are only sometimes reliable in preventing incidents. As a result, there is a pressing need for innovative and sustainable solutions to address this issue. The research project aims to develop an AI-based system to accurately detect elephants in video frames captured by CCTV/wildlife camera traps. The system will then send early warning messages to villages, accompanied by artificial sounds that can deter elephants from entering the villages. In addition, the system will collect and analyze data that the elephant conflict management team can use for informed decision-making in the future. The goal is to create a reliable and effective system that can significantly reduce human-elephant conflicts and provide a sustainable solution for mitigating this conservation challenge in Sri Lanka.

1.4 Research Aim

This research aims to minimize human conflict incidents in Sri Lanka. To achieve it, the study is trying to build an automatic elephant intrusion detection system to send an early-warning message to the nearest villages and block elephants using external artificial sound effects.

This research sought to build an AI-based Human Element Conflicts Mitigation System to achieve the research's aim. It will accurately detect elephants in the video, send early warning messages to villages to alert residents, and utilize artificial sounds, such as buzzing bees and firecrackers, to deter elephants from entering villages. The project aims to create a sustainable solution that can effectively reduce human-elephant conflicts, protect endangered elephant populations, and minimize property damage and human casualties caused by such conflicts.

This study is planned to create a long-term solution to reduce human-elephant conflicts. Minimizing conflicts, protecting endangered elephant populations, and reducing property damage and human casualties.

Furthermore, the HECMS collects and analyzes data on elephant movements, behaviors, and intrusion patterns. This data provides valuable insights for the elephant conflict management team, allowing them to make informed decisions and develop proactive strategies to prevent future conflicts. The HECMS is a tool for evidence-based decision-making, enabling more effective management of human-elephant conflicts in Sri Lanka.

HECMS is an excellent solution that combines AI-based video processing, early warning messages, and artificial sounds to mitigate human-elephant conflicts in Sri Lanka. It aims to create a sustainable and practical approach to minimize disputes, protect elephants, and promote peaceful coexistence between humans and wildlife.

This study aims to develop an AI-based system called HECMS to minimize human-elephant conflicts in Sri Lanka. HECMS will detect elephants in video frames, send early warning messages to villages, and use artificial sounds to deter elephants from entering human settlements. By analyzing elephant movements and behaviors, HECMS aims to provide

insights for proactive strategies to prevent conflicts, protect elephants, and promote peaceful coexistence between humans and wildlife.

1.5 Research Objectives

1.5.1 Main Objectives

- To compare existing elephant detection models.
- Develop an AI-based video processing system that accurately detects elephants in videos captured by CCTV cameras or wildlife camera traps, with high accuracy and low negative rates.
- To alert villagers when elephants enter their premises by sending real-time early warning messages, allowing villagers to take proactive measures to prevent conflicts.
- To block elephants from entering the villages by playing artificial sound effects such as buzzing bees and firecrackers; this will discourage elephants from entering human settlements.
- To build an AI model for elephant detection using deep learning algorithms.
- To develop a web management panel to facilitate human-elephant conflicts data for monitoring elephant movements, behaviors, and intrusion patterns to generate insights for the elephant conflict management team to support evidence-based decision-making and proactive strategies to prevent future conflicts.
- To test and evaluate the accuracy of system performance.
- Monitor and evaluate the performance and impact of the HECMS system over time and continuously improve the system.

1.5.2 Sub Objectives

- Build an AI model for elephant detection.
    - Collect daytime elephant images.
    - Label images
    - Cleanup image data

    - Prepare dataset.
    - Write a Python script to build the model with TensorFlow.
    - Train the model (AI Brain)
- Build a monitoring system in Python (Main System)
    - Write a Python script to infer the trained models.
    - Implement OpenCV to process the videos (Input)
    - Write logic to break videos into frames.
    - Process frames with the trained models to look for elephants from the video input.
    - Write logic to send SMS/Emails as early warning messages to the nearest villages.
    - Write logic to play artificial sounds of buzzing bees and monkeys to block (scare) the elephants.
    - Write a logic to update the database if elephants are found with time/date.
    - Write logic to play warning sound effects.
    - Write logic to turn on/off the alarm.
    - Write logic to switch between trained AI models.
    - Write logic to adjust the brightness of the frame.
    - Write a function to adjust the contrast of the frame.
    - Write a function to adjust the threshold of model accuracy.
    - Write a function to run all processes in a looping manner.
    - Write a function to stream videos from IP Camera.
- Build a web panel (Management System)
    - Design the web panel (UI/UX)
    - Create the web panel with MERN Stack.
    - Connect the monitoring system’s (Main System) database with this web panel to view information on elephant discovery.
- Testing the functionalities
    - Test AI model accuracy.
    - Test the main system functionalities.
    - Test the web panel functionalities.

1.6 Research Questions

Some research questions are arrived at from the objectives and aim of the research; this research has the following questions to be answered to produce high-quality research outcomes.

- **Question 01:** What are the existing elephant detection models, and how do they compare in terms of accuracy, efficiency, and applicability in the context?
- **Question 02:** How can this AI human-elephant conflict system be developed to detect elephants in videos accurately?
- **Question 03:** Which deep learning model architectures are most suitable for building an AI model for elephant detection?
- **Question 04:** What are the impacts of playing artificial sound effects such as buzzing bees and firecrackers in discouraging elephants from entering the villages?
- **Question 05:** How can the management panel be designed and developed, and what features and functionalities should it be included to facilitate data monitoring and analysis of elephants?
- **Question 06:** What are the performance and accuracy metrics to evaluate the system performance, and how can they be measured effectively?
- **Question 07:** What are the challenges and limitations of the AI model for elephant detection, the primary system for video processing, and the web panel for data management, and how can they be addressed to ensure reliable and best performance?
- **Question 08:** How can the system be tested comprehensively to ensure the accuracy of the AI model, the functionality of the main system, and the usability of the web panel, and what strategies can be employed to troubleshoot and optimize the system during testing?

Note: These research questions will be answered and described in the methodology section with implementation methods and evidence data.

1.7 Research Scope

Developing an AI-based video processing system: This research tries to develop an innovative and cutting-edge AI-based system that can accurately detect and identify elephants in videos captured by CCTV cameras or wildlife camera traps. The system's core objective is to achieve better accuracy and lesser negative rates in detecting elephant intrusions.

Real-time early warning messages: The research seeks to implement a state-of-the-art system that leverages real-time data to promptly send early warning messages to villagers when elephants enter their premises to help them proactively avoid conflicts.

Artificial sound effects for blocking elephants: The research tries to incorporate artificial sound effects, such as the buzzing of bees and the vibrant bursts of firecrackers, as an innovative idea to prevent elephants from entering villages.

Web management system for data analysis: The research aims to develop a sophisticated and user-friendly web management panel that empowers conservationists and wildlife management teams with robust tools for in-depth analysis of human-elephant conflict data. This management system will provide unprecedented insights into elephant movements, behaviors, and intrusion patterns, facilitating evidence-based decision-making and proactive strategy development to prevent future conflicts.

Testing and evaluation of system performance: The research tries to conduct comprehensive testing and evaluation of the accuracy and implementation of the developed system. This multifaceted assessment will encompass evaluating the system's accuracy in detecting elephant intrusions, scrutinizing the effectiveness of real-time early warning messages and artificial sound effects, and meticulously assessing the usability and functionality of the web management panel.

1.8 Research Significance

It is envisioned as an innovative and reliable system that utilizes cutting-edge technologies, such as artificial intelligence and computer vision, to detect elephants in video frames. The system will be able to process video data from CCTV/wildlife camera traps in real time and then analyze each frame to identify the elephants. Once elephants are detected, the system will automatically send early warning messages to the responsible villages, alerting them to take proactive actions to prevent damage.

The safety of human lives and their properties need to be considered, and Human-elephant conflicts often result in property damage, crop raiding, and human casualties; this research is implementing an AI-based system, with its early-warning messages and block elephants using variant sounds, has the potential to prevent or minimize such incidents, safeguarding the lives and livelihoods of local communities residing in conflict-prone areas. This can increase safety and security for human populations and protect their valuable property and crops from damage.

The system will collect and analyze data on elephant intrusion incidents, including the number of elephants detected and the locations of incidents. This data will be stored in a database and accessible by conservationists. The data will be valuable for making informed decisions and developing effective strategies to mitigate human-elephant conflicts in the future.

Community engagement and participation are other factors needed to be considered; this project sends early warning messages to; The local communities' involvement in this solution project can foster a sense of ownership, empowerment, and cooperation among people, leading to increased effectiveness and sustainability of the conflict management efforts.

Replicability and scalability are other factors that take place here; the findings and insights gained from this research project could be replicated and scaled up in different regions and countries facing similar human-elephant conflicts.

The project integrates various disciplines, including artificial intelligence, wildlife conservation, and community engagement. This multidisciplinary approach has the potential to generate new knowledge and insights at the intersection of different fields, fostering collaboration and innovation in addressing complex societal and environmental challenges.

1.9 Report Structure

- **Title Page:** *Contains logo of UoB, module name, student name, ID, and submission date.*
- **Abstract:** *Contains the overview of the entire thesis.*
- **Acknowledgment:** *Contains Brief statement of vote of thanks.*
- **Table of Contents:** *Contains contents & structure.*
- **List of Tables:** *Contains* a *list of table information.*
- **List of Figures:** *Contains* figures, diagrams, and illustrations information.
- **Keywords:** *Contains abbreviations of long words used in this thesis.*

Chapter: 01 Introduction: *Explains the basic details of the research to get started.*

- Introduction
- Research Background
- Research Problem
- Research Aim
- Research Objectives
- Research Scope
- Research Questions
- Research Significance
- Report Structure

Chapter: 02 Literature Review: *Explains previous studies and root causes of the problem.*

- Introduction
- Literature Review Body

- Research Gap
- Literature Review Conclusion

Chapter: 03 Methodology: *Explains research techniques and methods.*

- Introduction
- Research Philosophy
- Research Design
- Research Approach
- Research Strategy
- Project Design
- Data Collection Methods
- Data Analysis Methods
- Implementation
- Summary

Chapter: 04 Results: *Explains findings and research outcomes.*

- Introduction
- Project Outcome
- Project Testing
- Accuracy Analysis

Chapter: 05 Discussion: *Discuss the findings of the study.*

- Introduction
- Key Findings
- Interpretations
- Implications
- Unexpected Occurrences

Chapter: 06 Conclusion: *Explains the summary of the research.*

- Introduction
- Limitations
- Recommendations

- Future Work
- Closing Thoughts

References: *Contains sources and citations referred to in the thesis.*

Appendices: *Contains supplementary materials.*

Chapter 02: Literature Review

2.1 Introduction

This chapter contains the Human Elephant Conflicts Mitigation System's research literature review. It describes all the relevant information related to the research domain and includes literature from basic to advanced, covering non-technical to technical details.

The scope of this literature review is to provide a comprehensive overview of the existing research and literature related to human-elephant conflicts in Sri Lanka, including the causes, consequences, and current mitigation measures. The review will also aim to identify gaps or limitations in the existing literature and highlight areas requiring further research. Additionally, the literature review will examine the potential of using AI-based technologies, such as video processing and early warning systems, for mitigating human-elephant conflicts in Sri Lanka.

This chapter contains the literature review in the body section with multiple subheadings covering a specific part of the literature, such as "Elephant in Sri Lanka." or "Elephant and Tourism in Sri Lanka." Each subheading covers a literature review related to that part, and finally, it has a "Research Gap" section that explains what things are missing and will be covered in the future of this research. It ends with a literature review conclusion summarizing the entire review in one page.

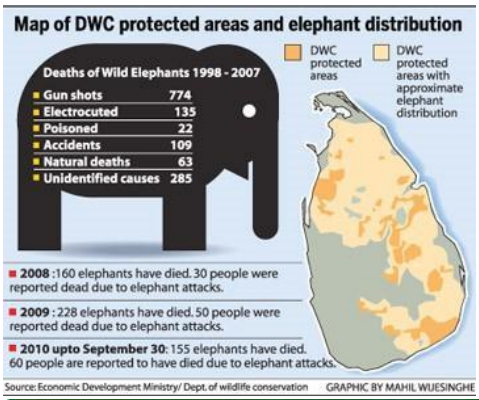
As a layout, this literature review covers the positive aspects of elephant tourism and the various types of elephants found in Sri Lanka. It also delves into the negative side of human-elephant conflicts, including incidents and struggles farmers face. Traditional methods used by villagers to mitigate conflicts are discussed, along with existing solutions by

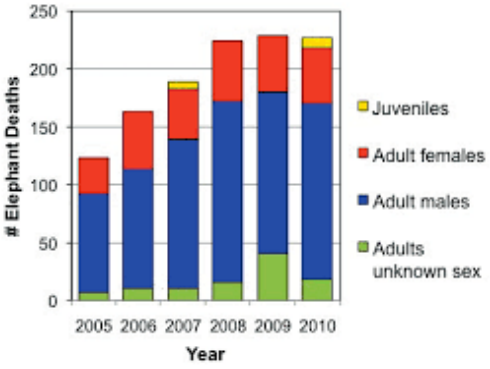
the Sri Lankan government. The review also explores technological solutions used in elephant-range countries globally and highlights the potential of AI and computer vision technology in mitigating human-elephant conflicts. The study identifies critical findings, trends, gaps, and limitations in the existing research and the potential contributions of AI and computer vision technology in addressing the human-elephant conflict in Sri Lanka.

2.2 Literature Review Body

2.2.1 Elephants in Sri Lanka

According to a research animal science paper, Sri Lankan elephant types are (Elephas maximus maximus) and are native to Sri Lanka. The Asian elephant (Elephas maximus) is categorized as Endangered in the IUCN Red List (IUCN, 2007). They have recognized subspecies of the Asian elephant (Campos-Arceiz et al., 2008). Among all the beasts of the world, the elephant is the only one demonstrating God's power and wisdom like no other. (TOPSELL, 2021). The Sri Lankan elephant population is currently mostly restricted to the dry zone within the north, east, and southeast of land. Elephants area unit gift in Udawalawe park, Yala park, Lunugamvehera park, Wilpattu park, and Minneriya park however conjointly live outside protected areas (*Article from* https://www.natureodyssey.com/). Sri Lanka holds a crucial position with relation to Asian elephant conservation. spill 100 percent of the world Asian elephant population in but a pair of of elephant vary (Leimgruber et al. 2003). Human–elephant conflict (HEC) is a key environmental issue in several Asian countries, including Sri Lanka. Incidents of HEC have significantly increased in Sri Lanka between 1991 and 2018, with 1734 human deaths reported in this period (281% increase), 4837 elephant deaths (1172% increase), 1053 human injuries (140% increase), and more than 23,000 property damage reports (1406% increase). The elephant is taken into account joined of the celebrities of the kingdom Animalia and may be a distinguished image of life conservation in Sri Lanka. The elephant has compete a central half in Sri Lankan history, culture, faith and mythology (and latterly politics) for over twenty-five centuries (Santiapillai et al., 2010). Tamed elephants have long been used for numerous forms of add the country as well as moving hardwoods, building and transportation hundreds.





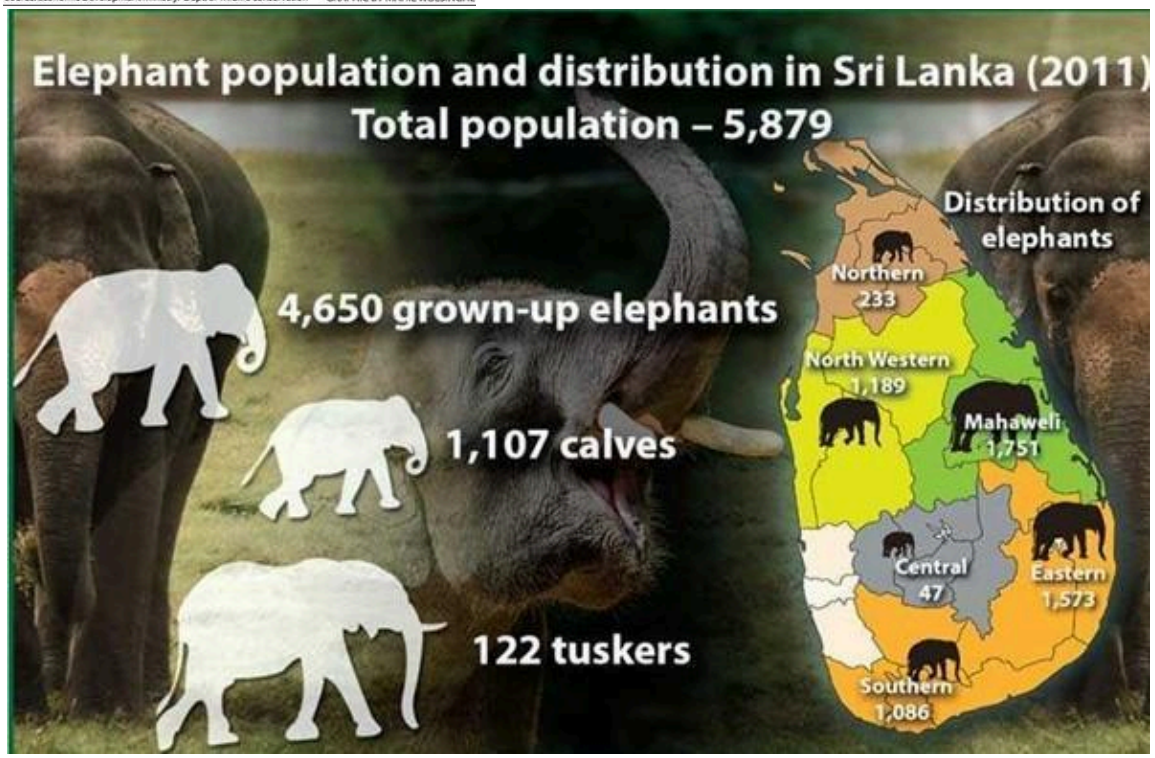


Figure 4: Elephant Population in Sri Lanka.

2.2.2 Elephants and Tourism in Sri Lanka

The island is famous for its tourism. Some of the favorite places visited by the tourists are Yala, UWNP, Pinnawala Elephant Orphanage, and the Minneriya-Kaudulla national parks *(ft.lk).* Most significantly, the general value of one wild elephant to Sri Lankan commercial enterprise is on the brink of Rs. Four million or $ twenty one,400 *(lankaenvironmentfund.org Dec 25, 2020).* Elephant-related tourism is worth over Rs. 1 billion or $ 5.8 million per year to Sri Lanka – over four times as much as tea! *(ft.lk).* According to the statistics, 1,100,435 international guests visited Sri Lanka in 2018, visited Sri Lanka's several life and national

parks. Sri Lanka earned Rs. 2,138,450,422.38 in entrance fees alone particularly the elephants were one amongst the favorite picks among them in addition *(ft.lk – 2018 census)*

Figure 5: Sri Lankan Elephant Tourism 1

Figure 6: Sri Lankan Elephant Tourism 2

Figure 7: Sri Lankan Elephant Tourism 3

Figure 8: Sri Lankan Elephant Tourism Image 4

2.2.3 Human-Elephants Conflicts in Sri Lanka

In conflicts between elephants and humans, crop-raiding, injuries and deaths to humans caused by elephants, and elephants killed by humans for reasons other than ivory are common *(Research Paper: wwf.panda.org and World Wildlife Fund). However, human-elephant conflicts also arise; 3,500 elephants require around 17,500 km2, or 27 percent of the total land area, while the protected areas in Sri Lanka cover only 12.5 percent of the land area (or 8,200 km2). The lack of sufficient land area for the existence of elephants is the foremost reason for human-elephant conflicts (Ceylon Expeditions). A total of 112 persons were killed due to elephant attacks in 2020 (xinhuanet.com).* The conflict threatens community livelihoods and food security and may even cause human fatalities. The ensuing economic stress to marginalized rural communities causes over $10 million in crop and property injury annually in the state (Sargent, 2021). The institution of protected areas that give feeding, breeding, and residing habitats to the elephants may play a key role in addressing human-elephant conflict in the country. These areas can physically separate humans and elephants. Moreover, effort corridors that connect fragmented habitats can offer additional areas for these mammals to graze. The presence of corridors supports the elephants in their seasonal migration and helps them search out food throughout the year *(Karunananda slycantrust.org)*.

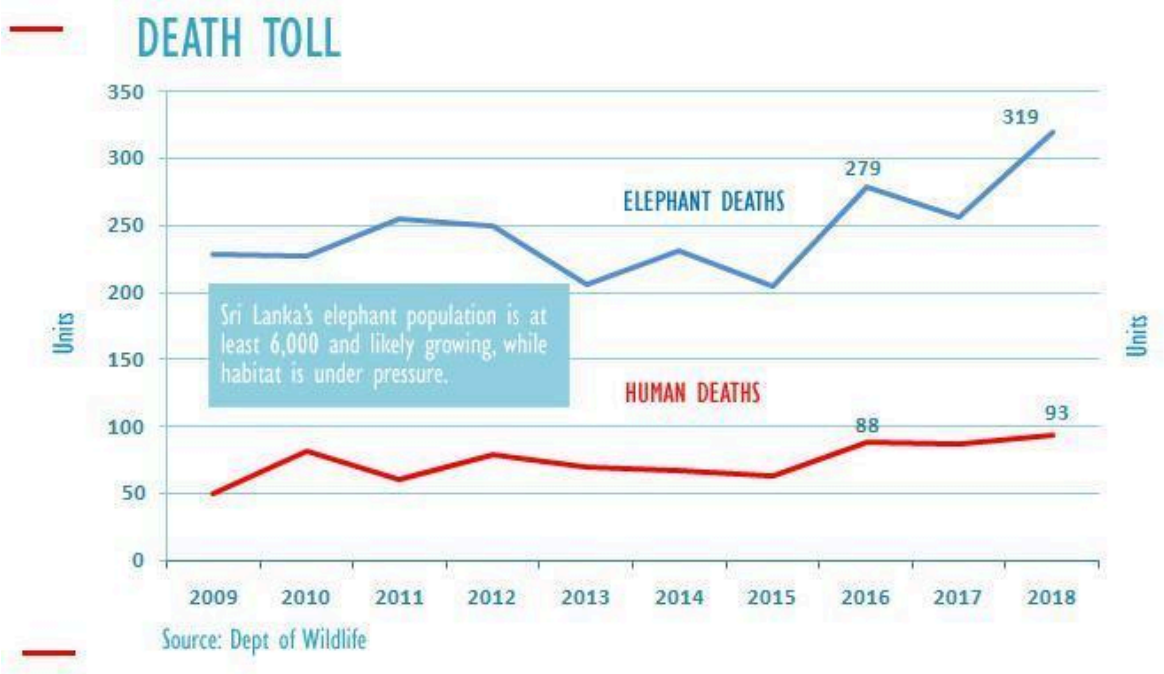


Figure 9: Human-Elephant Conflicts Chart

Figure 10: Human-Elephant Conflict Image 1

Figure 11: Human-Elephant Conflict Image 2

Figure 12: Human-Elephant Conflict Image 3

2.2.4 Elephants Destroy Farmer's Lives

According to the research, the farmer's earn and their life are affected by human-elephant conflicts: The overwhelming majority (89%) of the farming families seem to earn but Rs. 120,000 (US$ 1,200) each year. In 100% of cases, the annual financial gain ranges between Rs. 120,000 and Rs. 240,000, whereas only one earns quite Rs. 240,000 (US$ 2,400). The annual financial gain of the families seems to be sadly inadequate. The poorest forty-fifth earns only Rs. 3,000 (US$ 30) monthly. Because of the losses area unit involved, twenty-fifth incur associate degree annual loss of but Rs. 5,000 whereas thirty seconds incur between Rs. 5,000 to Rs. 20,000 a year. However, the bulk (43%) suffers associate degree annual loss of over Rs. 20,000 (US$ 200). Farmers everywhere around the globe do exaggerate their losses; even so, their losses are unit real and, therefore shouldn't be unheeded. (*Rukmali Athurupana October 2010*)

Figure 13: Elephant Destroy Corps 1

Figure 14: Elephant Destroy Corps 2

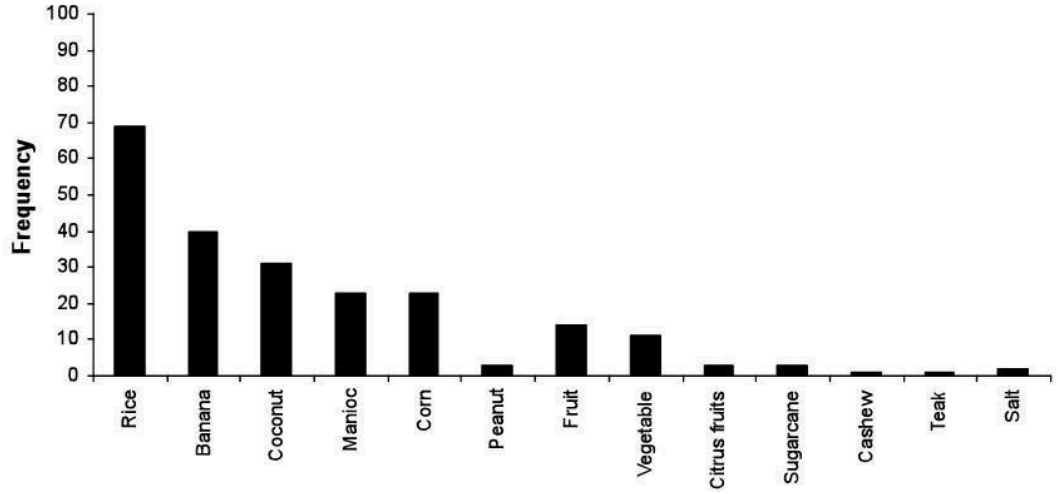


Figure 15: Elephant Preferred Foods

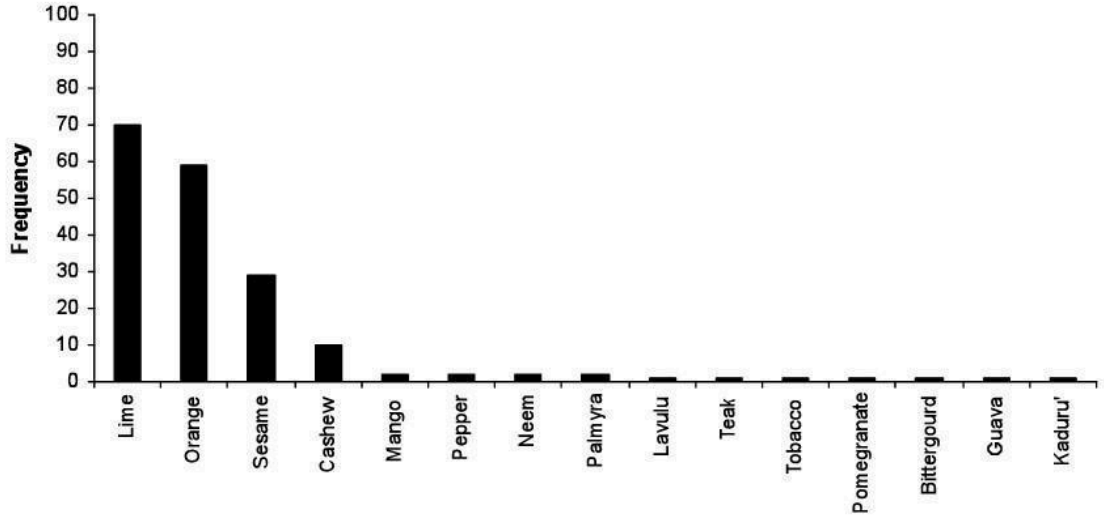


Figure 16: Elephant Avoid Foods

2.2.5 Existing Solutions for Human Elephant Conflicts

2.2.5.1 Building Electric Fences

Electric fence area units are the foremost effective human-elephant conflict mitigation measure if they're strategically set and well-enforced. Such fences seem to be the people's favorite answer to the elephant mitigation problem since they physically separate human and elephant territories. The IUCN states that this technique may be economically economical thanks to conserving these valued animals.

The electrical fences don't stop the impact of elephant attacks. Some studies have found that some elephants square measure 'habitual fence breakers' (Peiris & Sooriyabandara,

2014) alternative studies have shown that fences must be compelled to be electrified in most grassland elephant ranges since these square measure places wherever crop raiders are determined and protracted. However, other studies realize that electrification isn't required for the maximum amount in forest elephant ranges wherever elephants seem not to be thus persistent at crop offensive (Peiris & Sooriyabandara, 2014).

Figure 17: Elephant Breaks Electric Fence

2.2.5.2 Other Existing Natural Solutions

Farmers use external methods to scare the elephants, such as shouting and firecrackers to chase away elephants entering their fields. Building an automatic device to make sound is important because farmers usually stay up the whole night while guarding their cultivation which affects their day-to-day activities *(Karunananda slycantrust.org);* Elephants can be chased away by using bonfires, torches, and flashlights. Solar flashlights, which are sustainable, can be used to make this process more efficient.

Farmers can distribute the lights and place them strategically to cover the entire agricultural area. Light repels systems have demonstrated positive results in preventing elephants from entering cultivated lands, according to the *(Wildlife and Nature Protection Society of Sri*

Lanka's Annual Report 2019). To ensure the efficiency of the process, farmers must follow up on maintenance.

Growing unattractive crops: Elephants are fond of crops such as paddy. By growing less attractive crops together with these crops, farmers can reduce the impacts of elephants on their cultivation *(Karunananda slycantrust.org)*;

Translocation: Elephants that damage houses, feed on crops, and even kill humans can be tranquilized and transported to another location. After some time, translocated elephants return to their original territories, according to studies. As a result, this is not a long-term solution to the conflict between humans and elephants.

2.2.6 Technology-Based Solutions

Today, machine learning and artificial intelligence use is crucial as technology continues to grow. AI can solve most past unsolved circumstances with better implementation. There is a growing interest in artificial intelligence, and most tech personnel believe that AI is the future. In addition to reducing the human-elephant conflict, AI systems were and are developed that send alerts to the villages that enable them to dread elephants before they reach the destination of the system, especially the buzzy bee sound that is widely used to frighten elephants *(BINITA KHANAL Enschede, The Netherlands, June 2022)*.

AI system involves small cameras that work remotely, hidden in a tree above the reach of elephants. When the camera's motion sensor is triggered, it uses computer vision to detect

elephants in the frame. It transmits those images in near-real time to the cell phones of village guardians (WildEyes AI by NGO RESOLVE and CVEDIA).

The scalable, affordable system, funded partly by One Earth, uses new advances in computing technology to find elephants and transmit alerts to life managers and communities to forestall conflict things before they occur (WildEyes AI).

The Global Positioning System (GPS) collar system used today is another approach towards tracking elephants currently used only to identify migration patterns of elephants (Gamage & Wijesundara, 2014).

These days modern technology enables the ability to monitor elephants' movement with GSM and satellite images. The collected elephant movement data can be analyzed and monitored with the help of artificial intelligence algorithms. These software algorithms issue alert messages and facilitate rapid response if some elephants become unnaturally immobile. STE tracking app uses AI algorithms to visualize and track elephants and investigate their behavior (*Save the Elephants project: https://www.savetheelephants.org)*

In Africa, They are leveraging Artificial Intelligence and Machine Learning technologies with drone and camera to stop poaching issues. There is a vision processing platform called (Neurala's AI) that platform is AI-powered and that software can process terabytes of video data in real-time captured by drones, this software uses the deep neural network to track and predict the movement patterns of elephants and other animals (Neurala's AI: https://www.neurala.com).

in Africa, The WildEyes AI is a camera system that uses artificial intelligence technology to monitor African elephants and sends alerts in real time; helping to greatly reduce the risk of human-elephant conflict or enhance research and monitoring efforts; WildEyes AI's journey started in 2014 during the Great Elephant Census, a continent-wide survey of the entire African elephant population conceptualized (by Paul G. Allen).

In Cambodia, "Wild Earth Allies" is a community-based startup using Camera Traps to protect Asian Elephants. Camera traps are an important conservation tool, as they facilitate remote wildlife monitoring *(Wild Earth Allies 2021)*.

Figure 18: Camera Trap in Cambodia (Wild Earth Allies)

The way computers learn, it learns different variations within a species, as well as what an animal might look like in a wide variety of poses and from different perspectives (Arjan Wijnveen CEO of CVEDIA), According to Dinerstein and Wijnveen, this not only makes the AI better-placed to recognize its target, but also prepares the system to recognize it from any angle.

When a camera is deployed in the field, all of this training is stored in a deep neural network on the SD card. This is the same technology that allows a self-driving car to recognize a pedestrian in a crosswalk ahead *(Geib, 2020).*

The camera , designed by RESOLVE engineer Steve Gulick, is housed in a thin camouflaged strip that houses a motion detector and two sensors for different lighting conditions. It is powered by a battery that lasts at least one and a half years (WildEyes).

Believe the WildEyes can be integrated into existing deterrent systems. Her organization provides farmers with wildlife conflict training and deterrent kits developed by the organization Honeyguide, which use light, noise, and trunk-tickling chili pepper to turn African elephants away from fields humanely *(Silvia Ceppi, a scientific adviser with the NGO Instituto Oikos' East Africa office, Tanzania).*

ForestGuard AI is a platform built by CVEDIA for potential future use, where it could spot logging trucks as they enter forests, to stop illegal timber harvest before it happens (*CVEDIA*).

The experience has shown that farmers are willing to invest at least half the cost of powerful torches, which are the most expensive asset in the toolkit; farmers are willing to invest in crops-protection technologies that are convincing and proven to work *(Ceppi - Researcher).*

Perhaps, in the future, we can protect elephants while giving farmers more good nights' sleep their farms protected not by human eyes, but by AI *(RESOLVE was a co-founder of Mongabay's WildTech)*

Tracking imperiled elephants with cutting-edge technology, with the warden, we started tracking elephants for security in Meru National Park in 2003. Those were old-fashioned collars; WildTech spoke with Iain Douglas-Hamilton (O'Brien & Palminteri, 2015)

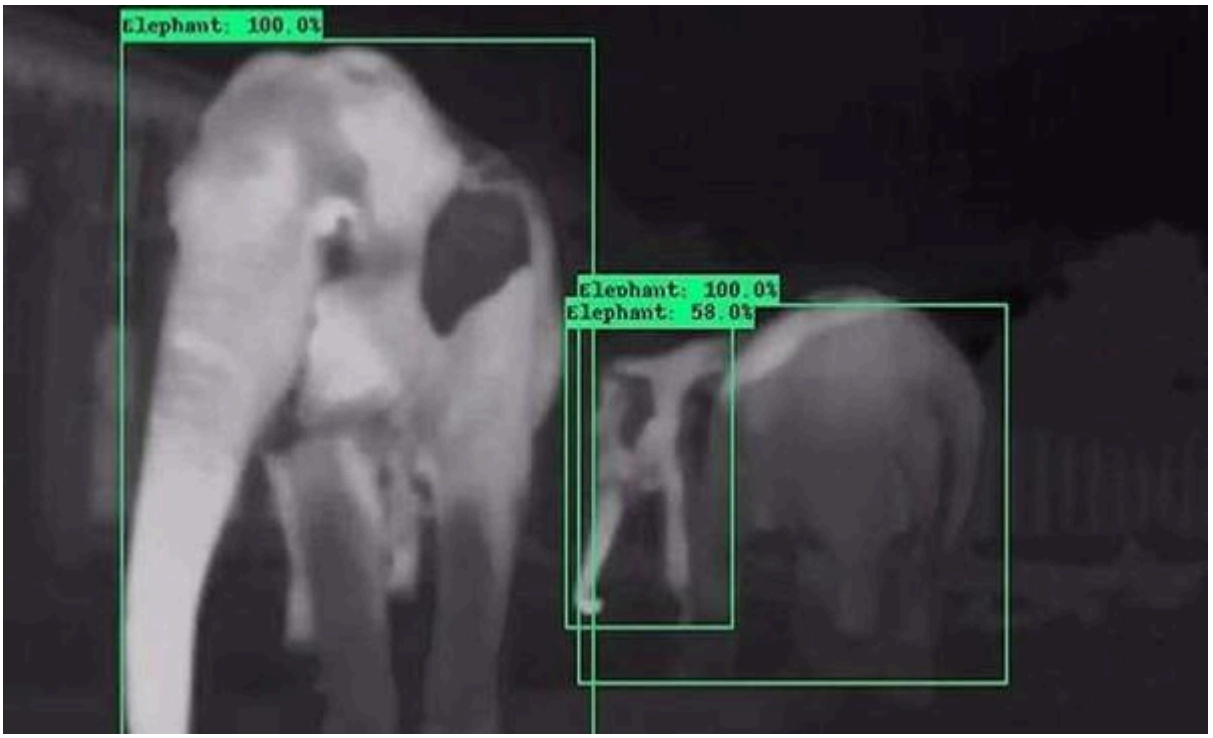


Figure 19: WildEyes Camera Trap Output

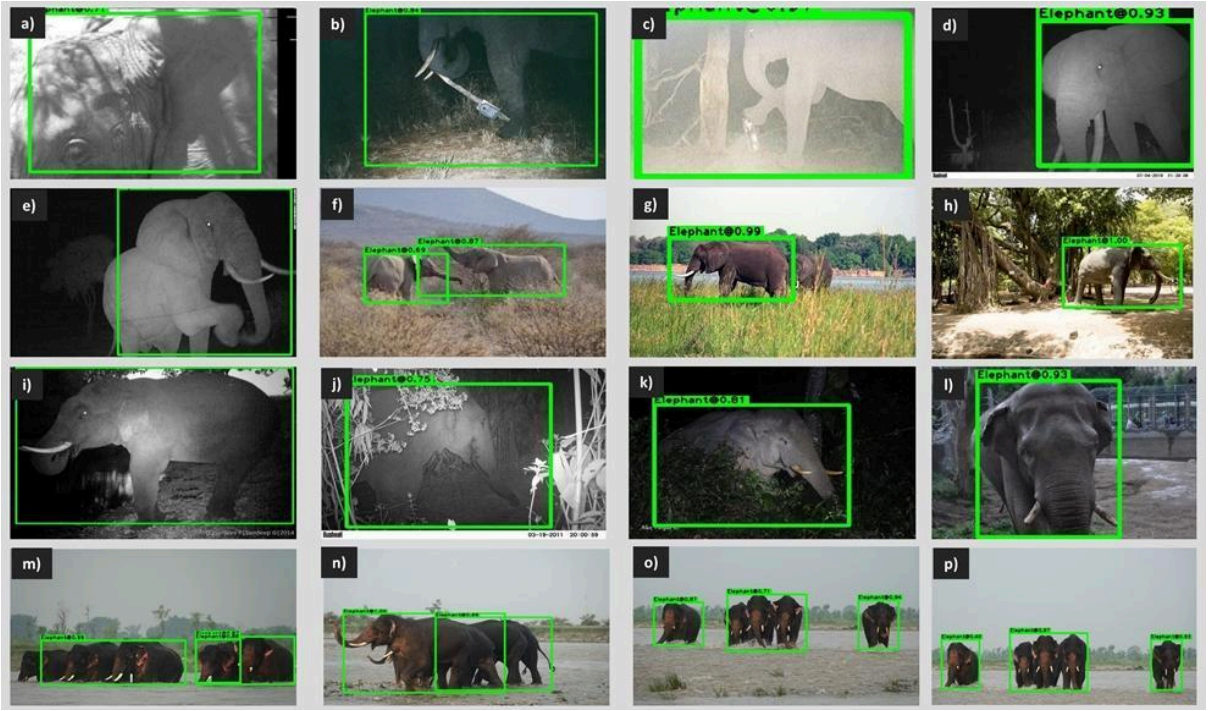


Figure 20: WildEyes Camera Trap Output

The Elephants and Bees Project is part of the Human Elephant Coexistence Program at Save the Elephants, which is based in Sagalla, Kenya, near Tsavo National Park. Our award-winning project investigates the use of Beehive Fences as a natural elephant deterrent, thereby assisting in the protection of farmers and farmland. The concept is based on our innovative research that uses elephants' fear of African honeybees to help reduce crop damage and other human-elephant conflicts *(Elephants and Bees Project: https://elephantsandbees.com/)*

Figure 21: Elephants and Bees Project

We spend 80 percent of our time at the Sagalla test site [near Tsavo East National Park], and we've been assisting new partners in Africa and Asia in establishing beehive fence projects [here's one from South Africa]. Of course, these fences will not solve all HEC issues (*Douglas-Hamilton WildTech) (O'Brien & Palminteri, 2015).*

2.2.7 Computer Vision Technology

A convolutional neural network (CNN) is an artificial deep neural network used for feed-forward to achieve computer vision-related tasks, for example, classification and identification of images / digital videos; CNN is like conventional neural networks but with deeper strata. This has weights, biases, and a nonlinear efficiency Switch (S, Ramesh, and Divya, 2016).

Computer vision is an interdisciplinary scientific field that deals with how computers can gain a high-level understanding of digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can *do (Zhanyu Ma; Haibin Ling; Yi-Zhe Song; Timothy Hospedales; Wei Jia; Yuxin Peng 25 June 2018)*

We are attempting to do the inverse in computer vision, that is, to describe the world we see in one or more images and reconstruct its properties, such as shape, illumination, and color distributions. Humans and animals can do this so effortlessly, whereas computer vision algorithms are notoriously prone to errors. People who have not worked in the field frequently underestimate the problem's difficulty. This misconception about vision dates back to the early days of Artificial Intelligence when it was thought that intelligence's cognitive (logic proving and planning) components were inherently more difficult than the perceptual components (Boden: Source: Computer Vision: Algorithms and Applications by Richard Szeliski).

Parts of the complexity of the human vision system are being replicated, allowing computers to identify and process objects in images and videos in the same way that humans do. Until recently, computer vision could only perform limited tasks. With advances in artificial intelligence and innovations in deep learning and neural networks, the field has been able to make significant strides in recent years, surpassing humans in some tasks related to object detection and labeling *(Ilija Mihajlovic)*

YOLO (you only look once) is an object detection algorithm that divides images into grids. Each grid cell is in charge of detecting objects within itself. Because of its speed and accuracy, YOLO is one of the most well-known object detection algorithms (Bouchard, 2020)

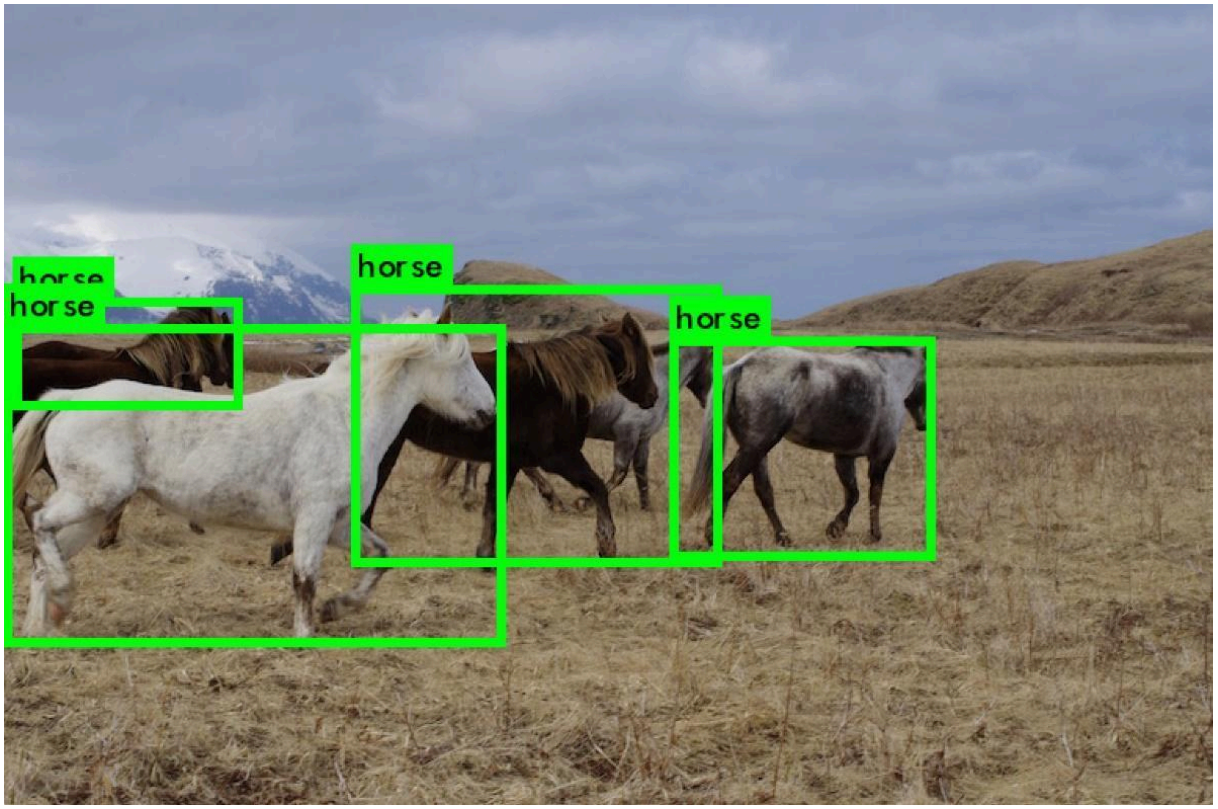
The field of computer vision is an intellectual frontier. It is exciting and disorganized, as is any border, and there is frequently no reliable authority to appeal to. Many useful ideas need

more theoretical foundations, and some theories are useless in practice; developed areas are widely dispersed, and one often appears completely inaccessible from the other. Nonetheless, we have attempted to present a fairly orderly picture of the field in this book *(Computer Vision: A Modern Approach by D. Forsyth, J. Ponce).*

Applications built on animal detection play a vital role in providing solutions to various real-life problems *(S. Sharma and D. J. Shah Jun. 2013.)*

To reduce observation costs, labor, and logistics, ecologists increasingly turn to greater automation to locate, count, and identify organisms in natural environments *(Pimm et al., 2015).*

Automating the Analysis of Camera Trap Images: Prior to the widespread adoption of deep learning systems, computer vision researchers devised several innovative and moderately successful methods for automating the analysis of animals from camera traps using raw pixel data from images *(Schneider et al., 2018)*



2.3 Research Gap

Despite significant efforts to mitigate human-elephant conflicts in Sri Lanka, the issue remains a persistent challenge with socio-economic and ecological implications. However, there needs to be more research to understand the potential of technology, specifically artificial intelligence (AI) and computer vision technology, as a novel approach to mitigate human-elephant conflicts in Sri Lanka.

Although the research provides better software solutions for human-elephant conflict issues in Sri Lanka, this research investigated a lot at the software level; when it comes to the hardware level, this research only suggests implementing the software on tiny micro-computers such as Jetson Nano. In the future, this research needs to study suitable hardware components such as FPGA or check its hardware to come up with.

While some studies have explored technology solutions for human-elephant conflict mitigation, there needs to be more research on the specific application of AI and computer vision technology in this context. The potential of these technologies to detect and monitor elephant movements, predict their behavior and provide real-time alerts to farmers and authorities still needs to be explored. Moreover, there needs to be more studies investigating these technologies' acceptability, feasibility, and effectiveness in the socio-cultural and environmental context of Sri Lanka, where elephants hold cultural and religious significance.

Addressing this research gap could contribute to developing innovative and sustainable approaches to mitigate human-elephant conflicts in Sri Lanka. By harnessing the potential of AI and computer vision technology, this research could provide valuable insights into the feasibility, effectiveness, and socio-cultural acceptability of technology-based solutions in human-elephant conflicts. Furthermore, this research could contribute to the broader literature on wildlife conservation, conflict mitigation, and the use of technology in biodiversity conservation efforts, with implications for other regions facing similar challenges with human-wildlife conflicts.

2.4 Literature Review Conclusion

Elephas and Maximus are the Sri Lankan subspecies of elephants living there, and elephants play a vital role in tourism and Sri Lankan culture, such as traditional religious events like "Kandy Perahera," etc. They are treated as a treasure in some circumstances.

Conversely, there is a severe problem separating humans and elephants: human-elephant conflicts. Wild elephants turn into enemies of humans and vice versa. This problem causes much damage to both sides, such as deaths and property damage, rural areas are people's lives, and Sri Lanka is slowly losing its natural beauty of elephant tourism.

Many techniques are used to avoid human-elephant conflicts in Sri Lanka, such as electric fences and natural methods of scaring elephants. There are many studies related to this problem with technology, such as IoT-based solutions. But students only do these kinds of research for academic purposes, less severe research; this causes the local Sri Lankan problems to keep existing in Sri Lanka.

This human-elephant conflict problem exists in many countries; in some countries, they use artificial intelligence; WildTech, ForestGuard AI, WildEyes, and *Wild Earth Allies* are similar startups using technology and artificial intelligence for this problem.

WildEyes is a popular community-based startup that solves this human-elephant conflict issue with technology and artificial intelligence in Africa. Wild *Earth Allies* is another similar startup in Cambodia for this problem; ForestGuard AI and WildTech are community-based organizations that work for wild animals and the environment with technology.

There is a project called Elephants and Bees (Beehive Fences). They are growing bees to avoid elephants because elephants fear buzzing bee sounds; it involves using beehives as a natural deterrent to keep elephants away from farmland.

According to the farmer’s interview, Sri Lanka villagers and farmers use firecrackers to avoid elephants entering the villages.

in a conservation or wildlife management setting. Computer vision and image processing technologies have revolutionized how elephants and other animals can be detected and monitored in their natural habitats, leading to more effective conservation efforts.

Chapter 03: Methodology

3.1 Introduction

This previous chapter discussed the literature review of the Human Elephant Conflicts Mitigation System research; this chapter explains the study's methodology and clearly describes how this study is methodologically implemented correctly.

This chapter first explains the research philosophy and the design patterns; then, it deeply dives into the research approach and strategy; once it has covered all the research aspects of the study, then it discusses the project design, such as how the project was structured, planned, and implemented.

This section also explained various aspects of the research process, including project description, project requirements, project architecture, database architecture (ER diagram), system structure (UML class diagram), system interactions (UML use case diagram), algorithms designs (flowcharts), project planning phase (WBS diagram), project wireframes, project user interfaces, data collection method, data analysis method, and implementation.

The research philosophy outlines the underlying beliefs and principles guiding the study, while the research design elucidates the overall plan and structure. The research approach and strategy highlight the method adopted to address the research questions or objectives and the strategies employed to gather and analyze data.

The project design section delves into how the project was conceptualized, structured, planned, and implemented, including project requirements, project architecture, database architecture, system structure, system interactions, algorithms designs, and project wireframes. The project planning phase is detailed through a work breakdown structure (WBS) diagram, which delineates the tasks, dependencies, and timeline for project completion.

The data collection and analysis methods describe how data was collected and analyzed to generate results and findings. Finally, the implementation section discusses the practical implementation of the research outcomes or findings in a real-world context. This chapter provides a comprehensive overview of the research methodology and guides the systematic and rigorous implementation of the research study

3.2 Research Philosophy

The research philosophy for the Artificial Intelligence-based Human Elephant Conflict Mitigation System is grounded in a pragmatic approach that combines positivist and interpretive paradigms. The research will adopt a positivist paradigm by utilizing quantitative methods to gather and analyze data on human-elephant conflicts in Sri Lanka, including the number of incidents, the effectiveness of the AI-based system in detecting elephants, and the impact of artificial sounds on deterring elephants from entering villages.

Problem-solving orientation: The research aims to provide a practical solution to the pressing issue of human-elephant conflicts in Sri Lanka. It focuses on developing an AI-based system to detect elephants effectively and send early warning messages to villages to prevent conflicts. The research prioritizes the implementation and effectiveness of the system in mitigating human-elephant conflicts.

Interdisciplinary approach: The research recognizes that addressing human-elephant conflicts requires a multidisciplinary approach that combines knowledge and expertise from fields such as computer vision, artificial intelligence, conservation biology, social sciences, and community engagement. The research will integrate insights from various disciplines to develop a comprehensive solution considering the issue's technical and social aspects.

Community engagement: The research acknowledges the importance of engaging with local communities, elephant conflict management teams, government agencies, and others throughout the research process. Their perspectives, experiences, and feedback will be considered in developing and evaluating the AI-based system. Collaborative partnerships with stakeholders will be fostered to ensure the relevance and sustainability of the proposed solution.

Evidence-based decision-making: The research adopts a data-driven approach to inform decision-making. Data on human-elephant conflicts, system performance, and stakeholder perspectives will be collected, analyzed, and interpreted to draw meaningful conclusions and recommendations. The research will strive to provide robust and reliable evidence to support the effectiveness and sustainability of the proposed solution.

3.3 Research Design

This project used the **quantitative method** as a research methodology since it involves many mathematical operations, countable data, and statistics. Usually, the primary scope of this project is to build an artificial intelligence model that can detect elephants, and it can also send early warning SMS to the nearest villages and an email to the authority. Therefore, this research involves a lot of data collection and annotation, data parameter switching, etc.

Typically, in machine learning, to build a better performance model, we need to have many data in different shapes of perspectives, and we also need to produce sample data augmentation to generate more data.

In this research, a better AI model must be developed. Developing a better model means more training, testing, evaluating, dataset increasing, data augmentation, parameter tuning, model tuning, and a lot of scripting and mathematical approaches are involved in this study to build a computer vision model; Therefore, a lot of quantitative methods will be used in this study to achieve the right destination.

To train a computer vision model, we need to feed many images of elephants; as this tries to build two different models, those are daytime and nighttime models, we need to collect datasets for both models; as a data collection method, this research is going to use the offline and online collection, to collect images of elephant, the researcher will visit the zoo and collect videos of an elephant then it will be converted as frames during the training process, this is the offline method, as online methods, this research will download some publicly available dataset of elephants.

This research also falls under the category of **experimental study** type since it involves much experimentation such as coding, building models, data analysis, this research process/analysis, and experiment data collected for this study. Therefore, this research goes under the category of experimental investigation.

This research falls under the category of **Applied Research** because this research involves practical problem-solving and action-oriented solutions. This is the **nature** of this research is applied research.

3.4 Research Approach

As a research approach **Deductive Approach** is chosen for the Human Elephant Conflicts Mitigation System HECMS because this research involves a lot of theories and methods to drive this research properly, there are several reasons why the deductive approach is well suitable.

For example, the research problem of human-elephant conflicts in Sri Lanka is clearly defined. The problem statement highlights the challenges posed by elephants entering villages, damaging crops, homes, and property, and causing harm or casualties to humans. The need for innovative and sustainable solutions to mitigate human-elephant conflicts is also emphasized. This well-defined problem allows for a **deductive approach** where the research can proceed systematically to test hypotheses and draw conclusions.

Many existing knowledge and theories can be applied. The research project aims to develop an AI-based system to detect elephants in video frames captured by CCTV/wildlife camera traps and send early warning messages to villages. This approach can leverage existing knowledge and theories in computer vision, artificial intelligence, and wildlife conservation to design and implement the system. Deductive reasoning can be used to build upon existing theories and research findings to develop a robust and effective solution.

The other important thing is that quantitative data can be collected and analyzed. This research project involves processing video frames from CCTV/wildlife camera traps and collecting data on elephant sightings, incidents, and system effectiveness. This quantitative data can be systematically collected and analyzed using statistical methods to test hypotheses and draw conclusions. Deductive reasoning can be used to develop hypotheses based on existing theories and research findings and then collect and analyze data to confirm or refute these hypotheses.

A deductive approach is suitable for the research project as it allows for systematic testing of hypotheses, leveraging existing knowledge and theories, collecting and analyzing quantitative data, rigorous evaluation of results, and potential generalizability of findings.

3.5 Research Strategy

This research uses the experiment strategy as a primary strategy approach, and some mixed research procedures and strategy methods are planned to be executed in this research.

Conduct a well-clear literature review process; This will provide a solid foundation for understanding the current state of knowledge in the field, identifying research gaps, and informing the development of the proposed AI-based system.

Conduct a field study to collect data on human-elephant conflicts, including the frequency, severity, and causes of disputes and the effectiveness of existing mitigation methods such as electric fences.

Analyze the collected data, including the number of elephant detections, incidents prevented, and other relevant variables, to assess the impact of the AI-based system on mitigating human-elephant conflicts. This could involve statistical analysis, data visualization, and interpretation of results to draw conclusions and make recommendations for further improvements.

Develop the AI model; after the requirement gathering and information are finished, the next step is to develop the AI-based system for detecting elephants in video frames captured by CCTV/wildlife camera traps. This would involve developing computer vision algorithms to accurately detect elephants, integrating the system with artificial sound effects, designing a mechanism for sending early warning messages to villages, and storing data in a database for further analysis; this process involves many experiments and trial and error.

Collaborate with local communities, wildlife authorities, and other stakeholders involved in human-elephant conflict management in Sri Lanka to ensure that the developed AI-based system aligns with their needs and priorities.

Prepare research reports, publications, and presentations to disseminate the study findings among the scientific community, policymakers, and practitioners. This could contribute to the knowledge base on human-elephant conflicts and technological solutions and help advocate adopting the AI-based system as a sustainable solution for mitigating human-elephant conflicts in Sri Lanka.

3.6 Project Design

3.6.1 Project Description

In Sri Lanka, there is no solid solution for this problem, and this project tries to implement artificial intelligence technology-based solutions for human-elephant conflicts. There are many similar technological solutions, but only the electric fence method is in operation; in Sri Lanka, many innovators are building IoT-based technologies for this problem; however, this research introduces artificial intelligence technology and tries to automate a manual process by machines.

This research tries to automate the following problem statement in some rural parts of Sri Lanka; Usually, in villages in Sri Lanka, primarily a farmer is assigned to the task of watching for elephants, especially in the nighttime; that watchman needs to watch for elephants if the elephant comes, that watchman needs to report to the nearest villages as an early warning alert. This method is done in Sri Lanka manually in rural areas to avoid human-elephant conflicts.

This research is willing to replace this process with artificial intelligence technology that implements an AI technology that could do the same with machines without manpower in an effective way. It can collect elephant data and store it in the database, and authorities use it manually for future decision-making. This research aims to build a computerized system to automate the watchman's work to identify elephants and alert the villagers to take proactive actions to prevent damage.

The research output will be a computer-vision-based artificial intelligence software program that can see the elephant with image processing technology and send an early warning message to the nearest villages to be alert. This system will block the elephants from entering the villages by playing the artificial sounds of a buzzing bee and firecrackers since the elephants fear them. This groundbreaking research aims to revolutionize how human-elephant conflicts are managed in rural Sri Lanka by harnessing the power of artificial intelligence; this project seeks to automate the manual process of watching for elephants and issuing early warnings to villagers to be alerted.

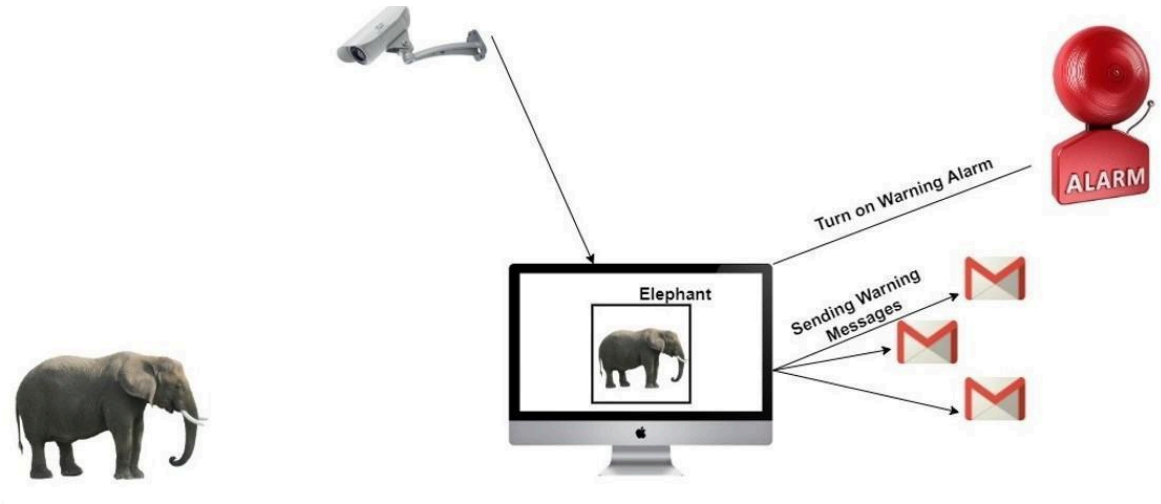


Figure 1 System Description Image

Technically, this system has two parts: a **monitoring system** and a **management panel**. The primary tasks of the monitoring system are **looking for elephants** by processing the vision from a camera source and **sending early warning messages** to the corresponding villages.

The primary tasks of the management panel are storing information on elephant detection and having the ability to set settings for the monitoring system example, the password can be changed from the management panel, and able to create more monitoring system accounts from the management panel; When creating monitoring system, it is called like “Create a New Device” in the management panel.

There are two detection methods used in the monitoring system using those are with motion detection method and with complete processing method; entire processing method is computationally expensive since it processes all the frames from the video source, and the second method is motion detection based, which can only process vision that has movement if any significant changes happen in the screen. It will send the frames for model processing; these settings and parameters can be set from the management panel.

The ultimate **scope** of the project is to explore a solution for the human-elephant conflicts with technology; this research is willing to build a computerized system that can send early warning messages to corresponding villages; this project tries to automate the manual work of watching for elephants without manpower but with machine power.

3.6.2 Project Requirements

The fundamental requirement for this research comes from the problem Sri Lanka and similar countries face in human-elephant conflicts. Therefore, the main requirements are clearly defined in this section.

The problem can be easily understood. This research tries to help rural area farmers and villagers by providing an artificial intelligence-based computerized system to solve human-elephant conflicts.

As a primary requirement, this research is trying to build an artificial intelligence-based Human Elephant Conflicts Mitigation system called “HECMS” This software contains two different parts, which are a “**Monitoring System**” and a “**Management System.**” This requirement section is divided into two parts to describe the requirements of each module in a separate section.

- **Monitoring System Requirements**
    - Build AI models that accurately detect elephants.
    - Must have authentication (login) screen before entering the main screen.
    - Must have an error box when the wrong credential is provided.
    - Able to send early warning SMS to the village authority’s phone number.
    - Able to send early warning Emails to conservationists (Conflicts Management Team Authority).
    - Able to play artificial sound effects of buzzing bees and firecrackers.
    - Able to play warning sounds to alert the nearest surroundings.
    - Able to update elephant conflict threat incidents in the database with time, date, and the number of elephants.
    - Able to update the conflict threat incidents images in the database.
    - Able to process video from an IP camera.
    - Able to process video from the device camera.
    - Able to process videos from external pre-recorded videos.
    - Able to adjust the detection threshold rate (even during the processing).
    - Able to turn off/on alarm (even during the processing).

- Able to change alarm sound effects (even during the processing).
- Able to change AI model architecture.
- Able to adjust the brightness of the video source (even during the processing).
- Able to adjust the contrast of the video source (even during the processing).
- Able to adjust the blurriness of the video source (even during the processing).
- Able to take snapshots (only during the processing).

- **Management System Requirements**
  - Must have two user account types: Super Admin and Device Admin.
  - Must have an authentication (login) feature.
  - Must have an error box when the wrong credential is provided.
  - **Super Admin**
    - Must have a beautiful dashboard.
    - Must have the following pages: Homepage, Data Management Page, Device Management Page, and Settings Page.
    - Must have a logout option.
    - Must mention “Super Admin” on the sidebar menu.
    - **Homepage (Dashboard)**
      - Must have a line chart that plots all conflict incidents traffic.
      - Must have a feature to filter the line chart month-wise.
      - Must show the total number of devices.
      - Must show the total number of elephant conflict incidents by all devices.
      - Must show the total number of captured elephants by all devices.
    - **Data Management Page**
      - Must show all conflict incidents data captured by all devices in a table (Device ID, Date, Time, Number of Elephants, and Image).
      - Must have a pagination pane (A single page must only have 15 data).
      - Must have a day-wise filter to list by a specific date.

        - Must have a feature to view the incident image on a new window.
    - **Device Management Page**
        - Must show all the device data in a table (Device ID, Device Name, Latitude, Longitude, authority email, authority phone, username, and password).
        - Must have the option to create a new device.
        - Must have a pagination pane (A single page must only have 15 data).
        - Must have the option to delete device data. (Deleting device data must also delete the associated device account credentials).
        - Before deleting a device's data, it should ask for confirmation.
        - Must have the option to edit device data.
        - Must have the option to view the device location on Google Maps.
    - **Settings Page**
        - Must have the option to change the password of Super Admin.
- **Device Admin**
    - Must have a beautiful dashboard.
    - Must have the following pages: Homepage, Data Management Page, Device Preferences Page, and Settings Page.
    - Must have a logout option.
    - Must mention "Device Admin" on the sidebar menu.
    - Must mention device info on the sidebar menu (Device Name, Device ID).
    - **Homepage (Dashboard)**
        - Must have a line chart that plots conflict incidents traffic of the current device.
        - Must have a feature to filter the line chart month-wise.
        - Must have a "View on Map" button to view the current device location.

        - Must show the total number of elephant conflict incidents by the current device.
        - Must show the total number of captured elephants by the current device.
    - **Data Management Page**
        - Must show conflict incidents data captured by the current device in a table (Device ID, Date, Time, Number of Elephants, and Image).
        - Must have a pagination pane (A single page must only have 15 data).
        - Must have a day-wise filter to list by a specific date.
        - Must have a feature to view the incident image on a new window.
    - **Device Preferences Page**
        - Must show the current device information (Device ID, Device Name, Latitude, Longitude, authority email, authority phone).
        - Must have the option to edit device data.
    - **Settings Page**
        - Must have the option to change the password of the current device.
- **Security Requirements for Management System**
    - Must have proper session handling for authentication (login).
    - Must validate forms on the server side as well as on the client side.
    - Must have proper routes with (GET/POST).
    - Must check session type before providing access to any pages.
    - Must have basic security protections.
    - Must have proper session destroyed when logging out.

3.6.3 Project Architecture

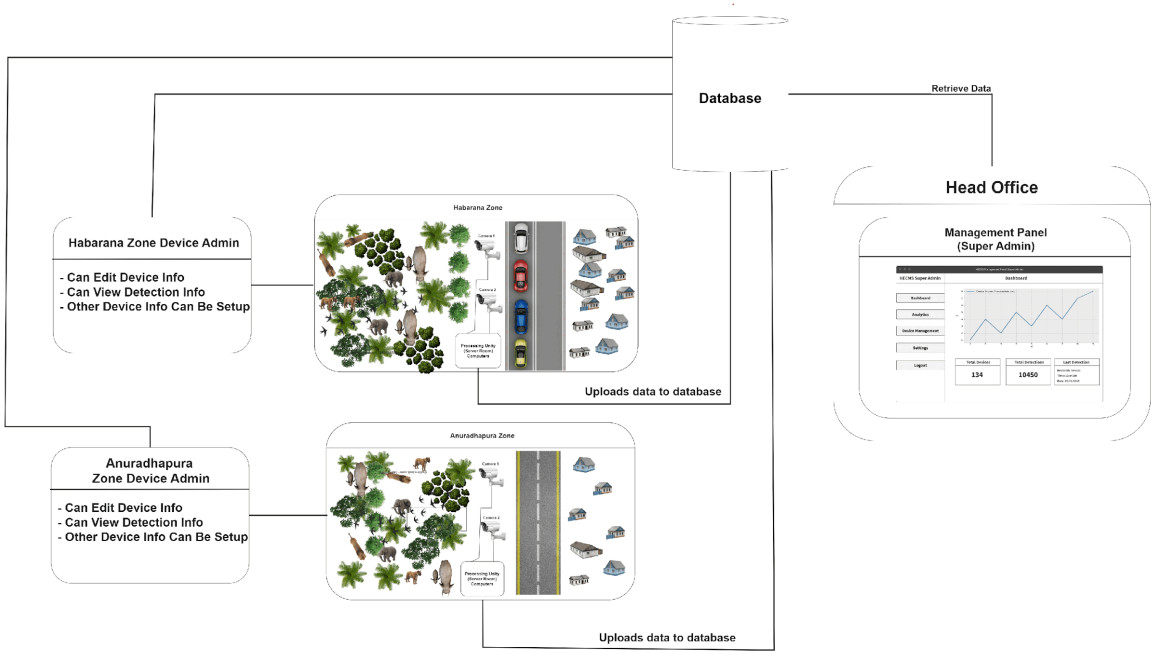


Figure 2 Graphical Architecture Diagram

The project's graphical architecture is built based on the collected requirements. It visually defines how the Human Element Conflict Mitigation System (HECMS) functions; it segments the different pieces of functions in a more illustrative manner.

There is a "Head Office" on the right side of the diagram, which means they will use the **Management System**, and all the conflict incidents data will be transferred to it and can be analyzed and accessed by the Head Office.

On the left side, there are two zones presented as examples, "Habarana Zone" and "Anuradhapura Zone" (Popular elephant conflict places in Sri Lanka), and each zone has a device admin. That device admin has basic access to the **Management System** through the **Device Admin** account. Each zone has a **Monitoring System** installed on a computing device that processes video frames to check elephant presence. If elephants are detected, it will send an SMS to the corresponding villager and an email to "Head Office," The incident data will be transmitted to Management System. Those can be accessed by "Head Office" and that device admin.

In the middle, the database is shared between Management and Monitoring systems to proceed with the granted operations.

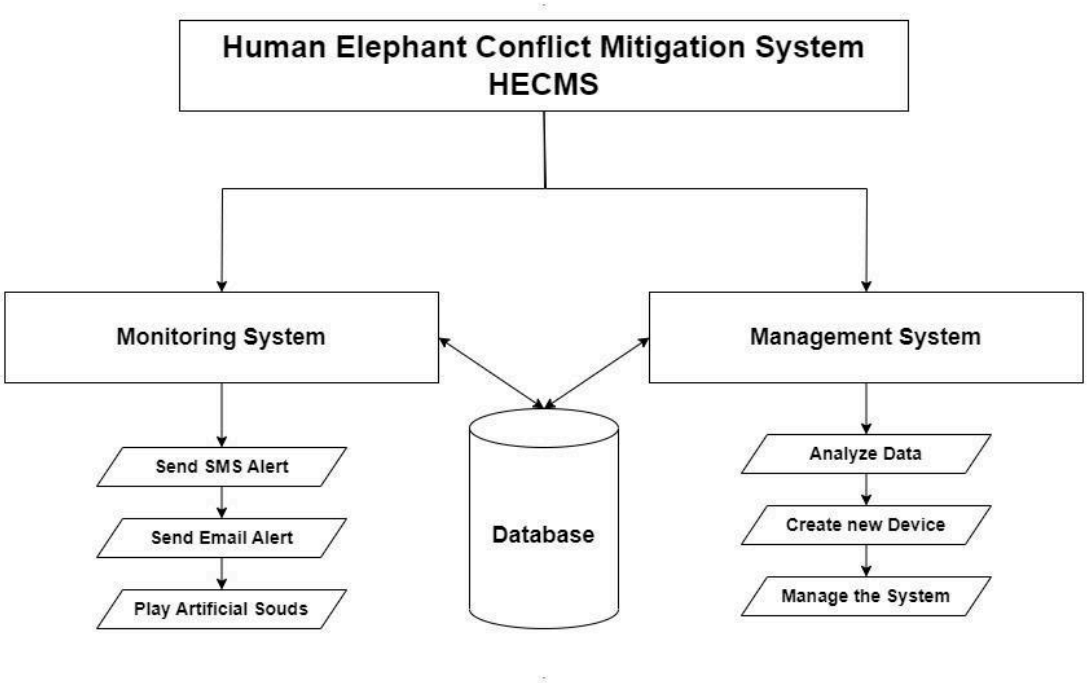


Figure 3 Structured-based Architecture Diagram

The above structured-based architecture diagram clearly explains how the Human Element Conflict Mitigation System (HECMS) functions structurally. HECMS is divided into two main sub-systems: the Monitoring System and the Management System.

Both systems have a shared database in the middle, and their main functionalities are also clearly mentioned; the monitoring system's primary functions are Elephant detection, SMS sending, Email Sending, artificial sound playing, and database updating, and the management system's functions are analyzed data, manage data, and create new devices, etc.

Both systems function together to achieve the goal of this research, and it seeks to minimize human-elephant conflicts by proactively detecting and managing elephant movements in conflict-prone areas. The close integration between the Monitoring and Management systems allows for a dynamic and adaptive approach to human-elephant conflict mitigation. The real-time information provided by the Monitoring System enables swift and informed decision-making by the Management System, leading to timely and targeted interventions to prevent conflicts or mitigate their impacts.

3.6.4 Database Architecture (ER Diagram)

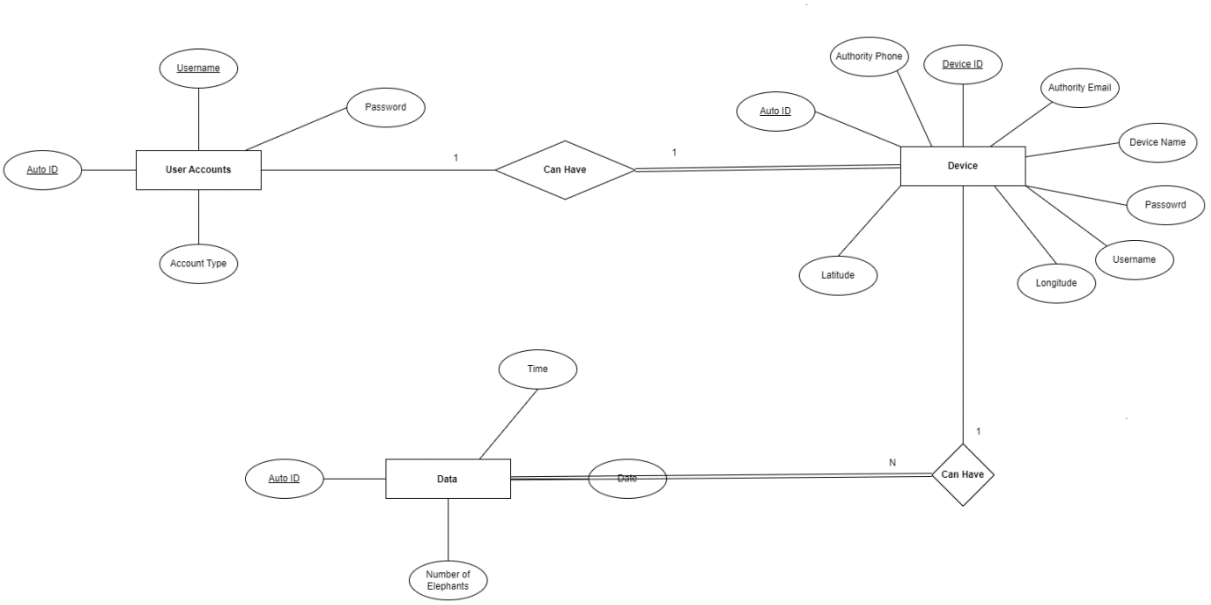


Figure 4 Database Structure (ER-Diagram)

The above ER (Entity-Relationship) diagram explains the structure of the database system in a graphical way; it is the visual representation of the database functions, attributes, and relationships between tables.

The above diagram clearly defines three main tables: Device, Data, and User Accounts. The Data table contains detection data, the Device table includes the monitoring system's device data, and the User Account table contains the user account details. This is the shared database utilized by both systems Monitoring and Management systems.

The relationships between the tables are shown above; for example, Data and Device have "One to Many" relationships, which means that the data table can have multiple devices, but the device table does not. Likewise, the relationship between the User Account table and Device table is "One to One, " meaning a particular device can be found once in the User Account table.

Regarding partial and full dependency, between Data and Device is partial dependency; all Data should depend on a device's data. But, all device data should not rely on all data detection data. Likewise, between User Account and Device is a partial dependency because all device data should depend on User Account Data. Still, all User Account data should not depend on all device data example: the super admin account does not need a device.

3.6.5 System Structure (UML Class Diagram)

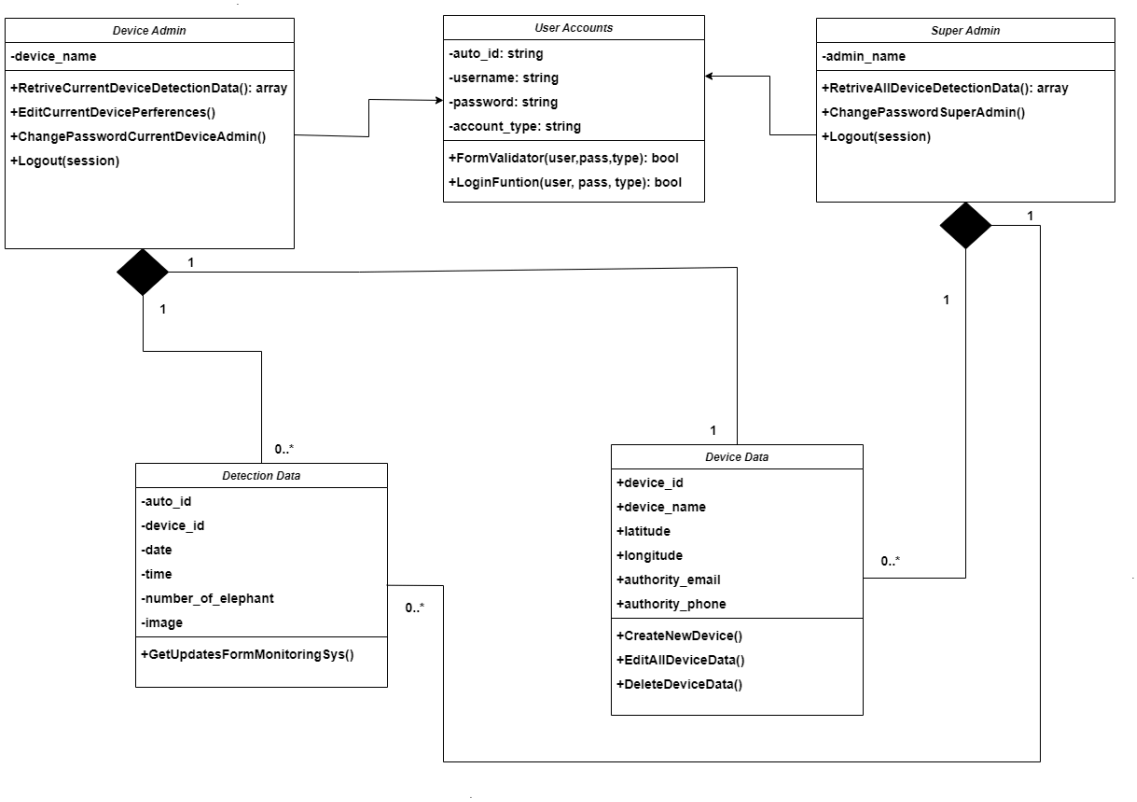


Figure 5 System Structure (UML Class Diagram)

This is the project system structure of the Human-Elephant Conflicts Mitigation System HECMS. It is a graphical representation that illustrates the system structure and relationships of classes in the HECMS system; This UML diagram explains the Object Oriented Concept that is used in this project, how those classes are used and what is the structure of that program paradiam and etc.

Five classes make up the whole system functioning smoothly: the User Account Class, Device Admin Class, Super Admin Class, Detection Data Class, and Device Data Class. The User Account Class has two main functions those FormValidator and LoginFunction; first, it runs the FormValidator function with the username, password, and user_account_type arguments to validate the credentials using the form validator function, then it sends the credentials to LoginFunction that returns a bool (True or False) if the certification is correct, it returns True otherwise, it returns False, based on this Boolean value, the system decides to open the dashboard or not.

The User Account has entities such as auto_id (auto-generated ID), username:string, password:string, and user_account_type:string; these are called attributes.

The Device Admin and Super Admin classes are derived from the User Account Class this means that the Device Admin and the Super Admin will have the same functions and attributes of the User Account class, from the Super Admin and Device Admin can call functions and features that are in the User Account class .

Super Admin has admin_name as an extra entity, and it has three primary functions are RetrieveAllDeviceDetectionData, ChangePasswordSuperAdmin, and Logout. The RetrieveAllDeviceDetectionData function is used to fetech data to the dashboard to the super admin account once the login is done. The ChangePasswordSuperAdmin function is used to change the credentials of the super admin account.

The Logout function takes one parameter, which is session; when logout the system, that function will destroy the session associated with the session layer, and it provides a proper session handling mechanism to the system.

The Device Admin class is similar to Super Admin, but it is limited; it has functions such as RetriveCurrentDeviceData, EditCurrentDeviceData, ChangeCurrentDevicePassword, Logout functions, RetriveCurrentDeviceData gets data of the current device in an array, and EditCurrenDeviceData has functions to edit the drive preferences and ChangeCurrentDevicePassword will allow to change device password.

Device Data class has attributes such as device_id, device_name, latitude, longitude, authority_email, and authoriy_phone and functions like EditDevice, CreateNewDevice, and DeleteDevic; thesee parts are public those can be accessed from the SuperAdmin side and can perform CURD operation on that class.

Detection Data class hold attributes such as auto_id, device_id, date, time and image of the incident, and a function called GetUpdatesFromMonitoringSys, which is used to clean up data from the monitoring system and allow the database to place correctly.

This UML diagram clearly defines how the Human Elephant Conflicts Mitigation System HECMS functions at the class level and what classes are co-related. It provides a comprehensive understanding of the whole system.

3.6.6 System Interactions (UML Use Case Diagram)

Figure 6 System Interactions (UML Use Case Diagram)

The above Use Case Diagram is a Unified Modeling Language (UML) behavioral diagram that represents the Elephant Conflict Mitigation System's interactions between users (Super Admin and Device Admin). The Use Case Diagram can illustrate the system's different use cases or functionalities and the users who interact with the system.

According to the above diagram, there are two actors: Super Admin and Device Admin; in the middle, there are functions that the system has; both actors use the login function. It has <<include>> and <<extend>> dependencies. When logging into a system, it first validates the form of username, password, and user_account_type for security reasons. Then after the validation is finished, the system will throw the data into the verification module, which will verify whether the user has access. Here the notable point is the validation and login verification functions are <<include>> dependencies which means that when a login takes

place, these two functions must be operated in a sequence. Still, there is a <<extend>> function that is Display Error Message; it only runs if the user is provided the wrong credentials.

Both accounts use the login function, Super Admin and Device Admin, and the session will differ based on the account type.

Creating a new device function is only accessed by the Super Admin; that function is used to create new devices admins; device admin cannot access this; only the super admin can access that function.

Delete Device Admin; super admin has another function: the super admin can delete device admin accounts; when deleting the accounts will delete all the related device admin data, and this function is only accessible by the Super Admin.

There is another function called Edit Device Admin that Super Admin and Device Admin can access; Super Admin can edit any device details, but the Device Admin can only edit its details. The device admin does not have the privileges to edit other device admin details, and it can edit its data only.

The function Update Detection Data is usually accessed by the Device Admin, especially from the Monitoring System. The primary job of this function is to transmit all the text and image data to the database.

Retrieve Detection Data is another function; its primary goal is to fetch data from the database and plot it on the dashboard. The Super Admin and Device Admin access this function, but when Device Admin access this function, it only allows fetching data sent by that current device; generally, in the device admin dashboard, it will only plot its data. But when the Super Admin access this function, it will fetch all the data sent by all the devices because Super Admin always plans all the data in the dashboard.

Super Admin and Device Admin can access the Change Password function; Device Admin can change its password, and the Super Admin can change the account password. This function is for only changing passwords of accounts.

In conclusion, this use-case UML diagram explains how the human-elephant conflict mitigation system functions and describes the system's primary functions.

3.6.7 Algorithms Designs (Flowcharts)

3.6.7.1 Login Process Flowchart

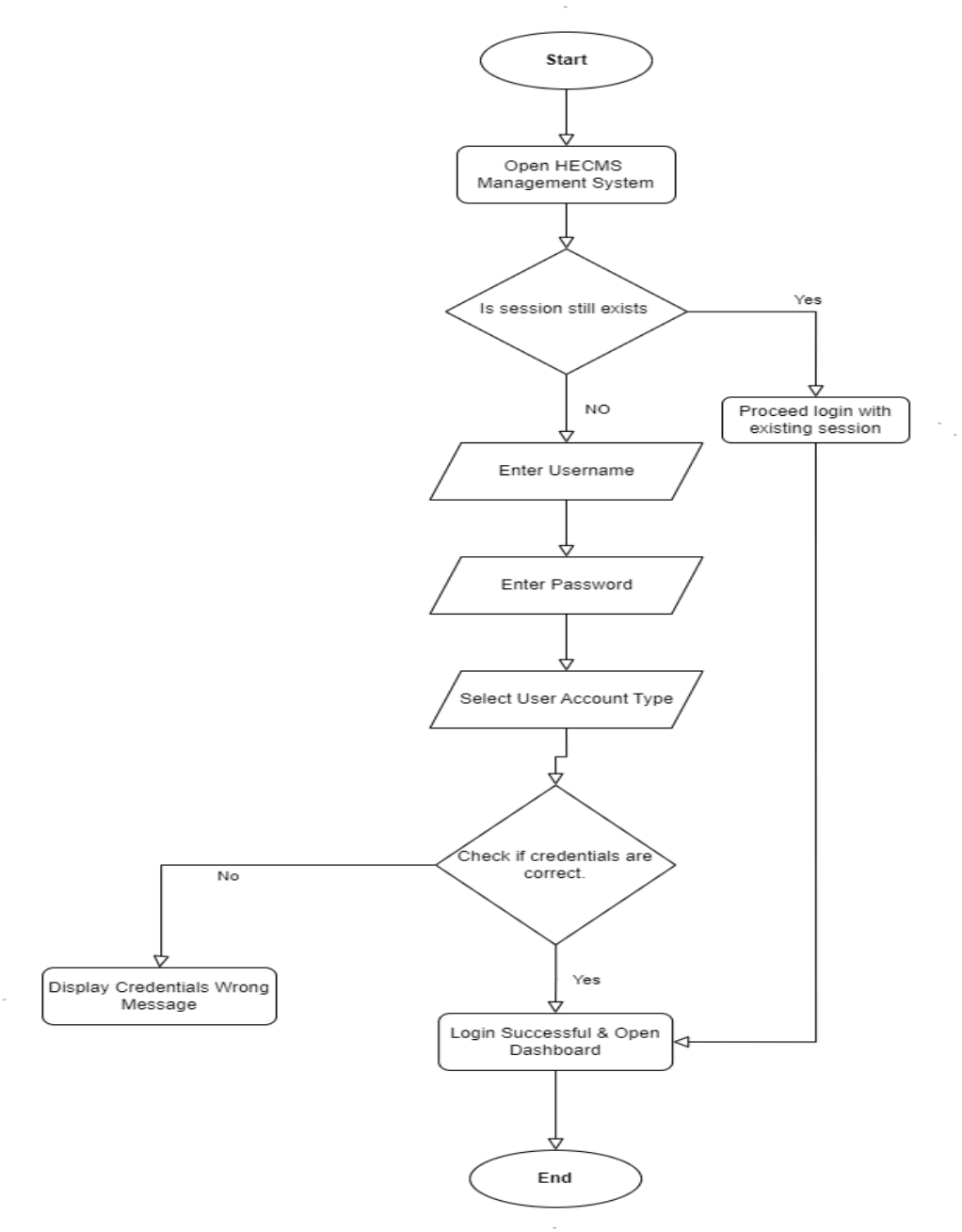


Figure 7 Login Process Flowchart

This flowchart explains how the login process happens in the HECMS system. At first, it checks the session, which means if it is already logged in, it will let the user go to the associated session's dashboard. Otherwise, it asks password and username to log in.

3.6.7.2 Elephant Detection Process Flowchart

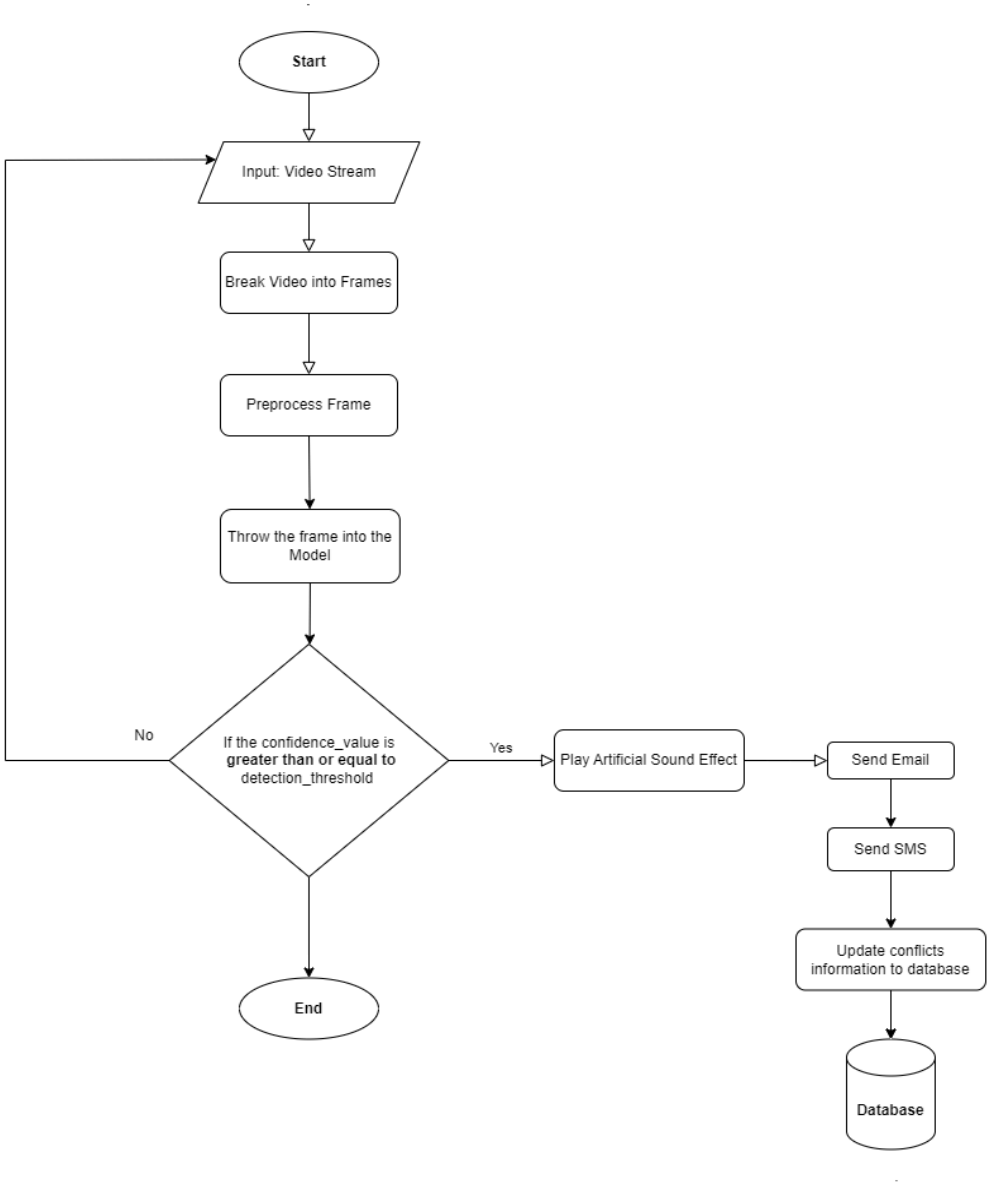


Figure 8 Elephant Detection Process Flowchart

The above flowchart explains the elephant model detection function in the HECMS project. First, it gets the video stream as the input source, then that video will be transmitted to the next layer to break the video into frames, then a frame will be sent to the preprocessing unit to clean up and then inverts the frame into a numerical vector, then the frame will be thrown to the neural network for processing, and then it outputs the confidence value if the confidence value is greater than or equal to the threshold value, it considered as a positive prediction, and then the system will play artificial sound, send SMS, email, and update database, otherwise the system processes each frame one by one in a looping manner.

3.6.8 Project Planning Phase (WBS Diagram)

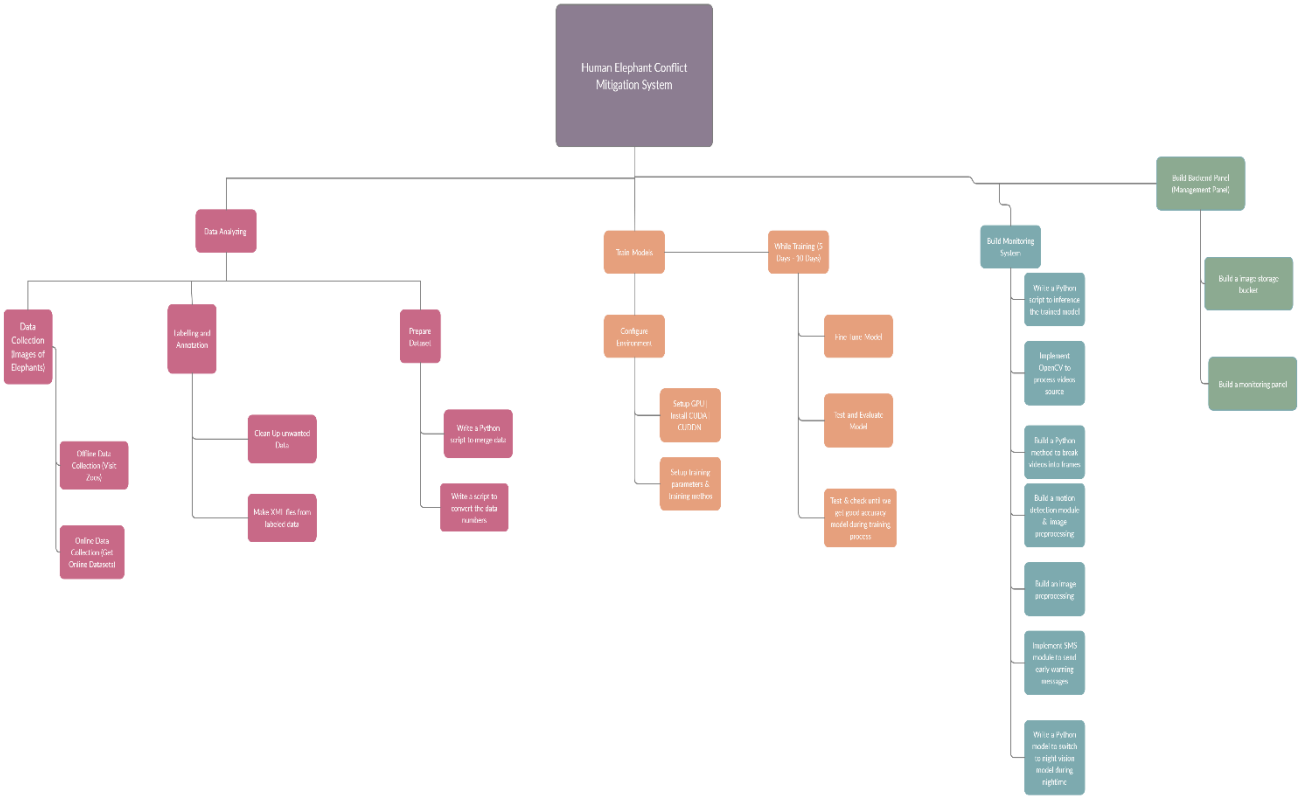
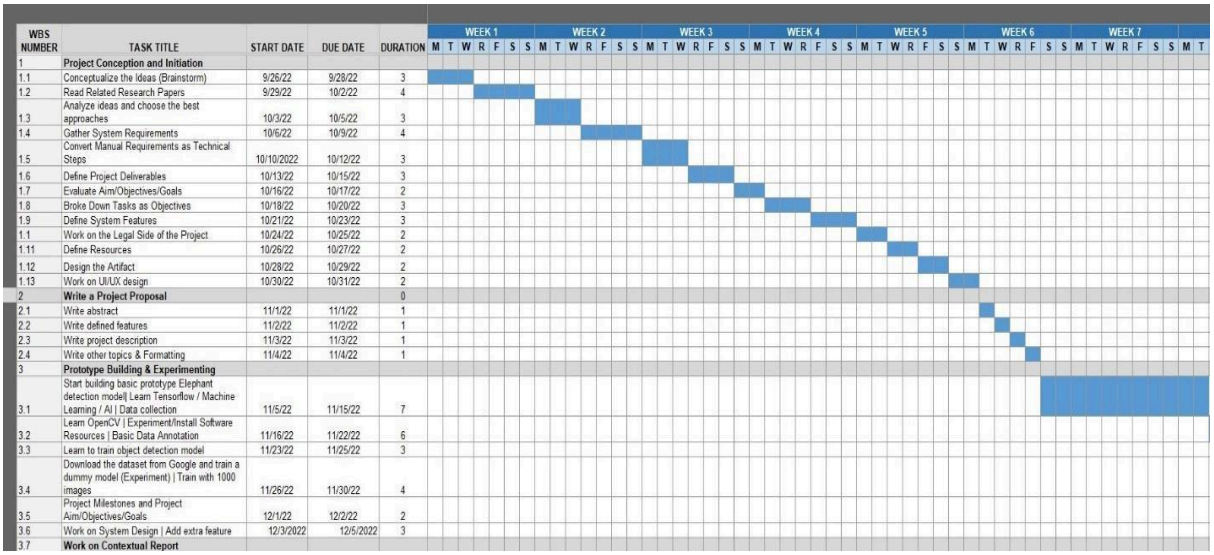


Figure Project Planning Phase (WBS Diagram)

3.6.9 Project Plan (Gantt Chart)

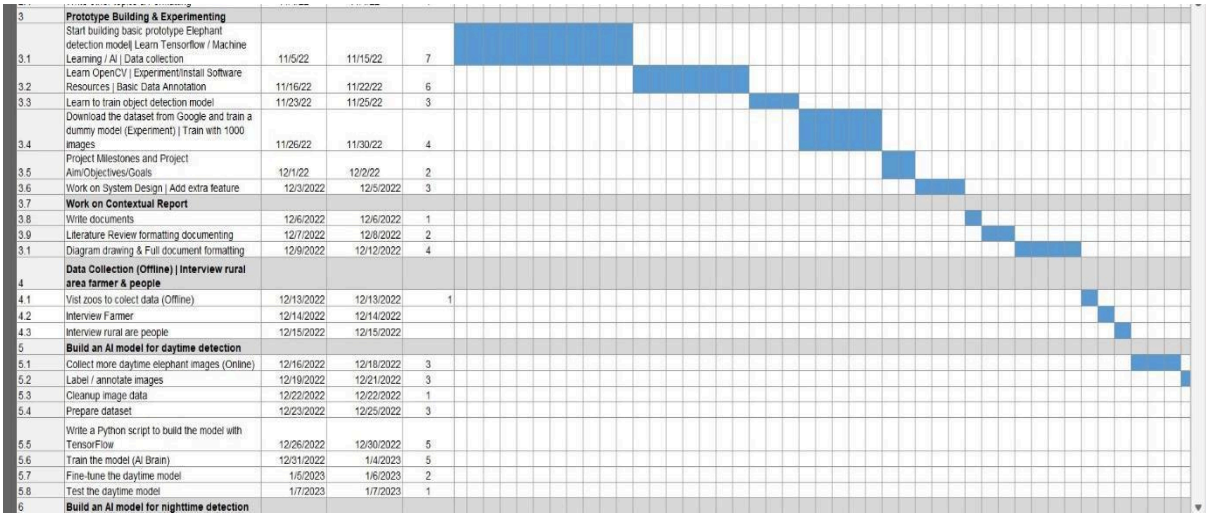
3.6.9.1 Page 1



WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION
1	**Project Conception and Initiation**			
1.1	Conceptualize the Ideas (Brainstorm)	9/26/22	9/28/22	3
1.2	Read Related Research Papers	9/29/22	10/2/22	4
1.3	Analyze ideas and choose the best approaches	10/3/22	10/5/22	3
1.4	Gather System Requirements	10/6/22	10/9/22	4
1.5	Convert Manual Requirements as Technical Steps	10/10/2022	10/12/22	3
1.6	Define Project Deliverables	10/13/22	10/15/22	3
1.7	Evaluate Aim/Objectives/Goals	10/16/22	10/17/22	2
1.8	Broke Down Tasks as Objectives	10/18/22	10/20/22	3
1.9	Define System Features	10/21/22	10/23/22	3
1.1	Work on the Legal Side of the Project	10/24/22	10/25/22	2
1.11	Define Resources	10/26/22	10/27/22	2
1.12	Design the Artifact	10/28/22	10/29/22	2
1.13	Work on UI/UX design	10/30/22	10/31/22	2
2	**Write a Project Proposal**			0
2.1	Write abstract	11/1/22	11/1/22	1
2.2	Write defined features	11/2/22	11/2/22	1
2.3	Write project description	11/3/22	11/3/22	1
2.4	Write other topics & Formatting	11/4/22	11/4/22	1
3	**Prototype Building & Experimenting**			
3.1	Start building basic prototype Elephant detection model| Learn Tensorflow / Machine Learning / AI | Data collection	11/5/22	11/15/22	7
3.2	Learn OpenCV | Experiment/Install Software Resources | Basic Data Annotation	11/16/22	11/22/22	6
3.3	Learn to train object detection model	11/23/22	11/25/22	3
3.4	Download the dataset from Google and train a dummy model (Experiment) | Train with 1000 images	11/26/22	11/30/22	4
3.5	Project Milestones and Project Aim/Objectives/Goals	12/1/22	12/2/22	2
3.6	Work on System Design | Add extra feature	12/3/2022	12/5/2022	3
3.7	**Work on Contextual Report**			

Figure 23: Gantt Chart Page 1

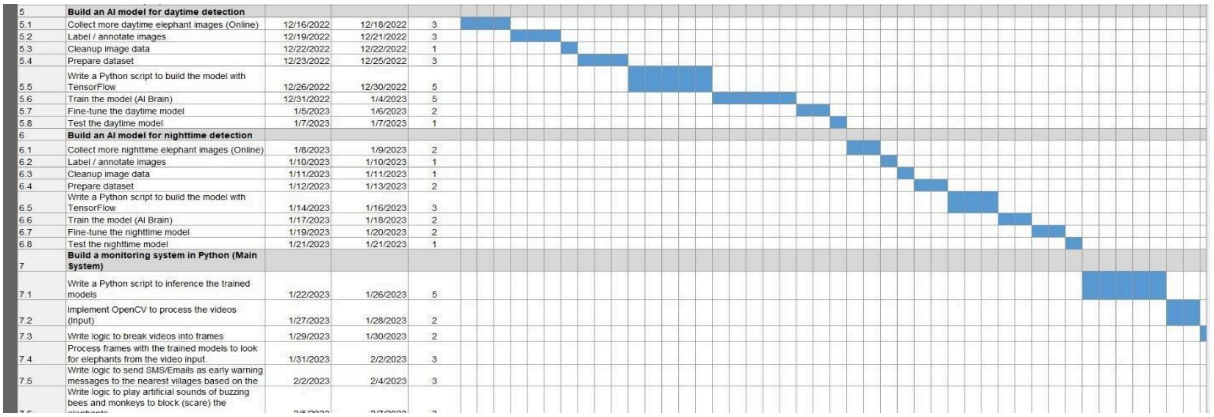
3.6.9.2 Page 2



3	**Prototype Building & Experimenting**			
3.1	Start building basic prototype Elephant detection model| Learn Tensorflow / Machine Learning / AI | Data collection	11/5/22	11/15/22	7
3.2	Learn OpenCV | Experiment/Install Software Resources | Basic Data Annotation	11/16/22	11/22/22	6
3.3	Learn to train object detection model	11/23/22	11/25/22	3
3.4	Download the dataset from Google and train a dummy model (Experiment) | Train with 1000 images	11/26/22	11/30/22	4
3.5	Project Milestones and Project Aim/Objectives/Goals	12/1/22	12/2/22	2
3.6	Work on System Design | Add extra feature	12/3/2022	12/5/2022	3
3.7	**Work on Contextual Report**			
3.8	Write documents	12/6/2022	12/6/2022	1
3.9	Literature Review formatting documenting	12/7/2022	12/8/2022	2
3.1	Diagram drawing & Full document formatting	12/9/2022	12/12/2022	4
4	**Data Collection (Offline) | Interview rural area farmer & people**			
4.1	Vist zoos to colect data (Offline)	12/13/2022	12/13/2022	1
4.2	Interview Farmer	12/14/2022	12/14/2022	
4.3	Interview rural are people	12/15/2022	12/15/2022	
5	**Build an AI model for daytime detection**			
5.1	Collect more daytime elephant images (Online)	12/16/2022	12/18/2022	3
5.2	Label / annotate images	12/19/2022	12/21/2022	3
5.3	Cleanup image data	12/22/2022	12/22/2022	1
5.4	Prepare dataset	12/23/2022	12/25/2022	3
5.5	Write a Python script to build the model with TensorFlow	12/26/2022	12/30/2022	5
5.6	Train the model (AI Brain)	12/31/2022	1/4/2023	5
5.7	Fine-tune the daytime model	1/5/2023	1/6/2023	2
5.8	Test the daytime model	1/7/2023	1/7/2023	1
6	**Build an AI model for nighttime detection**			

Figure 24: Gantt Chart Page 2

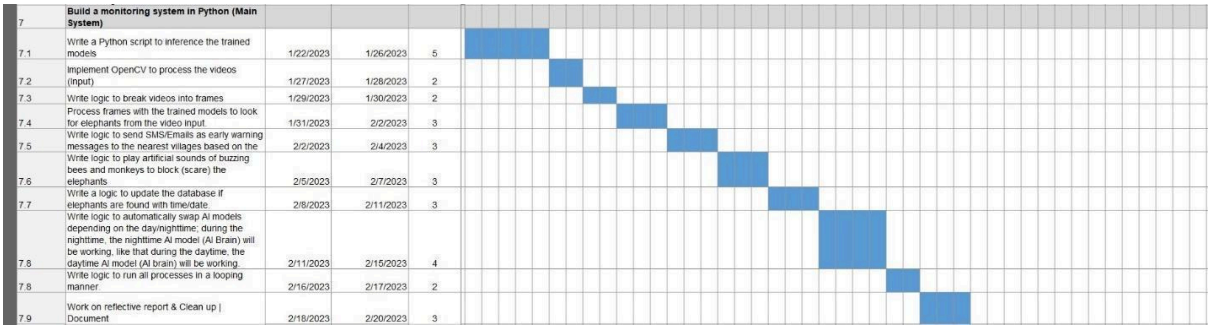
3.6.9.3 Page 3



5	**Build an AI model for daytime detection**			
5.1	Collect more daytime elephant images (Online)	12/16/2022	12/18/2022	3
5.2	Label / annotate images	12/19/2022	12/21/2022	3
5.3	Cleanup image data	12/22/2022	12/22/2022	1
5.4	Prepare dataset	12/23/2022	12/25/2022	3
5.5	Write a Python script to build the model with TensorFlow	12/26/2022	12/30/2022	5
5.6	Train the model (AI Brain)	12/31/2022	1/4/2023	5
5.7	Fine-tune the daytime model	1/5/2023	1/6/2023	2
5.8	Test the daytime model	1/7/2023	1/7/2023	1
6	**Build an AI model for nighttime detection**			
6.1	Collect more nighttime elephant images (Online)	1/8/2023	1/9/2023	2
6.2	Label / annotate images	1/10/2023	1/10/2023	1
6.3	Cleanup image data	1/11/2023	1/11/2023	1
6.4	Prepare dataset	1/12/2023	1/13/2023	2
6.5	Write a Python script to build the model with TensorFlow	1/14/2023	1/16/2023	3
6.6	Train the model (AI Brain)	1/17/2023	1/18/2023	2
6.7	Fine-tune the nighttime model	1/19/2023	1/20/2023	2
6.8	Test the nighttime model	1/21/2023	1/21/2023	1
7	**Build a monitoring system in Python (Main System)**			
7.1	Write a Python script to inference the trained models	1/22/2023	1/26/2023	5
7.2	Implement OpenCV to process the videos (Input)	1/27/2023	1/28/2023	2
7.3	Write logic to break videos into frames	1/29/2023	1/30/2023	2
7.4	Process frames with the trained models to look for elephants from the video input.	1/31/2023	2/2/2023	3
7.5	Write logic to send SMS/Emails as early warning messages to the nearest villages based on the	2/2/2023	2/4/2023	3
	Write logic to play artificial sounds of buzzing bees and monkeys to block (scare) the			

Figure 25: Gantt Chart Page 3

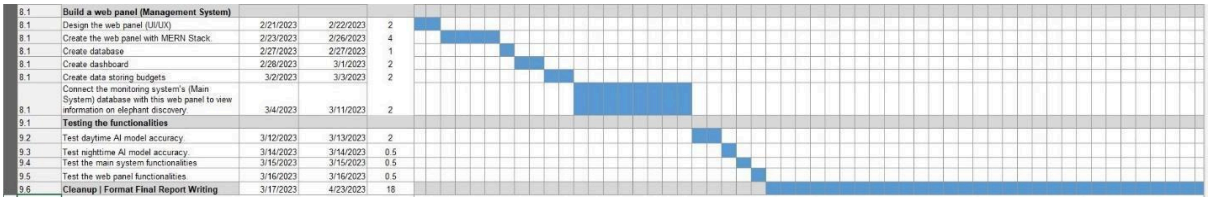
3.6.9.4 Page 4



7	**Build a monitoring system in Python (Main System)**			
7.1	Write a Python script to inference the trained models	1/22/2023	1/26/2023	5
7.2	Implement OpenCV to process the videos (Input)	1/27/2023	1/28/2023	2
7.3	Write logic to break videos into frames	1/29/2023	1/30/2023	2
7.4	Process frames with the trained models to look for elephants from the video input.	1/31/2023	2/2/2023	3
7.5	Write logic to send SMS/Emails as early warning messages to the nearest villages based on the	2/2/2023	2/4/2023	3
7.6	Write logic to play artificial sounds of buzzing bees and monkeys to block (scare) the elephants	2/5/2023	2/7/2023	3
7.7	Write a logic to update the database if elephants are found with time/date	2/8/2023	2/11/2023	3
7.8	Write logic to automatically swap AI models depending on the day/nighttime; during the nighttime, the nighttime AI model (AI Brain) will be working, like that during the daytime, the daytime AI model (AI brain) will be working.	2/11/2023	2/15/2023	4
7.8	Write logic to run all processes in a looping manner	2/16/2023	2/17/2023	2
7.9	Work on reflective report & Clean up | Document	2/18/2023	2/20/2023	3

Figure 26: Gantt Chart Page 4

3.6.9.5 Page 5



8.1	**Build a web panel (Management System)**			
8.1	Design the web panel (UI/UX)	2/21/2023	2/22/2023	2
8.1	Create the web panel with MERN Stack	2/23/2023	2/26/2023	4
8.1	Create database	2/27/2023	2/27/2023	1
8.1	Create dashboard	2/28/2023	3/1/2023	2
8.1	Create data storing budgets	3/2/2023	3/3/2023	2
8.1	Connect the monitoring system's (Main System) database with this web panel to view information on elephant discovery.	3/4/2023	3/11/2023	2
9.1	**Testing the functionalities**			
9.2	Test daytime AI model accuracy.	3/12/2023	3/13/2023	2
9.3	Test nighttime AI model accuracy.	3/14/2023	3/14/2023	0.5
9.4	Test the main system functionalities	3/15/2023	3/15/2023	0.5
9.5	Test the web panel functionalities	3/16/2023	3/16/2023	0.5
9.6	**Cleanup | Format Final Report Writing**	3/17/2023	4/23/2023	18

Figure 27: Gantt Chart Page 5

3.6.10 Project Wireframes

The wireframe is created based on the system requirements and system architecture using an online tool called “WireFramePro”, wireframes sketches are designed for Monitoring System and Management System Super Admin and Device Admin; it graphically explains the skeleton of the project nodes.

3.6.10.1 Monitoring System Login Page Wireframe

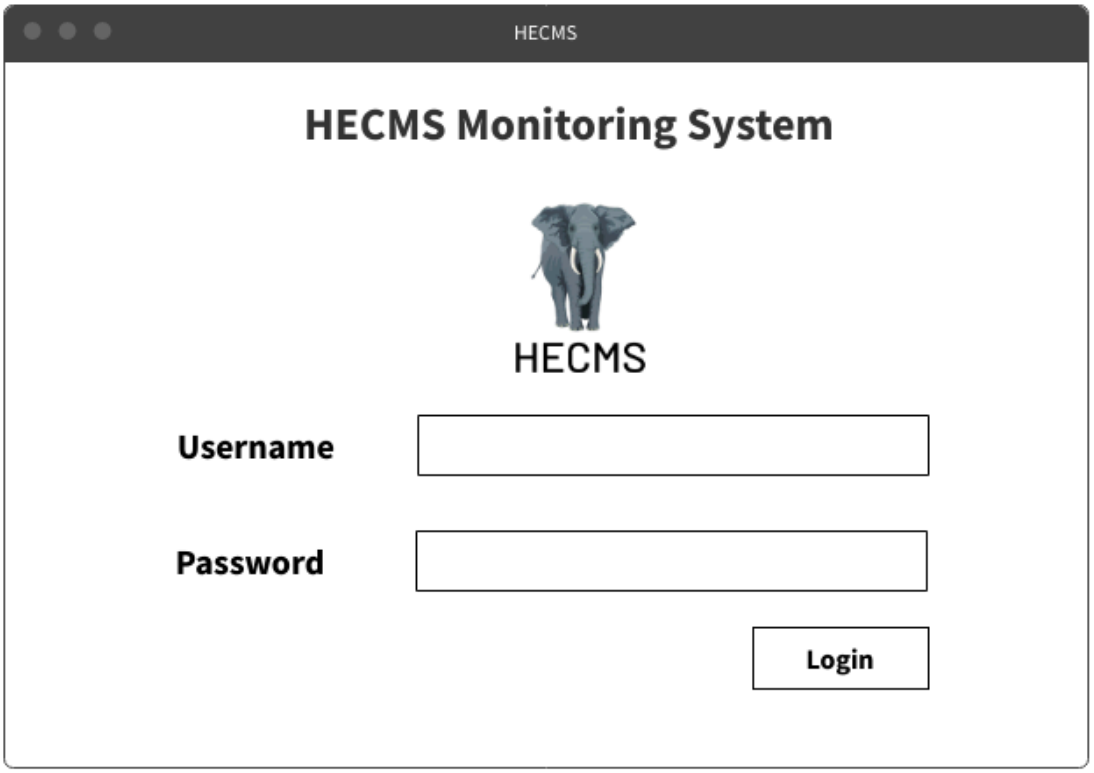


Figure 10 Monitoring System Login Page

This is the wireframe for the monitoring system login page, where the device must enter its password and username to log into the monitoring system; it has a text box for username and a password box for password.

3.6.10.2 Monitoring System Dashboard Wireframe

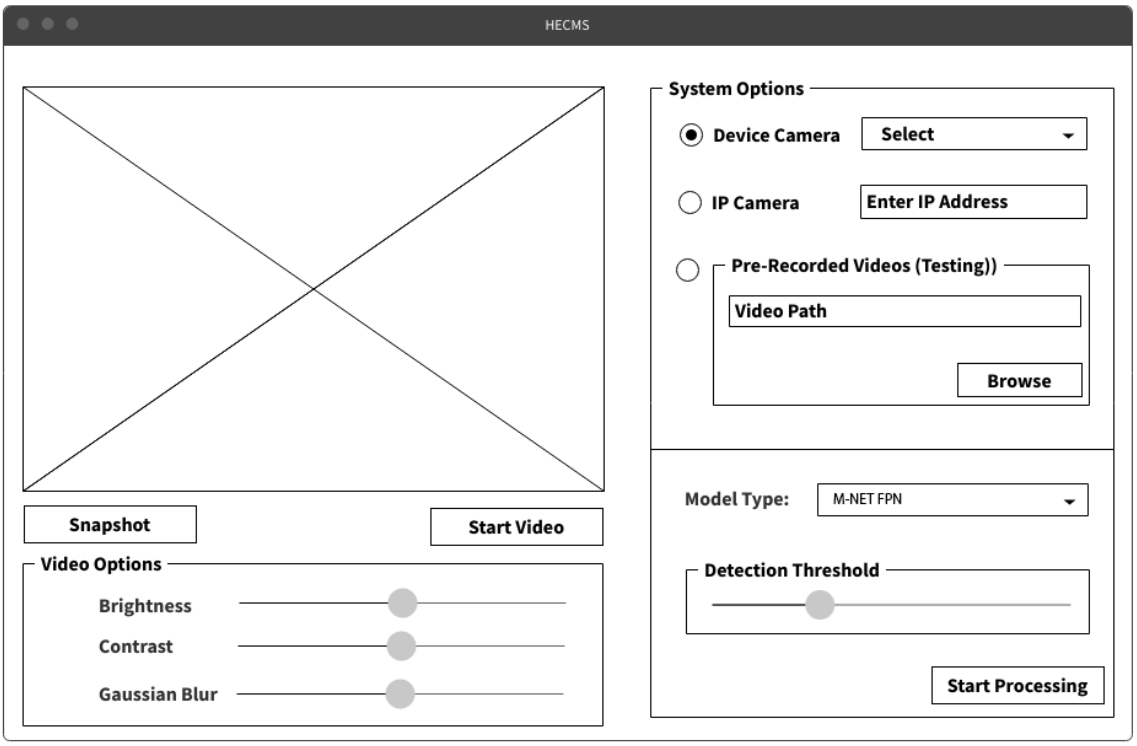


Figure 11 Monitoring System Dashboard Wireframe

All of the needed elements are placed in the wireframe of the dashboard of the monitoring system, such as a device camera selector, a textbox for the IP camera, a browse button to load the pre-recorded video, a select box for model architecture, a slider pane for detection threshold, Start Processing button is to start the processing and there many few sliders to adjust the color coordination of the frames such as brightness, contrast, and blur. There is a button for a snapshot.

These elements will be converted to the user interface and programmed as clickable elements in the relevant programming tools.

3.6.10.3 Management System Login Wireframe

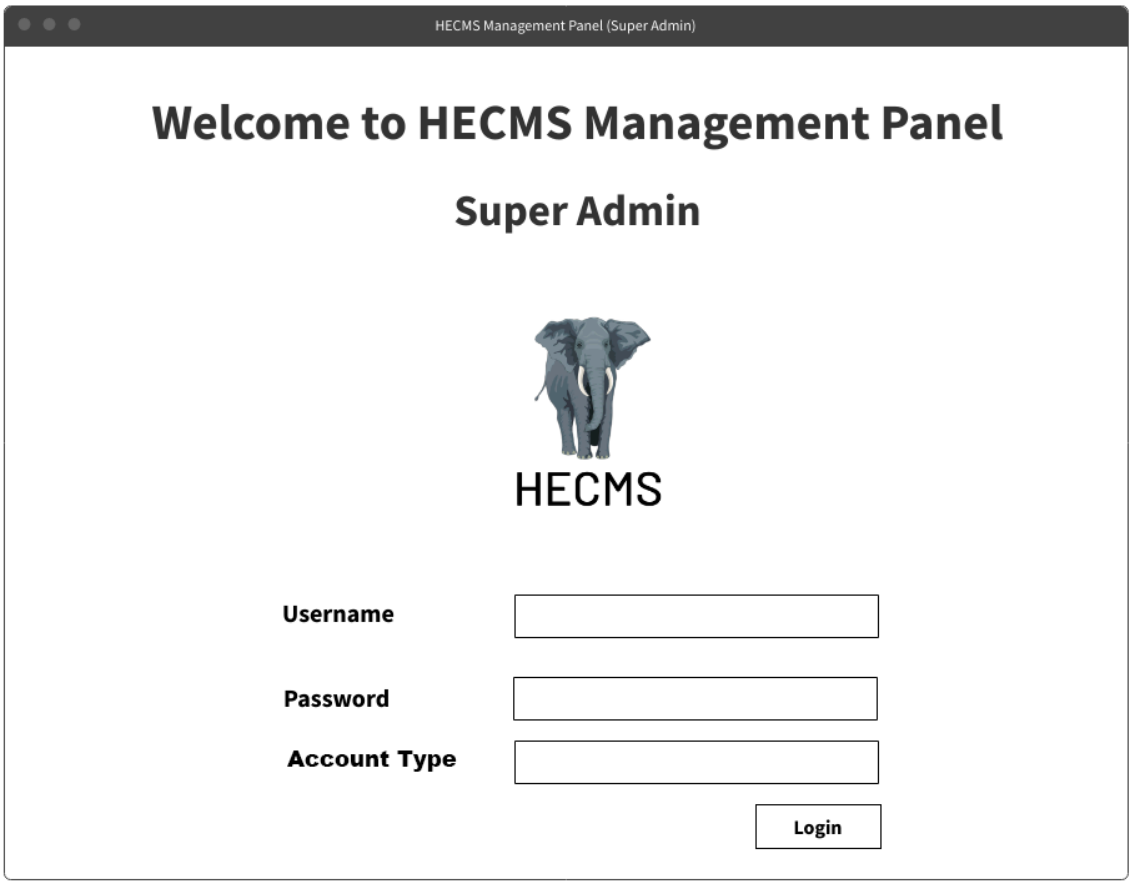


Figure 12 Management System Login Wireframe

This is the management system login page wireframe; it has three text elements for getting user inputs on username, password, and account type. There is a login button that button will be clickable in the final version. When clicking, it will check with the database if the account is present or if it has access. It will open the super admin dashboard, but now this wireframe has to be converted into the user interface; then it will be programmed as a clickable object in the appropriate programming language.

3.6.10.4 Management System Super Admin Dashboard Wireframe

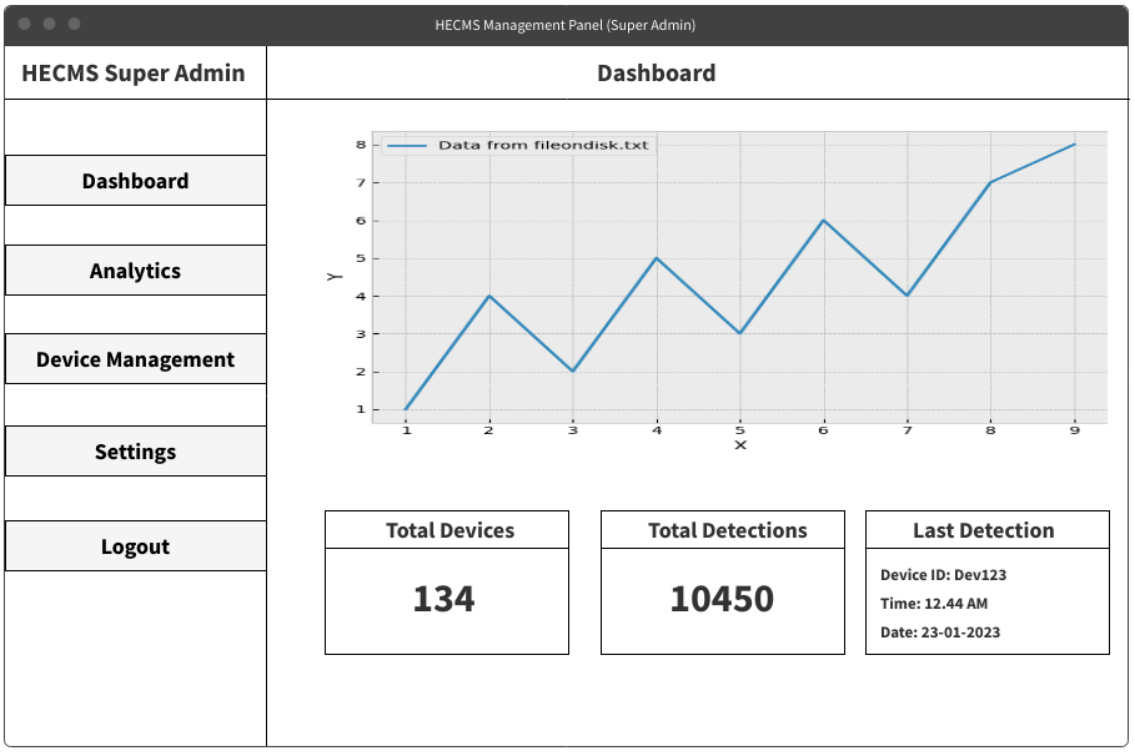


Figure 13 Management System Super Admin Dashboard Wireframe

This is the super admin management panel dashboard, which has functionalities and data visualizations; it has many elements in it; in the sidebar, it has pages like Dashboard, Data Management, Device Management, Settings, and Logout.

The dashboard has a nice-looking line chart that shows the element traffics data month-wise. That line chart will be animated with information such as total registered devices and conflicts incidents, the number of features detected, etc.; this wireframe will be converted as UI and then programmed to be a real software object.

3.6.10.5 Management System Data Management Super Admin Wireframe

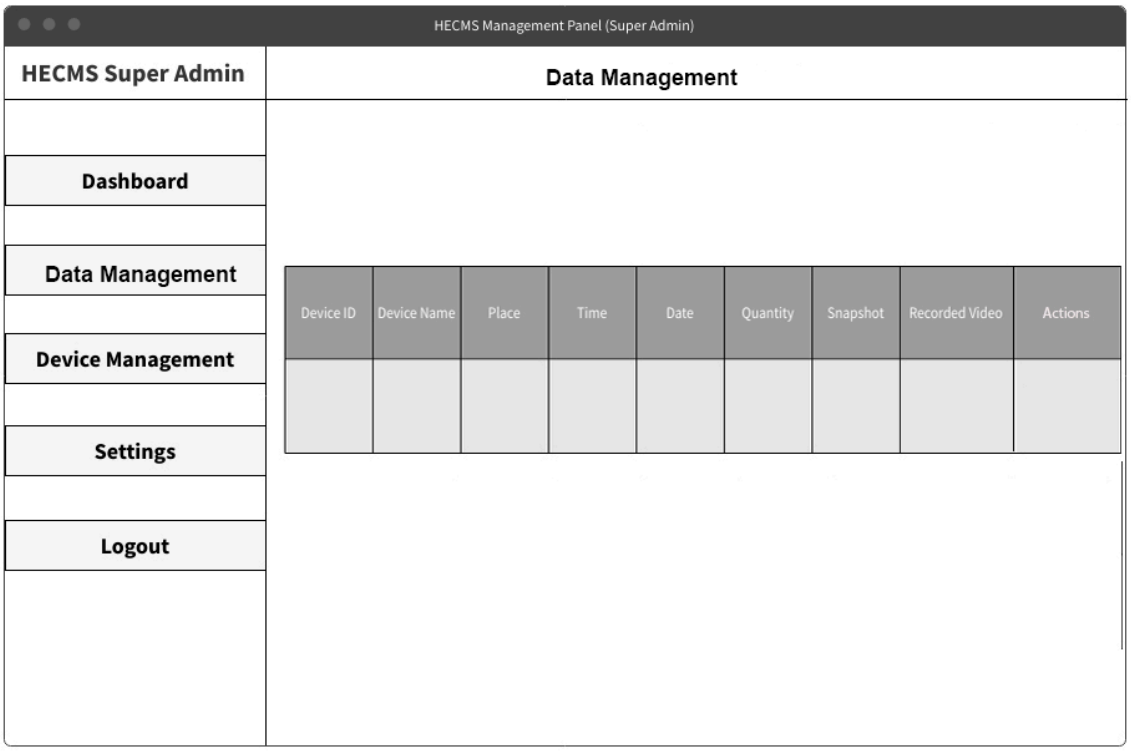


Figure 14 Management System Data Management Super Admin Wireframe

This is a super admin data management page wireframe; it contains collected data from the monitoring system such as DeviceID, Device Name, Location, Time, Date, Snapshot, etc., it stores the data in a nicely aligned table.

This wireframe will be converted as a user interface and programmed into natural software objects. This is the data management page where all human-elephant conflict data are transmitted.

3.6.10.6 Management System Device Management Super Admin Wireframe

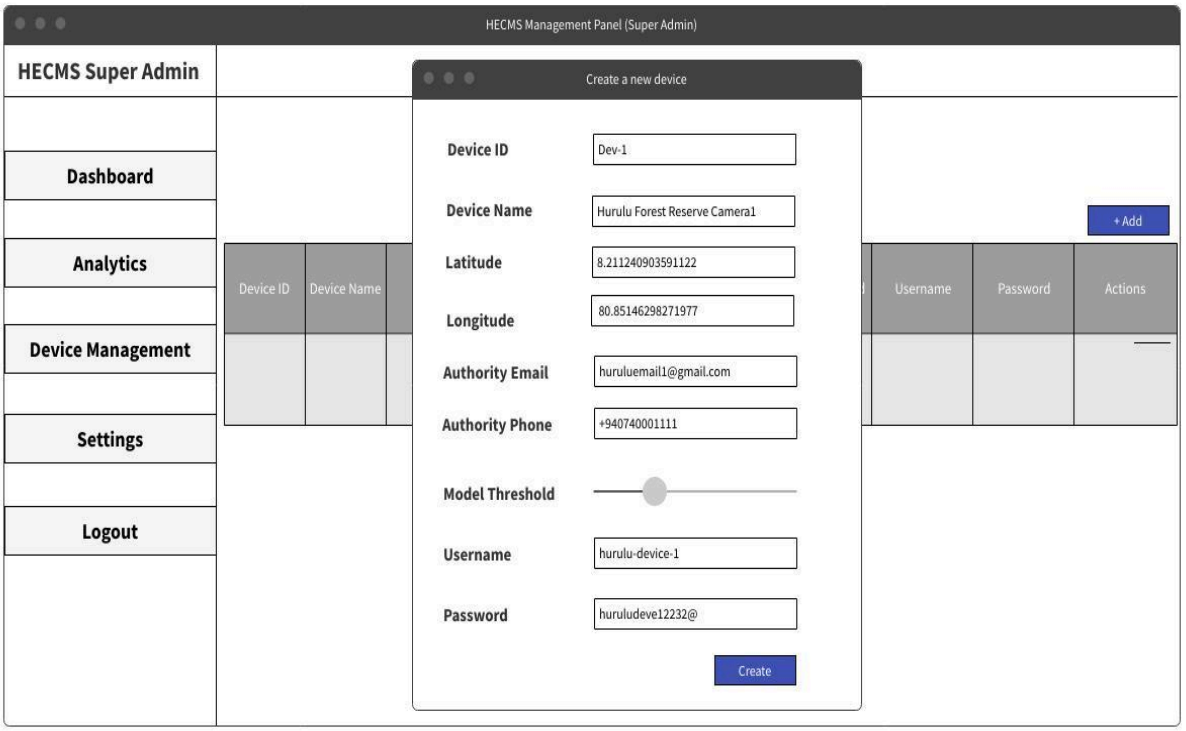


Figure 15 Management System Device Management Super Admin Wireframe

This is the wireframe of the device management page; it has many elements in it. This page shows all of the registered device's information such as Device ID, Device Name, Latitude, Longitude, Authority Email, Authority Phone Number, Model Threshold, Username, and Password, and there are some actions buttons such as delete, edit, and view on map.

Once a device is registered, that information will be shown here; then, the super admin can edit that information it is needed; for example, if the authority phone number is changed, the super admin can edit that to the new phone number. Likewise, the super admin can edit all the devices' fields.

3.6.10.7 Management System Settings Super Admin Wireframe

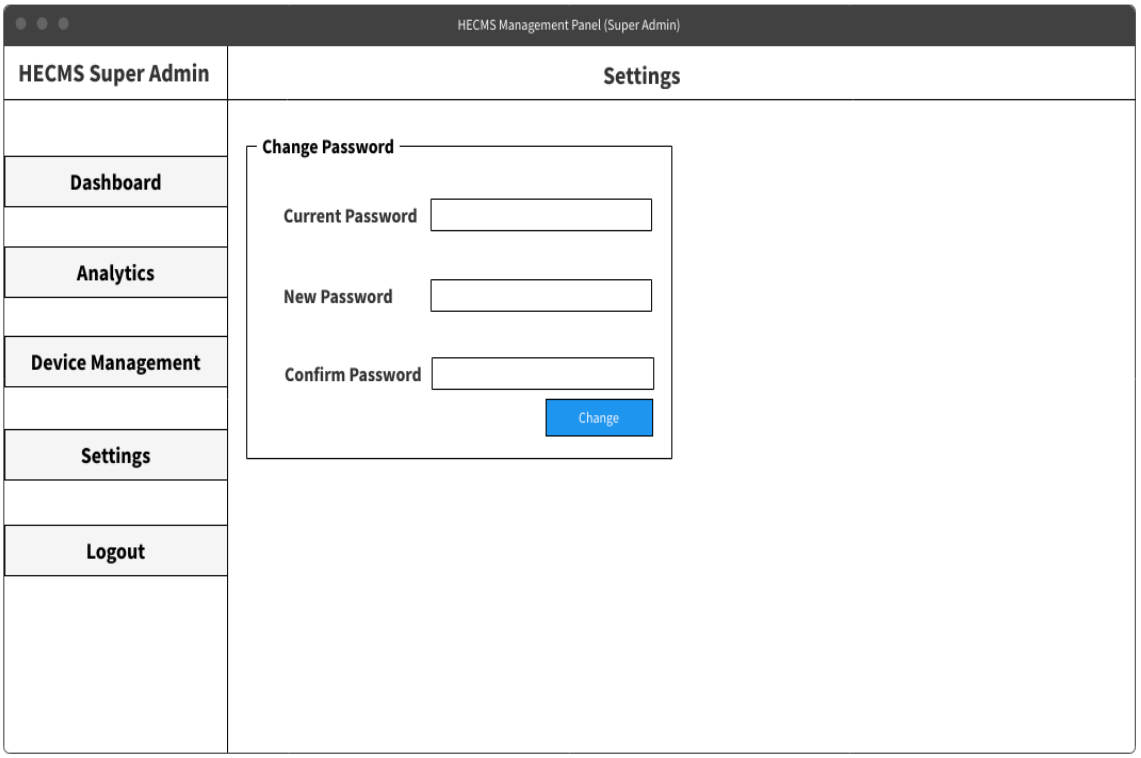


Figure 16 Management System Settings Super Admin Wireframe

This is the wireframe for the management system of the super admin; it has some fields such as current password, new password, and confirms password, so this page is used to change the credentials of the management system of the super admin.

3.6.10.8 Management System Login Page Device Admin Wireframe

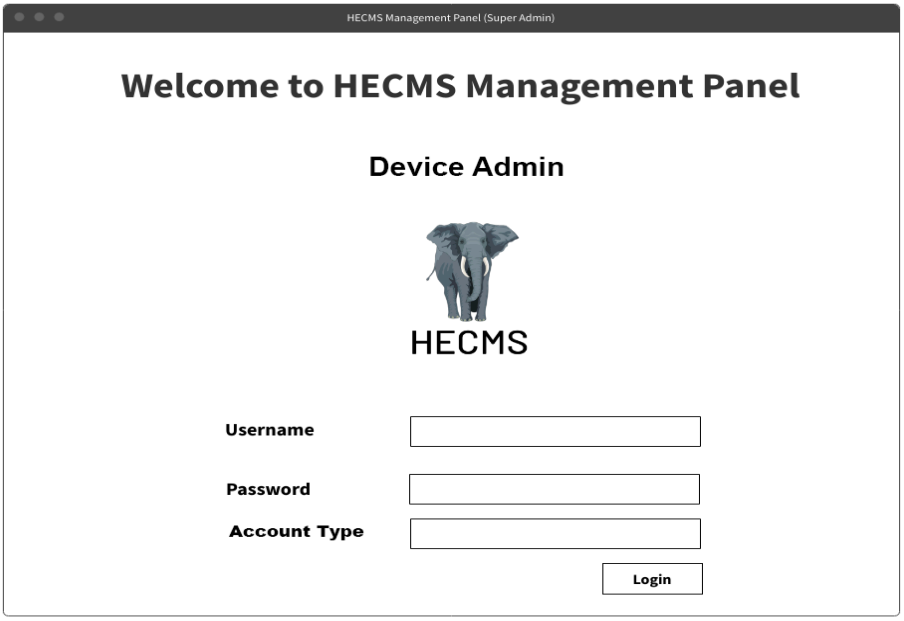


Figure 17 Management System Login Page Device Admin Wireframe

3.6.10.9 Management System Dashboard Page Device Admin Wireframe

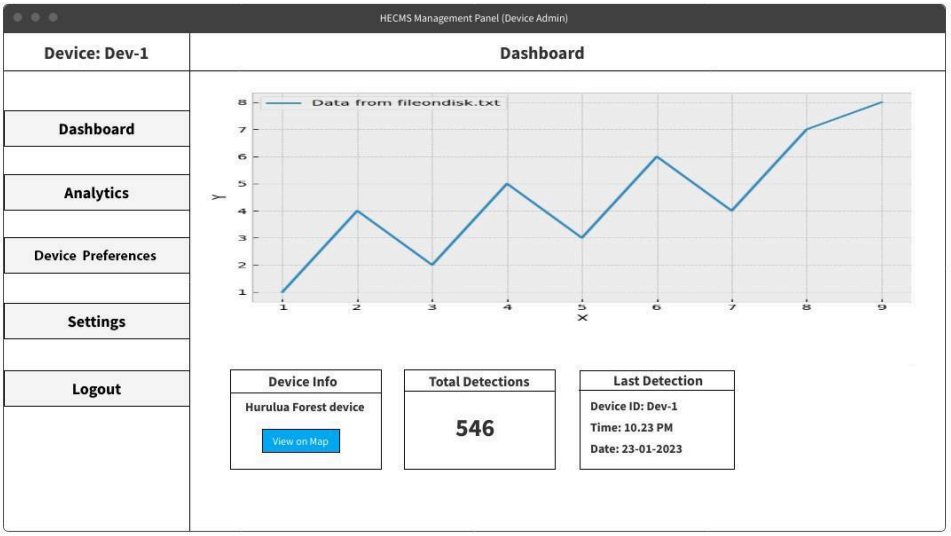


Figure 18 Management System Dashboard Page Device Admin Wireframe

This is the wireframe of the dashboard and login page, in the dashboard it will show the all of the element conflict incidents that are taken by the current devices that are logged in and it also has a button to view the location of the current device and it has total number incidents that are taken by the device.

3.6.10.10 Management System Data Management Page Device Admin Wireframe

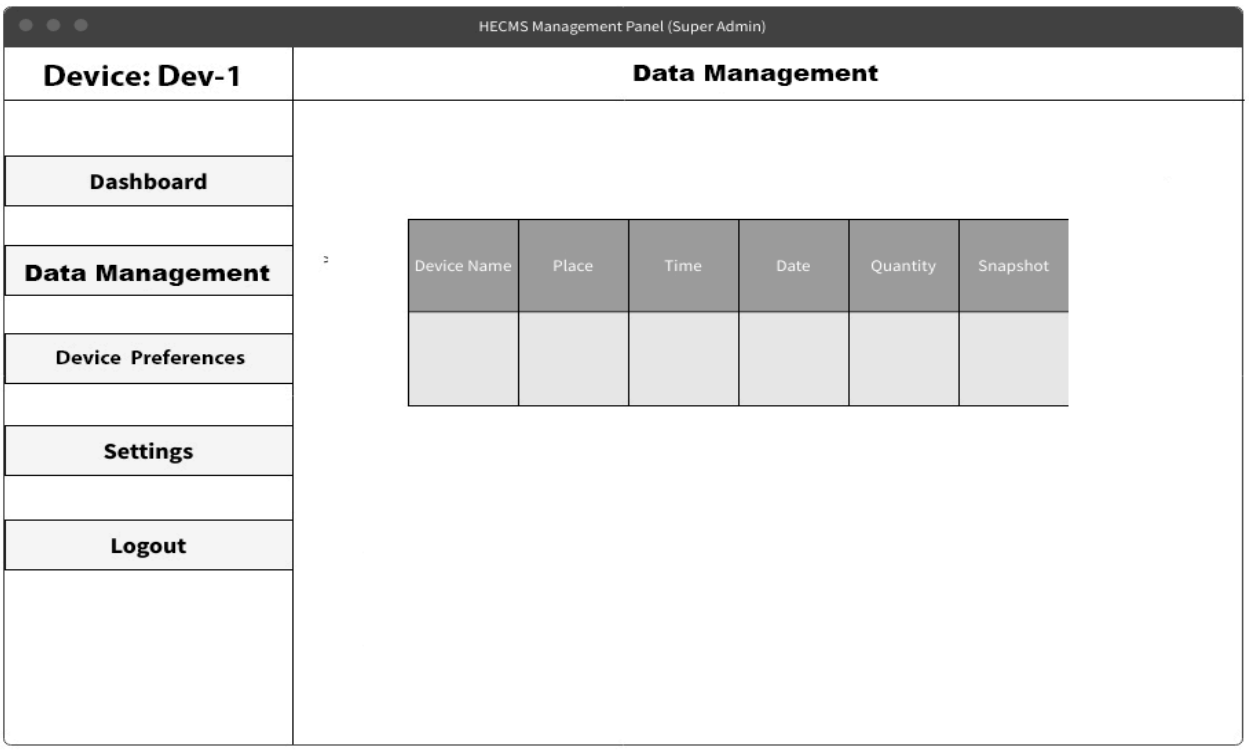


Figure 19 Management System Data Management Page Device Admin Wireframe

This is the wireframe of the data management page in the device admin, it stores all of the elephant detection information of the current device that is logged in, and the device admin can search for data.

3.6.10.11 Management System Device Preferences Page Device Admin Wireframe

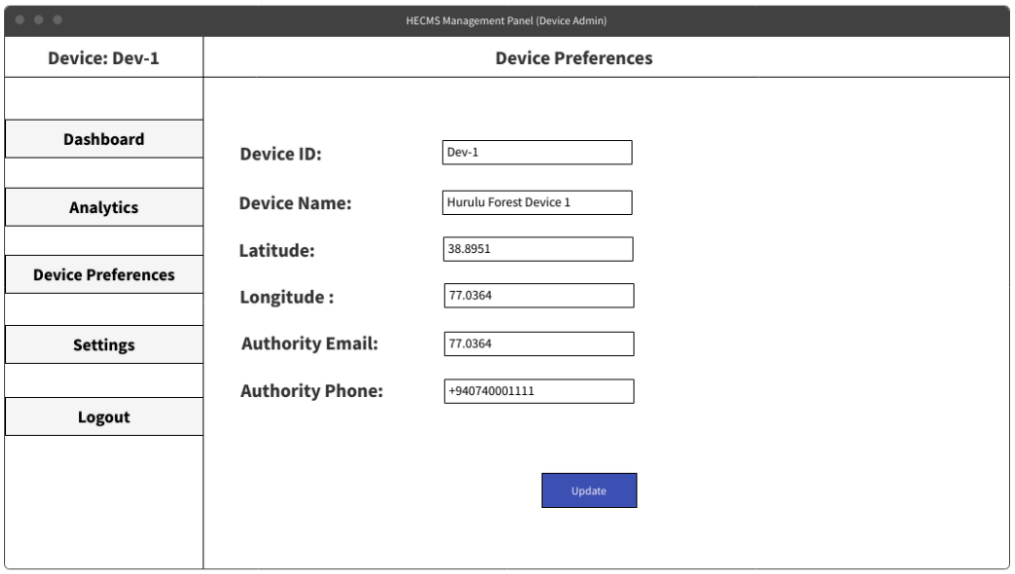


Figure 20 Management System Device Preferences Page Device Admin Wireframe

This is the wireframe of the management system device admin where the device admin can edit the device information such as device name, location, and authority email and authority passwords can be edited in this screen.

3.6.10.12 Management System Settings Page Device Admin Wireframe

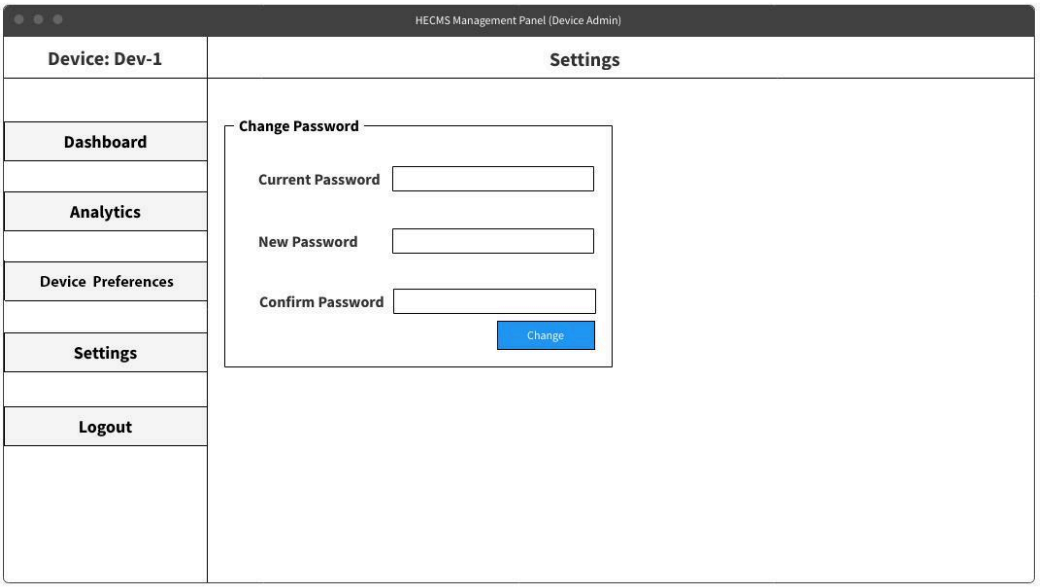


Figure 21 Management System Settings Page Device Admin Wireframe

This is the management system device admin settings page wireframe where the device admin can change the account password.

3.6.11 Project User Interfaces

The user interface is created based on the wireframe created from the system requirements and the system architecture using an online tool called “Figma”; these user interfaces are designed for Monitoring System and Management System Super Admin and Device Admin; it graphically explains the skeleton of the project nodes.

3.6.11.1 Monitoring System Login Page User Interface

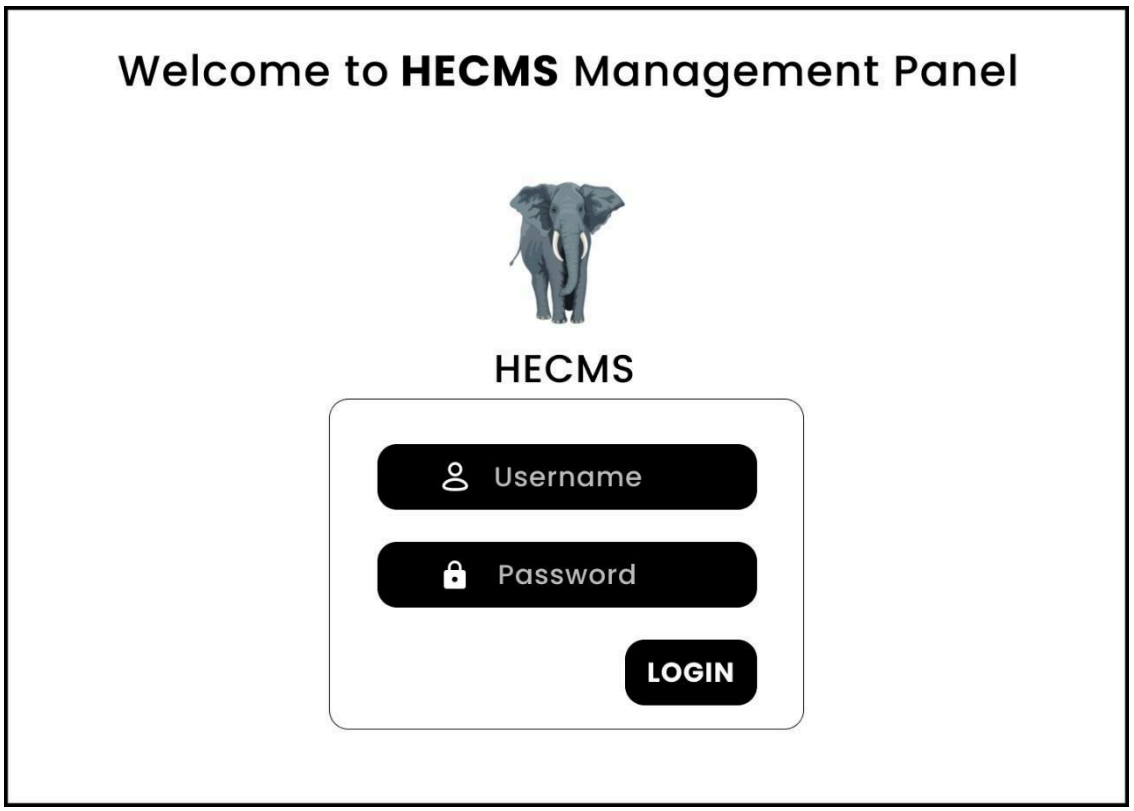


Figure 22 Monitoring System Login Page User Interface

This is the user interface for the monitoring system login page. The device must enter its password and username to log into the monitoring system; it has a text box for the username and a password box for the password.

3.6.11.2 Monitoring System Dashboard User Interface

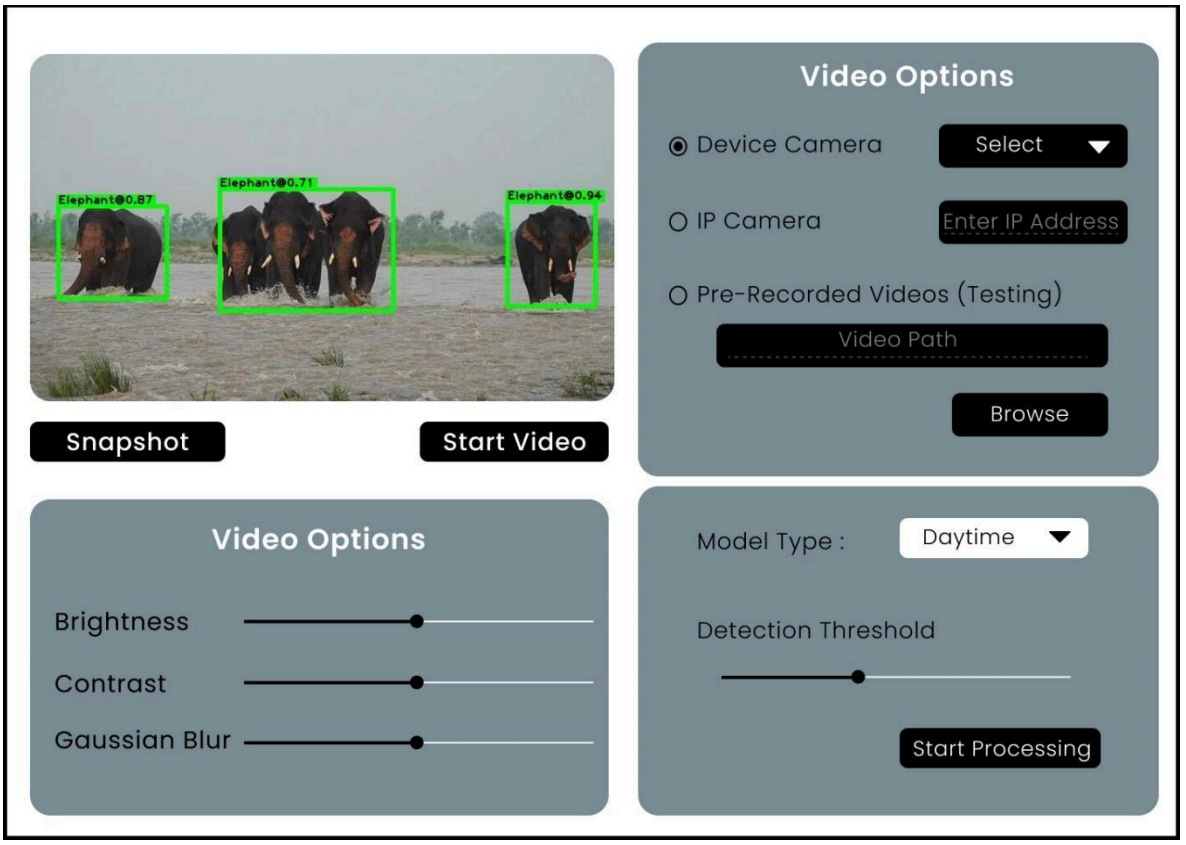


Figure 23 Monitoring System Dashboard User Interface

All of the necessary pieces, such as a device camera selector, a textbox for the IP camera, a browse button to load the pre-recorded video, a pick box for model architecture, and a slider pane for detection threshold, are included in the monitoring system's dashboard user interface. The Start Processing button initiates the processing, and a few sliders change the color coordination of the frames, such as brightness, contrast, and blur. A snapshot button is available.

These elements will be transferred to the user interface and written as clickable elements in the appropriate programming tools.

3.6.11.3 Management System Login User Interface

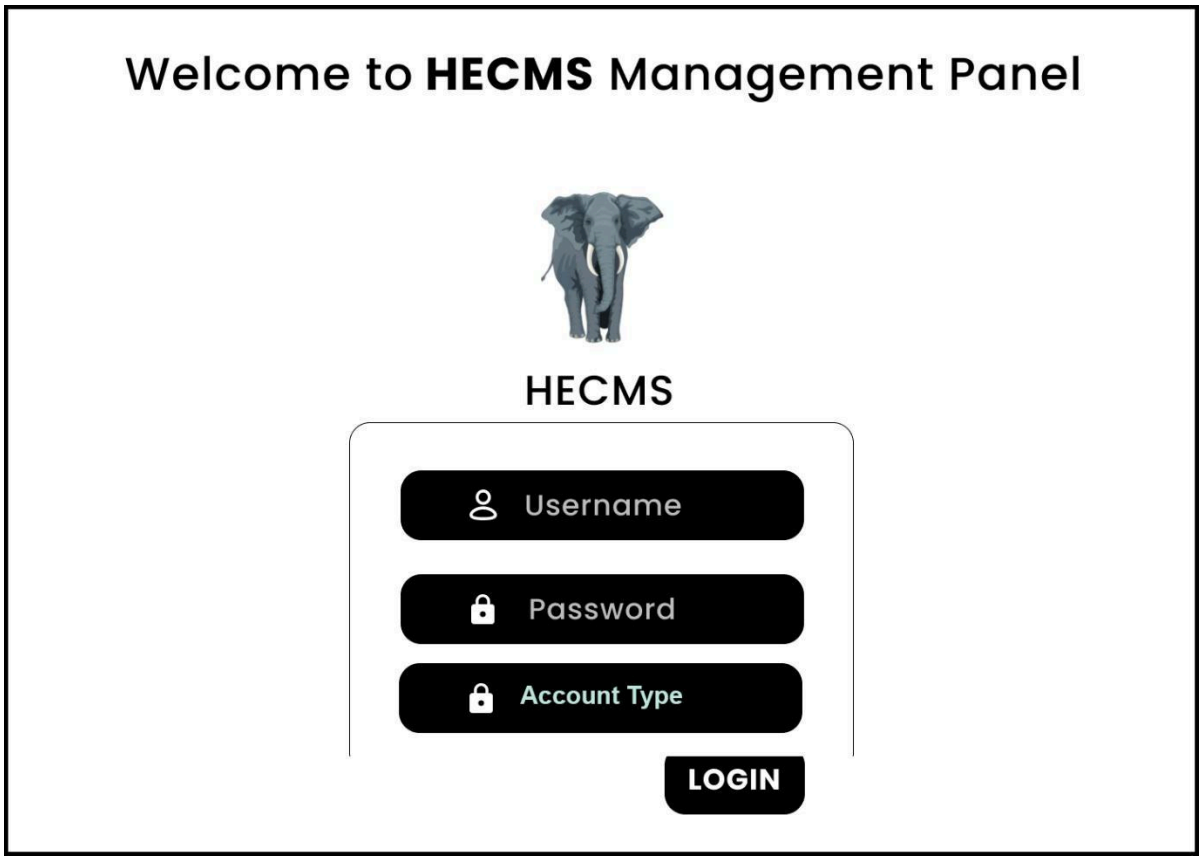


Figure 24 Management System Login User Interface

This is the user interface for the management system login page; it comprises three text elements for collecting user input on username, password, and account type. In the final version, there will be a login button that can be clicked. When you click, the database will check to see if the account is present and if it has access. It will open the super admin dashboard and be programmed in the proper programming language as a clickable object.

3.6.11.4 Management System Super Admin Dashboard User Interface

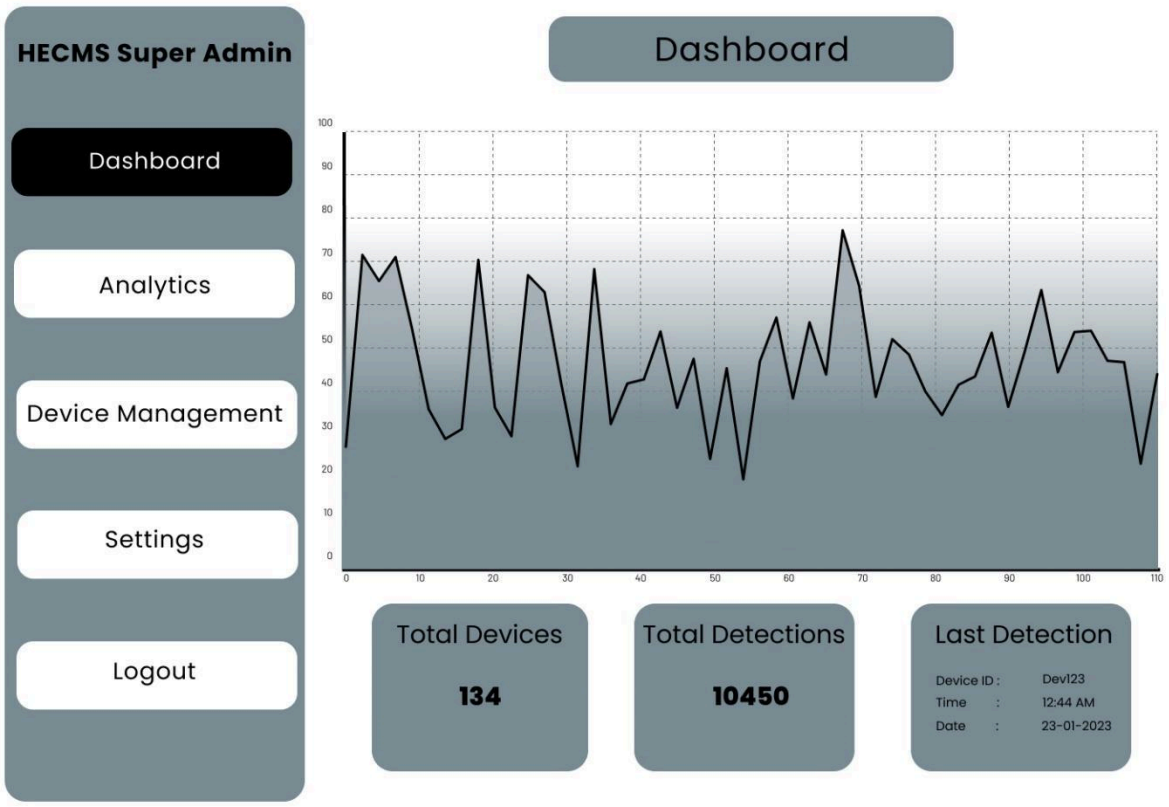


Figure 25 Management System Super Admin Dashboard User Interface

This is the super admin management panel dashboard, which includes capabilities and data visualizations; it contains numerous features, and the sidebar consists of sections such as Dashboard, Data Management, Device Management, Settings, and Logout.

The dashboard includes a visually appealing line chart that displays the element traffic data month by month. This line chart will be animated with information such as the total number of registered devices and conflicts, the number of features discovered, and so on; this user interface will be created to be a proper software object.

3.6.11.5 Management System Data Management Super Admin User Interface

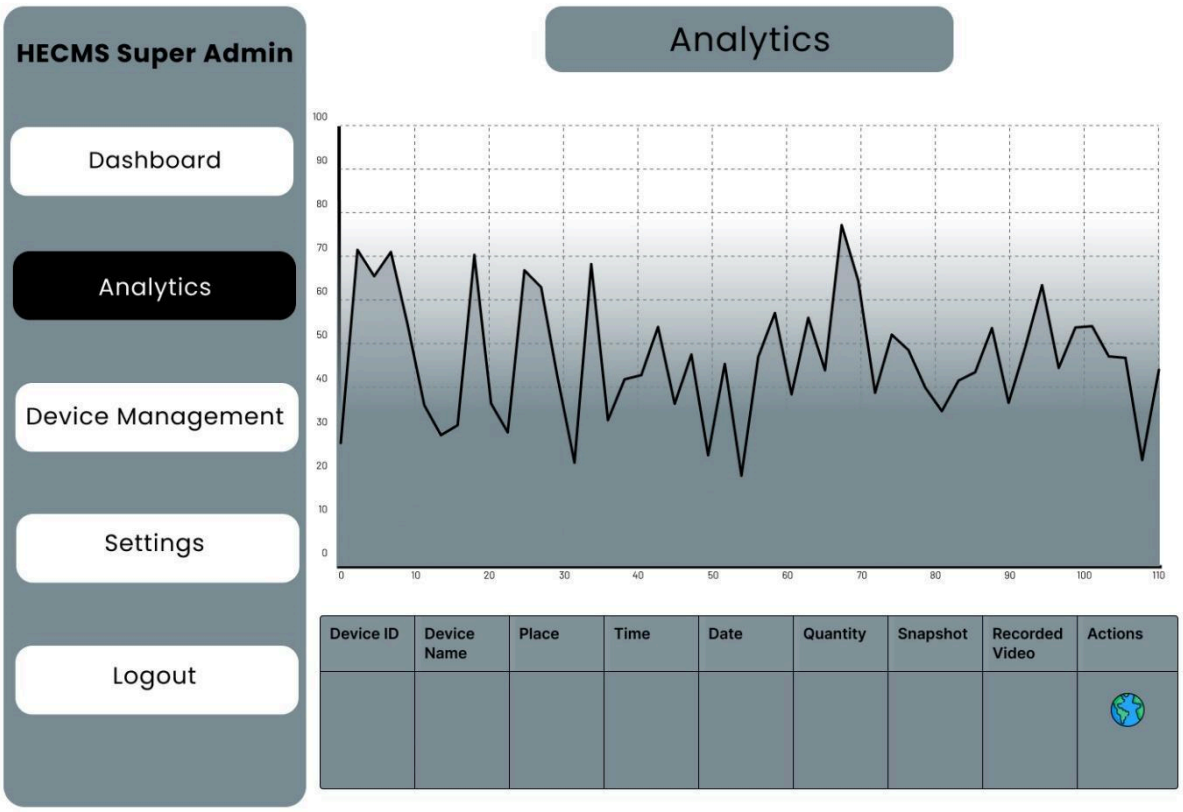


Figure 26 Management System Data Management Super Admin User Interface

This is a user interface for a super admin data management page; it contains data obtained from the monitoring system such as DeviceID, Device Name, Location, Time, Date, Snapshot, and so on, and it puts the data in a nicely aligned table.

This interface will be built into natural software objects. This is the data management page, from which all human-elephant conflict data is sent.

3.6.11.7 Management System Device Management Super Admin User Interface

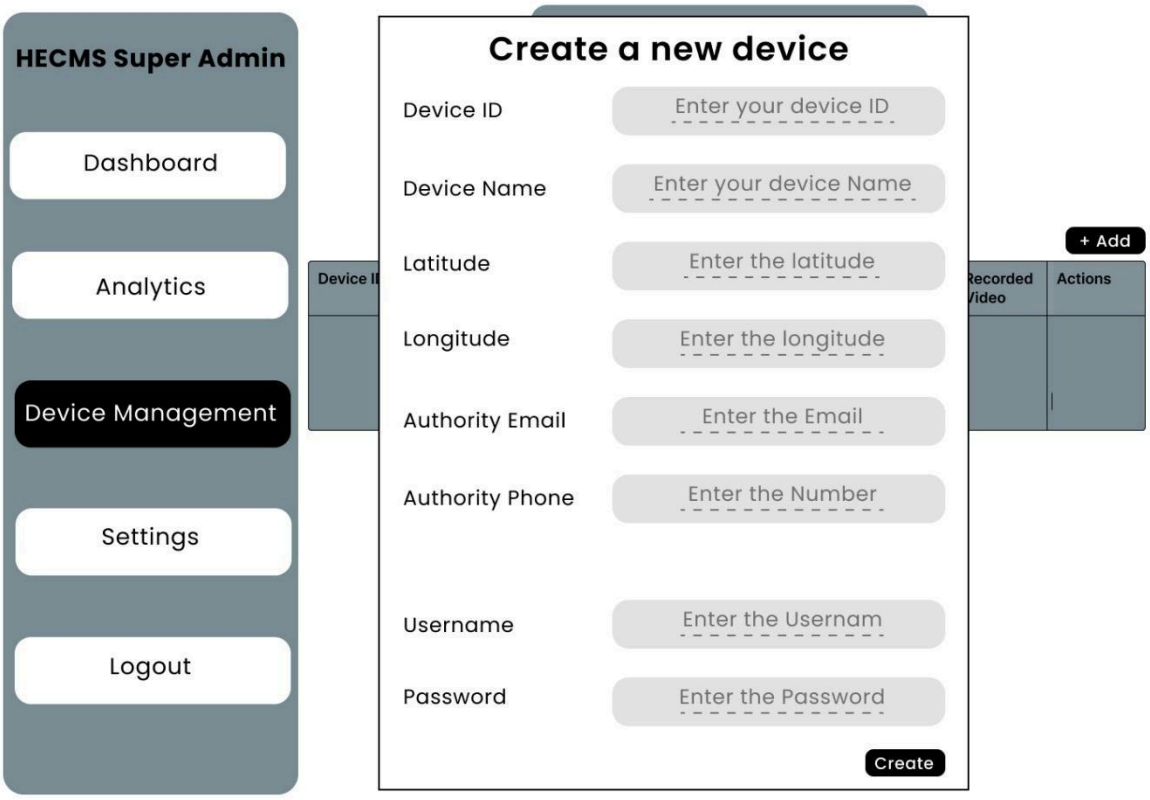


Figure 27 Management System Device Management Super Admin User Interface

This is the user interface for the device administration page, and it has numerous elements. This page displays information about all registered devices such as Device ID, Device Name, Latitude, Longitude, Authority Email, Authority Phone Number, Model Threshold, Username, and Password, as well as action buttons such as delete, modify, and see on map.

Once a device is registered, the information will be displayed here; the super admin can then amend that information if necessary; for example, if the authority phone number changes, the super admin can edit that to the new phone number. Similarly, the super admin has access to all device fields.

3.6.11.8 Management System Settings Super Admin User Interface

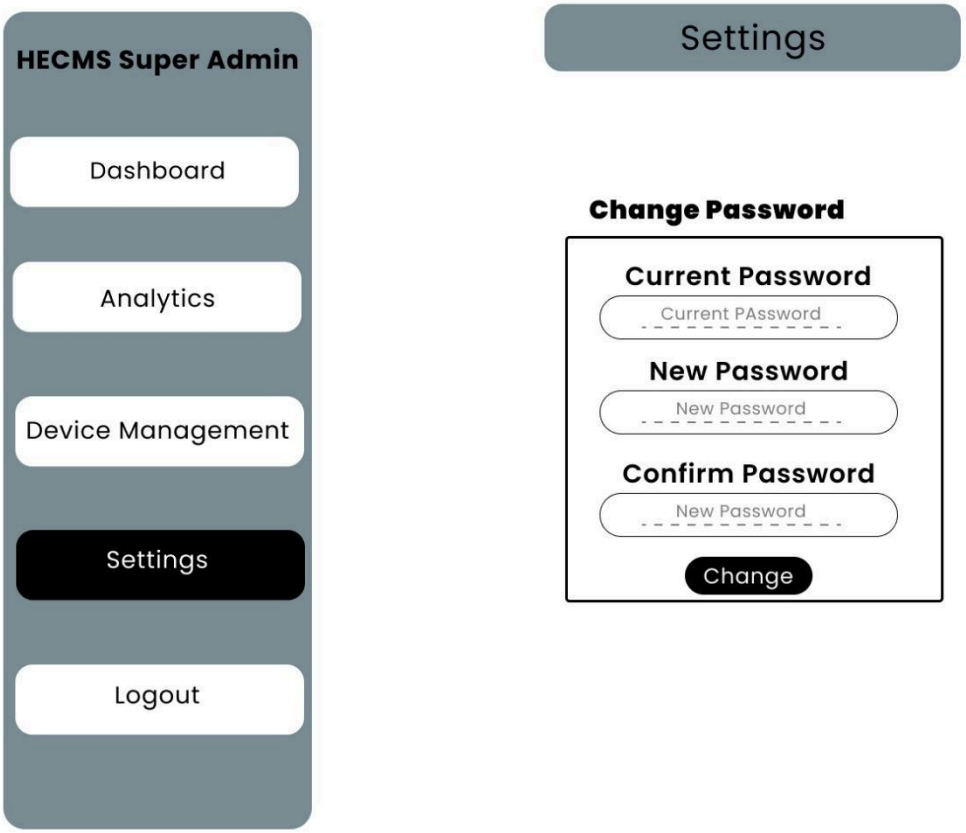


Figure 28 Management System Settings Super Admin User Interface

This is the user interface for the management system of the device admin; it has some fields such as current password, new password, and confirms password, so this page is used to change the credentials of the management system of the super admin.

3.6.11.9 Management System Login Page Device Admin User Interface

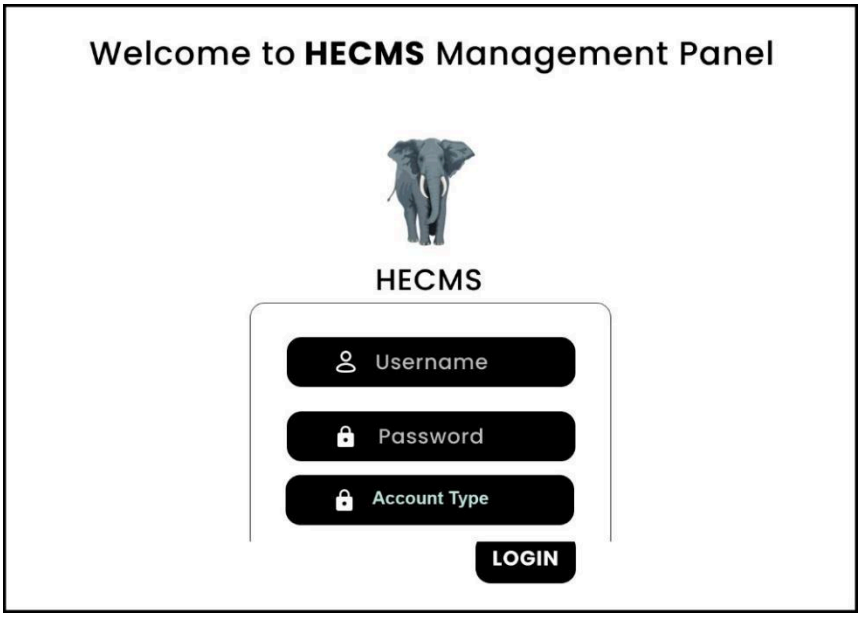


Figure 29 Management System Login Page Device Admin User Interface

3.6.11.10 Management System Dashboard Page Device Admin User Interface

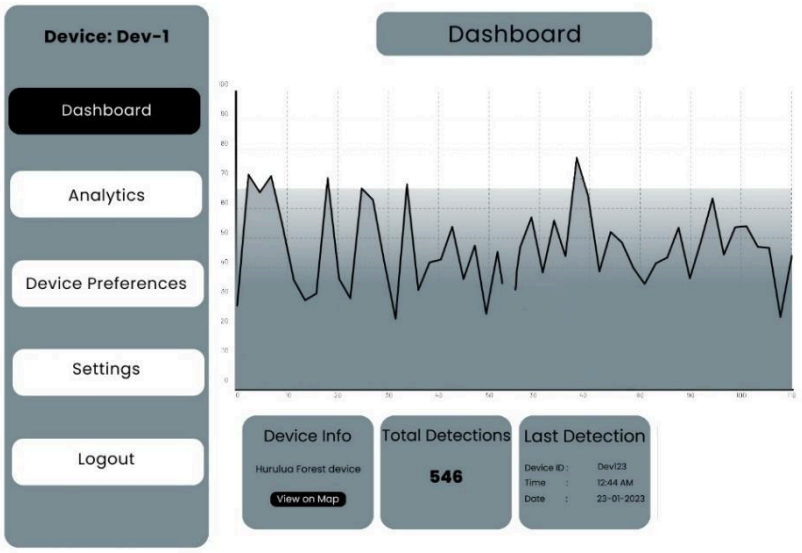


Figure 30 Management System Dashboard Page Device Admin User Interface

This is the user interface of the dashboard and login page; in the dashboard, it will show all the element conflict incidents taken by the current devices that are logged in. It also has a button to view the current device's location. It has the total number of incidents that are taken by the device.

3.6.11.11 Management System Data Management Page Device Admin User Interface

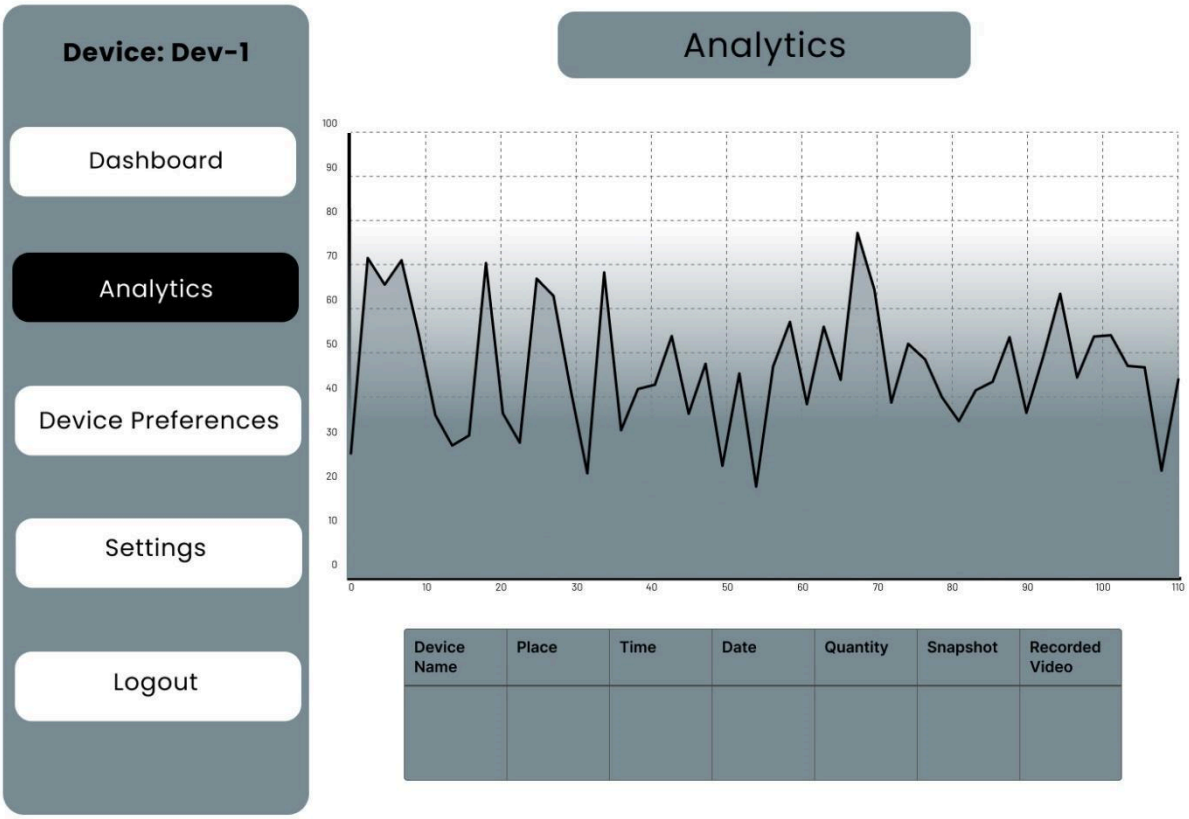


Figure 31 Management System Data Management Page Device Admin User Interface

This is the user interface of the data management page in the device admin, it stores all the elephant detection information of the current device that is logged in, and the device admin can search for data.

3.6.11.12 Management System Device Preferences Page Device Admin User Interface

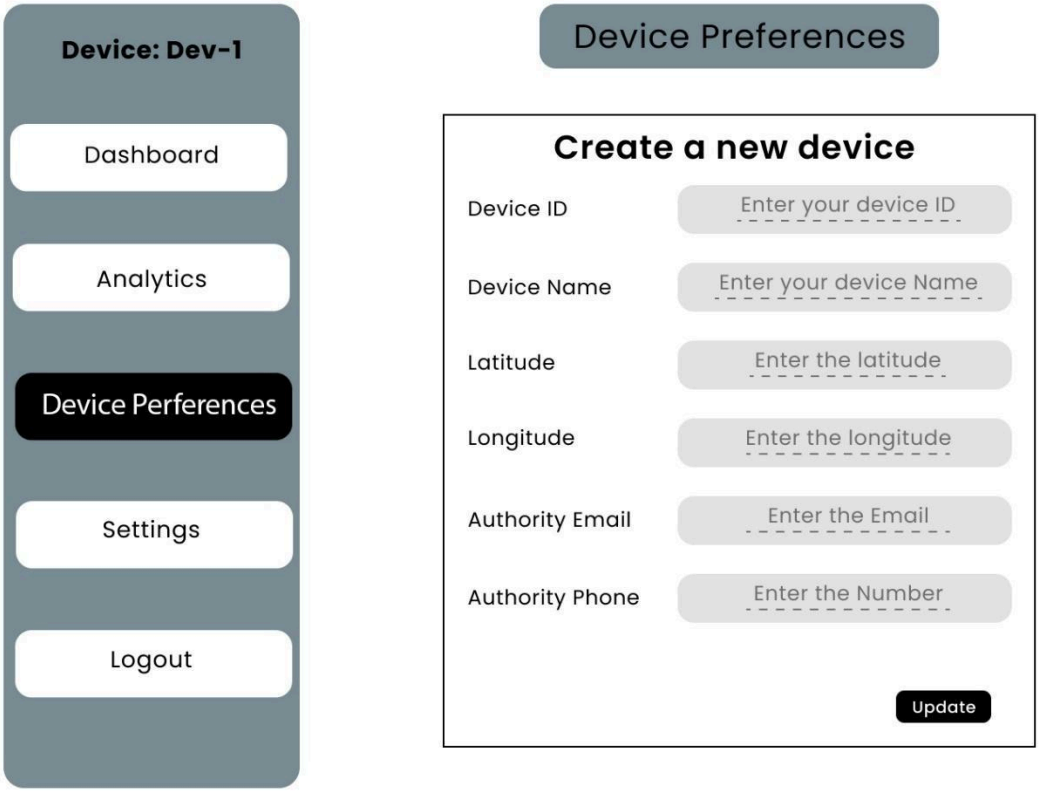


Figure 32 Management System Device Preferences Page Device Admin User Interface

This is the wireframe of the management system device admin where the device admin can edit the device information such as device name, location, and authority email, and authority passwords can be edited in this screen.

3.6.11.13 Management System Settings Page Device Admin User Interface

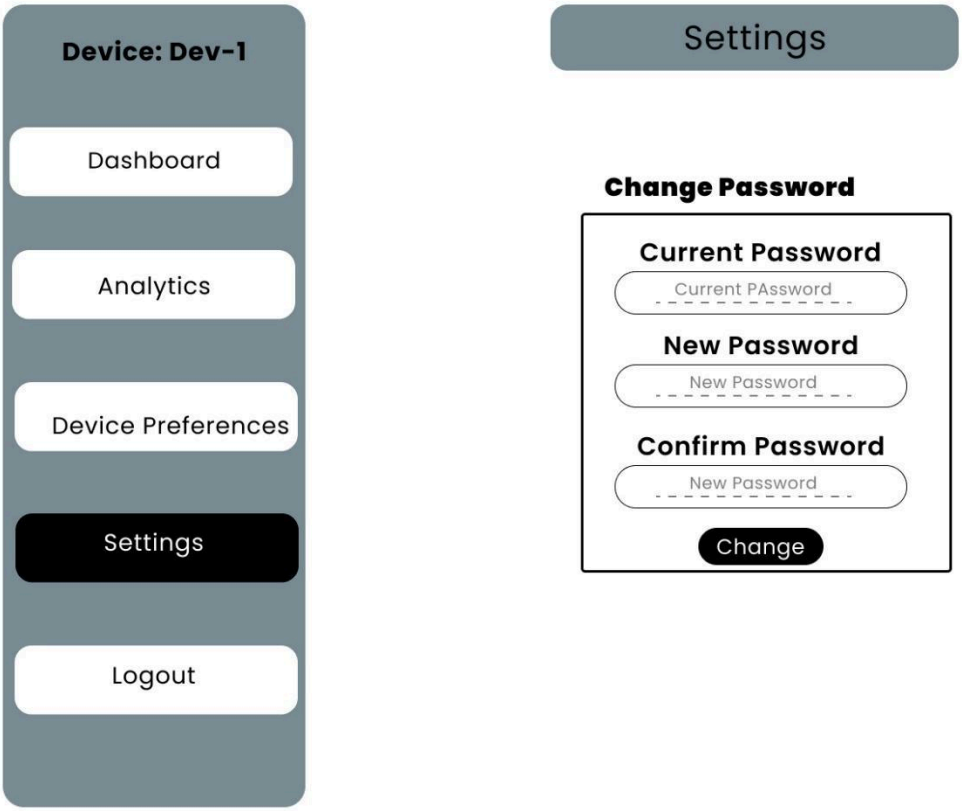


Figure 33 Management System Settings Page Device Admin Wireframe

This is the management system device admin settings page user interface where the device admin can change the account password.

3.7 Data Collection Method

Interviews: This research uses the interview method to collect information from farmers and people in rural areas; from the discussion, the researcher can get some ideas about how villagers use manual techniques to block the elephants, such as firecrackers and loud noises. The researcher learned that a watchman (villager/farmer) sits and watches for elements at night. If elephants find in their eyes, he manually alerts the villages. From the interview, basic requirements are clearly defined.

Online Resources: On the other side, many research papers and similar literature were thoroughly studied to understand the problem, and more information and system methodologies were extracted from those sources.

3.7.1 How the data was collected for dataset Preparation to train AI models.

One of the challenging parts of this research is preparing the dataset for model training. This research uses many methodologies to gather data from different sources to build a better dataset; data collection sources are listed below.

- Open Images Dataset V7
- Google Images
- Open Source Kaggle Datasets
- YouTube Elephant Videos

Open Images Dataset V7: An open-source computer vision dataset covers 9 million images with labels spanning thousands of object categories; from this dataset, only the data elephant category was downloaded; this dataset cannot be downloaded manually to download one specific category. The researchers used the OIDv4_Toolkit, an open-source tool that can be used to download one class of data; this is how 15% of the data is collected.

Google Images: in this method, the researcher manually downloads all the public images of elephants on Google, and many public free image search websites are also used to download, but it has more useless ideas of elements such as toys inappropriate images; from this, only 5% of the data were collected.

Open Source Kaggle Datasets: Some free, open-source elephant datasets were available on the Kaggle platform; the researcher downloaded three different datasets from this platform, and 20% of the data were collected from this method.

YouTube Elephant Videos: There were 124 videos of elephants downloaded from YouTube; this is a significant data source; from YouTube, 60% of data are gathered.

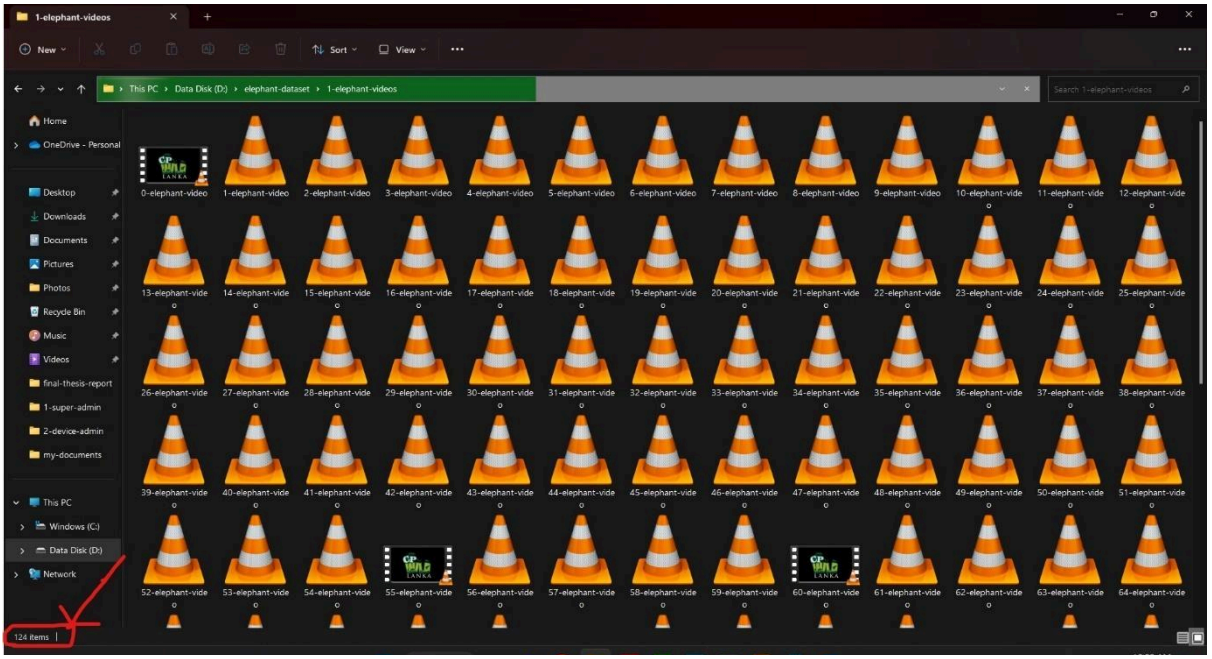


Figure 34 Elephants YouTube Video (124)

From YouTube, there are 124 elephant videos collected with 17.6 hours durations; all videos are high quality and can extract clear images. The researcher was able to get enough and a much data from YouTube.

As a result, there were **20416** elephant images collected; those are **10208** daytime images and **10208** night-time images. The next "Data Analysis Method" section explains how this data is analyzed and converted as night vision images and how the dataset was prepared in more detail.

In conclusion, the data collection is primarily done with the interview method to get essential information. Many research papers, articles, journals, and public sources are studied. The dataset is prepared using many techniques, such as scraping public images, downloading public datasets, and YouTube videos.

3.8 Data Analysis Method

The above section clearly explains how the data was collected, such as requirements and datasets for model training. This section will explain how the data was analyzed and how those raw YouTube videos are extracted into frames.

It also covers how the collected data are inverted into night vision to produce a nighttime dataset. To achieve this, the research had to involve a lot of programming and Python script writing; those matters will be covered thoroughly in this section.

The interview data was brainstormed to extract system requirements; Much information was collected from public sources such as online articles, research papers, and journals. Those data are analyzed and brainstormed a lot to extract requirements.

3.8.1 How the data was analyzed for dataset Preparation to train AI models.

After the data collection is done for the dataset preparation, the data were needed to be analyzed and inverted into a different format for training models; there is a need to augment the data to make more night vision samples; these techniques are done with Python scripting and more technical involvement.

As mentioned in the data collection section, many sources are used to collect data, such as Kaggle open-source datasets, Google online images, YouTube videos, and Open Images Dataset V7.

After collecting and merging all data into one directory, the next step was to clean up and remove unwanted and useless data, such as toy elephants or elephants in different positions that are not needed; this deleting process is done manually by viewing images into collages and removing useless samples.

Two analyzing scripts are implemented in Python, one is to extract frames from raw YouTube videos, and the other one is to convert all the collected data into night vision. The researcher wrote those two scripts to analyze the data; by utilizing the power of programming, the researcher achieved the best data-analyzing outputs.

3.8.2 Python script to extract frames from YouTube videos.

```
1. import cv2
2. import os
3.
4. folder_path = "input-videos/"
5. folder_path_output = "outputs/"
6.
7. index = 0
8. for file_name in os.listdir(folder_path):
9.     if file_name.endswith((".mp4", ".avi", ".mov")):
10.           file_path = os.path.join(folder_path, file_name)
11.           cap = cv2.VideoCapture(file_path)
12.           fps = int(cap.get(cv2.CAP_PROP_FPS))
13.           time_counter = 0
14.
15.           while True:
16.               ret, frame = cap.read()
17.
18.               if not ret:
19.                   break
20.               time_counter += 1
21.
22.               if time_counter == 3 * fps:
23.
24.                   cv2.imwrite(f"{folder_path_output}/image-{in
   dex}.jpg", frame)
25.                   index = index +1
26.                   print(f"image-{index} Saved!")
27.
28.                   time_counter = 0
29.
30.           cap.release()
31.
```

Figure 35 Python script to extract frames from YouTube video.

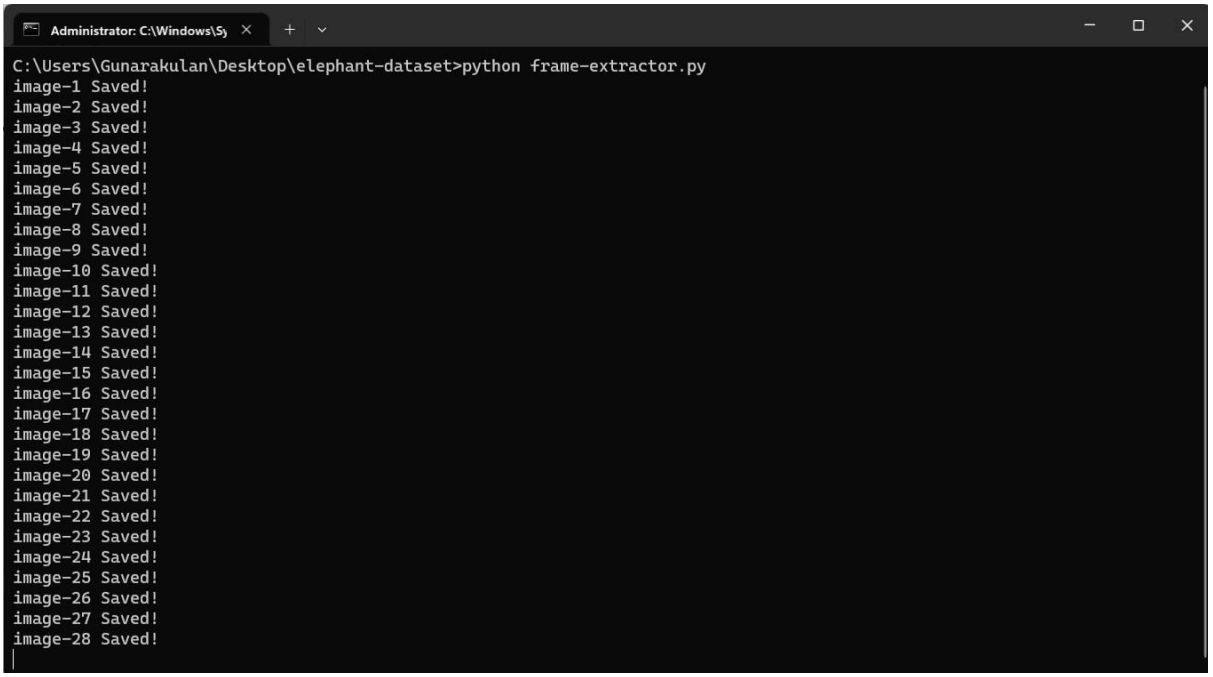


Figure 36 Python script to extract frames from YouTube video outputs.

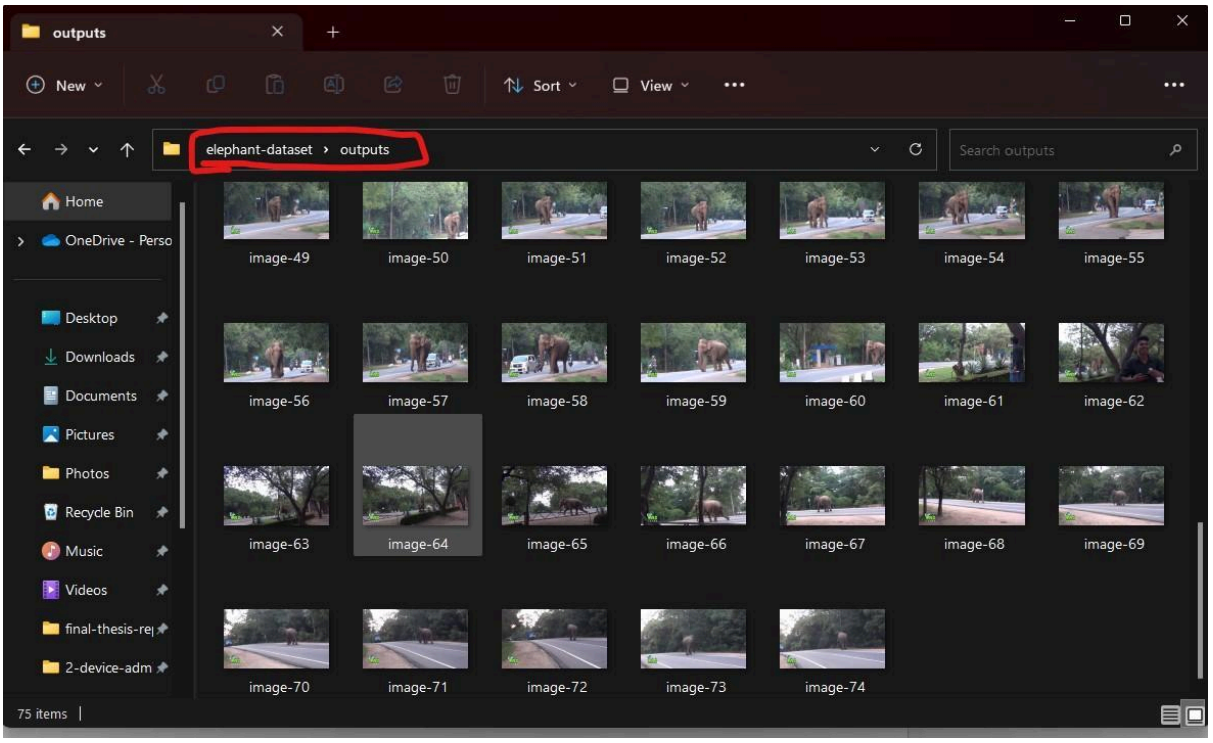


Figure 37 Python script frames extraction output folder.

The script is designed to extract frames from raw YouTube videos using the OpenCV module in Python. The extracted frames are saved as JPEG images for further analysis or processing. The script takes video files from a specified input folder, processes them, and saves the

frames in an output folder. It loads videos from the “1-elephant-videos” folder, then processes and saves the extracted frames in the “outputs” folder.

The input video files should be placed under the "1-elephant-videos" to use the script. folder. it will then process videos and save the extracted frames in an output folder called "outputs.” It then automatically loops the input folder for valid video files. After that, once a video file is identified, the script reads each frame from the video, processes it, and saves it as a JPEG image in the output folder.

The script is designed to be easy to use and highly customizable. The FPS, output and input folders can adjust to control the number of frames extracted or modify the output file names to suit the needs. The script keeps track of the frames processed and saves every third frame from striking a balance between extracting enough frames for analysis and avoiding excessive frames that may consume storage space and processing time. Below is a step-by-step explanation of how this script is executed.

- **Step 1:** The necessary libraries, cv2 (OpenCV) for video processing and os for file system operations, are imported.
- **Step 2:** The input and output folder paths are defined using the variables folder_path and folder_path_output.
- **Step 3:** Variables such as index (for keeping track of the image file names), fps (for storing the frames per second of the input videos), and time_counter (for counting the frames processed) are initialized.
- **Step 4:** The script loops over the files in the input folder using the os.listdir() function
- **Step 5:** For each file in the input folder, the script checks if the file has a valid video file extension (.mp4, .avi, .mov) using the endswith() method.
- **Step 6:** The script uses the cv2.VideoCapture() function to open the video file for reading. The file path is constructed using the os.path.join() function to concatenate the folder path and file name.
- **Step 7:** The script enters a while loop that reads each frame from the video using the cap.read() function. If the frame is read successfully (ret is True), the script increments the time_counter by 1.
- **Step 8:** The script checks if the time_counter reaches three times the video’s frames per second (fps). If it does, it saves the current frame as a JPEG image in the output

folder using the cv2.imwrite() function, and a "Saved!" message is printed to indicate progress.

- **Step 9:** After saving the image, the time_counter is reset to 0 to start counting frames again.
- **Step 10:** The script releases the video capture using cap.release() to free up system resources.
- **Step 11:** The script repeats the above steps for all video files in the input folder.

The script provides a convenient way to extract frames from raw YouTube videos using OpenCV in Python, allowing for further analysis or processing of the frames for various computer vision tasks, in order to this script, some important dependencies need to be installed such as OpenCV and Python must be installed.

8.8.3 Python script to convert all images into night vision.

```
1. import os
2. import cv2
3. import numpy as np
4.
5. folder_path = 'input-dataset/'
6.
7. index = 0
8.
9. for filename in os.listdir(folder_path):
10.       file_path = os.path.join(folder_path, filename)
11.       img_gray = cv2.imread(file_path, cv2.IMREAD_GRAYSCALE)
12.
13.       if img_gray is None:
14.           continue
15.
16.       current_brightness = np.mean(img_gray)
17.
18.       divider_value = 0
19.
20.       if current_brightness <=70:
21.           divider_value = 1
22.       elif current_brightness >= 71 and current_brightness <=1
   00:
23.           divider_value = 1.3
24.       elif current_brightness >= 101 and current_brightness <=
   150:
```

```
            divider_value = 2
        elif current_brightness >= 151 and current_brightness <=
    200:
            divider_value = 3
        elif current_brightness >= 201 and current_brightness <=
    250:
            divider_value = 4
        elif current_brightness >= 251 and current_brightness <=
    300:
            divider_value = 4.5
        elif current_brightness >= 301 and current_brightness <=
    350:
            divider_value = 5.3
        elif current_brightness >= 351 and current_brightness <=
    400:
            divider_value = 6.3
        elif current_brightness >= 401 and current_brightness <=
    450:
            divider_value = 7.1
        elif current_brightness >= 451 and current_brightness <=
    500:
            divider_value = 8.0
        elif current_brightness >= 501:
            divider_value = 10

        img = img_gray // divider_value

        print(f"{filename}: {index}")
        cv2.imwrite(f"output-data/{filename}", img)
        index = index + 1

```

Figure 38 Python script to convert all images into night vision.

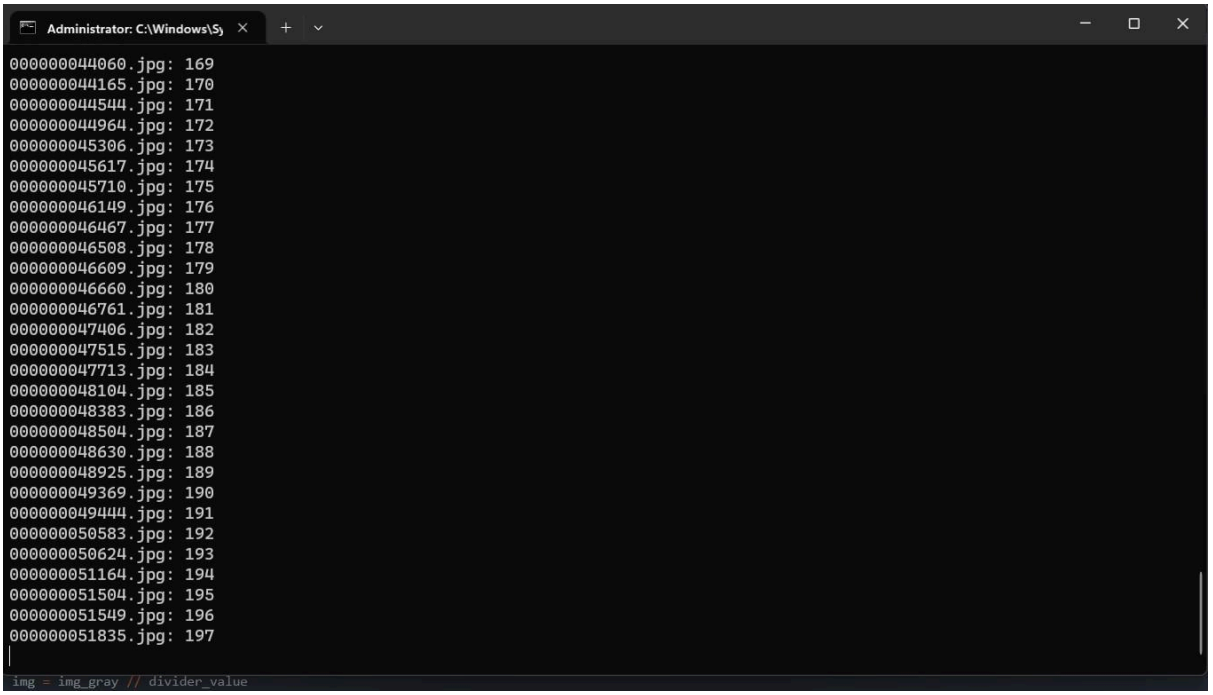


Figure 39 Python script to convert all images into night vision output CMD.

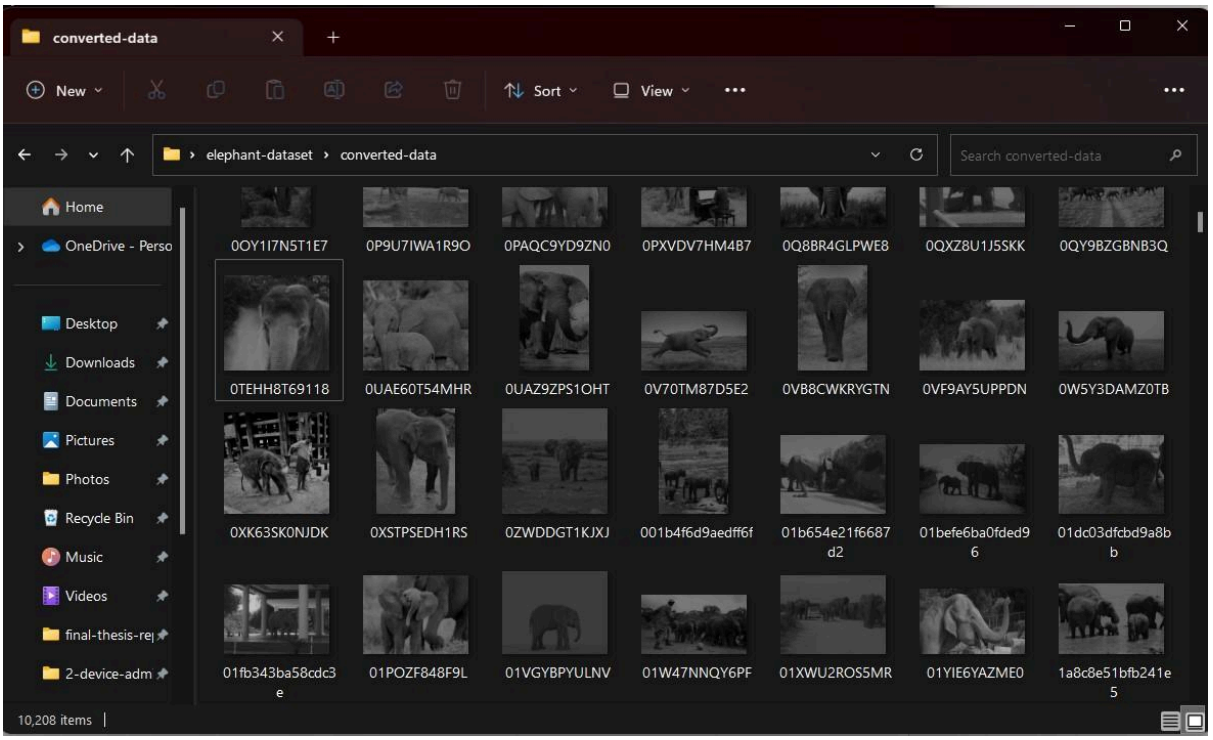


Figure 40 Python script to convert all images into a night vision Output Folder.

This script is used to convert daytime images into IR nighttime versions artificially; it is not accurate like real IR (infrared) images; the idea is the simulate that. So, this script reduces and adjusts the brightness and another color gradient to convert images to look like IR.

- **Step 01:** The necessary libraries, os, cv2, and numpy, are imported at the beginning of the script.
- **Step 02:** The folder path where the input images are located is defined using the folder_path variable.
- **Step 03:** An index variable index is initialized to keep track of the processed images.
- **Step 04:** The script enters a for loop to iterate through each file in the specified folder (folder_path).
- **Step 05:** For each file, the full file path is generated using the os.path.join() function by combining the folder_path and the filename.
- **Step 06:** The grayscale image is read from the file path using the cv2.imread() function with the flag cv2.IMREAD_GRAYSCALE, which converts the image to grayscale. The resulting grayscale image is stored in the variable img_gray.
- **Step 07:** The current brightness of the grayscale image is calculated by finding the mean pixel value using the np.mean() function, and it is stored in the variable current_brightness.
- **Step 08:** Based on the value of current_brightness, the script determines a divider value to adjust the image's brightness. It uses a series of conditional statements to map the brightness level to a specific divider value.
- **Step 09:** The image's brightness is adjusted by dividing the pixel values of the grayscale image by the determined divider value using the integer division operator (//), and the resulting image is stored in the variable img.
- **Step 10:** The processed image is saved in a new folder named "converted-data" with the same filename using the cv2.imwrite() function.
- **Step 11:** The index variable is incremented to keep track of the processed images.
- **Step 12:** The script prints the filename of the processed image along with the index using the print() function to show the processing progress.
- **Step 13:** The loop continues for the remaining files in the folder until all images are processed.

- **Step 14:** Once the loop is completed, the script finishes its execution, and the processed images are saved in the "converted-data" folder, ready for further use.

3.9 Implementation

The last two previous sections discussed how the data was collected and analyzed; this section outlines how the implementation procedures are executed to build the research outcome; it also discusses the working methods, programming methods, and the materials and tools used during the research and development period.

Implementing the research findings as a software product is divided into two main parts: the monitoring system and the management system. The monitoring system is developed as a software product to act as a camera trap in the jungle. This monitoring system will utilize advanced computer vision algorithms and machine learning techniques to process images and videos captured by the camera trap and automatically detect and identify wildlife species in the jungle.

Conversely, the management system is hosted on a server as a web application. It is the central data management, analysis, and visualization hub. The management system receives the processed data from the monitoring system, stores it in a database, and provides a user-friendly interface for super admin and device admins to access and analyze the collected data. It has advanced data visualization and reporting to gain insights and make informed decisions based on the managed wildlife monitoring data.

The software product is designed to be user-friendly, considering the needs and requirements of wildlife researchers, conservationists, and others. It is intended to be scalable, flexible, and customizable, allowing users to configure and adapt the system to their specific monitoring needs and requirements. It will also prioritize data security and privacy, with robust authentication and authorization mechanisms to ensure the collected data is protected and only accessible to authorized users.

The software was implemented to be a powerful tool for wildlife monitoring and conservation efforts, enabling more efficient and accurate data collection, analysis, and decision-making. It will provide valuable insights into wildlife species' behavior, distribution,

and abundance, helping researchers and conservationists better understand and protect our precious natural resources.

To build this software, many tools are used; most of the tools are open-source and free to use, and those tools made up this research to achieve the expected outcome; there are some available tools used to perform general things such as documenting the works, etc., and there are many technical tools used such as programming languages, programming libraries, etc., those tools and platforms helped to implement this research as a software product. Below are all the tools, platforms, and technology stacks mentioned in this research and development.

3.9.1 Tools and Platforms

- General Tools
    - MS Word - For documentation works.
    - MS Excel - For data collection and project planning.
    - MS PowerPoint – For making presentations.
    - OBS Studio Video Recorder – To record videos.
    - OIDv4 ToolKit – To download Datasets.
- Technology Stack
    - Python – Used as a major programming language for the monitoring system.
    - Tkinter - Used to build the UI for the monitoring system.
    - TensorFlow – Used to train and run AI models.
    - OpenCV – Used to perform computer vision tasks.
    - MySQL – Used for the database.
    - PyAutoGUI – Used to test the monitoring system automatically.
    - Selenium – Used to test the management system automatically.
    - Pygame – Used to play artificial sounds.
    - UUID – Used to generate random ids when saving images to name them.
    - base64 – Used to convert image as a base64 to upload to the database.
    - JSON – Used to store system credentials.
    - Keras – Used to model running.

- Pyshorteners – Used to short Google Maps URLs when sending SMS.
- Object Detection Framework – Used as the main tool for elephant detection.
- Laravel – Used to build the Management System.

3.9.2 Implementation Methodology

This project used the Agile methodology for development since it involves artificial intelligence technology; typically, AI models must be tested and evaluated frequently to achieve the best accuracy. Therefore, this development, in a similar way, had to be tested many times to assess the model to produce a better-quality model.

This research cannot use any single-shot approach like a waterfall; in this development process, there will be more failures to overcome and produce a better model; this research needs to stick with a development method goes until it finds the best result; therefore, this research is going to use the prototyping methodology under the Agile framework.

During the model training process, at each stage, there will be a testing and model accuracy evaluation; if the model does not perform well, then another cycle of training with different approaches and parameters or with a different dataset to be used; this cycle will be repeated until the model gets the excellent results in detection.

Most of the time, these three development methods, planning, and research will be respected to achieve the destination.

Sometimes, if the computer vision detection architecture does not work, this research needs to identify the best architecture; more changes will occur during development.

The primary objective is not only about training a better model but also about building the best solution for the human-elephant conflict. To do that successfully, this research must build an additional software platform, such as a backend panel for elephant details monitoring and a monitoring system that inferences trained the model and send early warning messages if an elephant is found and it plays artificial sounds to block the elephants entering the villages.

3.10 Summary

In summary, the methodology section of this research provides a comprehensive overview of the various aspects and steps to conduct the study successfully. The **pragmatic approach** was chosen as the research philosophy, the **quantitative method** was chosen as the research design, and the **deductive approach** was selected as the research approach. Many strategies were carefully chosen to align with the research objectives and ensure the validity and reliability of the findings. The project design, description, requirements, and architecture were meticulously planned and organized to provide a clear roadmap for the implementation phase.

The database architecture and system structure, represented through ER and UML class diagrams, were designed to ensure efficient data management and system functionality. The system interactions were explained using the UML use case diagrams, highlighting the various scenarios and interactions between users (Super Admin/Device Admin) and the system. Algorithms were designed using flowcharts, visually representing the logical flow and decision-making processes of the system's functional executions, such as login and model detection flow.

The project planning phase was meticulously carried out using WBS diagrams and a Gantt chart to outline the tasks and their responsibilities, ensuring smooth project execution to conduct this research without any issues.

Wireframes and user interfaces were designed to visually represent the project's interface, ensuring user-friendly interactions and seamless usability of the project. Data collection and analysis methods were chosen carefully to ensure the accuracy and reliability of the research findings.

In conclusion, the methodology section provides a comprehensive and systematic approach to conducting the research project, outlining the research design, project requirements, system architecture, data collection, and analysis methods. The implementation phase will

be based on the foundation in the methodology section, ensuring a robust and reliable research study.

Chapter 04: Results

4.1 Introduction

The results section of the research paper presents the Human Elephant Conflict Mitigation System's findings and outcomes of the study. It provides a detailed and objective account of the collected, analyzed, and interpreted data during the research process. The Results section contains the critical component of the research.

This section contains information such as the project outcomes; the project is divided into two parts the monitoring system and the management system; this result has results on both sections.

It clearly explains the output of this research, such as how the system was built and what features there are; it also briefly describes how the system's functionalities are working and their results, etc.

The results section also contains detailed information on the testing results; three types of testing were held on this project: automatic, manual, and security.

In the automatic testing, the framework Selenium was used for the management system, and in the monitoring system, PyAutoGUI was used; it is a desktop application automation framework. The testing scripts are written in Python programming with PyAutoGUI and Selenium.

In the manual testing phase, some testing was held that could not be done automatically, so the results were thoroughly tested and evaluated to provide a better research outcome; in the security testing, there is some essential security best practices were tested, and the development of the security testing provided best results too.

In summary, this result section contains information on the system outputs and how the system's functionalities were implemented, how the results were tested, and all; it also has testing results of the AI model that explain the accuracy and performance of the models.

There are three types of testing done such as automatic testing, manual testing, and security testing, and those results also provided the best results.

4.2 Project Outcome

The Project Outcome section presents the project results, highlighting the achieved outcomes, functionalities, and other relevant aspects. This section provides a comprehensive overview of outcomes achieved through the project implementation. It showcases the project's final product or deliverables, showcasing the project's functional components, features, and capabilities.

As an outcome, this research achieved all the features mentioned in the requirement section of the project design. Monitoring System Management systems all are developed successfully; the observing system is built with all components, and the management system is divided into two different admins, the Super Admin and Device Admin; all the built features are modularized below, and there are going to explain in sperate sections with more details.

- **Monitoring System**
    - Login Option
    - Processing Panel
    - Video Options Panel
    - All features
- **Management System**
    - ***Super Admin***
        - Login Option
        - Dashboard
        - Data Management
        - Device Management
        - Settings
    - ***Device Admin***
        - Login Option
        - Dashboard
        - Data Management

- Device Preferences
- Settings

4.2.1 Monitoring System

The monitoring system is the primary system that runs on the ground to analyze video frames, detects elephants, sends warning email/SMS, and updates the conflict data to the management system. The result of the system is perfect, and it is built with so many functionalities and features. This section will explain the module and parts of the monitoring step by step.

4.2.1.1 Login Screen – Monitoring System

This is the login screen of the monitoring system. Login credentials are created in the management system as Device Admin, and that credential can be used to login into the system.

Features

- Successful login when credentials are matched.
- Display an error message when a wrong credential is provided.
- Clear textboxes automatically after a wrong credential is provided.

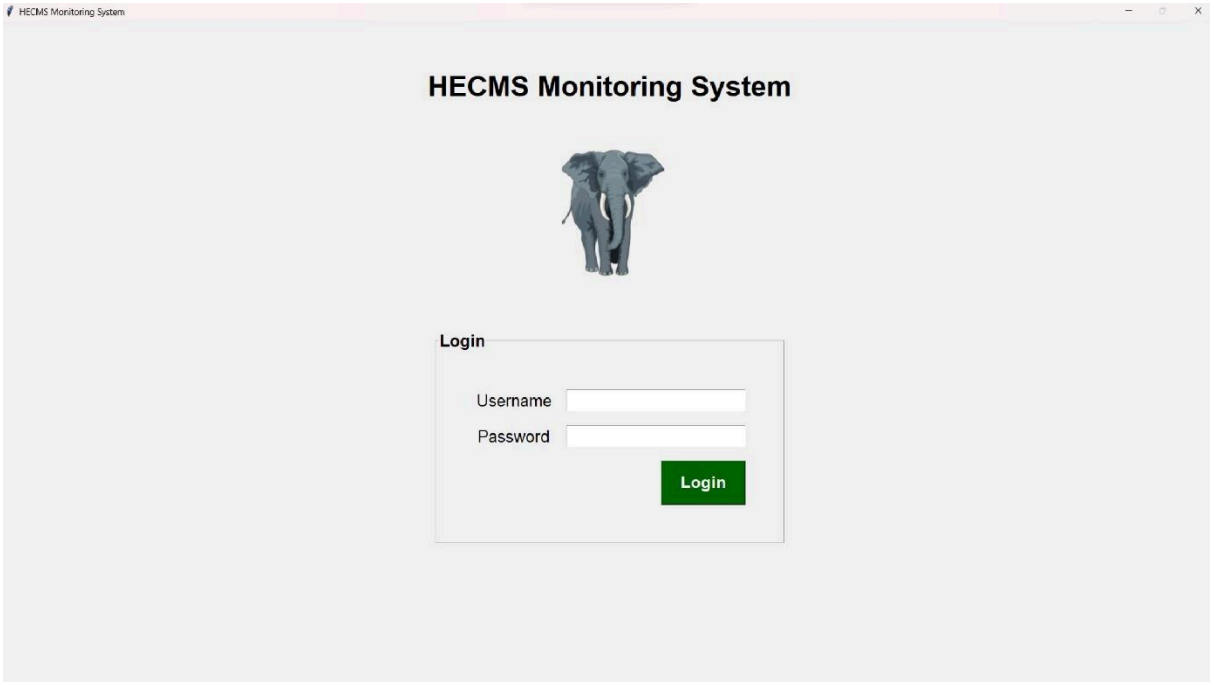


Figure 41 Login Screen – Monitoring System (Results)

4.2.1.2 Dashboard – Monitoring System

This is the main dashboard of the monitoring system, and this dashboard contains a lot of features and functionalities. It is the primary system with AI-powered models to detect elephants in real-time, send SMS and email, and update the database, this is the dashboard, and this dashboard was with Python and Tkinter framework.

This dashboard allows video processing from various sources, including the device camera, IP camera, pre-recorded videos, or browsing and selecting videos. There is an input video streams feature for real-time monitoring or analyzing recorded videos for post-event analysis. There are many features that are listed below.

Features

- Process video from the device camera.
- Process video from IP camera.
- Process video from pre-recorded videos.
- Browse to select videos.
- Have a screen to show the output on the left side.
- Snapshot button to take snaps of the screen.

- Threshold slider to adjust the detection rate.
- Alarm turns on/off feature.
- Sound effects can be changed.
- Model architecture can be changed.
- The brightness of the video can be adjusted.
- The Contrast of the video can be adjusted.
- The Blurriness of the video can be adjusted.
- Start Process button to begin the process.
- Stop Process button to stop the process.
- Restore Default Settings button to clear the changed options.

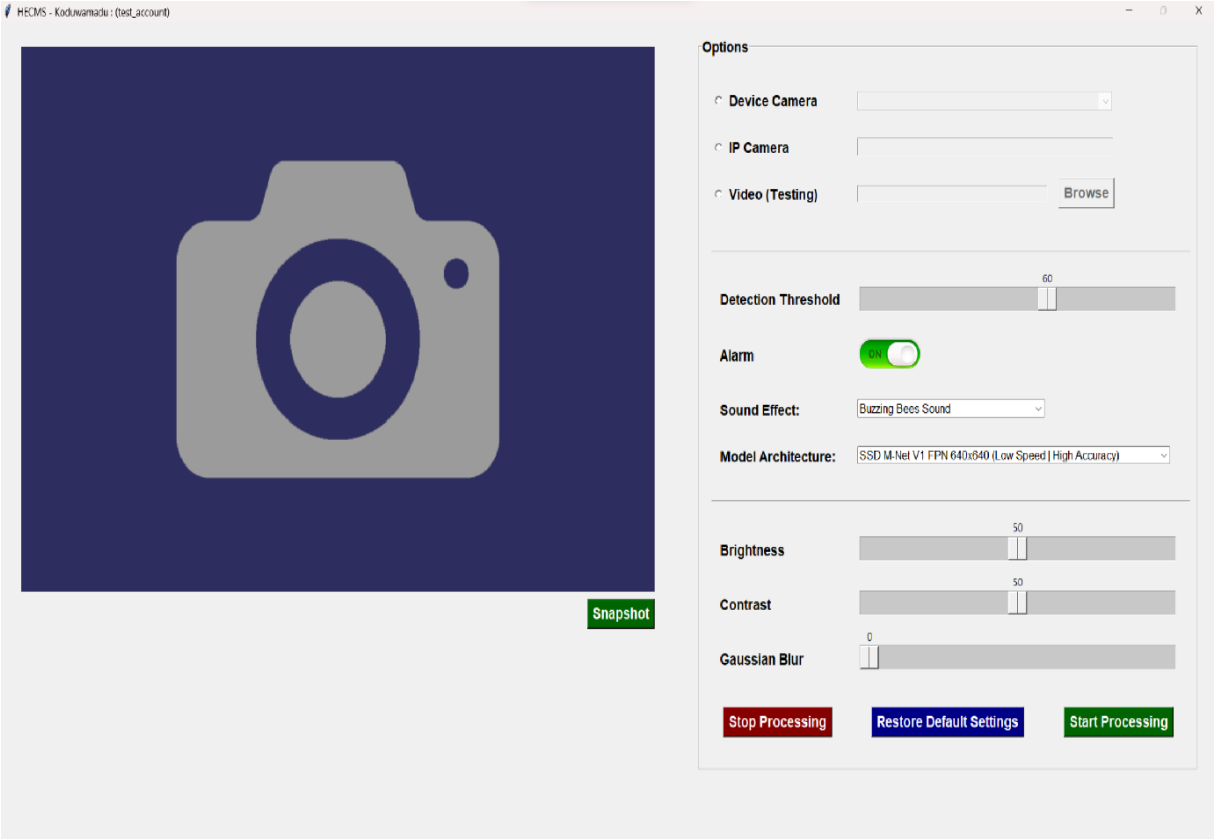


Figure 42 Dashboard – Monitoring System (Results)

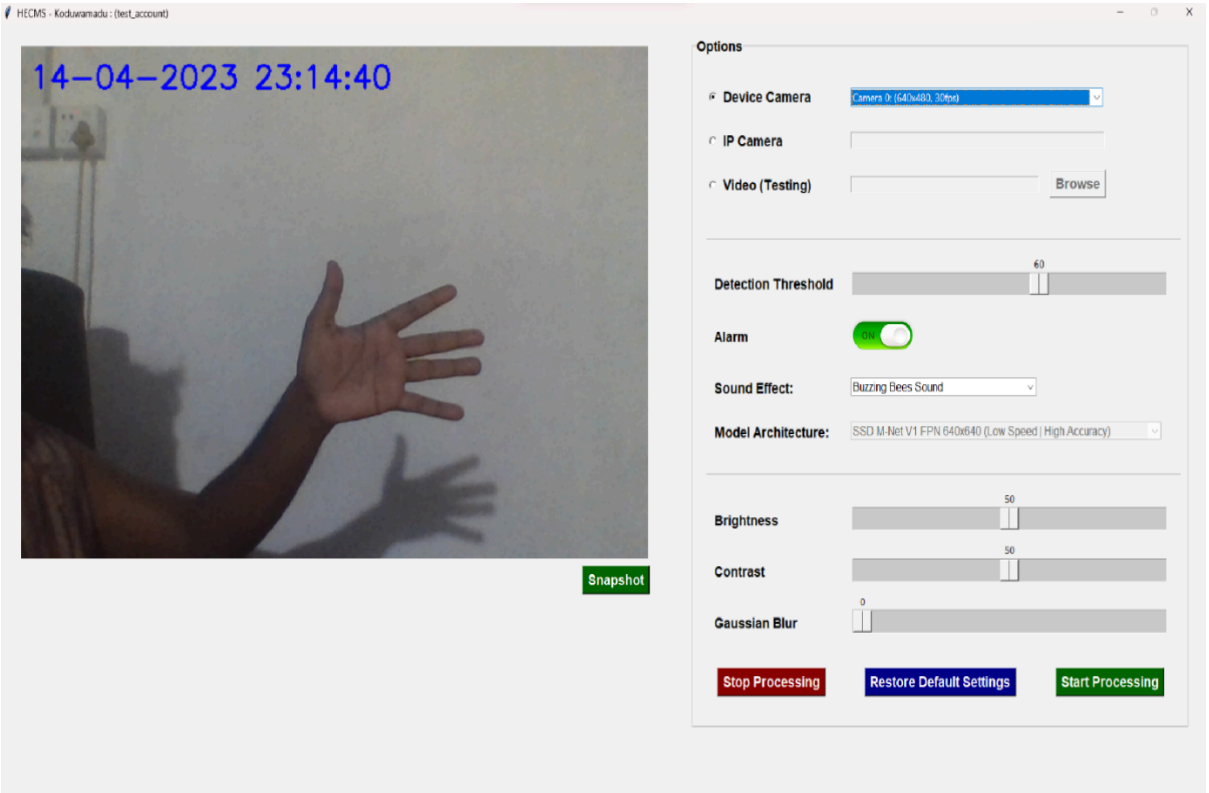


Figure 43 Dashboard – Monitoring System Runs with Camera (Results)

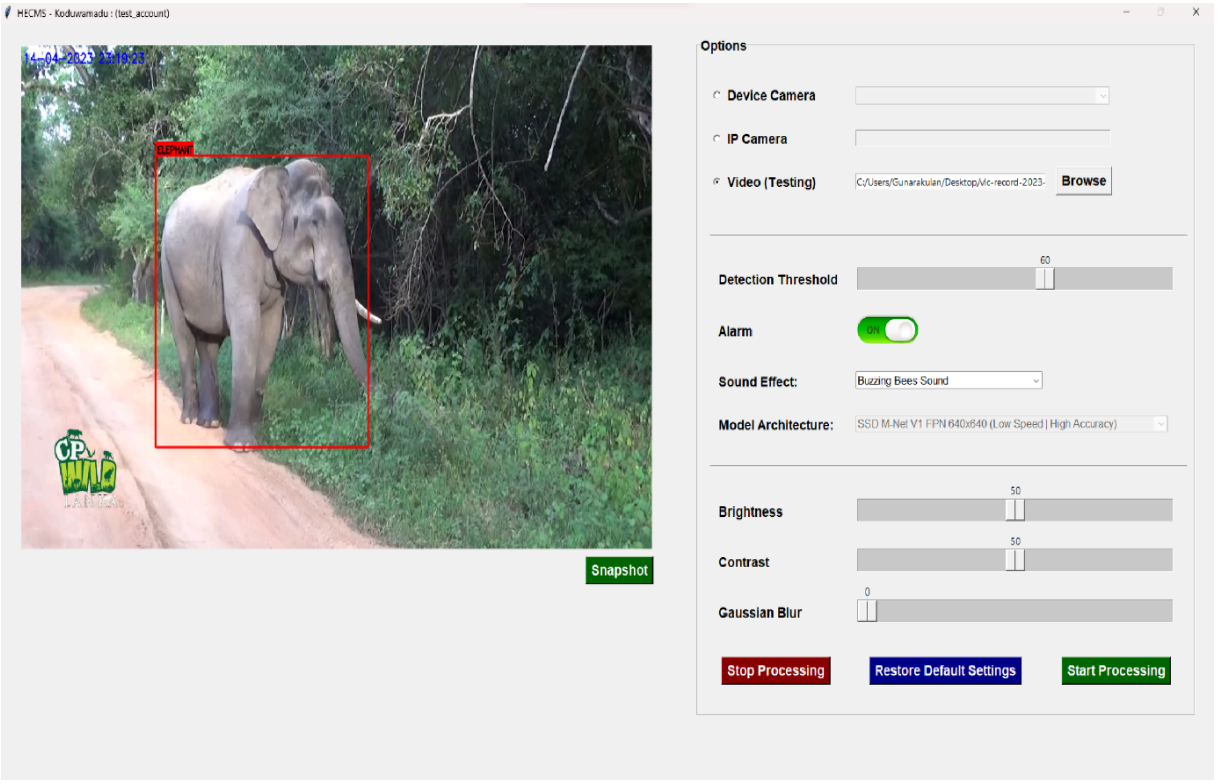


Figure 44 Monitoring System Runs with Pre-recorded Videos (Results) Image 1

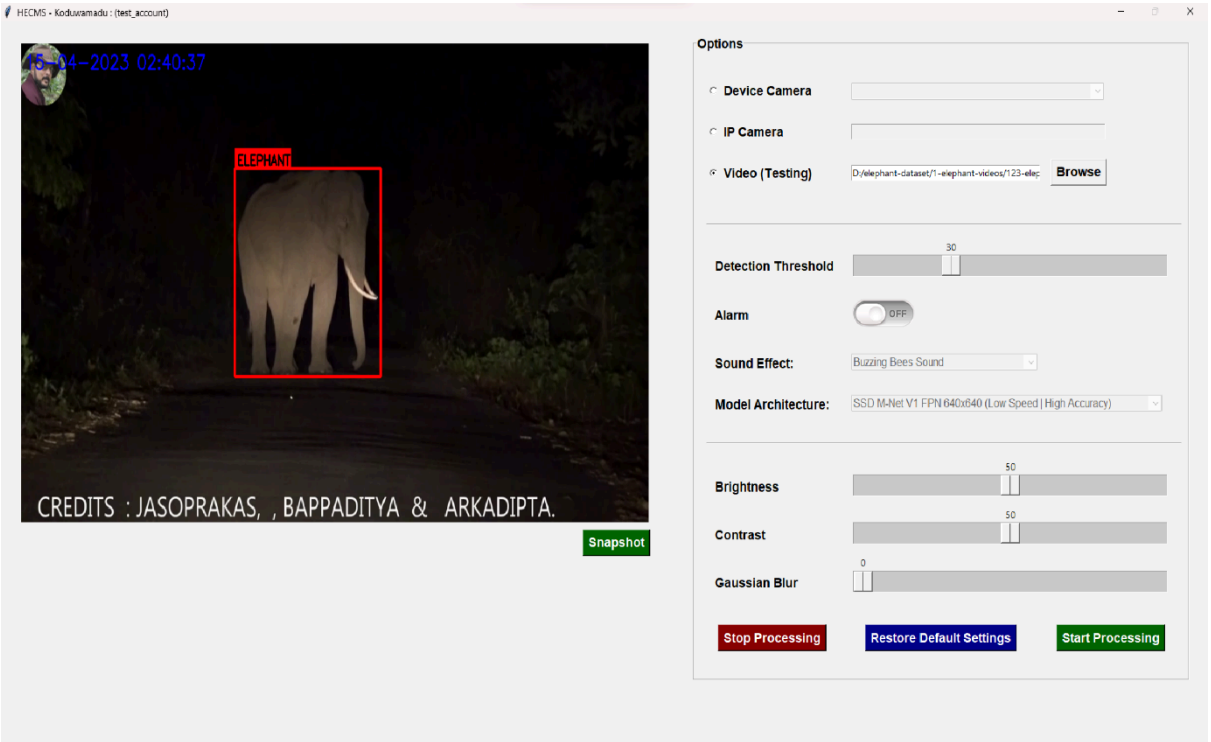
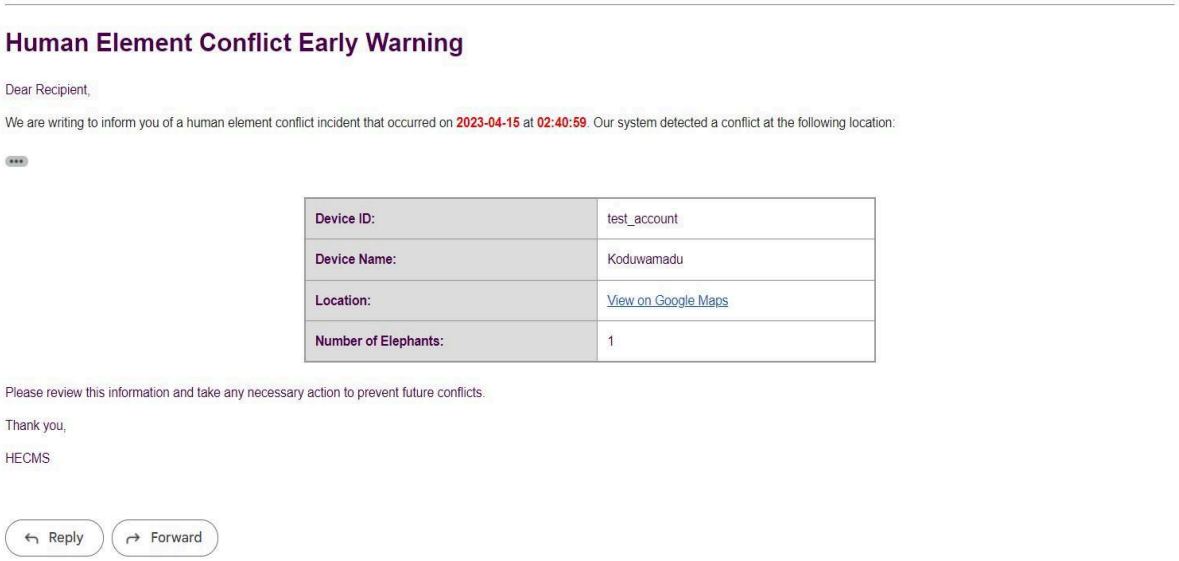


Figure 45 Monitoring System Runs with Pre-recorded Videos (Results) Image 2



Human Element Conflict Early Warning

Dear Recipient,

We are writing to inform you of a human element conflict incident that occurred on **2023-04-15** at **02:40:59**. Our system detected a conflict at the following location:

Device ID:	test_account
Device Name:	Koduwamadu
Location:	View on Google Maps
Number of Elephants:	1

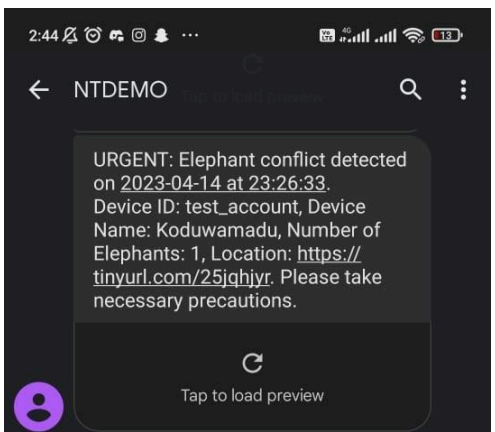
Please review this information and take any necessary action to prevent future conflicts.

Thank you,

HECMS

Reply Forward

Figure 46 Monitoring System Early Warning Email



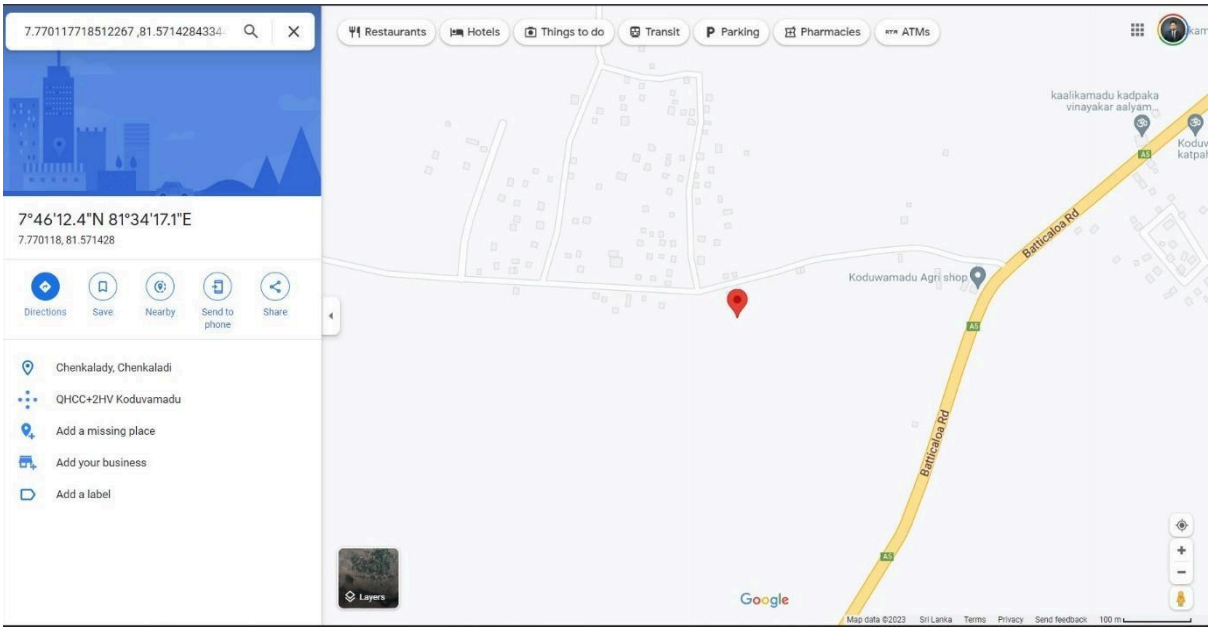
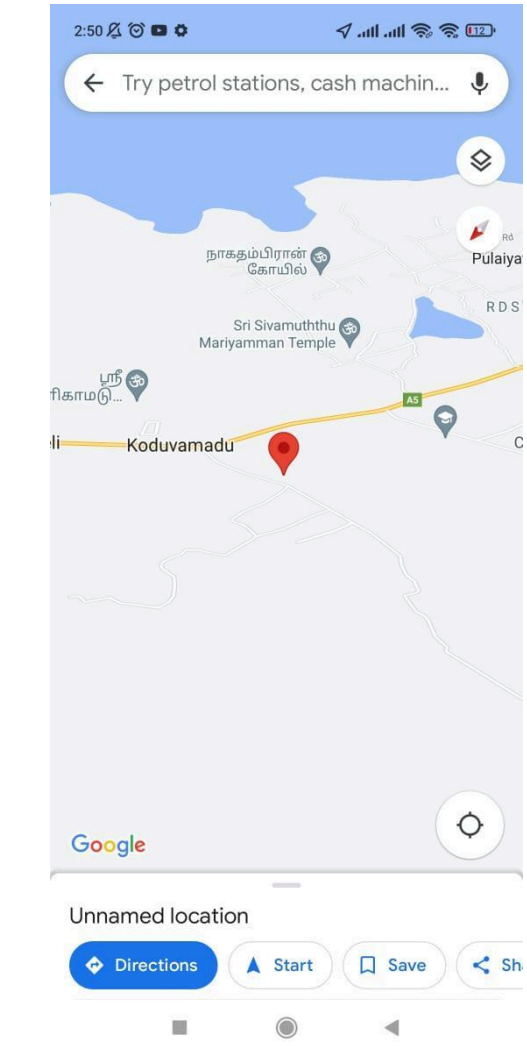


Figure 48 Monitoring System Location Sent via Email



4.2.2 Management System

The management system is the secondary system used by the head office of the conflict management team to analyze the data. It is the core of the system because all the data will be transmitted to it and even monitoring system login accounts can be one created in the management system. There are two accounts in the management system, those are Super Admin is only used by the head office, the other one is Device Admin, which could be created by the management system and can be used in the monitoring system and well as in the management system to see its detection data.

4.2.2.1 Login Page – Management System

The Management System login page has two different account types: Super Admin and Device Admin.

Features

- Able to select which account type that is wanted to be logged in.
- Successful login when credentials are matched.
- Display an error message when a wrong credential is provided.
- Clear textboxes automatically after a wrong credential is provided.

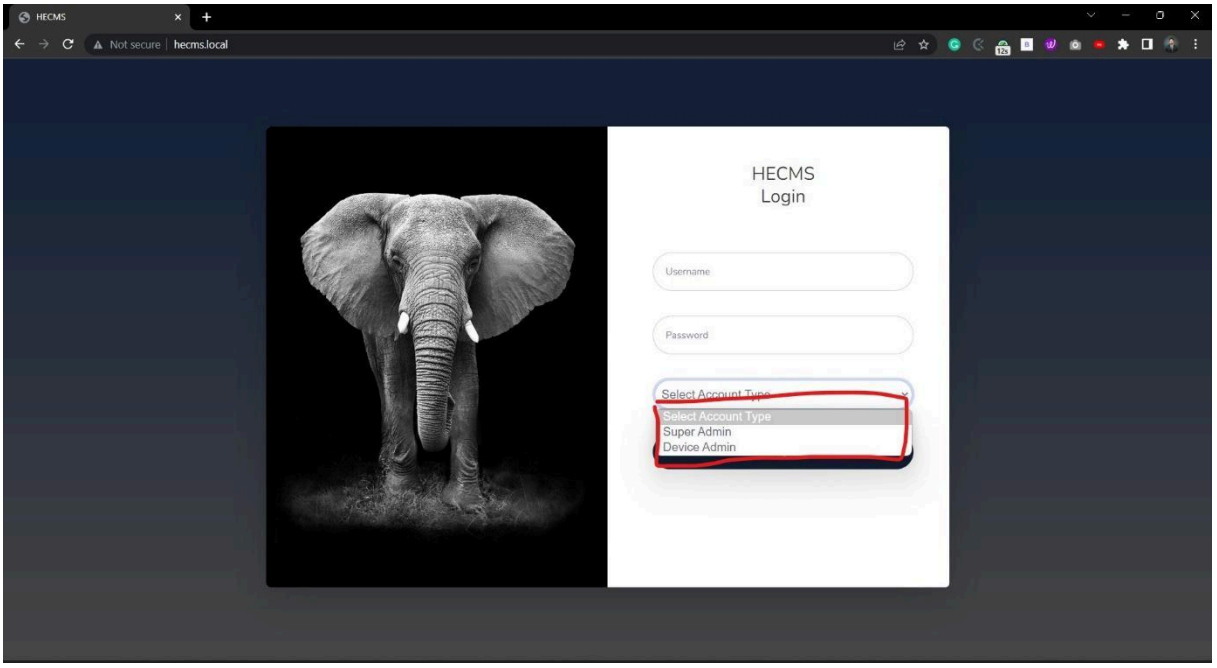


Figure 50 Login Page – Management System (Results)

4.2.2.2 Homepage (Super Admin) – Management System

This is the Management System Homepage, also known as the dashboard of the Super Admin; it has so many features and functionalities. A line chart plots the elephant data by monthly; This Super Admin will get all the data transmitted by all the registered devices. Apart from the functional-based features, there are many UI-based features, such as an animative line chart, nice fade-in / face-out effects on hovering, etc. All the functional features are listed below.

The sidebar has all the Super Admin pages, such as Data Management, Device Management, and Settings. Data management has all the data transmitted by all the devices; device management has all the created device information and has functionalities like creating new devices, deleting devices editing devices and viewing device locations, etc.

Features

- Plot all the detection data on a monthly wise line chart.
- A monthly-wise filter is built on the right top of the line chart.
- Display the total number of devices created in the system.
- Display the total number of elephant conflict incidents identified by all devices.
- Display the total number of elephants detected by all devices.

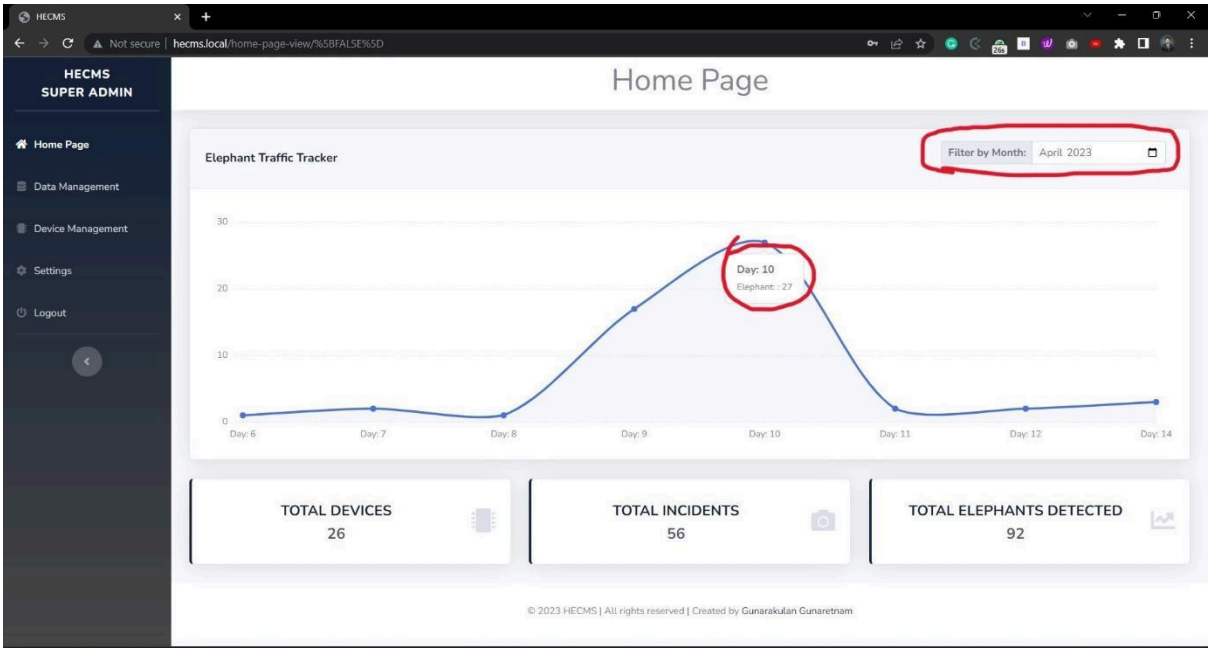


Figure 51 Homepage (Super Admin) – Management System (Results)

4.2.2.3 Data Management Page (Super Admin) – Management System

This is the Management System data management page of Super Admin; This page fetches all the human-elephant conflicts data from the database that were transmitted by all the devices. It has so many features to view data. This page has a table with columns such as Device ID, Date, Time, Number of Elephants Detected, and Image; It has a built-in day-wise filter option to fetch data from the database on a specific date, and there is an option to view image on new tap to see it more clear with zoom options.

Features

- Show all conflict incidents data captured by all devices in the table.
- Have a pagination panel that splits data by 15 to speed up the system.
- Have a day-wise filter to list by a specific date.
- Have a radio button to switch between filter options.
- Show all data as a radio button option.
- Have a feature to view the incident image on a new window with zoom ability.

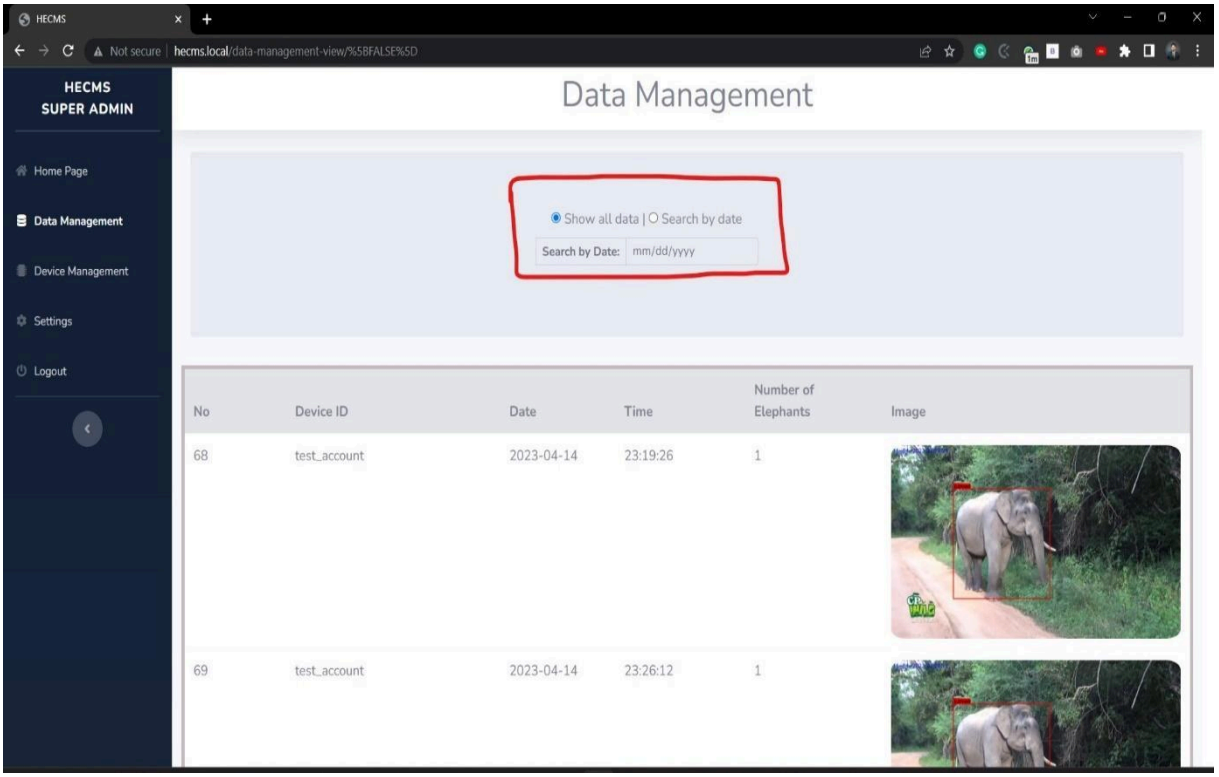


Figure 52 Data Management Page (Super Admin) – Management System (Results) Image 1

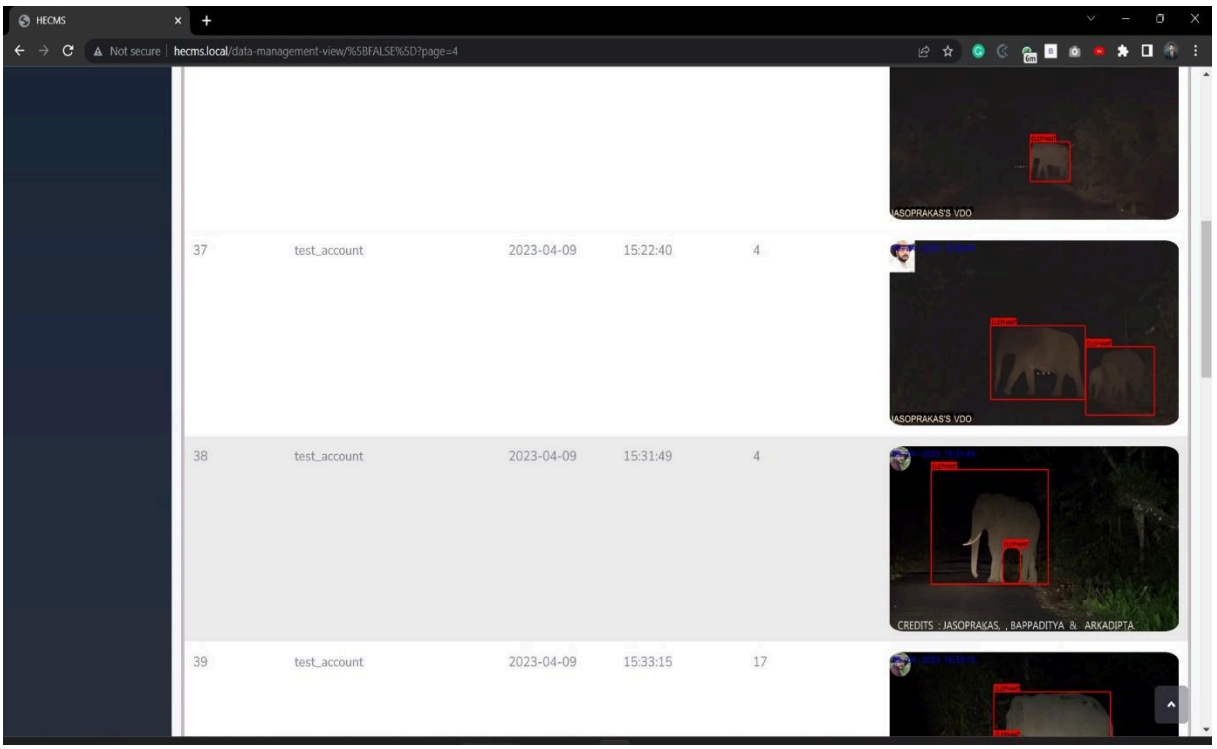


Figure 53 Data Management Page (Super Admin) – Management System (Results) Image 2

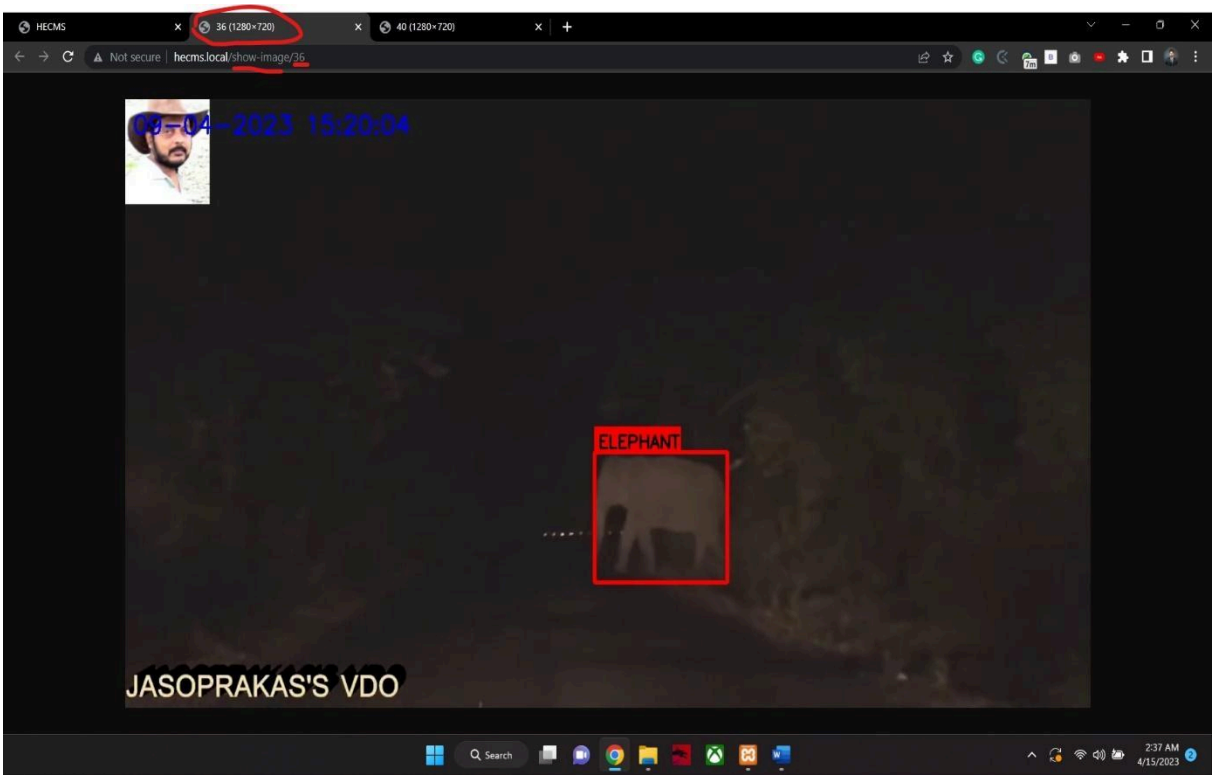


Figure 54 Data Management Page (Super Admin) – Management System (Results) Image 3 (View Image in New Tab)

4.2.2.4 Device Management Page (Super Admin) – Management System

This is the Management System device management page of Super Admin; It is the core part of the system because device creation, deletion, updating, all device-related functions, and data are managed by this page.

Features

- Show all the device data in a table.
- Have a feature to create a new device.
- Have the option to delete devices.
- Have the option to edit/update devices.
- Have a pagination panel that splits data by 15 to speed up the system.
- Have a feature to ask for confirmation before deleting devices completely.
- Have the option to view the device's location on Google Maps.
- Have a feature to show a successful message when a successful operation occurs.
- Have a feature to show an error message when an error occurs.

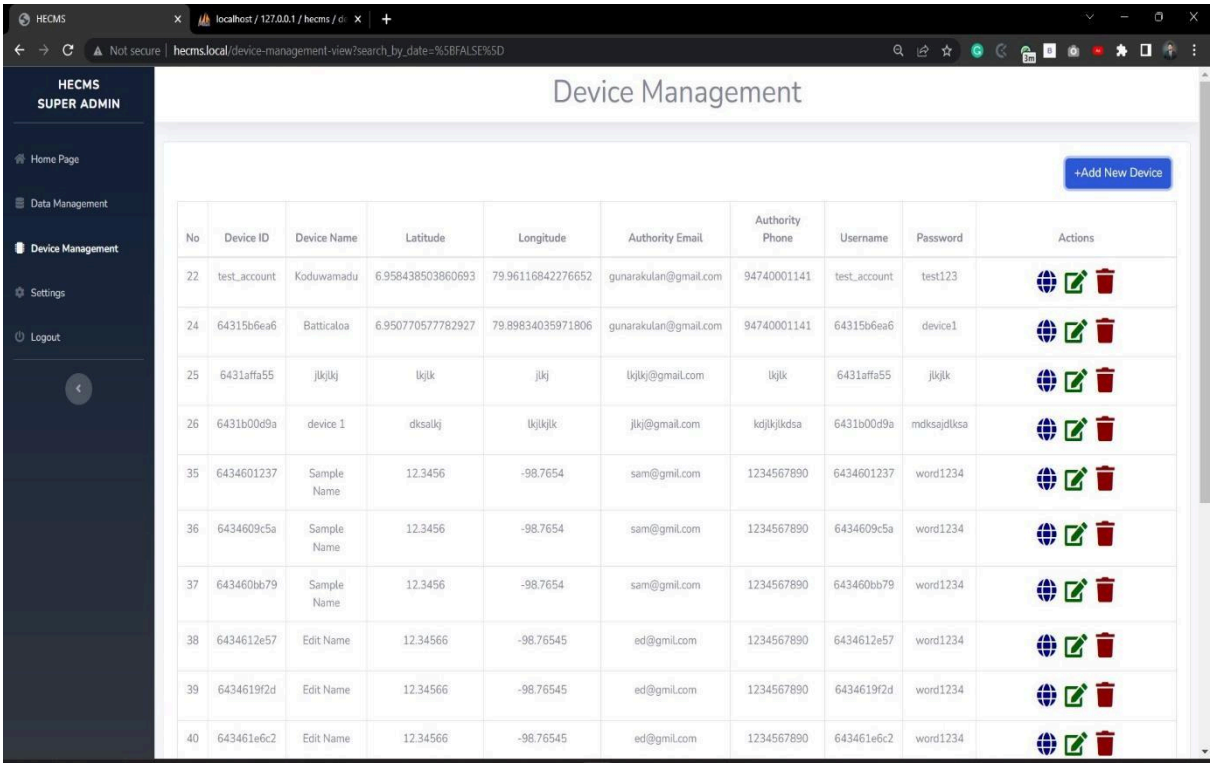


Figure 55 Device Management Page (Super Admin) – Management System (Results)

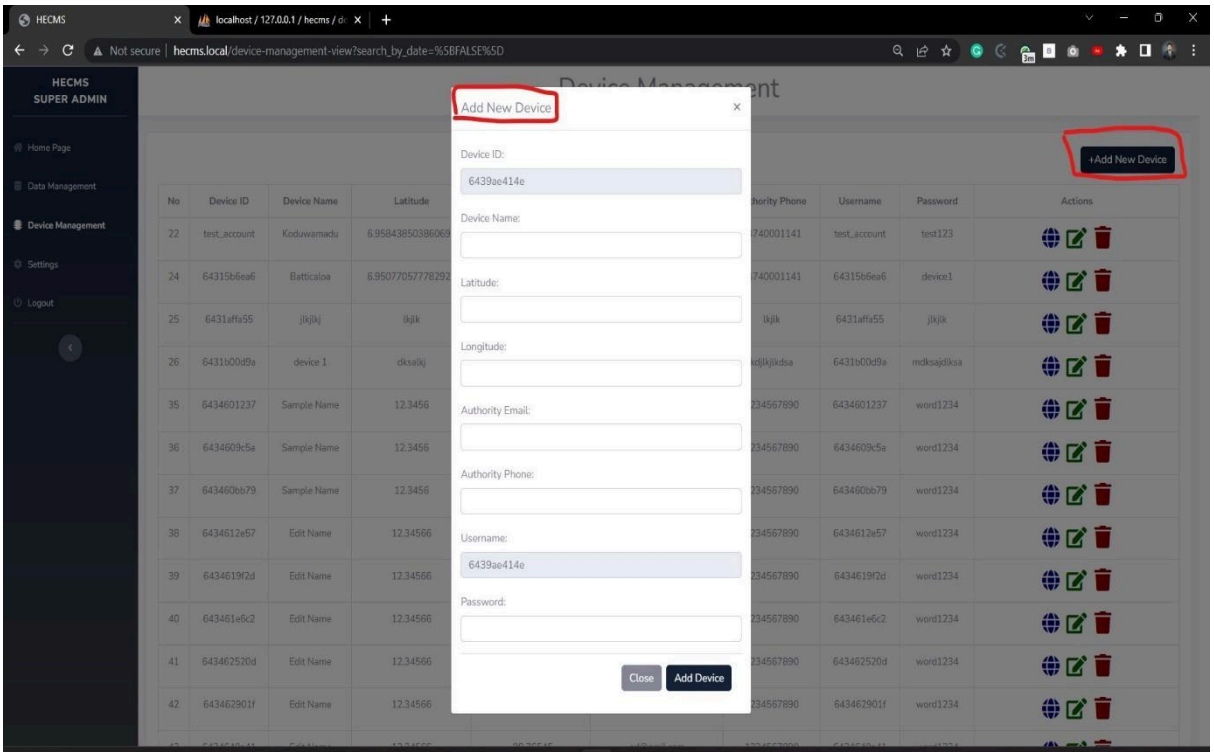


Figure 56 Device Management Page (Super Admin) – Management System (Create a new Device) (Results)

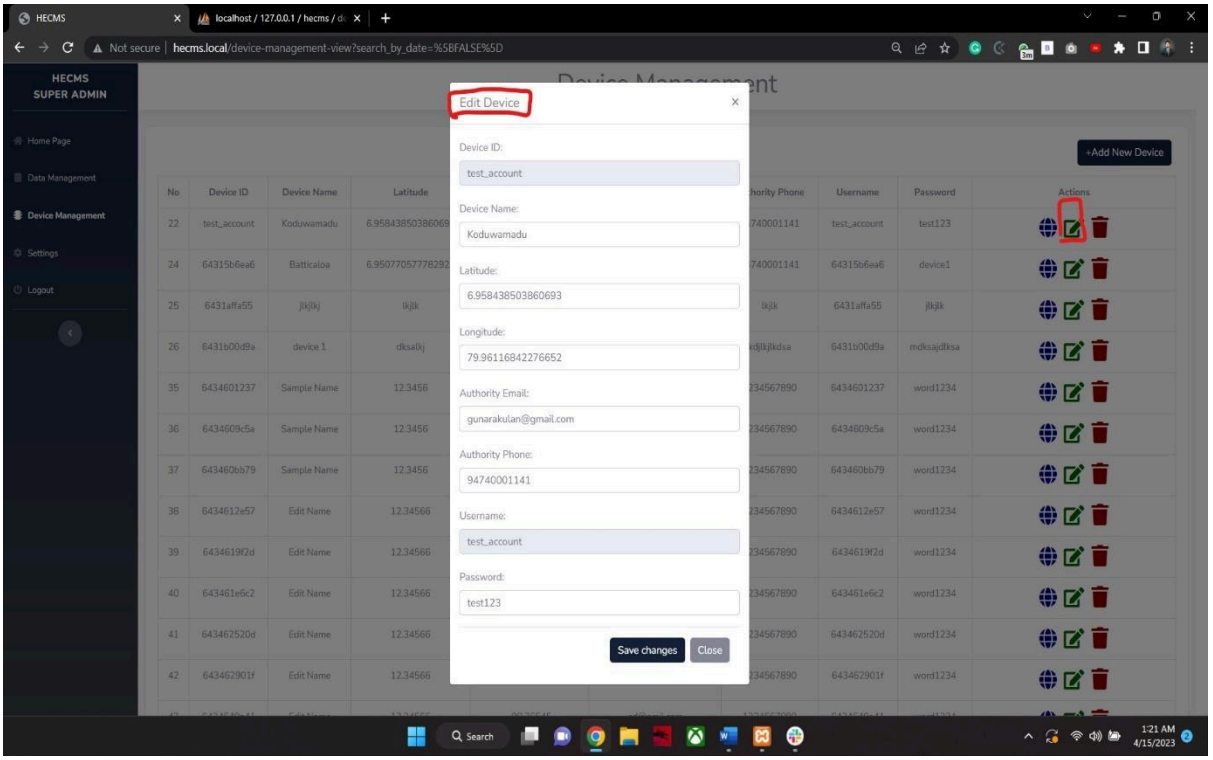


Figure 57 Device Management Page (Super Admin) – Management System (Edit Device) (Results)

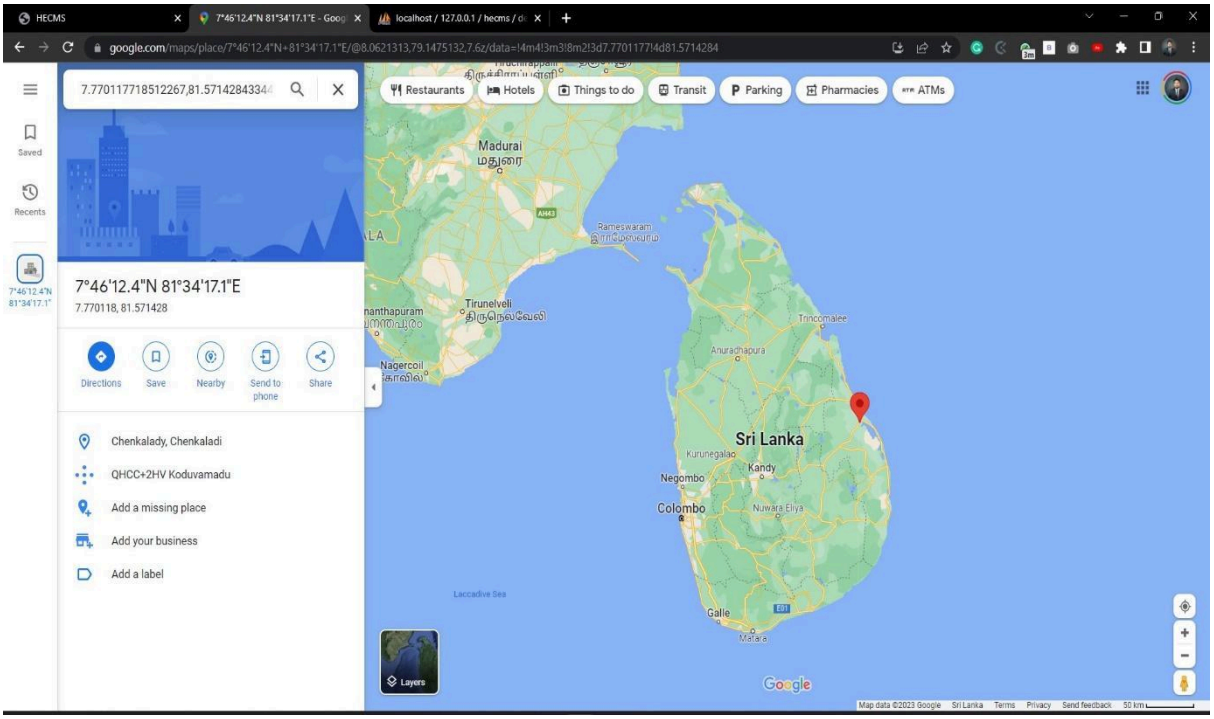


Figure 58 Device Management Page (Super Admin) – Management System (Device Location) (Results) Image 1

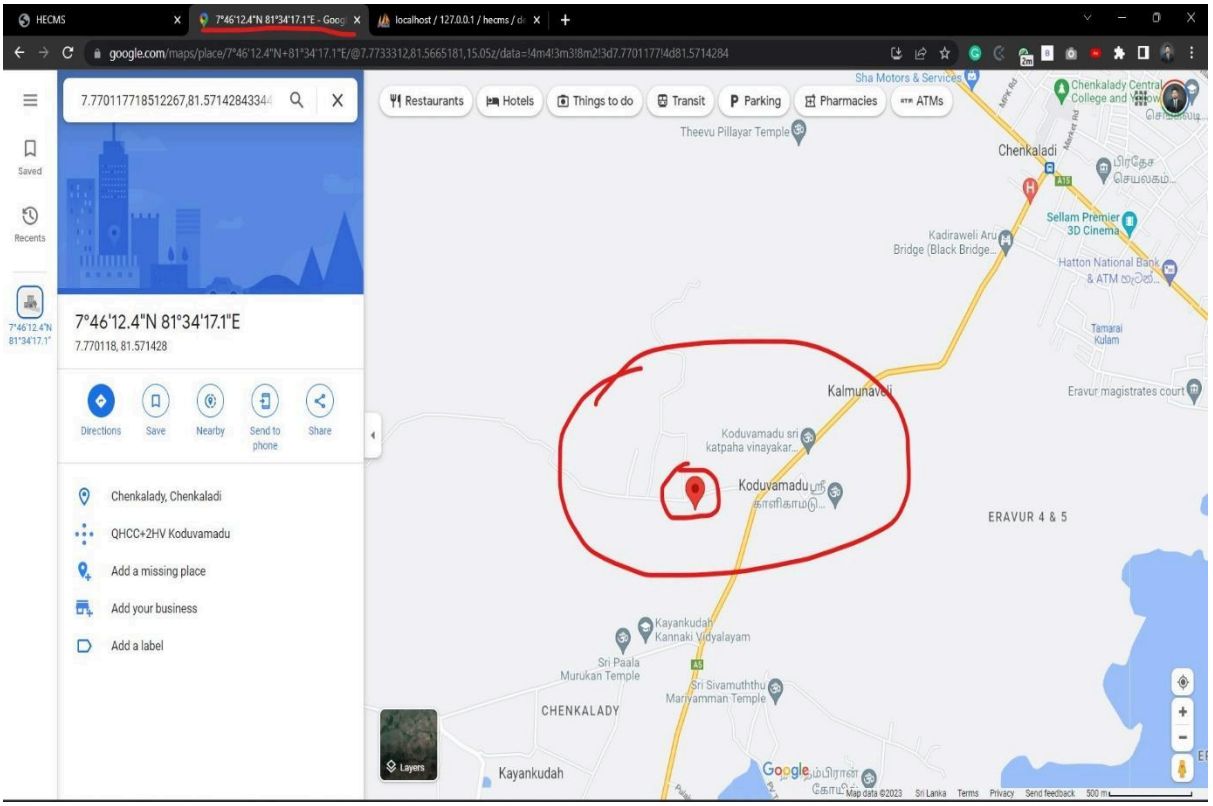


Figure 59 Device Management Page (Super Admin) – Management System (Device Location) (Results) Image 2

4.2.2.5 Device Settings Page (Super Admin) – Management System

This is the Management System setting page of the Super Admin; it has features to change the password of the Super Admin; in the future, a feature will be added to the settings page to make it more standardized.

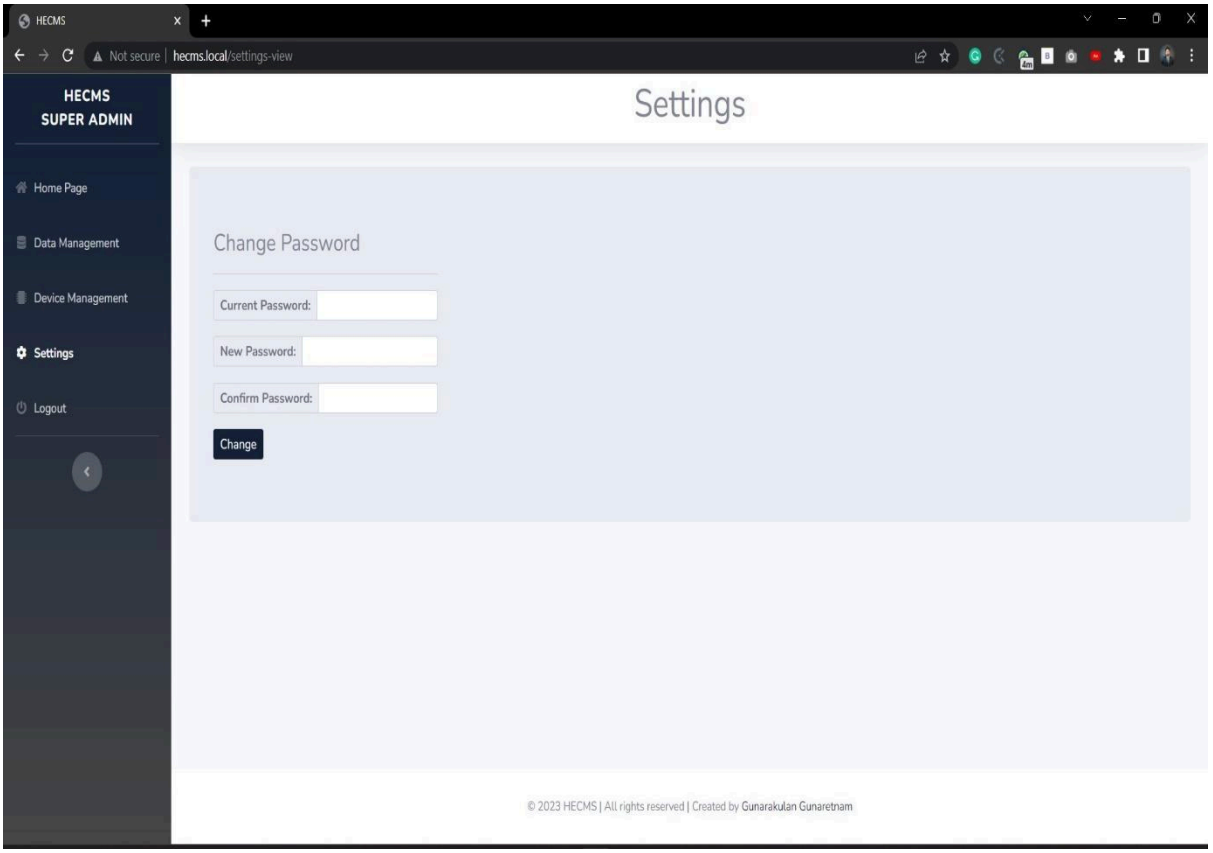


Figure 60 Device Settings Page (Super Admin) – Management System

Features

- Change the password of the Super Admin.
- Have a feature to show a successful message when a successful operation occurs.
- Have a feature to show an error message when an error occurs.
- Have a feature to check the confirmation password.
- Have a feature to check the current password.

4.2.2.6 Homepage (Device Admin) – Management System

This is the Management System Homepage, also known as the dashboard of the Device Admin; it has so many features and functionalities related to the device. A line chart plots the elephant data monthly; This Device Admin will only get the data transmitted by its thin client. Apart from the functional-based features, there are many UI-based features, such as an animative line chart, nice fade-in / face-out effects on hovering, etc. All the functional parts are listed below.

The sidebar has all the Device Admin pages, such as Data Management, Device Preferences, and Settings. Data management has the data transmitted by the current device through a thin client; Device Preferences have information about the current device and can be edited. In the side bar, it also shows the device information such as device name and ID.

Features

- Plot the detection data of the currently logged-in device on a monthly-wise line chart.
- A monthly-wise filter is built on the top of the line chart.
- Have a button to locate the location of the currently logged-in device.
- Display the total number of elephant conflict incidents identified by the current device.
- Display the total number of elephants detected by current device.

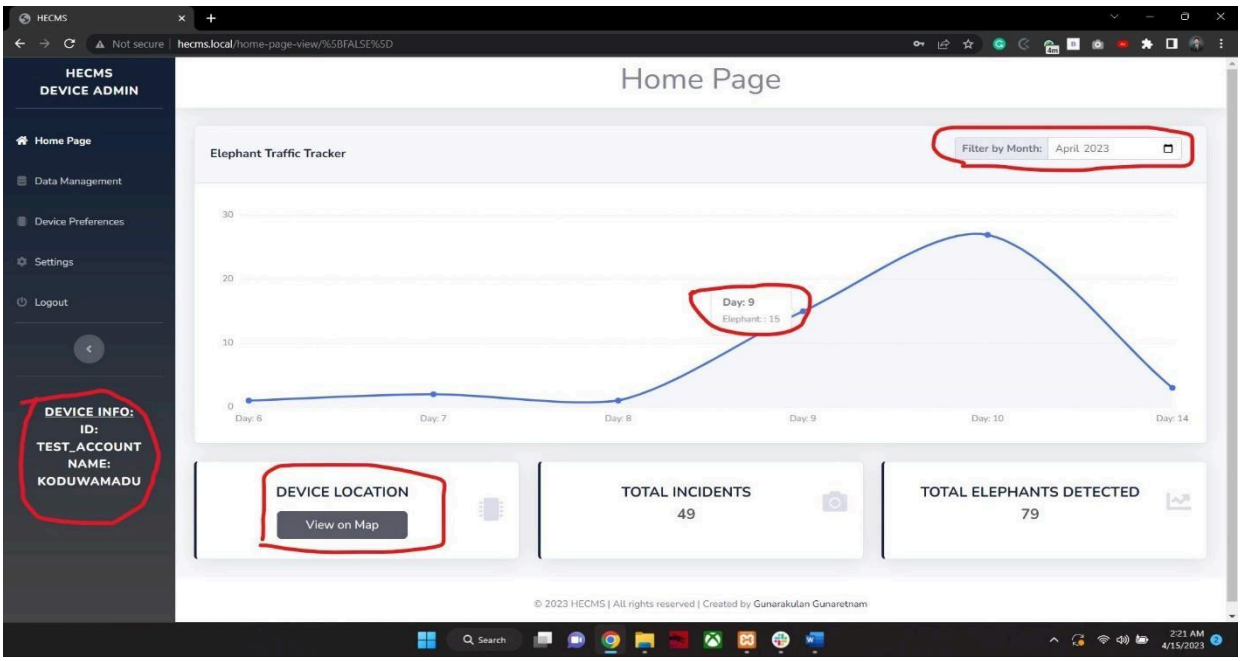


Figure 61 Homepage (Device Admin) – Management System (Results)

4.2.2.7 Data Management Page (Device Admin) – Management System

This is Device Admin's Management System data management page; This page fetches currently logged-in device elephant conflicts data from the database transmitted by its thin client. It has so many features to view data. This page has a table with columns such as Date, Time, Number of Elephants Detected, and Image; It has a built-in day-wise filter option to fetch data from the database on a specific date, and there is an option to view images on new tap to see it more clear with zoom options.

Features

- Show all conflict incidents data captured by the current device in the table.
- Have a pagination panel that splits data by 15 to speed up the system.
- Have a day-wise filter to list by a specific date.
- Have a radio button to switch between filter options.
- Show all data of the current device as a radio button option.
- Have a feature to view the incident image on a new window with zoom ability.

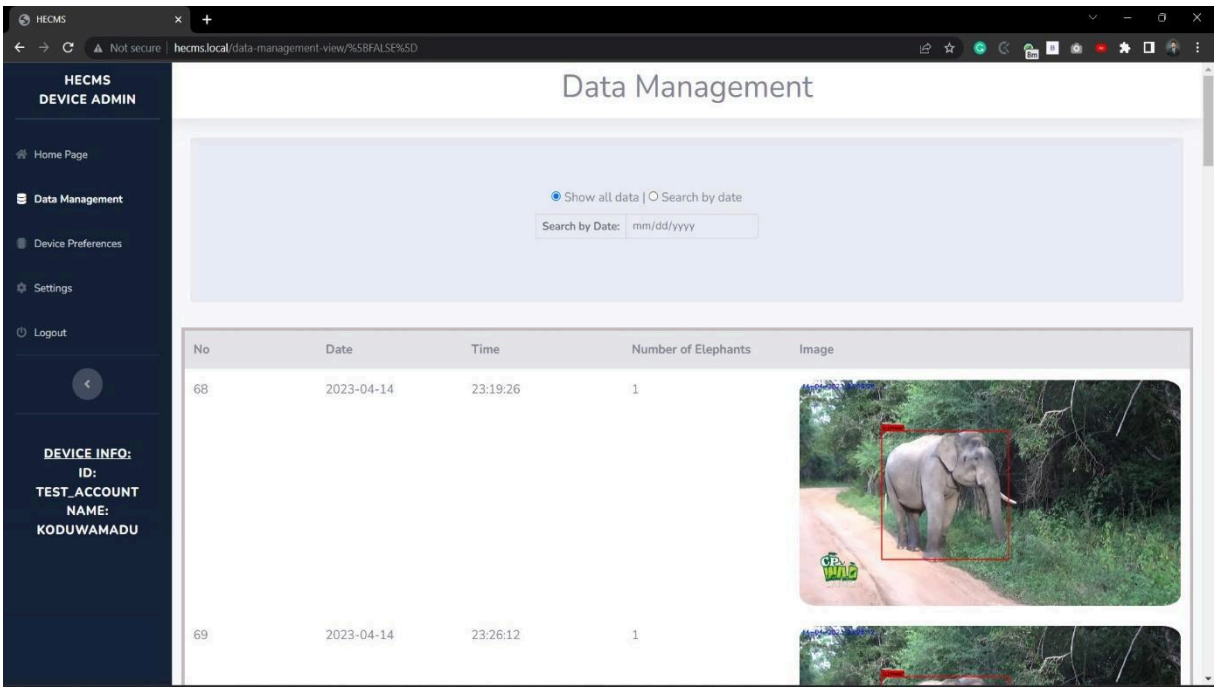


Figure 62 Data Management Page (Device Admin) – Management System (Results) Image 1

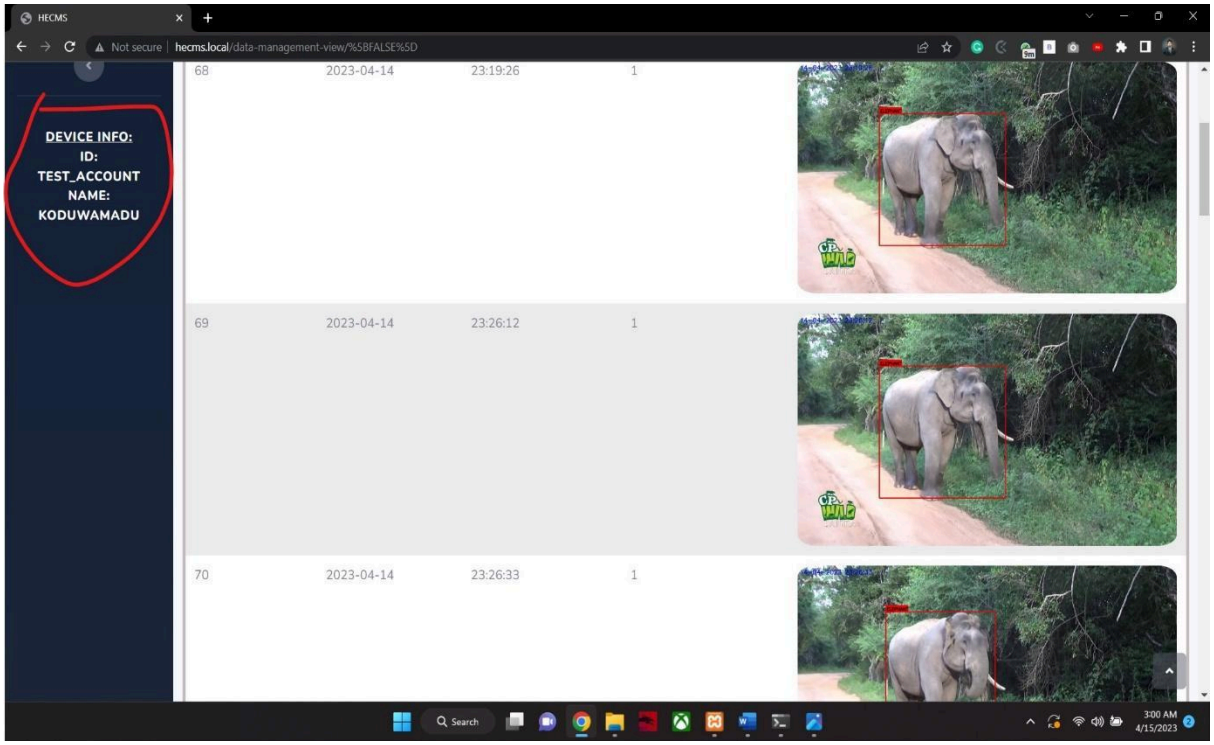


Figure 63 Data Management Page (Device Admin) – Management System (Results) Image 2

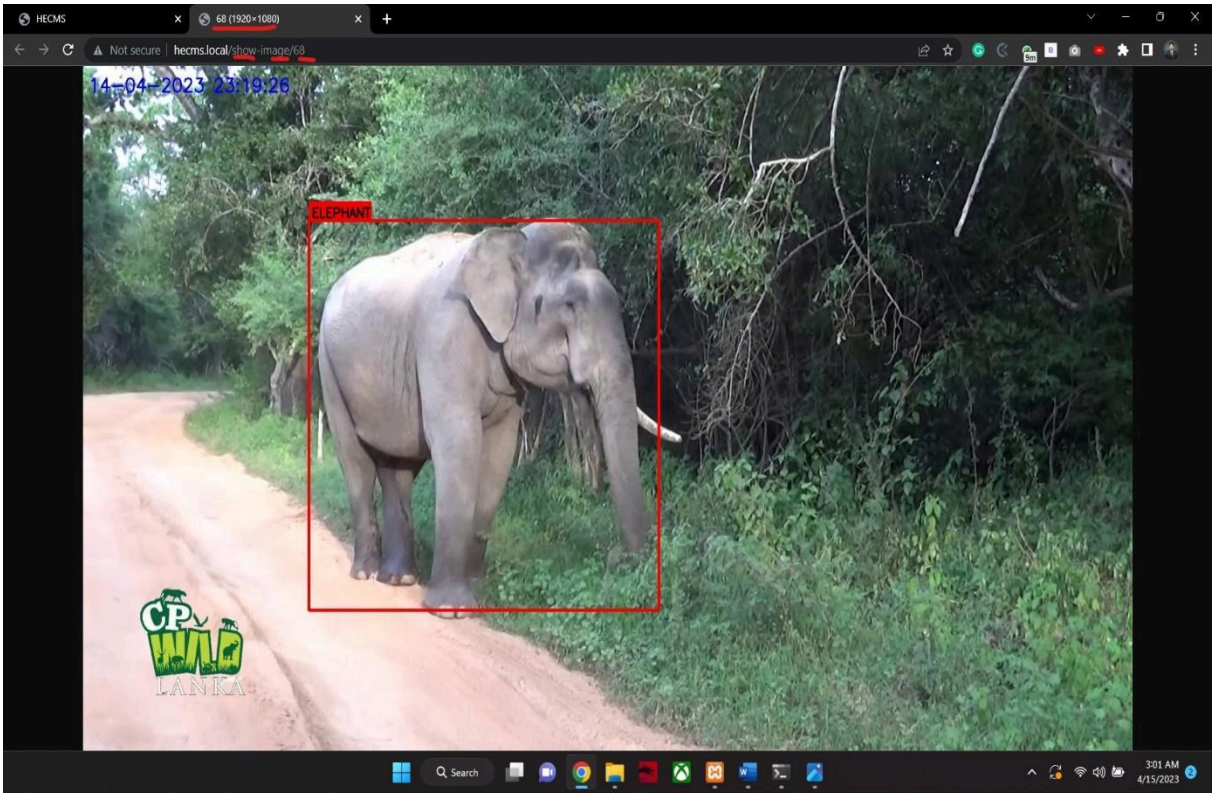


Figure 64 Data Management Page (Device Admin) – Management System (Results) Image 3 (View Image in New Tab)

4.2.2.8 Device Preferences Page (Device Admin) – Management System

This is Device Admin's Management System device preferences page; It is used to update/edit current device information.

Features

- Show the current device information.
- Have an option to edit device data.
- Have a feature to show a successful message when a successful operation occurs.
- Have a feature to show an error message when an error occurs.

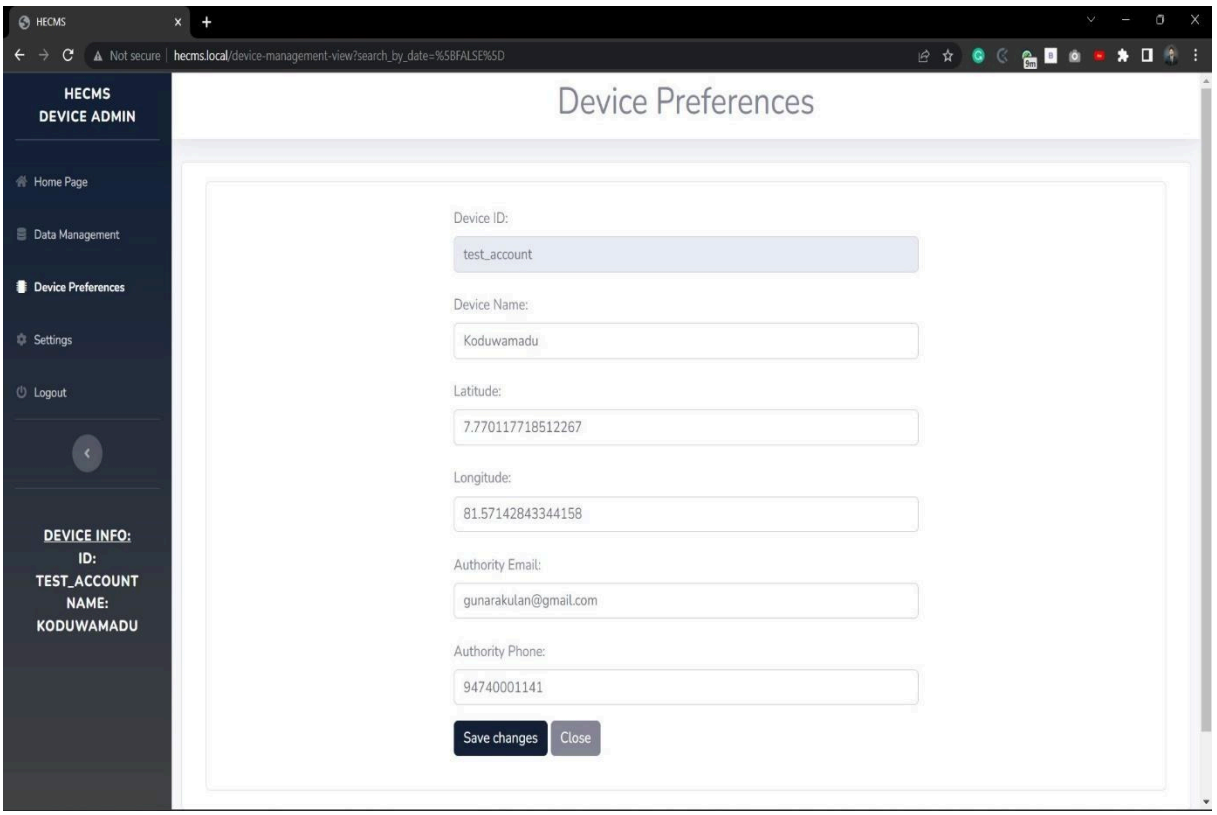


Figure 65 Device Preferences Page (Device Admin) – Management System (Results)

4.2.2.9 Device Settings Page (Device Admin) – Management System

This is the Management System setting page of the Device Admin; it has features to change the password of the currently logged-in Device Admin; in the future, a part will be added to the settings page to make it more standardized.

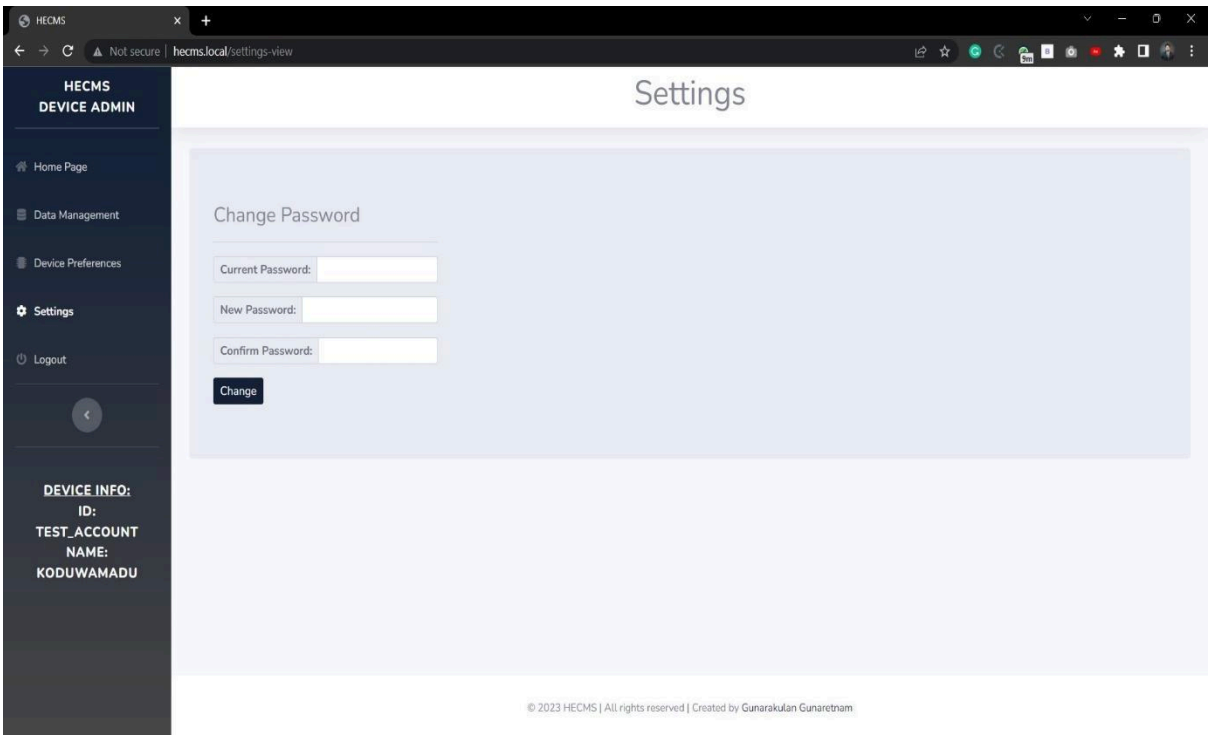


Figure 66 Device Settings Page (Device Admin) – Management System

Features

- Change the password of the Super Admin.
- Have a feature to show a successful message when a successful operation occurs.
- Have a feature to show an error message when an error occurs.
- Have a feature to check the confirmation password.
- Have a feature to check the current password.

4.3 Project Testing

The previous section described the outcome of the research. This part will test the output system and many testing methodologies applied to the system to evaluate its performance. Automatic, manual, and security testing was done against the system to assess its performance and accuracy.

Automated testing techniques were used to evaluate the system's performance. Mechanical tests were designed to check the system's output against expected outcomes. This involved running many test cases with different input scenarios to validate the accuracy and consistency of the system's responses. Automated testing tools were used to automate the testing process, which helped identify any potential issues or inconsistencies in the system's output quickly and efficiently.

Manual testing was also conducted to assess the Human Elephant Conflict Mitigation System's performance. Manually reviewed the system's output to validate its accuracy and ensure it met the desired outcomes. Many testing scenarios were provided as inputs to analyze the corresponding work to verify the system's response. This involved careful examination of the system's output for correctness, relevancy, and coherence. Manual testing helped identify discrepancies or inconsistencies in the system's output that may have yet to be detected through automated testing.

Security testing assesses the system's security measures and identifies Human Elephant Conflict Mitigation System vulnerabilities. This involved testing the system for potential security breaches, such as unauthorized access.

The combination of automated, manual, and security testing helped identify the performance and accuracy of the Human Elephant Conflict Mitigation System. It provides a comprehensive evaluation of the system features. Any issues or inconsistencies identified during the testing phase were addressed and resolved to ensure the system met the desired performance standards and consistently produced accurate output.

4.3.1 Automatic Testing

To perform the automated system testing, **Selenium** and **PyAutoGUI** frameworks were used to write testing scripts in **Python.** Two different separate scripts were written to test both systems. To test the Monitoring System, PyAutoGUI framework was used with Python to write a sequence testing script with the **HTML Parser** to produce the testing report in HTML format. To test the Management System, the Selenium framework was used with Python to write a script with HTML Parser to create the testing report in HTML format.

4.3.1.1 Automatic Testing - Monitoring System

The monitoring system's testing script was in the **artifact folder** of the following directory "test\0-test-case-for-hecms-monitoring-system" directory. Two Python files are in that directory: "main.py" and "utilities.py." main.py is used to run the testing script, and the utilities.py contains the test cases written in **Python** and **PyAutoGUI**. Below, a sample test case function shows that function is used to test the login feature of the monitoring system.

```
1. def test_login_succeed():
2.
3.     time.sleep(3)
4.     pyautogui.press('tab')
5.     pyautogui.typewrite("test_account")
6.
7.     pyautogui.press('tab')
8.     pyautogui.typewrite("test123")
9.     pyautogui.press('tab')
10.
11.        image1 = pyautogui.screenshot(f'{test_folder}\\screensho
   ts\\test-login-succeed-image1.png')
12.        pyautogui.press('space')
13.
14.        time.sleep(1)
15.
16.        image2 = pyautogui.screenshot(f'{test_folder}\\screensho
   ts\\test-login-succeed-image2.png')
17.
18.        if image1 == image2:
19.            return False
20.        else:
21.            return True
22.
```

Figure 67 A sample testing case written in Python and PyAutoGUI to test the login success in Monitoring System

Once the “main.py” test script is completed, it generates a testing folder with the date and time under the following directory “0-test-case-for-hecms-monitoring-system\results”, inside that folder, there will be a file called “test-results.html” and a folder called “screenshots.” The screenshot folder contains testing snapshots, and the test-results.html has the automatic testing result report.

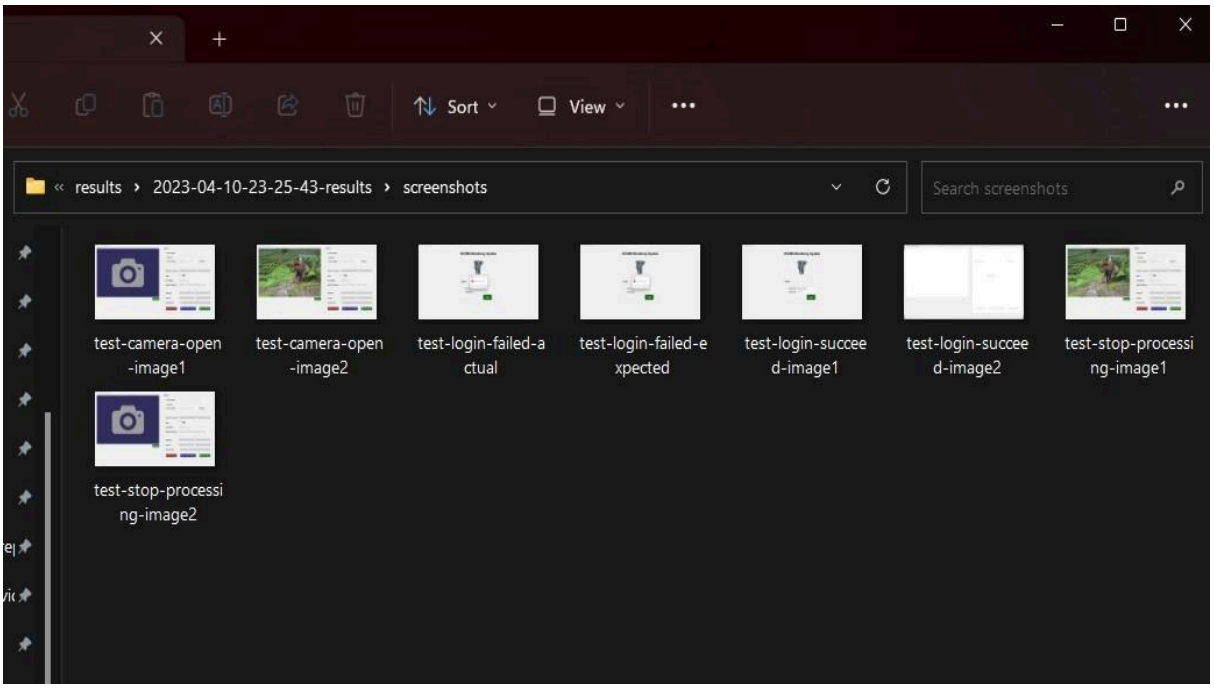


Figure 68 Screenshot folder of the monitoring system testing report directory.

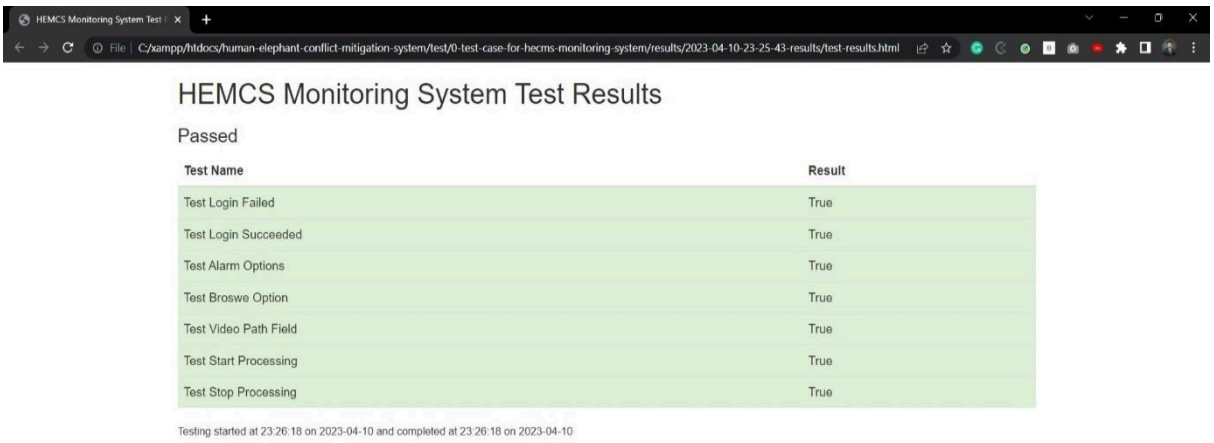


Figure 69 Monitoring System Automatic Testing Report

4.3.1.2 Automatic Testing – Management System

The management system's testing script was in the **artifact folder** of the following directory "test\ 1-test-case-for-hecms-management-system" directory. Two Python files are in that directory: "main.py" and "utilities.py." main.py is used to run the testing script, and the utilities.py contains the test cases written in **Python** and **Selenium**. Below, two-sample test case functions are given: login test case and create a new device test case.

```
1. def test_super_admin_password_passed():
2.
3.   try:
4.           # Find the username field by name and enter the value
5.           username_field = driver.find_element(By.NAME, 'username')
6.           username_field.send_keys('admin')
7.
8.           # Find the password field by name and enter the value
9.           password_field = driver.find_element(By.NAME, 'password')
10.                  password_field.send_keys('admin')
11.
12.                  # Find the account_type select element by name
   and select "Super Admin" option
13.                  account_type_select =
   Select(driver.find_element(By.NAME, 'account_type'))
14.                  account_type_select.select_by_visible_text('Super
   Admin')
15.
```

```
16.                 # Find the login button by name and click it
17.                 login_button = driver.find_element(By.NAME,
    'login')
18.                 login_button.click()
19.                 # Take a screenshot and save it to a file
20.                 screenshot_file =
    f"{test_folder}/screenshots/1-super-admin-login-passed-test.png"
21.                 driver.save_screenshot(screenshot_file)
22.
23.                 return True
24.
25.         except Exception as e:
26.                 return False
```

Figure 70 A sample testing case written in Python and Selenium to test the login function in Management System

```
1. def test_super_admin_create_a_device_function():
2.
3.   try:
4.           # Find the button element
5.           button = WebDriverWait(driver,
    10).until(EC.presence_of_element_located((By.XPATH,
    "//button[text()='+Add New Device']")))
6.           button.click()
7.
8.           time.sleep(2)
9.
10.                 # Find the form fields and enter sample data
11.                 driver.find_element(By.ID,
    "deviceName").send_keys("Sample Name")
12.                 driver.find_element(By.ID,
    "latitude").send_keys("12.3456")
13.                 driver.find_element(By.ID,
    "longitude").send_keys("-98.7654")
14.                 driver.find_element(By.ID,
    "authorityEmail").send_keys("sample@email.com")
15.                 driver.find_element(By.ID,
    "authorityPhone").send_keys("1234567890")
16.                 driver.find_element(By.ID,
    "password").send_keys("word1234")
17.
18.                 screenshot_file =
    f"{test_folder}/screenshots/8-super-admin-device_management_cerat
    e_a_device_function-test.png"
19.                 driver.save_screenshot(screenshot_file)
20.                 # Submit the form
21.                 driver.find_element(By.XPATH,
    "//button[text()='Add Device']").click()
22.
23.                 return True
24.
25.         except Exception as e:
26.                 return False
```

Figure 71 A sample testing case written in Python and Selenium to test the create a new device function in Management System

Once the “main.py” test script is completed, it generates a testing folder with the date and time under the following directory “1-test-case-for-hecms-management-system\results”, inside that folder, there will be a file called “test-results.html” and a folder called “screenshots.” The screenshot folder contains testing snapshots, and the test-results.html has the automatic testing result report.

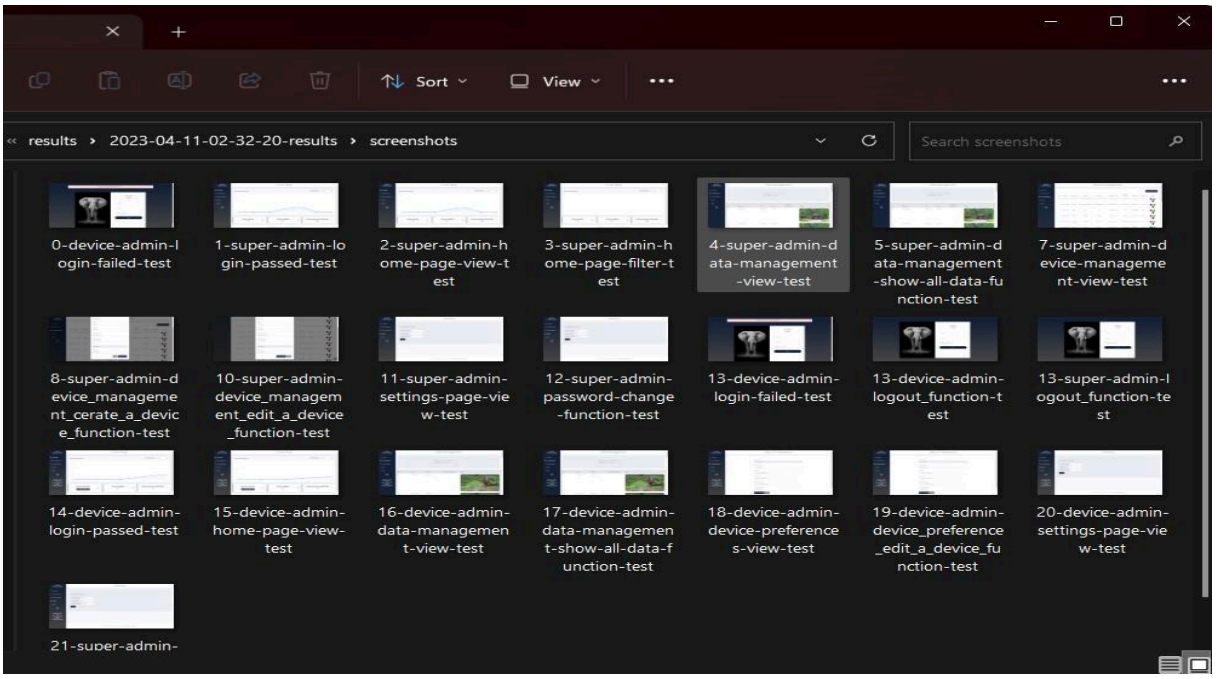


Figure 72 Screenshot folder of the Management System testing report directory

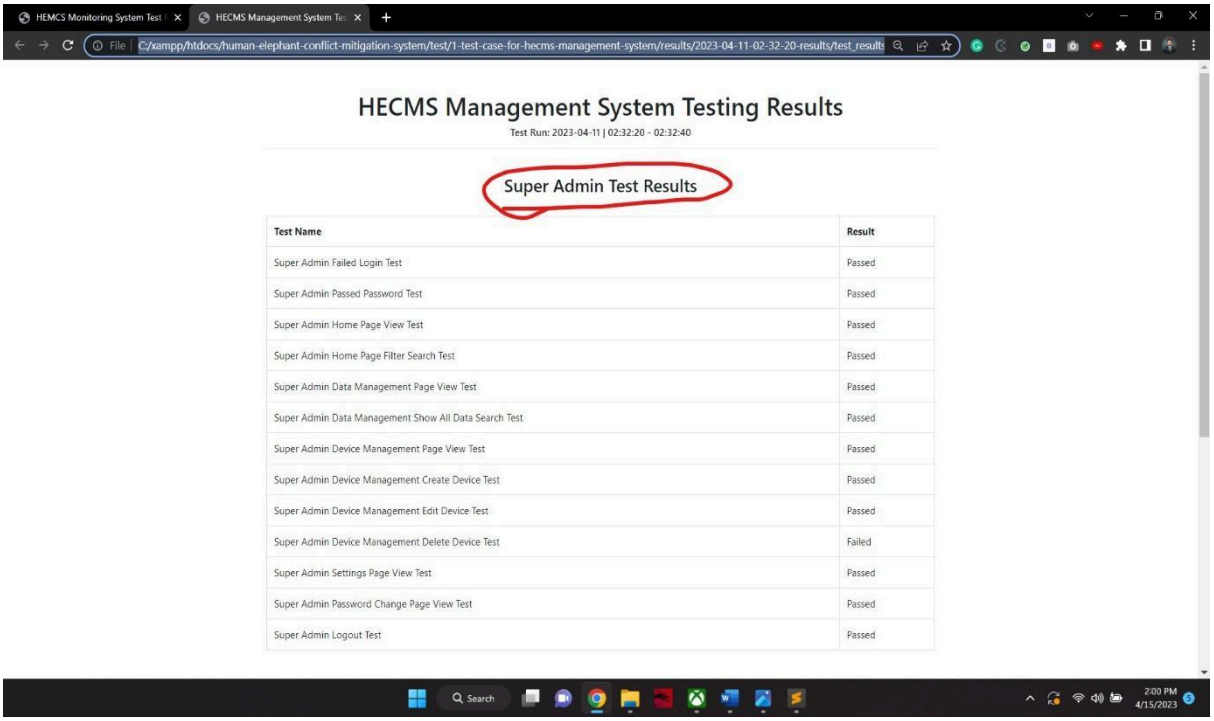


Figure 73 Management System Super Admin Automatic Testing Report

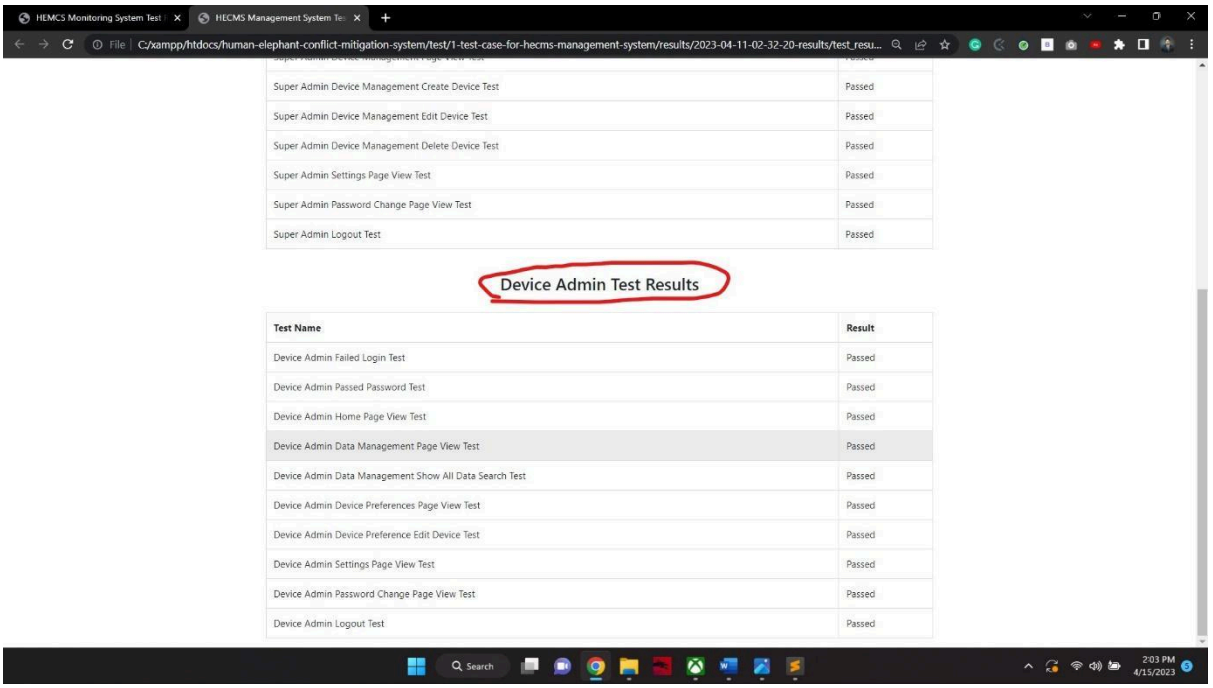


Figure 74 Management System Device Admin Automatic Testing Report

In summary, the automatic testing script was interconnected with the GitHub integration pipeline to test the system every time the system gets a new development update to confirm that all the functionalities are working fine after adding a new feature.

By utilizing the automatic testing script integrated with the GitHub integration pipeline, It can be confidently confirmed that the system functions correctly after every update. This approach helps to identify and fix any issues early in the development process, resulting in a more robust and reliable design.

The automatic testing script was developed for both systems to ensure performance and usability of the system. It helps to organize the system testing processes and saves a much time on manual testing.

After the automatic testing was completed, manual tests were also conducted to verify the system's ability and flexibility further. This combination of automatic and manual testing helps ensure that the system functions as intended and meets the desired performance and usability standards.

4.3.2 Manual Testing

The monitoring and the management systems were tested manually to ensure that they were flexible to use by real users. Below are some testing cases with evidence provided; both system results are found below.

4.3.2.1 Monitoring System Test Cases

Table 1 Test Case 1 - Monitoring System

Test Scenario: Log in to the monitoring system with the wrong credentials.	
Expected Results: Must not allow login into the system; it should say credentials wrong.	
Entities	**Input Value**
Username	Ewer232
Password	Dsadsa3213
Results: Passed	

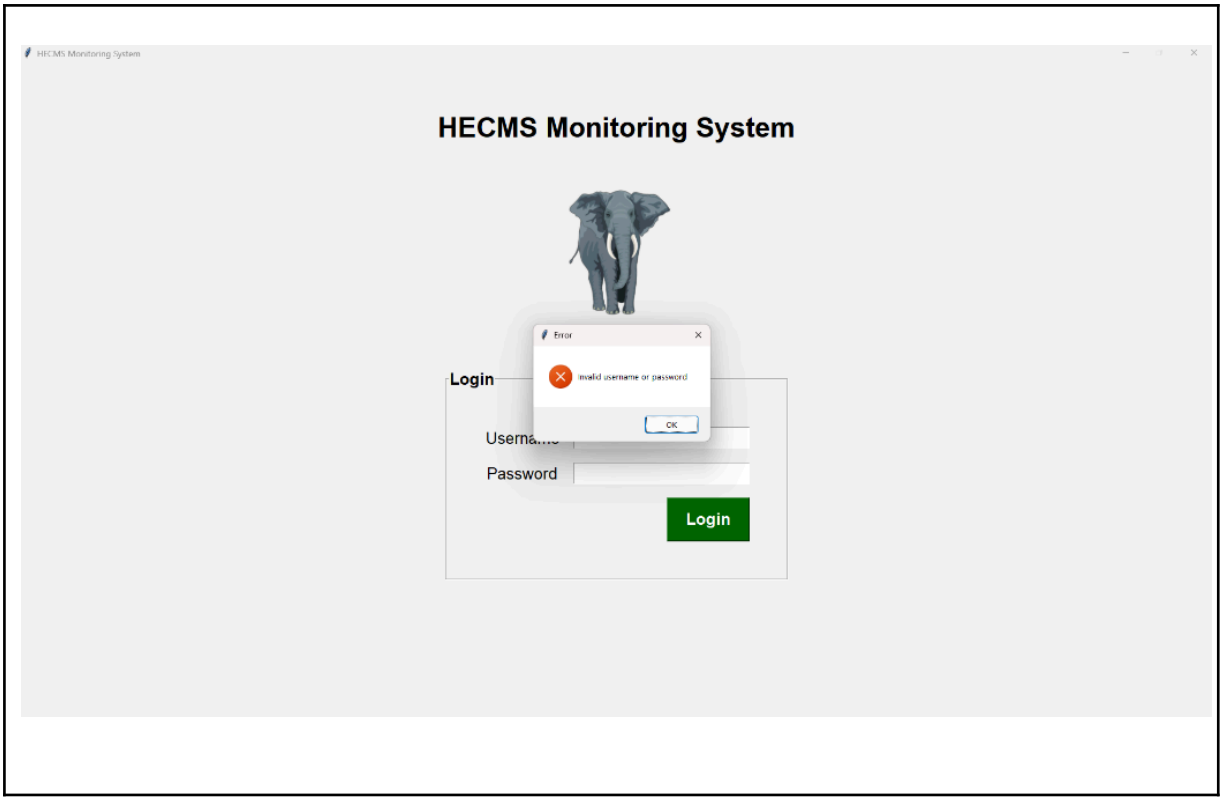


Table 2 Test Case 2 - Monitoring System

Test Scenario: Log in to the monitoring system with the right credentials.	
Expected Results: Must login and enter into the dashboard	
Entities	**Input Value**
Username	test_account
Password	test123
Results: Passed	

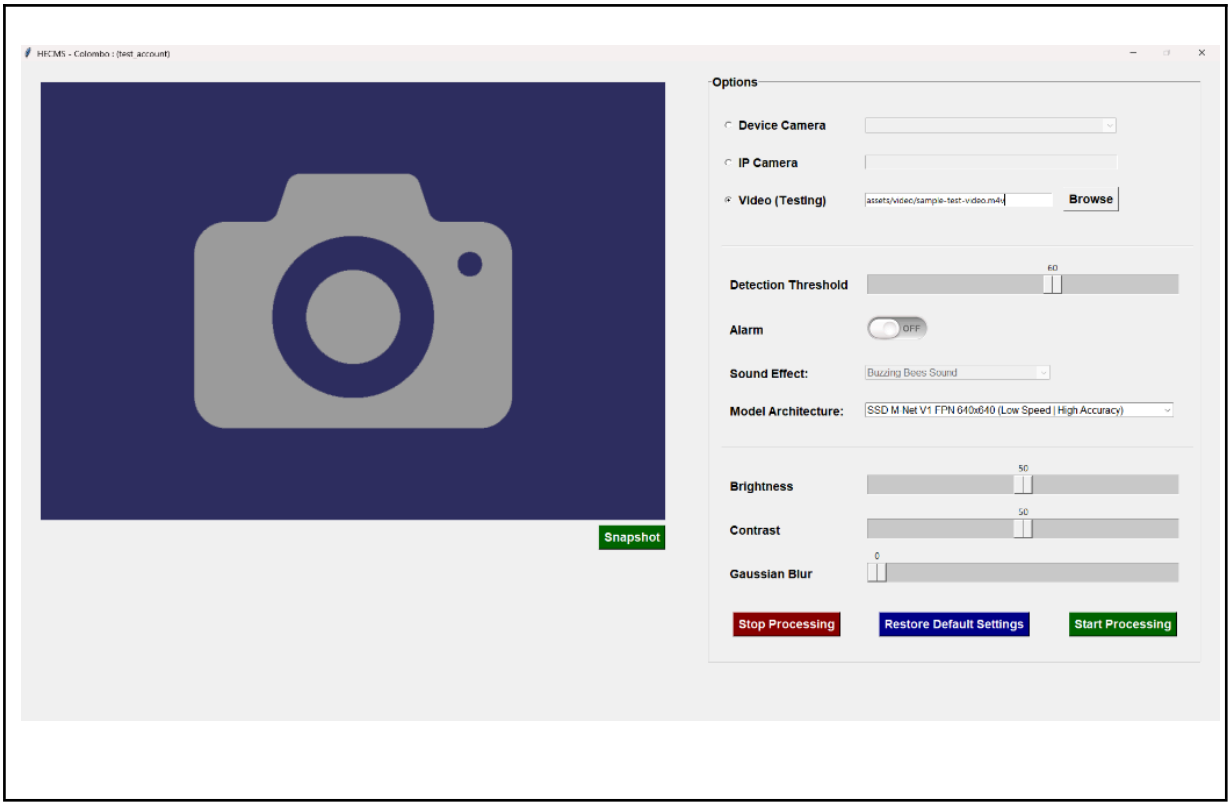


Table 3 Test Case 2 - Monitoring System

Test Scenario: Start the processing without selecting a video source.	
Expected Results: It must say please provide a video source to start the process.	
Entities	**Input Value**
Start Processing Button	CLICKED
Results: Passed	

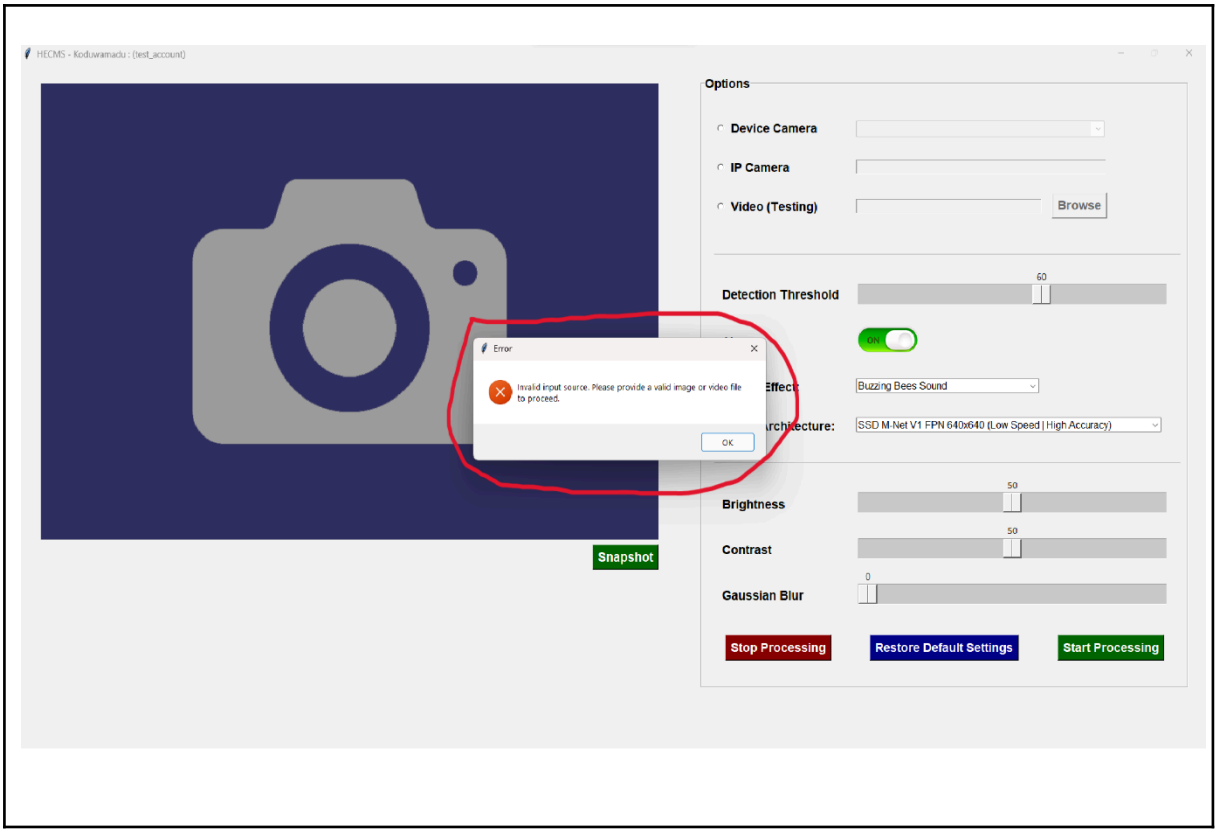


Table 4 Test Case 3 - Monitoring System

Test Scenario: Start the processing by selecting a video source.	
Expected Results: It must start processing.	
Entities	**Input Value**
Start Processing Button	CLICKED
Results: Passed	

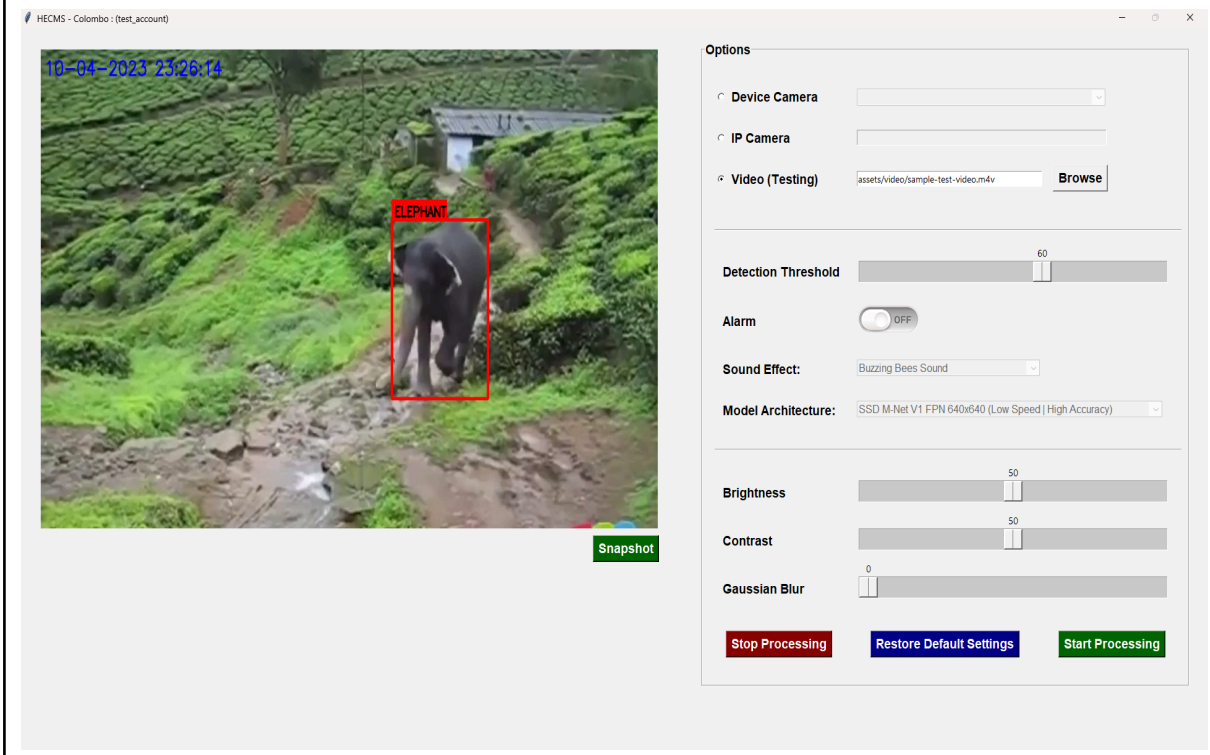


Table 5 Test Case 4 - Monitoring System

Test Scenario: Increase/decrease the detection threshold to test the model accuracy function.	
Expected Results: When decreasing, more false positive results should occur; more false negative results occur when increasing.	
Entities	**Input Value**
Detection Threshold	CHANGED = 0 | CHANGED = 100

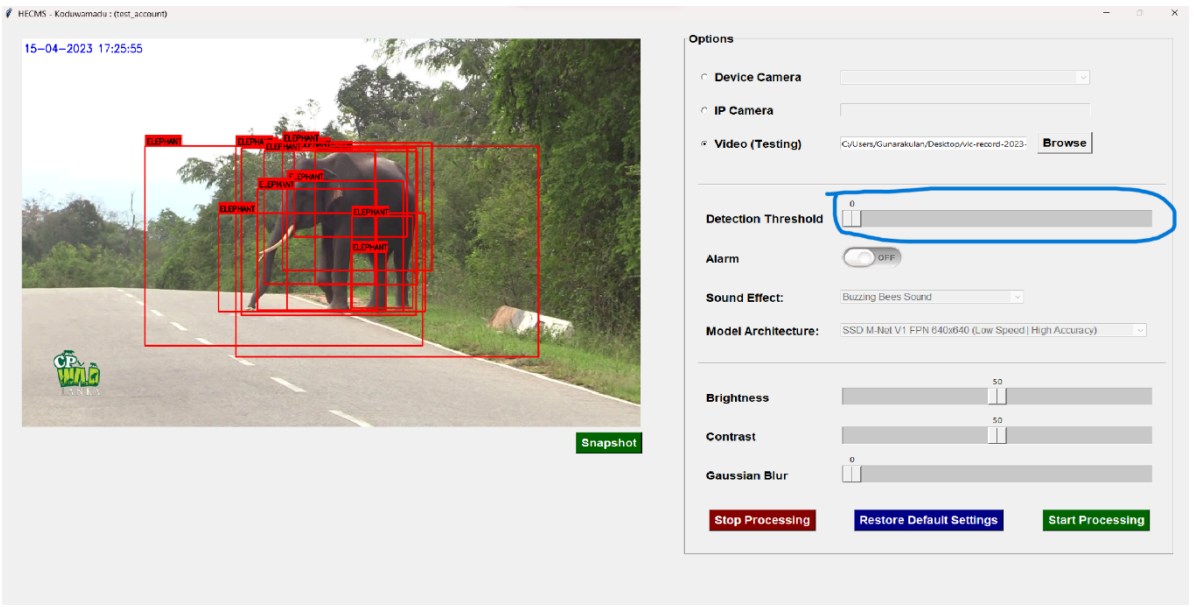
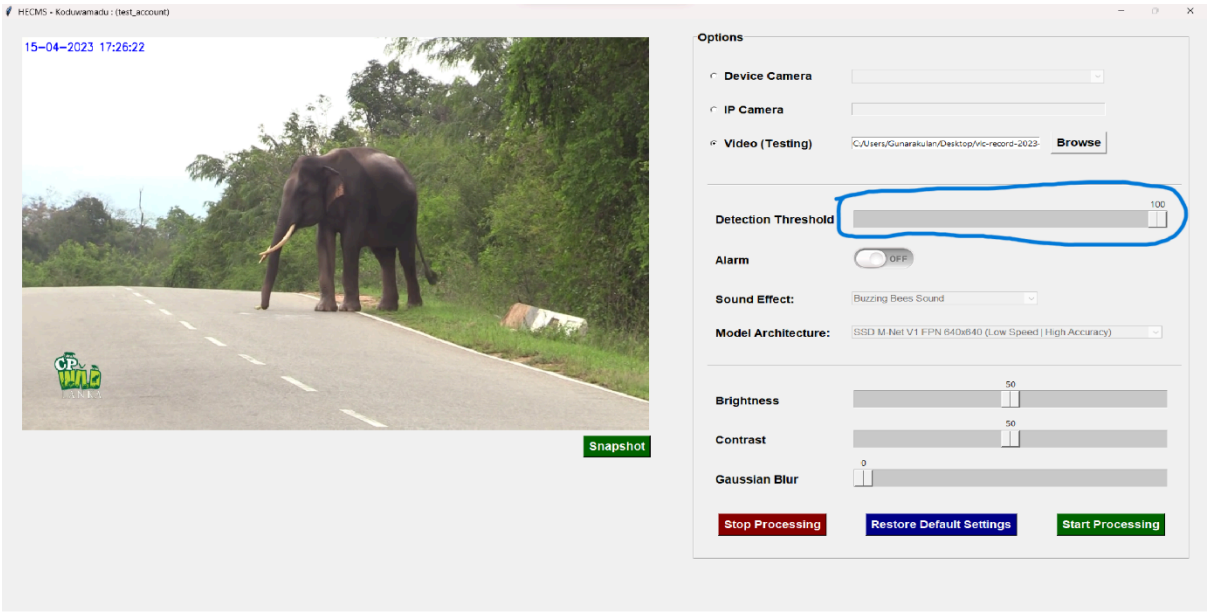
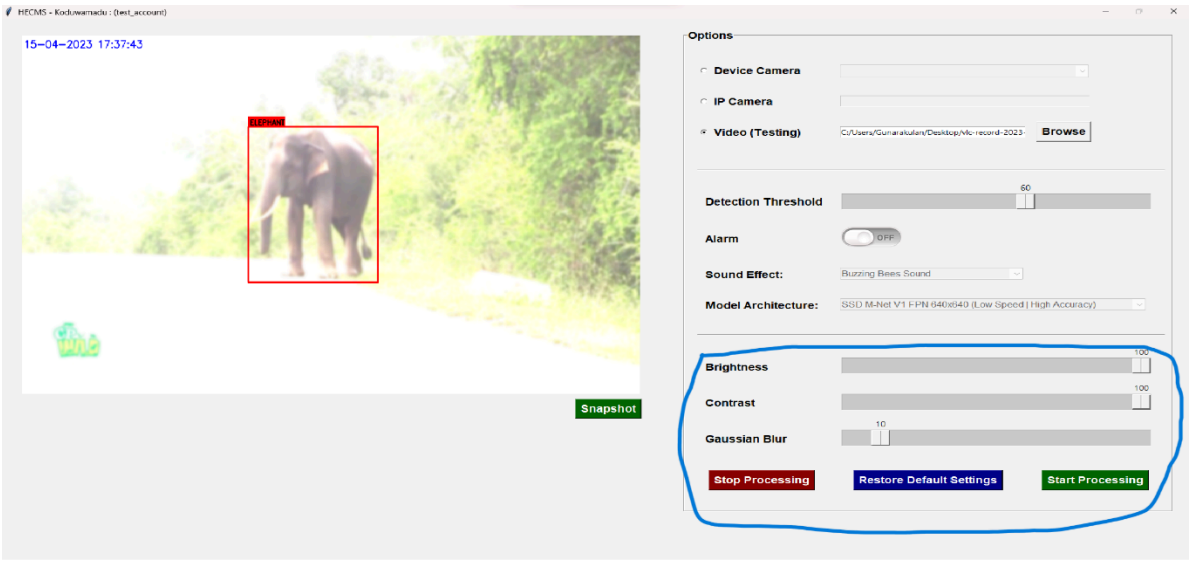
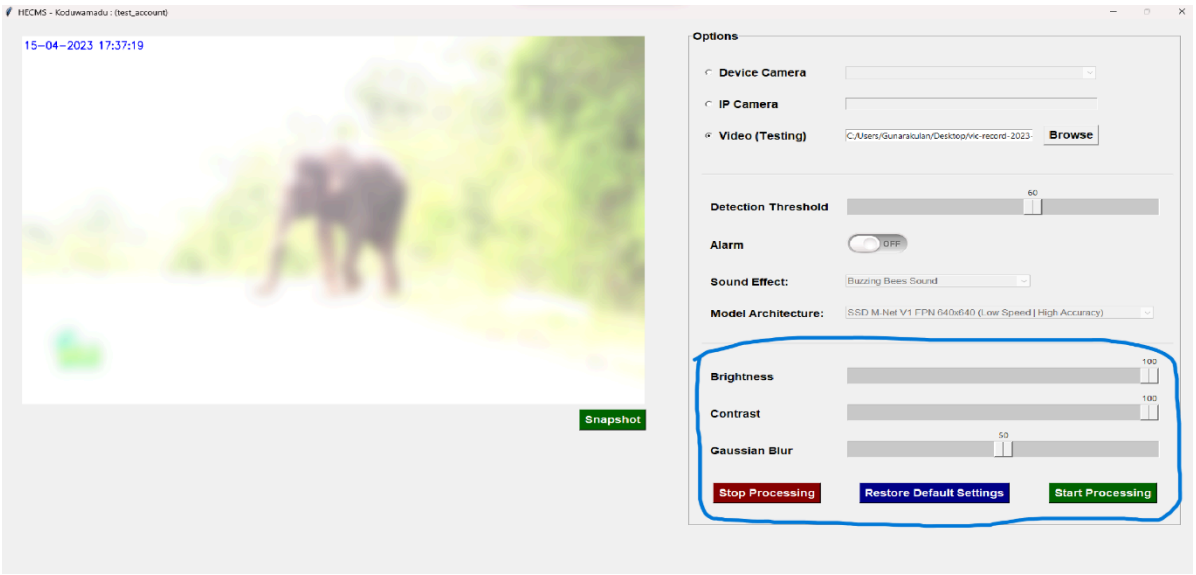
Results: Passed
   

Table 6 Test Case 5 - Monitoring System

Test Scenario: Adjust brightness / contrast / blur to test the video source effects.	
Expected Results: The video should be affected when adjusting brightness/contrast/blur.	
Entities	**Input Value**

Brightness	CHANGED = 100
Contrast	CHANGED = 100
Blur	CHANGED = 10 | CHANGED = 50
Results: Passed	





4.3.2.2 Management System Test Cases

Table 7 Test Case 1 - Management System : Super Admin

Test Scenario: Log in to the management system with the right credentials.	
Expected Results: Must login and enter into the dashboard	
Entities	**Input Value**
Username	admin
Password	admin
Account Type	Super Admin
Results: Passed	

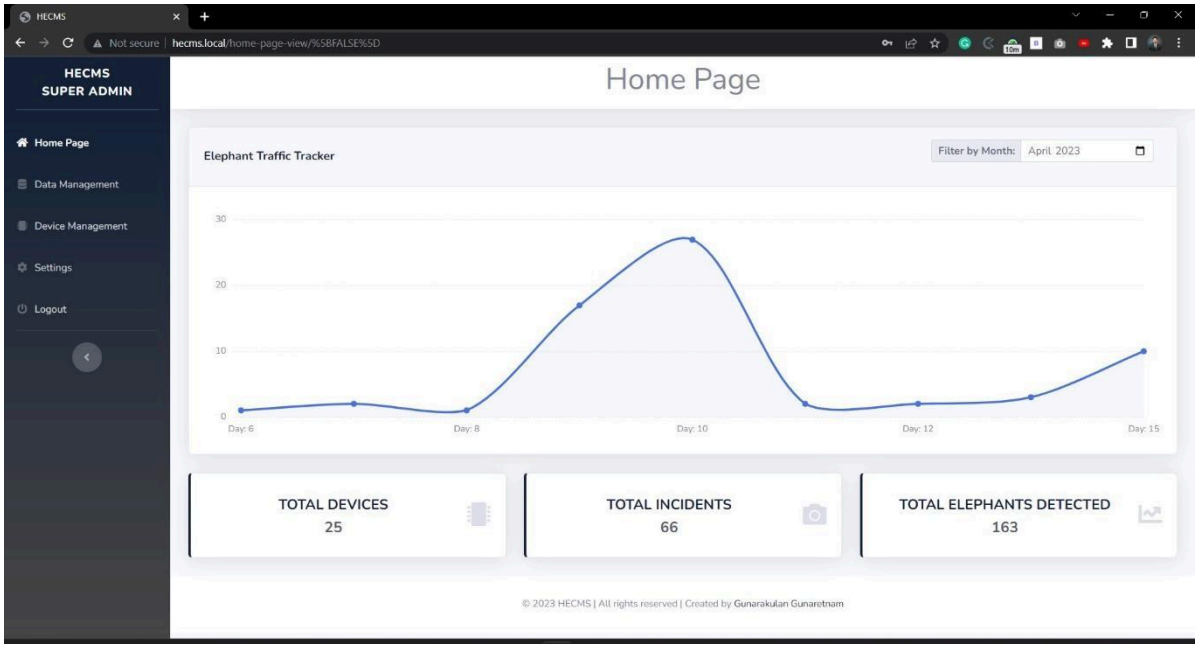
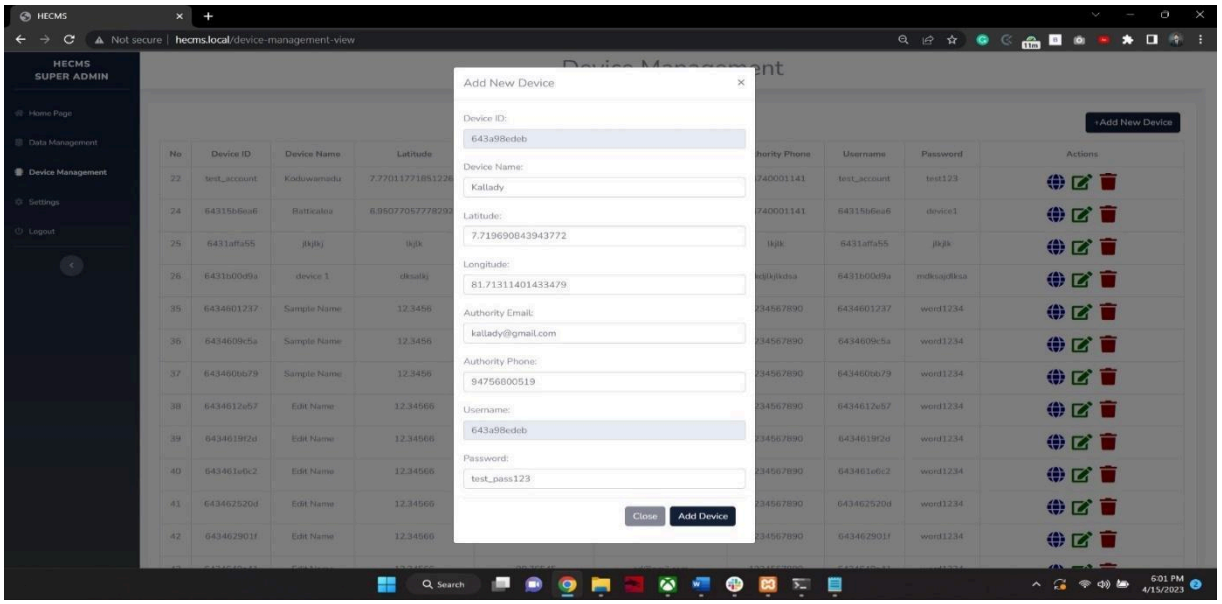


Table 8 Test Case 2 - Management System: Super Admin

Test Scenario: Create a new device to check that function execute correctly.	
Expected Results: A new device should be created and added to the table.	
Entities	**Input Value**
Device Name | Latitude | Longitude | Authority Email | Authority Phone | Password	Kallady | 7.719690843943772 | 81.71311401433479 | kallady@gmail.com | 94756800519 | test_pass123
Results: Passed	



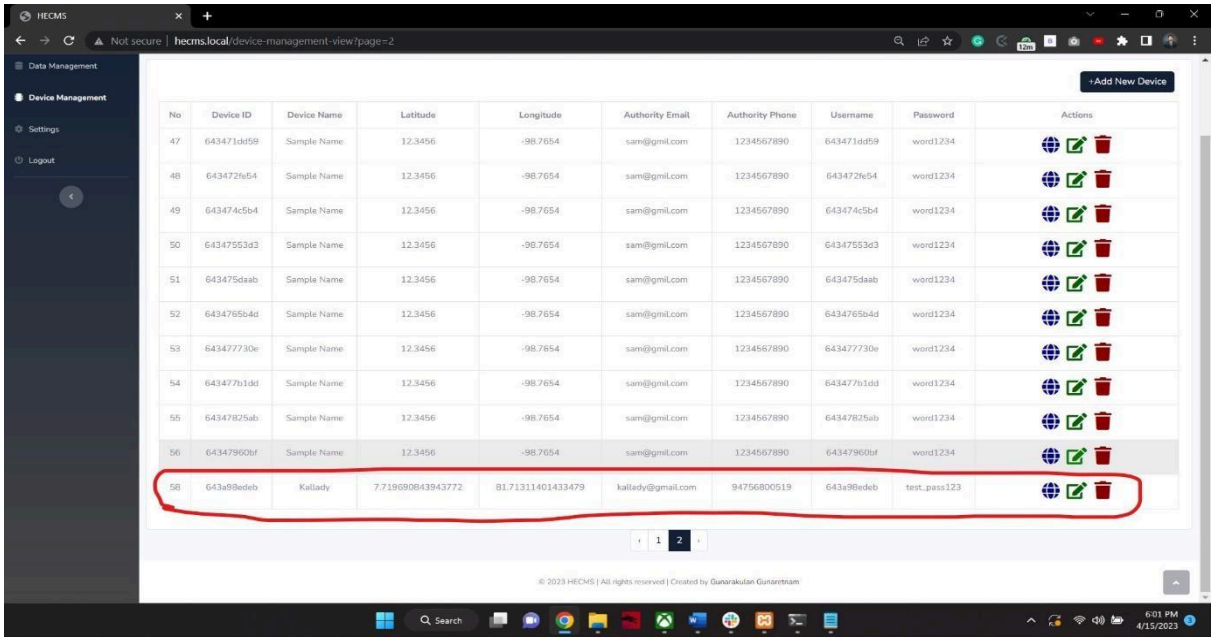
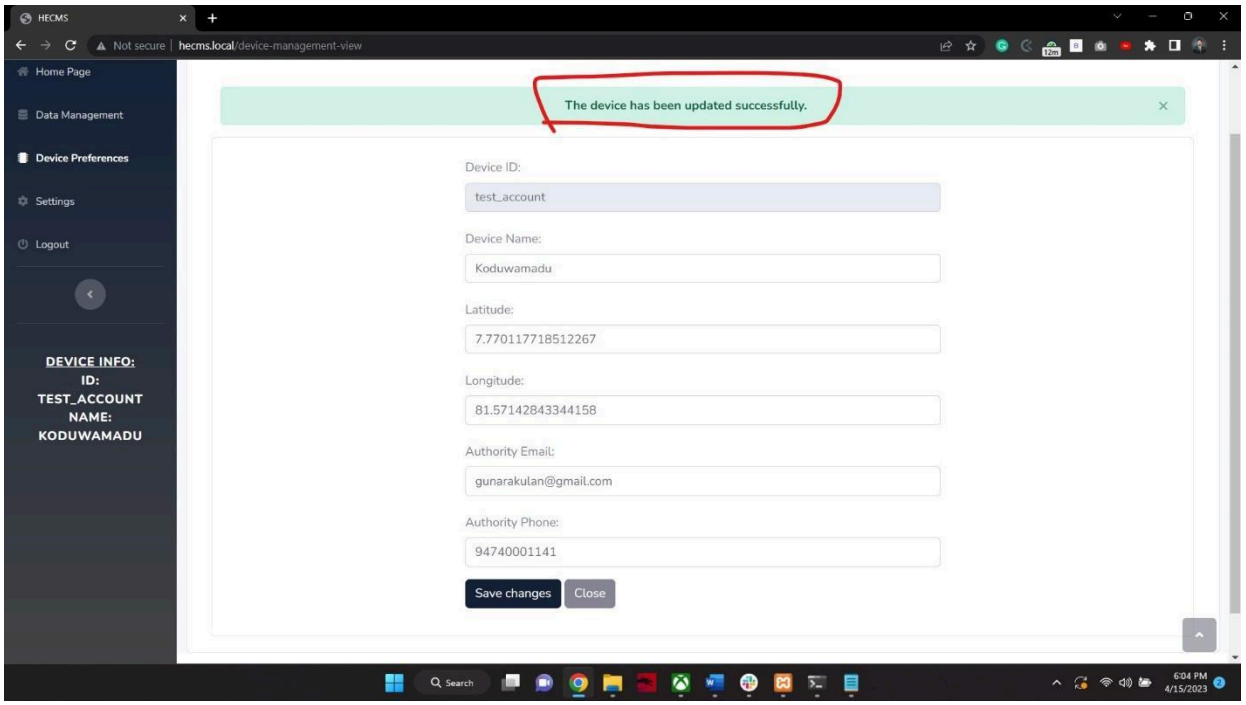


Table 9 Test Case 3 - Management System: Device Admin

Test Scenario: Update device details in Device Admin.	
Expected Results: The device details must be updated.	
Entities	**Input Value**
Authority Phone	94740001141
Results: Passed	



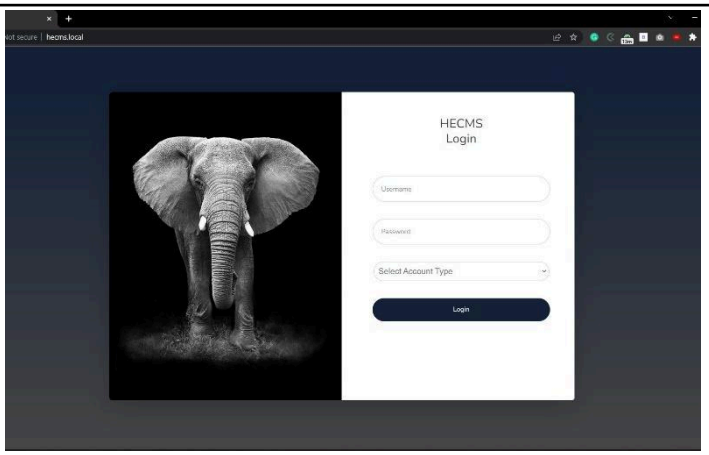
4.3.3 Security Testing

The last two previous sections showed how the manual and automatic testing was done, this section is going to discuss basic security testing; security testing is only applicable to the management system.

As already mentioned in the implementation section of the methodology chapter, the Management System was built with the Laravel framework. Typically, Laravel takes care of most of the security best practices. At the app level, most of the essential security practices are covered by default in the framework itself. Anyhow, the hosting of the server security must be considered, but at the movement, the app was hosted on localhost.

The developer needs to worry about some of the security best practices when building with Laravel; below, a security test is shown to evaluate that the researcher followed the best practice on session handling "Secure Session Management". The test scenario is to try to access the dashboard with "URL" (http://hecms.local/home-page-view/%5BFALSE%5D); if it allows opening the dashboard, it will fill; otherwise, it will be passed.

Table 10 Open the dashboard URL when the session is logout - Monitoring System

Test Scenario: Open the dashboard URL when the session is logout to test whether the session is handled correctly in programming.	
Expected Results: It must not allow to enter into the dashboard, it should throw the user to the login page again.	
Entities	**Input Value**
URL Search Bar	http://hecms.local/home-page-view/%5BFALSE%5D
Results: Passed	
	

Chapter 05: Discussion

5.1 Introduction

The discussion section of this research presents an in-depth analysis of the key findings, interpretations, and implications of the study on addressing human-elephant conflicts in Sri Lanka through the development of an Artificial Intelligence (AI)-based system.

The research highlights that despite conservation efforts and tourism opportunities associated with Sri Lanka's wild elephants, human-elephant conflicts remain a persistent and pressing issue, resulting in human casualties, property damage, and financial losses for farmers.

The existing methods, such as electric fences, have limitations in effectively preventing incidents, and there is a need for innovative and sustainable solutions. The proposed AI-based system, as discussed in the previous chapters, has the potential to mitigate human-elephant conflicts by utilizing AI and computer vision technology for elephant detection and real-time early warning alerts via SMS/Email to villages.

The system also collects and analyzes data for informed decision-making and proactive strategies. The discussion section further interprets the research findings, underscores the implications of the proposed approach for wildlife conservation and conflict resolution, and highlights the need for continued research and implementation to protect Sri Lanka's unique and endangered elephant population.

5.2 Key Findings

In the previous chapter all the research results are described and well-explained, most of the key findings are already clearly defined in the results section, this section is going about the important key finding that this research has provided.

The study may identify the strengths and weaknesses of existing solutions for the Human Elephant Conflict problem in Sri Lanka. The government uses the electric fence method that needs to be fixed, and most elephants can break that. That was a big issue; there is a need for a technological solution to this problem; this is one of the findings of the results.

This research indicates that a real-time early warning system for human-elephant conflict prevention via SMS/Email is a more productive solution, as mentioned. In the introduction chapter, in rural parts of Sri Lanka, a farmer/villager manually sits and watches for elephants at night; if they find an elephant, they alert the village people; this research found that automating this process would be much more productive and effective.

This research found that AI computer vision technology can be used to detect elephants. That technology can be appropriately applied to solve the Human Elephant Conflicts in Sri Lanka.

This research indicates that elephants are afraid of bees; therefore, implementing such a system that plays the bee sounds effect may block the elephants.

This research found that villagers and framers block elephants using firecrackers and loud sounds; therefore, playing firecracker sound effects and big loud sounds may block the elephants from entering.

This research indicates the application of artificial sounds effects to loud sounds and firecrackers and bees can be automated in a systematic way to prevent elephants entering tge villages.

This research found and suggested that systematically making this human-elephant conflict problem, such as a management system for conflict team head office and a monitoring system for sub-team, can be a good idea to prevent this issue. If this issue is handled as a

team in a structured organizational way instead of having only electric fences, if this approach is implemented, there will be a considerable change to tackle the impacts.

This research indicates that analyzing elephants' movements and storing those data will help the human-conflict team to predict and prevent future issues with elephants and humans from reducing the impact.

This research also indicates that counting elephants and storing their conflict incidents conflict management team to be more productive and help produce a more robust solution.

This research proves that according to the literature review, the WilidEye community innovation startup in Africa used a similar approach and gained so many good results; therefore, implementing such a system in Sri Lanka can also solve the problem.

This research indicates that developing a web management panel facilitates collecting and analyzing human-elephant conflict data, providing insights into elephant movements, behaviors, and intrusion patterns for evidence-based decision-making and proactive strategies.

This research found that using the TensorFlow Object Detection model to train an elephant detection system and some deep-learning algorithms can be used, such as m-net fpn.

This research indicates that Sri Lankan elephant (Elephas maximus maximus), which is a subspecies of the Asian elephant and is categorized as Endangered on the IUCN Red List, they should be protected.

It indicates that human-elephant conflict (HEC) is a significant environmental issue in Sri Lanka, with incidents of HEC increasing between 1991 and 2018, resulting in human deaths, elephant deaths, human injuries, and property damage reports.

Farmers in Sri Lanka, who often have low annual incomes, are particularly affected by human-elephant conflicts, with crop raiding by elephants causing financial losses and livelihood challenges for them.

5.3 Interpretations

This research aims to address the persistent issue of human-elephant conflicts in Sri Lanka through the development of an Artificial Intelligence (AI)-based system. The research highlights that despite conservation efforts and tourism opportunities associated with Sri Lanka's wild eleprovidingn, human-elephant conflicts remain a pressing issue. Thousands of incidents are reported annually, resulting in human casualties and property damage. The existing methods, such as electric fences, are only sometimes reliable in preventing incidents, and there is a need for innovative and sustainable solutions.

The effectiveness of the AI-based system could help in many ways to reduce the damage made by the elephants, as already mentioned in the literature review, watching for elephants is done manually by a villager or farmer. If it is automated with machines, that would be much faster, SMS will be sent quickly to village authority to alert people to take proactive actions to prevent the causes.

The system will send early warning messages to the corresponding villages, accompanied by artificial sounds of buzzing bees and firecrackers that can deter elephants from entering the villages. The system will also collect and analyze data, such as the number of elephants and incidents, which the elephant conflict management team can use for informed decision-making in the future.

The interpretation of the proposed research is that it offers a promising and technologically advanced solution to mitigate human-elephant conflicts in Sri Lanka. By utilizing AI and computer vision technology, the system has the potential to detect elephants accurately and provide timely warnings to villages, enabling proactive measures to prevent damage. Additionally, the system's data collection and analysis capabilities can aid in developing evidence-based strategies for managing human-elephant conflicts in the long term.The research underscores the need for innovative and sustainable solutions to address this pressing conservation challenge and offers a valuable contribution to wildlife conservation and conflict resolution. Further research and implementation of the proposed system can

potentially lead to significant reductions in human-elephant conflicts and contribute to protecting Sri Lanka's unique and endangered elephant population.

5.4 Implications

According to the literature review, this emphasizes the endangered status of Sri Lankan elephants and the importance of conservation efforts to protect their population and humans, there is a need to build a such a harmless system.

Then, the system can reduce the number of human-elephant conflicts by providing early warning messages to villages when elephants are detected in video frames. It can allow villagers to take proactive actions to prevent damage,

According to the literature review, Sri Lankan elephants are afraid of firecrackers, bees, and loud sounds; therefore, this system scares away elephants using the artificial sounds of buzzing bees and firecrackers. By preventing elephants from entering villages, the system can help reduce property damage, crop loss, and the risk of human casualties, thus mitigating the negative impacts of human-elephant conflicts on local communities.

The solution can be a sustainable and cost-effective solution for human-elephant conflict mitigation. Traditional methods, such as electric fences, have limitations, as elephants can sometimes break through or bypass them. In contrast, the proposed AI-based system can accurately detect elephants in video frames and provide real-time early warnings without physical barriers. This can reduce the reliance on manpower for human-elephant conflict mitigation, which can be challenging to sustain long-term due to costs and human resources.

Then the system can contribute to data collection and analysis for informed decision-making by the elephant conflict management team. By recording and storing information on the number of elephants and incidents, the system can provide valuable data that the elephant conflict management team can analyze. This data can inform future decisions, such as identifying high-risk areas, understanding elephant behavior patterns, and evaluating the effectiveness of mitigation measures. This can lead to evidence-based strategies and policies for human-elephant conflict management, promoting a more informed and systematic approach to handling the issue.

Acknowledging that the proposed AI-based system may also have limitations and potential challenges is essential. For example, the accuracy of elephant detection in video frames may depend on various factors such as lighting conditions, camera quality, and elephant behavior.

5.5 Unexpected Occurrences

During the development stage of the research, many unexpected occurrences took place that made the research progress slow. In the implementation section, the technology stack was messed up. At the time of the Monitoring System's development, it was previously planned to build with the PyQt framework for UI implementation. However, PyQt needs more support for threaded programming, which was more needed for the project, then instead of PyQt, Tkinter was selected.

Many false positives occurred during the model evolution and training, and those things were sorted out by fine-tuning and working on the project dataset. Many technical dependency issues come across, but those are also sorted effectively.

Both systems use the same database, which is one of the vast technical issues they come across; when one method is using it, the other is unable to access that; this issue was sorted by changing the entire technology stack of the management system, It was planned to build with Nodejs MERN related technologies, because of so many cases, it was changed to be developed with Laravel.

The MySQL database can store base64 values, and all detection images are transmitted as base64 to update in the correct format. This caused so many issues; the database was getting much more significant, but that issue was solved by storing the images as compressed “.bin” files.

During the development stage, there are not many more elephant images on night vision cameras. Therefore, it is challenging to sort out this issue; all-day images are artificially augmented to look like night vision images.

Chapter 06: Conclusion

6.1 Introduction

The previous chapter contains detailed discussions on the research findings, interpretations, implications etc. This chapter is the final chapter which encloses the research with the closing tag.

This chapter focuses on three key aspects: limitations, recommendations, and future work. Limitations highlight areas that require further development to enhance system performance and scalability based on the research findings. These limitations provide valuable insights for future researchers to build upon and improve existing knowledge.

It discusses limitations, recommendations, and future works and finally ends with the closing thoughts. The limitation concerns the things that must be developed further to improve the system performance and scalability of the research.

This chapter explores future research future works that might expand the scope and impact of the research. These options for additional investigation allow students in the future to explore further into the research topic and unearth fresh insights.

This chapter concludes with closing thoughts, summarizing the overall research journey. It serves as a closing tag, encapsulating the research study and providing a sense of culmination to the research process.

6.2 Limitations

This research only focuses on developing software solutions without involving hardware development; it is trying to build software applications that can mitigate the human-elephant conflicts on the software level, and that software can be installed on any tiny hardware device such as Jetson Nano or any AI computed microprocessor. However, this study did not go deep into building a hardware device for it, such as FPGA.

The research project explicitly targets Sri Lankan elephants. The findings may not directly apply to other elephant populations or wildlife conservation contexts in different regions or countries because Sri Lankan rural communities reported that elephants fear firecrackers and loud noises. In foreign countries, elephants may have different scenarios according to their nature.

The accuracy of elephant detection in the system heavily relies on the video quality captured by CCTV or wildlife camera trap devices. Poor video quality, such as low resolution or blurry images, may result in lower accuracy in detecting elephants, leading to false positives or false negatives.

The project may face regulatory, ethical, and legal considerations related to the use of AI, data privacy, and wildlife conservation practices, which may impact the implementation and outcomes of the research.

The research is mainly focusing on automating the manual procedures as the computerized solution that villagers and farmers sit and watch for elephants; if they find elephants, they alert the village people, and they will use firecrackers and loud sounds to block elephants from entering; this did not study deeply into harms behind this method. It is just automating the manual to automatic with some conflicts data storing and analyzing techniques.

Using artificial sounds, such as buzzing bees and firecracker sounds, to scare away elephants may cause noise pollution in the surrounding areas. This could disrupt the natural habitat of other wildlife and impact the local community's quality of life.

The system may require investment in hardware, software, and ongoing operational costs, including electricity, internet connectivity, and system maintenance to ensure long-term usability, including funding and resource allocation, which could be challenging.

The accuracy and effectiveness of the proposed system may be affected by weather and environmental factors, such as heavy rain, fog, dense vegetation, or challenging terrain. These types of things may impact the performance of the AI algorithms.

The proposed system may require continuous monitoring, maintenance, and updates to ensure optimal performance. This could involve regular software updates, calibration of hardware devices, and ongoing surveillance of the system's accuracy and effectiveness. Maintaining high performance and functionality over time may require dedicated efforts and resources.

6.3 Recommendation

The Human Elephant Conflict Mitigation System should be continuously researched and developed to improve its accuracy, efficiency, and effectiveness. This could involve exploring new technologies, algorithms, and approaches to detect elephants better, enhance early warning messages, and improve the management system dashboard.

Collaboration among researchers, conservation organizations, government agencies, local communities, and other stakeholders is critical to the system's effectiveness. Partnerships may help in the collection of more data, the sharing of resources, and the leveraging of knowledge to address the complex issue of human-elephant conflicts.

This system should be field tested and validated in several parts of Sri Lanka to ensure its effectiveness in various landscapes, weather situations, and human-elephant conflict scenarios. Field testing can assist in identifying any limitations or issues in the system and refining it as needed.

Capacity-building activities should be done to successfully teach community members and wildlife authorities to use and manage the system. Training on system operation, data administration, dispute resolution tactics, and creating awareness about the value of human-elephant coexistence could be included.

Long-term monitoring and evaluation of this system should be carried out to examine its influence on minimizing human-elephant conflicts and its long-term viability. Monitoring and evaluation can aid in identifying problems and areas for improvement and guide future research and development initiatives.

This system should be developed for scalability and replicability to extend to other regions or countries dealing with similar human-elephant conflict situations. This may entail creating a modular and adaptable system architecture that can be easily altered and copied in many circumstances.

6.4 Future Work

As the current research mainly focuses on software solutions, future work could explore the development of specialized hardware devices, such as Field-Programmable Gate Arrays (FPGAs) or other sensor technologies, to enhance the accuracy and effectiveness of the system, such as integrating hardware components with the software algorithms to create a more robust and efficient method for detecting and mitigating human-elephant conflicts.

Future research should dig deeper into the impact of artificial sounds on elephants, such as firecrackers, buzzing bees, and loud noises. This could include conducting acoustic impact studies on elephants to measure the potential harm caused by these sounds, such as their behavior, stress levels, and overall well-being, to identify negative effects and develop mitigation methods to consider the welfare of both humans and elephants by better understanding the impact of these artificial noises on elephants.

Since the proposed system aims to automate manual procedures that involve local communities, future work could involve community engagement and education. This could include working closely with villagers and farmers to understand their perspectives, concerns, and needs, and involving them in the system's development, implementation, and monitoring.

As the system requires ongoing operational costs and maintenance, future work could involve investigating strategies for sustainable deployment and maintenance of the system in the long term. This could include exploring funding mechanisms, resource allocation strategies, and operational models to ensure the system's continued usability, performance, and functionality over time.

Future work could involve conducting thorough impact assessments with collected data by the management system in mitigating human-elephant conflicts, such as developing new mitigation strategies with the analyzed data, and it could include monitoring and evaluating the system's performance over an extended period and assessing its impact on wildlife behavior, local communities, and the natural environment. Based on the findings, appropriate mitigation measures could be developed and implemented to address any adverse consequences of the system.

6.5 Closing Thoughts

In conclusion, the human-elephant conflict in Sri Lanka poses a significant challenge to humans and endangered elephant populations. Despite conservation efforts and tourism opportunities, human-elephant conflicts remain a pressing issue, resulting in casualties, property damage, and economic losses for local communities.

The existing methods to mitigate these conflicts, such as electric fences, are only sometimes reliable in preventing incidents, indicating the need for innovative and sustainable solutions.

The research outcome is an AI-based Human Elephant Conflict Mitigation System that utilizes camera and computer vision technology to accurately detect elephants in video frames captured by CCTV/wildlife camera traps.

The system then sends early warning messages to villages, accompanied by artificial sounds that can deter elephants from entering the villages. The system also collects and analyzes data that the elephant conflict management team can use for informed decision-making in the future.

This research project provides a technologically advanced approach to addressing the human-elephant conflict issue in Sri Lanka. Automating the early warning system and utilizing AI technology can significantly reduce human-elephant conflicts, minimize human casualties, and prevent property damage. Moreover, the system's data collection and analysis capabilities can provide valuable insights to the elephant conflict management team, enabling them to make informed decisions and implement effective future mitigation strategies.

The successful implementation of this AI-based Human Elephant Conflict Mitigation System has the potential to contribute to the conservation of endangered elephant species in Sri Lanka, protect local communities from harm and economic losses, and sustain the country's natural beauty and tourism industry.

This research project serves as a valuable contribution to the field of wildlife conservation and inspires further innovation in addressing human-elephant conflicts and other conservation challenges worldwide.

References

bees, elephantsand (no date) Elephants & bees, Elephants & Bees. Available at: https://elephantsandbees.com/ (Accessed: November 12, 2022).

Bouchard, L. (2020) "Introduction to YOLOv4 | SOTA Real-Time Object Detection in 2020," medium, 31 May. Available at: https://medium.com/what-is-artificial-intelligence/the-yolov4-algorithm-introduction-to-you-only-look-once-version-4-real-time-object-detection-5fd8a608b0fa (Accessed: November 16, 2022).

Campos-Arceiz, A. et al. (2008) "Behavior rather than diet mediates seasonal differences in seed dispersal by Asian elephants," Ecology, 89(10), pp. 2684–2691. Available at: https://doi.org/10.1890/07-1573.1.

Ceylon Expeditions (no date) Sri Lanka human – elephant conflict: Wildlife holidays sri lanka, Ceylon Expeditions. Available at: https://www.ceylonexpeditions.com/sri-lanka-human-elephant-conflict (Accessed: November 23, 2022).

Demographic details of annual elephant deaths recorded 2005-2010 (DWC data). (no date) researchgate. Available at: https://www.researchgate.net/figure/Demographic-details-of-annual-elephant-deaths-recorded-2005-2010-DWC-data_fig4_258286914.

EscVM (no date) ESCVM/OIDv4_ToolKit: Download and visualize single or multiple classes from the huge open images V4 dataset, GitHub. Available at: https://github.com/EscVM/OIDv4_ToolKit (Accessed: April 7, 2023).

Gamage, A. and Wijesundara, M. (2014) "A solution for the elephant-human conflict," 2014 Texas Instruments India Educators' Conference (TIIEC) [Preprint]. Available at: https://doi.org/10.1109/tiiec.2014.036.

Geib, C. (2020) "New artificial intelligence could save both elephant and human lives," news.mongabay.com. Sue Palminteri WildTech Reporting Fellowship, 17 September. Available at: https://news.mongabay.com/2020/09/new-artificial-intelligence-could-save-both-elephant-and-human-lives/ (Accessed: December 3, 2022).

Issues: Human-elephant conflict (no date) WWF. Available at: https://wwf.panda.org/discover/knowledge_hub/endangered_species/elephants/asian_elephants/areas/issues/elephant_human_conflict/ (Accessed: December 5, 2022).

Karunananda, A. (no date) Addressing human-elephant conflict in Sri Lanka - blog - knowledge hub, RSS. Available at: https://www.slycantrust.org/blog-posts-knowledge/addressing-human-elephant-conflict-in-sri-lanka (Accessed: November 12, 2022).

KORALAGA, D.I.L.A.N.T.H.I. (no date) "Sri Lanka: Food security impacted by flawed economic policies," www.orfonline.org. Available at: https://www.orfonline.org/expert-speak/sri-lanka-food-security-impacted-by-flawed-economic-policies/ (Accessed: November 14, 2022).

Map of elephants (no date) archives sundayobserver. Available at: https://archives.sundayobserver.lk/2010/10/17/fea25.asp.

O'Brien, C. and Palminteri, S. (2015) “Tracking imperiled elephants with cutting-edge technology: an interview with Iain Douglas-Hamilton,” https://wildtech.mongabay.com/, 2 December. Available at: https://wildtech.mongabay.com/2015/12/tracking-imperiled-elephants-with-cutting-edge-technology-an-interview-with-iain-douglas-hamilton/ (Accessed: November 16, 2022).

Peiris, U.K.L. and Sooriyabandara, M.G.C. (2014) “Effectiveness of power fences on preventing human elephant conflict in Sri Lanka,” Proceedings of International Forestry and Environment Symposium, 18. Available at: https://doi.org/10.31357/fesympo.v18i0.1865.

Premarathna, K.S., Rathnayaka, R.M. and Charles, J. (2020) “An elephant detection system to prevent human-elephant conflict and tracking of elephant using Deep Learning,” 2020 5th International Conference on Information Technology Research (ICITR) [Preprint]. Available at: https://doi.org/10.1109/icitr51448.2020.9310798.

Pyautogui (2023) PyPI. Available at: https://pypi.org/project/PyAutoGUI/ (Accessed: March 25, 2023).

Research guides: Organizing academic research papers: Types of research designs (2023) Types of Research Designs - Organizing Academic Research Papers - Research Guides at Sacred Heart University. Available at: https://library.sacredheart.edu/c.php?g=29803&p=185902#:~:text=The%20research%20design%20refers%20to,measurement%2C%20and%20analysis%20of%20data. (Accessed: April 7, 2023).

Santiapillai, C. et al. (2010) “An assessment of the human-elephant conflict in Sri Lanka,” Ceylon Journal of Science (Biological Sciences), 39(1), p. 21. Available at: https://doi.org/10.4038/cjsbs.v39i1.2350.

Sargent, C. (2021) oneearth, 25 October. Available at: https://www.oneearth.org/wildeyes-ai-helping-to-save-wild-elephants-and-prevent-human-elephant-conflict/ (Accessed: November 2, 2022).

Schneider, S., Taylor, G.W. and Kremer, S. (2018) “Deep Learning Object Detection Methods for ecological camera trap data,” 2018 15th Conference on Computer and Robot Vision (CRV) [Preprint]. Available at: https://doi.org/10.1109/crv.2018.00052.

Selenium with python¶ (no date) Selenium with Python - Selenium Python Bindings 2 documentation. Available at: https://selenium-python.readthedocs.io/(Accessed: April 9, 2023).

Shovic, J.C. and Simpson, A) “Python all-in-one.”

Take online courses. earn college credit. Research Schools, Degrees & Careers (2023) Study.com | Take Online Courses. Earn College Credit. Research Schools, Degrees & Careers. Available at: https://study.com/learn/lesson/research-methodology-examples-approaches-techniques.html#:~:text=A%20research%20approach%20is%20the,%2C%20qualitative%2C%20and%20mixed%20methods. (Accessed: April 5, 2023).

Thy, N. (2021) Using camera traps to understand and protect Asian elephants in Cambodia, Wild Earth Allies. Available at: https://wildearthallies.org/camera-traps-elephants/ (Accessed: December 06, 2022).

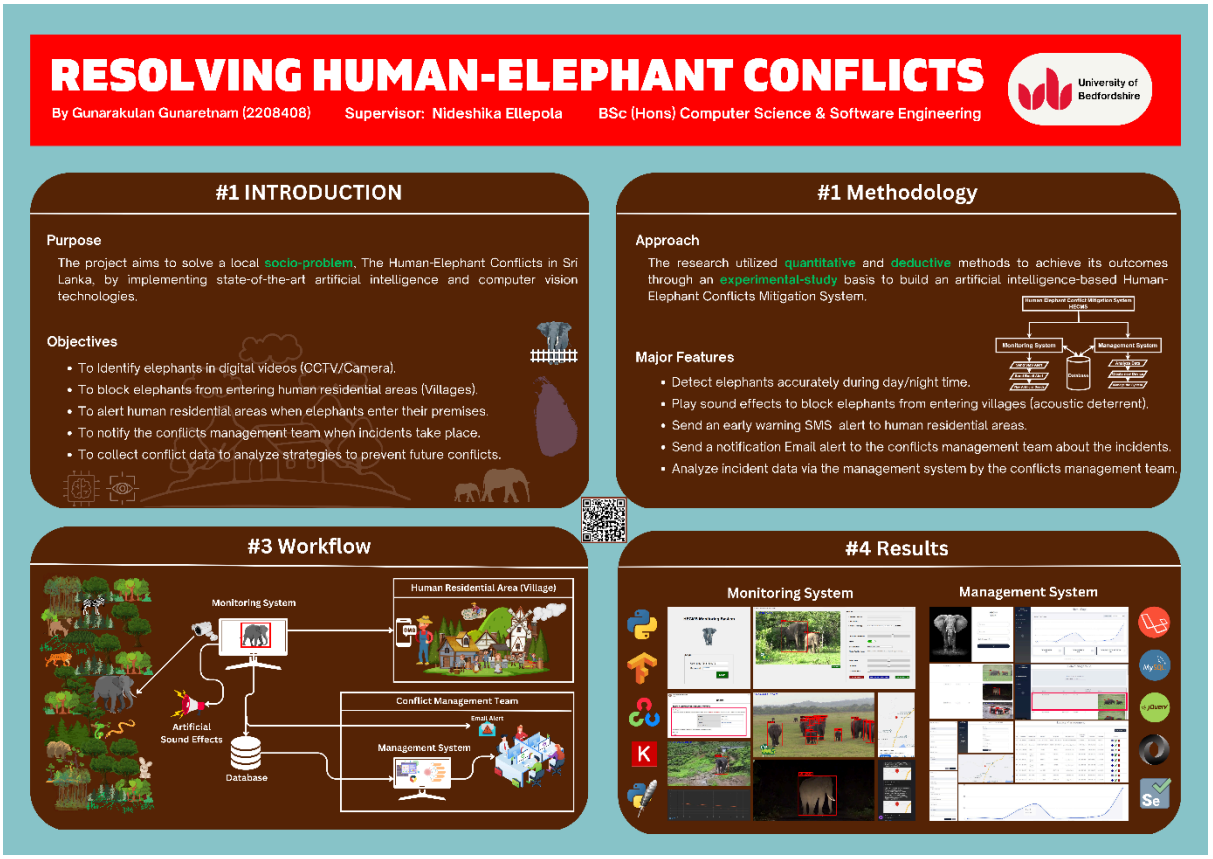
TOPSELL, E.D.W.A.R.D. (2021) History of four-footed beasts, serpents and insects: Describing at large... their true and lively figure, their several names. 1st edn. S.l.: FORGOTTEN BOOKS.

What is a wild elephant's economic worth? (2020) Daily FT. Available at: https://www.ft.lk/Columnists/What-is-a-wild-elephant-s-economic-worth/4-704195 (Accessed: December 04, 2022).

(2016) YouTube. YouTube. Available at: https://www.youtube.com/watch?v=kqkVStFLzL4 (Accessed: April 13, 2023).

(2016) YouTube. YouTube. Available at: https://www.youtube.com/watch?v=FlBFdEgrTBM (Accessed: April 10, 2023).

Appendix A: Research Poster



Appendix B: Project Estimation

Project	Human Elephant Conflicts Migration System	
Client	SLIIT	
Team	Gunarakulan Gunaretnam	
Reference Document	Document	
Feature No	**Requirement**	**Hours [GR]**
	General Works	
	Get Started + Project Management	
	Kickstart (meetings, cleanup, files creations, initializations)	20
	Project Management	50
	DevOPs + Migration + Integration + Version Control	15
	QA Testing	15
	Total hours	**100**
	Designs: UX/UI & Wireframe	
	Wireframes Designs for Monitoring System	5
	Wireframes Designs for Super Admin System	5
	Wireframes Designs for Backend Admin System	5
	UI/UX Designs for Monitoring System	5
	UI/UX Designs for Super Admin System	5
	UI/UX Designs for Backend Admin System	5
	Buffer	5
	Total hours	**35**
	Monitoring System	
	Data Collocation & Training: Preparations | Annotations | Labeling | Training	
	Dataset research	10
	Daytime Data collocation, annotations, labeling	8
	Nightime Data collocation, annotations, labeling	8
	Train daytime model	20
	Train nighttime model	20
	Buffer	10
	Total hours	**76**

Feature Development (More Info)	
Convert the UI/UX	10
Write a script to inference the daytime model: Able to detect elephants in the daytime	5
Write a script to inference the nighttime model: Able to detect elephants in the nighttime	5
Able to play artificial sound effects	6
Able to send early warning SMS to authorities	6
Able to send early warning emails to authorities	6
Able to update elephant detection records in the database (Number of Elephants | Time and Date | Camera ID | Camera Place)	6
Able to take snapshots when elephants are detected and update the database	5
Able to record videos when elephants are detected and update the database	15
Able to update real-time video streaming to the backend panel	25
Able to detect all the cameras that are attached to the deployed device	6
Able to process videos from attached cameras	4
Able to process live videos from an IP camera	4
Able to process pre-recorded videos	4
Able to adjust the threshold value to find a better accuracy	4
Able to switch between daytime and nighttime models depending on the current time of the deployed device (Automatic)	4
Able to switch between daytime and nighttime model (Manual)	4
Able to take screenshots in the monitoring system	2
Buffer	50
Total hours	**171**

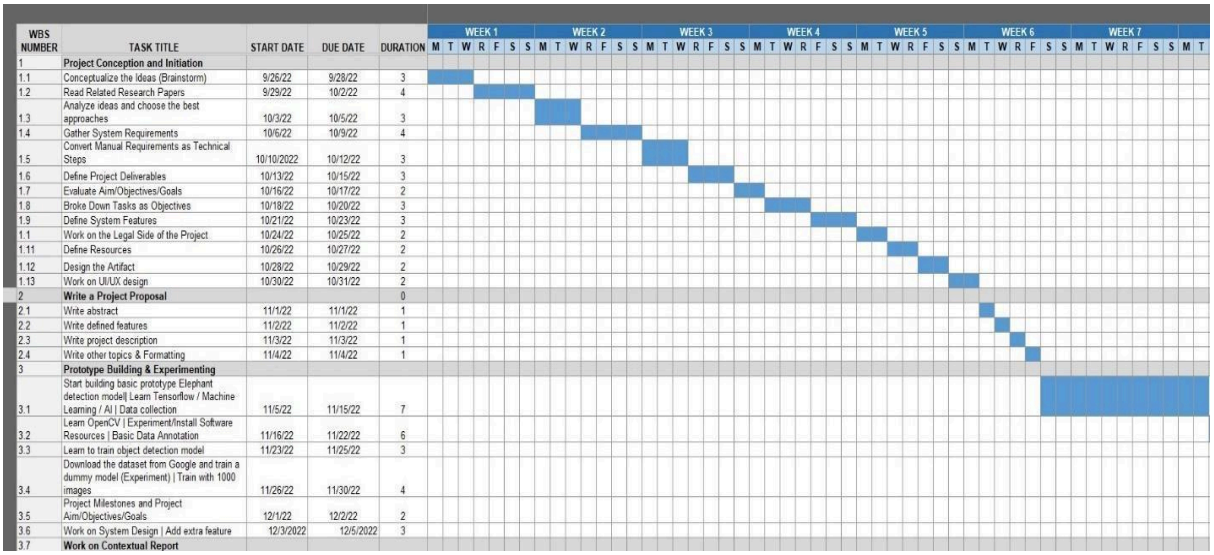
Management Panel

Super Admin

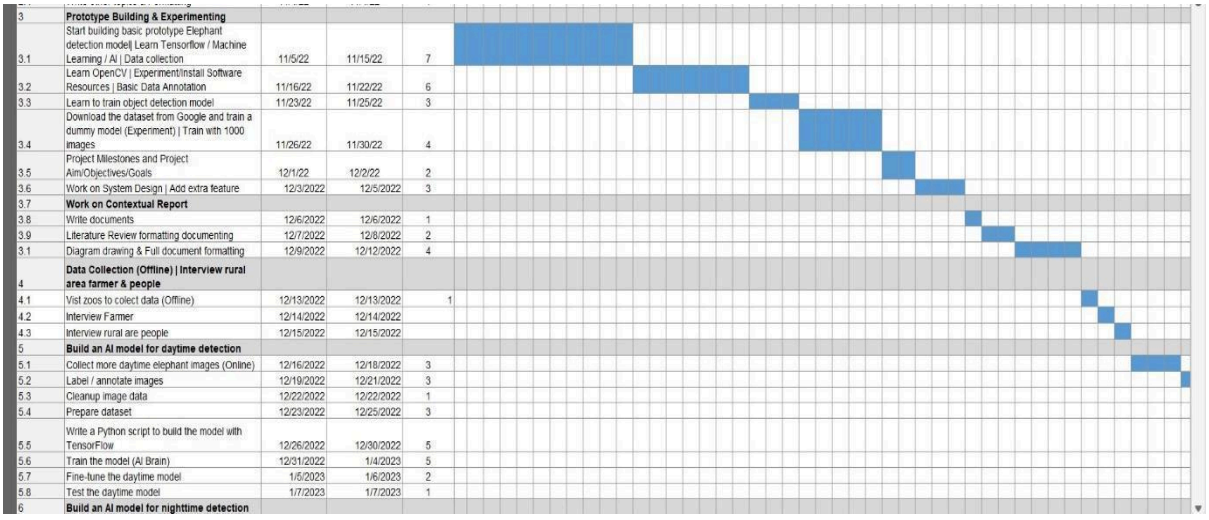
Feature Development	
UI/UX: Convert Login Page | Dashboard | Analytics| Device Management | Settings Page | Logout	15
Dashboard Page: Total devices | Total detections | Last Detection | Daywise chart	10
Analytics Page: All detection data should be displayed on the analytics page [More Info]	3
Analytics Page: The same daywise analytics chart should be displayed on the top of the analytics page	10
Device Management Page: Create a new monitoring device [More Info]	10
Device Management Page: Show all created monitoring devices [More Info]	4

Device Management Page: Search for a device [More Info]	3
Device Management Page: Delete a device [More Info]	3
Device Management Page (Sub Device): Able to click & get into a device & see info (Auto Login without credentails) [More Info]	5
Device Management Page (Sub Device): Each device page info should be open in a new tab [More Info]	3
Device Management Page (Sub Device | Dashboard): Device live streaming [More Info]	10
Device Management Page (Sub Device | Setting Module): Edit device information [More Info]	4
Settings Page: Change password of the super admin account	2
Logout Page: A button should there to logout the super admin	2
Buffer	50
Total hours	**134**
Backend Admin	
Feature Development	
UI/UX: Convert Login Page | Dashboard | Analytics | Device Preferences | Settings Page | Logout	10
Dashboard Page: Device Basic Info | Total detections | Last Detection | Daywise chart	5
Analytics Page: All detection data should be displayed on the analytics page [More Info]	4
Analytics Page: The same daywise analytics chart should be displayed on the top of the analytics page	4
Dashboard Page: Device live streaming [More Info]	10
Device Preferences: Edit device info	5
Setting Page: Edit credentails [More Info]	10
Logout Page: A button should there to logout the super admin	3
Buffer	50
Total hours	**101**
Overall Project Buffer	**100**
TOTAL HOURS	717
TOTAL BUFFER	265
WITHOUT BUFFER	452
PROJECT DURATION	6-8 Months

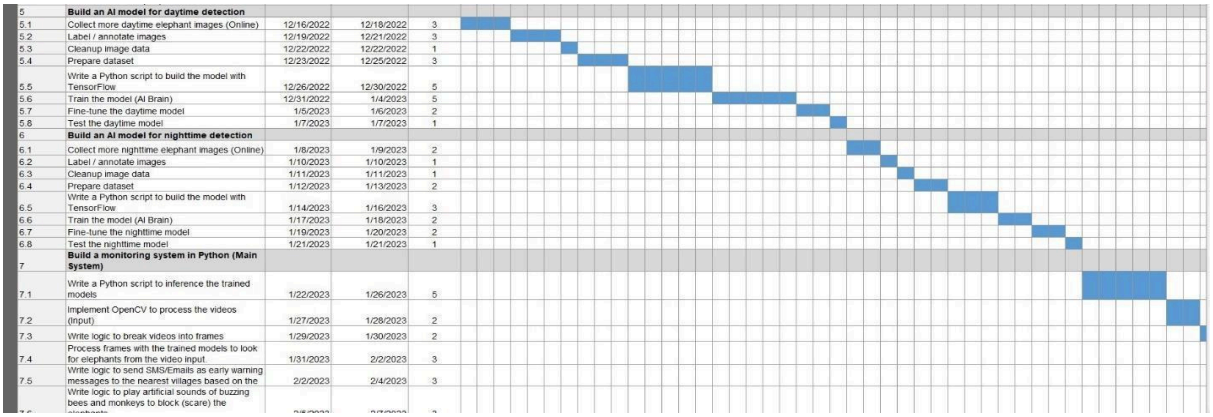
Appendix C: Project Gantt Chart



WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION
1	**Project Conception and Initiation**			
1.1	Conceptualize the Ideas (Brainstorm)	9/26/22	9/28/22	3
1.2	Read Related Research Papers	9/29/22	10/2/22	4
1.3	Analyze ideas and choose the best approaches	10/3/22	10/5/22	3
1.4	Gather System Requirements	10/6/22	10/9/22	4
1.5	Convert Manual Requirements as Technical Steps	10/10/2022	10/12/22	3
1.6	Define Project Deliverables	10/13/22	10/15/22	3
1.7	Evaluate Aim/Objectives/Goals	10/16/22	10/17/22	2
1.8	Broke Down Tasks as Objectives	10/18/22	10/20/22	3
1.9	Define System Features	10/21/22	10/23/22	3
1.1	Work on the Legal Side of the Project	10/24/22	10/25/22	2
1.11	Define Resources	10/26/22	10/27/22	2
1.12	Design the Artifact	10/28/22	10/29/22	2
1.13	Work on UI/UX design	10/30/22	10/31/22	2
2	**Write a Project Proposal**			0
2.1	Write abstract	11/1/22	11/1/22	1
2.2	Write defined features	11/2/22	11/2/22	1
2.3	Write project description	11/3/22	11/3/22	1
2.4	Write other topics & Formatting	11/4/22	11/4/22	1
3	**Prototype Building & Experimenting**			
3.1	Start building basic prototype Elephant detection model| Learn Tensorflow / Machine Learning / AI | Data collection	11/5/22	11/15/22	7
3.2	Learn OpenCV | Experiment/Install Software Resources | Basic Data Annotation	11/16/22	11/22/22	6
3.3	Learn to train object detection model	11/23/22	11/25/22	3
3.4	Download the dataset from Google and train a dummy model (Experiment) | Train with 1000 images	11/26/22	11/30/22	4
3.5	Project Milestones and Project Aim/Objectives/Goals	12/1/22	12/2/22	2
3.6	Work on System Design | Add extra feature	12/3/2022	12/5/2022	3
3.7	**Work on Contextual Report**			



3	**Prototype Building & Experimenting**			
3.1	Start building basic prototype Elephant detection model| Learn Tensorflow / Machine Learning / AI | Data collection	11/5/22	11/15/22	7
3.2	Learn OpenCV | Experiment/Install Software Resources | Basic Data Annotation	11/16/22	11/22/22	6
3.3	Learn to train object detection model	11/23/22	11/25/22	3
3.4	Download the dataset from Google and train a dummy model (Experiment) | Train with 1000 images	11/26/22	11/30/22	4
3.5	Project Milestones and Project Aim/Objectives/Goals	12/1/22	12/2/22	2
3.6	Work on System Design | Add extra feature	12/3/2022	12/5/2022	3
3.7	**Work on Contextual Report**			
3.8	Write documents	12/6/2022	12/6/2022	1
3.9	Literature Review formatting documenting	12/7/2022	12/8/2022	2
3.1	Diagram drawing & Full document formatting	12/9/2022	12/12/2022	4
4	**Data Collection (Offline) | Interview rural area farmer & people**			
4.1	Vist zoos to colect data (Offline)	12/13/2022	12/13/2022	1
4.2	Interview Farmer	12/14/2022	12/14/2022	
4.3	Interview rural are people	12/15/2022	12/15/2022	
5	**Build an AI model for daytime detection**			
5.1	Collect more daytime elephant images (Online)	12/16/2022	12/18/2022	3
5.2	Label / annotate images	12/19/2022	12/21/2022	3
5.3	Cleanup image data	12/22/2022	12/22/2022	1
5.4	Prepare dataset	12/23/2022	12/25/2022	3
5.5	Write a Python script to build the model with TensorFlow	12/26/2022	12/30/2022	5
5.6	Train the model (AI Brain)	12/31/2022	1/4/2023	5
5.7	Fine-tune the daytime model	1/5/2023	1/6/2023	2
5.8	Test the daytime model	1/7/2023	1/7/2023	1
6	**Build an AI model for nighttime detection**			



5	**Build an AI model for daytime detection**			
5.1	Collect more daytime elephant images (Online)	12/16/2022	12/18/2022	3
5.2	Label / annotate images	12/19/2022	12/21/2022	3
5.3	Cleanup image data	12/22/2022	12/22/2022	1
5.4	Prepare dataset	12/23/2022	12/25/2022	3
5.5	Write a Python script to build the model with TensorFlow	12/26/2022	12/30/2022	5
5.6	Train the model (AI Brain)	12/31/2022	1/4/2023	5
5.7	Fine-tune the daytime model	1/5/2023	1/6/2023	2
5.8	Test the daytime model	1/7/2023	1/7/2023	1
6	**Build an AI model for nighttime detection**			
6.1	Collect more nighttime elephant images (Online)	1/8/2023	1/9/2023	2
6.2	Label / annotate images	1/10/2023	1/10/2023	1
6.3	Cleanup image data	1/11/2023	1/11/2023	1
6.4	Prepare dataset	1/12/2023	1/13/2023	2
6.5	Write a Python script to build the model with TensorFlow	1/14/2023	1/16/2023	3
6.6	Train the model (AI Brain)	1/17/2023	1/18/2023	2
6.7	Fine-tune the nighttime model	1/19/2023	1/20/2023	2
6.8	Test the nighttime model	1/21/2023	1/21/2023	1
7	**Build a monitoring system in Python (Main System)**			
7.1	Write a Python script to inference the trained models	1/22/2023	1/26/2023	5
7.2	Implement OpenCV to process the videos (input)	1/27/2023	1/28/2023	2
7.3	Write logic to break videos into frames	1/29/2023	1/30/2023	2
7.4	Process frames with the trained models to look for elephants from the video input.	1/31/2023	2/2/2023	3
7.5	Write logic to send SMS/Emails as early warning messages to the nearest villages based on the	2/2/2023	2/4/2023	3
	Write logic to play artificial sounds of buzzing bees and monkeys to block (scare) the			

7	**Build a monitoring system in Python (Main System)**				
7.1	Write a Python script to inference the trained models	1/22/2023	1/26/2023	5	
7.2	Implement OpenCV to process the videos (input)	1/27/2023	1/28/2023	2	
7.3	Write logic to break videos into frames	1/29/2023	1/30/2023	2	
7.4	Process frames with the trained models to look for elephants from the video input	1/31/2023	2/2/2023	3	
7.5	Write logic to send SMS/Emails as early warning messages to the nearest villages based on the	2/2/2023	2/4/2023	3	
7.6	Write logic to play artificial sounds of buzzing bees and monkeys to block (scare) the elephants	2/5/2023	2/7/2023	3	
7.7	Write a logic to update the database if elephants are found with time/date	2/8/2023	2/11/2023	3	
7.8	Write logic to automatically swap AI models depending on the day/nighttime; during the nighttime, the nighttime AI model (AI Brain) will be working, like that during the daytime, the daytime AI model (AI brain) will be working.	2/11/2023	2/15/2023	4	
7.8	Write logic to run all processes in a looping manner	2/16/2023	2/17/2023	2	
7.9	Work on reflective report & Clean up | Document	2/18/2023	2/20/2023	3	

8.1	**Build a web panel (Management System)**			
8.1	Design the web panel (UI/UX)	2/21/2023	2/22/2023	2
8.1	Create the web panel with MERN Stack	2/23/2023	2/26/2023	4
8.1	Create database	2/27/2023	2/27/2023	1
8.1	Create dashboard	2/28/2023	3/1/2023	2
8.1	Create data storing budgets	3/2/2023	3/3/2023	2
8.1	Connect the monitoring system's (Main System) database with this web panel to view information on elephant discovery.	3/4/2023	3/11/2023	2
9.1	**Testing the functionalities**			
9.2	Test daytime AI model accuracy.	3/12/2023	3/13/2023	2
9.3	Test nighttime AI model accuracy.	3/14/2023	3/14/2023	0.5
9.4	Test the main system functionalities	3/15/2023	3/15/2023	0.5
9.5	Test the web panel functionalities	3/16/2023	3/16/2023	0.5
9.6	**Cleanup | Format Final Report Writing**	3/17/2023	4/23/2023	18