PAPER NAME	AUTHOR
42_SudeeshES.docx	Sudeesh E S

WORD COUNT	CHARACTER COUNT
9952 Words	58740 Characters
PAGE COUNT	**FILE SIZE**
80 Pages	3.3MB
SUBMISSION DATE	**REPORT DATE**
May 14, 2024 12:46 PM GMT+5:30	May 14, 2024 12:47 PM GMT+5:30

● 22% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 16% Internet database
- Crossref database
- 21% Submitted Works database
- 6% Publications database
- Crossref Posted Content database

ABSTRACT

The Online Crime Reporting System using Python Django is a web-based platform designed to empower individuals to report criminal activities and incidents in a convenient and efficient manner. This project aims to bridge the gap between the general public and law enforcement in Kottayam district, ensuring a seamless process for reporting and addressing crimes. The system provides an easy-to-use interface where users can submit detailed information about criminal incidents, suspicious activities, or safety concerns.

Users can safely register and submit complete incident details, including location, time, description, and multimedia proof like photos, thanks to its user-friendly interface. Using a centralized database guarantees data accessibility and accuracy, which increases the effectiveness of law enforcement initiatives. Administrators can effectively analyze, prioritize, and manage incidents with the help of a dedicated panel. It is expected that the deployment of "Echo" in the Kottayam region will promote enhanced public-law enforcement contact, leading to quicker and more efficient responses to criminal incidents. "Echo" is essentially a noteworthy technology innovation that aims to improve public safety and welfare in communities by streamlining crime reporting procedures.

List of Abbreviation

HTML	–	Hyper Text Markup Language
CSS	–	Cascading Style Sheet
SQLite	–	Structured Query Language Lite
UML	–	Unified Modelling Language
JS	–	JavaScript
AJAX	–	Asynchronous JavaScript and XML Environment
RDBMS	_	Relational Database Management System
IDE	_	Integrated Development Environment
BDD	_	Behavioral-Driven Development
UAT	_	User Acceptance Test
URL	_	Uniform Resource Locator
IP	_	Internet Protocol

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

"Echo" is a cutting-edge online crime reporting system designed with Python Django with the goal of making it easier to report crimes and incidents in the Kottayam region. It facilitates the reporting and resolution of crimes by acting as a link between the public and law enforcement. Its easy-to-use features allow people to safely register and submit detailed incident information, such as location, time, description, and multimedia proof like images. By using a centralized database to guarantee data accuracy and accessibility, the technology improves the effectiveness of law enforcement initiatives. Administrators can efficiently analyze, prioritize, and manage reported incidents with the use of a dedicated panel.

The project focuses security and scalability by utilizing Django's strong framework. Its authentication system ensures user privacy and data protection. "Echo" is a cutting-edge online crime reporting system designed with Python Django with the goal of making it easier to report crimes and incidents in the Kottayam region. By enhancing contact between the public and law enforcement, its deployment in the Kottayam area is expected to result in more prompt and efficient responses to criminal situations. All things considered, "Echo" is a big step toward using technology to improve community well-being and public safety.

1.2 PROJECT SPECIFICATION

The "ECHO" Crime Reporting System is meticulously designed, employing a state-of-the-art technology stack. The front-end leverages HTML and CSS to deliver an intuitive user interface, while the back-end relies on Python/Django, ensuring a robust and adaptable foundation.

In the user module, the system prioritizes security with a secure login/logout system and offers profile management features, allowing users to create, modify, and manage their profiles. Anonymity is a key consideration, providing an option for users to submit incidents anonymously.

The admin module equips administrators with essential tools for user account management, employer verification, role definition, and an overview dashboard displaying crucial metrics for effective administration.

The incident reporting module offers a comprehensive reporting form, enabling users to submit detailed incident reports. Multimedia uploads are supported, allowing users to provide additional evidence such as photos and videos. The system also allows users to categorize incidents, facilitating efficient resource allocation.

The system's architecture seamlessly integrates HTML and CSS for an intuitive user interface, while Python/Django ensures efficient data processing. The inclusion of robust libraries enhances the overall capabilities of the system.

Security is a paramount consideration, with the implementation of secure user authentication and authorization to protect user data and ensure privacy.

User experience is central to the project, emphasizing a user-friendly interface for both the public and law enforcement. Clear navigation and a positive overall experience are key design principles.

Anticipating future developments, "ECHO" envisions updates that may include advanced features like AI-driven incident categorization or enhanced analytics for better crime prevention.

Rigorous testing procedures cover functionality, security, and user experience, including user acceptance testing. "Echo" aims to be a reliable solution, simplifying and enhancing the crime reporting experience for all users involved.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

A system study is a critical phase in the lifecycle of any software or organizational project. It is a systematic and structured process of analysing, understanding, and evaluating the existing systems, processes, and technologies within an organization. The primary objective of a system study is to identify inefficiencies, shortcomings, and areas for improvement. This process serves as the foundation for making informed decisions about implementing new systems, software, or process enhancements.

System studies involve a detailed examination of the organization's current operations, which can range from business processes to information systems and technologies in use. The goal is to gain a comprehensive understanding of how the organization operates and how it can be optimized.

2.2 EXISTING SYSTEM

The current crime reporting landscape faces challenges characterized by fragmented reporting processes, data management inefficiencies, and delayed responses from law enforcement agencies. A closer examination exposes gaps in incident documentation, hindering the timely and effective collaboration between the public and law enforcement. Existing systems often lack a unified and systematic approach, resulting in prolonged response times and missed opportunities to address incidents promptly. Despite attempts to modernize, these systems struggle to adapt to the evolving dynamics of crime reporting, contributing to a disconnect between the community and law enforcement. The demand for a more responsive and streamlined system in the crime reporting ecosystem is evident.

2.2.1 NATURAL SYSTEM STUDIED

The existing natural system in crime reporting relies heavily on traditional, manual processes, leading to several challenges in effective incident management. Law enforcement agencies often grapple with paper-based documentation, hindering the seamless flow of information and creating data management issues. The reliance on manual reporting by the public results in delays in incident reporting and response times. Additionally, the lack of a unified and automated system contributes to difficulties in tracking and prioritizing reported incidents. The "ECHO" system aims to address these shortcomings by introducing automation and modernizing the crime reporting process. Through streamlined and automated workflows, it seeks to enhance data management, reduce reporting delays, and improve the overall efficiency of incident handling by law enforcement agencies. The goal is to create a more responsive and integrated system.

.

2.2.2 DESIGNED SYSTEM STUDIED

In the realm of online crime reporting, various designed systems have been examined, revealing both limitations and advantages. Many existing systems may lack comprehensive features, user-friendly interfaces, and efficient data management practices. The examination of these systems aims to distill effective elements that can be integrated into the proposed "ECHO" system for a more robust solution. Existing designed systems in crime reporting often focus on specific aspects and may overlook the holistic needs of both the public reporting incidents and law enforcement managing these reports. One common issue is the complexity of interfaces, which can pose challenges for users in navigating seamlessly through the reporting process. "ECHO" strives to overcome these challenges by simplifying the user experience, ensuring that technological advancements are harnessed to create an accessible and efficient platform for both the public and law enforcement agencies. The goal is to provide a user-friendly interface that enhances the overall experience of reporting and managing incidents.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Data Management Challenges: The current crime reporting system grapples with data management issues. The extensive information generated, including incident details, evidence, and reports, is often managed through manual and paper-based methods. This reliance can result in errors, inconsistencies, data duplication, and information loss, compromising the overall integrity of the crime data.
- Missed Incident Reports: The existing crime reporting system may not effectively reach individuals who witness or experience criminal activities but choose not to actively report them. Additionally, the system might struggle to keep track of reported incidents that require further follow-up or investigation, leading to missed opportunities in addressing potential criminal activities.
- Manual Screening and Review: Human intervention is often necessary in the screening and review process of reported incidents. This manual approach, undertaken by law enforcement or administrative staff, can be time-consuming, tedious, and subjective. This manual screening may result in overlooked or delayed assessments of the severity and urgency of reported incidents.
- Inefficient Communication Channels: Communication between the public reporting incidents and law enforcement within the existing system may lack a centralized and streamlined approach. This can result in delays, misunderstandings, and a lack of transparency, making it challenging for both the public and law enforcement agencies to stay informed about the

progress and resolution of reported incidents. Improved communication channels are essential to enhance the effectiveness of the crime reporting and resolution process

2.4 PROPOSED SYSTEM

The Online Crime Reporting System using Python Django is a web-based platform designed to empower individuals to report criminal activities and incidents in a convenient and efficient manner. This project aims to bridge the gap between the general public and law enforcement in Kottayam district, ensuring a seamless process for reporting and addressing crimes. The system provides an easy-to-use interface where users can submit detailed information about criminal incidents, suspicious activities, or safety concerns. The system offers several key features, including user registration and management, and administrative interfaces for Admin. Upon registration, users can log in securely and submit comprehensive details related to the incident, such as location, time, descriptions, and even multimedia evidence like photos. These reports are then stored in a centralized database, ensuring data integrity and easy access. The admin is equipped with a dedicated administrative panel where they can review, prioritize, and manage reported incidents.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- **Streamlined Reporting Process:** The "Echo" system simplifies the crime reporting process by providing an intuitive and user-friendly interface. Citizens can easily document incidents, upload relevant evidence like photos or videos, and submit reports with minimal effort. This streamlined process reduces the barriers to reporting, encouraging more people to report criminal activities promptly.

- **Faster Response Time:** By automating and optimizing the incident documentation and reporting process, "Echo" significantly reduces response times for law enforcement agencies. With real-time incident alerts and streamlined data collection, law enforcement can quickly assess and prioritize incidents, dispatching resources more efficiently and effectively to address urgent situations.

- **Enhanced Collaboration Between Public and Law Enforcement:** "Echo" fosters better collaboration and communication between the public and law enforcement agencies. The system allows for two-way communication, enabling law enforcement to request additional information or updates from the reporter, and citizens to receive updates on the status of their reports. This collaborative approach builds trust and strengthens community-police relationships.

- **Data-Driven Insights for Decision Making:** With a robust back-end infrastructure, "Echo" collects and analyzes data from reported incidents, providing law enforcement agencies with valuable insights and trends. This data-driven approach enables agencies to make informed decisions, allocate resources more effectively, and implement targeted crime prevention strategies based on real-time data and analytics.

- **Safer and More Connected Community:** Ultimately, "Echo" contributes to creating a safer and more connected community by empowering citizens to actively participate in crime prevention and law enforcement efforts. By making the reporting process more accessible and efficient, and by fostering collaboration between the public and law enforcement, "Echo" helps build a community where residents feel safer, more informed, and more engaged in creating a secure environment for everyone. By leveraging modern technology and innovative approaches, "Echo" aims to establish a new standard for crime reporting and law enforcement collaboration, paving the way for safer communities and more efficient law enforcement operations.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is the cornerstone of any significant project, and the development of a project is no exception. It is a rigorous and systematic examination of the proposed project, assessing various dimensions that collectively determine its viability and potential for success. The development of an Online crime reporting System holds the promise of transforming and optimizing the crime reporting processes easy. However, this transformation begins with a comprehensive feasibility study, a process that is multifaceted and essential to the success of the project.

- Technical feasibility
- Economic feasibility
- Behavioural feasibility

3.1.1 Economic Feasibility

Economic feasibility is the pragmatic evaluation of the financial aspects associated with developing and maintaining the system. The financial aspect examines the costs associated with system development, implementation, and long-term operation. It factors in initial development costs, software licensing fees, hardware procurement, and ongoing expenses like maintenance and support. By calculating potential return on investment (ROI), organizations can gauge whether the project aligns with their financial objectives considering factors like reduced costs, time savings, improved user experience, and enhanced operational efficiency. This involves a meticulous examination whether the organizations operate within budgetary constraints. It ensures that the project is financially sustainable without straining the organization's resources.

3.1.2 Technical Feasibility

The technical feasibility of the system is crucial to determine whether it can be practically implemented from a technological standpoint. Evaluating various aspects is essential for its success. Firstly, ensure that your organization has the necessary technology infrastructure or can acquire it to support the platform, including web hosting, databases, and server capacity. This aspect evaluates whether an organization possesses the technical capabilities and infrastructure required to develop and maintain the system. It considers factors such as software development tools, hardware, and database management. It encompasses several key aspects that are crucial.

3.1.3 Behavioral Feasibility

Behavioral feasibility refers to the analysis and assessment of whether the proposed project or system is acceptable and practical from the perspective of its intended users and stakeholders. It focuses on understanding and predicting how individuals and groups will respond to and interact with the project, considering their attitudes, preferences, behavior, and willingness to adopt and use the system. Behavioral feasibility aims to identify potential barriers, resistance, or challenges that might arise during the implementation and operation of the project, and it seeks to ensure that the project aligns with the needs and expectations of its target audience.

Examining the platform's acceptance and usability by its target users is one of the behavioral feasibility criteria in online crime reporting . To learn more about what users want, need, and anticipate from an online gift shop, you can conduct user surveys or focus groups. The platform's functionality and design can be tailored to the preferences of users by considering their behavior and adoption trends. To make sure that users have a seamless and satisfying experience, user training and onboarding resources may be required.

3.1.4 Feasibility Study Questionnaire

Project Overview:

The Project entitled “Online Crime Reporting System” aims to enhance the efficiency and transparency of crime reporting and investigation processes. It facilitates the entire crime management lifecycle, from reporting incidents to case resolution, providing a seamless experience for law enforcement, administrators, and the public.

To What Extent the System is Proposed For:

The proposed system is primarily designed for law enforcement agencies, officers, and the general public. It empowers citizens to report crimes, assists law enforcement in managing cases, and provides administrators with tools to oversee the system effectively.

Specify the Viewers/Public Involved in the System:

The viewers/public involved in the system include administrators, users, visitors.

List of Modules Included in the System:

a) Administrator Module
b) User Module
c) Visitor Module
d) Law Enforcement

Q: What challenges are you currently facing in managing criminal incidents that this system aims to address?

A: Our current crime management processes lack efficiency and transparency. We struggle with the timely reporting of incidents, coordination among law enforcement agencies, and maintaining a comprehensive database of criminal activities.

Q: How is user authentication and authorization currently managed within your law enforcement operations?

A: User authentication primarily relies on secure access credentials. We must ensure secure user authentication by utilizing robust authentication systems and following security best practices to protect sensitive data.

Q: Could you describe the types of crimes and incidents you handle, and how are they currently managed?

A: Currently, these incidents are managed through manual reporting, paper documentation, and legacy systems. There is a need for a centralized system to streamline incident reporting.

Q: How do you currently monitor the performance of your law enforcement operations, including case resolution and public satisfaction?

A: Performance monitoring is currently decentralized, making it challenging to assess the overall effectiveness of our operations. We lack a centralized dashboard for real-time insights. The system aims to implement performance tracking and feedback mechanisms to enhance operations.

Q: What is the expected timeline for implementing the crime reporting system?

A: We anticipate that the implementation of the system will take around 12-18 months, considering the complexity of integrating with existing systems and ensuring data security.

Q: What budget considerations are there for this project?

A: We have allocated a budget that includes software development costs, hardware infrastructure, security measures, training, and ongoing operational expenses.

Q: Do you have any specific security and privacy requirements for sensitive crime data, ensuring compliance with data protection regulations?

A: Yes, we have stringent security and privacy requirements to protect sensitive crime data, including encryption, access controls, and compliance with relevant data protection regulations.

Q: Are there any existing software or systems that need to be integrated with this crime reporting system?

A: Yes, we have existing systems for criminal records, evidence management, and incident reporting that need to be integrated with the new crime reporting system for seamless data exchange and efficiency.

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel Core i3
RAM	-	4.00 GB
SSD	-	256 GB

3.2.2 Software Specification

Front End	-	HTML, CSS
Back End	-	Python, Django
Database	-	SQLite
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 Django

Django stands as a high-level Python web framework, renowned for its ability to simplify and expedite the web development process. Offering a robust toolkit, libraries, and established patterns, Django facilitates the creation of secure, scalable, and maintainable web applications. Following the Model-View-Controller architectural paradigm, Django underscores the separation of concerns, enhancing code reusability and maintainability. A cornerstone of Django lies in its potent Object-Relational Mapping (ORM) layer, abstracting database interactions and enabling developers to engage with the database through Python objects and methods. This abstraction streamlines database operations, eliminating the need for intricate SQL queries. Notably, Django boasts a built-in administration interface, automating the generation of forms, views, and Create, Read, Update, Delete functionality for database models. This feature expedites development by furnishing a ready-to-use backend for efficient data management. Moreover, Django advocates for modular development through its application structure. Developers can craft modular and pluggable apps, fostering reusability across multiple projects, curbing code redundancy, and amplifying overall productivity.

3.3.2 SQLite

SQLite is an open-source, lightweight, and self-contained relational database management system (RDBMS) that excels in simplicity and minimalism. Designed for embedded systems and scenarios where a standalone database is required, SQLite allows users to store, organize, and manage structured data with a focus on efficiency. Its self-contained nature means that it requires no separate server process and operates directly on the application's data files. SQLite is often integrated into mobile applications, desktop software, and small to medium-scale web projects due to its low overhead and ease of deployment. Despite its lightweight design, SQLite supports a significant subset of SQL for data manipulation and retrieval, making it a versatile and accessible choice for developers. Its small footprint, speed, and compatibility across various platforms contribute to its popularity in scenarios where a robust yet lightweight database solution is needed. The active community surrounding SQLite ensures ongoing support, frequent updates, and a wealth of documentation, solidifying its position as a reliable and pragmatic database solution for diverse applications

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

System design is a critical phase in the software development process, serving as the bridge between high-level requirements and actual implementation. It involves the detailed planning, structuring, and specification of a software system's architecture, components, and functionalities. The primary goal of system design is to transform the abstract concepts and ideas gathered during the requirements analysis phase into a concrete, well-defined blueprint for building the software. Effective system design is crucial for the successful development and deployment of software systems. It minimizes the risk of misunderstandings, errors, and costly revisions by providing a clear and comprehensive plan that guides the development team throughout the implementation phase.

4.2 UML DIAGRAM

A key tool in software engineering, Unified Modelling Language (UML) is well-known for its ability to visually express intricate systems and procedures. It offers a uniform set of graphical notations that make it easier to illustrate different facets of the behaviour and structure of a system. UML, which was developed via the cooperation of industry professionals, is now widely used in both academics and business. UML has gained widespread acceptance and adoption in both academia and industry. It serves as a powerful communication tool, enabling stakeholders, including developers, designers, and clients, to attain a shared understanding of system architecture, design, and functionality. It is an effective tool for communication that helps all parties involved—developers, designers, and clients—to comprehend the architecture, functioning, and design of the system. As a lingua franca that cuts over language boundaries and guarantees a consistent way to communicate complex software concepts, UML diagrams eventually improves the productivity and efficacy of the software development processClass Diagram

- Object Diagram
- Use case Diagram
- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- State Chart Diagram
- Deployment Diagram
- Component Diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. Fundamentally, they give an organized way to recognize and characterize the numerous functions a system provides and the ways in which different actors or entities can access these functions. Actors are shown alongside the particular use cases they interact with, whether they are representing systems, users, or other things. Actor-use case associations shed light on the nature of these interactions and the duties and responsibilities of each component in the system. Along with improving stakeholder communication, this comprehensive visual depiction offers a clear blueprint for system functionality, setting the groundwork covering the stages of the software life cycle during development. Taken as a whole, Use Case Diagrams play a crucial part in coordinating development activities with user expectations, guaranteeing that the finished software system successfully and efficiently accomplishes its intended function. A use case diagram contains four components.

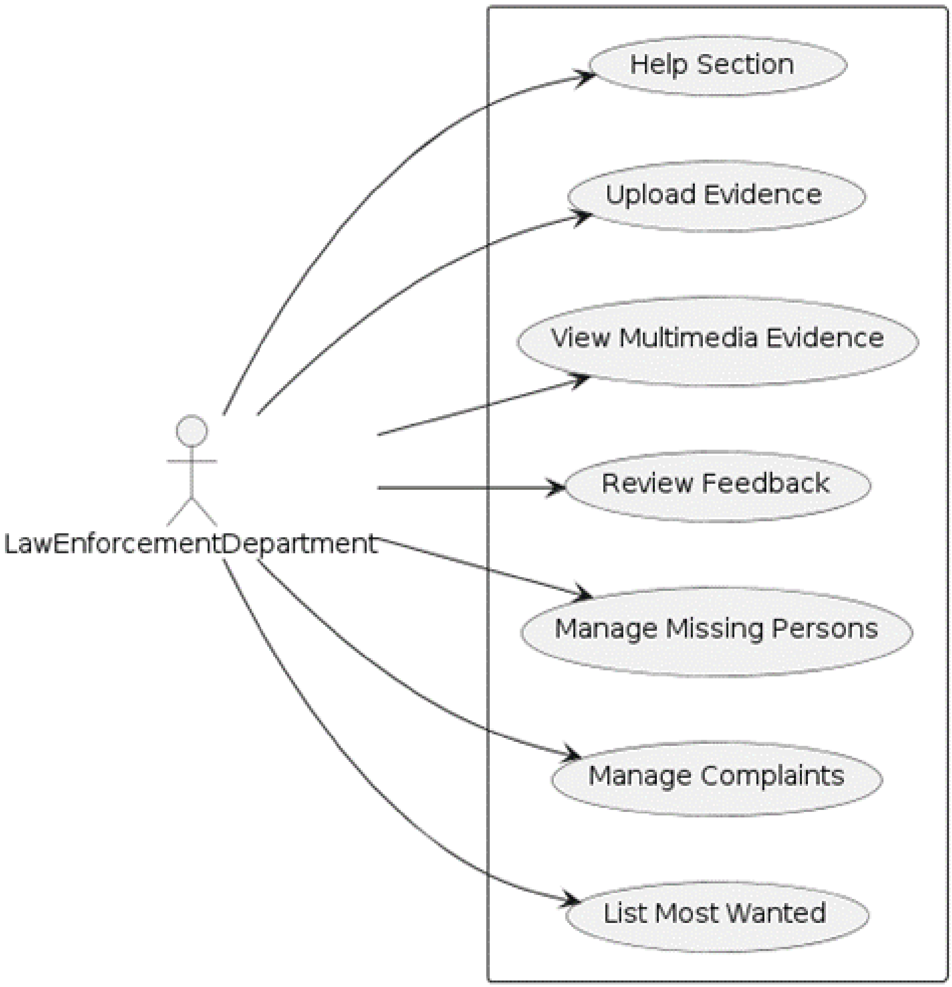
- Actor Definition: Clearly define and label all actors involved in the system. Actors are outside entities that communicate with the system.

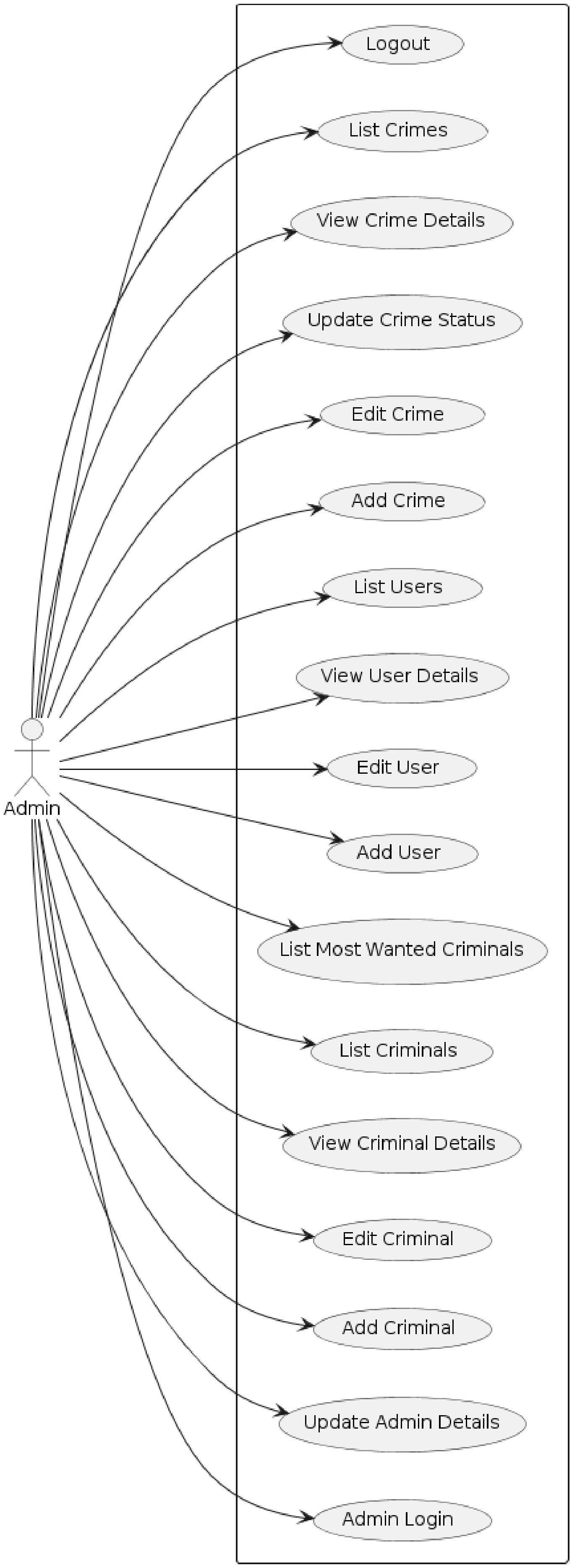
- Use Case Naming: Use descriptive names for use cases to accurately convey the functionality they represent.

- Association Lines: Use solid lines to represent associations between actors and use cases. This signifies the interaction between entities.

- System Boundary: Draw a box around the system to indicate its scope and boundaries. This defines what is inside the system and what is outside.

- Relationships: To express similar functionality shared by several use cases, utilise "include" relationships. To demonstrate alternative or additional functionality, use "extend" relationships


Help Section
Upload Evidence
View Multimedia Evidence
Review Feedback
Manage Missing Persons
Manage Complaints
List Most Wanted
LawEnforcementDepartment


Logout
List Crimes
View Crime Details
Update Crime Status
Edit Crime
Add Crime
List Users
View User Details
Edit User
Add User
List Most Wanted Criminals
List Criminals
View Criminal Details
Edit Criminal
Add Criminal
Update Admin Details
Admin Login
Admin

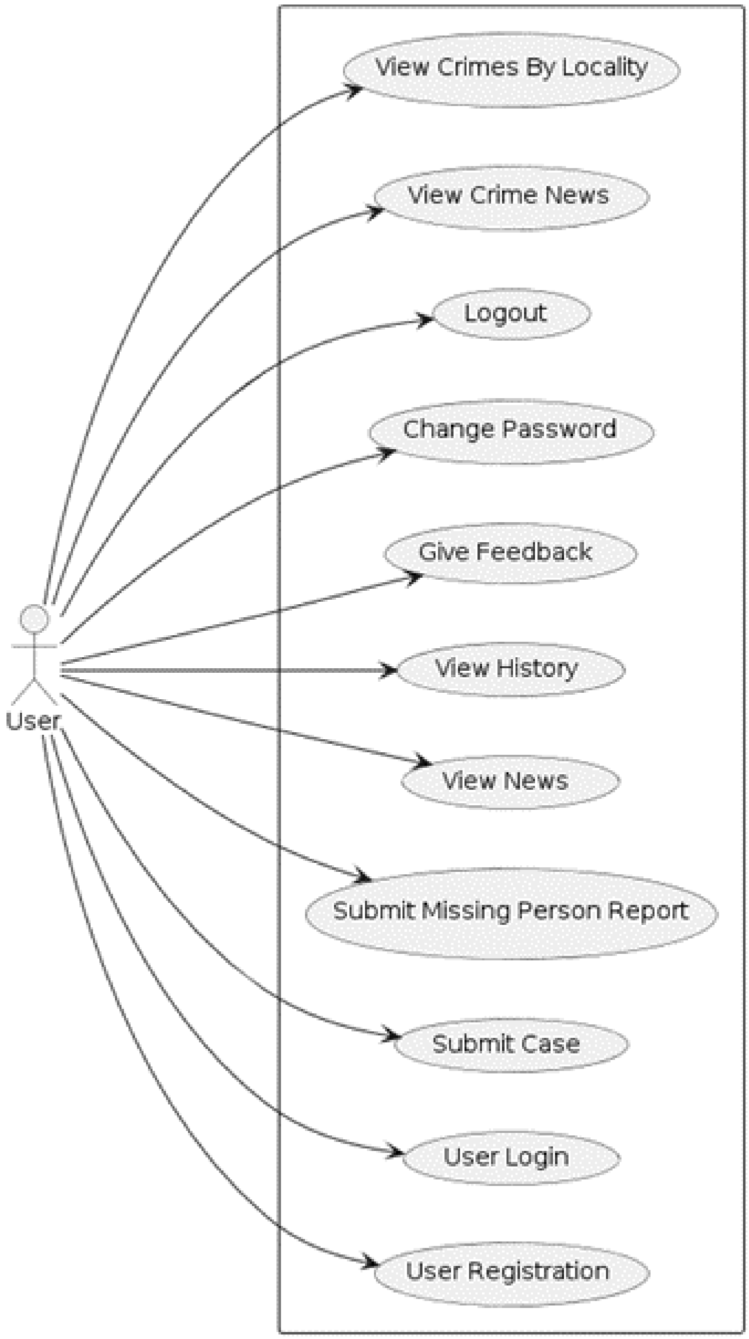


Fig 1: Use case diagram for Echo

4.2.2 SEQUENCE DIAGRAM

Sequence Diagrams serve as dynamic models in software engineering, illustrating the orderly progression of interactions between various components or parts of a system. They highlight the sequence in which messages are sent and received, providing insight into the system's behaviour over time. A vertical axis is used to depict actors and objects, and arrows are used to show the order and direction of the messages. Lifelines show the existence of actors or objects throughout the interaction by projecting vertical lines from them. These diagrams are an essential tool for comprehending the temporal elements of a software process and for visualizing the behaviour of systems. Sequence diagrams help stakeholders understand how various components work together to accomplish particular functionalities, which improves communication between development teams and stakeholders. These diagrams employ lifelines to depict the entities involved, and arrows to indicate the flow of messages or actions, aiding developers and stakeholders in comprehending the temporal aspects of system processes. In addition to helping identify possible bottlenecks or inefficiencies, this thorough representation offers a framework for improving system performance later on in the software development process.

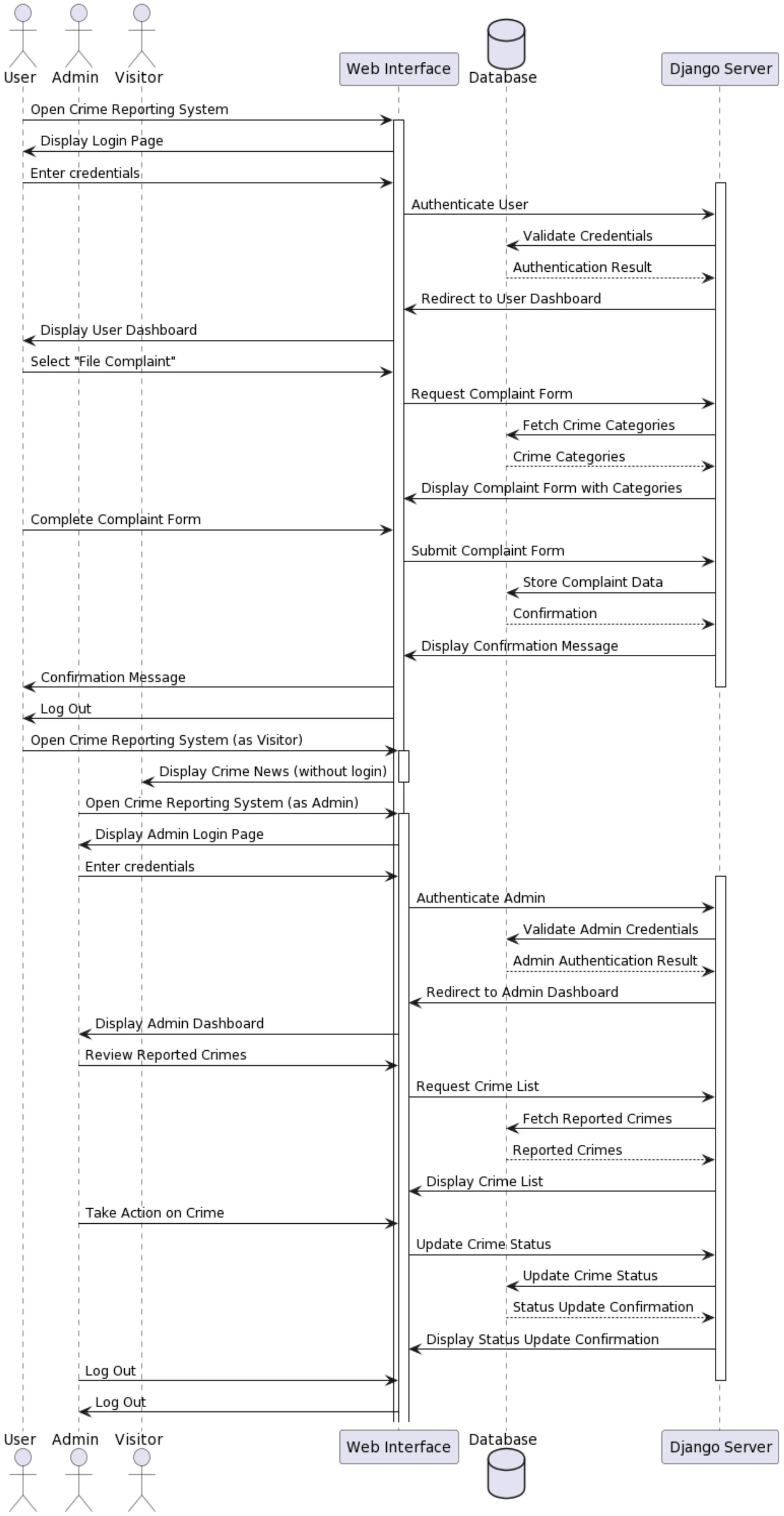


Fig 2: Sequence diagram for Echo

4.2.3 State Chart Diagram

The State Chart Diagram, which illustrates the several stages of an object's existence and the transitions between them, is a crucial component of the UML. It demonstrates how an entity undergoes state transitions and responds to external stimuli by acting dynamically. Every state denotes a unique stage in the object's life, while transitions show the circumstances that lead to changes in a state. The object's lifetime begins and ends at its initial and final states. Concurrent states are possible in orthogonal regions, allowing for the simultaneous capture of several facets of an object's action. The structured depiction of complicated behaviours is made possible by hierarchical states. Actions associated with entering or departing a state are represented by entry and exit actions. Guard criteria also guarantee that transfers take place only in certain situations. Events trigger these transitions, causing the system or object to move from one state to another. Actions, conditions, and triggers associated with each transition can also be specified. State chart diagrams are essential for comprehending and constructing the dynamic behaviour of systems, which helps create software that is reliable and responsive.

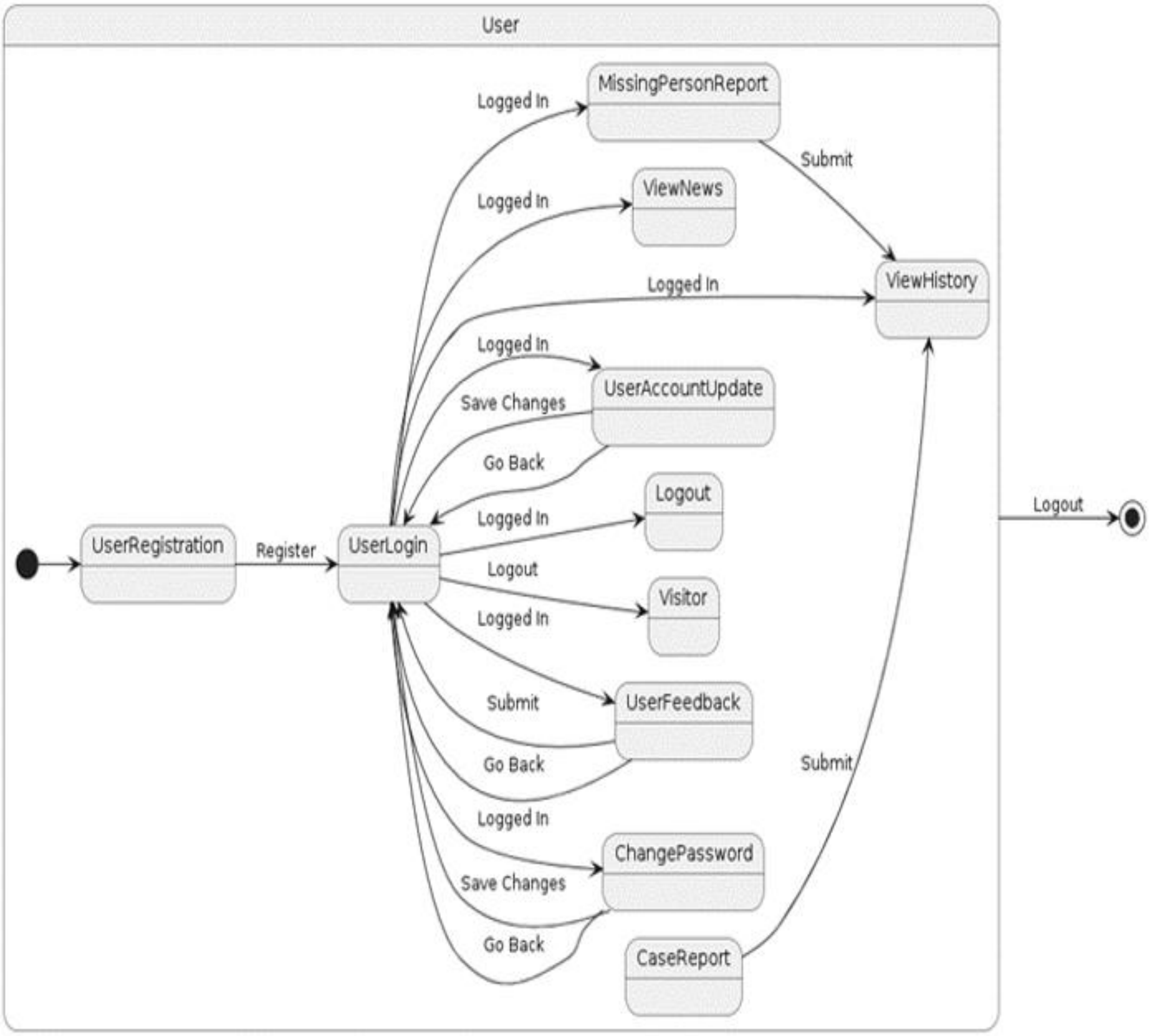
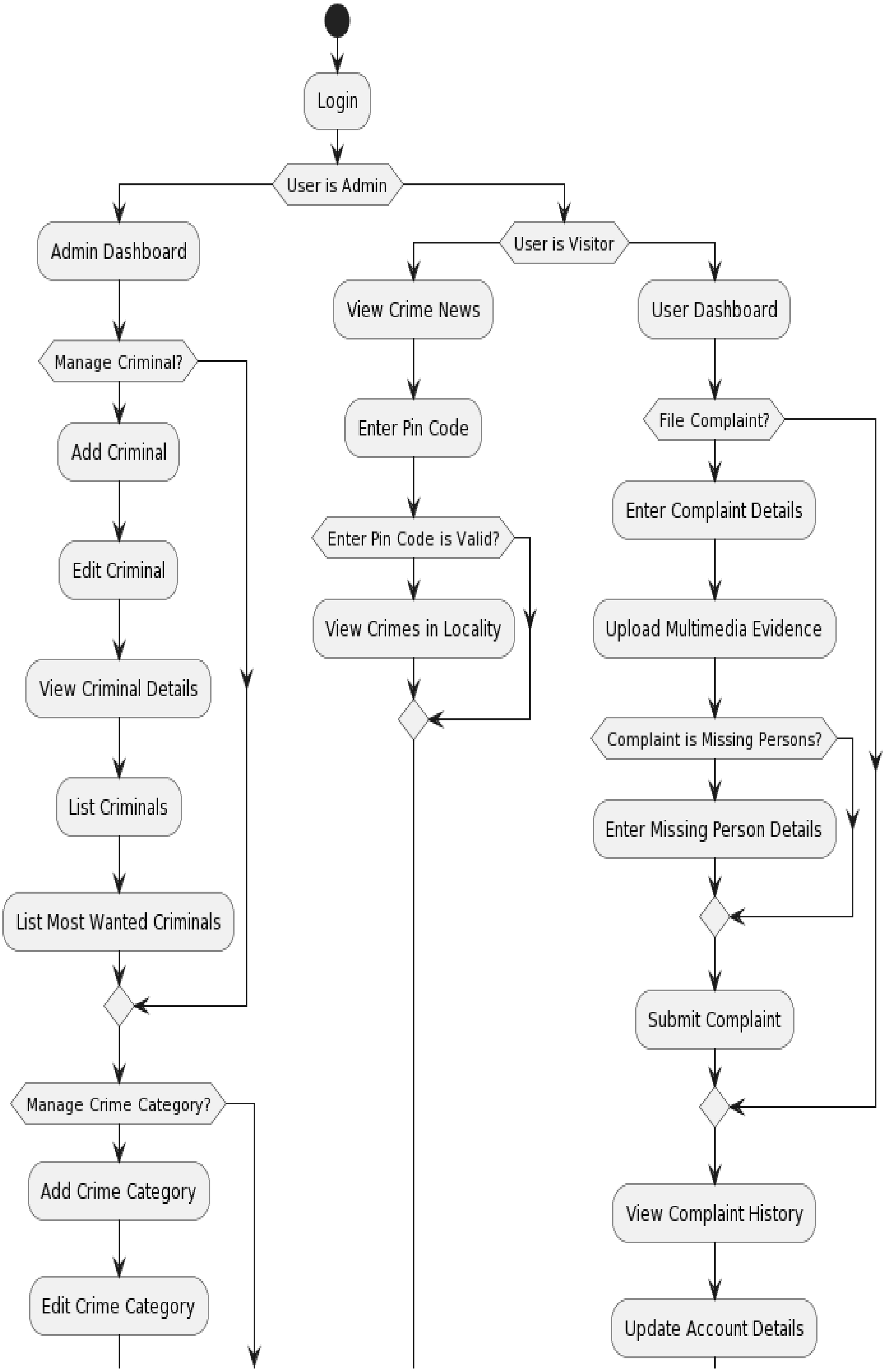


Fig 2: State Chart diagram for Echo

4.2.4 Activity Diagram

It is a visual representation within UML that guidelines the way, the activities and actions in a process, flow. It employs various symbols to depict tasks, decision points, concurrency, and control flows. Rectangles signify activities or tasks, while diamonds represent decision points, allowing for conditional branching. Arrows indicate the flow of control from one activity to another. Forks and joins denote concurrency, where multiple activities can occur simultaneously or in parallel. Swimlane segregate activities based on the responsible entity, facilitating clarity in complex processes. Initial and final nodes mark the commencement and completion points of the activity. Decision nodes use guards to determine the path taken based on conditions. Synchronization bars enable the coordination of parallel activities. Control flows direct the sequence of actions, while object flows depict the flow of objects between activities. Activity Diagrams serve as invaluable tools for understanding, modeling, and analyzing complex workflows in systems and processes. They offer a structured visual representation that aids in effective communication and system development


Login
User is Admin
Admin Dashboard
Manage Criminal?
Add Criminal
Edit Criminal
View Criminal Details
List Criminals
List Most Wanted Criminals
Manage Crime Category?
Add Crime Category
Edit Crime Category
User is Visitor
View Crime News
Enter Pin Code
Enter Pin Code is Valid?
View Crimes in Locality
User Dashboard
File Complaint?
Enter Complaint Details
Upload Multimedia Evidence
Complaint is Missing Persons?
Enter Missing Person Details
Submit Complaint
View Complaint History
Update Account Details

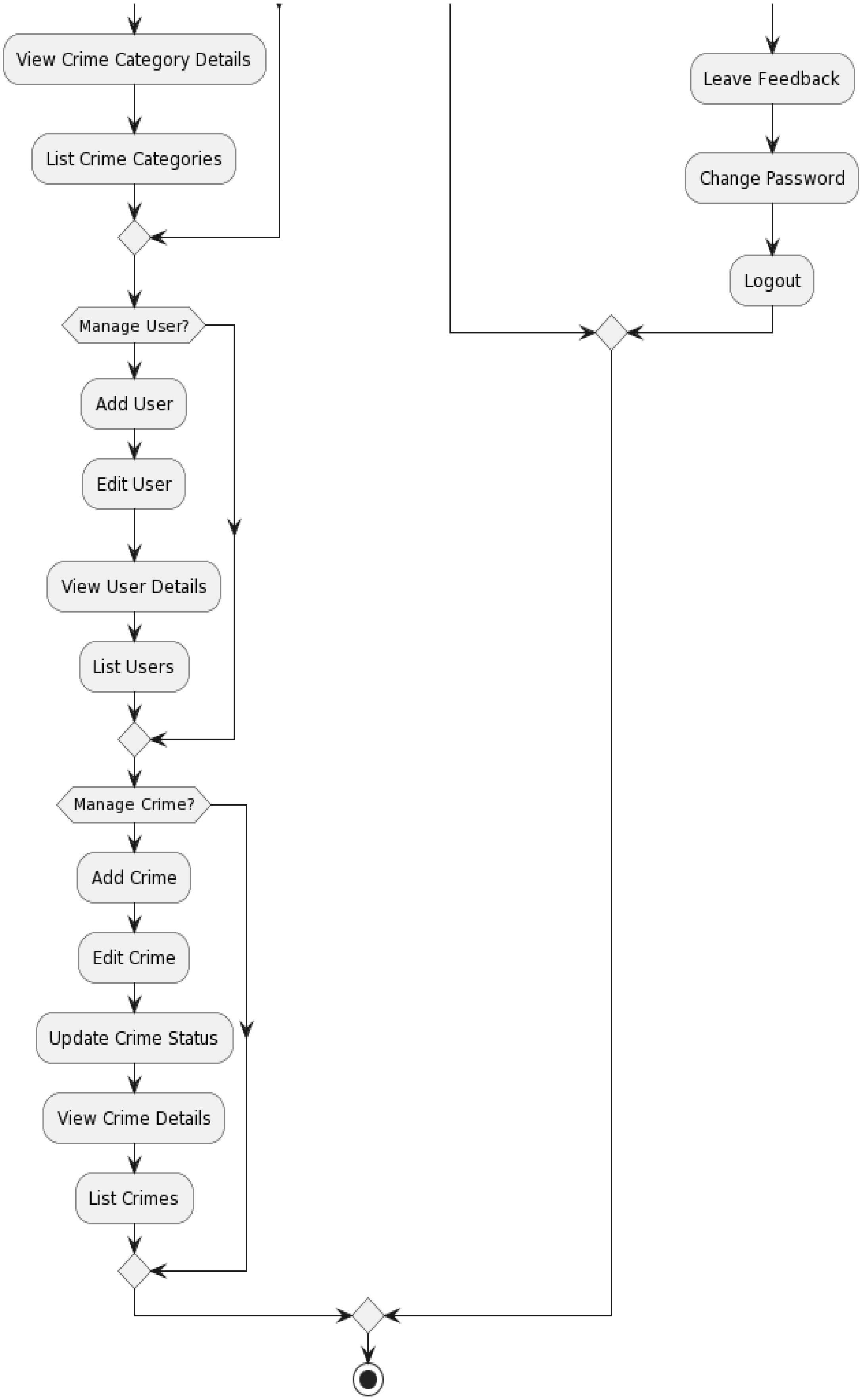


Fig 4: Activity diagram for Echo

4.2.5 Class Diagram

Class diagram represents the static view of the application. Classes, depicted as rectangles, encapsulate data and behavior within a system. Associations between classes indicate relationships, showcasing how they interact. Multiplicity notations specify the cardinality of associations. Inheritance is denoted by an arrow indicating the subclass inheriting from a super-class. Aggregation and composition illustrate whole-part relationships between classes. Interfaces, depicted as a circle, outline the contract of behavior a class must implement. Stereotypes provide additional information about a class's role or purpose. Dependencies highlight the reliance of one class on another. Association classes facilitate additional information about associations. Packages group related classes together, aiding in system organization..

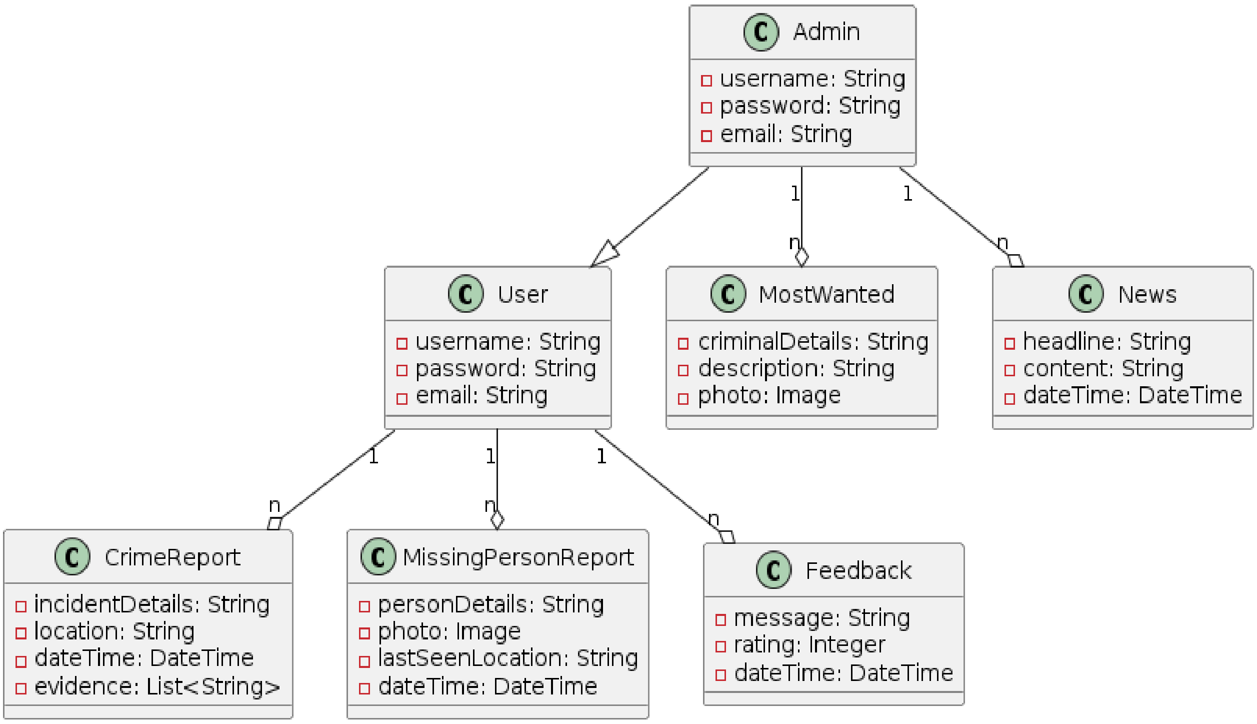


Fig 5: Class diagram for Echo

4.2.6 Object Diagram

It, in UML, provides a quick look of a system at a specific time, displayingthe instances of classes and their relationships. Links between items show relationships between them and emphasize their interactions. The number of instances engaged in associations is indicated by multiplicity notations. Attributes and their associated values show the status of the object. Object Diagrams provide a thorough understanding of runtime interactions, which helps with testing and system comprehension. They give a concrete depiction of class ties by concentrating on real world examples. Object Diagrams are comparable to Class Diagrams, except they place more emphasis on actual instances than class definitions. They are useful instruments for confirming that classes and associations function as anticipated in practice and validating system design. System validation relies heavily on object diagrams to make sure that the functions of system components and how they interact to match the requirements and intended design.

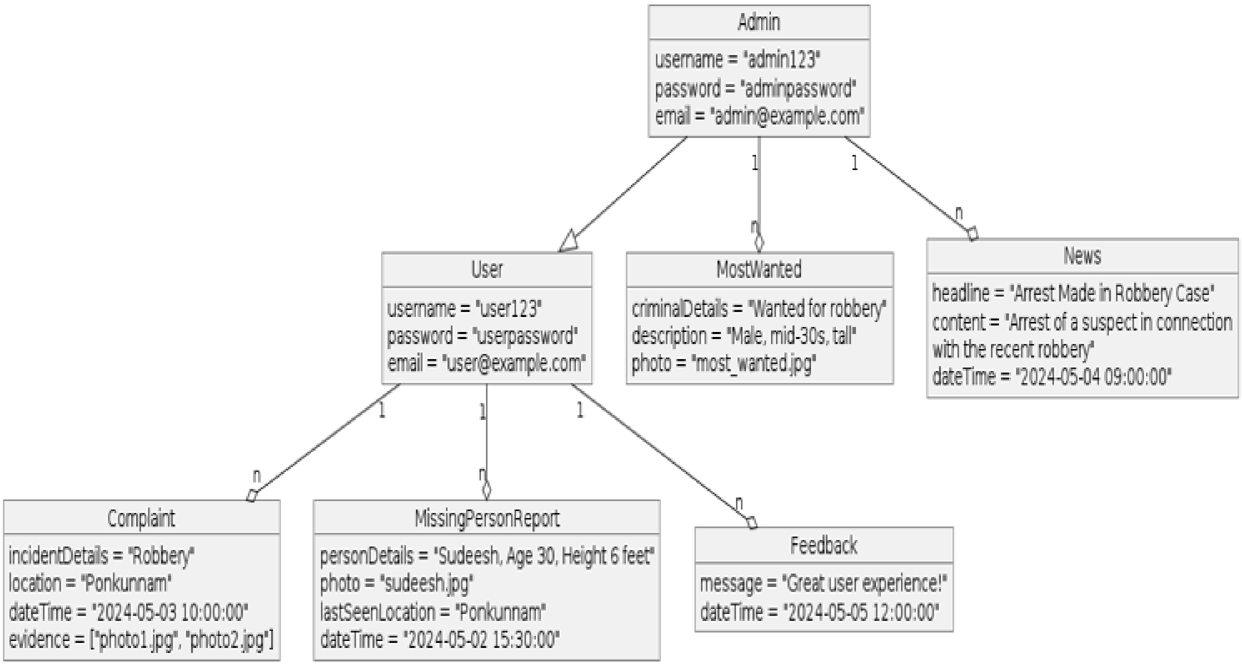

Admin
username = "admin123"
password = "adminpassword"
email = "admin@example.com"
1
1
n
n
User
username = "user123"
password = "userpassword"
email = "user@example.com"
MostWanted
criminalDetails = "Wanted for robbery"
description = "Male, mid-30s, tall"
photo = "most_wanted.jpg"
News
headline = "Arrest Made in Robbery Case"
content = "Arrest of a suspect in connection with the recent robbery"
dateTime = "2024-05-04 09:00:00"
1
1
1
n
n
n
Complaint
incidentDetails = "Robbery"
location = "Ponkunnam"
dateTime = "2024-05-03 10:00:00"
evidence = ["photo1.jpg", "photo2.jpg"]
MissingPersonReport
personDetails = "Sudeesh, Age 30, Height 6 feet"
photo = "sudeesh.jpg"
lastSeenLocation = "Ponkunnam"
dateTime = "2024-05-02 15:30:00"
Feedback
message = "Great user experience!"
dateTime = "2024-05-05 12:00:00"

Fig 6: Object Diagram for Echo

4.2.7 Component Diagram

Component diagrams have various behaviours and personalities. Components, depicted as rectangles, encapsulate modules, classes, or even entire systems. Dependencies between components are displayed through arrows, signifying the reliance of one component on another. Interfaces, represented by a small circle, outline the services a component offers or requires. Connectors link interfaces to denote the required or provided services. Ports, depicted as small squares, serve as connection points between a component and its interfaces. Stereotypes provide additional information about the role or purpose of a component. Deployment nodes indicate the physical location or environment in which components are deployed. Component Diagrams are instrumental in system design, aiding in the organization and visualization of system architecture

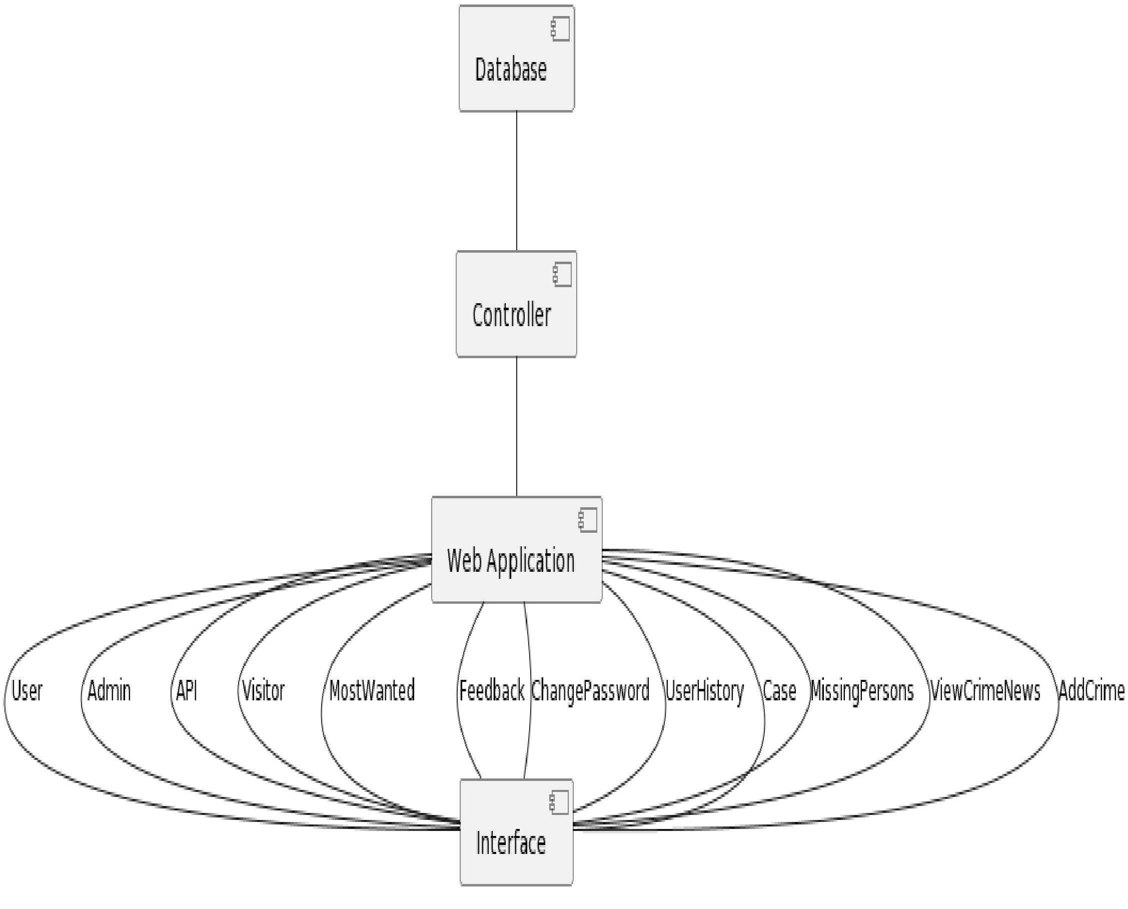


Fig 7: Component Diagram for echo

4.2.8 Deployment Diagram

It is a crucial facet of UML, provides a visual representation of the physical architecture, showcasing hardware nodes and components. Nodes, representing hardware entities like servers or devices, are depicted as rectangles. Artifacts, denoted by rectangles with a folded corner, represent software components or files deployed on nodes. Associations between nodes and artifacts indicate the deployment of software on specific hardware. Dependencies illustrate the reliance of one node on another. Communication paths, shown as dashed lines, represent network connections between nodes. Stereotypes provide additional information about the role or purpose of nodes and artifacts. Deployment Diagrams are instrumental in system planning, aiding in the visualization and organization of hardware and software components. They emphasize the allocation of software modules to specific hardware nodes, ensuring efficient utilization of resources. Overall, Deployment Diagrams perform an important role in orchestrating physical infrastructure of complex applications.

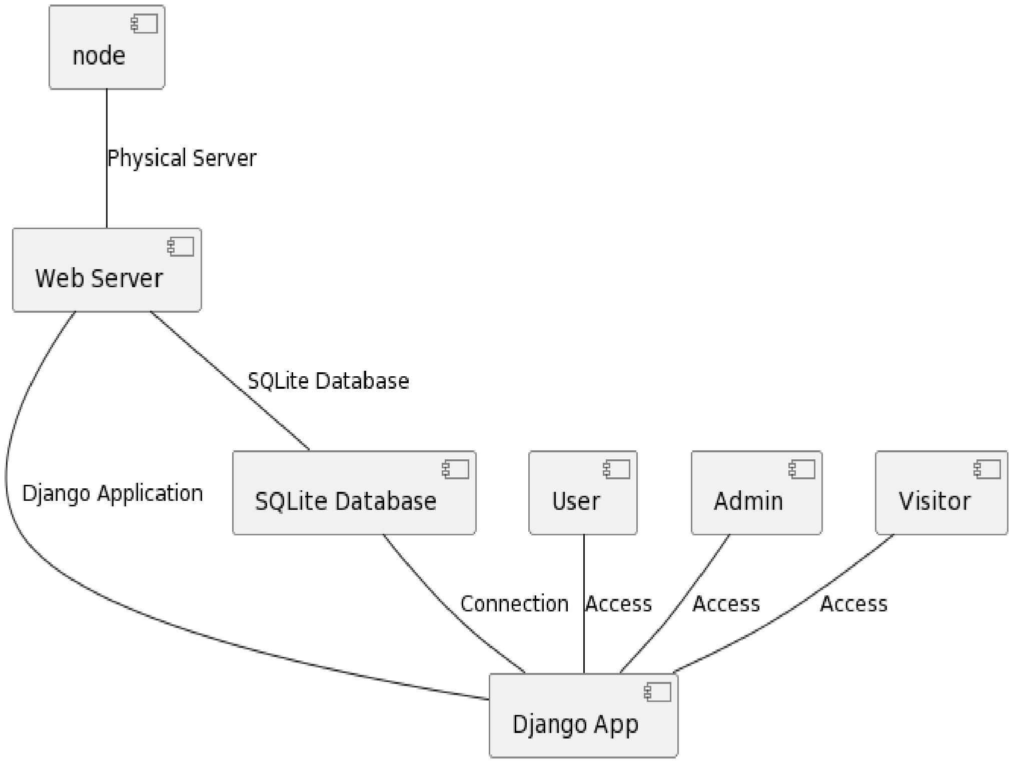
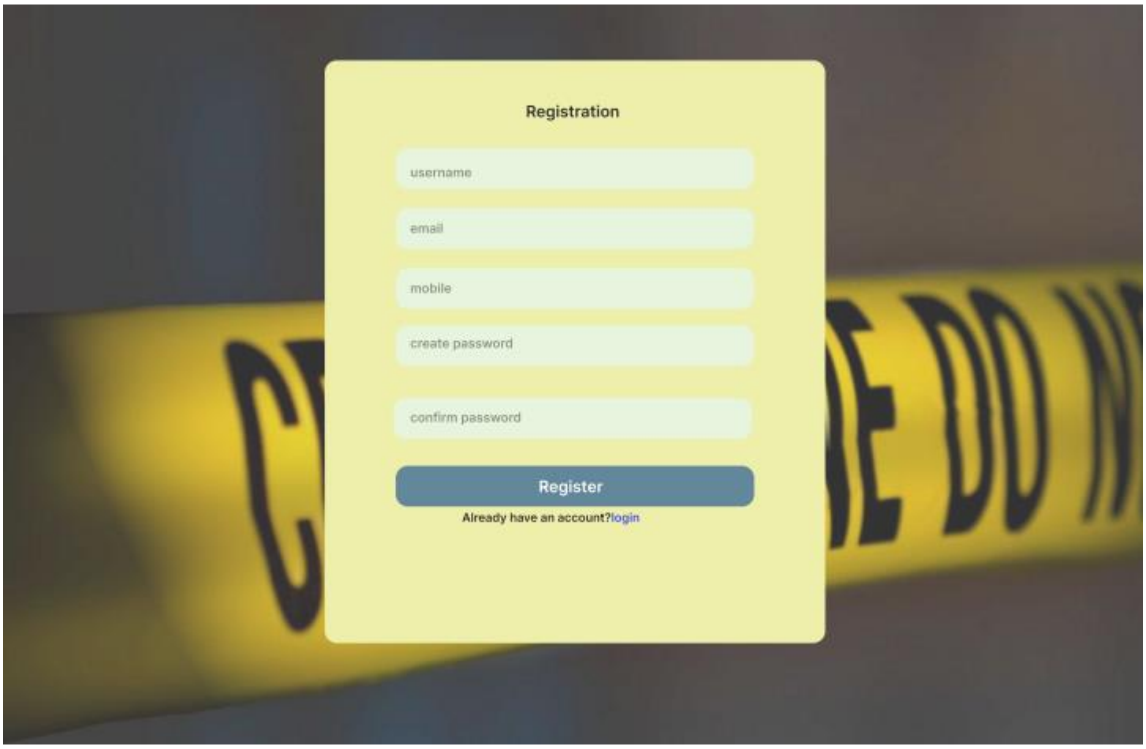


Fig 8: Deployment Diagram

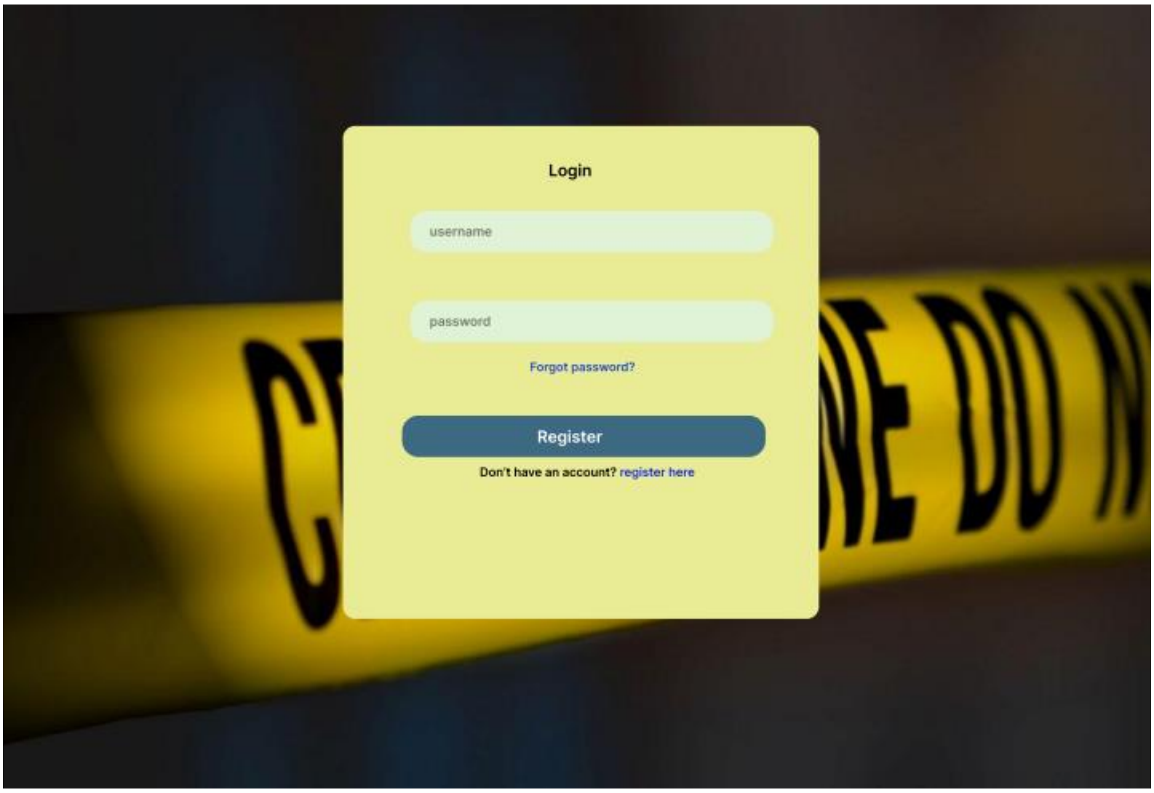
4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: Registration Page

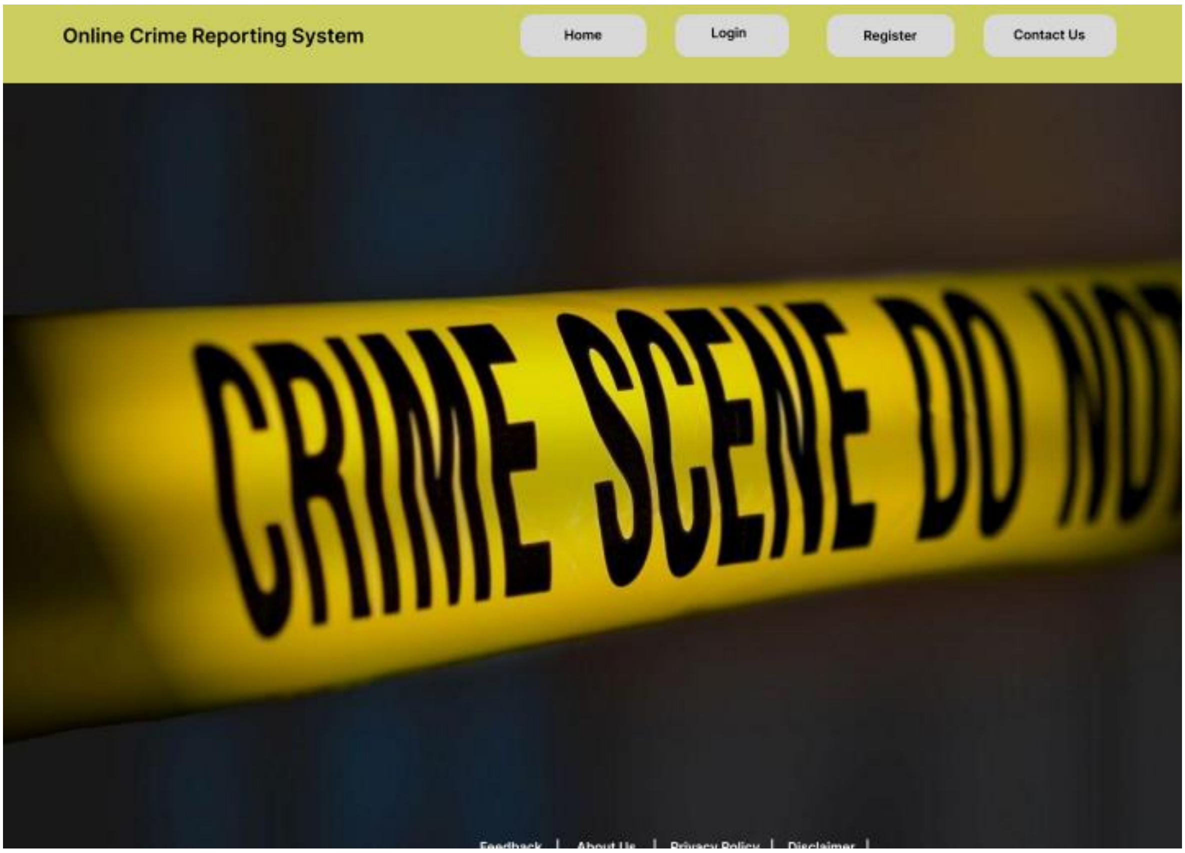


Already have an account?login

Form Name: Login Page



Form Name: Home Page


Online Crime Reporting System
Home
Login
Register
Contact Us
CRIME SCENE DO NOT
Feedback | About Us | Privacy Policy | Disclaimer |

4.4 DATABASE DESIGN

Database Design is important component in realm of management and software development. It works by the thoughtful and systematic organization of data to ensure efficient storage, retrieval, and manipulation. A well-designed database serves as the backbone of applications, enabling them to handle large volumes of information with speed and accuracy. This process encompasses defining the structure, relationships, and constraints of data entities, optimizing for performance and scalability. Effective database design is pivotal in minimizing redundancy, ensuring data integrity, and providing a foundation for robust data analytics. It involves a deep understanding of business requirements and user needs, translating them into a coherent and logical data model. The goal of a sound database design is to create a reliable, scalable, and maintainable system that supports the organization's objectives and facilitates seamless information flow.

4.4.1 Relational Database Management System (RDBMS)

Data is represented via a relational model as a set of relationships. Every relationship can be thought of as a dataset or table of values. Tables are referred to as relations, rows as tuples, and column headers as attributes in relational model nomenclature. Tables make up a relational file, and each one has its own name. A set of linked values is represented by a row in the chart.

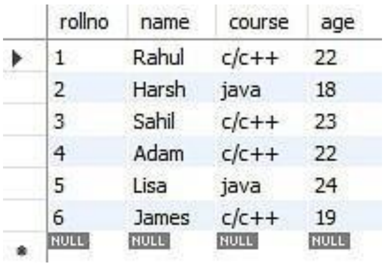
4.4.2 Normalization

To reduce the impact of future changes on data structures, data are arranged simply. Normalisation is a structured procedure that divides big tables into smaller ones, removes unnecessary fields, and upholds integrity. Its goal is to prevent anomalies from occurring when data is updated, deleted, and inserted. In data modelling, normal form is predicated on two ideas: keys and relationships. A foreign key discovers a record from a separate table, whereas a primary key uniquely identifies a row in a table. In order to produce a logical arrangement of data into appropriate tables that the user can readily correlate, all tables are normalised till the third normal form. By removing repetitive groupings of data, normalisation helps to reduce the load on computer resources and prevent data redundancy.

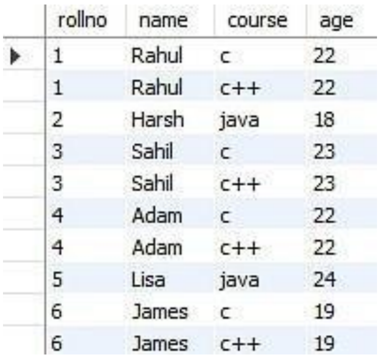
First Normal Form (1NF)

If a table has an atomicity of 1, it is said to be in its first normal form. Atomicity in this instance forbids a single cell from storing more than one value. It is limited to one value and one attribute. The usage of composite attributes, multi-valued attributes, and their combinations is forbidden in the first normal form.

Example: The table of students' records below includes details on each student's age, course, course number, and roll number. You can see that the course column in the students record table has two values. As a result, it deviates from the First Normal Form.



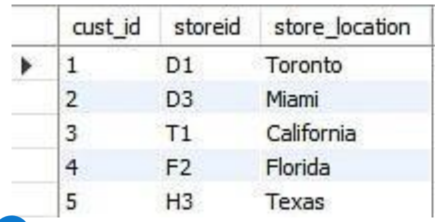
rollno	name	course	age
1	Rahul	c/c++	22
2	Harsh	java	18
3	Sahil	c/c++	23
4	Adam	c/c++	22
5	Lisa	java	24
6	James	c/c++	19
NULL	NULL	NULL	NULL



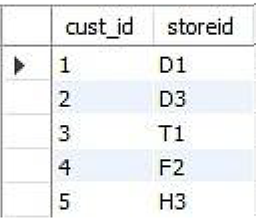
rollno	name	course	age
1	Rahul	c	22
1	Rahul	c++	22
2	Harsh	java	18
3	Sahil	c	23
3	Sahil	c++	23
4	Adam	c	22
4	Adam	c++	22
5	Lisa	java	24
6	James	c	19
6	James	c++	19

Second Normal Form (2NF)

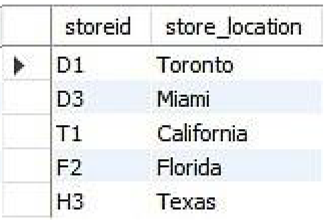
A table must initially be in First Normal Form in order to be in Second Normal Form. Additionally, the table cannot have partial dependency, which is the condition that the correct subset of the candidate key yields a non-prime attribute. Let's now use an example to further grasp the Second Normal Form. To get the table to Second Normal Form, you have to split it into two parts. After that, you'll see the tables below:



cust_id	storeid	store_location
1	D1	Toronto
2	D3	Miami
3	T1	California
4	F2	Florida
5	H3	Texas



cust_id	storeid
1	D1
2	D3
3	T1
4	F2
5	H3

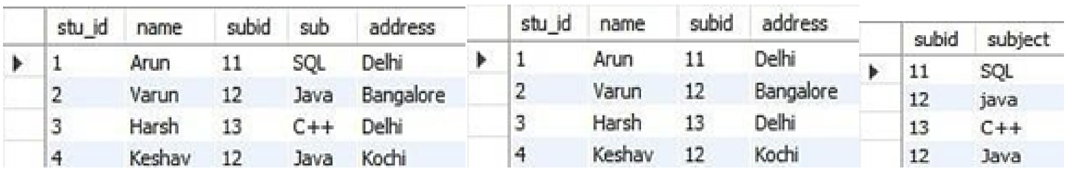


storeid	store_location
D1	Toronto
D3	Miami
T1	California
F2	Florida
H3	Texas

The column store location is completely dependent on the main key of that table, storied, as you have removed the partial functional reliance from the location table.

Third Normal Form (3NF)

To achieve the Third Normal Form, a table must first satisfy the Second Normal Form. The second criterion is that there should be no transitive dependency for non-prime attributes. This means that non-prime attributes, which are not part of the candidate key, should not depend on other non-prime attributes in the table. A transitive dependency occurs when attribute A determines attribute C indirectly through attributes B and B determines C (where B does not determine A). Consider a table with student information including student name, subject ID, student ID, and subject. In this table, subject ID and subject are determined by student ID. Therefore, subject is determined by student ID through subject ID. This table fails to meet the third normal form condition due to this transitive functional dependency. To convert the table to the third normal form, it should be split as follows:



stu_id	name	subid	sub	address
1	Arun	11	SQL	Delhi
2	Varun	12	Java	Bangalore
3	Harsh	13	C++	Delhi
4	Keshav	12	Java	Kochi

stu_id	name	subid	address
1	Arun	11	Delhi
2	Varun	12	Bangalore
3	Harsh	13	Delhi
4	Keshav	12	Kochi

subid	subject
11	SQL
12	java
13	C++
12	Java

4.4.3 Sanitization

Data sanitization is a crucial aspect of web development, particularly when dealing with forms that require users to input personal information which is then sent to the database. Any data submitted in an invalid format has the potential to compromise the security of the DBMS. Therefore, to prevent hackers from gaining access to the database, it is essential to sanitize and filter all user- entered data before sending it to the database. By ensuring that user-entered data is properly formatted, meets certain criteria, and is free from malicious input, developers can prevent errors and security vulnerabilities in their applications.

4.4.4 Indexing

Indexing decreases the amount of disc accesses needed to execute a query, thereby improving database speed. It's a data structure method used to quickly find and retrieve data from a database. Indexes are created in databases by using many columns. The main key of the table is duplicated or has the potential to be duplicated in the search key, which is usually the first column. The sorted order of these variables makes it easier to retrieve related data quickly. It's crucial to remember that the data itself might not be kept in an ordered fashion.

4.5 TABLE DESIGN

1. User

Primary Key: **user_id**

No:	Field name	Datatype	Key Constraints	Description of the field
1	user_id	AutoField	Primary Key	Primary Key of the user.
2	email	EmailField	unique=True	Email address of the user.
3	username	CharField	Not Null	Username of the user.
4	firstname	CharField	Not Null	First name of the user.
5	lastname	CharField	Not Null	Last name of the user.
6	role	CharField	Not Null	User role (User, admin).

2. Feedback

Primary Key: **feedback_id**

Foreign Key: **user** references table **User**

No:	Field name	Datatype	Key Constraints	Description of the field
1	feedback_id	IntegerField	Primary Key	Feedback ID
2	user	CharField	Foreign Key	Username
3	message	CharField	Not Null	Content
4	date	DateTimeField	Not Null	Date submitted

3. Complaint

Primary Key: **complaint_id**

Foreign Key: **applicant_id** references table **User, police_id** reference**s table PoliceStation**

No:	**Field name**	**Datatype**	**Key Constraints**	**Description of the field**
1	complaint_id	InetegerField	Primary Key	Complaint ID
2	applicant_id	IntegerField	Foreign Key	ID of user filing the complaint
3	police_id	CharField	Foreign Key	ID of the investigating police station
4	date	DateField	Not Null	Date when the complaint was filed
5	idate	DateField	Not Null	Date of the incident
6	time	CharField	Max length: 6	Time of the incident
7	address	CharField	Max length=100	Address where the incident occurred
8	landmark	CharField	Max length=50	Landmark near the incident location
9	charge	CharField	Max length=100	Description of the charges
10	victim	CharField	Max length=10	Name of the victim
11	ifname	CharField	Max length=50	First name of the informer
12	ilname	CharField	Max length=50	Last name of the informer
13	dob	DateField	Not Null	Date of birth of the informer
14	iaddress	CharField	Max length=100	Address of the informer
15	status	BooleanField	Default=False	Status of the complaint

4. MissingPerson

Primary Key: **missingperson_id**

Foreign Key: **user** references table **User**

No:	Field	Data Type	Key Constraints	Description of the field
1	missingperson_id	IntegerFild	Primary Key	Missing person ID
2	user	CharField	Foreign Key	User ID
3	gendar	CharField	Not Null	Gender
4	description	TextField	Not Null	Description of the missing person
5	photo	ImageField	Default=current time	Image of missing person
6	status	CharField	Not Null	Status of the complaint
7	age	IntegerField	Not Null	Age

5. Table: Blog

Primary key: BID

No:	Field Name	Data Type	Key Constraints	Description of the field
1	BID	IntegerField	Primary Key	Post id
2	title	CharField	Not Null	Title of the post the admin is sharing
3	content	TextField	Not Null	Content of the blog the admin is posting
4	published_date	DateTimeField	Default=current time	Publish date of the post
5	image	ImageField	Not Null	Images related to the content that the admin is sharing

6. Table: PoliceStation

Primary key: station_id

No:	Field name	Datatype	Key Constraints	Description of the field
1	station_id	CharField	Primary key	Station id
2	station_name	CharField	Max length=30	Station name
3	password	CharField	Not Null	password
4	email_id	EmailField	Not Null	Email address of the station
5	address	CharField	Max length=100	Address of the station
6	phone	IntegerField	Max length=15	Phone number of the station
7	pic	ImageField	Not Null	Image of the police station

7. Table: Comment

Primary key: cid

No:	Field	Type	Key Constarints	Description
1	cid	IntegerField	Primary Key	Id of the comment
2	post_id	ForeignKey	Not Null	Post id
3	author_id	ForeignKey	Not Null	Author id
4	text	TextField	Not Null	Comment content
5	created_date	DateTimeField	Not Null	Comment posted date

8. Table: MostWanted

Primary key: mid

No:	Field	DataType	Key Constarints	Description of the field
1	mid	IntegerField	Primary Key	unique identifier for each record
2	full_name	CharField	Max length=30	Full name of the wanted person
3	alias	CharField	Max length=50	Alias or nickname of the wanted person
4	date_of_birth	DateField	Not Null	Date of birth of the wanted person
5	place_of_birth	CharField	Not Null	Place of birth of the wanted person
6	gender	CharField	Max length=10	Gender of the wanted person
7	nationality	CharField	Max length=30	Nationality of the wanted person
8	height	CharField	Max length=20	Height of the wanted person
9	weight	CharField	Max length=20	Weight of the wanted person
10	eye_color	CharField	Max length=20	Eye color of the wanted person
11	hair_color	CharField	Max length=20	Hair color of the wanted person
12	distinguishing_features	TextField	Max length=100	Any distinguishing features
13	offenses_committed	TextField	Max length=100	Offenses committed by the wanted person
14	last_known_location	CharField	Not Null	Last known location of the wanted person
15	rewards_offered	CharField	Not Null	Rewards offered for information leading to capture
16	recent_photographs	ImageField	Not Null	Recent photographs of the wanted person

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the methodical process of executing a program to find any possible bugs or problems. A well-designed test case has a high probability of identifying problems that were previously overlooked. When a test uncovers an error that was previously unknown, it is deemed successful. A test can identify software defects if it runs according to plan and achieves its goals. The test indicates whether the computer program is running at its best and according to its planned functionality. A computer program can be evaluated using three main methods: computational complexity analysis, correctness evaluation, and implementation efficiency evaluation.

5.2 TEST PLAN

A comprehensive document that outlines the approach, parameters, goals, means, timetable, and anticipated results for a particular testing project is called a test plan. It serves as a framework for conducting testing operations, ensuring that each step of the process is carefully planned out and carried out. Additionally, the test plan establishes the roles and responsibilities of team members, outlines the required testing environment, and sets forth the criteria for the successful completion of testing activities. This document plays a pivotal role in ensuring that the testing phase is conducted in a structured and effective manner, ultimately contributing to the overall success of the project.

The testing levels include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

Overall, defining the scope, identifying the test environment, defining the test cases, generating the test scripts, running the tests, recording the results, analyzing the results, and reporting the results are all steps in the Selenium test plan creation process. By doing the following actions, you can make sure that your testing is thorough and efficient and that you are able to spot and fix any issues before they become serious ones.

5.2.1 Unit Testing

Unit Testing is not only a meticulous examination of discrete units within a softwaresystem but also an indispensable quality assurance measure. This phase serves a crucial foundation for the testing process, where the focus lies on isolating and scrutinizingindividual units of code. The objective remains unwavering: to verify that each unit performs its designated function accurately, yielding precise outputs for predefined inputs. Moreover, Unit Testing operates independently, detached from other components, and any external dependencies are either emulated or replaced by "mock" objects, ensuring controlled evaluation. This meticulous process establishes a robust foundation for the software, confirming that each unit functions reliably and adheres meticulously to its predefined behavior.

5.2.2 Integration Testing

It stands as a pivotal phase in the software testing process, dedicated to scrutinizing the interactions and interfaces among diverse modules or components in a software system. Primary objective is ascertaining that individual units of code seamlessly converge to create a unified and functional system. In stark contrast to unit testing, which assesses individual units in isolation, integration testing delves into the interplay between these units, with a keen eye for any disparities, communication glitches, or integration hurdles. By subjecting the integrated components to rigorous testing, development teams aim to affirm that these elements function cohesively, addressing any potential issues before deployment. This systematic evaluation is instrumental in ensuring that the software operates as an integrated whole, free from any unforeseen conflicts or errors that may arise from the convergence of individual modules

5.2.3 Validation Testing or System Testing

It places end-users at the forefront of evaluation, ensuring that the software aligns precisely with their anticipated needs and expectations. This phase stands distinct from other testing methodologies, as its primary objective is to authenticate that the software, in its final form,serves its intended purpose seamlessly within the real-world scenarios it was designed for. As a culmination of the testing process, Validation Testing carries the responsibility of confirming that the software not only meets the defined technical specifications but also delivers genuine value to its users. It does so by scrutinizing the software against the backdrop of actual usage, thereby fortifying its readiness for deployment.

5.2.4 Output Testing or User Acceptance Testing

It also known as Results Validation, is a critical phase in the software testing process.Its primary focus is to verify the correctness and accuracy of the output generated by a software application. The goal is to make sure that the system produces the expected results for a given set of inputs and conditions

Key aspects of Output Testing include:

- Comparison with Expected Results
- Test Case Design

- Validation Criteria
- Data Integrity
- Precision and Completeness

5.2.5 Automation Testing

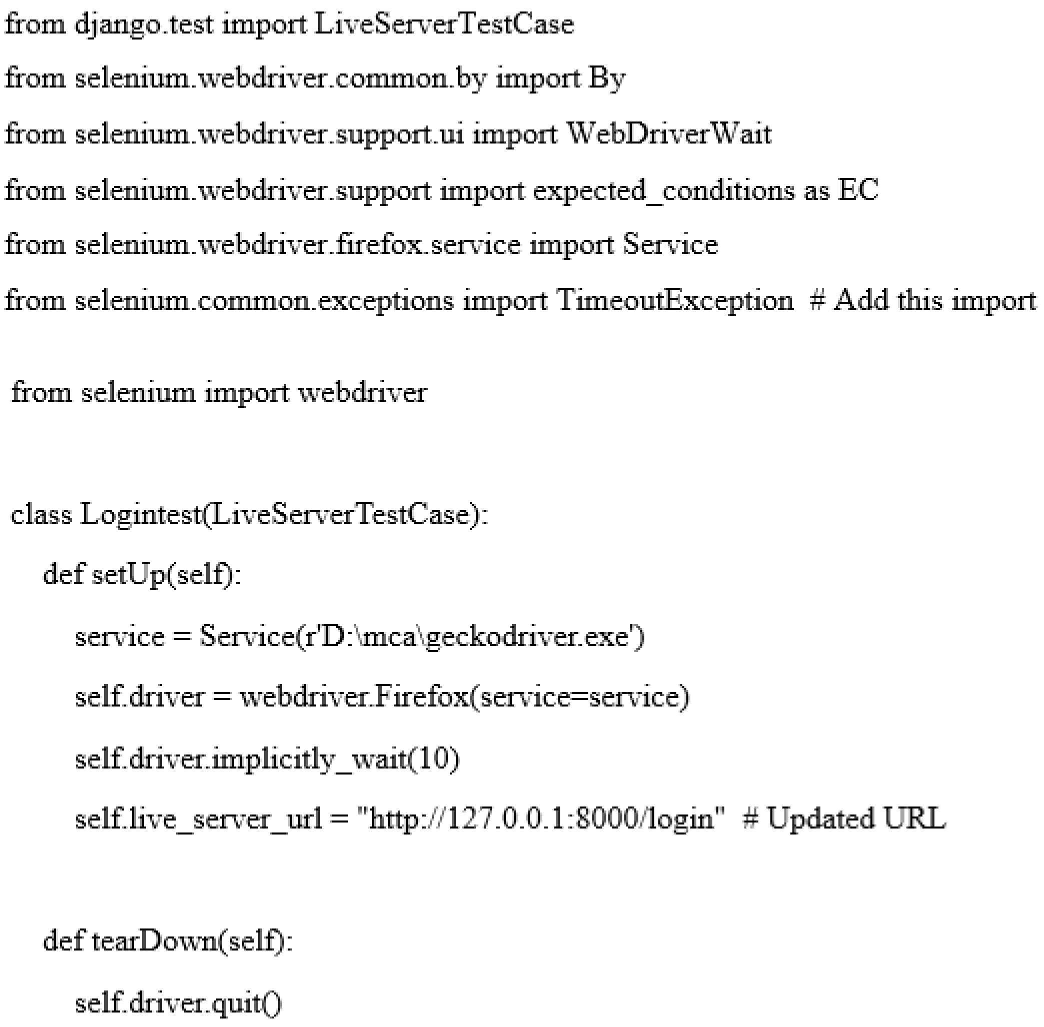
It stands as a cornerstone in the testing process, harnessing power of automated tools and scripts to meticulously execute test cases. In stark contrast to manual testing, which hinges on human intervention, automation testing brings forth a streamlined approach, employing software to conduct repetitive, intricate, and time-consuming tests. This methodology not only heightens operational efficiency but also significantly diminishes the likelihood of human error, ensuring precise and reliable results. Moreover, it empowers thorough testing across a diverse array of scenarios and configurations, from browser compatibility to load and performance assessments.

5.2.6 Selenium Testing

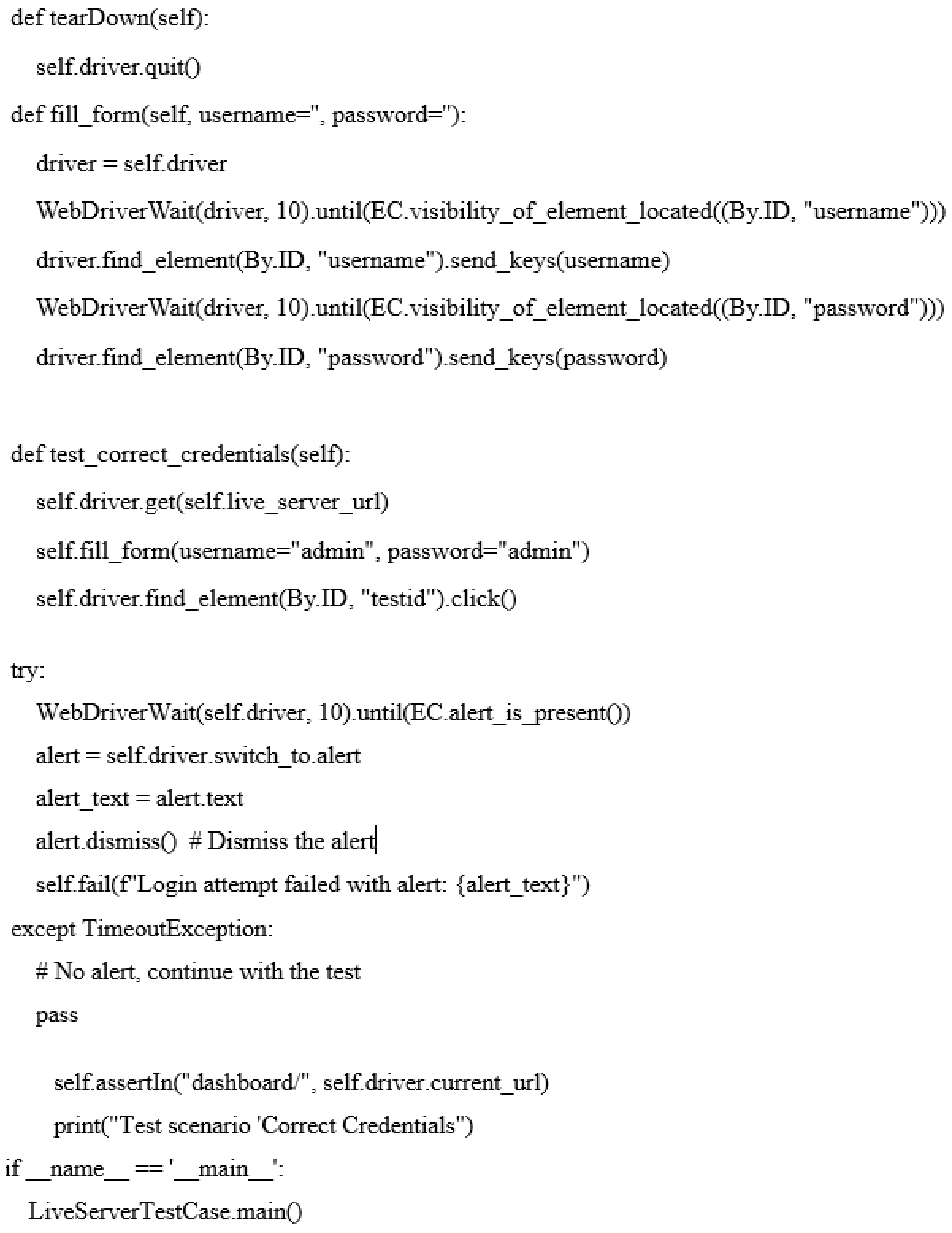
Selenium stands as a freely available, open-source automated testing framework, serving the purpose of validating web applications across diverse browsers and platforms. Its versatility lies in the ability to craft Selenium test scripts employing various programming languages such as Java, C#, Python, among others. The process of conducting tests using Selenium is commonly referred to as Selenium testing. Given its composite nature, Selenium comprises contributions from a multitude of developers. Highlighted below are pivotal individuals who have significantly impacted the Selenium project. Furthermore, Selenium's adaptability enables seamless integration with an array of testing frameworks, tools, and technologies. This includes pairing with testing frameworks like TestNG or JUnit to leverage advanced test management and reporting capabilities. Additionally, Selenium seamlessly interfaces with popular build tools, source control systems, and defect tracking systems, amplifying its utility and efficiency in the software testing landscape

Test Case 1: Admin login

code

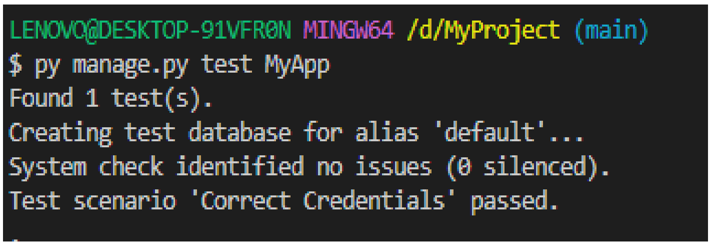


```
from django.test import LiveServerTestCase
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.firefox.service import Service
from selenium.common.exceptions import TimeoutException  # Add this import

 from selenium import webdriver


 class Logintest(LiveServerTestCase):
     def setUp(self):
         service = Service(r'D:\mca\geckodriver.exe')
         self.driver = webdriver.Firefox(service=service)
         self.driver.implicitly_wait(10)
         self.live_server_url = "http://127.0.0.1:8000/login"  # Updated URL

     def tearDown(self):
         self.driver.quit()
```



```
 def tearDown(self):
    self.driver.quit()
 def fill_form(self, username='', password=''):
    driver = self.driver
    WebDriverWait(driver, 10).until(EC.visibility_of_element_located((By.ID, "username")))
    driver.find_element(By.ID, "username").send_keys(username)
    WebDriverWait(driver, 10).until(EC.visibility_of_element_located((By.ID, "password")))
    driver.find_element(By.ID, "password").send_keys(password)


 def test_correct_credentials(self):
    self.driver.get(self.live_server_url)
    self.fill_form(username="admin", password="admin")
    self.driver.find_element(By.ID, "testid").click()

 try:
    WebDriverWait(self.driver, 10).until(EC.alert_is_present())
    alert = self.driver.switch_to.alert
    alert_text = alert.text
    alert.dismiss()  # Dismiss the alert
    self.fail(f"Login attempt failed with alert: {alert_text}")
 except TimeoutException:
    # No alert, continue with the test
    pass

      self.assertIn("dashboard/", self.driver.current_url)
      print("Test scenario 'Correct Credentials")
if __name__ == '__main__':
   LiveServerTestCase.main()
```

Screenshot:



```
LENOVO@DESKTOP-91VFR0N MINGW64 /d/MyProject (main)
$ py manage.py test MyApp
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test scenario 'Correct Credentials' passed.
```

Test Report:

Test Case 1

Project Name: ECHO	
Login Test Case	
Test Case ID: 1	**Test Designed By:** Sudeesh E S
Test Priority (Low/Medium/High): High	**Test Designed Date: 16/04/2024**
Module Name: Login Screen	**Test Executed By:** Ms. Nimmy Francis
Test Title: Admin Login	**Test Execution Date: 16/04/2024**
Description: Verify login with valid username and password	

Pre-Condition: User has valid username and password

Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)
1	Navigation to Login Page	login page successfully loaded and visible,	Index features should be displayed	Login page displayed	Pass
2	Provide Valid username	User Name: admin	Admin should be able to Login	Admin Logged in and navigated to their corresponding Dashboard	Pass
3	Provide Valid Password	admin			
4	Click on Login button				

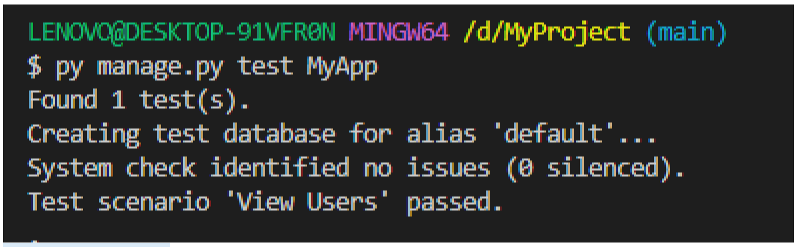
Post-Condition: Admin is validated with database and successfully login into account. The Account session details are logged in database.

Test Case 2: View Users

Code

```
from django.test import LiveServerTestCase
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.firefox.service import Service
from selenium.common.exceptions import TimeoutException
from selenium import webdriver


class Logintest(LiveServerTestCase):
    def setUp(self):
        service = Service(r'D:\mca\geckodriver.exe')
        self.driver = webdriver.Firefox(service=service)
        self.driver.implicitly_wait(10)
        self.live_server_url = "http://127.0.0.1:8000/login"  # Updated URL
def test_correct_credentials(self):
    self.driver.get(self.live_server_url)
    self.fill_form(username="admin", password="admin")
    self.driver.find_element(By.ID, "testid").click()
    try:
      WebDriverWait(self.driver, 10).until(EC.alert_is_present())
      alert = self.driver.switch_to.alert
      alert_text = alert.text
      alert.dismiss()  # Dismiss the alert
      self.fail(f"Login attempt failed with alert: {alert_text}")
  except TimeoutException:
      pass
  self.assertIn("users_list/", self.driver.current_url)
  print("Test scenario 'View Users' passed.")
```

Screenshot:



```
LENOVO@DESKTOP-91VFR0N MINGW64 /d/MyProject (main)
$ py manage.py test MyApp
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test scenario 'View Users' passed.
```

Test report:

Test Case 2

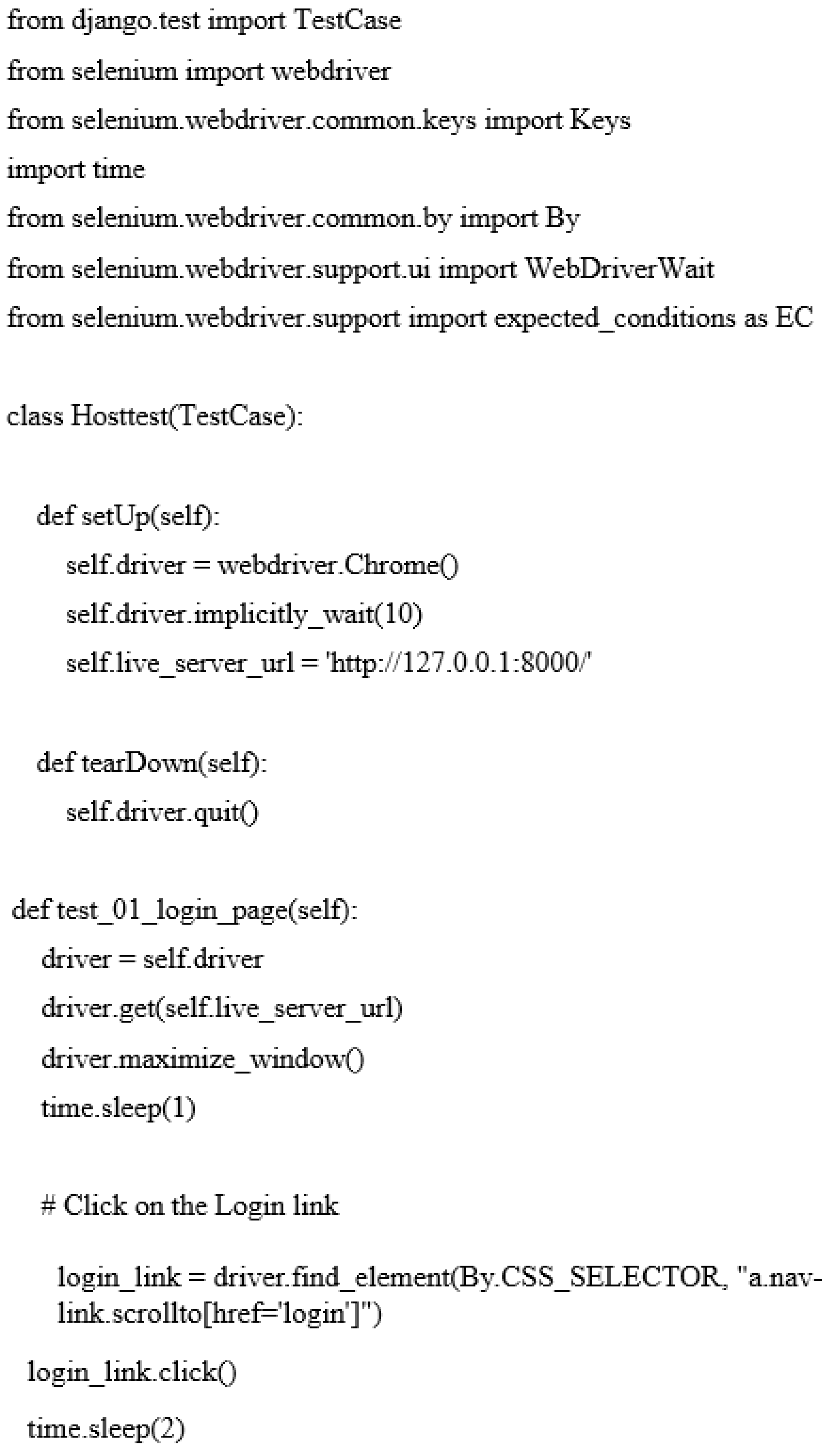
Project Name: ECHO	
View users Test Case	
Test Case ID: 2	**Test Designed By: Sudeesh E S**
Test Priority (Low/Medium/High): High	**Test Designed Date: 16/04/2024**
Module Name: User	**Test Executed By: Ms. Nimmy Francis**
Test Title: View Users	**Test Execution Date: 16/04/2023**
Description: Admin listing all users in the system	
Pre-Condition: User has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Index features should be displayed	Login page displayed	Pass
2	Provide Valid username	User Name: admin	user should be able to Login	user is Logged in and is navigated to their corresponding Dashboard	Pass
3	Provide Valid Password	admin			
4	Click on Login button		Successfull navigation to the admin dashboard	Successfull navigation to the admin dashboard	pass
5	Click on view users button	librarian	Show all users	users are displayed	Pass

Post-Condition: User is successfully logged into the system and user view is implemented.

Test Case 3: Citizen dashboard

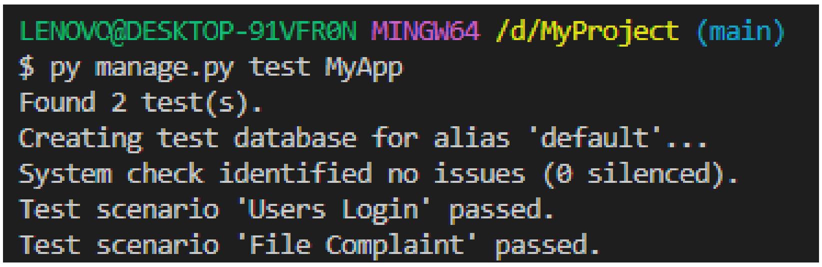
Code



```
from django.test import TestCase
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

class Hosttest(TestCase):

    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(10)
        self.live_server_url = 'http://127.0.0.1:8000/'

    def tearDown(self):
        self.driver.quit()

 def test_01_login_page(self):
     driver = self.driver
     driver.get(self.live_server_url)
     driver.maximize_window()
     time.sleep(1)

     # Click on the Login link
       login_link = driver.find_element(By.CSS_SELECTOR, "a.nav-
       link.scrollto[href='login']")
   login_link.click()
   time.sleep(2)
```

```
# Enter email and password
    email =
    driver.find_element
    (By.CSS_SELECTOR, "input[type='text'][name='username']")
    email.send_keys
    ("CI_Erumeli")  # Updated email

    password =
    driver.find_element
    (By.CSS_SELECTOR,"input[type='password'][name='password'])
    password.send_keys
    ("Sushisan@234")  # Updated password
    time.sleep(2)

# Click on the Sign In button
submit_button =
driver.find_element
(By.CSS_SELECTOR, "button#testid")
submit_button.click()
time.sleep(2)
```

Screenshot:



```
LENOVO@DESKTOP-91VFR0N MINGW64 /d/MyProject (main)
$ py manage.py test MyApp
Found 2 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test scenario 'Users Login' passed.
Test scenario 'File Complaint' passed.
```

Test Case 3

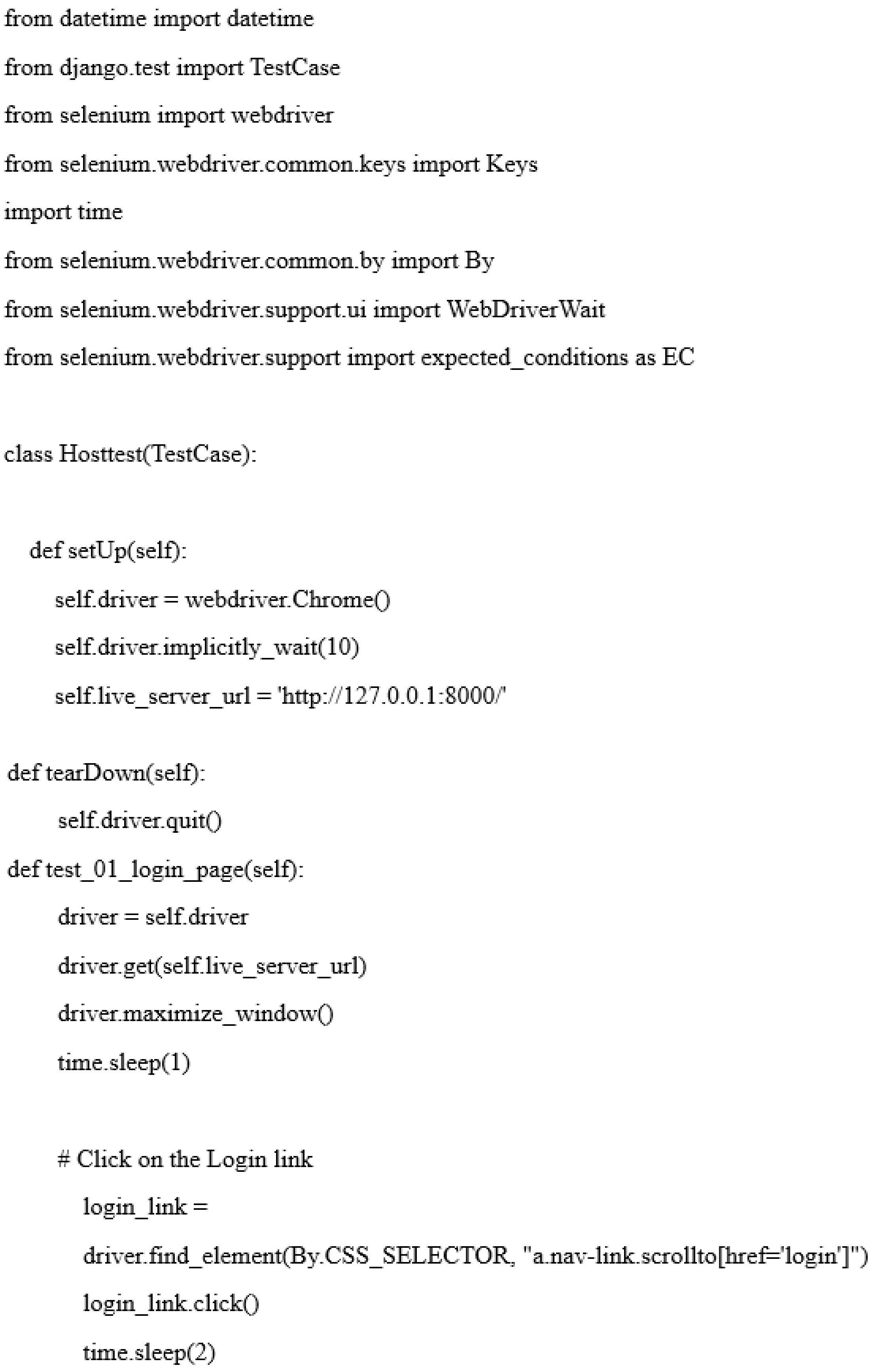
Project Name: ECHO	
Citizen dashboard Test Case	
Test Case ID: 3	**Test Designed By: Sudeesh E S**
Test Priority(Low/Medium/High):High	**Test Designed Date: 16/04/2023**
Module Name: Citizen	**Test Executed By: Ms. Nimmy Francis**
Test Title: Citizen	**Test Execution Date: 16/04/2023**
Description: User functionalities.	
Pre-Condition: User has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Index features should be displayed	Login page displayed	Pass
2	Provide Valid Jobseeker email address	User Name: sudeesh43	User should be able to Login	User Logged in and navigated to their respective Dashboard	Pass
3	Provide Valid Password	Sushi@123			
4	Click on Login button				
5	User is Navigated to dashboard		User should be able to view dashboard	User should be able to view his dashboard	Pass
6	User click File Complaints Button		User should be able file a complaint	Complaint form displayed	Pass
7	Display Complaints list		File complaint	Successfully filed complaint	Pass

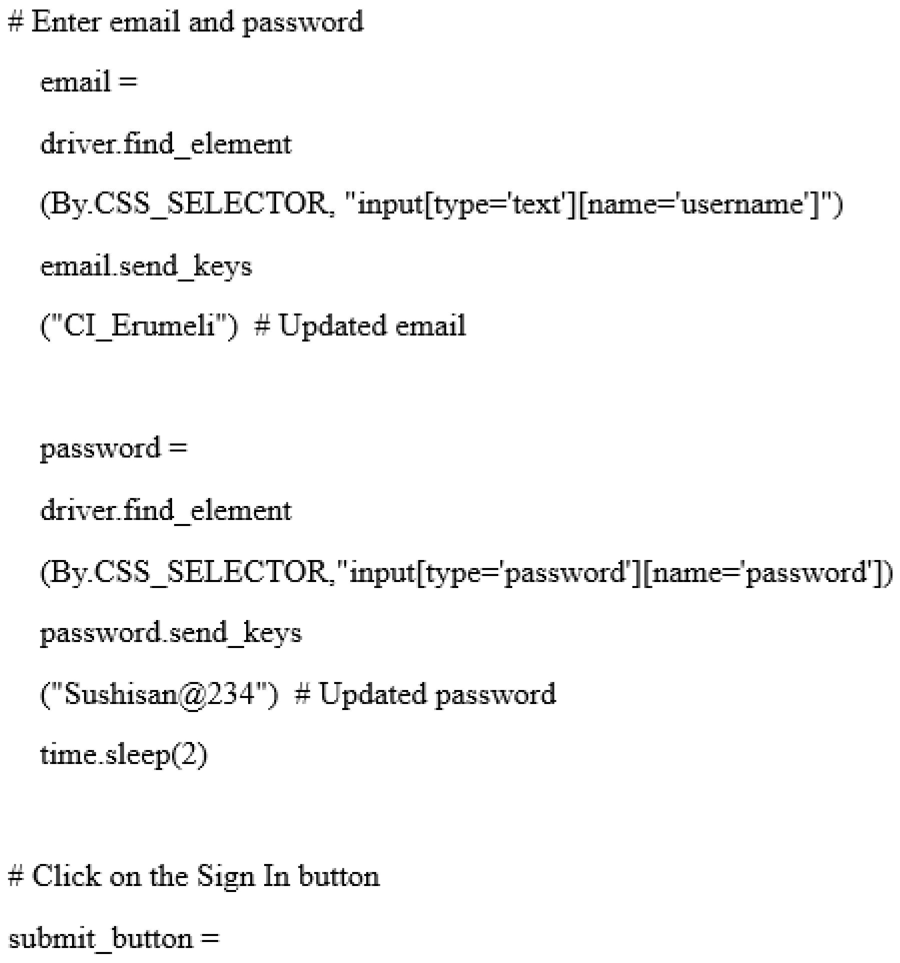
Post-Condition: User successfully logged into their account and is able to file Complaints.

Test Case 4: Missing person List

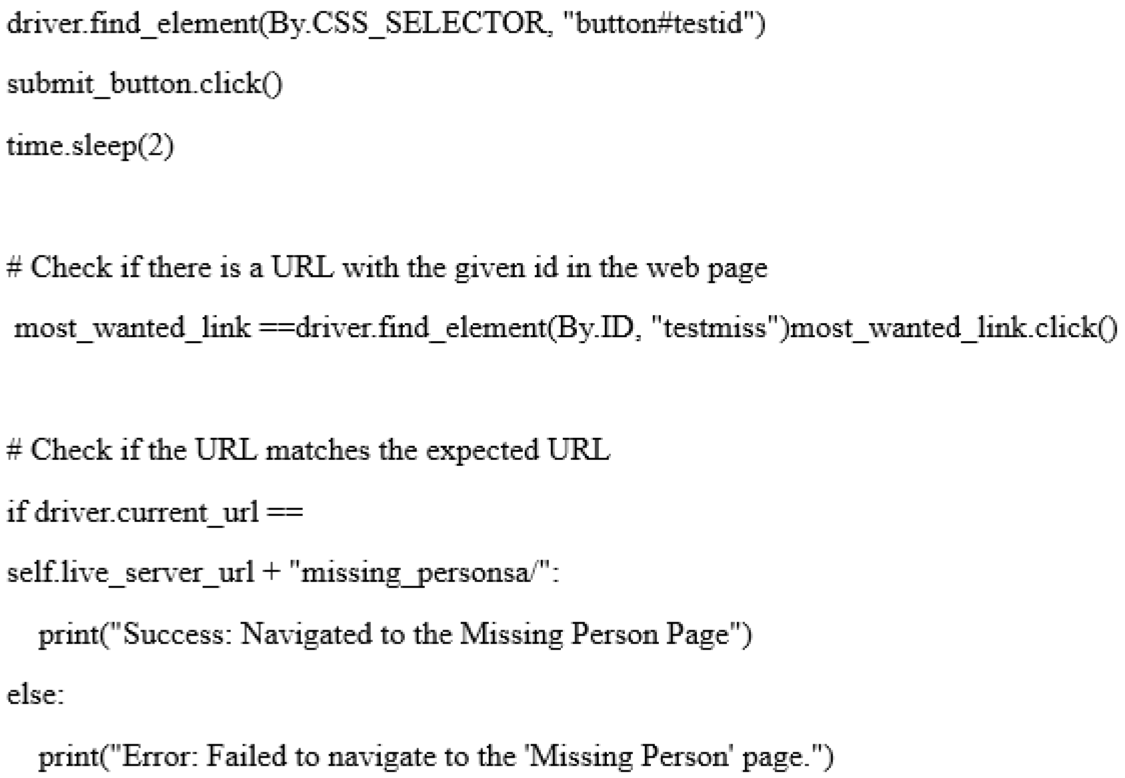
Code



```
from datetime import datetime
from django.test import TestCase
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

class Hosttest(TestCase):

    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(10)
        self.live_server_url = 'http://127.0.0.1:8000/'
 def tearDown(self):
        self.driver.quit()
 def test_01_login_page(self):
        driver = self.driver
        driver.get(self.live_server_url)
        driver.maximize_window()
        time.sleep(1)

        # Click on the Login link
            login_link =
            driver.find_element(By.CSS_SELECTOR, "a.nav-link.scrollto[href='login']")
            login_link.click()
            time.sleep(2)
```

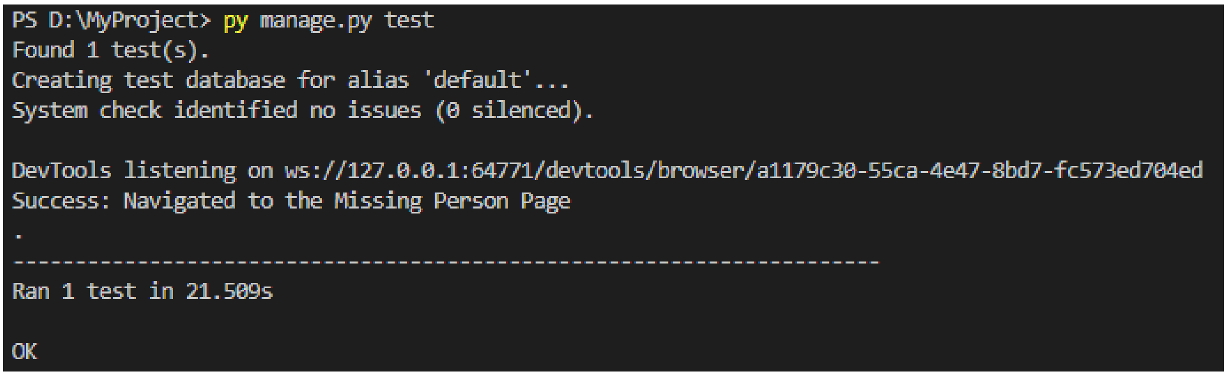


```
# Enter email and password
    email =
    driver.find_element
    (By.CSS_SELECTOR, "input[type='text'][name='username']")
    email.send_keys
    ("CI_Erumeli")  # Updated email


    password =
    driver.find_element
    (By.CSS_SELECTOR,"input[type='password'][name='password'])
    password.send_keys
    ("Sushisan@234")  # Updated password
    time.sleep(2)

# Click on the Sign In button
submit_button =
```



```
 driver.find_element(By.CSS_SELECTOR, "button#testid")
 submit_button.click()
 time.sleep(2)

 # Check if there is a URL with the given id in the web page
  most_wanted_link ==driver.find_element(By.ID, "testmiss")most_wanted_link.click()

 # Check if the URL matches the expected URL
 if driver.current_url ==
 self.live_server_url + "missing_personsa/":
     print("Success: Navigated to the Missing Person Page")
 else:
     print("Error: Failed to navigate to the 'Missing Person' page.")
```

Screenshot:



```
PS D:\MyProject> py manage.py test
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:64771/devtools/browser/a1179c30-55ca-4e47-8bd7-fc573ed704ed
Success: Navigated to the Missing Person Page
.
----------------------------------------------------------------------
Ran 1 test in 21.509s

OK
```

Test report:

Test Case 4

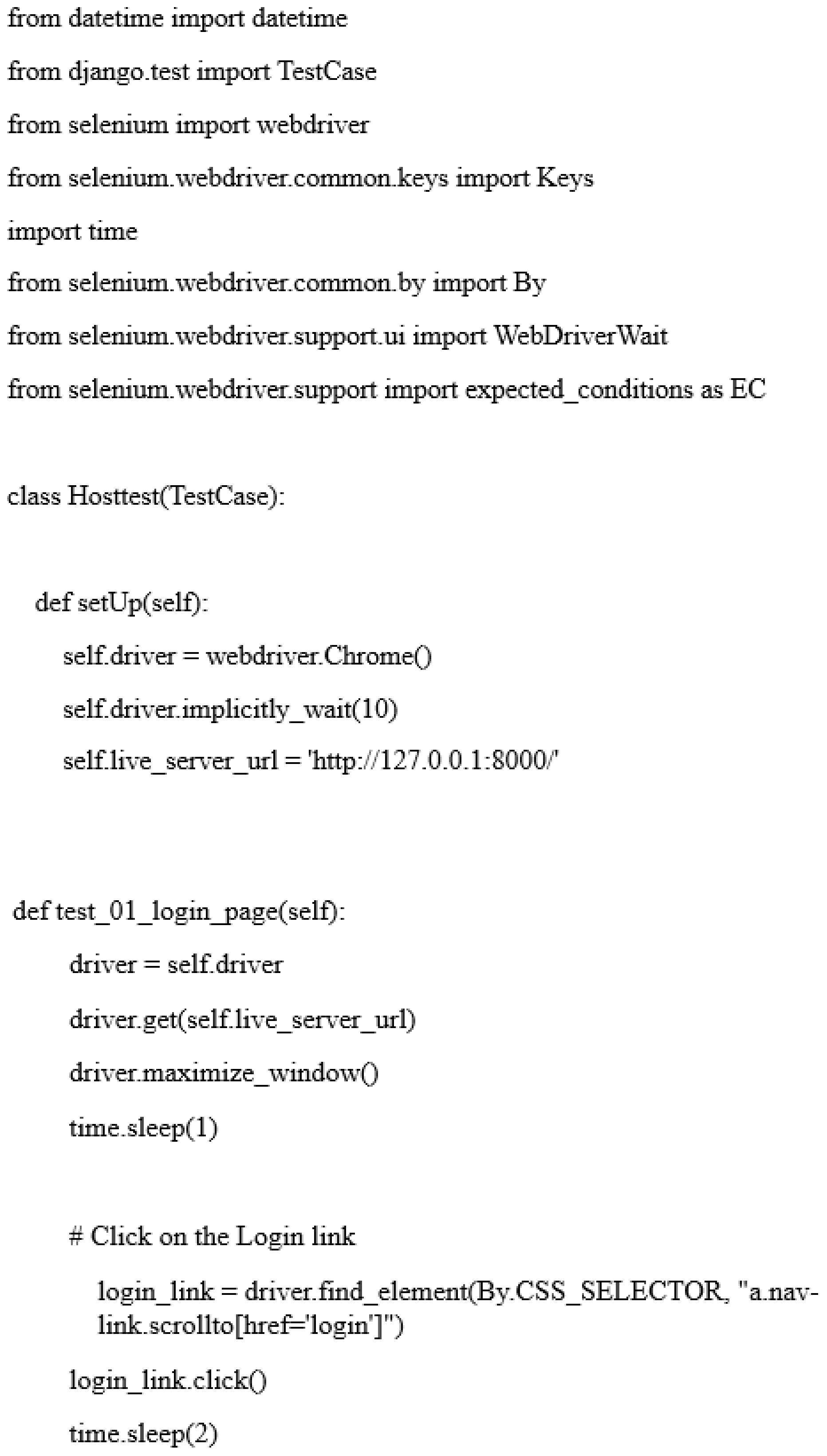
Project Name: ECHO	
Missing Person Test Case	
Test Case ID: 4	**Test Designed By: Sudeesh E S**
Test Priority (Low/Medium/High): High	**Test Designed Date: 16/04/2024**
Module Name: Missing Person	**Test Executed By: Ms. Nimmy Francis**
Test Title: Missing person	**Test Execution Date: 16/04/2024**
Description: View missing person list	
Pre-Condition: User has valid username and password	

Step	Test Step	Test Data	Expect Result	Actual Result	Status (Pass/ Fail)
1	Navigation to Login Page		Index features should be displayed	Login page displayed	Pass
2	Provide Valid user name	User Name: CI_Erumeli	User should be able to Login	User Logged in and navigated to their corresponding Dashboard	Pass
3	Provide Valid Password	sushisan@234			
4	Click on Login button				
5	Click on the missing person list button		User should be able to see the missing persons list	Login success	Pass

Post-Condition: User is logged into the system and successfully submitted the feedback form.

Test Case 5: Most Wanted view

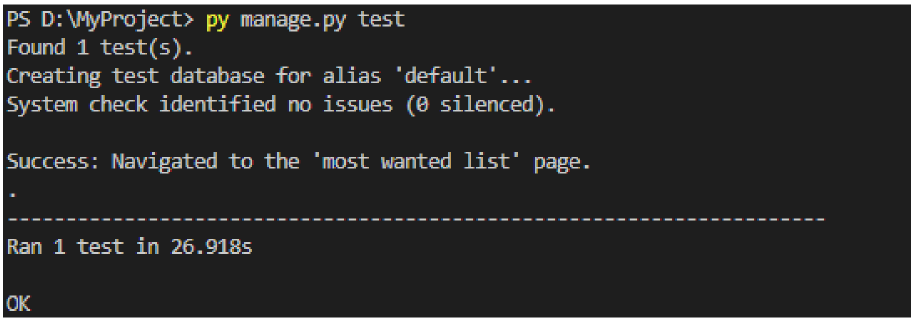
Code



```
from datetime import datetime

from django.test import TestCase

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

import time

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected_conditions as EC


class Hosttest(TestCase):


    def setUp(self):

        self.driver = webdriver.Chrome()

        self.driver.implicitly_wait(10)

        self.live_server_url = 'http://127.0.0.1:8000/'



 def test_01_login_page(self):

         driver = self.driver

         driver.get(self.live_server_url)

         driver.maximize_window()

         time.sleep(1)


         # Click on the Login link

             login_link = driver.find_element(By.CSS_SELECTOR, "a.nav-
             link.scrollto[href='login']")

         login_link.click()

         time.sleep(2)
```

```
# Enter email and password
   email =
   driver.find_element
   (By.CSS_SELECTOR, "input[type='text'][name='username']")
   email.send_keys
   ("CI_Erumeli")  # Updated email

   password =
   driver.find_element
   (By.CSS_SELECTOR,"input[type='password'][name='password'])
   password.send_keys
   ("Sushisan@234")  # Updated password
   time.sleep(2)

 # Check if there is a URL with the given id in the web page
  most_wanted_link = driver.find_element(By.ID, "testmost")most_wanted_link.click()

 # Check if the URL matches the expected URL

 if driver.current_url == self.live_server_url + "missing_personsa/":
    print("Success: Navigated to the Most Wanted Page")
 else:
    print("Error: Failed to navigate to the Most Wanted page.")
```

Screenshot:



```
PS D:\MyProject> py manage.py test
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).

Success: Navigated to the 'most wanted list' page.
.
----------------------------------------------------------------------
Ran 1 test in 26.918s

OK
```

Test report:

Test Case 5

Project Name: ECHO					
Most wanted Test Case					
Test Case ID: 5			**Test Designed By: Sudeesh E S**		
Test Priority (Low/Medium/High): High			**Test Designed Date: 16/04/2024**		
Module Name: Most wanted			**Test Executed By: Ms. Nimmy Francis**		
Test Title: Most wanted List			**Test Execution Date: 16/04/2024**		
Description: View most wanted list					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expect Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page	Login Page should be visible	Index features should be displayed	Login page displayed	Pass
2	Provide Valid user name	User Name: CI_Erumeli	User should be able to Login	User Logged in and navigated to their corresponding Dashboard	Pass
3	Provide Valid Password	sushisan@234			
4	Click on Login button				
5	After reaching dashboard go to most wanted button		Most wanted people list	View success	Pass
Post-Condition: User is logged into the system and successfully viewed the most wanted list.					

CHAPTER 6

IMPLEMENTATION

6.1INTRODUCTION

Project implementation is the phase where plans and strategies transform into tangible actions and outcomes. It marks the transition from theoretical concepts to practical application. This pivotal stage requires meticulous planning, resource allocation, and a dedicated team to execute tasks according to the established timeline and objectives. In this phase, the project team translates the project's blueprints into real-world activities, ensuring that each step aligns with the overarching goals. Effective project implementation demands clear communication, robust leadership, and a keen eye for detail. This introduction sets the stage for a comprehensive understanding of the project implementation process, emphasizing its significance in achieving the envisioned goals. As we delve deeper, we will explore key components, strategies, and best practices that contribute to successful project implementation.

6.2 IMPLEMENTATION PROCEDURES

6.2.1 User Training

User training is a critical component of ensuring the effective utilization of any website. It involves imparting the necessary knowledge and skills to end-users, enabling them to navigate and utilize the website efficiently. This training equips users with a comprehensive understanding of the website's features, functions, and capabilities. Through hands-on sessions and guided tutorials, users learn how to perform tasks, customize settings, and troubleshoot common issues. Moreover, user training fosters confidence and proficiency, empowering individuals to maximize their productivity while using the application.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness, it is essential to provide training on the new application software to the user. This training should include the underlying philosophy of using the new system, such as the flow of screens, screen design, the type of help available on the screen, the types of errors that may occur while entering data, and the corresponding validation checks for each entry, and ways to correct the data entered. Additionally, the training should cover information specific to the user or group, which is necessary to use the system or part of the system effectively. It is important to note that this training may differ across different user groups and levels of hierarchy.

6.2.3 System Maintenance

System maintenance is a crucial aspect of ensuring the seamless operation and longevity of any website application. It encompasses a series of tasks aimed at monitoring, optimizing, and troubleshooting the underlying infrastructure on which the application runs. This includes activities such as regular performance monitoring, and data backups. Additionally, system maintenance involves identifying and rectifying any potential vulnerabilities or inefficiencies that may impede the website’s performance. Proactive maintenance measures contribute to a stable and secure environment, minimizing the risk of unexpected downtime or data loss.

6.2.4 Hosting

Hosting a website involves making a website accessible to users on the internet by storing its files on a server and configuring domain settings to point to that server. When hosting a website, the most typical choice is to find a hosting provider, set up server space, upload website files, configure domain settings, and ensure security measures are in place..

Amazon Elastic Compute Cloud (EC2) Amazon Elastic Compute Cloud (EC2) stands as a pivotal component within Amazon Web Services (AWS), allowing users to lease virtual servers, or instances, within the cloud. This service permits adaptable scaling of computing resources based on demand, offering a spectrum of instance types optimized for various workloads. EC2 provides comprehensive control over server instances, supports multiple operating systems, and furnishes features promoting security, scalability, and cost-efficiency. Users can conveniently manage their instances via the AWS Management Console or APIs. Ultimately, AWS EC2 empowers businesses to operate applications and services reliably, scalability, and economically in the cloud.

Procedure for hosting a website on AWS:

Step 1: Create an AWS Account: Start by creating an AWS account, if you haven't already. Open the EC2 dashboard by browsing to the AWS Management Console.

Step 2: Build an EC2 Instance: Initiate a new EC2 instance by selecting the "Launch Instance" button. Select the Ubuntu AMI (Amazon Machine Image) and choose a suitable instance type aligned with your project's requirements.

Step 3: Configure Instance Settings: Tailor instance parameters such as quantity, networking configurations, and storage preferences. Integrate storage volumes for secure storage of project files and data.

Step 4: Set Up Security Group: Establish a new security group or utilize an existing one to define firewall rules. Ensure accessibility to ports 22 (SSH) for remote access and 8000 (or any required port for your application) for web traffic.

Step 5: Launch Instance: Review instance setup and initiate the EC2 instance launch process. Choose or create a pair of keys to gain access to SSH.

Step 6: Connect to the Instance: Upon instance launch, establish an SSH connection using the public IP address or DNS name of your instance along with the key pair for secure access.

Step 7: Clone Project Repository: Install Git on the EC2 instance and clone the project repository from your Git repository using the git clone command.

Step 8: Install Python and Django: Deploy Python and Django on the EC2 instance using the apt package manager for Python installation and pip for installing Django along with any other necessary Python packages.

Step 9: Install Dependencies: Install essential packages and dependencies for running the website, including database drivers and Django extensions.

Step 10: Configure Django Settings: Update the Django settings file (settings.py) with the requisite database configuration, static file settings, and other project-specific configurations.

Step 11: Run Django Server: To start the Django development server, type python3 manage.py runserver 0.0.0.0:8000 into your script, operating the server on port 8000 accessible on all network interfaces.

Step 12: Test Your Website: Access a web browser and navigate to the public IP address or DNS name of your EC2 instance followed by the designated port number (:8000). Confirm website accessibility and functionality.

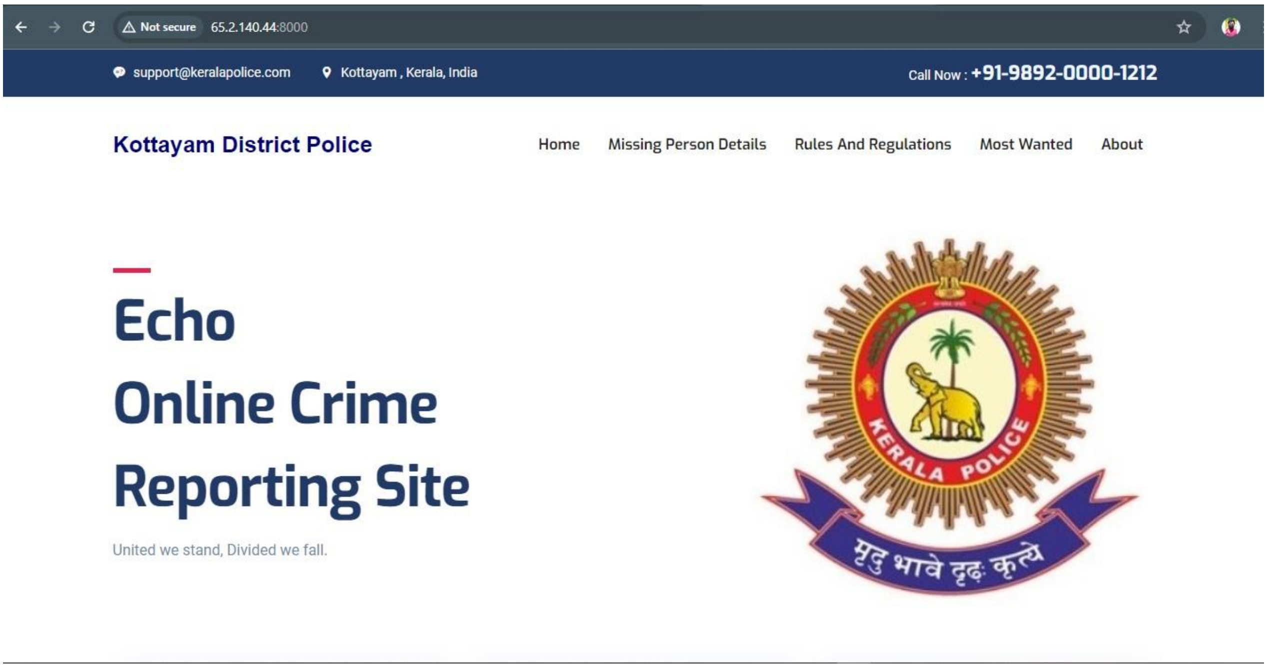
Step13: Domain Name Configuration (Optional): If you possess a domain name, configure DNS settings to point to the public IP address of your EC2 instance.

Step14: SSL/TLS Certificate Setup (Optional): Enable HTTPS for your website by setting up an SSL/TLS certificate using AWS Certificate Manager or a third-party provider.

Step15: Monitor Your EC2 Instance: Utilize AWS CloudWatch or other monitoring utilities to oversee EC2 instance performance, security, and uptime. Regularly update instance and software for enhanced security and reliability.

Hosted Website:

Home Page:



Not secure 65.2.140.44:8000

support@keralapolice.com Kottayam , Kerala, India Call Now : +91-9892-0000-1212

Kottayam District Police

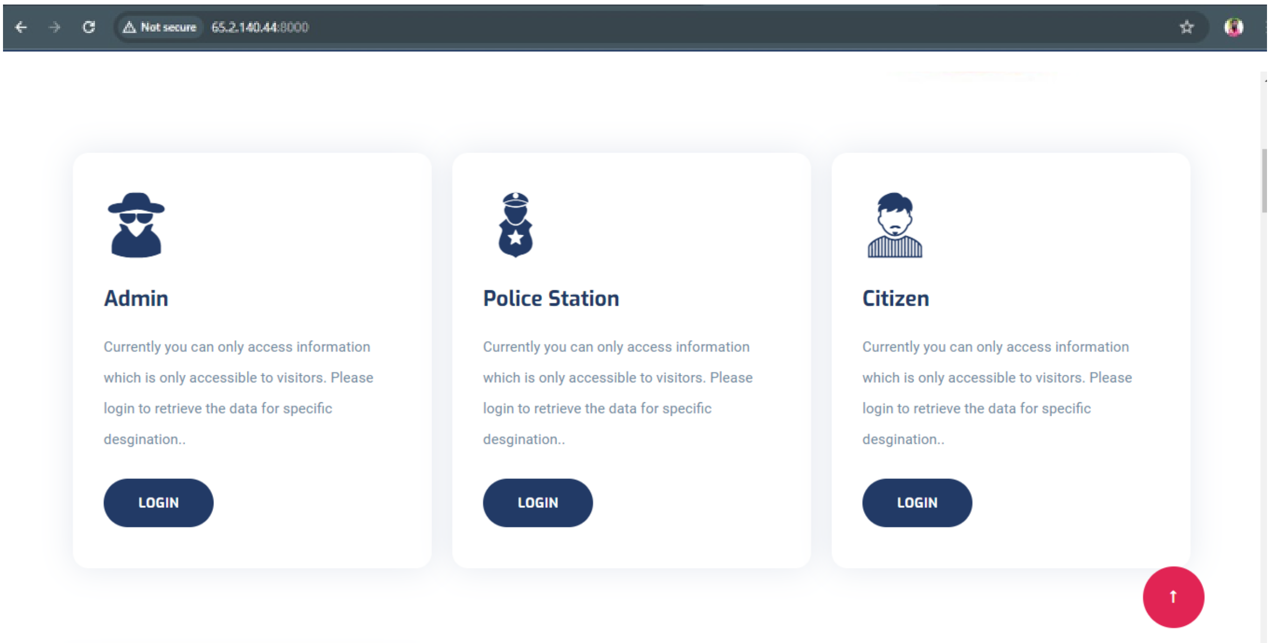
Echo

Online Crime

Reporting Site

United we stand, Divided we fall.

Login:



Not secure 65.2.140.44:8000

Admin

Currently you can only access information which is only accessible to visitors. Please login to retrieve the data for specific desgination..

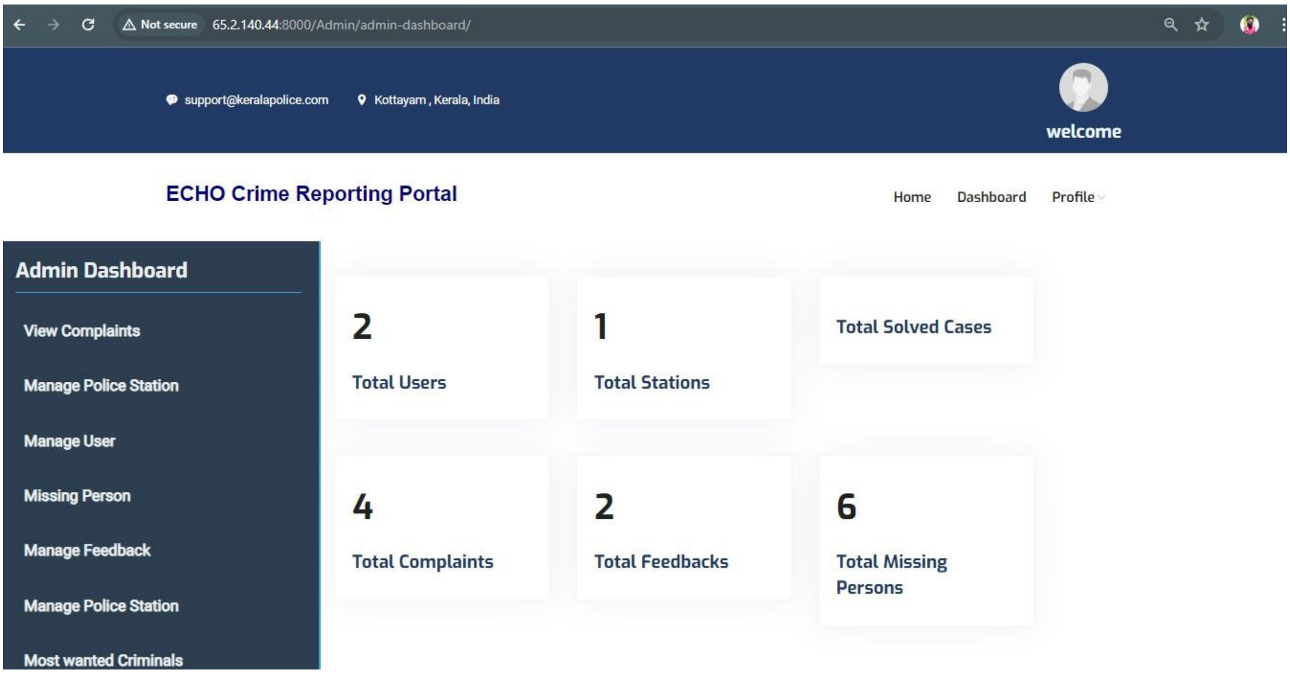
Police Station

Currently you can only access information which is only accessible to visitors. Please login to retrieve the data for specific desgination..

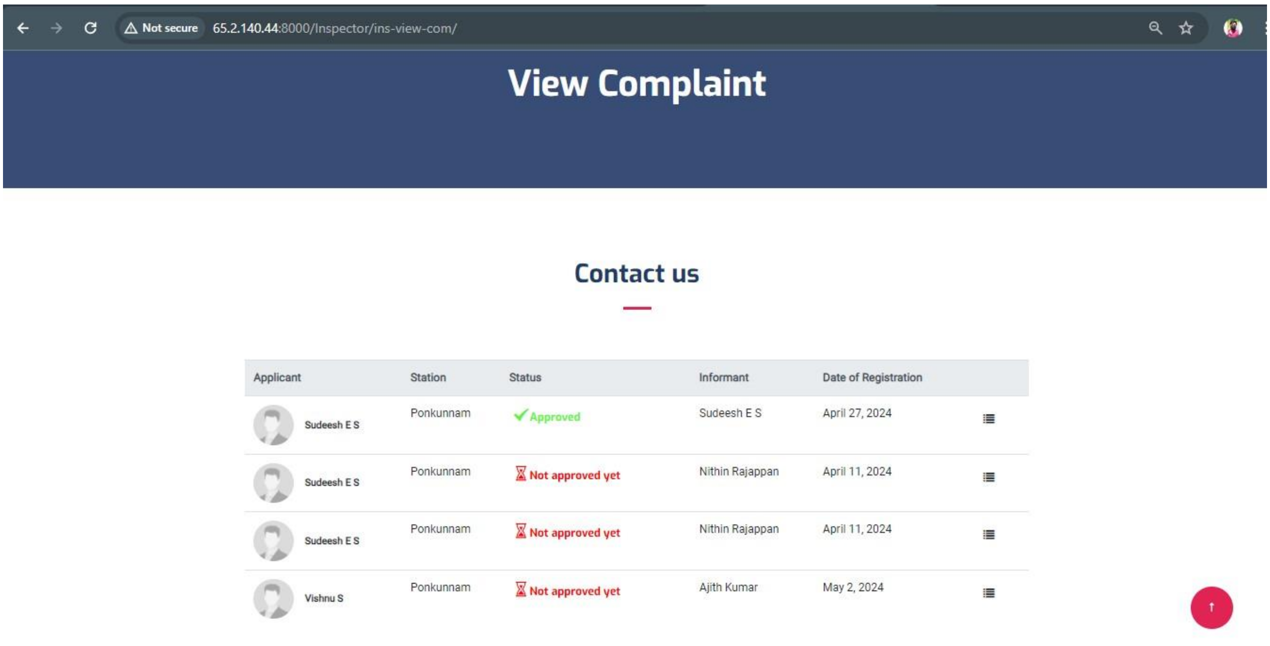
Citizen

Currently you can only access information which is only accessible to visitors. Please login to retrieve the data for specific desgination..

Admin dashboard:



View Complaints:



Applicant	Station	Status	Informant	Date of Registration	
Sudeesh E S	Ponkunnam	✓ Approved	Sudeesh E S	April 27, 2024	☰
Sudeesh E S	Ponkunnam	⌛ Not approved yet	Nithin Rajappan	April 11, 2024	☰
Sudeesh E S	Ponkunnam	⌛ Not approved yet	Nithin Rajappan	April 11, 2024	☰
Vishnu S	Ponkunnam	⌛ Not approved yet	Ajith Kumar	May 2, 2024	☰

Project link: http://65.2.140.44:8000/

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In conclusion, the "Online Crime Reporting System" named ECHO stands as a groundbreaking initiative in enhancing public safety and law enforcement collaboration within Kottayam district. Leveraging the powerful Django framework, ECHO provides a user-friendly web-based platform that empowers individuals to report criminal activities conveniently and efficiently. The project not only bridges the gap between the general public and law enforcement but also establishes a seamless process for reporting and addressing crimes. The system's key features, including user registration, incident reporting, and administrative interfaces for both users and admins, contribute to a comprehensive solution. By utilizing HTML and CSS for the front-end and Python Django for the back-end, ECHO ensures a secure, scalable, and intuitive experience for users of all technical backgrounds. ECHO aspires not only to meet but to exceed industry standards, contributing to a paradigm shift in how crimes are reported, managed, and addressed in the ever-evolving landscape of public safety.

7.2 FUTURE SCOPE

Anticipating the future, ECHO envisions a path of continuous innovation and evolution to navigate the dynamic landscape of online crime reporting. Specifically, ECHO aims to introduce advanced features for enhancing the analysis of incident reports, automating the categorization of criminal activities, and providing law enforcement with more efficient tools for prioritization and management. Moreover, ECHO is poised to explore additional dimensions of multimedia handling. Enhancing the platform's capabilities to process and analyze multimedia evidence, such as photos and videos submitted by users, will contribute to a more comprehensive and efficient crime reporting system.

In alignment with a user-centric approach, ECHO remains committed to soliciting and incorporating user feedback. This iterative process ensures that the platform not only meets but exceeds the expectations of both the public and law enforcement. By staying abreast of technological advancements and responding proactively to user needs, ECHO positions itself as a frontrunner in redefining the landscape of online crime reporting, making it an indispensable tool for fostering community safety and collaboration.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Design and Implementation of Online Crime Reporting System, Joy I. Eze, Elijah O. Omidiora, Published in: International Journal of Computer Applications 2014.

- Crime Reporting and Criminal Identification Using Online Reporting System, Priya H. Patil, Mr. Aniket S. Potpote, International Jounal of Advanced Research in Computer Engineering & Technology 2014.

- Secure and Efficient Online Crime Reporting System for Community Policing, Mohammed A. Aibinu, Mustafa Man, Rania A. Kora, Ali Selamat, Journal of Network and Computer Applications 2016.

WEBSITES:

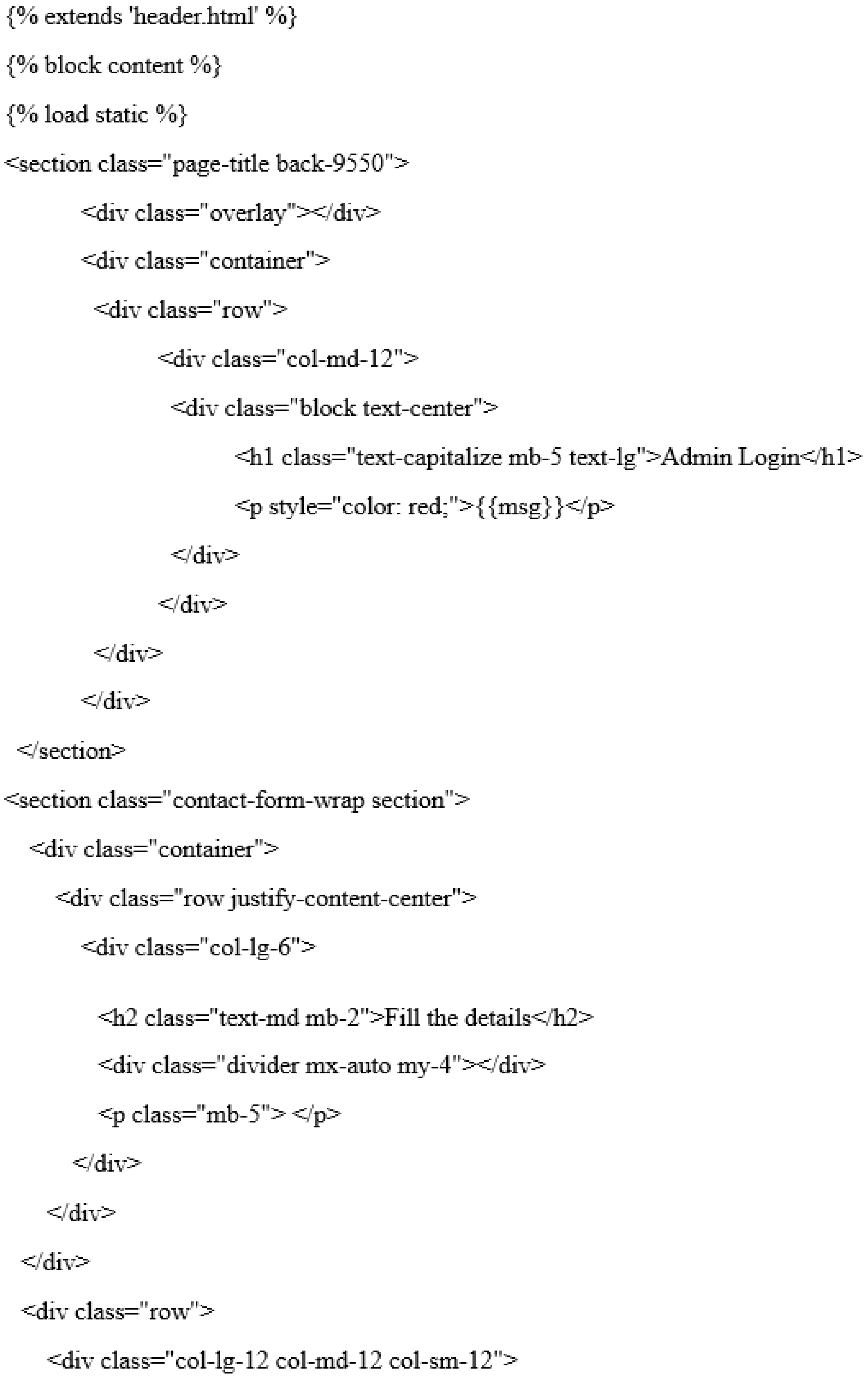
- www.chat.openai.com
- www.djangoproject.com
- www.github.com
- www.stackoverflow.com
- www.w3schools.com

CHAPTER 9

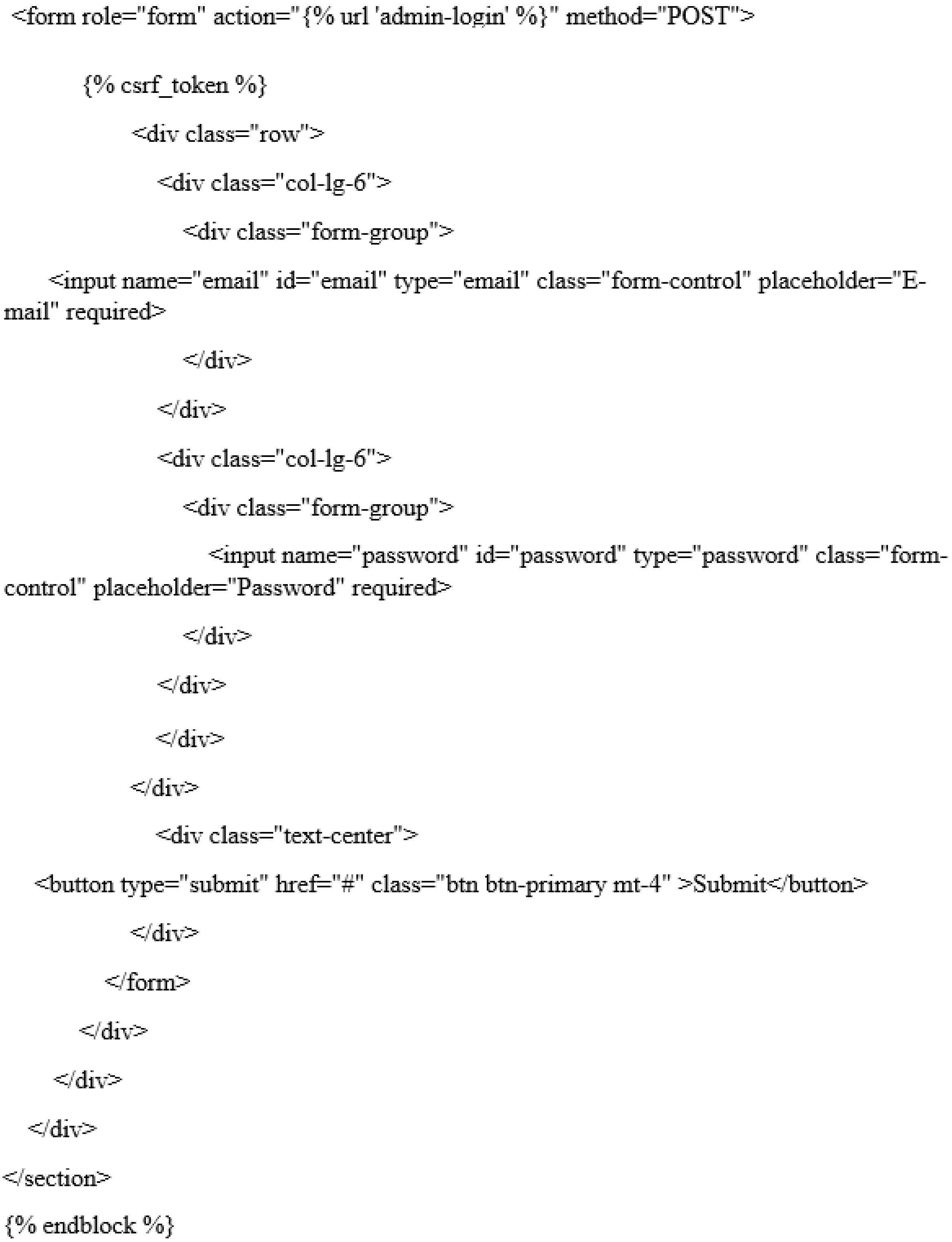
APPENDIX

9.1 Sample Code

Login

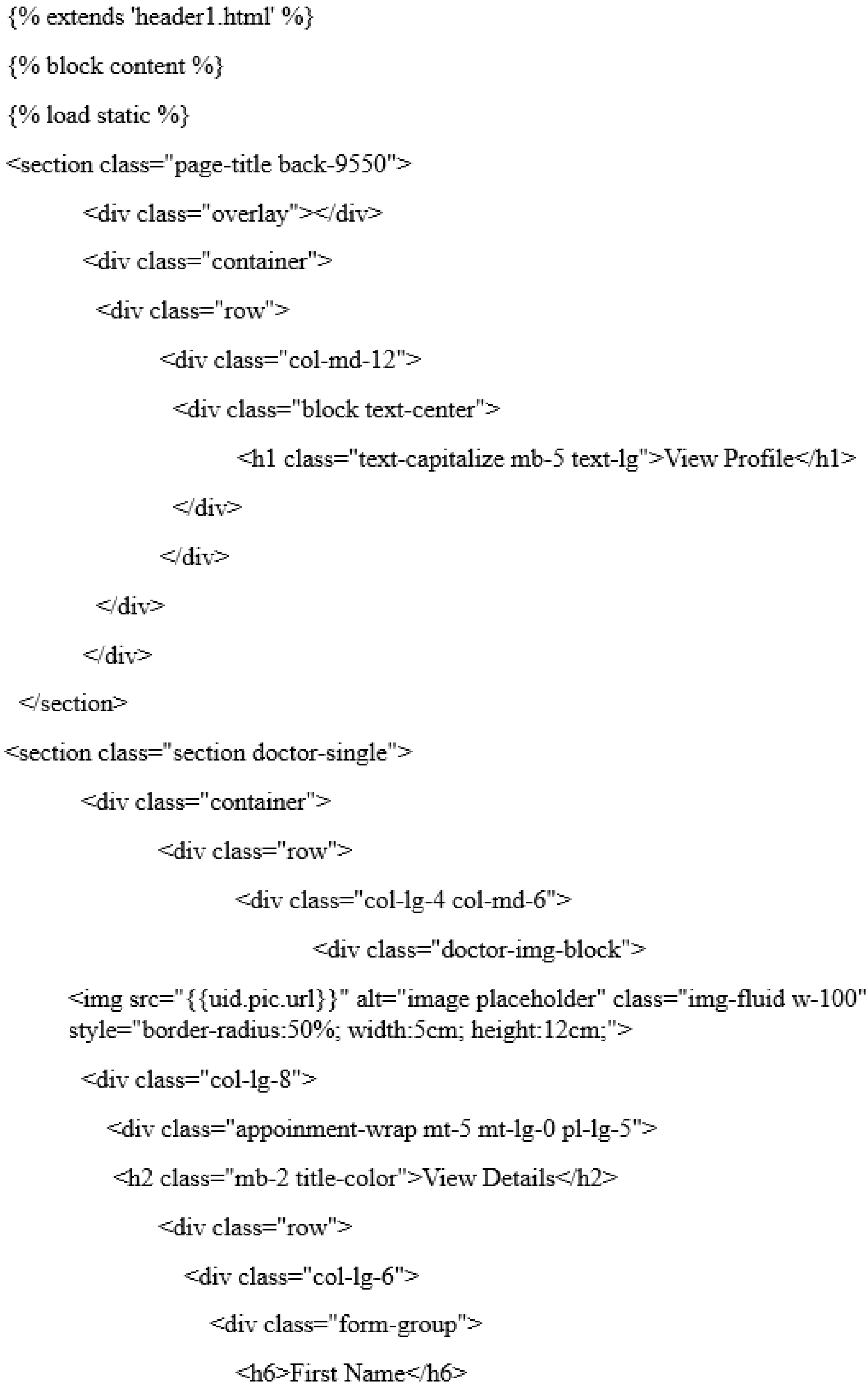


```
{% extends 'header.html' %}
{% block content %}
{% load static %}
<section class="page-title back-9550">
            <div class="overlay"></div>
            <div class="container">
              <div class="row">
                        <div class="col-md-12">
                          <div class="block text-center">
                                    <h1 class="text-capitalize mb-5 text-lg">Admin Login</h1>
                                    <p style="color: red;">{{msg}}</p>
                          </div>
                        </div>
              </div>
            </div>
  </section>
<section class="contact-form-wrap section">
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-lg-6">
              <h2 class="text-md mb-2">Fill the details</h2>
              <div class="divider mx-auto my-4"></div>
              <p class="mb-5"> </p>
           </div>
       </div>
   </div>
   <div class="row">
       <div class="col-lg-12 col-md-12 col-sm-12">
```

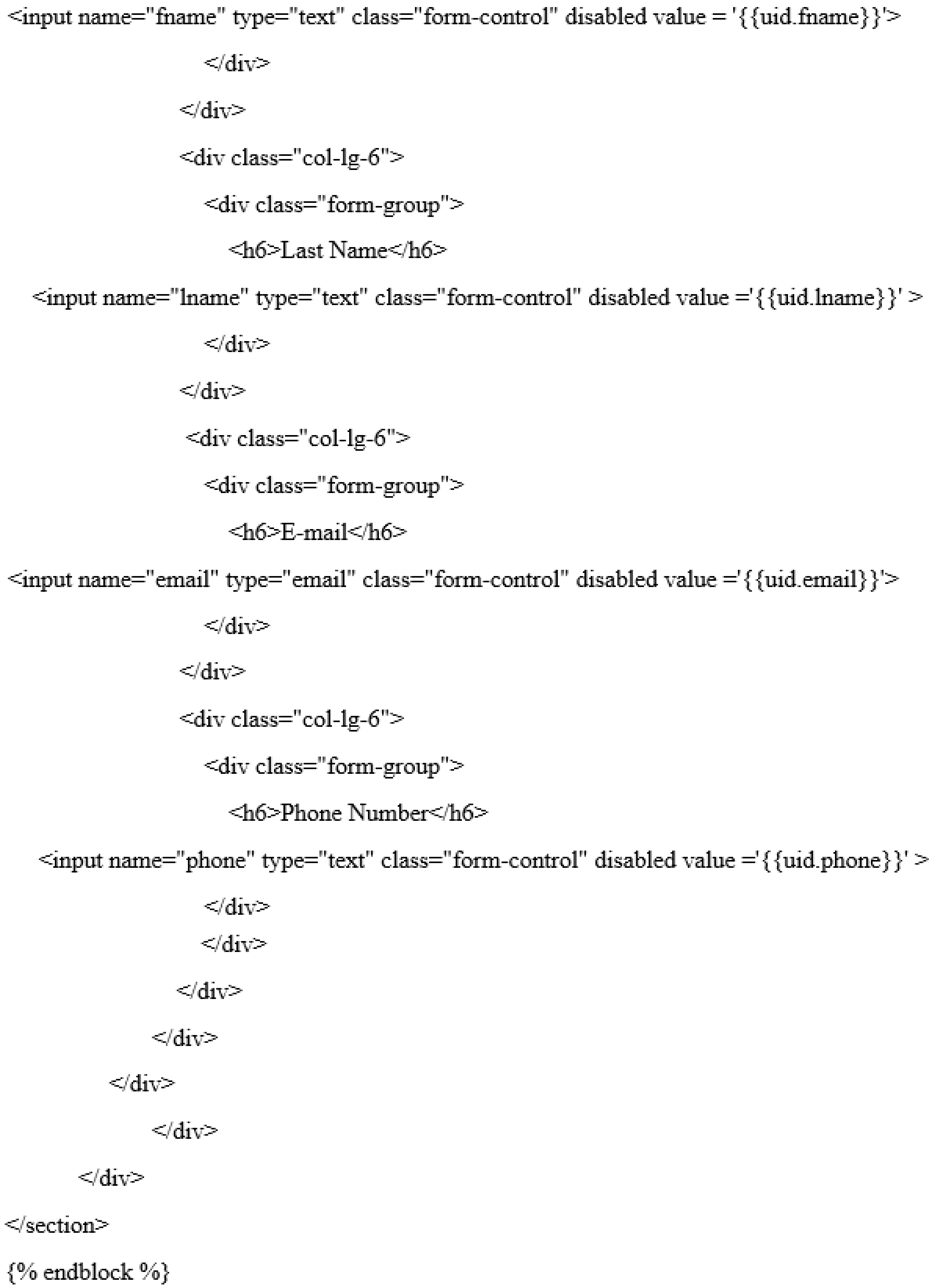


```
 <form role="form" action="{% url 'admin-login' %}" method="POST">

            {% csrf_token %}
                    <div class="row">
                        <div class="col-lg-6">
                            <div class="form-group">
       <input name="email" id="email" type="email" class="form-control" placeholder="E-
mail" required>
                            </div>
                        </div>
                        <div class="col-lg-6">
                            <div class="form-group">
                                <input name="password" id="password" type="password" class="form-
control" placeholder="Password" required>
                            </div>
                        </div>
                        </div>
                    </div>
                        <div class="text-center">
     <button type="submit" href="#" class="btn btn-primary mt-4" >Submit</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</section>
{% endblock %}
```

Registration

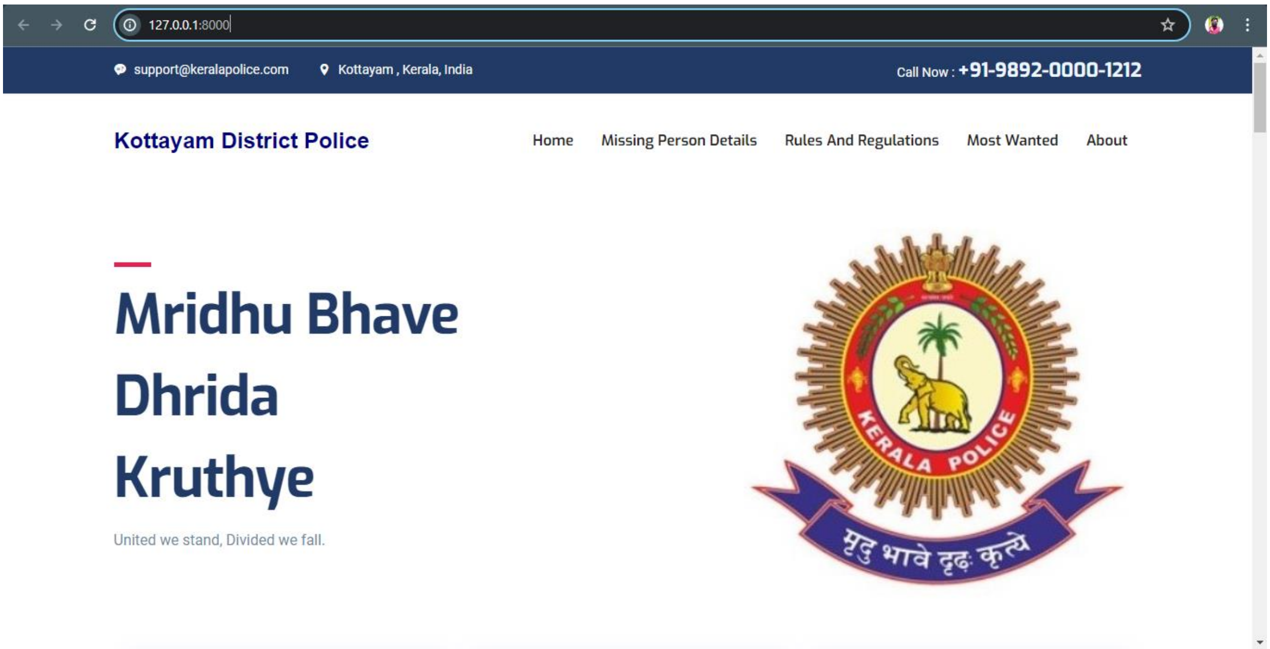


```
{% extends 'header1.html' %}

{% block content %}

{% load static %}

<section class="page-title back-9550">

        <div class="overlay"></div>

        <div class="container">

          <div class="row">

                <div class="col-md-12">

                  <div class="block text-center">

                        <h1 class="text-capitalize mb-5 text-lg">View Profile</h1>

                  </div>

                </div>

          </div>

        </div>

  </section>

<section class="section doctor-single">

        <div class="container">

                <div class="row">

                        <div class="col-lg-4 col-md-6">

                                <div class="doctor-img-block">

      <img src="{{uid.pic.url}}" alt="image placeholder" class="img-fluid w-100"
      style="border-radius:50%; width:5cm; height:12cm;">

        <div class="col-lg-8">

          <div class="appoinment-wrap mt-5 mt-lg-0 pl-lg-5">

           <h2 class="mb-2 title-color">View Details</h2>

                <div class="row">

                   <div class="col-lg-6">

                      <div class="form-group">

                        <h6>First Name</h6>
```



```
<input name="fname" type="text" class="form-control" disabled value = '{{uid.fname}}'>
                    </div>
                  </div>
                  <div class="col-lg-6">
                    <div class="form-group">
                      <h6>Last Name</h6>
  <input name="lname" type="text" class="form-control" disabled value ='{{uid.lname}}' >
                    </div>
                  </div>
                   <div class="col-lg-6">
                    <div class="form-group">
                      <h6>E-mail</h6>
<input name="email" type="email" class="form-control" disabled value ='{{uid.email}}'>
                    </div>
                  </div>
                  <div class="col-lg-6">
                    <div class="form-group">
                      <h6>Phone Number</h6>
   <input name="phone" type="text" class="form-control" disabled value ='{{uid.phone}}' >
                    </div>
                    </div>
                  </div>
                </div>
            </div>
                </div>
        </div>
</section>
{% endblock %}
```

9.2 Screen Shots



127.0.0.1:8000

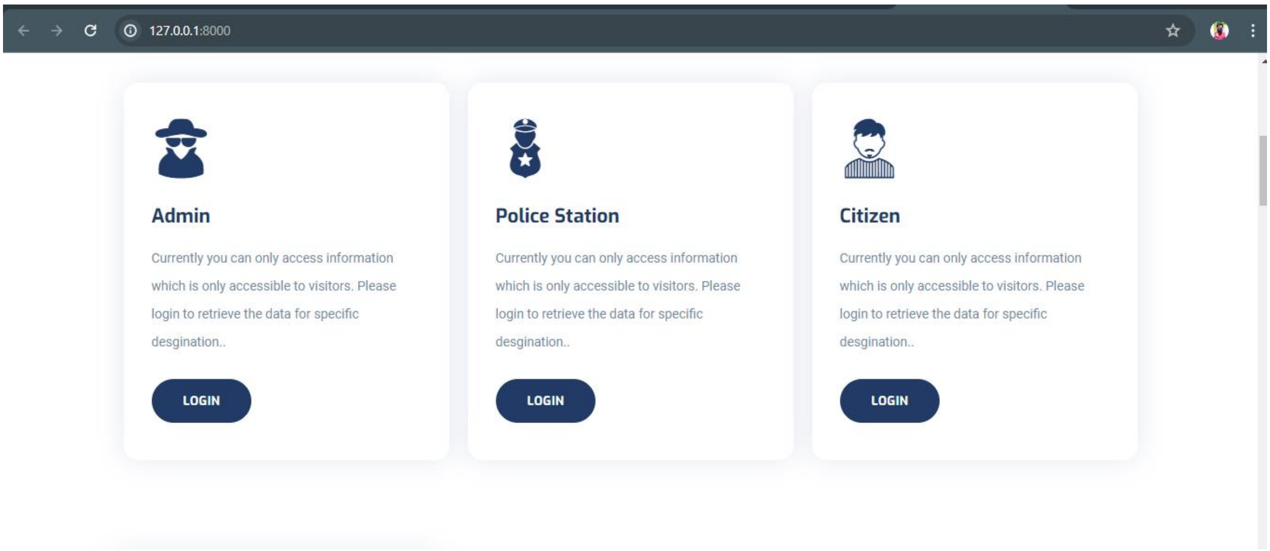
support@keralapolice.com Kottayam , Kerala, India Call Now : +91-9892-0000-1212

Kottayam District Police

Mridhu Bhave Dhrida Kruthye

United we stand, Divided we fall.

Fig 1: Home Page



127.0.0.1:8000

Admin

Currently you can only access information which is only accessible to visitors. Please login to retrieve the data for specific desgination..

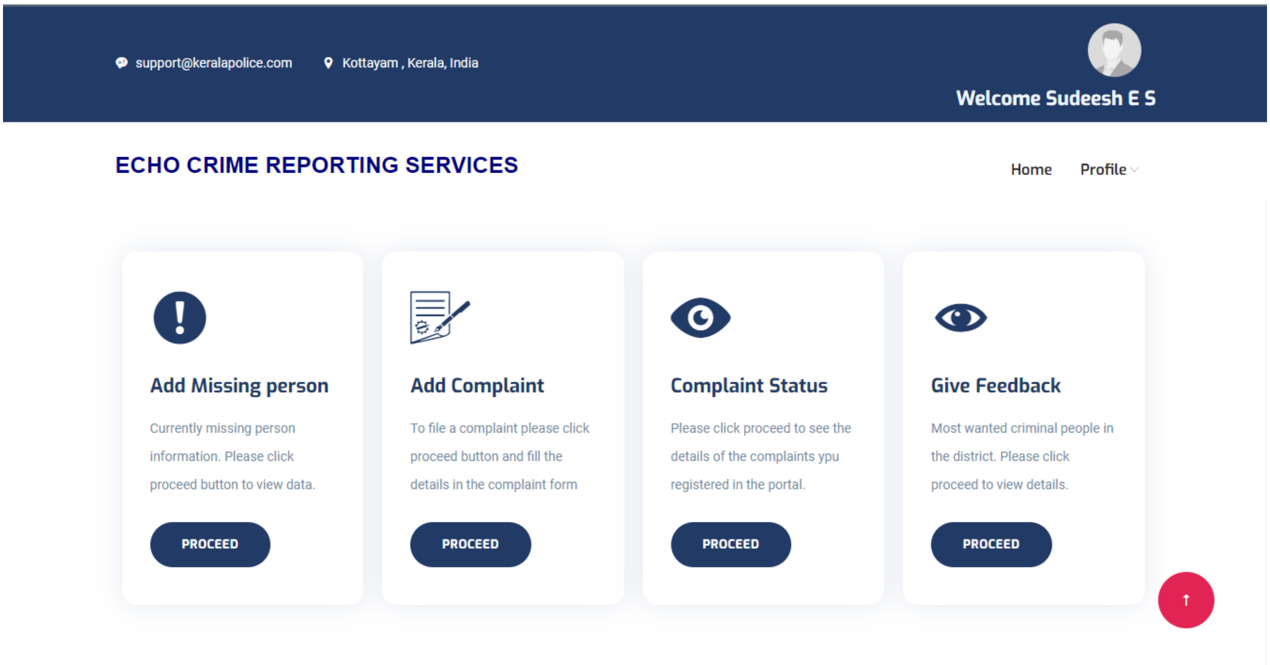
Police Station

Currently you can only access information which is only accessible to visitors. Please login to retrieve the data for specific desgination..

Citizen

Currently you can only access information which is only accessible to visitors. Please login to retrieve the data for specific desgination..

Fig 2: Login Page



ECHO CRIME REPORTING SERVICES

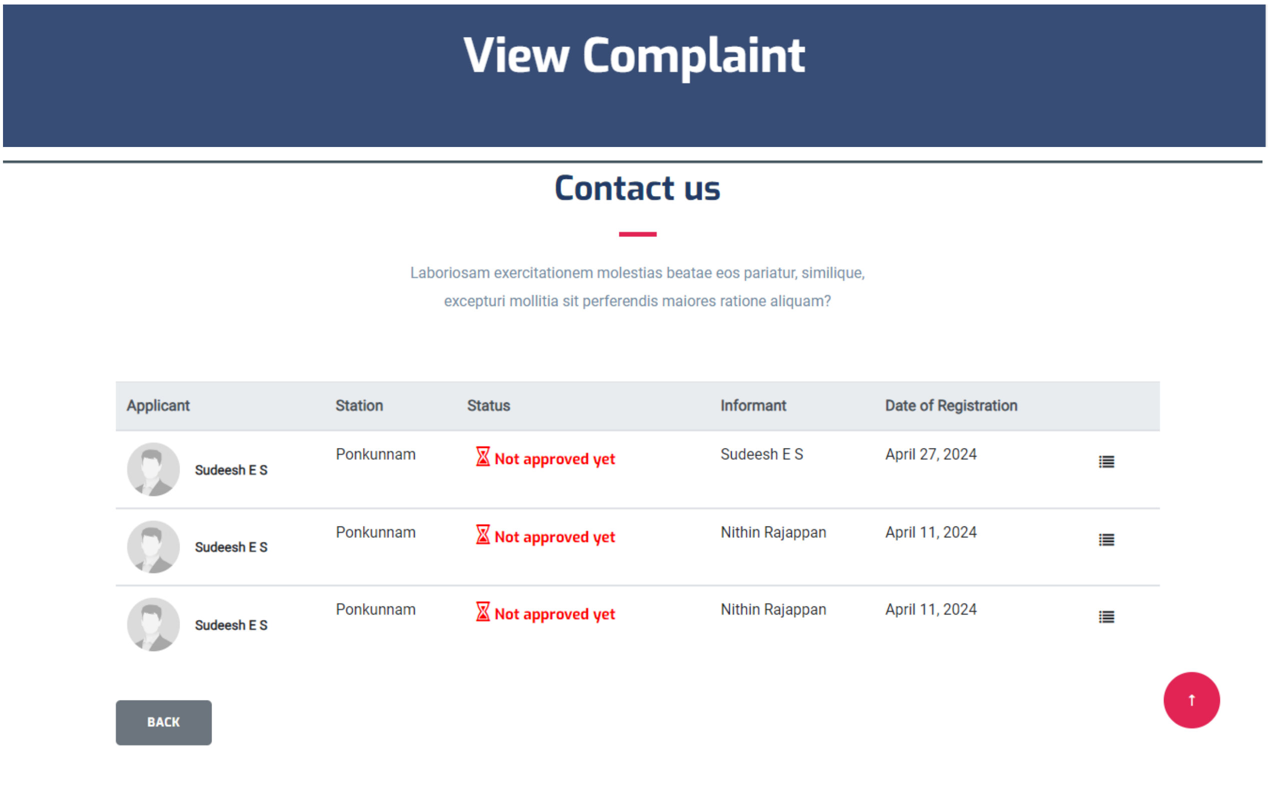
Add Missing person
Currently missing person information. Please click proceed button to view data.

Add Complaint
To file a complaint please click proceed button and fill the details in the complaint form

Complaint Status
Please click proceed to see the details of the complaints ypu registered in the portal.

Give Feedback
Most wanted criminal people in the district. Please click proceed to view details.

Fig 3: User home Page



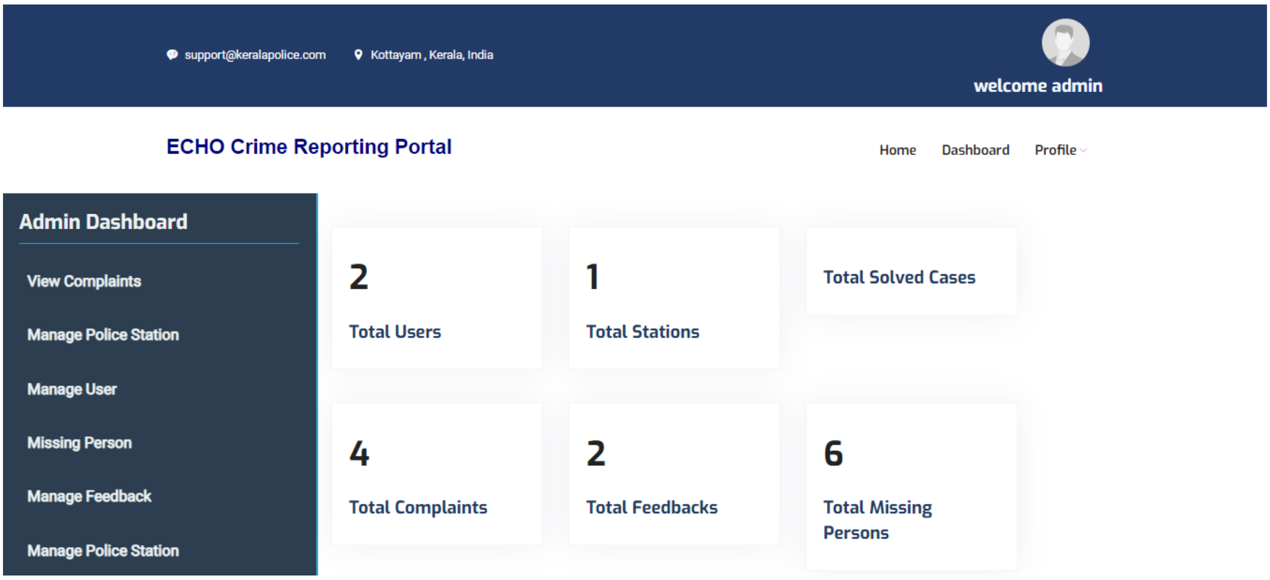
View Complaint

Contact us

Laboriosam exercitationem molestias beatae eos pariatur, similique, excepturi mollitia sit perferendis maiores ratione aliquam?

Applicant	Station	Status	Informant	Date of Registration	
Sudeesh E S	Ponkunnam	⌛ Not approved yet	Sudeesh E S	April 27, 2024	≣
Sudeesh E S	Ponkunnam	⌛ Not approved yet	Nithin Rajappan	April 11, 2024	≣
Sudeesh E S	Ponkunnam	⌛ Not approved yet	Nithin Rajappan	April 11, 2024	≣

Fig 4: Complaint List Page



support@keralapolice.com Kottayam , Kerala, India

welcome admin

ECHO Crime Reporting Portal

Home Dashboard Profile

Admin Dashboard

- View Complaints
- Manage Police Station
- Manage User
- Missing Person
- Manage Feedback
- Manage Police Station

2 Total Users

1 Total Stations

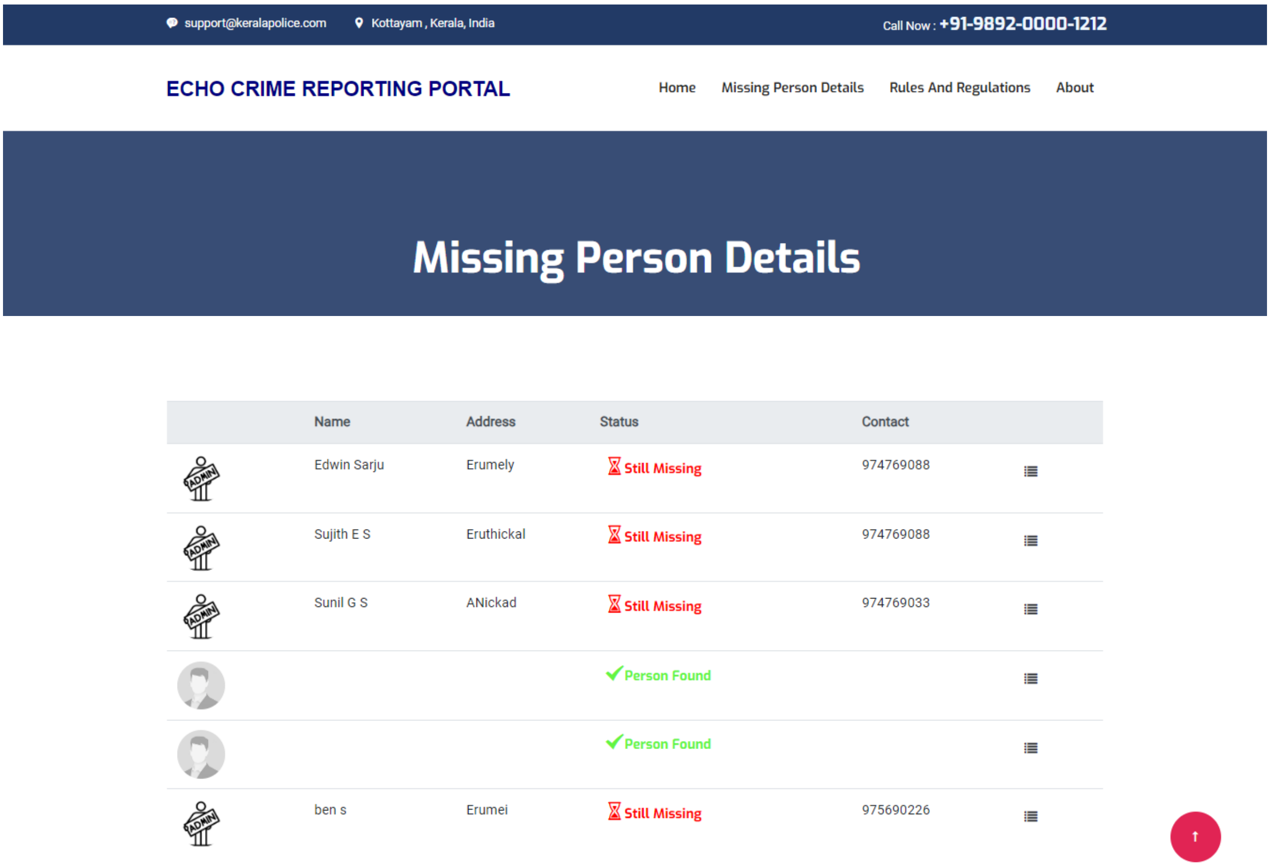
Total Solved Cases

4 Total Complaints

2 Total Feedbacks

6 Total Missing Persons

Fig 4: Admin dashboardt Page



support@keralapolice.com Kottayam , Kerala, India Call Now : +91-9892-0000-1212

ECHO CRIME REPORTING PORTAL

Home Missing Person Details Rules And Regulations About

Missing Person Details

	Name	Address	Status	Contact	
	Edwin Sarju	Erumely	Still Missing	974769088	≡
	Sujith E S	Eruthickal	Still Missing	974769088	≡
	Sunil G S	ANickad	Still Missing	974769033	≡
			✓ Person Found		≡
			✓ Person Found		≡
	ben s	Erumei	Still Missing	975690226	≡

Fig 4:Missing person Page

● **22% Overall Similarity**

Top sources found in the following databases:

- 16% Internet database
- 6% Publications database
- Crossref database
- Crossref Posted Content database
- 21% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**College of Banking and Financial Studies on 2023-04-19** Submitted works	7%
2	**St. Ignatius High School on 2022-07-19** Submitted works	2%
3	**ministriesofaidesinternational.org** Internet	1%
4	**St. Ignatius High School on 2022-07-19** Submitted works	1%
5	**St. Ignatius High School on 2022-07-19** Submitted works	<1%
6	**simplilearn.com** Internet	<1%
7	**issuu.com** Internet	<1%
8	**University of the West Indies on 2024-04-12** Submitted works	<1%

9	Nizwa College of Technology on 2022-12-29 Submitted works	<1%
10	Colorado Technical University Online on 2017-11-23 Submitted works	<1%
11	College of Haringey, Enfield and North East London on 2024-05-03 Submitted works	<1%
12	Kingston University on 2024-04-15 Submitted works	<1%
13	Asia Pacific International College on 2023-08-16 Submitted works	<1%
14	Westford School of Management on 2024-01-03 Submitted works	<1%
15	University of Greenwich on 2024-04-29 Submitted works	<1%
16	riogrande.cs.tcu.edu Internet	<1%
17	irjmets.com Internet	<1%
18	B.S.Abdur Rahman Crescent Institute of Science & Technology on 202... Submitted works	<1%
19	CSU, San Jose State University on 2015-05-18 Submitted works	<1%
20	coursehero.com Internet	<1%

21	Higher Education Commission Pakistan on 2015-06-09 Submitted works	<1%
22	Higher Education Commission Pakistan on 2019-06-07 Submitted works	<1%
23	TAFE Queensland Brisbane on 2022-09-04 Submitted works	<1%
24	University of Leicester on 2017-01-17 Submitted works	<1%
25	Gulf College Oman on 2015-01-06 Submitted works	<1%
26	Open University of Mauritius on 2022-06-18 Submitted works	<1%
27	NCC Education on 2012-05-17 Submitted works	<1%
28	University of Stirling on 2024-04-16 Submitted works	<1%
29	Birkbeck College on 2023-05-01 Submitted works	<1%
30	Middle East College of Information Technology on 2013-05-26 Submitted works	<1%
31	CSU, San Jose State University on 2016-04-28 Submitted works	<1%
32	University of Western Australia on 2023-10-14 Submitted works	<1%

33	First City University College on 2023-05-04 Submitted works	<1%
34	University of Bedfordshire on 2018-01-03 Submitted works	<1%
35	ijarsct.co.in Internet	<1%
36	networkencyclopedia.com Internet	<1%
37	studentsrepo.um.edu.my Internet	<1%
38	University of Hull on 2022-04-28 Submitted works	<1%
39	techtarget.com Internet	<1%
40	ajce.irins.org Internet	<1%
41	docshare.tips Internet	<1%
42	dspace.nau.edu.ua Internet	<1%
43	deskera.com Internet	<1%
44	ifg.cc Internet	<1%

45	Colorado Technical University on 2023-07-06 Submitted works	<1%
46	Fakultet elektrotehnike i računarstva / Faculty of Electrical Engineering... Submitted works	<1%
47	RDI Distance Learning on 2012-11-14 Submitted works	<1%
48	Taylor’s Education Group on 2018-06-17 Submitted works	<1%
49	The British College on 2022-01-27 Submitted works	<1%
50	University of College Cork on 2019-11-12 Submitted works	<1%
51	conference.cusat.ac.in Internet	<1%
52	King's College on 2024-04-12 Submitted works	<1%
53	NCC Education Services on 2009-09-01 Submitted works	<1%
54	Universiti Sains Islam Malaysia on 2017-12-11 Submitted works	<1%
55	University of East London on 2013-12-11 Submitted works	<1%
56	ir.jkuat.ac.ke Internet	<1%

57	mu.ac.in Internet	<1%
58	repository.cuilahore.edu.pk Internet	<1%
59	vdocuments.mx Internet	<1%
60	CSU, Los Angeles on 2021-11-18 Submitted works	<1%
61	Henley College Coventry, Coventry on 2019-03-01 Submitted works	<1%
62	Berlin School of Business and Innovation on 2022-10-10 Submitted works	<1%
63	Colorado Technical University Online on 2014-09-17 Submitted works	<1%
64	Colorado Technical University Online on 2015-03-13 Submitted works	<1%
65	Inlämnad till PSB Academy (ACP eSolutions) den 2013-03-12 Submitted works	<1%
66	The University of Wolverhampton on 2024-04-20 Submitted works	<1%