ECHO
Online Crime Reporting System

Mini Project Report

Submitted by

Sudeesh E S

Reg. No.: AJC22MCA-2092

In Partial Fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS
(MCA TWO YEAR)
[Accredited by NBA]

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC, Kanjirappally, Kottayam, Kerala – 686518]

2023-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**ECHO**" is the bona fide work of **SUDEESH E S (Regno: AJC22MCA-2092)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Ms. Nimmy Francis	**Ms. Meera Rose Mathew**
Internal Guide	**Coordinator**

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **“ECHO”** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date:	**SUDEESH E S**
KANJIRAPPALLY	**Reg: AJC22MCA-2092**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude and to our esteemed Manager **Rev. Fr. Dr. Mathew Paikatt** and dedicated Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of for the invaluable support and guidance extended to us by our Head of the Department, **Rev. Fr. Dr. Rubin Thottupurathu Jose**. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Nimmy Francis** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

SUDEESH E S

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,
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.

Project is mainly divided into 3 modules:

- Admin
- User
- Visitor

Users must be registered to use the system. Upon registration, users can log insecurely 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. The project leverages the Django framework, which provides a robust foundation for building secure and scalable web applications. Django's authentication system ensures user privacy and data protection. Furthermore, the project focuses on creating an intuitive and user-friendly interface, making it accessible to individuals of all technical background.

CONTENT

List of Abbreviation

AJAX	- Asynchronous JavaScript
API	- Application Programming Interface
ATS	- Applicant Tracking System
CSS	- Cascading Style Sheet
HTML	- Hyper Text Markup Language
IDE	- Integrated Development Environment
JS	- Java Script
JSON	- JavaScript Object Notation
OAuth	- Open Authorization
ORM	- Object-Relational Mapping
PyPI	- Python Package Index
REST	- Representational State Transfer
SQL	- Structured Query Language
UML	- Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

In the fast-paced evolution of the job market, the demand for a sophisticated, efficient, and user-friendly platform facilitating the connection between employers and job seekers has become paramount. "ECHO" stands out as an innovative web application built on the Django framework, specifically crafted to tackle the prevailing inefficiencies and challenges within the hiring process. Its central objective is to deliver an enhanced experience for both employers and job seekers, ultimately refining and optimizing the entire recruitment journey.

"ECHO" endeavors to stand as a catalyst for positive transformation, enhancing overall efficiency and satisfaction within the intricate realm of recruitment processes.

In its commitment to innovation, "ECHO" seeks to bridge the existing gaps and bottlenecks, providing a seamless interface that facilitates a more intuitive and streamlined interaction between employers and job seekers. Through this advanced yet accessible platform, the aim is not merely to meet industry standards but to set a new benchmark for excellence in recruitment solutions. As the job market continues to evolve, "ECHO" stands poised to redefine and elevate the hiring experience, contributing to a paradigm shift in how employers and job seekers connect and engage throughout the recruitment journey.

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 job provider module empowers law enforcement by facilitating job posting management, candidate search, and efficient navigation through a user-friendly interface.

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

"Echo" endeavors to revolutionize the crime reporting landscape by introducing an innovative approach to incident documentation and law enforcement collaboration. Addressing the current challenges in reporting criminal activities, this system aims to streamline the reporting process, reducing response times and enhancing overall efficiency in addressing incidents. With a modern front-end interface and a robust back-end infrastructure, "Echo" aspires to establish a new standard for seamless, data-driven interactions between the public and law enforcement, contributing to a safer and more connected community.

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 "ECHO" system addresses critical challenges in the current recruitment landscape by leveraging advanced technologies. It focuses on enhancing data management, reducing missed opportunities, and streamlining the screening process. The proposed system aims to replace manual and paper-based data handling with an automated approach, ensuring accuracy and efficiency in storing, retrieving, and analyzing diverse data generated throughout the hiring process.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- **Enhanced Job Discovery for Candidates:** "ECHO" acts as a catalyst for candidates, facilitating the discovery and application for career opportunities aligned with their preferences. The system's user-friendly interface and efficient processes simplify the job search experience, enabling candidates to navigate and access relevant opportunities with ease.

- **Streamlined Recruitment Processes for Employers:** "ECHO" streamlines recruitment processes, enhancing efficiency and transparency for employers. Intuitive workflows and tools empower recruiters to focus on identifying the most suitable candidates, reducing the time and effort traditionally associated with hiring.
- **Improved Communication and Collaboration:** The proposed system emphasizes enhanced communication and collaboration between employers and candidates. "ECHO" provides clearer insights into application statuses and interview processes, fostering a more transparent and communicative environment. This ensures that all stakeholders remain well-informed throughout the recruitment process.
- **Interactive UI:** The system incorporates an interactive user interface, enhancing the overall user experience and satisfaction for both employers and job seekers. It provides easy-to-use web interfaces, offering valuable and actionable information and insights about the hiring process.

.

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 Employee Recruitment System holds the promise of transforming and optimizing the recruitment processes of organizations. 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 Economical 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 recruitment costs, time savings, improved candidate quality, 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

Technical feasibility involves study to establish the technical capability of the system being created to accomplish all requirements to the user. The system should be capable of handling the proposed volume of data and provide users and operating environment to increase their efficiency. 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 for the successful development and deployment of the system such as Infrastructure Assessment, compatibility, scalability, data security and privacy.

3.1.3 Behavioral 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 recruitment costs, time savings, improved candidate quality, 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.

3.1.4 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

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: We handle a wide range of criminal activities. 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, investigation, and case management.

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: Are there any machine learning or analytics features you envision for this crime reporting system?

A: While not a primary focus, we see potential for machine learning modules to enhance crime pattern recognition, suspect identification, and predictive policing. These capabilities could improve the efficiency of our crime management processes.

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

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- Intel Core i3
RAM	- 4 G B
Hard disk	- 2 5 6 G B

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, Django, HTML5, AJAX, J Query, Python, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PYTHON

Python is a high-level programming language that is frequently used to create a wide variety of software applications, from web development and data analysis to scientific computing and artificial intelligence. Python is a well-liked option for project development because of its simplicity, usability, and versatility. Python is the perfect language for project development because of its enormous standard library and ecosystem of third-party libraries, tools, and frameworks. Python's syntax is simple to understand and read, and because its code is frequently shorter than that of other programming languages, projects can be developed more quickly. Python also provides a wide range of debugging, profiling, and testing tools that make it simpler for developers to find and fix problems rapidly.

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.

3.3.3Django

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.

.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. They enhance communication by providing a visual representation that is easily understood by technical and non-technical stakeholders alike. They aid in analysis and design by capturing system requirements, relationships, and interactions. UML diagrams also facilitate software development by serving as a blueprint for developers to implement and test systems.

UML includes the following nine diagrams.

• Use case diagram

• Sequence diagram

• State chart diagram

• Activity diagram

• Class diagram

• Object diagram

• Component diagram

• Deployment diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modelling Language), a standard notation for the modelling of real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalogue updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

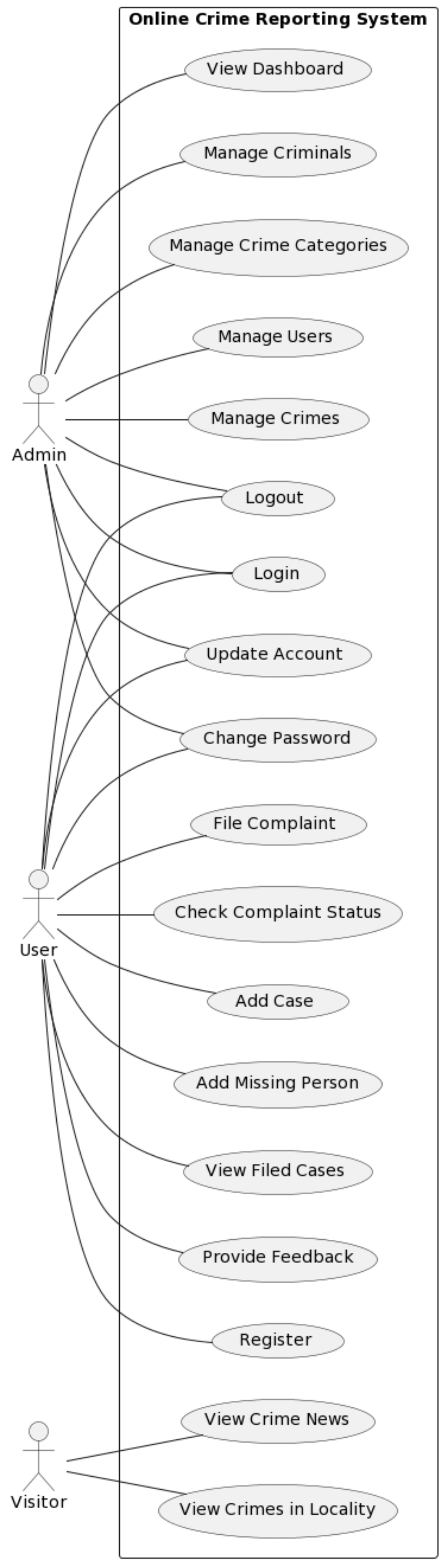


Fig 1: Use case diagram for ‘ECHO ‘

4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems. In a sequence diagram, objects are represented as vertical lifelines, and messages are depicted as horizontal arrows between the lifelines. The arrows indicate the flow of communication, including method calls, responses, and signals. The sequence of messages helps to depict the dynamic behavior of the system and how different objects collaborate to achieve specific functionality. Sequence diagrams are useful for understanding and analyzing the interactions within a system, especially in scenarios where multiple objects or actors are involved. They allow software developers and designers to visualize the chronological order of events, identify potential bottlenecks or issues, and ensure that the system behaves as intended. Overall, sequence diagrams serve as a valuable tool in software development for modeling and understanding the dynamic behavior of systems, providing a visual representation that illustrates the interactions and order of events between different components or objects within the system. 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.

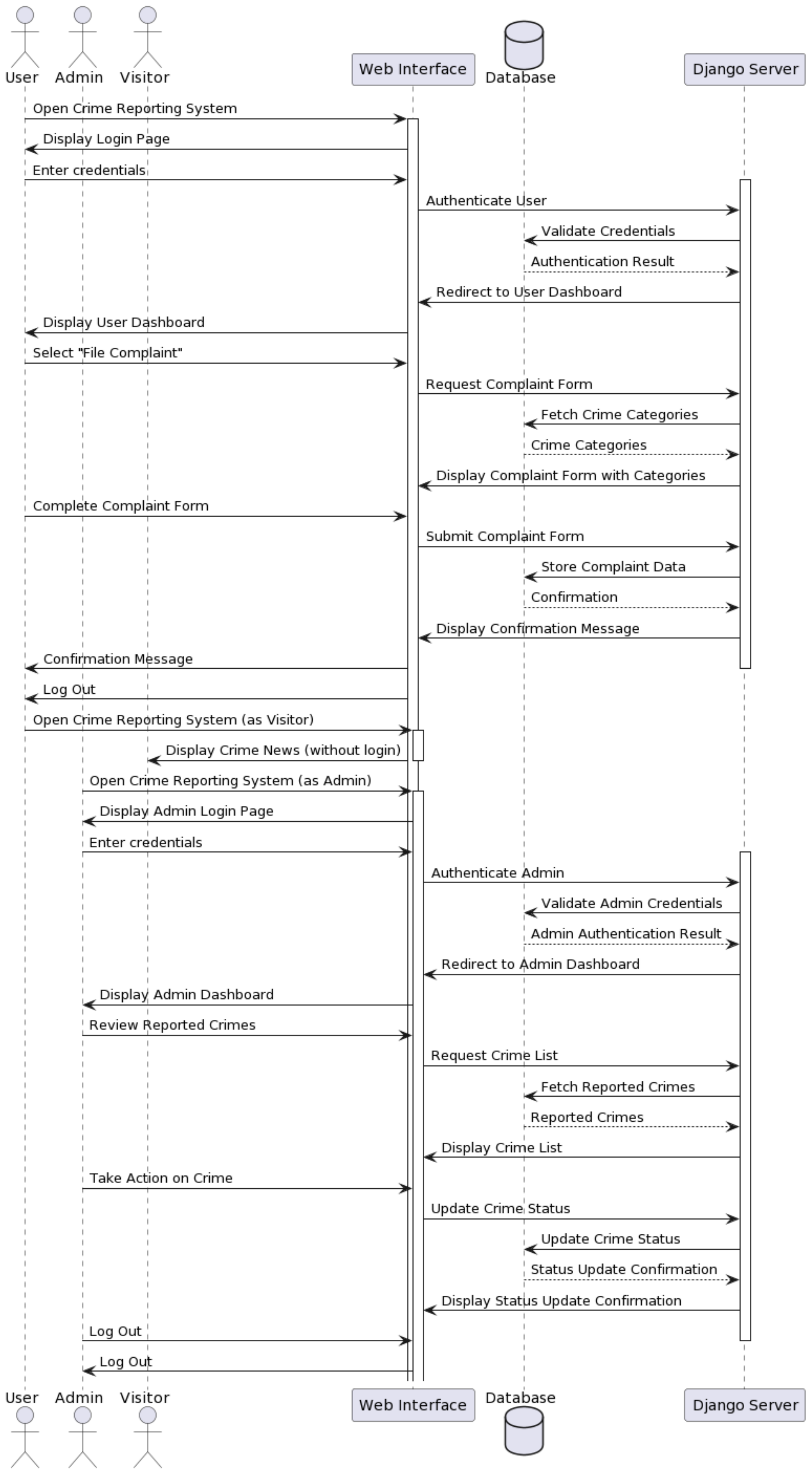


Fig 2: Sequence diagram for ‘ECHO ‘

4.2.3 State Chart Diagram

State Chart Diagram is a type of UML (Unified Modeling Language) diagram used to model the behavior of a system or object over time. A State Chart Diagram shows the various states that an object or system can be in, and how it transitions from one state to another. Each state represents a particular condition or mode that the object or system can be in, and may have associated actions or behaviors. Transitions between states are triggered by events, which may be external (such as a user input) or internal (such as a timer). State Chart Diagrams can be used to model complex systems with many states and transitions, such as user interfaces, control systems, or business processes. State Chart Diagrams are often used in conjunction with other UML diagrams, such as Use Case Diagrams or Class Diagrams. State Chart Diagrams can be used to communicate system behavior to stakeholders, including developers, testers, and users. State Chart Diagrams can be implemented using various programming languages and tools.State Chart Diagrams can help improve system reliability and maintainability by providing a clear and concise model of system behavior that can be used for testing, debugging, and maintenance. In a state chart diagram, states are represented as rounded rectangles, and transitions between states are depicted as arrows. 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.

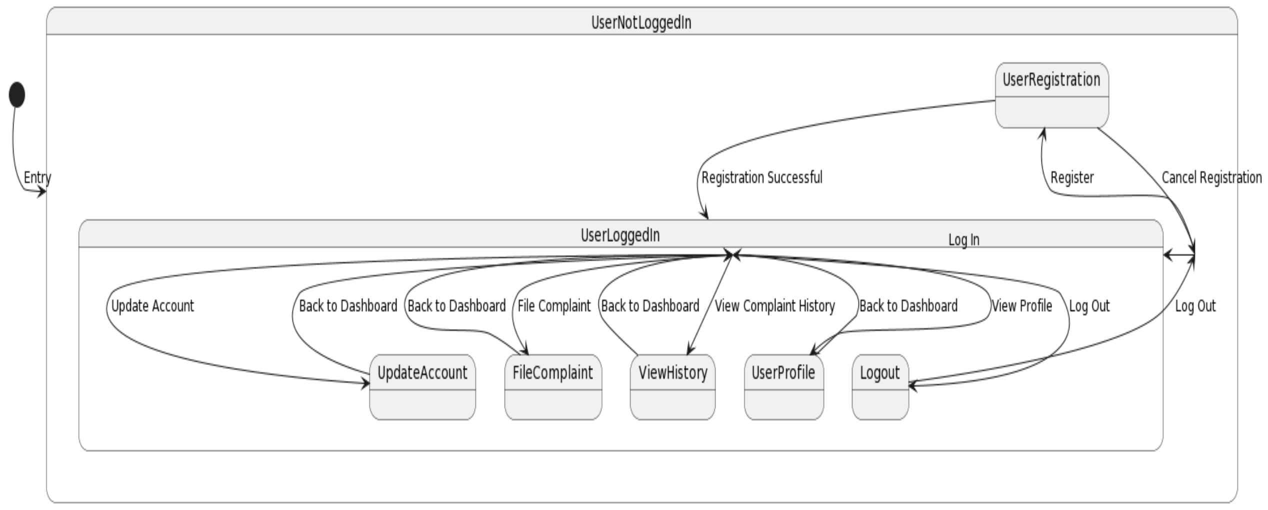
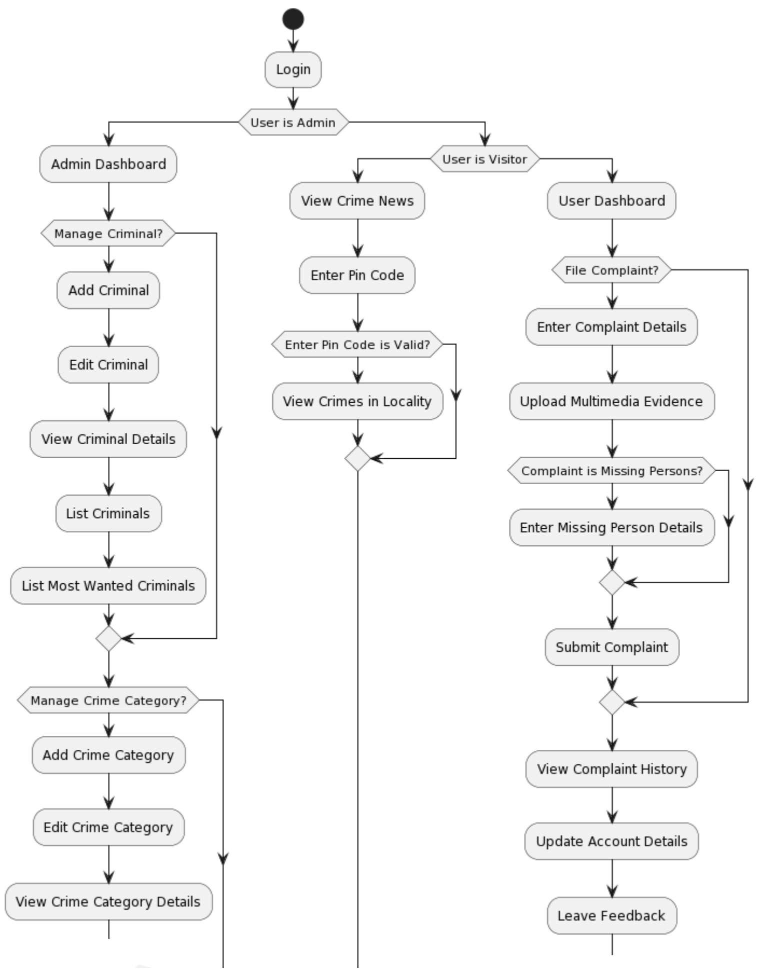


Fig 3: State Chart diagram for ‘ECHO ‘

4.2.4 Activity Diagram

Activity diagrams depict how different levels of abstraction of activities are linked to provide a service. Typically, an event should be completed by some activities, particularly when the activity is intended to do multiple separate goals that need coordination. Another typical requirement is how the events in a single use case interact with one another, particularly in use cases where operations may overlap and require coordination. It may also be used to show how a collection of interrelated use cases interacts to reflect business operations. In an activity diagram, activities are represented as rounded rectangles, and arrows indicate the flow of control between activities. Decision points are depicted as diamonds, where different paths or branches can be taken based on conditions or criteria. Forks and joins represent parallel or concurrent execution of activities. Swim lanes can also be used to group related activities performed by specific actors or roles. Activity diagrams are valuable for modelling and analyzing complex processes or workflows. They help to visualize the sequence of steps, dependencies, and decision points within a system, providing a clear understanding of how the process operates. This aids in identifying potential bottlenecks, inefficiencies, or areas for optimization.



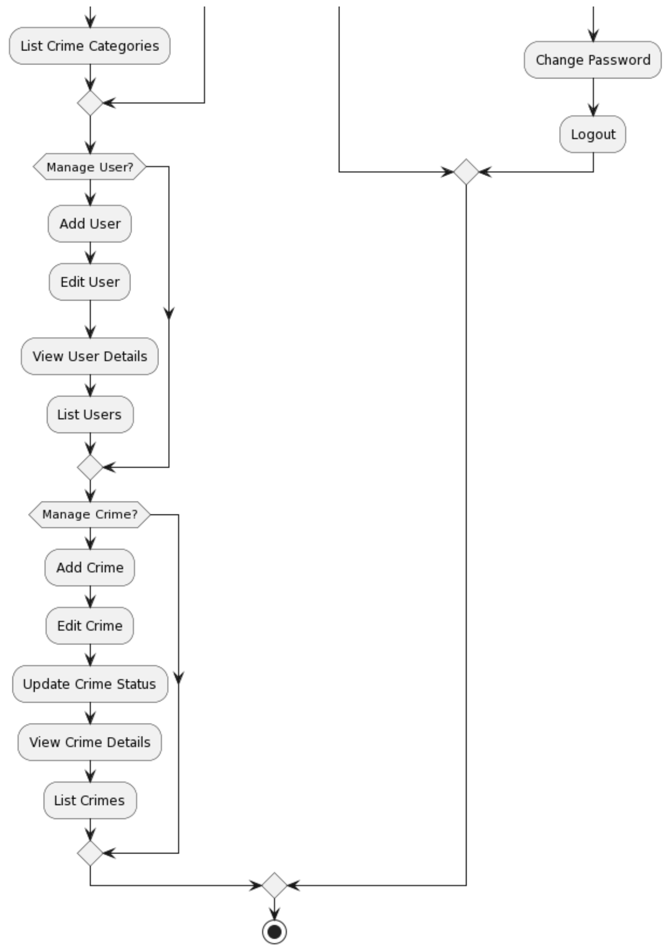


Fig 4: Activity diagram for ‘ECHO ‘

4.2.5 Class Diagram

Class diagram is a static diagram. It represents the static view of the application. Class diagrams are useful for visualizing, describing, and documenting various system components as well as for writing executable code for software applications. A class diagram describes the constraints imposed on the system together with the properties and operations of a class. The only UML diagrams that can be directly converted into object-oriented languages are class diagrams, which are extensively utilized in the designing of object-oriented systems. An assortment of classes, interfaces, affiliations, partnerships, and limitations are displayed in a class diagram. Class diagrams are widely used in software development to design and document the structure of object-oriented systems. Key elements in a class diagram include associations, which represent relationships between classes. Aggregation and composition relationships are used to represent whole-part relationships between classes. They aid in system design, modelling, and documentation, allowing software developers and stakeholders to understand the system's architecture, dependencies of different classes.

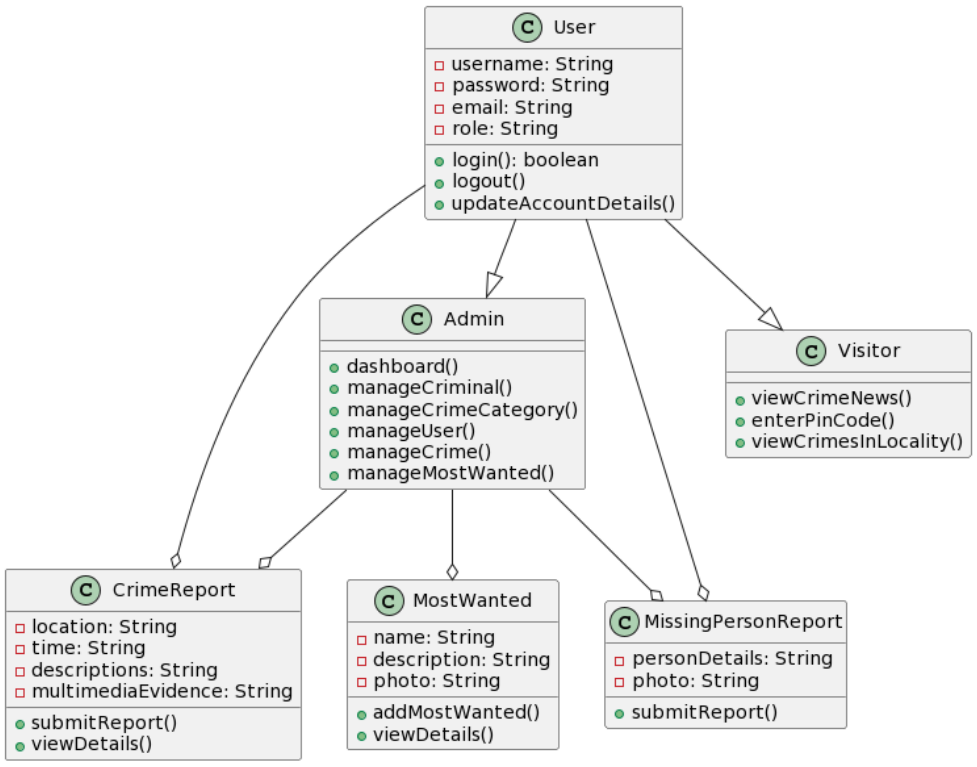


Fig 5: Class diagram for 'ECHO '

4.2.6 Object Diagram

A chart represents an instance of a chart. The basic concepts of art and art are the same. The diagram also represents a static view of the system, but this static view is a snapshot of the system at a particular time. A diagram is used to illustrate a group of objects and their relationships as an example. The intended use of the image should be clearly understood. Charts are used in the same way as charts. The difference is that the diagram represents an abstract model of classes and their relationships. However, a painting represents a sample of a particular time and is authentic in nature. This means that the graph is close to the actual system behavior. The goal is to capture a static view of the system at a given time. In an object diagram, objects are depicted as individual instances of classes, usually with their attributes and current values. The relationships between objects are shown using lines and arrows, representing associations, dependencies, etc. Object diagrams are useful for visualizing and understanding the structure and relationships between objects within a system. They help to depict real-world scenarios or specific instances of a system, enabling software developers and stakeholders to gain insights into how objects collaborate and interact. One of the key benefits of object diagrams is that they provide a concrete representation of objects and their states. Object diagrams are particularly valuable for testing and debugging, as they help to analyze and understand the behavior of specific instances or scenarios.

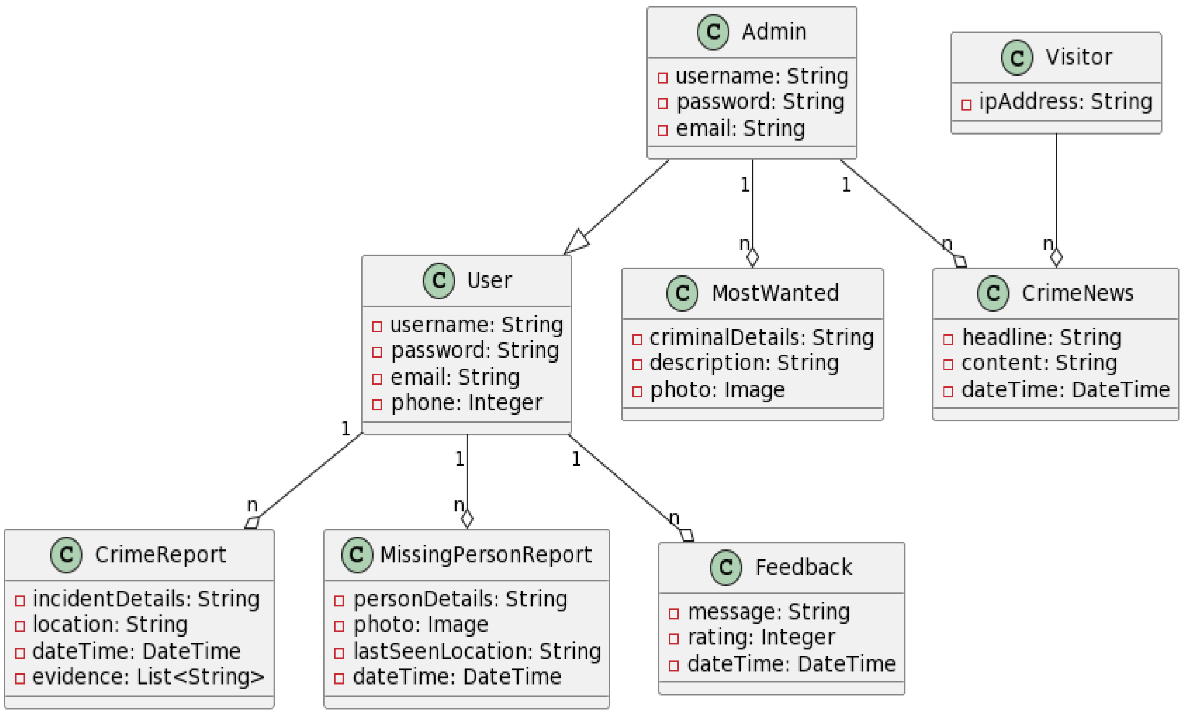


Fig 6: Object diagram for 'ECHO '

4.2.7 Component Diagram

Component diagrams have different behaviors and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams may also be used to construct systems that can be run. Key elements in a component diagram include interfaces, which define the contract or communication protocol between components. Dependencies represent the reliance of one component on another, indicating that changes in one component may impact others. Component diagrams also show the provided and required interfaces of each component, specifying the services or functionalities it offers or requires. Component diagrams are commonly used in software development and system design to depict the overall structure and organization of a system.

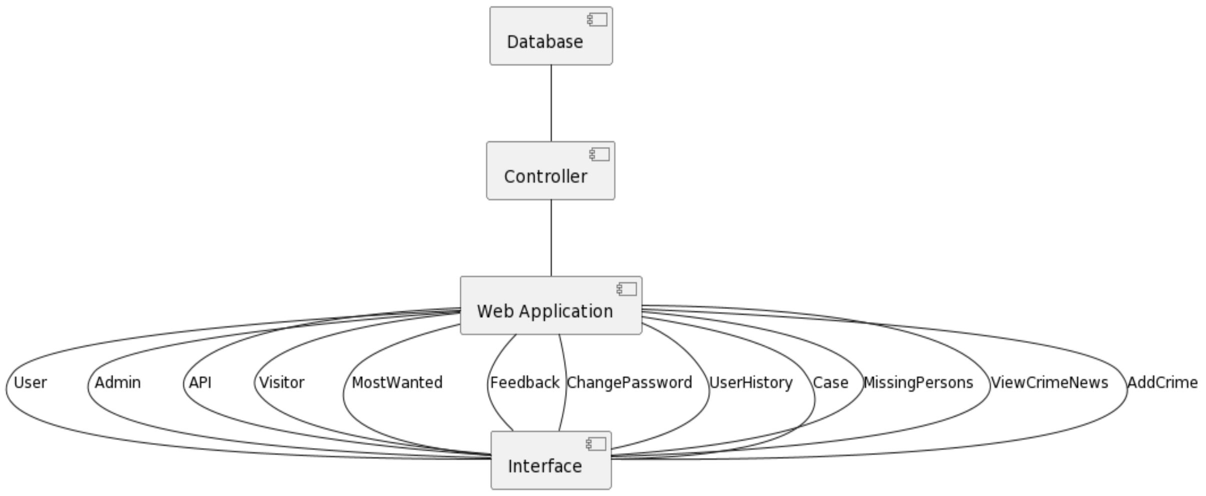


Fig 7: Component diagram for ‘ECHO ‘

4.2.8 Deployment Diagram

An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system. By using it, you can comprehend how the hardware will physically deliver the system. In contrast to other UML diagram types, which primarily depict the logical Key elements in a deployment diagram include nodes, which represent the hardware or software execution environments, and their relationships with components and artifacts. Communication paths, such as network connections or communication protocols, are also depicted to illustrate the interaction and data flow between nodes. Deployment diagrams are commonly used in software development and system architecture to design, communicate, and document the deployment strategy of a system. They assist in planning the allocation of resources, identifying potential points of failure, and ensuring the scalability and efficiency of the system's deployment. In summary, deployment diagrams provide a visual representation of how components and artifacts are deployed on nodes within a system.

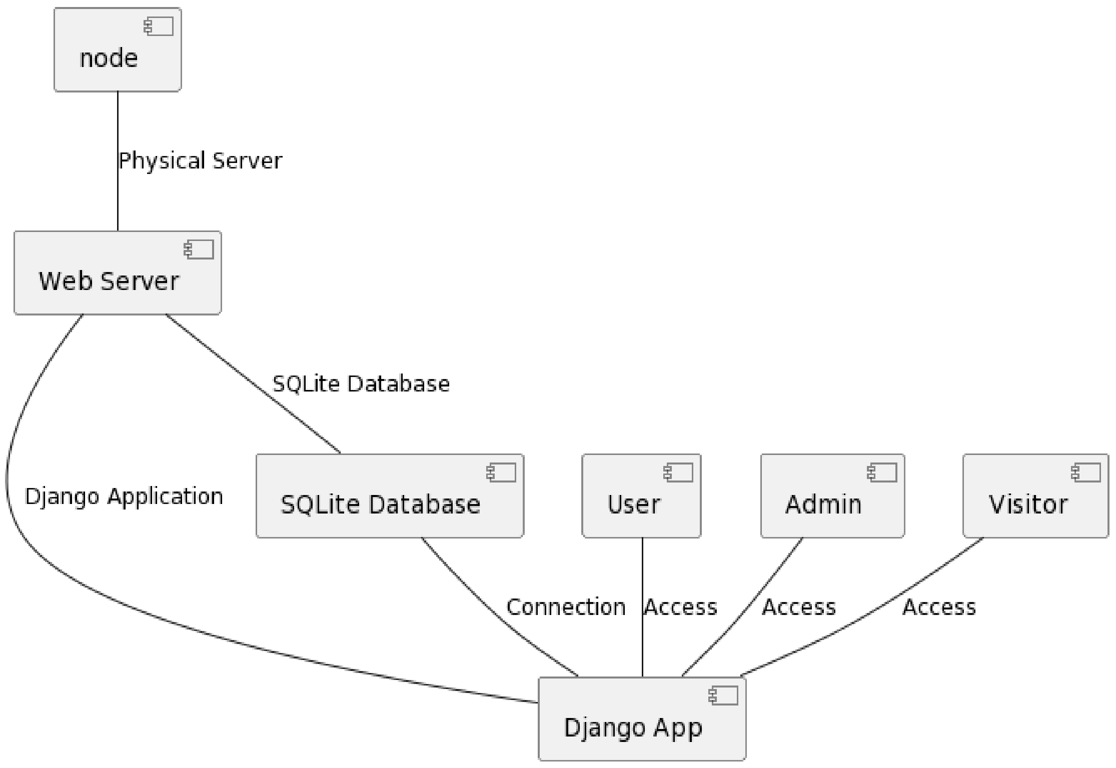


Fig 8: Deployment diagram for ‘ECHO ‘

4.2.9 Collaboration Diagram

A collaboration diagram in Unified Modeling Language (UML) is a visual representation of how objects interact with each other to achieve a specific task or goal within a system. It illustrates the dynamic relationships and collaborations between objects, showcasing the flow of messages exchanged during runtime. Objects are represented as nodes, and the interactions are depicted through arrows indicating the direction of communication. The primary focus is on visualizing the relationships and dependencies between objects, making collaboration diagrams valuable for understanding the runtime behavior of a system. By providing a clear and concise view of object interactions, collaboration diagrams enhance communication among stakeholders, aiding in the design and development of software systems.

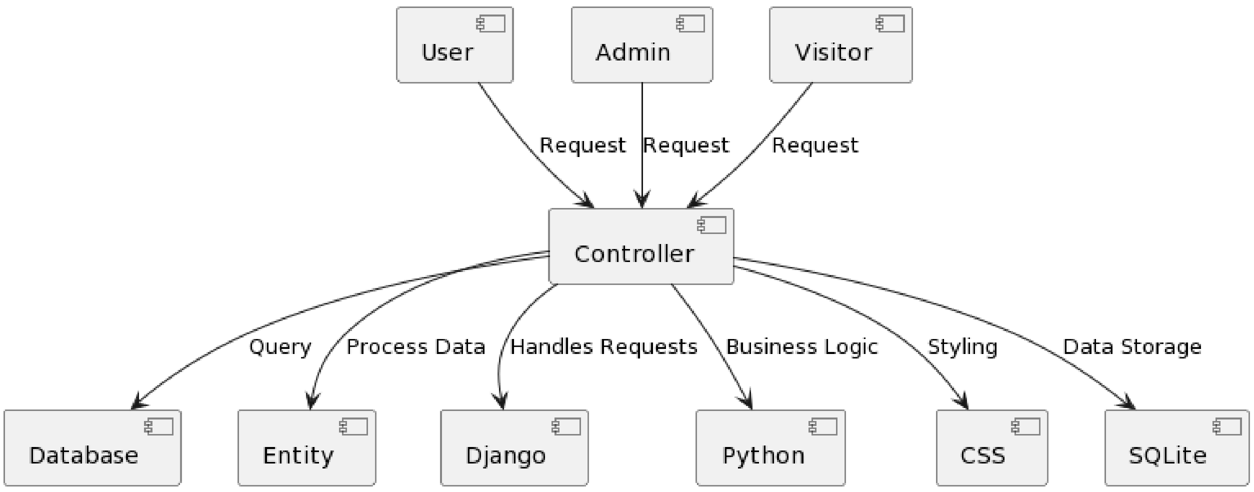
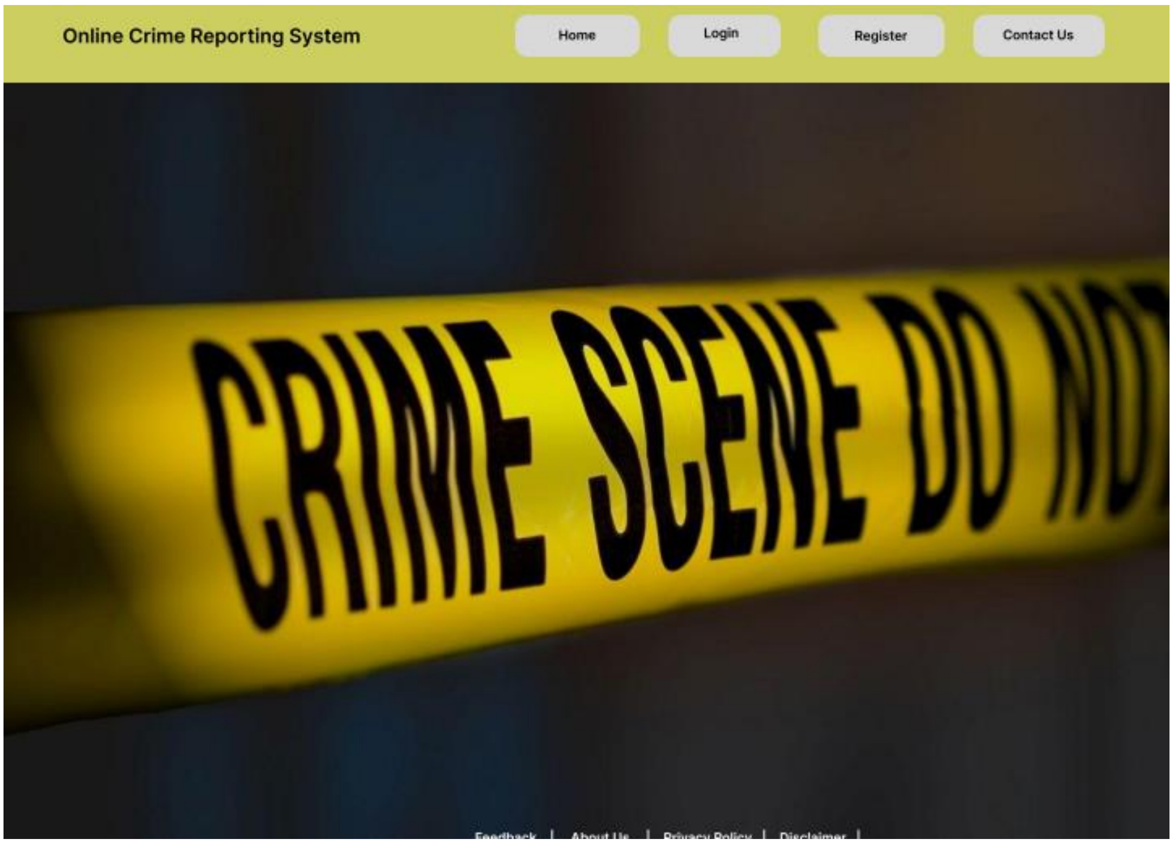


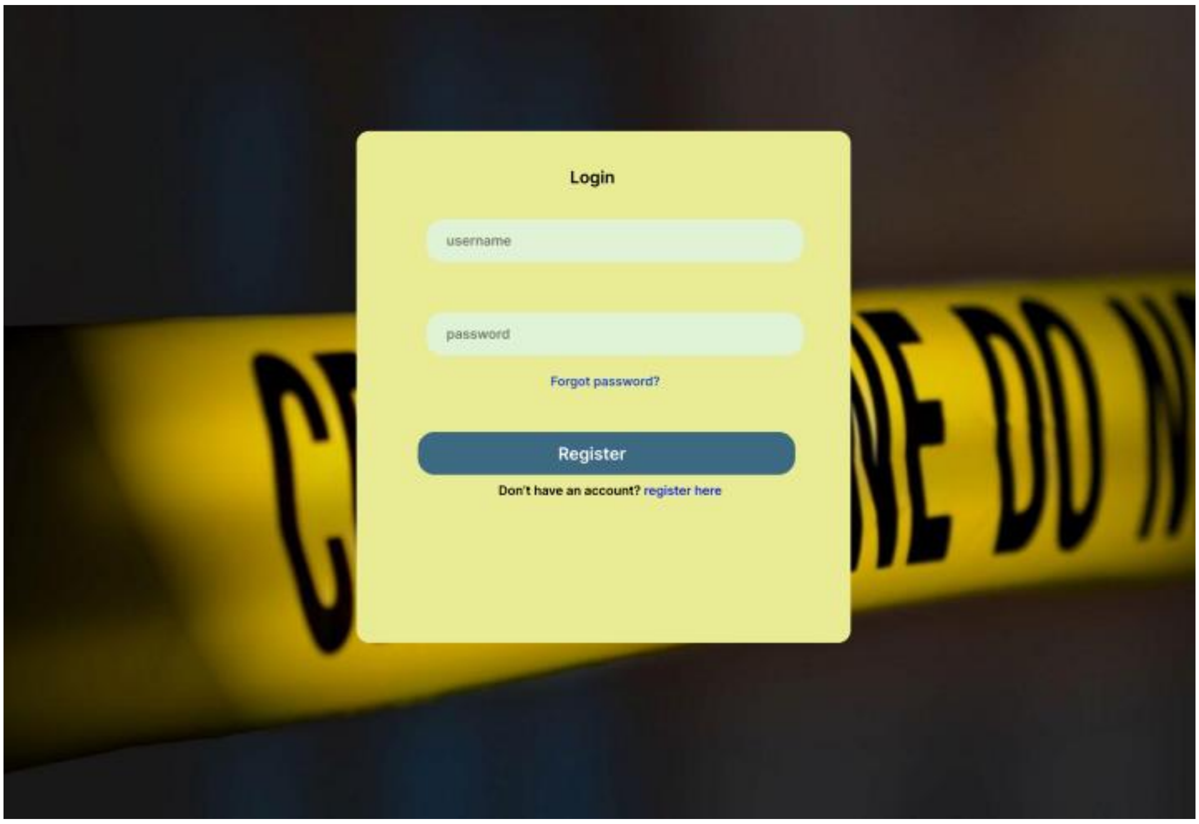
Fig 9: Collaboration diagram for 'ECHO '

4.3 USER INTERFACE DESIGN USING FIGMA

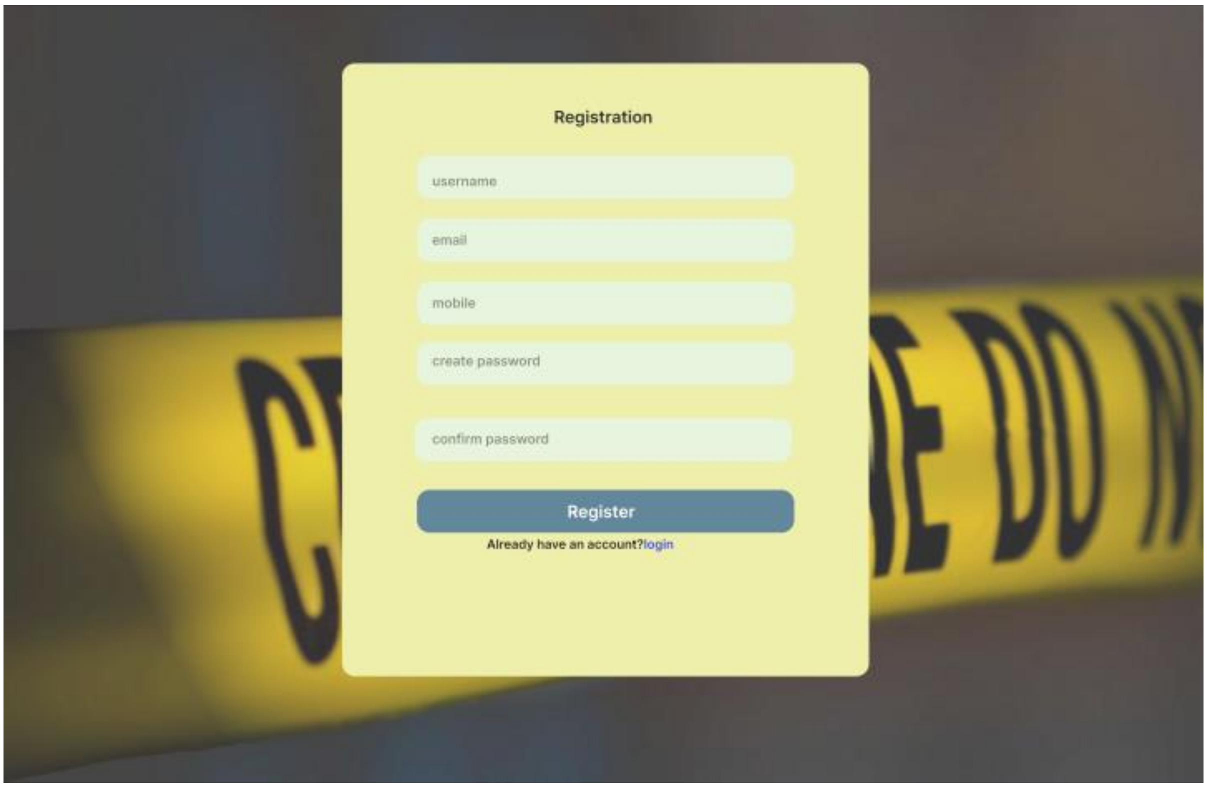
Home


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

Login



Register



4.4 DATABASE DESIGN

A database is a process with the ability to store information to retrieve stored information from users in a quantity and quality. Data is the purpose of any database and must be protected. A database is created in two steps. Write out the user's needs in order to build a document that satisfies them as precisely as you can. This process, known as data-level creation, is separate from each DBMS. This data-level design is converted into a design for the specific DBMS being used to carry out the queries in the second stage. The details of the exact DBMS to be used are included in this step, which is referred to as physical layer design. System design and database design are complementary processes.

4.4.1 Relational Database Management System (RDBMS)

Data is represented via a relational model as a set of relationships. Every relationship is like a table of values or datasets. In relational model terminology, rows are called tuples, column headings are called attributes, and tables are relations. A relational file consists of tables, each of which is given a unique name. A row in the chart represents a group of related values.

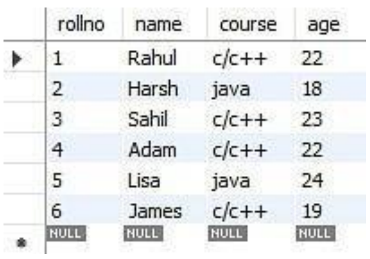
4.4.2 Normalization

The table is a relationship. Rows in a table are called tuples. A tuple is an ordered collection of n elements. Lines are called attributes. Relationships are established between all tables in the database. This ensures the integrity of information and the relationship between organizations. Domain D is a set of atomic charges. One way to clarify the author is to determine the file type by extracting the main file that created the name. It's also helpful to give a field a name to help define its value.

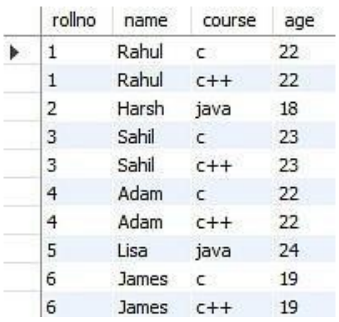
First Normal Form (1NF)

If a table's atomicity is 1, it is said to be in its first normal form. Atomicity dictates that a single cell cannot contain multiple values in this instance. It can only include one attribute with a single value. The multi-valued attribute, composite attribute, and their combinations are forbidden by 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.



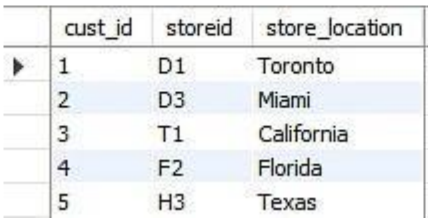
rollno	name	course	age
1	Rahul	c/c++	22
2	Harsh	java	18
3	Sahil	c/c++	23
4	Adam	c/c++	22
5	Lisa	java	24
6	James	c/c++	19
NULL	NULL	NULL	NULL



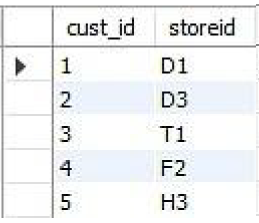
rollno	name	course	age
1	Rahul	c	22
1	Rahul	c++	22
2	Harsh	java	18
3	Sahil	c	23
3	Sahil	c++	23
4	Adam	c	22
4	Adam	c++	22
5	Lisa	java	24
6	James	c	19
6	James	c++	19

Second Normal Form (2NF)

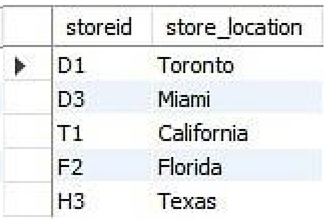
The first requirement for a table to be in Second Normal Form is that it must be in First Normal Form; the table should also not have partial dependency, which means that the correct subset of the candidate key should give a non-prime attribute. Now, let's understand the Second Normal Form using an example. You must divide the table into two pieces in order to get it to Second Normal Form. You will then the tables given 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



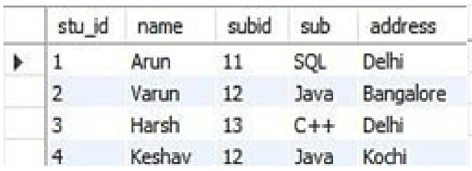
storeid	store_location
D1	Toronto
D3	Miami
T1	California
F2	Florida
H3	Texas

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.

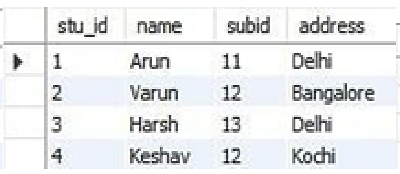
Third Normal Form (3NF)

The table must first be in the Second Normal Form in order for it to be in the Third Normal Form. Non-prime attributes that are not a part of the candidate key should not depend on other non- prime characteristics in a table, according to the second requirement, which states that there should be no transitive reliance for non-prime attributes. As a result, a transitive dependency is a functional dependency in which A determines C indirectly due to A B and B C (where B A is not true).

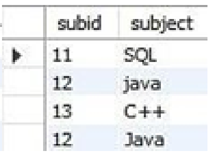
A student table with their student ID, name, subject ID, and subject is provided below. In the student table shown above, sub_id and sub_id are determined by stu_id. As a result, sub is determined by stu_id via sub_id. As a result, the table is said to have a transitive functional dependency and does not meet the third normal form requirement. Now, split the table as indicated below to convert it to the third normal form:



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 entails removing or erasing data from a storage device on purpose and permanently to make sure it cannot be restored. Normally, when data is removed from storage media, the medium is not truly wiped, and an attackerwho gains access to the device can recover the data. Serious questions about security and data privacy are raised by this. Sanitization involves cleaning storage media so that no data remains on the device and that no data can be recovered even with cutting-edge forensic techniques.

4.4.4 Indexing

By reducing the number of disc accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes.

• The search key, which is the first column, contains a duplicate of the table's primary key or potential primary key. These values are kept in sorted order so that it is easy to get the corresponding data.

Note: The data may or may not be kept in sorted order.

• The address of the disc block on which that specific key value is stored is held by a series ofpointers in the second column, which is designated as the Data Reference or Pointer.

4.5 TABLE DESIGN

1. User

Primary Key: **user_id**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	user_id	AutoField	Primary Key (PK)	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	first_name	CharField	Not Null	First name of the user.
5	last_name	CharField	Not Null	Last name of the user.
6	role	CharField	max_length=20	User role (User, admin).

2. Complaint

Primary Key: **complaint_id**

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

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	complaint_id	AutoField	Primary Key (PK)	Complaint_ID
2	user	CharField	Foreign Key	Date of Birth
3	name	CharField	max_length=50	Victim name
4	place	CharField	max_length=10	Incident Location
5	description	CharField	max_length=25	Complaint Description
6	evidence	ImageField	Not Null	Evidence photo

3. Criminal

Primary Key: **criminal_id**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	criminal_id	AutoField	Primary Key (PK)	Criminal ID
2	name	CharField	max_length=50	Name
3	description	TextField	max_length=150	Description
4	photo	ImageField	Not Null	Criminal photo

4. Feedback

Primary Key: **feedback_id**

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

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	feedback_id	IntegerField	Primary Key (PK)	Feedback ID
2	message	CharField	Not Null	Content
3	date	DateTimeField	Not Null	Date submitted

5. MissingPerson

Primary Key: **missingperson_id**

Foreign Key: **user_id** references table **User**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	missingperson_id	AutoField	Primary Key (PK)	Missing person ID
2	user_id	IntegerField	ForeignKey(FK)	User ID
3	gender	CharField	Not Null	Gender
4	description	TextField	Not Null	Description
5	photo	ImageField	default=timezone.now	Image of missing person
6	status	CharField	Not Null	Status
7	age	IntegerField	Not Null	Age

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner. Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checkingthat what has been specified is what the user actually wanted.

5.2 TEST PLAN

An extensive document that describes the testing strategy and approach for a specific project or product is called a test plan. Selenium is one of the most popular tools for automating tests when it comes to web application testing. A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. Therefore, the test plan should include information on the mean time to failure, the cost to locateand correct the flaws, the residual defect density or frequency of occurrence, and thenumber of test work hours required for each regression test.

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 concentrates verification efforts on the software component or module, which is the smallest unitof software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. The unit testing's disclosed scope and the tests' relative complexity. Unit testing is white-box focused, and numerous components may be tested simultaneously. To guarantee that data centers and exits the program unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during all phases of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at leastonce, boundary conditionsare evaluated. Finally, each path for managing errors is examined. Before starting any other test, tests of data flow over a module interface must be completed.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performingunit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing. The functional requirements of the software are the main emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every program requirement. Errors in data structures or external data access, erroneous or missing functions, interface errors, performance issues, initialization issues, and termination issues are all types of faults that black box testing looks for.

5.2.4 Output Testing or User Acceptance Testing

User approval of the system under consideration is tested; in this case, it must meet the needs of the company. When developing, the program should stay in touch with the user and perspective system to make modifications as needed. With regard to the following points.

- Input Screen Designs
- Output Screen Designs

The aforementioned testing is carried out using a variety of test data. The preparation of test data is essential to the system testing process. The system under investigation is then put to the test using the prepared test data. When testing the system, test data issues are found again and fixed using the testing procedures described above. The fixes are also logged for use in the future.

5.2.5 Automation Testing

A test case suite is executed using specialized automated testing software tools as part of the software testing technique known as automation testing. The test stages are meticulously carried out by a human performing manual testing while seated in front of a computer. Additionally, the automation testing software may generate thorough test reports, compare expected and actual findings, and enter test data into the System Under Test. Software test automation necessitates significant financial and material inputs. Repeated execution of the same test suite will be necessary during subsequent development cycles.

5.2.6 Selenium Testing

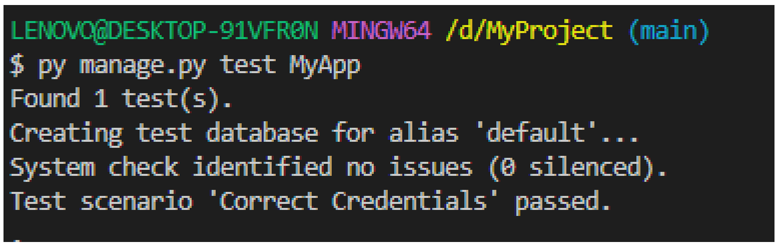
Selenium is a free (open-source) automated testing framework used to verify web applications across different browsers and platforms. You can use multiple programming languages to create Selenium test scripts like Java, C#, Python, etc. Testing done using Selenium testing tool is usually referred to as Selenium testing Since Selenium is a collection of different tools, it also had different developers. Below are the key people who have made significant contributions to the Selenium project. Moreover, Selenium's extensibility allows integration with other testing frameworks, tools, and technologies. It can be combined with testing frameworks like TestNG or JUnit for advanced test management and reporting. It integrates well with popular build tools, source control systems, and defect tracking systems.

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 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: 03/12/2023**		
Module Name: Login Screen			**Test Executed By: Ms. Nimmy Francis**		
Test Title: Admin Login			**Test Execution Date: 03/12/2023**		
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		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.					

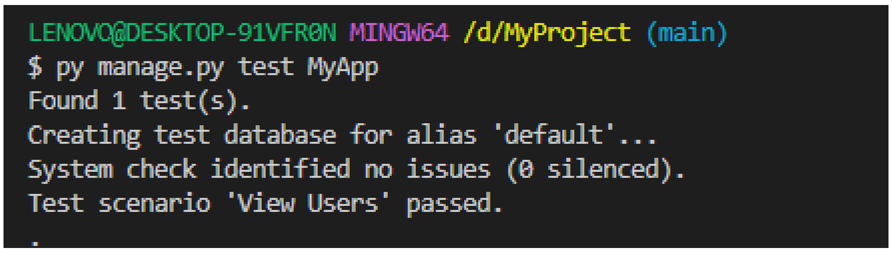
Test Case 2: View Users

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 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 '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	
Search Test Case	
Test Case ID: 2	**Test Designed By: Sudeesh E S**
Test Priority (Low/Medium/High): High	**Test Designed Date: 03/12/2023**
Module Name: User	**Test Executed By: Ms. Nimmy Francis**
Test Title: View Users	**Test Execution Date: 03/12/2023**
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				
5	Click on view users button	librarian	Show all users	users are displayed	Pass

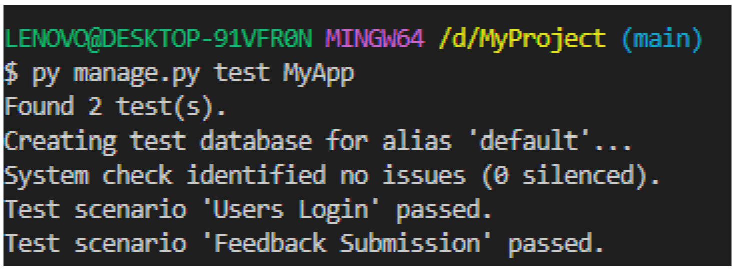
Post-Condition: User is successfully logged into the system and user view is implemented.

Test Case 3: User Feedback

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 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="sudeesh43", password="sushi@123")
        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("user_landing/", self.driver.current_url)
        print("Test scenario 'Users Login' passed.")
        print("Test scenario 'Feedback Submission' passed.")

    def test_feedback_submission(self):
        feedback_url = self.live_server_url + 'http://127.0.0.1:8000/submit_feedback/'
        self.driver.get(feedback_url)

        try:
            WebDriverWait(self.driver, 10).until(
                EC.visibility_of_element_located((By.ID, "feedback_text"))
            )
        except TimeoutException as e:
            print(f"TimeoutException: {e}")
            print(f"Current URL: {self.driver.current_url}")
            print(f"Current Title: {self.driver.title}")
            raise
        try:
            feedback_text_input = self.driver.find_element(By.ID, "feedback_text")
            feedback_text_input.send_keys("This is a test feedback.")
        except NoSuchElementException as e:
            print(f"NoSuchElementException: {e}")
            print(f"Current URL: {self.driver.current_url}")
            print(f"Current Title: {self.driver.title}")
            raise
        print("Test scenario 'Feedback Submission' passed.")

if __name__ == '__main__':
    LiveServerTestCase.main()
```

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 'Feedback Submission' passed.
```

Test report:

Test Case 3	
Project Name: ECHO	
Search Test Case	
Test Case ID: 3	**Test Designed By: Sudeesh E S**
Test Priority (Low/Medium/High): High	**Test Designed Date: 03/12/2023**
Module Name: Submit feedback	**Test Executed By: Ms. Nimmy Francis**

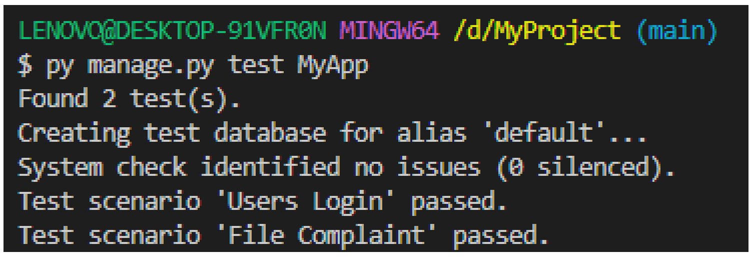
Test Title: Feedback			**Test Execution Date: 03/12/2023**		
Description: User submitting the feedback form					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expect Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Index features should be displayed	Login page displayed	Pass
2	Provide Valid Job Provider email address	User Name: sudeesh43	User should be able to Login	User Logged in and navigated to their corresponding Dashboard	Pass
3	Provide Valid Password	sushi@123			
4	Click on Login button				
5	Entering feedback in textbox	This is a test feedback	Submit the feedback	Feedback submitted successfully	Pass
Post-Condition: User is logged into the system and successfully submitted the feedback form.					

Test Case: 4

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 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="sudeesh43", password="sushi@123")

        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("user_landing/", self.driver.current_url)
        print("Test scenario 'Users Login' passed.")
    def test_file_complaint(self):
        complaint_url = f"{self.live_server_url}/file_complaint/"
        self.driver.get(complaint_url)
        try:
            WebDriverWait(self.driver, 10).until(
                EC.visibility_of_element_located((By.ID, "name"))
            )
        except TimeoutException as e:
            print(f"TimeoutException: {e}")
            print(f"Current URL: {self.driver.current_url}")
            print(f"Current Title: {self.driver.title}")
            raise
        try:
            name_input = self.driver.find_element(By.ID, "name")

            place_input = self.driver.find_element(By.NAME, "place")

            description_input = self.driver.find_element(By.NAME, "Description")

            name_input.send_keys("John Doe")

            place_input.send_keys("Sample Place")

            description_input.send_keys("This is a test complaint.")

            # Submit the form

            submit_button = self.driver.find_element(By.ID, "submitButton")

            submit_button.click()

            # Wait for the success message or redirect

            WebDriverWait(self.driver, 10).until(

                EC.url_matches("/success_page/")  # Adjust the URL or success condition
accordingly

            )

  # Assert success (You can customize this based on your application behavior)

self.assertIn("success", self.driver.current_url)

self.assertEqual("Complaint submitted successfully!", self.driver.find_element(By.ID,
"successMessage").text)

        except NoSuchElementException as e:

            print(f"NoSuchElementException: {e}")

            print(f"Current URL: {self.driver.current_url}")

            print(f"Current Title: {self.driver.title}")

            raise
        print("Test scenario 'File Complaint' passed.")

if __name__ == '__main__':

    LiveServerTestCase.main()
```

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 Report:

Test Case 4					
Project Name: ECHO					
Block User Test Case					
Test Case ID: 4			**Test Designed By: Sudeesh E S**		
Test Priority(Low/Medium/High):High			**Test Designed Date: 03/12/2023**		
Module Name: View Companies			**Test Executed By: Ms. Nimmy Francis**		
Test Title: View Companies			**Test Execution Date: 03/12/2023**		
Description: Jobseeker can view the list of Companies.					
Pre-Condition: Jobseeker 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	Pass
3	Provide Valid Password	Sushi@123			

4	Click on Login button			Dashboard	
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 Companies list		File complaint	Successfully filed complaint	Pass
Post-Condition: User successfully logged into their account and is able to file Complaints.					

CHAPTER 6
IMPLEMENTATION

6.1 INTRODUCTION

The implementation phase of a project is where the design is transformed into a functional system. It is a crucial stage in ensuring the success of the new system, as it requires gaining user confidence that the system will work effectively and accurately. User training and documentation are key concerns during this phase. Conversion may occur concurrently with user training or at a later stage. Implementation involves the conversion of a newly revised system design into an operational system. During this stage, the user department bears the primary workload, experiences the most significant upheaval, and has the most substantial impact on the existing system. Poorly planned or controlled implementation can cause confusion and chaos. Whether the new system is entirely new, replaces an existing manual or automated system, or modifies an existing system, proper implementation is essential to meet the organization's needs. System implementation involves all activities required to convert from the old to the new system. The system can only be implemented after thorough testing is done and found to be working according to specifications. System personnel evaluate the feasibility of the system. Implementation requires extensive effort in three main areas: education and training, system testing, and changeover. The implementation phase involves careful planning, investigating system and constraints, and designing methods to achieve changeover.

6.2 IMPLEMENTATION PROCEDURES

Software implementation is the process of installing the software in its actual environment and ensuring that it satisfies the intended use and operates as expected. In some organizations, the software development project may be commissioned by someone who will not be using the software themselves. During the initial stages, there may be doubts about the software, but it's important to ensure that resistance does not build up. This can be achieved by:

- Ensuring that active users are aware of the benefits of the new system, building their confidence in the software.
- Providing proper guidance to the users so that they are comfortable using the application.

Before viewing the system, users should know that the server program must be running on the server. Without the server object up and running, the intended process will not take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

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

The maintenance phase is a crucial aspect of the software development cycle, as it is the time when the software is actually put to use and performs its intended functions. Proper maintenance is essential to ensure that the system remains functional, reliable, and adaptable to changes in the system environment. Maintenance activities go beyond simply identifying and fixing errors or bugs in the system. It may involve updates to the software, modifications to its functionalities, and enhancements to its performance, among other things. In essence, software maintenance is an ongoing process that requires continuous monitoring, evaluation, and improvement of the system to meet changing user needs and requirements.

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. A pivotal focus for advancement lies in the strategic integration of artificial intelligence (AI) into the platform. 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"
  Authors: Joy I. Eze, Elijah O. Omidiora
  Published in: International Journal of Computer Applications
  Year: 2014Link to

- "Crime Reporting and Criminal Identification Using Online Reporting System"
  Authors: Priya H. Patil, Mr. Aniket S. Potpote
  Published in: International Journal of Advanced Research in Computer Engineering & Technology
  Year: 2014Link to

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

WEBSITES:

- www.chat.openai.com
- www.djangoproject.com
- www.github.com
- www.glassdoor.com
- www.google.com
- www.hackerrank.com
- www.indeed.com
- www.jobvite.com
- www.linkedin.com
- www.monster.com
- www.phind.ch
- www.stackoverflow.com
- www.w3schools.com

CHAPTER 9

APPENDIX

9.1 Sample Code

home.html

```
{% load static  %}
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Echo - Online Crime Reporting</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="{% static 'assets/img/police.png' %}" rel="icon">
  <link href="{% static 'assets/img/apple-touch-icon.png' %}" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link href="{% static
'https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Jost:30
0,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,70
0i' %}" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="{% static 'assets/vendor/aos/aos.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/bootstrap-icons/bootstrap-icons.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/boxicons/css/boxicons.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/glightbox/css/glightbox.min.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/remixicon/remixicon.css' %}" rel="stylesheet">
  <link href="{% static 'assets/vendor/swiper/swiper-bundle.min.css' %}" rel="stylesheet">

  <!-- Template Main CSS File -->
```

```
 <link href="{% static 'assets/css/style.css' %}" rel="stylesheet">

</head>

<body>

 <!-- ======= Header ======= -->
 <header id="header" class="fixed-top ">
  <div class="container d-flex align-items-center">

   <h1 class="logo me-auto"><a href="index.html">ECHO</a></h1>

   <nav id="navbar" class="navbar">
    <ul>
     <li><a class="nav-link scrollto active" href="#hero">Home</a></li>
     <li><a class="nav-link scrollto" href="#about">About</a></li>
     <li><a class="nav-link scrollto" href="register/">Register</a></li>
     <li><a class="nav-link   scrollto" href="login/">Login</a></li>
     <li><a class="nav-link scrollto" href="#team">Team</a></li>
     <li class="dropdown"><a href="#"><span></span> <i class="bi bi-chevron-
down"></i></a>
      <ul>
       <li><a href="#">Drop Down 1</a></li>
       <li class="dropdown"><a href="#"><span>Deep Drop Down</span> <i class="bi bi-
chevron-right"></i></a>
        <ul>
        </ul>
       </li>
       <li><a href="#">Drop Down 2</a></li>
       <li><a href="#">Drop Down 3</a></li>
       <li><a href="#">Drop Down 4</a></li>
      </ul>
     </li>
     <li><a class="nav-link scrollto" href="#contact">Contact</a></li>
```

```
          <li><a class="getstarted scrollto" href="#about">Get Started</a></li>
        </ul>
        <i class="bi bi-list mobile-nav-toggle"></i>
      </nav><!-- .navbar -->

    </div>
  </header><!-- End Header -->

  <!-- ======= Hero Section ======= -->
  <section id="hero" class="d-flex align-items-center">

    <div class="container">
      <div class="row">
        <div class="col-lg-6 d-flex flex-column justify-content-center pt-4 pt-lg-0 order-2 order-lg-
1" data-aos="fade-up" data-aos-delay="200">
          <h1>Kottayam District Police</h1>
          <h2>Ensure safety and reduce disorders,Reduce crime and the fear of crimeContribute to
delivery of justice, which secures and maintains public confidence in the rule of law</h2>
          <div class="d-flex justify-content-center justify-content-lg-start">
            <a href="#about" class="btn-get-started scrollto">Get Started</a>
            <a href="{% static 'https://www.youtube.com/watch?v=jDDaplaOz7Q' %}"
class="glightbox btn-watch-video"><i class="bi bi-play-circle"></i><span>Watch
Video</span></a>
          </div>
        </div>
        <div class="col-lg-6 order-1 order-lg-2 hero-img" data-aos="zoom-in" data-aos-
delay="200">
          <img src="{% static 'assets/img/logo_kerala.png' %}" class="img-fluid animated" alt="">
        </div>
      </div>
    </div>

  </section><!-- End Hero -->
```

```
<main id="main">

  <!-- ======= Clients Section ======= -->
  <section id="clients" class="clients section-bg">
    <div class="container">

      <div class="row" data-aos="zoom-in">

        <div class="col-lg-2 col-md-4 col-6 d-flex align-items-center justify-content-center">
          <img src="{% static 'assets/img/clients/client-1.png' %}" class="img-fluid" alt="">
        </div>

        <div class="col-lg-2 col-md-4 col-6 d-flex align-items-center justify-content-center">
          <img src="{% static 'assets/img/clients/client-2.png' %}" class="img-fluid" alt="">
        </div>

        <div class="col-lg-2 col-md-4 col-6 d-flex align-items-center justify-content-center">
          <img src="{% static 'assets/img/clients/client-3.png' %}" class="img-fluid" alt="">
        </div>

        <div class="col-lg-2 col-md-4 col-6 d-flex align-items-center justify-content-center">
          <img src="{% static 'assets/img/clients/client-4.png' %}" class="img-fluid" alt="">
        </div>

        <div class="col-lg-2 col-md-4 col-6 d-flex align-items-center justify-content-center">
          <img src="{% static 'assets/img/clients/client-5.png' %}" class="img-fluid" alt="">
        </div>

        <div class="col-lg-2 col-md-4 col-6 d-flex align-items-center justify-content-center">
          <img src="{% static 'assets/img/clients/client-6.png' %}" class="img-fluid" alt="">
        </div>

      </div>
```

```
   </div>
  </section><!-- End Cliens Section -->

  <!-- ======= About Us Section ======= -->
  <section id="about" class="about">
   <div class="container" data-aos="fade-up">

    <div class="section-title">
     <h2>About Us</h2>
    </div>

    <div class="row content">
     <div class="col-lg-6">
      <p>
       Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore
       magna aliqua.
      </p>
      <ul>
       <li><i class="ri-check-double-line"></i> Ullamco laboris nisi ut aliquip ex ea commodo
consequat</li>
       <li><i class="ri-check-double-line"></i> Duis aute irure dolor in reprehenderit in
voluptate velit</li>
       <li><i class="ri-check-double-line"></i> Ullamco laboris nisi ut aliquip ex ea commodo
consequat</li>
      </ul>
     </div>
     <div class="col-lg-6 pt-4 pt-lg-0">
      <p>
       culpa qui officia deserunt mollit anim id est laborum.
      </p>
      <a href="#" class="btn-learn-more">Learn More</a>
     </div>
    </div>
   </div>
  </section><!-- End About Us Section -->
  <section id="why-us" class="why-us section-bg">
   <div class="container-fluid" data-aos="fade-up">

    <div class="row">

     <div class="col-lg-7 d-flex flex-column justify-content-center align-items-stretch  order-2
order-lg-1">

      <div class="content">
       <h3>Eum ipsam laborum deleniti <strong>velit pariatur architecto aut
nihil</strong></h3>
       <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Duis aute irure dolor in reprehenderit
       </p>
      </div>

      <div class="accordion-list">
       <ul>
        <li>
         <a data-bs-toggle="collapse" class="collapse" data-bs-target="#accordion-list-
1"><span>01</span> Non consectetur a erat nam at lectus urna duis? <i class="bx bx-chevron-
down icon-show"></i><i class="bx bx-chevron-up icon-close"></i></a>
         <div id="accordion-list-1" class="collapse show" data-bs-parent=".accordion-list">
          <p>
           Feugiat pretium nibh ipsum consequat. Tempus iaculis urna id volutpat lacus laoreet
non curabitur gravida. Venenatis lectus magna fringilla urna porttitor rhoncus dolor purus non.
          </p>
         </div>
        </li>
```

```
                <li>
                  <a data-bs-toggle="collapse" data-bs-target="#accordion-list-2"
class="collapsed"><span>02</span> Feugiat scelerisque varius morbi enim nunc? <i class="bx
bx-chevron-down icon-show"></i><i class="bx bx-chevron-up icon-close"></i></a>
                  <div id="accordion-list-2" class="collapse" data-bs-parent=".accordion-list">
                    <p>
                      Dolor sit amet consectetur adipiscing elit pellentesque habitant morbi. Id interdum
velit laoreet id donec ultrices. Fringilla phasellus faucibus scelerisque eleifend donec pretium. Est
pellentesque elit ullamcorper dignissim. Mauris ultrices eros in cursus turpis massa tincidunt dui.
                    </p>
                  </div>
                </li>

                <li>
                  <a data-bs-toggle="collapse" data-bs-target="#accordion-list-3"
class="collapsed"><span>03</span> Dolor sit amet consectetur adipiscing elit? <i class="bx bx-
chevron-down icon-show"></i><i class="bx bx-chevron-up icon-close"></i></a>
                  <div id="accordion-list-3" class="collapse" data-bs-parent=".accordion-list">
                    <p>Eleifend mi in nulla posuere sollicitudin aliquam ultrices sagittis orci. Faucibus
pulvinar elementum integer enim. Sem nulla pharetra diam sit amet nisl suscipit. Rutrum tellus
pellentesque eu tincidunt. Lectus urna duis convallis convallis tellus. Urna molestie at elementum
eu facilisis sed odio morbi quis
                    </p>
                  </div>
                </li>
              </ul>
            </div>
          </div>
          <div class="col-lg-5 align-items-stretch order-1 order-lg-2 img" style='background-image:
url("assets/img/why-us.png");' data-aos="zoom-in" data-aos-delay="150">&nbsp;</div>
        </div>
      </div>
    </section><!-- End Why Us Section -->
<div class="container footer-bottom clearfix">
```

```
      <div class="copyright">
        &copy; Copyright <strong><span>Arsha</span></strong>. All Rights Reserved
      </div>
      <div class="credits">
        Designed by <a href="{%static 'https://bootstrapmade.com/' %}">BootstrapMade</a>
      </div>
    </div>
  </footer><!-- End Footer -->

  <div id="preloader"></div>
  <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-
arrow-up-short"></i></a>
  <script src="{% static 'assets/vendor/aos/aos.js'%}"></script>
  <script src="{% static 'assets/vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>
  <script src="{% static 'assets/vendor/glightbox/js/glightbox.min.js' %}"></script>
  <script src="{% static 'assets/vendor/isotope-layout/isotope.pkgd.min.js' %}"></script>
  <script src="{% static 'assets/vendor/swiper/swiper-bundle.min.js' %}"></script>
  <script src="{% static 'assets/vendor/waypoints/noframework.waypoints.js' %}"></script>
  <script src="{% static 'assets/vendor/php-email-form/validate.js' %}"></script>
  <script src="{% static 'assets/js/main.js' %}"></script>
</body>
</html>
```

login.html

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <link rel="stylesheet" type="text/css" href="{% static 'login.css' %}">
</head>
<body>
    <div class="login-container">
```

```
        <h1>Login</h1>
        {% if error_message %}
        <p class="error-message">{{ error_message }}</p>
        {% endif %}
        {% if not request.user.is_authenticated %}
        {% comment %} <script>
            // Show an alert box when the user is logged out
            window.onload = function() {
                alert('You have been successfully logged out.');
            };
        </script> {% endcomment %}
        {% endif %}
        <form method="post">
            {% csrf_token %}
            <input type="text" placeholder="username" name="username" id="username" required>
            <input type="password" placeholder="password" id="password" name="password"
required>
            <input type="submit" id="testid" value="Login"><br>
            <span><center>Forget Password? <a href="{% url 'password_reset'
%}">Reset</a><center></span>
            <p><h5><center>Not a User?<a href="{% url 'register' %}">Register
Here</a><center><h5></p>
        </form>
    </div>
</body>
</html>
```

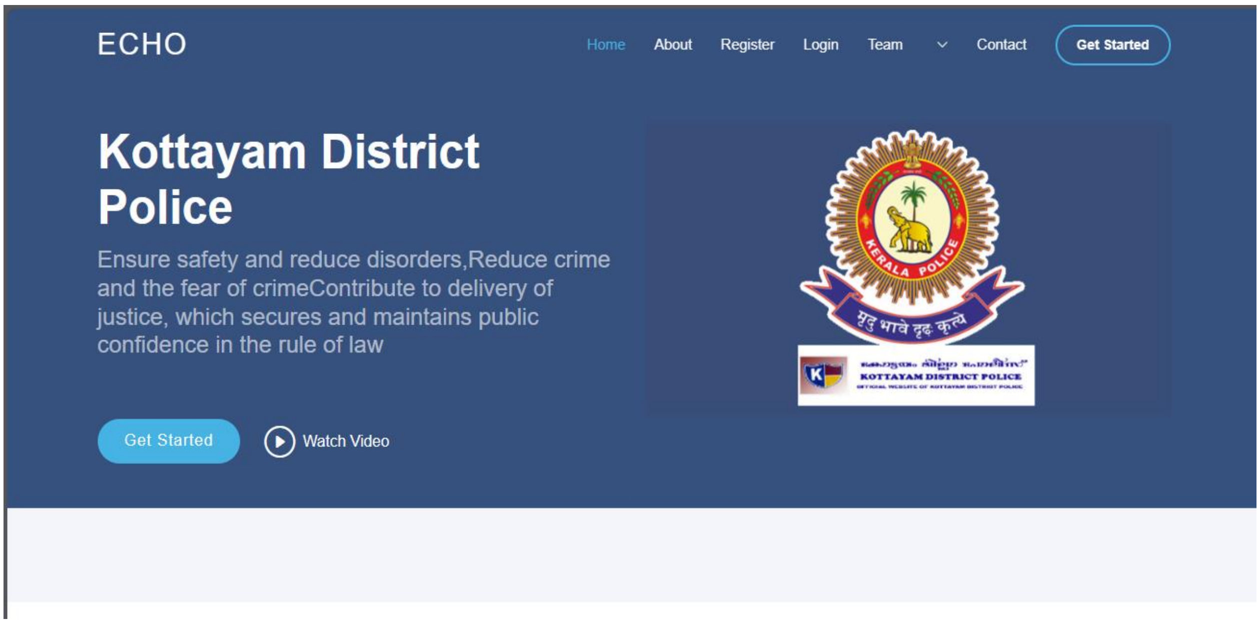
filecomplaint.html

```
{% load static %}
<html lang="en">
<head>
  <!-- {% block title %}Report{% endblock title %} -->
</head>
<div class="login-box">
```

```
   <form id="locform" method="post" action="{% url 'file_complaint' %}"
enctype="multipart/form-data">
      {% csrf_token %}
      <div class="demo-wrap">
         <img class="demo-bg" src="assets/img/logo_kerala.png" alt="">
      </div>
   <h3>Report Complaint</h3>
   <form id="complaintForm">
      <div class="form-label" style="font-size: 75%;">
         <p><strong>Name of victim(s)</strong>: <br />
            <input id="name" placeholder="Name" name="name" style="font-size: 90%;"
type="text" required>
         </p>
         <p><strong>Place of the incident</strong>: <br />
            <input name="place" style="font-size: 90%;" type="text" placeholder="Enter address"
required>
         </p>

         <p><strong>Describe the incident</strong>: <br />
            <textarea name="Description" rows="12" style="width: 95%; font-size:
90%;"></textarea>
         </p>

         <p><strong>Evidence:</strong> <br />
            <select name="PhysicalEvidence" id="physicalEvidence" style="font-size: 90%;"
onchange="togglePhotoUpload()">
               <option value="yes">Yes</option>
               <option value="no">No</option>
            </select>
         </p>

         <p id="photoUpload" style="display: none;"><strong>Upload a photo of the
incident:</strong> <br> <br>
            <input name="fileUpload" type="file" accept="image/png, image/jpeg, image/jpg">
            </p>
        </div>
        <p id="message"></p>
        <!-- Move style of button into stylesheet, once the button is nailed down... -->
        <p style="text-align: center;">
            <button id="submitButton" name="submitButton" type="submit">
                Submit This Report
            </button>
        </p>
    </form>
</div>

<script>
    document.addEventListener("DOMContentLoaded", function () {
        var physicalEvidenceDropdown = document.getElementById("physicalEvidence");
        var photoUpload = document.getElementById("photoUpload");

        physicalEvidenceDropdown.value = "no";

        if (physicalEvidenceDropdown.value === "no") {
            photoUpload.style.display = "none";
        }

        function togglePhotoUpload() {
            if (physicalEvidenceDropdown.value === "yes") {
                photoUpload.style.display = "block";
            } else {
                photoUpload.style.display = "none";
            }
        }
        physicalEvidenceDropdown.addEventListener("change", togglePhotoUpload);
        togglePhotoUpload();
    });
    document.addEventListener("DOMContentLoaded", function () {
        var successMessage = document.getElementById("successMessage");
        document.getElementById("complaintForm").addEventListener("submit", function (event) {
            event.preventDefault();
            successMessage.style.display = "block";
        });
    });
</script>
</body>
</html>
```

9.1 Screen Shots

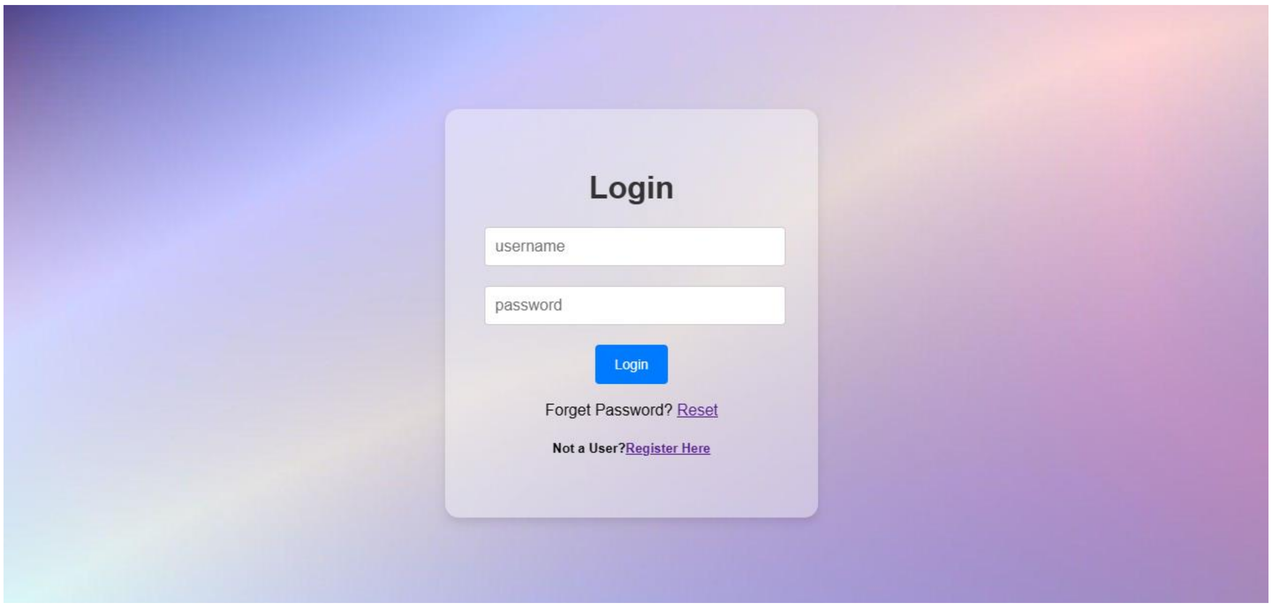
Home



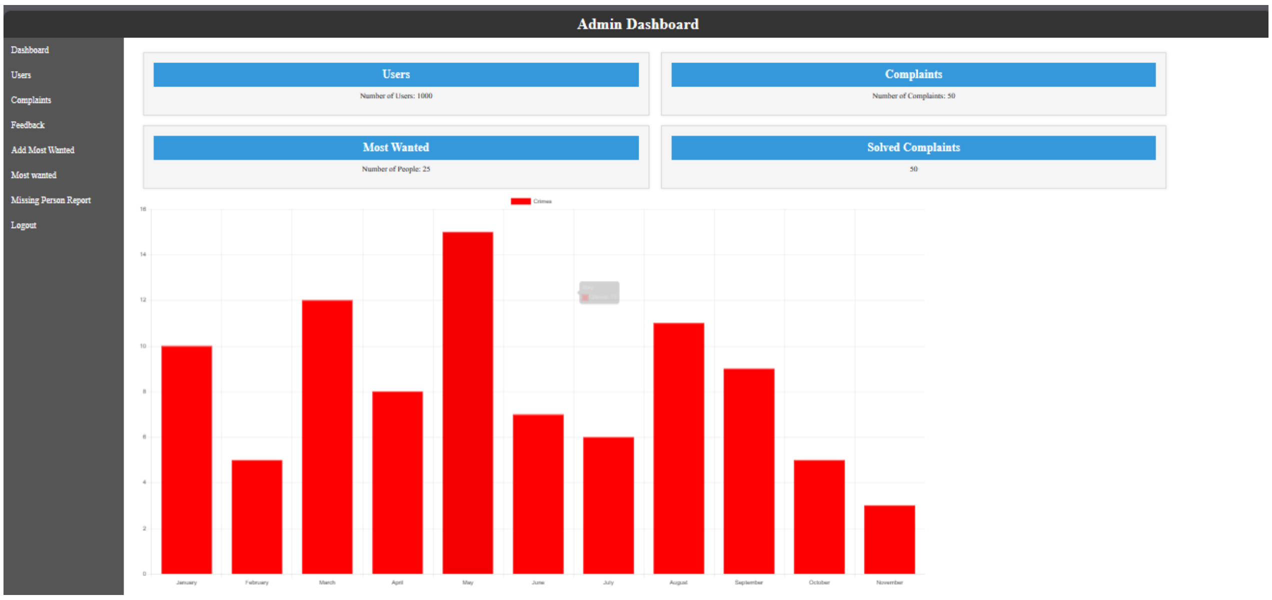
Kottayam District Police

Ensure safety and reduce disorders,Reduce crime and the fear of crimeContribute to delivery of justice, which secures and maintains public confidence in the rule of law

Login page



Admin Dashboard


Admin Dashboard

Users

Number of Users: 1000

Complaints

Number of Complaints: 50

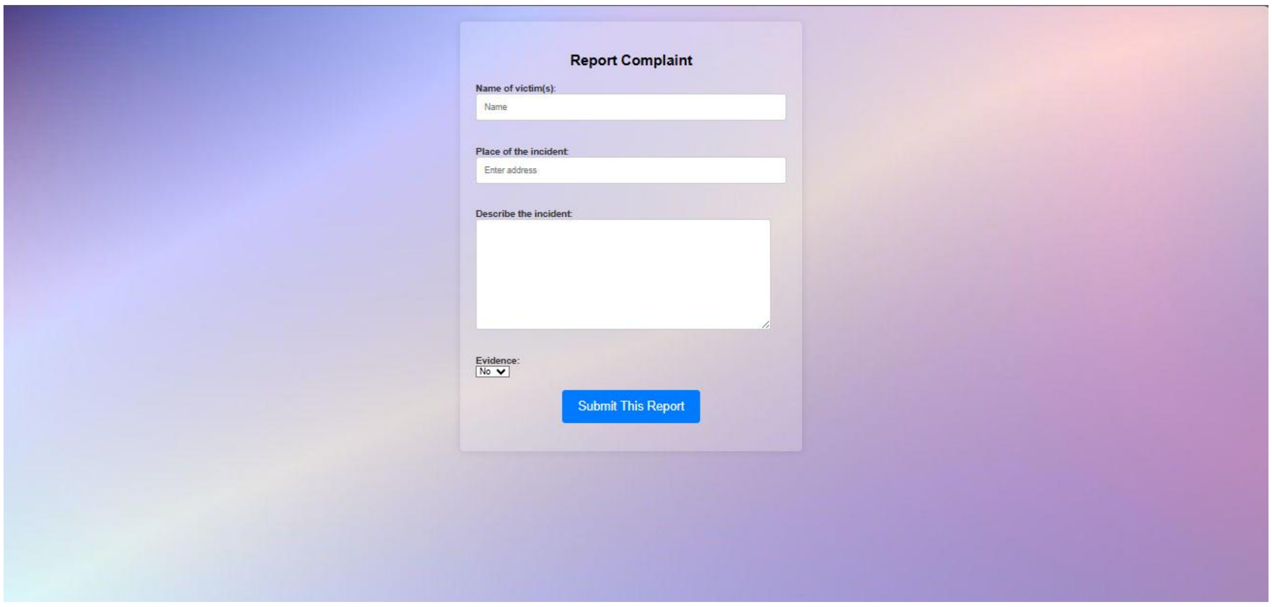
Most Wanted

Number of People: 25

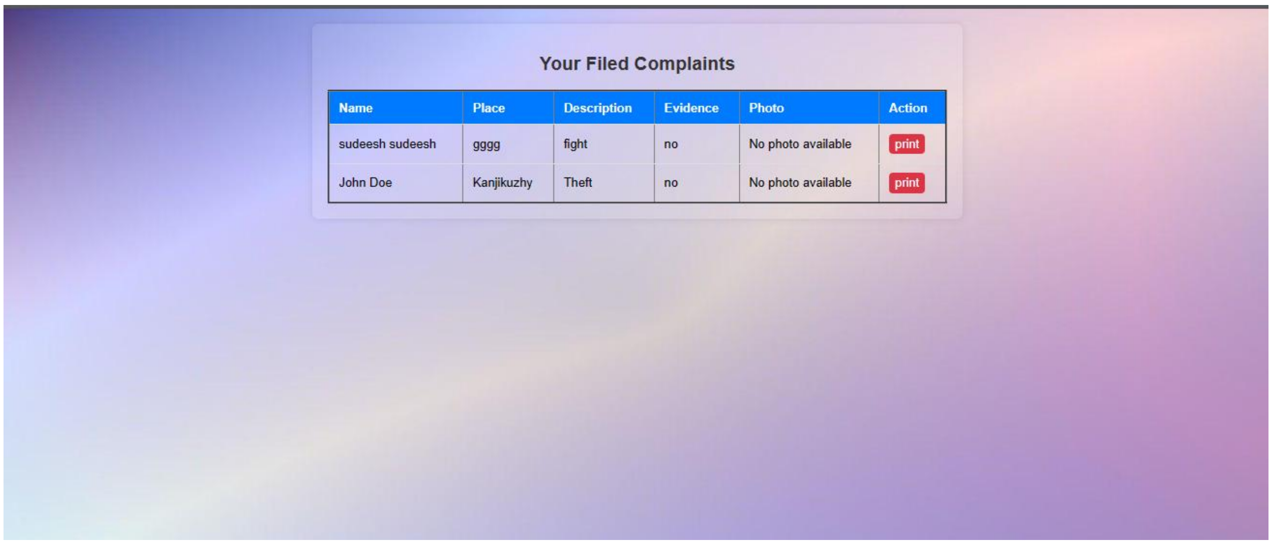
Solved Complaints

50

File Complaint



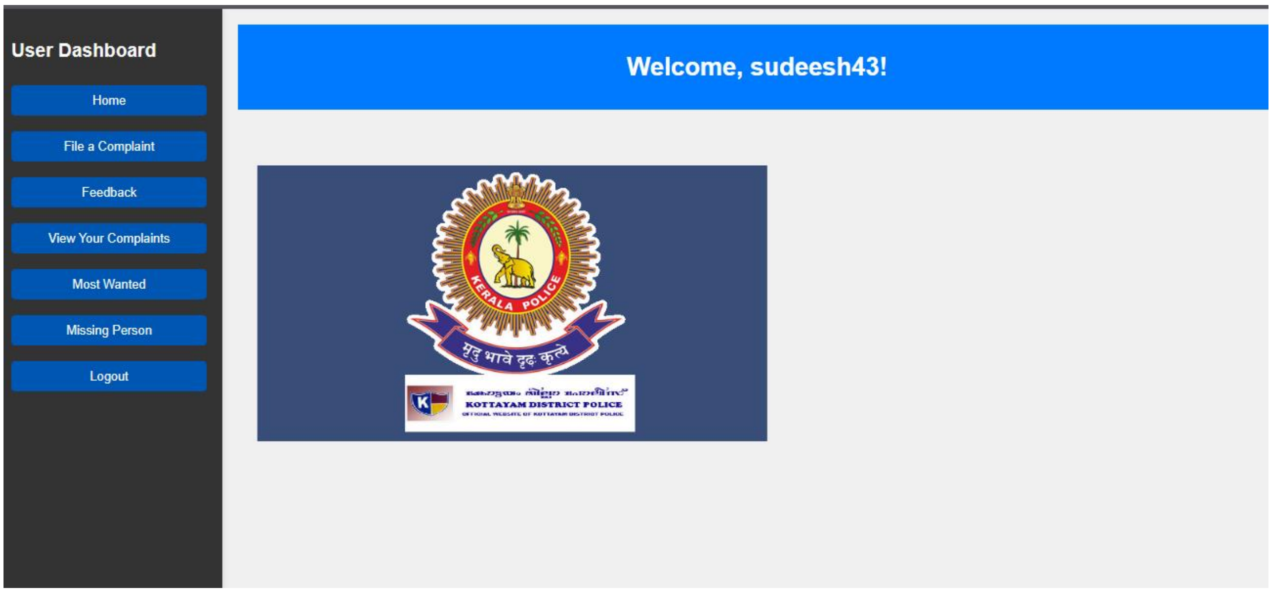
View Complaints



Your Filed Complaints

Name	Place	Description	Evidence	Photo	Action
sudeesh sudeesh	gggg	fight	no	No photo available	print
John Doe	Kanjikuzhy	Theft	no	No photo available	print

User dashboard


User Dashboard
Home
File a Complaint
Feedback
View Your Complaints
Most Wanted
Missing Person
Logout
Welcome, sudeesh43!
KERALA POLICE
मृदु भावे दृढ कृत्ये
KOTTAYAM DISTRICT POLICE