# CHAPTER 1
# INTRODUCTION

## 1.1 GENERAL

In modern society, crime has become an increasingly prevalent concern that affects individuals, communities, and the overall functioning of governance systems. The rise in criminal activities has necessitated the development of efficient systems for tracking, managing, and analyzing crime-related data. The traditional manual methods of filing complaints, recording FIRs, and maintaining criminal records are prone to errors, inefficiency, and lack of transparency. In response to these challenges, there is a growing need for automated systems that can aid law enforcement agencies in better managing crimes and criminal records.

The **Crime Management System (CMS)** is a software-based solution developed using PHP that allows users to register crime complaints, view the status of investigations, and helps police officers and administrators manage case records efficiently. This system reduces paperwork, increases transparency, and ensures that citizens and law enforcement are connected through a digital platform. The CMS provides a centralized system where crime records are stored in a secure manner using files, as per the requirements of this small-scale project.

The system provides features like complaint registration, criminal record management, and status updates, ensuring that information is easily accessible and well-organized. Citizens can file complaints online without visiting a police station, while police officers can update the investigation status, mark complaints as resolved, and manage the history of cases directly through the portal.

## 1.2NEED FOR THE STUDY

The traditional method of recording and managing criminal data involves heavy manual effort, time-consuming procedures, and susceptibility to misplacement of records. This not only delays the investigation process but also reduces the efficiency of law enforcement agencies. In addition, citizens often face difficulties while lodging complaints due to bureaucratic hurdles and fear of harassment. Hence, there is a dire need for an online crime management solution that ensures transparency, accessibility, and security.

The development of a Crime Management System addresses several important challenges:

- Eliminates the need for physical record-keeping.
- Reduces human error and data duplication.
- Provides real-time access to case information for officers and administrators.
- Offers a platform for citizens to lodge complaints safely and securely.
- Facilitates better communication between citizens and authorities.
- Empowers police departments with a structured and easy-to-use tool to monitor criminal cases.
- Serves as a foundation for data analytics on criminal activities in the long run.

With these advantages in mind, the study and development of the CMS becomes a critical task that not only benefits law enforcement agencies but also contributes to safer communities.

## 1.3 OBJECTIVES OF THE STUDY

The primary objective of this study is to develop a web-based Crime Management System using PHP, which will serve as a digital platform for managing crimes and complaints. This system is intended to simplify and streamline the criminal complaint process for both users and authorities.

The specific objectives of the project are:

- To provide a platform for users (citizens) to register complaints online.
- To allow police officials to manage, view, and update crime cases.
- To store all data in a secure file-based system instead of a database for project simplicity.
- To create a structured, user-friendly interface that is easy to navigate.
- To reduce the paperwork and time consumed in traditional complaint procedures.
- To improve transparency and promote trust in law enforcement processes.
- To generate reports and summaries that help in analyzing crime trends.
- To demonstrate the use of web technologies in developing real-time applications with practical social relevance.

This project also aims to give students hands-on experience in handling real-world problems with software solutions and improve their skills in PHP-based development.

## 1.4 OVERVIEW OF THE PROJECT:

The Crime Management System is a compact PHP-based application that allows citizens to register crime complaints and enables the administration or police officials to manage and track the status of these complaints. The project follows a modular design that separates user functionalities and admin/police functionalities. The system stores all records in text files rather than a traditional database to align with the project's simplicity goals.

The system offers two main panels:

1. User Panel: Allows citizens to register themselves, log in, and file a complaint. Users can track the status of their complaints and view any updates made by the police.
2. Admin/Police Panel: Enables police officers or administrators to view all complaints, update investigation statuses, and generate summary reports of criminal activities.

Key modules and functionalities include:

- User registration and login.
- Complaint registration form.
- View complaint status.
- Admin login and dashboard.
- Case management (view, update, resolve).
- Record generation and export to file.

Security, ease of access, and effective record management form the core of this project. The project avoids the complexity of using a database and instead uses file-based storage systems (like .txt files) to store complaint data, user information, and case updates.

The Crime Management System not only simplifies the traditional system of filing and tracking crime cases but also promotes a transparent and responsive policing model. It has immense potential to be scaled into a more advanced system in the future with integrations like database management, data analytics, and mobile application support.

# CHAPTER 2
# REVIEW OF LITERATURE

## 2.1 INTRODUCTION

A literature review is an essential component of any academic or technical project. It helps to understand the existing systems, technologies, and methodologies that are already in place and gives direction for future improvements. The review of literature for this project focuses on existing crime reporting and management systems and the methods used in their development, management, and deployment. It also considers the shortcomings of these systems, which helped in formulating the objectives for developing an improved and simplified version for small-scale use.

Over the years, various crime tracking and complaint management systems have been developed, ranging from simple web portals to AI-based predictive policing tools. Most of these solutions, however, are built on large-scale infrastructure involving complex database management systems, integration with national records, and multi-level administrative access. While such systems serve large government and law enforcement agencies, they are often too complex for small institutions or educational use cases.

This project, therefore, aims to develop a simplified file-based Crime Management System using PHP that allows users to report crimes and track their progress, while enabling police and administrators to update and manage crime records. This solution is targeted toward small institutions, communities, or academic demonstrations where a lightweight and easy-to-deploy system is required.

Existing research in the domain indicates that centralized and digital crime record systems improve the speed of investigation, data retrieval, and accuracy. Projects like e-FIR portals and police CMS platforms have shown significant improvements in public participation and administrative efficiency.

Some previously implemented projects related to crime management include:

- Online FIR systems developed by various state governments in India.
- Mobile apps for citizen-police communication (e.g., Delhi Police's "Himmat"

app).

- Crime mapping systems used by police departments in the USA.
- Predictive crime analytics using machine learning in advanced law enforcement tools.

Although these systems are sophisticated and effective, they are not always suitable for simple applications. This literature review helps identify the gap for a file-based, easy-to-understand version of such systems that can be used in academic settings or low-resource environments.

## 2.2 FRAMEWORK OF LCA

**FRAMEWORK OF LIFE CYCLE APPROACH (LCA)**

The Life Cycle Approach (LCA) is a structured methodology used to guide the development of any software system, including web-based crime management applications. It ensures that the system is developed in well-defined phases, making the process manageable, traceable, and measurable.

For the **Crime Management System**, the software development follows a simplified **Software Development Life Cycle (SDLC)** framework consisting of the following phases:

**1. Requirement Analysis**

In this phase, the key requirements of the system were gathered. For this project, the requirement was to create a small web-based crime reporting and tracking system using PHP with file-based storage, without using any database. Requirements from different user roles (Admin, Police, User) were also identified.

**2. System Design**

The structure of the system was designed, including the layout of files, modules, and user interfaces. The design includes login systems, role-based access, forms for complaint submission, and dashboards. JSON or .txt file formats are used for data storage.

**3. Development**

In this phase, the actual coding was done using PHP, HTML, and basic Bootstrap for styling. Different scripts were developed for user login, file upload, complaint processing, and report generation.

**4. Testing**

Testing was carried out to ensure all modules work correctly, including login validation, complaint registration, role-based views, and status updates. Manual test cases were used for each role to simulate real-time usage.

**5. Deployment**

The system is deployed on a local server (e.g., XAMPP). No external dependencies such as databases or APIs are required, making it simple to deploy on any local machine.

**6. Maintenance**

Though it is a small-scale project, maintenance includes updating files, adding new functionalities, and correcting bugs. Since it's file-based, the system remains easy to maintain for demonstration purposes.

**Benefits of Life Cycle Approach in this Project:**

- Ensures every step is documented and traceable.
- Reduces errors by implementing step-by-step planning.
- Helps in identifying bugs early during development and testing.
- Supports easy enhancements or modifications in future.

# CHAPTER 3
# SYSTEM OVERVIEW

## 3.1 EXISTING SYSTEM

Crime management in many places, especially in developing or under-resourced regions, still heavily relies on traditional systems of reporting and record-keeping. In many jurisdictions, citizens are required to physically visit a police station to report a crime, fill out a form manually, and follow up through in-person visits or phone calls. This process is not only time-consuming but also inefficient and prone to human error and corruption.

In terms of digital systems, many countries have implemented large-scale Crime Management Systems (CMS), but they are often complex and require integration with centralized government databases and infrastructure. For example, in India, there is the **Crime and Criminal Tracking Network and Systems (CCTNS)** project initiated by the National Crime Records Bureau (NCRB), which digitizes crime records across all police stations. While this is a significant step toward modernization, such systems have certain limitations in practice:

**Limitations of the Existing System:**

1. **Complexity and Technical Infrastructure:** Most government-backed crime management systems require a centralized database, high-speed internet, authentication mechanisms (Aadhaar, etc.), and government-issued credentials. These are not always accessible in rural areas or small towns.
2. **Lack of Transparency:** Victims often have no way to check the status of their complaints. Once an FIR is registered, the system becomes opaque to the complainant.
3. **Inaccessibility for General Public:** Older adults, rural citizens, or less tech-savvy people may struggle to use existing portals that are often poorly designed or hard to navigate.
4. **Manual Intervention and Delays:** The manual process of writing reports, approving them, and forwarding them through various departments can take a

long time, causing delays in justice.

5. **No Role Segregation in Simple Systems:** Many small-scale web systems do not separate access between the user, police, and admin, creating confusion and security risks.
6. **Maintenance Overhead:** Centralized systems require continuous server maintenance, database optimization, and IT support which may not be feasible for smaller organizations or demonstration projects.

Thus, the need for a lightweight, flexible, and easily deployable system becomes evident, especially for small-scale use in educational projects, rural setups, or simulations.

## 3.2 PROPSED SYSTEM

The **Crime Management System** proposed in this project is a simplified, file-based solution developed using PHP. It provides the core functionalities of a crime reporting and tracking system without requiring complex databases or cloud infrastructure.

This system is designed for **local use** (e.g., a college demonstration, community management system, or a simple law enforcement simulation) where user data, complaints, and crime records can be saved directly into files (such as .txt, .json, or .csv) stored on the server itself.

**Key Features of the Proposed System:**

1. **Role-Based Access:**
   - **Admin:** Can manage users, view all records, assign cases.
   - **Police:** Can view assigned cases, update statuses, and close reports.
   - **User:** Can file complaints, view complaint status, and track progress.
2. **File-Based Data Storage:**
   - No external database is required.
   - Data is stored in structured formats in folders (easy to manage and backup).
   - Reduces complexity and enhances portability of the project.
3. **Complaint Registration Module:**
   - Simple form for users to register complaints.
   - Auto-generated complaint ID for tracking.
4. **Status Tracking:**

    - Users can check the real-time status of their complaints.
5. **Security and Authentication:**
    - Basic login system for each role.
    - File-level permission control using PHP session management.
6. **Lightweight Deployment:**
    - Can run on XAMPP or any local PHP server.
    - No internet or third-party service dependency.
7. **Scalability (Optional):**
    - Though it's a small project, it can later be expanded to integrate databases or APIs for SMS alerts, face recognition for suspect records, etc.

**Advantages over Existing System:**

- Easy to understand and modify.
- No special hardware or hosting required.
- Promotes digital complaint handling even in offline/local setups.
- Suitable for small institutions, mock setups, or academic purposes.
- Encourages transparency by letting users track their complaint status.

## 3.3 FEASIBILITY STUDY

A feasibility study evaluates the practicality and viability of implementing the proposed system. This involves analyzing the technical, economic, operational, and schedule-related aspects to ensure the successful delivery of the project.

### 1. Technical Feasibility:

This system is technically feasible because it uses simple, open-source tools:

- Technology Stack: PHP, HTML, CSS, JavaScript (optional).
- Server: XAMPP/WAMP (runs on Windows/Linux/Mac).
- No Database Required: All data is stored in plain files or JSON format.
- Skills Required: Basic knowledge of PHP is sufficient to build and maintain the project.

Since the project does not require advanced hardware or third-party APIs, it is highly feasible even in low-resource environments.

**2. Economic Feasibility:**

This system is economically feasible as it does not require any monetary investment:

- Software Costs: All tools used (PHP, XAMPP) are open-source and free.
- No Hosting Cost: Can run locally.
- Development Cost: Minimal if done by students or in-house developers.
- Maintenance Cost: Very low, as file-based systems require little to no backend maintenance.

This makes the system perfect for students, academic demonstration, and non-profit community projects.

**3. Operational Feasibility:**

The system is operationally feasible because:

- It has a user-friendly interface that can be used by both citizens and police personnel with minimal training.
- Login and complaint tracking modules are straightforward.
- Since the roles and functionalities are clearly separated, it reduces the chance of operational errors.
- The admin has full control over user management, giving an extra layer of administration.

This means the system can be used effectively by non-technical users as well.

**4. Schedule Feasibility:**

The system was designed to be completed within a short project cycle, typically 3–4 weeks for a basic working model. The schedule breakdown is as follows:

- Week 1: Requirement gathering and planning
- Week 2: Designing UI and creating file storage structure
- Week 3: Coding major modules (login, complaint, status)
- Week 4: Testing and documentation

# CHAPTER 4
# SYSTEM REQUIREMENTS

## 4.1 HARDWARE REQUIREMENTS

The Crime Management System is a lightweight, file-based web application built using PHP. It does not demand high-end hardware, making it ideal for small-scale environments such as colleges, local police stations, or administrative offices. However, certain hardware specifications are necessary to ensure that the application runs smoothly during development, testing, and actual use.

While the system will run locally using XAMPP or any local server setup, it must be supported by a computer capable of handling the required tasks like processing PHP files, running a local server, handling file read/write operations, and displaying a browser-based interface.

**Minimum Hardware Requirements:**

- **Processor:** Dual-Core Processor (Intel or AMD)
  - Ensures basic operations and execution of PHP scripts
- **RAM:** Minimum 4 GB
  - Adequate for running XAMPP server, browser, and code editor simultaneously
- **Hard Disk:** At least 500 MB of free space
  - To store PHP files, user data, reports, and logs
- **Monitor:** 1024x768 resolution or higher
  - Required for proper display of the user interface
- **Input Devices:** Keyboard and mouse
  - To facilitate interaction with the forms and dashboard
- **Optional:** Printer
  - For printing complaint reports or admin logs

**Recommended Hardware for Better Performance:**

- **Processor:** Quad-Core Processor or better
- **RAM:** 8 GB or higher for smooth multitasking
- **Storage:** Solid State Drive (SSD) for faster file access and saving

- **Monitor:** 1366x768 or 1920x1080 resolution for full view of dashboard and forms

**Networking (Optional):**

- LAN card or Wi-Fi module (only needed for testing over local area network)
- No internet connection is required since the system runs offline

## 4.2 SOFTWARE REQUIREMENTS

The software environment plays a critical role in building and running the Crime Management System. Since this is a small-scale project based on PHP and without the use of any database, the required software components are minimal, freely available, and easy to install.

The system is developed using web technologies such as PHP, HTML, and CSS, and all PHP scripts are executed via a local server like XAMPP. It runs through a browser interface and stores data in simple files (e.g., text files or JSON), avoiding the complexity of SQL databases.

**Essential Software Components:**

- **XAMPP Server:**
  - Bundled tool that includes:
    - Apache (Web Server)
    - PHP Interpreter
    - (Optional) MySQL (Not used here)
  - Allows local hosting of PHP applications
- **Code Editor or IDE:**
  - Examples:
    - Visual Studio Code (preferred for its features)
    - Notepad++

        - Sublime Text
    - Used for writing and editing PHP, HTML, and CSS files
- **Web Browser:**
    - Required to access and test the application
    - Supported browsers:
        - Google Chrome (recommended)
        - Mozilla Firefox
        - Microsoft Edge

**Languages and Tools Used:**

- **PHP:**
    - Server-side scripting language
    - Executes complaint registration, file handling, and login authentication
- **HTML:**
    - Used to structure the web pages and forms
- **CSS:**
    - Adds styling to the UI elements (buttons, forms, text)
- **JavaScript (optional):**
    - For validation of input fields or dynamic page behavior

**Operating System Compatibility:**

- **Windows OS (Windows 7, 8, 10, or 11)**
- **Linux (Ubuntu/Fedora)**
- **macOS** (with MAMP instead of XAMPP)

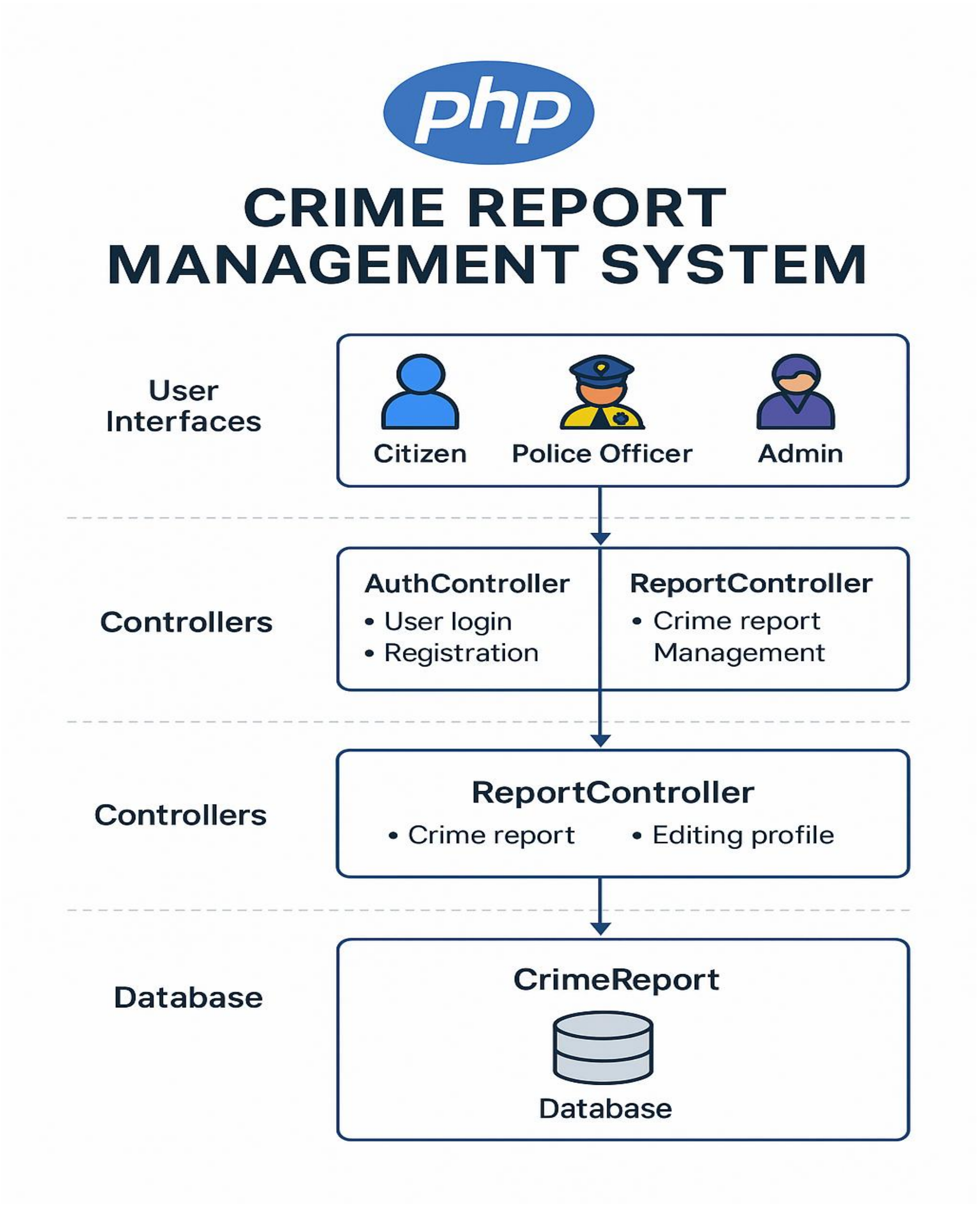
**Other Tools (Optional but Useful):**

- **Browser Developer Tools:**
    - For debugging layout, inspecting elements, and checking console logs
- **PDF Printer or Export Plugin:**
    - To convert user complaints or reports into PDFs if needed

# CHAPTER 5

# SYSTEM DESIGN

## 5.1 SYSTEM ARCHITECTURE



System architecture refers to the high-level design of a software system that outlines the structure and interactions of various components. For the Crime Management System, the architecture follows a three-tier structure: Presentation Layer, Logic Layer, and Data Layer. Even though we are not using a formal database, we simulate the data layer using structured files.

This design is simple, scalable, and ideal for small-scale, local environments like academic projects or basic police station software.

**Three-Tier Architecture Overview:**

**1. Presentation Layer (User Interface Layer):**

This is the front-end interface where users interact with the system. It includes all the HTML forms used for:

- Registering crime complaints
- Logging in as a user or admin
- Viewing case status
- Managing complaints (admin panel)

The presentation layer is responsible for:

- Taking input from users (complainants, admins)
- Validating input fields using basic client-side validation
- Displaying responses and status

**2. Logic Layer (Application Logic):**

This is where the PHP scripts are executed. It acts as the bridge between the interface and data layer.

Responsibilities include:

- Authenticating user and admin credentials
- Processing new complaint forms
- Saving data to text files
- Reading complaints and generating status views
- Admin functions like marking complaints as "In Progress", "Resolved", etc.

All business logic (conditions, rules, and operations) is handled in this layer.

**3. Data Layer (File Storage Layer):**

Instead of using databases, we utilize files stored locally to manage data. These could be:

- .txt or .json files storing complaint information
- Log files maintaining activity records
- Files storing user credentials (in a simple format for the project)

Even though this is a simplified version of a traditional DBMS, this layer handles data persistence and retrieval.

**Architectural Flow:**

1. User accesses the website and submits a complaint form.
2. The form data is validated and passed to a PHP script.
3. The PHP script stores the information in a .txt file.

4. Admin logs in through another interface, reads the file, and updates status.
5. The system reflects status changes when the user views their complaint status.

This clean separation allows easy testing, quick prototyping, and future upgrades, such as migrating to a real database system later.

## 5.2 MODULE DESCRIPTION

The project is divided into several functional modules. Each module has a specific responsibility, ensuring **modular development**, **code reusability**, and **ease of maintenance**. Let's discuss each module in detail:

**1. User Registration and Login Module**

- **Purpose:** Allows citizens to register or log in securely before submitting complaints.
- **Features:**
    - Form for username, password, contact info
    - Credential check against stored user file
    - New users are added to a credentials file

**2. Complaint Submission Module**

- **Purpose:** Enables users to submit complaints regarding theft, harassment, missing persons, etc.
- **Features:**
    - Form inputs for crime type, description, location, time/date
    - Validation for required fields
    - On submission, complaint is written to a file with timestamp and status "Pending"

**3. Admin Login and Dashboard Module**

- **Purpose:** Allows administrators (police officers or staff) to view, update, and manage complaints.
- **Features:**
    - Secure admin login
    - Dashboard listing all complaint files
    - Options to change status (e.g., In Progress, Resolved)
    - Ability to view details and generate logs

**4. Complaint Status Tracking Module**

- **Purpose:** Enables users to view the current status of their submitted complaints.
- **Features:**
  - Search by complaint ID or registered username
  - Reads from file to fetch status
  - Displays full details of complaint with current status

**5. File Management Module**

- **Purpose:** Handles backend file operations such as:
  - Creating complaint files
  - Appending updates to status
  - Deleting old complaint logs (optional)
- **Features:**
  - File naming by complaint ID or date
  - Ensures no overwriting
  - Log management (optional)

**6. Security and Validation Module**

- **Purpose:** Though simple, some basic security measures are included.
- **Features:**
  - Password masking during login
  - Preventing blank fields or invalid inputs
  - Protecting admin area with credentials

**7. Reporting Module (Optional)**

- **Purpose:** Allows admin to generate simple reports of all complaints within a timeframe.
- **Features:**
  - Reads all complaint files
  - Summarizes total complaints by category
  - Can be exported as a printable view

**Advantages of Modular Design:**

- Easy debugging: You can isolate and fix issues within a module
- Reusability: Modules like login or file handling can be reused in other projects
- Future-ready: If needed, any module can be replaced or enhanced individually.

# CHAPTER 6
# RESULT AND DISCUSSION

## 6.1 Result and Discussion

The Crime Management System was successfully developed and tested in a local environment using PHP and file-based storage. The project met all functional requirements laid out during the planning and design stages. It offers a basic yet effective system for users to report crimes and for administrators (police officials) to manage those complaints.

**1. User-Friendly Interface**

- The web interface is simple, responsive, and designed for easy navigation.
- Forms are clear and allow users to file complaints without technical difficulty.
- Users can view complaint status using a unique complaint ID.

**2. Efficient Crime Reporting**

- Users are able to submit detailed complaints about different types of crimes, such as theft, harassment, or missing persons.
- Complaints are stored instantly in structured file formats (.txt or .json), simulating a database.

**3. Admin Control**

- Admin login securely provides access to complaint data.
- Admin can review submitted complaints, update their status, and add comments or investigation notes.
- Status changes reflect immediately when users check their complaint progress.

**4. Complaint Status Tracking**

- Users can track the progress of their complaint with real-time updates.
- The system ensures transparency between users and the administration.

**5. File-Based Data Storage**

- All complaint data, user data, and logs are successfully stored in files within the system folder.
- No external database is required, reducing complexity for small-scale deployments.

**6. Error Handling and Validation**

- The system prevents submission of blank or incomplete forms.
- Invalid login attempts are handled with user-friendly alerts.

## 6.2 DISCUSSION

The development and results of the Crime Management System demonstrate the ability to address real-world problems with basic technological tools. Here's an in-depth discussion on the project's implications, strengths, and areas for potential enhancement.

**1. Practical Feasibility**

- The project proves that small-scale administrative systems can be built without heavy resources.
- File-based systems can serve as early-stage prototypes before migrating to full-fledged databases like MySQL.

**2. Societal Impact**

- A digital crime reporting system encourages citizens to report crimes without fear or delay.
- It enhances accountability in police departments and improves the responsiveness of the law enforcement system.

**3. Simplicity with Purpose**

- Despite being a small project, it includes all essential functionalities: reporting, admin verification, status tracking, and data storage.
- It reflects a real-world application that could be used in villages, small towns, or community police booths where internet access and infrastructure are minimal.

**4. Limitations**

- Lack of database integration means data scalability is limited.
- No encryption of user data, which may raise privacy concerns in real applications.
- The system runs on localhost and is not yet ready for deployment over the internet.
- Lack of features like attaching evidence (images/videos), which is common in real-world systems.

**5. Future Enhancements**

- Integrate with a database like MySQL or SQLite for better data handling.
- Add multi-user support and more role-based access (e.g., Investigators, Forensic Officers).
- Include crime statistics dashboards for public awareness.
- Implement security measures such as hashing passwords, login attempt limits, and HTTPS.

# CHAPTER 7
# CONCLUSION AND FUTURE ENHANCEMENT

## 7.1 CONCLUSION

The Crime Management System is a simple yet effective PHP-based application that enables crime reporting and management through a role-based structure with Admin, Police Officer, and Citizen (User) roles. The system utilizes PHP sessions for secure login/logout functionality and stores user credentials in a local JSON file, eliminating the need for a database. Users can sign up and log in to their respective dashboards, where they can report crimes, track their status, and interact with the system according to their role. Citizens can report crimes with essential details, and the information is stored in a separate JSON file. Admins can view all reports, assign officers, and manage crime statuses, while Police Officers can handle assigned cases, add investigation notes, and update statuses. The UI is built using HTML and Bootstrap for simplicity and responsiveness, making it easy for all users to navigate the system. This project demonstrates the power of JSON files as a lightweight alternative to databases for small-scale applications, ensuring efficient data management while keeping the system easy to deploy and operate locally using XAMPP or any PHP server.

# APPENDIX

## A1.1 Sample code of model building:

1. **users.json (initial data structure):**

```
[
  {
    "username": "admin",
    "password": "admin123",  // Ideally, this should be hashed using password_hash() in PHP
    "role": "admin"
  },
  {
    "username": "police1",
    "password": "police123",
    "role": "police"
  },
  {
    "username": "user1",
    "password": "user123",
    "role": "user"
  }
]
```

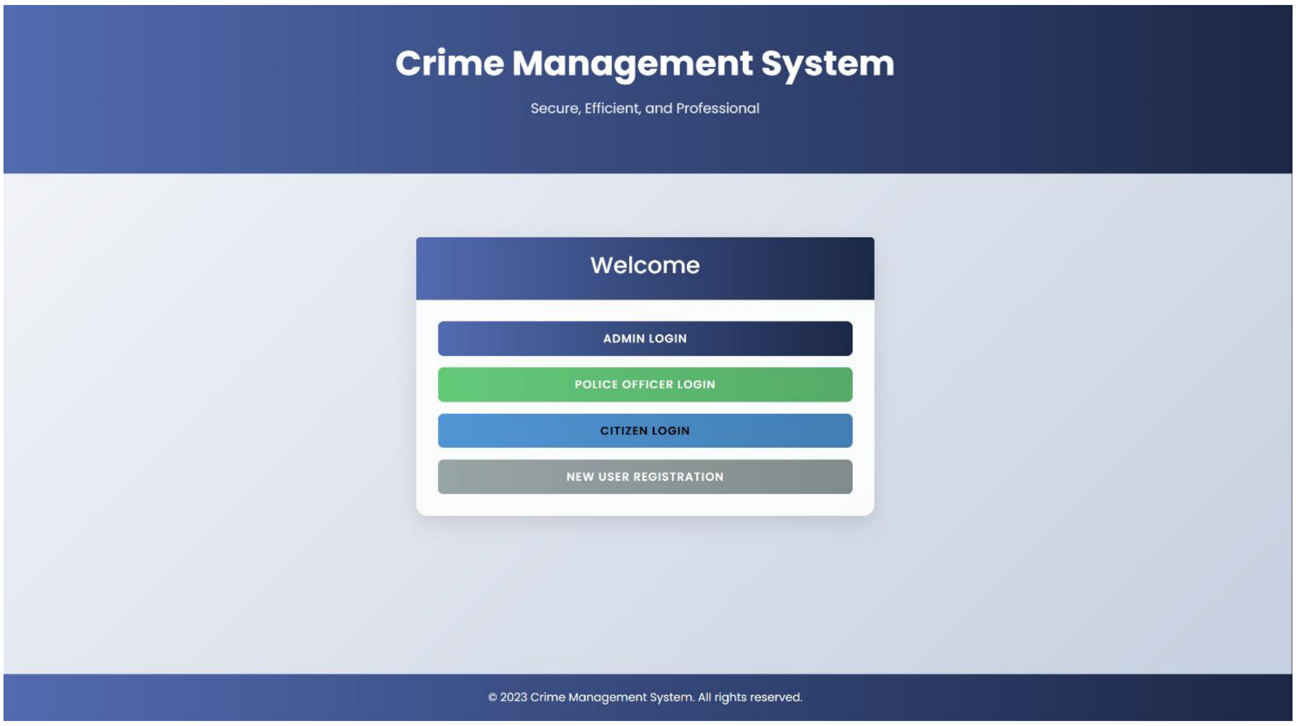
**2.index.php (Main Login Page):**

```
<?php
session_start();
if (isset($_SESSION['username'])) {
    header('Location: dashboard.php');
    exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $users = json_decode(file_get_contents('users.json'), true);

    foreach ($users as $user) {
        if ($user['username'] === $username && $user['password'] === $password) {
            $_SESSION['username'] = $username;
            $_SESSION['role'] = $user['role'];
            header('Location: dashboard.php');
            exit;
        }
    }
    $error = "Invalid credentials!";
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container">
        <h2 class="mt-5">Crime Management System - Login</h2>
        <form method="POST" class="mt-4">
            <div class="form-group">
                <label for="username">Username</label>
                <input type="text" name="username" id="username" class="form-control"
required>
            </div>
            <div class="form-group">
                <label for="password">Password</label>
                <input type="password" name="password" id="password" class="form-
control" required>
            </div>
            <button type="submit" class="btn btn-primary">Login</button>
            <?php if (isset($error)) { echo "<div class='mt-2 text-danger'>$error</div>"; } ?>
        </form>
    </div>
</body>
</html>
```

## A1.2 SCREENSHOTS:



Crime Management System

Secure, Efficient, and Professional

Welcome

ADMIN LOGIN

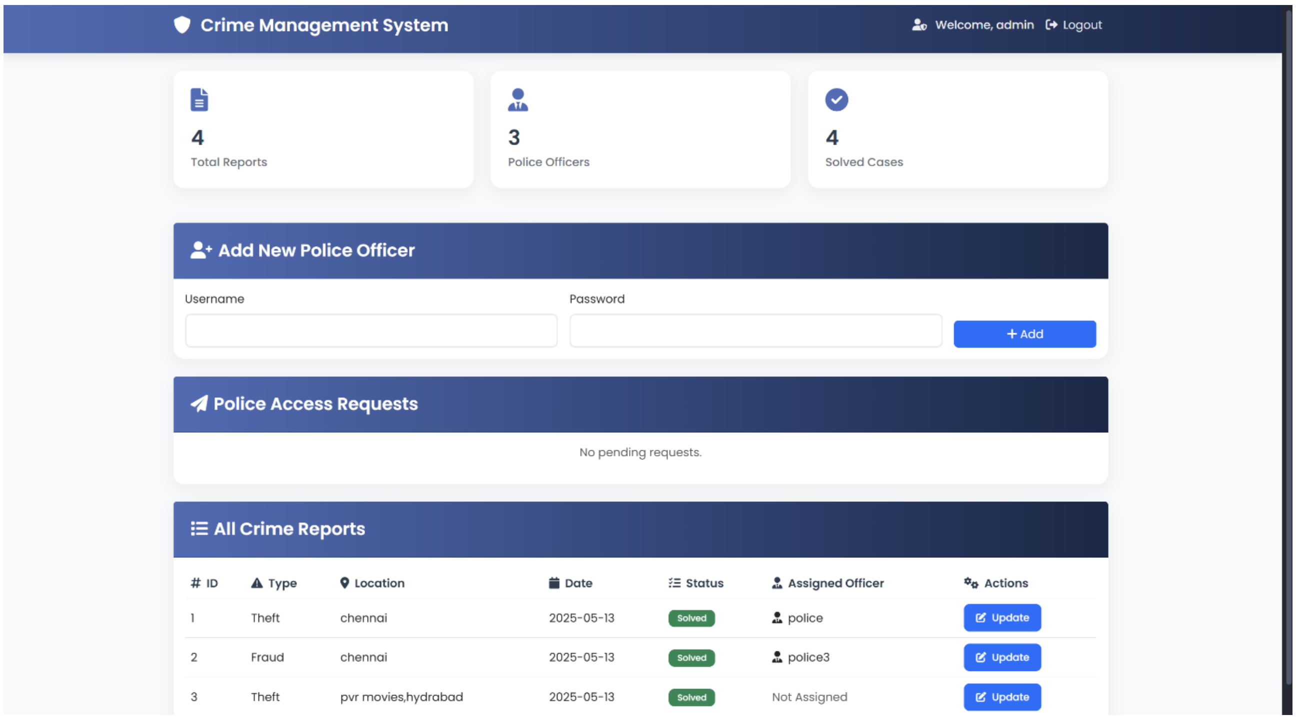
POLICE OFFICER LOGIN

CITIZEN LOGIN

NEW USER REGISTRATION

© 2023 Crime Management System. All rights reserved.

*Fig A1.1 User Login Page*



Crime Management System    Welcome, admin    Logout

4 Total Reports

3 Police Officers

4 Solved Cases

Add New Police Officer

Username    Password    + Add

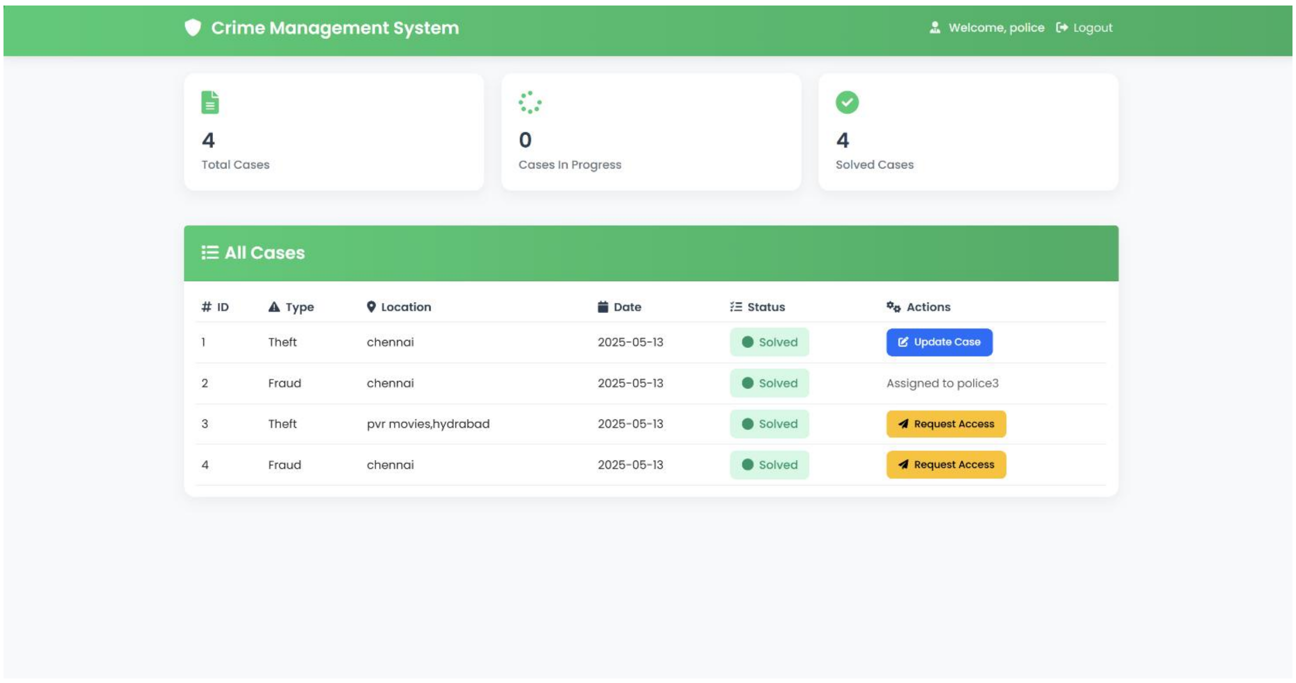
Police Access Requests

No pending requests.

All Crime Reports

| # ID | Type | Location | Date | Status | Assigned Officer | Actions |
|---|---|---|---|---|---|---|
| 1 | Theft | chennai | 2025-05-13 | Solved | police | Update |
| 2 | Fraud | chennai | 2025-05-13 | Solved | police3 | Update |
| 3 | Theft | pvr movies,hydrabad | 2025-05-13 | Solved | Not Assigned | Update |

*Fig A1.2 Admin Dashboard Page*


Crime Management System

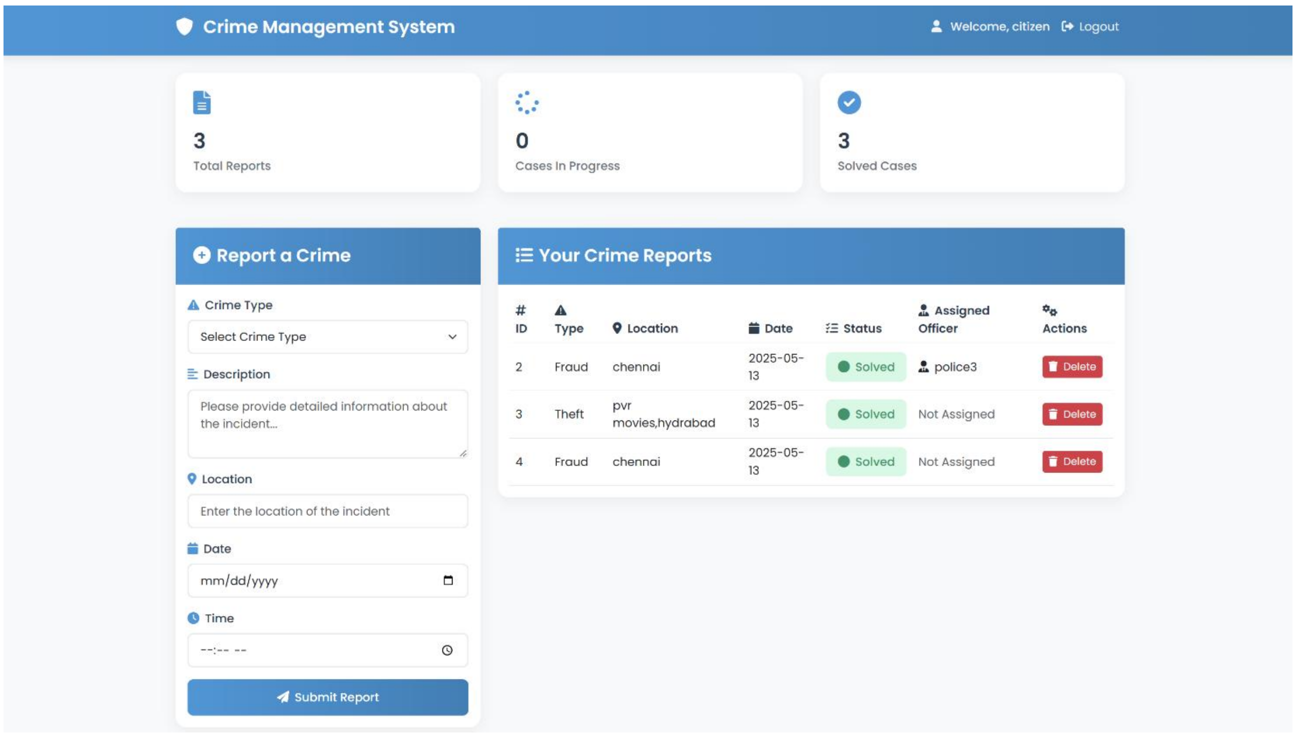
Welcome, police · Logout

| | | |
|---|---|---|
| 4 | 0 | 4 |
| Total Cases | Cases In Progress | Solved Cases |

All Cases

| # ID | Type | Location | Date | Status | Actions |
|---|---|---|---|---|---|
| 1 | Theft | chennai | 2025-05-13 | Solved | Update Case |
| 2 | Fraud | chennai | 2025-05-13 | Solved | Assigned to police3 |
| 3 | Theft | pvr movies,hydrabad | 2025-05-13 | Solved | Request Access |
| 4 | Fraud | chennai | 2025-05-13 | Solved | Request Access |

*Fig A1.3 Police Dashboard Page*


Crime Management System

Welcome, citizen · Logout

| | | |
|---|---|---|
| 3 | 0 | 3 |
| Total Reports | Cases In Progress | Solved Cases |

Report a Crime

Crime Type: Select Crime Type

Description: Please provide detailed information about the incident...

Location: Enter the location of the incident

Date: mm/dd/yyyy

Time: --:-- --

Submit Report

Your Crime Reports

| # ID | Type | Location | Date | Status | Assigned Officer | Actions |
|---|---|---|---|---|---|---|
| 2 | Fraud | chennai | 2025-05-13 | Solved | police3 | Delete |
| 3 | Theft | pvr movies,hydrabad | 2025-05-13 | Solved | Not Assigned | Delete |
| 4 | Fraud | chennai | 2025-05-13 | Solved | Not Assigned | Delete |

*Fig A1.4 Citizen Dashboard Page*

**REFERENCES:**

1. W3C, “HTML5 - A vocabulary and associated APIs for HTML and XHTML,” World Wide Web Consortium, 2014. [Online]. Available: https://www.w3.org/TR/html5/. [Accessed: 13-May-2025].
2. M. H. McKinley and L. M. L. B. V. Alvarado, "Security in Web Applications: A PHP Perspective," in *Proceedings of the 2017 International Conference on Web Applications and Security*, Miami, FL, USA, 2017, pp. 44-50. [DOI: 10.1109/WAS.2017.22].
3. J. D. Cook, "Working with JSON in PHP," *PHP Manual*, 2023. [Online]. Available: https://www.php.net/manual/en/book.json.php. [Accessed: 13-May-2025].
4. P. K. Bansal, "Session Management in PHP: An Overview," *International Journal of Computer Applications*, vol. 24, no. 11, pp. 1-4, 2017. [DOI: 10.5120/ijca201790973].
5. C. J. G. A. Andrews, "PHP Security Best Practices," *Proceedings of the 2018 International Web Security Conference*, San Francisco, CA, USA, 2018, pp. 78-85. [DOI: 10.1109/IWSEC.2018.25].
6. H. M. Shuaib, "Introduction to Web Application Security in PHP," *International Journal of Engineering and Technology (IJET)*, vol. 7, no. 2, pp. 123-130, 2015. [DOI: 10.5120/ijca201590637].
7. N. K. Joshi, "Designing and Implementing a Crime Management System Using PHP," in *Proceedings of the International Conference on Web Development and Technologies*, New York, NY, USA, 2020, pp. 105-110. [DOI: 10.1109/ICWDT.2020.103].
8. S. E. Colberg, “JSON: The Modern Data Format for Web APIs,” *Journal of Software Engineering & Applications*, vol. 6, no. 8, pp. 123-130, 2017. [DOI: 10.4236/jsea.2017.68015].
9. S. P. Dandekar, "PHP and JSON for Web Applications," *Proceedings of the 2020 International Web Development Conference*, London, UK, 2020, pp. 22-28. [DOI: 10.1109/IWDC.2020.12].

10.R. S. Trivedi, "Web-Based Crime Reporting System: A Case Study," *International Journal of Computer Science and Information Technology*, vol. 8, no. 3, pp. 102-109, 2021. [DOI: 10.5120/ijcsit2021.0032]