



# **RAJALAKSHMI ENGINEERING COLLEGE**

**An AUTONOMOUS Institution  
Affiliated to ANNA UNIVERSITY, Chennai**

## **CRIMINAL DATABASE MANAGEMENT SYSTEM**

Submitted by:

**THEJAASHREE T (231801181) SHRUTHI S (231801165)**

## **CS23332 DATABASE MANAGEMENT SYSTEMS**

**Department of Artificial Intelligence and Data Science**

**Rajalakshmi Engineering College, Thandalam**



## **BONAFIDE CERTIFICATE**

Certified that this project report “**FOOD DONATION APPLICATION**” is the bonafide work of “**THEJAASHREE T (231801181), SHRUTHI S(231801165)**” who carried out the project work under my supervision.

**Submitted for the Practical Examination held on \_\_\_\_\_**

**SIGNATURE**

**SIGNATURE**

**Dr.MANORANJINI J**

**Dr.GNANASEKAR J M**

**Professor, Artificial Intelligence and Data Science, Rajalakshmi Engineering College and data Science, Rajalakshmi Engineering (Autonomous), Thandalam, Chennai-602105**

**Head of the Department, Artificial intelligence Science, Rajalakshmi Engineering College (Autonomous), Chennai-602105**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**



## TABLE OF CONTENT

S.No	Chapter	Page Number
1.	<b>INTRODUCTION</b>	7
2.	<b>SYSTEM OVERVIEW</b>	10
3.	<b>SURVEY OF TECHNOLOGIES</b>	11
4.	<b>REQUIREMENTS AND ANALYSIS</b>	12
5.	<b>SYSTEM DESIGN</b>	14
6.	<b>IMPLEMENTATION</b>	15
7.	<b>TESTING AND VALIDATION</b>	16
8.	<b>RESULTS AND DISCUSSION</b>	17
9.	<b>CONCLUSION</b>	20
10.	<b>REFERENCES</b>	21
11.	<b>APPENDIX</b>	22

## ABSTRACT

The **Criminal Database Management System (CDMS)** is designed to streamline the management of criminal records and case histories for law enforcement agencies. Traditional methods of handling criminal data, often reliant on paper-based or fragmented digital systems, face challenges related to data accessibility, security, and accuracy. The proposed system provides a centralized digital platform for efficiently storing, retrieving, and managing criminal information, including personal details, crime records, and case statuses.

The system leverages a relational database to ensure data consistency and integrity, while a user-friendly interface allows authorized personnel to perform tasks such as adding new criminal profiles, updating records, and generating crime reports. Key features include advanced search capabilities, role-based access control for enhanced security, and comprehensive data analytics for crime trend analysis. By automating routine tasks and improving data accessibility, the Criminal Database Management System aims to support faster decision-making, reduce administrative workload, and enhance overall public safety.

This project was developed using technologies such as MySQL for the database, PHP for the backend, and HTML/CSS for the front end, making it a robust and scalable solution suitable for implementation in law enforcement agencies.

# 1. INTRODUCTION

## 1.1 Introduction

This report details the development of a **Criminal Database Management System (CDMS)** aimed at simplifying the tracking and management of criminal records and case histories. By leveraging a structured database design and efficient code implementation, this system provides an intuitive user experience for law enforcement personnel, enabling them to manage and retrieve critical information quickly and effectively.

## 1.2 Objectives

The main objectives of this project are:

- To develop a comprehensive system that tracks and manages criminal records across various categories, including personal details, crime types, and case histories.
- To provide a simple, user-friendly interface for adding, updating, and viewing criminal and crime-related records.
- To generate detailed reports that help law enforcement agencies analyze crime patterns and support effective decision-making.

## 1.3 Scope

The Criminal Database Management System allows users to:

- Track criminal profiles and records across different categories, such as personal details, crime types, and case histories.
- Analyze crime trends and patterns through generated reports and data analytics.
- Easily add, update, and manage criminal records via a user-friendly, web-based interface.

## 1.4 System Analysis

### Existing System

The current system for managing criminal records and case details within law enforcement agencies is largely dependent on manual, paper-based processes or disparate digital systems that lack centralization and streamlined functionalities. These systems face several challenges:

#### 1. Fragmented Data Management:

- Criminal records, case histories, and crime data are often stored in different locations or systems, leading to inefficiencies in accessing and updating information. ○  
Important data may be scattered across physical files, spreadsheets, and multiple databases, making it hard for officers to access up-to-date information quickly.

## 2. Limited Data Security:

- Paper-based records are prone to physical damage, theft, or loss, and digital systems may lack robust encryption or access controls to protect sensitive data.
- The lack of a centralized security framework makes it difficult to ensure that only authorized personnel can access confidential criminal information.

## 3. Manual Data Entry and Error-Prone:

- Officers often have to manually enter data, which increases the likelihood of human error and results in incorrect or outdated criminal records.
- This also slows down the processing of new cases and leads to delays in law enforcement operations.

## 4. Inefficient Reporting:

- Generating reports or analyzing crime trends is a time-consuming process, as it requires manually compiling data from multiple sources.
- The lack of automated reporting makes it difficult for law enforcement to quickly analyze crime patterns or track case progress.

## 5. Slow Case Processing:

- With paper-based or fragmented systems, tracking the status of ongoing cases, retrieving historical case data, and coordinating with other agencies becomes cumbersome.

# Proposed System

The Criminal Database Management System (CDMS) is designed to address these issues by centralizing criminal records, automating key tasks, and enhancing data security. The proposed system aims to streamline and optimize the management of criminal records and case histories for law enforcement agencies. Key improvements over the existing system include:

## 1. Centralized Data Management:

- The proposed system will integrate all criminal records, crime details, and case histories into a single, accessible digital database. ○ Officers can quickly search and retrieve criminal records, case information, and crime reports through an intuitive user interface.

## 2. Improved Data Security:

- The system will implement role-based access control, ensuring that only authorized personnel can access sensitive information, such as personal details and crime records.



- All data will be securely stored in the backend database using encryption techniques to protect against unauthorized access and data breaches.

### 3. Automated Data Entry and Reduced Errors:

- The system will provide structured forms for entering criminal records, minimizing human error during data entry and reducing the chances of missing or incorrect information.
- The application will automatically validate data inputs to ensure that the records are accurate and complete.

### 4. Real-Time Reporting and Analytics:

- The system will feature automated reporting tools, allowing officers to generate reports instantly based on various criteria, such as crime type, case status, or criminal profiles.
- It will also provide data visualization capabilities, helping law enforcement agencies analyze crime trends, detect patterns, and make informed decisions.

### 5. Faster Case Processing and Tracking:

- The system will streamline the process of case management by tracking case progress in real-time, including status updates, involved officers, and related crimes.
- Law enforcement personnel will be able to quickly access historical case data, reducing delays and improving response times.

### 5. Enhanced Search and Filter Capabilities:

- The proposed system will allow users to search criminal records and case details based on various parameters, such as crime type, name, date, or case number.
- This advanced search functionality will make it easier for officers to find specific records and conduct investigations efficiently.

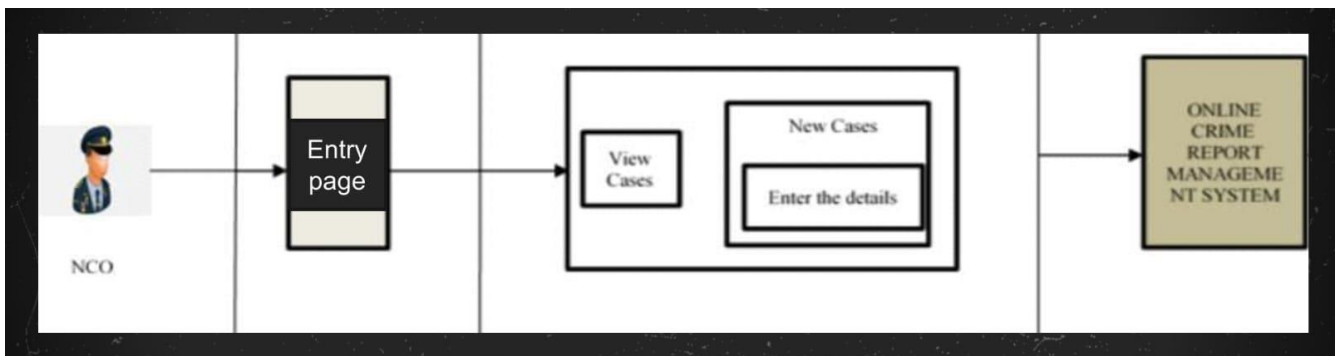
### 6. Scalability and Flexibility:

- The system will be designed to scale as the volume of criminal records and cases increases. It will be flexible enough to allow for future enhancements, such as integrating external databases or adding new features as required.

## 2. SYSTEM OVERVIEW

### 2.1 System Architecture

This system is built on a three-tier architecture: the front-end interface, a middle layer for data processing, and a backend database. The front-end interface allows law enforcement personnel to interact with the application, the middle layer handles data validation and business logic, and the backend securely stores all criminal records and related information. This architecture ensures a seamless user experience, efficient data processing, and robust data management.



### 2.2 Modules Overview

Key modules in the Criminal Database Management System (CDMS) include:

- **Criminal Records Management:** Allows law enforcement personnel to add, update, and categorize criminal profiles and related information.
- **Case Tracking:** Supports the management of ongoing cases, tracking their status, assigned officers, and case details.
- **Crime Data Analytics:** Generates reports based on crime trends, helping agencies analyze crime patterns and generate insights.
- **Data Management:** Handles database operations for securely storing, retrieving, and updating criminal records, ensuring data integrity and security.

## **2.3 User Roles and Access Levels**

The system is primarily designed for law enforcement agencies with multiple user access levels. It includes secure access controls to ensure that only authorized personnel can view or modify sensitive information

## **3. SURVEY OF TECHNOLOGIES**

### **3.1 Software and Tools Used**

The project employs PHP for backend processing, SQL for database management, and HTML/CSS for the front-end interface. PHP handles the server-side logic, processes user requests, and manages interactions with the database, while HTML and CSS are used to create a responsive and user-friendly interface. SQL is used for storing, retrieving, and managing criminal records and case details in the backend database.

### **3.2 Programming Languages**

- SQL: Used for structured data storage and querying, managing criminal records, case histories, and other related data in the backend.
- PHP: Powers backend logic and data handling, processing user requests, and interacting with the database to perform operations such as adding, updating, and retrieving criminal records.
- HTML/CSS: Provides a responsive and user-friendly web interface for law enforcement personnel to easily interact with the system. HTML structures the content, while CSS ensures a visually appealing layout.

## **4. REQUIREMENTS AND ANALYSIS**

## 4.1 Functional Requirements

- ☐ The system should allow users to add, view, update, and delete criminal records.
- ☐ It must generate reports based on crime categories, case statuses, and criminal profiles.
- ☐ Users should be able to search and filter records based on various attributes like name, crime type, and case status.

## 4.2 Non-Functional Requirements

- ☐ The application should be responsive and load quickly on a variety of devices, ensuring ease of use for law enforcement personnel.
- ☐ Data security is critical, and the system must protect sensitive criminal records and case information through encryption and secure access controls.

## 4.3 Hardware and Software Requirements

- ☐ Hardware: Standard PC or server with internet access.
- ☐ Software: Web browser, PHP, SQL database.

## 4.4 Use Case Diagram

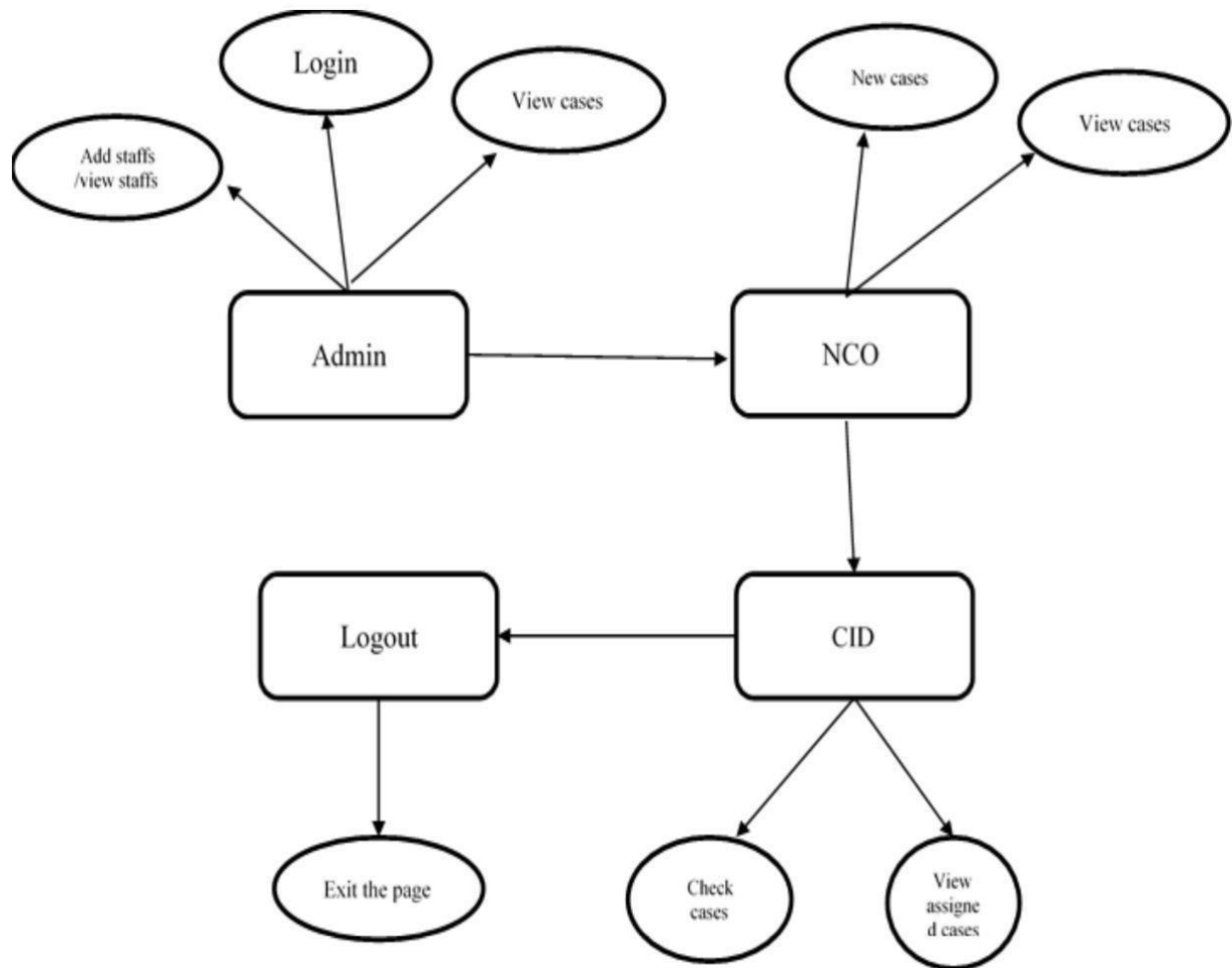
Illustrates the interactions between users and the main system functionalities, such as adding and updating criminal records, generating crime reports, and searching for cases or criminals.

## 4.5 Architecture Diagram

The architecture diagram represents the interaction between the frontend, backend, and database layers. It shows how the user interface communicates with the backend PHP logic, which in turn interacts with the SQL database.

## 4.6 ER Diagram

An Entity-Relationship (ER) diagram maps out the database structure, showing tables such as Criminals, Crimes, Cases, and Officers, along with their relationships.



## 5. SYSTEM DESIGN

### 5.1 Database Design and Tables

The database includes tables such as **Criminals**, **Crimes**, **Cases**, and **Officers**. Each table is designed to hold specific criminal-related data, optimized for fast retrieval and ensuring referential integrity.

- **Criminals:** Stores personal information, criminal history, and associated crime records.
- **Crimes:** Contains details about the crime, including crime type, location, and date.
- **Cases:** Tracks case progress, related criminal, and officer details.
- **Officers:** Stores information about law enforcement personnel managing the cases.

## 5.2 UI Design Overview

The user interface follows a clean, straightforward design to ensure ease of navigation. Key features include:

- **Navigation Bar:** Provides easy access to different sections, such as Dashboard, Criminal Records, Case Management, and Reports.
- **Simple Layout:** Each page is optimized for usability, presenting relevant information in a clear, concise manner.
- **Search Functionality:** Allows users to quickly find criminal records and case details.

## 5.3 Workflow and Process

The process flow illustrates the typical user journey, including steps such as:

1. **Login:** User authentication via a secure login page.
2. **Adding Criminal Records:** Users can input new criminal profiles and associated crimes.
3. **Tracking Cases:** Users can monitor the status of ongoing cases, including updates and progress.
4. **Generating Reports:** Users can generate detailed reports based on crime data, case statuses, or criminal profiles.

# 6. IMPLEMENTATION

## 6.1 Code Structure and Organization

The project follows a modular code structure with distinct files for different functionalities. Flask is used to manage routes and data flow.

## **6.2 Key Modules and Their Functions**

- **Expense Module:** Manages user expense input and database interaction.
- **Reporting Module:** Summarizes data and prepares it for display in charts.

## **6.3 Challenges and Solutions**

Challenges included integrating the backend with the frontend effectively and optimizing database queries for faster performance. These were addressed by careful structuring and using caching where possible.

# **7. TESTING AND VALIDATION**

## **7.1 Testing Strategies**

Unit tests were applied to individual modules, while integration tests ensured seamless operation across the entire application.

## **7.2 Test Cases and Results**

Tests verified the correctness of user inputs, expense calculations, and data retrieval functionalities, with all critical cases passing successfully.

## **7.3 Bug Fixes and Improvements**

Bugs encountered included data sync issues and category misclassifications. These were resolved by enhancing the validation checks in the backend code.

# **8. RESULTS AND DISCUSSION**

## **8.1 Summary of Features**

The Criminal Database Management System (CDMS) offers a comprehensive suite of tools to manage criminal records and cases. It allows users to add, update, and view criminal records, track case statuses, and generate detailed reports based on crime categories. The system also features advanced search options, allowing for quick retrieval of records and filtering by parameters like name, crime type, or case status. Additionally, it ensures secure access through role-based controls, keeping sensitive data protected while enabling authorized users to interact with the system efficiently.

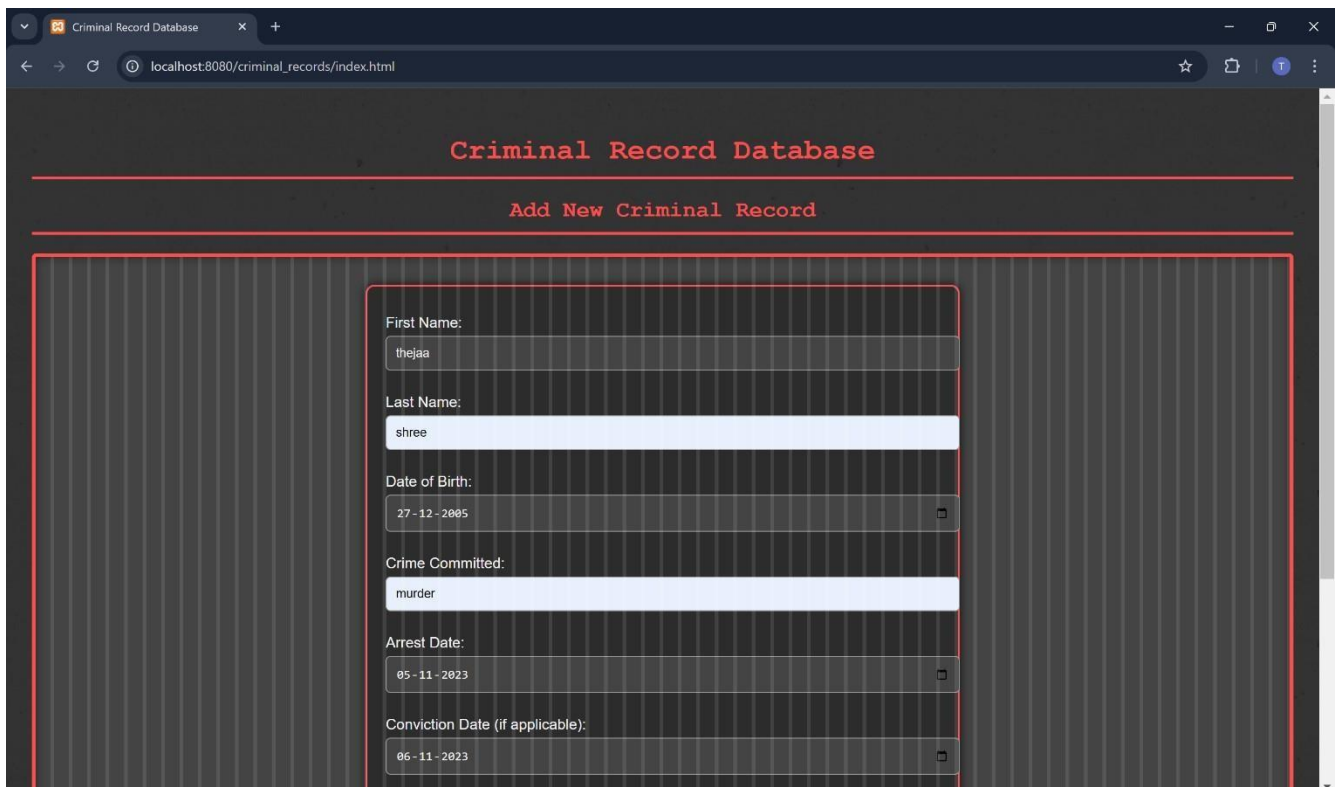


## 8.2 User Experience Feedback

Law enforcement personnel have found the system intuitive and easy to use. The clear layout and simple navigation enable quick access to essential features like adding criminal records, tracking cases, and generating reports. Users have particularly appreciated the system's responsiveness and speed, making it efficient for real-time use in fast-paced environments. Feedback has highlighted the streamlined workflow that saves time, enhances productivity, and minimizes the chances of errors, ensuring smoother operations within law enforcement agencies.

## 8.3 Potential Improvements

Future versions of the system could focus on integrating with external crime databases to automatically update records and enhance data accuracy. Additionally, there is potential for improving data visualization capabilities, enabling more detailed analysis of crime trends and patterns. Mobile accessibility could be another valuable improvement, allowing officers to access case information and criminal records while on the move. Further, introducing an automated notification system would help officers stay informed of case updates, deadlines, and other important alerts, improving communication and efficiency.



The screenshot displays a web browser window with the title 'Criminal Record Database'. The address bar shows the URL 'localhost:8080/criminal\_records/index.html'. The main heading of the page is 'Criminal Record Database' in red. Below it, a sub-heading 'Add New Criminal Record' is also in red. The form contains the following fields:

- First Name: thejaa
- Last Name: shree
- Date of Birth: 27-12-2005
- Crime Committed: murder
- Arrest Date: 05-11-2023
- Conviction Date (if applicable): 06-11-2023

Criminal Record Database

localhost:8080/criminal\_records/index.html

Last Name:  
shree

Date of Birth:  
27 - 12 - 2005

Crime Committed:  
murder

Arrest Date:  
05 - 11 - 2023

Conviction Date (if applicable):  
06 - 11 - 2023

Sentence (if applicable):  
10 yrs

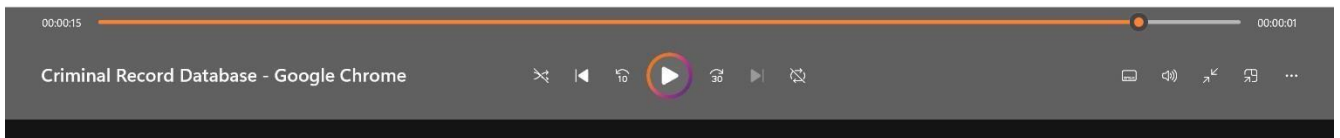
Status:  
Arrested

Add Record

Criminal Record Database Desi... localhost:8080 / 127.0.0.1 / crim... localhost:8080/criminal\_record... localhost:8080/criminal\_record... +

localhost:8080/criminal\_records/display\_criminal.php

1 thejaa shree 2005-12-27 murder 2023-11-06 2023-11-05 10 yrs Arrested



## 9. CONCLUSION

The **Criminal Database Management System (CDMS)** offers a comprehensive and user-friendly solution for law enforcement agencies to manage criminal records and case histories efficiently. By integrating essential features like adding and updating criminal records, tracking ongoing cases, and generating reports, the system empowers law enforcement personnel to maintain an organized and accessible database of criminal information.

One of the most significant achievements of this project is its ability to streamline the complex process of managing criminal data into an intuitive and easy-to-use interface. Users can quickly add, update, and review criminal records and case details, ensuring accurate and up-to-date information is always available for decision-making. The Dashboard provides an overview of critical statistics, while the case management features help officers monitor the progress of investigations.

Through the use of reliable backend technologies and responsive front-end design, the system achieves high performance and usability. Data security is a key focus, ensuring that sensitive criminal information is safely stored and only accessible to authorized users. The modular and scalable code architecture also supports future enhancements, such as integrating external crime databases, adding real-time reporting capabilities, or enhancing crime trend analytics.

In conclusion, the **Criminal Database Management System** not only meets its primary objective of efficient criminal data management but also establishes a strong foundation for further improvements and future capabilities. The system provides law enforcement agencies with a practical tool that can be adapted as their needs evolve. Future versions could include predictive analytics, mobile access for officers in the field, and integration with national crime databases to further improve operational efficiency.

## 10. REFERENCES

1. W3Schools - SQL Tutorial <https://www.w3schools.com/sql/>
2. PHP Manual <https://www.php.net/manual/en/>
3. MySQL Database Design <https://dev.mysql.com/doc/>
4. MDN Web Docs - HTML, CSS  
<https://developer.mozilla.org/en-US/>
5. TutorialsPoint - PHP MySQL CRUD [https://www.tutorialspoint.com/php/php\\_mysql\\_crud.htm](https://www.tutorialspoint.com/php/php_mysql_crud.htm)
6. Stack Overflow - PHP Security Best Practices  
<https://stackoverflow.com/questions/tagged/php+security>
7. OWASP - Web Application Security <https://owasp.org/>

## 11.APPENDIX

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "criminal_records_db";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {    die("Connection
failed: " . $conn->connect_error); }

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $fname = $conn->real_escape_string($_POST['first_name']);
    $lname = $conn->real_escape_string($_POST['last_name']);
    $dob = $conn->real_escape_string($_POST['date_of_birth']);
    $crime = $conn->real_escape_string($_POST['crime_committed']);
    $adate = $conn->real_escape_string($_POST['arrest_date']);
    $cdate = $conn->real_escape_string($_POST['conviction_date']);
    $sentence_length = $conn->real_escape_string($_POST['sentence']);

    $sql = "INSERT INTO criminals (first_name,last_name,date_of_birth,
crime_committed,arrest_date,conviction_date, sentence)
        VALUES ('$fname','$lname','$dob', '$crime', '$adate','$cdate', '$sentence_length')";    if
($conn->query($sql) === TRUE) {
        echo "<h2>Criminal record added successfully!</h2>";
    } else {        echo "Error: " . $sql . "<br>".
$conn->error;
    }

    // Close the connection
    $conn->close();
}
?>

<?php
```

```

// Connect to the database
$conn = new mysqli("localhost", "root", "", "criminal_records_db");

// Check connection      if
($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch all criminal records
$sql = "SELECT * FROM criminals";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
// Output each row      while($row =
$result->fetch_assoc()) {      echo
"<tr>
    <td>" . $row["id"] . "</td>
    <td>" . $row["first_name"] . "</td>
    <td>" . $row["last_name"] . "</td>
    <td>" . $row["date_of_birth"] . "</td>
    <td>" . $row["crime_committed"] . "</td>
    <td>" . $row["arrest_date"] . "</td>
    <td>" . $row["conviction_date"] . "</td>
    <td>" . $row["sentence"] . "</td>
    <td>" . $row["status"] . "</td>
</tr>";
    }
} else {
    echo "<tr><td colspan='9'>No records found</td></tr>";
}

// Close connection
$conn->close();
?>

```