



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Title of the Project

Criminal Record Management System (CRMS)

Abstract of the project

The Criminal Record Management System (CRMS) is designed to centralize crime-related information, providing fast and easy access to law enforcement agencies. The system aims to streamline the process of registering and managing criminal records, ensuring efficient data handling across different law-keeping bodies.

The CRMS is compatible with mobile and tablet devices, allowing officers to register criminals and access records from anywhere, improving on-the-go efficiency. By offering real-time data availability, the system ensures quick responses in critical situations.

Initially implemented in Mexico City, the CRMS has proven its value in enhancing law enforcement coordination. Implementing a similar system in the Indian subcontinent would help address the region's high crime rate, offering better crime tracking and response capabilities.

In summary, the CRMS enhances the effectiveness of law enforcement by improving data management, preventing information loss, and ensuring faster, more accurate decision-making.

Keywords

Generic Keywords

Databases, Middleware, Programming, Crime Information System

Specific Technology Keywords

Node.js, Express.js, PostgreSQL, HTML, JavaScript, HTTML, CSS and Javascript, JWT, REST API, NeonDB, Vercel

Project Type keywords

Analysis, Design, Implementation, Testing, Graphical User Interface, Security



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Functional components of the project

Following is a list of functionality of the system. More functionality that you find appropriate can be added to this list. And, in places where the description of functionality is not adequate, you can make appropriate assumptions and proceed.

Users of the System: CRMS will have two types of users: Law Enforcement Personnel (e.g., police officers, investigators) who will have access to criminal records and case management features, and Administrators who will oversee the system, manage user roles, and ensure data integrity.

Main Functionalities:

- **Login and Authentication:** This functionality allows users to securely log into the system using their credentials. It includes measures for password recovery and ensures that access is restricted based on user roles.
- **Criminal Record Entry and Updates:** Users can add new criminal records or update existing ones, including details about suspects, charges, and case outcomes. This feature ensures that all information is current and accurate.
- **Case Management:** This allows users to manage cases by tracking their status, assigning responsibilities, and adding notes or evidence. It provides a comprehensive view of ongoing investigations and helps streamline workflows.
- **Crime Statistics Dashboard:** This feature presents visual representations of crime data, including trends, types of crimes, and statistics. It aids law enforcement in understanding crime patterns and making informed decisions.
- **Suspect Profiling:** Users can create and maintain detailed profiles of suspects, including personal information, criminal history, and known associates. This helps in building comprehensive dossiers for investigations.
- **Audit Logs:** This functionality tracks all user actions within the system, creating a record of changes made to data. Audit logs enhance accountability and allow for the monitoring of any unauthorized access or modifications.
- **Search Functionality:** Users can efficiently search for records based on various criteria, such as names, case numbers, or specific crime types. This ensures quick access to relevant information when needed.
- **Contact Information:** This section provides users with essential contact details for relevant departments, legal authorities, and support services, ensuring easy communication when required.
- **Help Section:** The help section offers guidance on using the system, including FAQs, troubleshooting tips, and user manuals. This ensures that users can navigate the system effectively and resolve issues independently.

Steps to Start-off the Project

The system will be developed using Node.js for the backend, HTML, CSS and JAVASCRIPT for the frontend, and PostgreSQL for the database.

Steps to start the project include:



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

1. Familiarize with Node.js, Express.js, PostgreSQL, and HTML, CSS and Javascript.
2. Define the roles (police officer, admin) and the type of records (e.g., criminal records, case files).
3. Design the database and system architecture.
4. Ensure robust security features and user-friendly design.

Requirements

Hardware requirements

Number	Description	Alternatives (If available)
1	PC with 5 GB hard disk and 512 MB RAM	Not-Applicable

Software requirements

Number	Description	Alternatives (If available)
1	Windows 10/Linux OS	Not Applicable
2	NeonDB for PostgreSQL	Not Applicable
3	Node.js and Express.js	Not Applicable
4	JWT for Authentication	Not Applicable
5	HTML, CSS and Javascript	Not Applicable
6	Vercel for Deployment	Not Applicable

Manpower requirements

1 student can complete this in 2 to 3 months if he/ she works fulltime on it.

Milestones and Timelines

Number	Milestone Name	Milestone Description	Timeline Week no. from the start of the project	Remarks
1	Requirements Specification	Complete the detailed specification of the CRMS, including user	2-3	Attempt should be made to add some more relevant functionalities other than



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

		roles, functionalities, and system requirements. A document detailing the same should be written and a presentation made.		those that are listed in this document.
2	Technology familiarization	Gain an understanding of the technologies required for the implementation, including Node.js, Express.js, PostgreSQL, HTML, CSS, and JavaScript.	4-5	Presentations should focus on practical applications rather than purely theoretical concepts.
3	Database creation	Create a database schema with at least 100 users and 50 criminal records. Ensure the database is optimized for performance and security.	5-7	Finalizing the database schema is crucial for facilitating development and testing.
4	High-level and Detailed Design	Develop high-level and detailed designs for the system, including flowcharts, user interface designs, and detailed pseudocode for core functionalities.	7-9	Each design scenario should correlate with specified requirements, ensuring comprehensive coverage of the system's functionality.
5	Implementation of the front-end of the system	Develop the front-end of the system using HTML, CSS, and JavaScript, including user authentication, dashboards, and forms for managing criminal records and cases.	10-12	Begin formulating a test plan to ensure coverage of functionalities during the testing phase.
6	Integrating the front-end with the database	Connect the front-end developed in the previous milestone to the back-end services, allowing for data retrieval and updates	12-13	Integration should prepare the system for full testing.



Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

		in the database.		
7	Integration Testing	Conduct comprehensive testing of the system by executing all test cases developed in the previous milestones, ensuring functionality and security.	14-15	Allocate additional time to address any issues that arise during testing.
8	Final Review	Resolve any issues identified during the testing phase, ensuring that the system is fully functional and ready for final deployment.	16-18	Conduct a final review to ensure all requirements are met, providing justification for any unmet specifications.

Guidelines and References

Here are the HTTP links for each of the references related to your ****Criminal Record Management System (CRMS)**** project:

1. Node.js Documentation

<https://nodejs.org/en/docs/>

2. Express.js Guide

<https://expressjs.com/en/starter/installing.html>

3. PostgreSQL Documentation

<https://dev.PostgreSQL.com/doc/>

4. HTML Documentation

<https://developer.mozilla.org/en-US/docs/Web/HTML>

5. CSS Documentation

<https://developer.mozilla.org/en-US/docs/Web/CSS>

6. JavaScript Guide

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

7. JWT (JSON Web Tokens)

<https://jwt.io/introduction/>

8. RESTful API Design

<https://restfulapi.net/>