Name :-   Aumkumar H. Sheth

Enrollment no. :-   23SI002UG05932

Topic :-   Crimson Vault

Guided by :-   Prof. Bhavesh K Hadiyal

Stream :-   BCA (Sem-5)

College :-   Om Vindhyavasini College of IT & Mgt.

Acknowledgment :-

I **Aumkumar H. Sheth** would like to take this opportunity to express our sincere gratitude to all those who supported us in successful completion of our project, _**"Crimson Vault."**_
I extend my heartfelt thanks to my faculty guide **Prof. Bhavesh K Hadiyal**, whose valuable guidance, constructive feedback, and encouragement have been instrumental throughout the different phases of this project. His expertise and insights have helped us refine our ideas, structure our approach, and maintain focus on the project's objectives.
I also wish to thank my institution **OMVVIM College** for providing me the necessary resources, learning environment, and technical support that enabled me to carry out this work effectively.
My appreciation also goes to our peers, mentors, and friends who shared their suggestions and offered constructive criticism during the requirement analysis, feasibility study, and information-gathering stages. Their contributions helped us enhance the scope, security, and usability aspects of the project.
I acknowledge the significance of the reference materials, online resources, and existing platforms that inspired and guided me in shaping the _**"Crimson Vault"**_ into a practical and feasible solution.
Finally, I am deeply grateful to my family for their constant encouragement, patience, and moral support, which kept us motivated to complete this project successfully.
This project would not have been possible without the collective efforts, guidance, and support of all the individuals and institutions mentioned above.

Project Overview :-

The **<u>Crimson Vault</u>** is a secure and centralized digital platform designed to record and manage civil and criminal case data, including charges, trial proceedings, and judgments. The system aims to provide timely, accurate, and reliable access to sensitive legal information exclusively for authorized law enforcement and judicial personnel.

The platform's scope extends beyond simple record-keeping by offering advanced features such as a *Predictive Case Analytics*, **a** *Secure Evidence Linking Module*, **and** *User-Specific Workflow Customization.* These tools enhance decision-making, improve efficiency, and ensure the integrity of the justice process.

Built using HTML, CSS, and JavaScript for the front end, with PHP and MySQL for secure back-end processing and data storage, the Crimson Vault ensures both usability and security. Strict adherence to legal standards, encryption protocols guarantees data protection and compliance.

Ultimately, this project addresses the critical need for a centralized, accessible, and secure platform for managing case-related information. By bridging existing gaps in data management and ensuring role-based access, the Crimson Vault supports law enforcement officers, admins, and legal personnel in performing their duties with greater efficiency, accuracy, and accountability.

Index :-

1. Project Profile :-

1.1 Project Definition :-

The ***Crimson Vault*** is defined as a web-based, role-specific information management system that centralizes the storage, processing, and retrieval of civil and criminal case data. It serves as a structured digital repository where authorized law enforcement and judicial personnel can securely record case details, upload and access evidence, and track trial proceedings and judgments.

The system establishes clear boundaries by focusing exclusively on **authorized access, legal data security, and structured workflows**. Unlike public-facing platforms, it is not intended for general use but is restricted to verified users such as police officers, administrators, and legal professionals. Its primary aim is to replace fragmented manual or semi-digital processes with an integrated, automated, and legally compliant solution.

By definition, the project combines **front-end technologies (HTML, CSS, JavaScript)**, **back-end processing (PHP)**, and a **secure MySQL database** to deliver a reliable, scalable, and encrypted environment. It incorporates advanced modules like **Law & Statute Cross-Referencing**, **Predictive Case Analytics**, and **Secure Evidence Linking**, which collectively define its uniqueness as both a data repository and a decision-support tool.

In summary, the Crimson Vault is a **specialized legal data management platform**, strictly scoped for judicial and law enforcement use, ensuring efficiency, transparency, and compliance while safeguarding sensitive case-related information.

1.2 Background :-

In the modern justice system, the secure handling of legal data has become increasingly critical. Civil and criminal case information is often scattered across multiple systems, stored in physical records, or managed through isolated digital solutions. This fragmentation leads to delays, inefficiencies, and, in some cases, the risk of data loss or unauthorized access. Law enforcement officers and judicial personnel frequently face challenges in retrieving timely and accurate case details, which directly impacts the speed and quality of legal proceedings. Currently, government initiatives such as the *National Cyber Crime Reporting Portal* provide a platform for reporting specific incidents like cybercrimes. However, these solutions do not comprehensively address the broader need for a centralized, secure, and role-based system to manage diverse legal data—ranging from charges and trial proceedings to judgments and multimedia evidence.

The increasing volume of criminal and civil cases, coupled with the sensitive nature of associated evidence, highlights the urgent requirement for a controlled environment where authorized users can collaborate efficiently. Traditional manual processes not only consume significant time and resources but also lack the safeguards necessary to protect confidential information in compliance with national data laws.

The *Crimson Vault* project was conceptualized to bridge this gap. It provides a dedicated platform designed exclusively for authorized law enforcement and legal personnel, ensuring secure storage, efficient retrieval, and intelligent analysis of case records. By integrating

technologies such as **PHP, MySQL, HTML, CSS, and JavaScript** with advanced features like **encryption, predictive analytics, and statute cross-referencing**, the system aims to modernize legal data management and support transparency, accountability, and operational efficiency in the justice process.

1.3 Problem Statement :-

The current legal and law enforcement ecosystem suffers from the absence of a centralized, secure, and efficient platform for managing case-related data. Critical information such as charges, trial proceedings, judgments, and evidence is often fragmented across multiple systems or recorded manually, leading to delays, redundancy, and inconsistencies. Existing portals, such as the National Cyber Crime Reporting Portal, address only specific categories of crimes and lack the ability to serve as a comprehensive repository for diverse legal cases. This lack of integration results in challenges for police officers, administrators, and legal professionals, who require timely access to accurate data for decision-making, investigation, and judicial processes. Furthermore, the sensitive nature of legal information demands strict security, compliance with data protection laws, and role-based access, which many existing systems fail to provide.

Therefore, there is a pressing need for a unified digital platform that ensures **secure storage, structured workflows, intelligent data analysis, and reliable access** to legal case information, exclusively for authorized personnel.

1.4 Objectives :-

Centralized Data Management
- To design and implement a secure digital platform for recording and managing civil, criminal, and cybercrime case data, including charges, trial proceedings, judgments, and associated evidence.

Role-Based Access and Security
- To ensure that only authorized personnel (police officers, administrators, and legal professionals) can access the system, with role-specific permissions and strong security features such as Multi-Factor Authentication (MFA) and AES-256 encryption.

Workflow Efficiency
- To streamline legal data workflows by providing user-specific dashboards, automated notifications, and structured forms for efficient data entry and retrieval.

Integration of Advanced Features
- To develop intelligent modules such as:
  - *Predictive Case Analytics* to identify patterns and suggest related cases.
  - *Secure Evidence Linking Module* for encrypted storage and controlled access to multimedia evidence.

Compliance and Legal Standards
- To adhere to the Indian IT Act, judicial data retention policies, and national data protection laws, ensuring the platform remains legally compliant.

User Experience and Accessibility
- To build an intuitive, multilingual interface (English and Hindi) that is simple, professional, and accessible to a wide range of authorized users.

Scalability and Reliability
- To design the system for nationwide use with provisions for future scalability, offline access in low-connectivity areas, and potential international expansion.

Operational Support and Accountability
- To provide audit logs, activity tracking, and feedback mechanisms that ensure transparency, accountability, and continuous system improvement.

1.5 Methodology Overview:-

The development of the *Crimson Vault* will follow a **systematic and phased methodology**, ensuring that the platform is secure, functional, and aligned with stakeholder requirements. The methodology integrates standard software development practices with a focus on security, usability, and compliance.

1. **Requirement Analysis**
   - Collect and analyze requirements from stakeholders, including police officers, administrators, and legal personnel.
   - Define functional requirements such as user role management, case management, evidence linking, and analytics, alongside non-functional requirements like security, scalability, and performance.
   - Document compliance needs under the Indian IT Act and judicial data retention policies.

2. **System Design**
   - Prepare the system architecture, database schema, and user interface designs.
   - Design modules for *Law & Statute Cross-Referencing*, *Predictive Case Analytics*, *Secure Evidence Linking*, and *Audit Logs*.
   - Define workflows for each user role with role-based dashboards.

3. **Technology Selection**
   - Front-End: **HTML, CSS, JavaScript** for responsive and interactive user interfaces.
   - Back-End: **PHP** for server-side processing and secure data handling.
   - Database: **MySQL** for structured storage and retrieval of sensitive data.
   - Security: **AES-256 encryption**, **Multi-Factor Authentication**, and secure APIs for database integration.

4. **System Development**
   - Implement modules in iterative phases using a modular approach.
   - Integrate user authentication, case management forms, evidence upload, and dashboard features.
   - Ensure secure synchronization between the front-end and database.

5. **Testing and Validation**
   - Conduct functional testing for each module and integration testing for overall workflows.
   - Use anonymized data for testing sensitive features to maintain compliance.
   - Perform security testing (login, encryption, access control) and performance testing (response time within 2–3 seconds).

6. **Deployment**
   o Host the system on a secure server with SSL encryption and role-based access control.
   o Provide training to intended users for smooth adoption.
   o Establish backup and recovery mechanisms to prevent data loss.
7. **Maintenance and Future Enhancements**
   o Monitor system performance, security logs, and user feedback.
   o Provide continuous updates for scalability, additional language support, and possible integration with mobile applications.
   o Plan for long-term improvements, including advanced AI-driven analytics and international expansion.

1.6 Process of System :-

1. User Login & Authentication
- Authorized users log in with Multi-Factor Authentication, and access is granted based on roles (police, admin, legal personnel).

2. Case Entry & Evidence Upload
- Police officers create new case records using structured forms and upload documents, images, or videos securely.

3. Case Management
- Data is stored in an encrypted MySQL database.
- Admins oversee users and system activity, while legal personnel access case details in read-only mode.

4. Audit & Monitoring
- All user actions are logged for accountability and compliance.

5. Data Retrieval & Access
- Users search and retrieve case records quickly through a secure, multilingual dashboard.

6. Backup & Maintenance
- Regular backups, scalability options, and system updates ensure reliability and future growth.

1.7 System Requirement :-

1. Software Requirements
For Users (Client Side):
- **Operating System:** Windows 10 / Linux
- **Web Browser:** Latest version of Chrome, Firefox, or Edge
- **Other Tools:** PDF reader, Email client

For Admins (Server Side):
- **Operating System:** Windows 10 / Linux
- **Backend Environment:** PHP 8.x or above
- **Database Server:** MySQL 8.x or above
- **Web Server:** Apache / Nginx
- **IDE/Development Tools:** VS Code / PHPStorm / Git
- **Testing Tools:** Postman, XAMPP/WAMP for local testing

2. Hardware Requirements
For Users (Client Side):
- **Device:** Intel i3 or higher, 4 GB RAM, 500 GB HDD
- **Internet:** Stable broadband connection (minimum 10 Mbps)

For Admins (Server Side):
- **System:** Intel i5 or higher, 8 GB RAM, SSD recommended
- **Server:** Quad-Core CPU, 16 GB RAM, 1 TB SSD, RAID setup
- **Network:** 100 Mbps+ Internet connection, Secure LAN
- **Backup Device:** External storage / Cloud backup solution

2. Analysis :-

2.1 Limitation of existing system :-

Unrestricted Access Across Platforms

- In the existing setup, data access is not properly restricted. For example, any police officer can view cases uploaded by another officer, leading to confidentiality risks.

Lack of Role-Based Categorization

- Legal personnel are grouped under a single role without further categorization, which prevents assigning specific privileges or workflows according to their responsibilities.

Insufficient Access Control Mechanisms

- The current system does not enforce strict, role-specific permissions, which can result in unauthorized or unnecessary access to sensitive information.

Data Security Concerns

- Without proper segregation of roles and permissions, the possibility of data misuse, accidental modification, or exposure of confidential case records increases.

Operational Inefficiency

- The absence of refined role management leads to overlapping responsibilities, lack of accountability, and inefficient use of system resources.

2.2 Scope of the project :-

The scope of the ***Crimson Vault*** project is to design and implement a secure, centralized digital platform for the recording, management, and retrieval of civil, criminal, and cybercrime case data. The system will serve as a role-based environment, accessible only to authorized personnel such as police officers, administrators, and legal professionals.

Within this scope, the project will include:

- **Case Data Management:** Structured entry of charges, trial proceedings, judgments, and related information.
- **Evidence Handling:** Secure uploading, linking, and encryption of multimedia evidence.
- **Role-Based Access:** Defined permissions for police officers (case entry), admins (system management), and legal personnel (read-only access and analytics).
- **Advanced Features:** Modules such as *Law & Statute Cross-Referencing*, *Predictive Case Analytics*, and *Notifications & Alerts* to improve decision-making and efficiency.
- **Audit & Compliance:** Maintenance of detailed activity logs, adherence to the Indian IT Act, and support for long-term judicial data retention.
- **User Experience:** A clean, multilingual dashboard with tailored workflows for different roles.

The project will **not** be open for public access; it is strictly intended for authorized law enforcement and judicial use. Initially, the platform will focus on national-level implementation but will be designed with scalability for future expansion to additional regions and international scope.

2.3 Requirements :-

1. Functional Requirements
- **User Role Management:** Define and enforce permissions for police officers, admins, and legal personnel.
- **Authentication & Authorization:** Secure login with Multi-Factor Authentication (MFA).
- **Case Management Module:** Add, update, and retrieve complete case records.
- **Secure Evidence Linking Module:** Encrypt and control access to multimedia evidence.
- **Law & Statute Cross-Referencing:** Automatically link relevant legal sections to case data.
- **Predictive Case Analytics:** Suggest related cases and classify new cases using AI tools.
- **Notifications & Alerts:** Inform users about case updates, deadlines, and new evidence.
- **Audit Logs:** Track all user actions for accountability and compliance.
- **Feedback Mechanism:** Allow users to report issues and suggest improvements.

2. Non-Functional Requirements
- **Security:** AES-256 encryption, secure APIs, strict access control.
- **Legal Compliance:** Adherence to the Indian IT Act, data retention, and privacy laws.
- **Performance:** Average data retrieval within 2–3 seconds.
- **Scalability:** Support nationwide use and future international expansion.
- **Availability:** High uptime with planned maintenance.

- **Usability:** Clean, multilingual, and role-based dashboards.
- **Accessibility:** Government accessibility standards, multilingual support (English, Hindi).
- **Offline Access:** Limited read-only functionality during network outages.
- **Maintainability:** Modular design to allow easy updates and testing with anonymized data.

2.4 Feasibility Study :-

The *Crimson Vault* project was carefully examined for its feasibility across multiple dimensions to ensure its successful development and deployment.

1. Technical Feasibility
The technologies required for building this system—**HTML, CSS, JavaScript** for the interface, **PHP** for server-side processing, and **MySQL** for database management—are well-established, open-source, and widely supported. Features such as statute cross-referencing, predictive analytics, and secure evidence handling can be implemented with available libraries and frameworks. Security mechanisms like **AES-256 encryption**, **Multi-Factor Authentication**, and audit logs are technically achievable, making the project realistic from a development standpoint.

2. Economic Feasibility
The project is economically viable due to its reliance on open-source tools, eliminating licensing costs. The primary expenses include domain registration, hosting, security certificates, and development effort. Since much of the development can be handled at a low

cost (self-learned/freelance), the budget remains within a manageable range, making the system affordable while delivering high value to law enforcement and legal institutions.

3. Operational Feasibility

The system is designed around defined user roles—police officers, admins, and legal personnel—ensuring smooth integration into existing workflows. Features such as structured forms, dashboards, and notification systems make the platform user-friendly and efficient. Multilingual support (English and Hindi) increases accessibility, while the secure handling of evidence and legal records ensures user acceptance. With minimal training, users will be able to operate the system effectively.

4. Legal Feasibility

Given the sensitive nature of legal data, the system strictly adheres to the **Indian IT Act**, judicial data retention policies, and privacy regulations. Role-based access, anonymized test data, and encrypted evidence management ensure compliance with national laws and protect against misuse. This compliance framework guarantees that the project can be legally adopted by law enforcement and judicial bodies.

5. Schedule Feasibility

The project can be developed within a **3–4 month timeframe**, provided that requirement gathering, design, development, and testing phases are executed in sequence with proper project management. Since the technologies used are standard and well-documented, development risks are minimal. With a modular approach, features can be implemented and tested incrementally, ensuring timely completion.

2.5 Tools and Technologies Used :-

Tools and Technologies Used

The Crimson Vault project utilizes a combination of front-end and back-end technologies, along with specific tools for development, testing, and security.

- **Front-End:**
    - **HTML:** Used for structuring the content.
    - **CSS:** Used for styling and visual presentation.
    - **JavaScript:** Used to add interactivity and enhance user engagement.
- **Back-End:**
    - **PHP:** Serves as the server-side scripting language for managing data processing, handling secure user inputs, and dynamically rendering content.
    - **MySQL:** The database for structured storage and retrieval of sensitive data, including crime reports, case details, and user access logs.
- **Security:**
    - **AES-256 Encryption:** Essential for data at rest and in transit.
    - **Multi-Factor Authentication (MFA):** A crucial security feature, especially for sensitive user and administrative accounts.
    - **Secure APIs:** Used for integration with existing government and legal databases.
- **Development and Testing Tools:**
    - **IDE/Development Tools:** VS Code, PHPStorm, Git.
    - **Web Server:** Apache or Nginx.
    - **Testing Tools:** Postman, XAMPP/WAMP for local testing.

2.6 Architecture or Workflow :-

The following is an diagram of architecture of _**Crimson Vault**_ :-
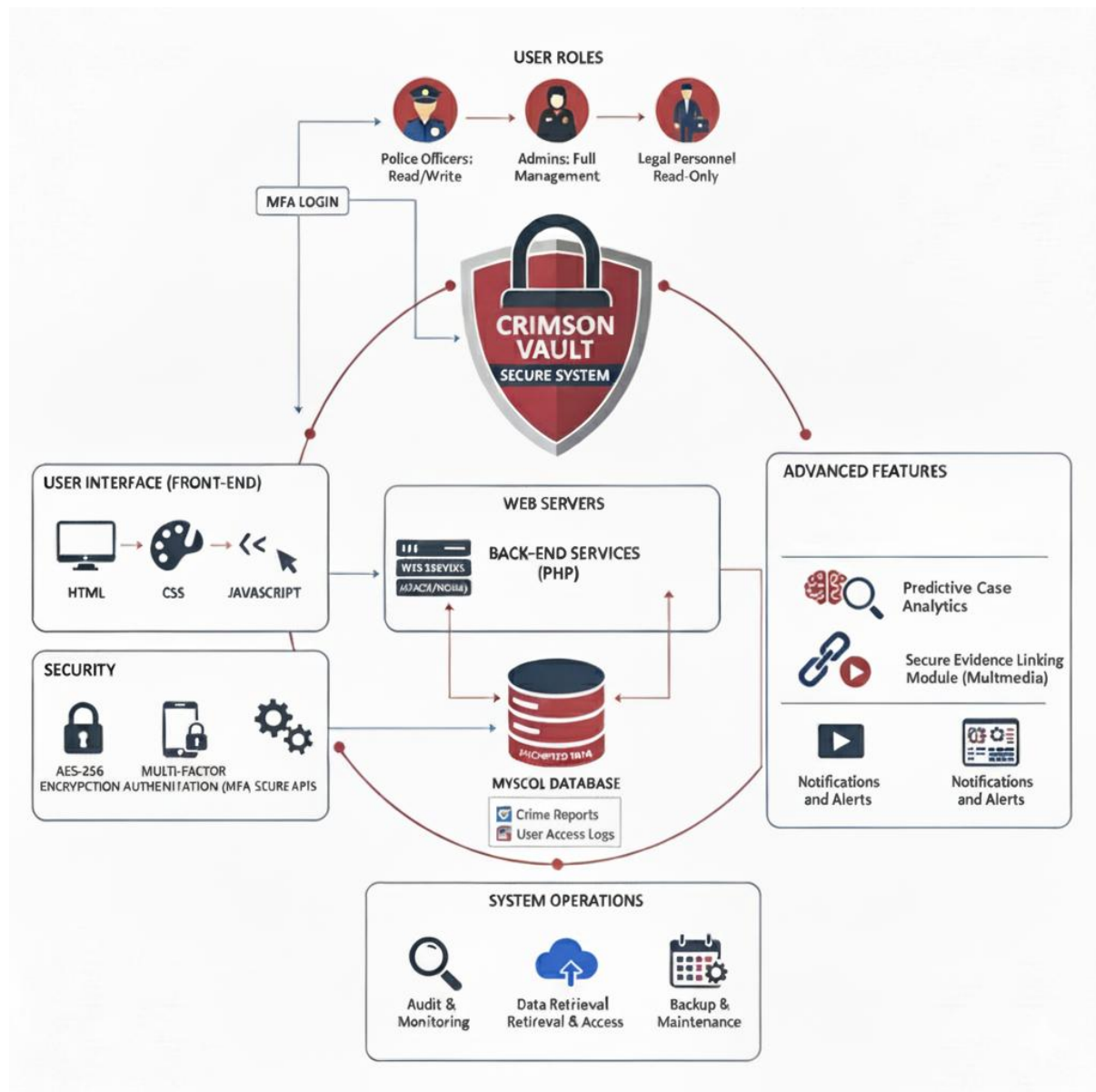
Workflow

The system's workflow is systematic and phased, designed to ensure a secure, functional, and user-centric experience.

1. **User Login & Authentication:** Authorized users, who include police officers, admins, and legal personnel, log in using Multi-Factor Authentication (MFA). Access is granted based on their specific roles.

2. **Case Entry & Evidence Upload:** Police officers are responsible for creating new case records using structured forms. They can also securely upload documents, images, and videos. The system also supports secure bulk data uploads.
3. **Case Management:** The uploaded data is stored in an encrypted MySQL database.
   o **Admins** have full management capabilities and can oversee users and system activity.
   o **Police officers** have read/write access for their cases.
   o **Legal personnel** have primarily read-only access to relevant data.
4. **Advanced Features:**
   o **Law & Statute Cross-Referencing Engine:** This module automatically links cases to relevant legal statutes.
   o **Predictive Case Analytics:** A simple AI tool that suggests related past cases or helps classify new crimes.
   o **Secure Evidence Linking Module:** Provides encrypted storage and controlled, role-based access to confidential multimedia evidence.
   o **User-Specific Workflow Customization:** Allows for tailored dashboards for different user roles, enhancing efficiency.
   o **Notifications and Alerts:** A system to send timely updates on case proceedings, new evidence, and deadlines.
5. **Audit & Monitoring:** A comprehensive log of all user activity is maintained for accountability, audit trails, and security monitoring.
6. **Data Retrieval & Access:** Users can search and retrieve case records quickly, ideally within a few seconds, through the secure and multilingual dashboard.
7. **Backup & Maintenance:** The system includes regular backups, scalability options, and system updates to ensure reliability and future growth.

2.7 Data Flow Diagrams (DFD) :-

Level 0 :-

Data Flow Diagram
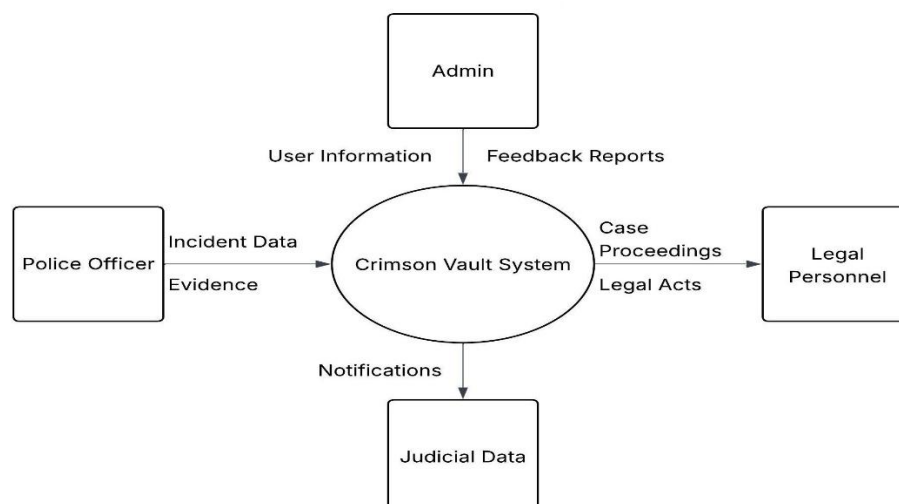
Level 1 :-

Level 1 DFD Diagram
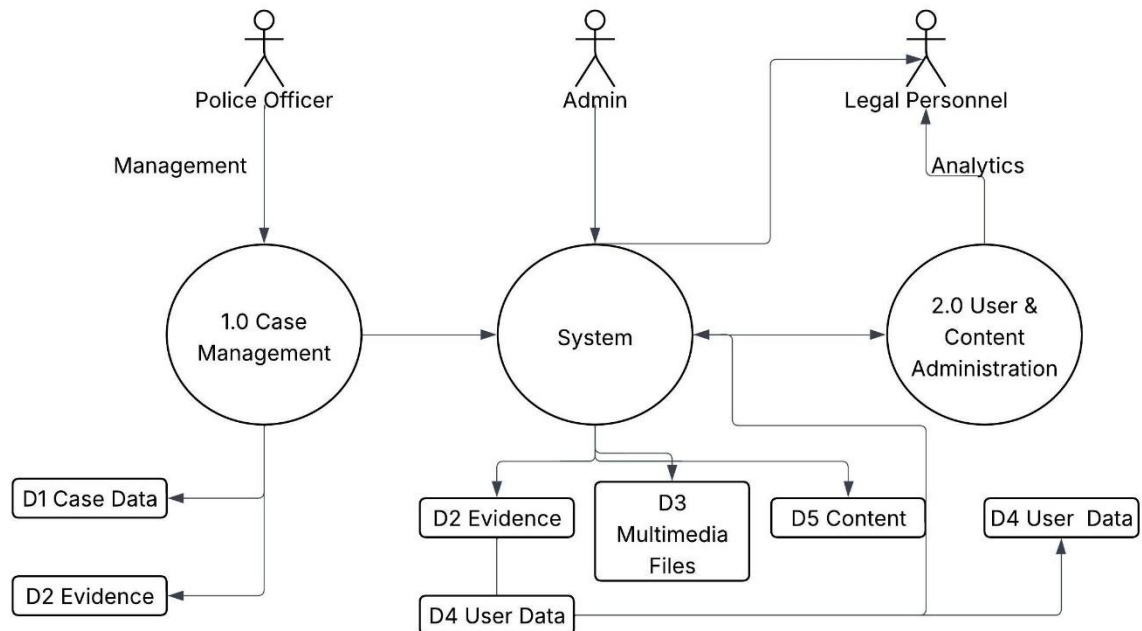
2.8 UML (Unified Modelling Language) :-

Use Case :-
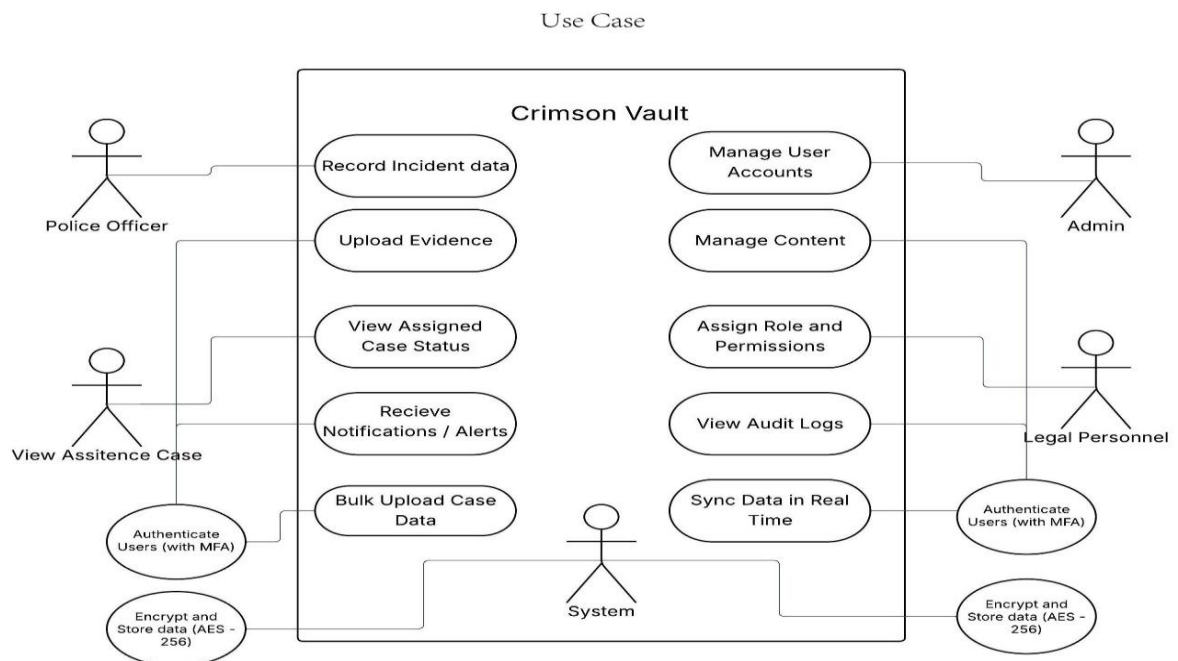
Class Diagram :-

Class Diagram

Officer

officerID : int
name : string
rank : string
department : string

submitIncident() : void
recieveNotifications() : void

One ———————————— Many

Incident

incidentID : int
tyoe : string
description : string
status : string

addEvidence() : void
getcaseDetails() : void

Many

Criminal

criminalId : int
criminalName : string
numberofCases : int
caseType : string
numberofJudgements : int

criminalData() : void

Many

One

Many

Admin

adminID : int
username : string
password : string

generateReports() : void

Evidence

evidenceID : int
description : string
filePath : string

uploadFile() : void

Activity Diagram :-

Start → User Login & Authentication (MFA) → Role Verification

- **Police Officer:** Add New Case → Upload & Encrypt Evidence → Manage Accused Profiles → Edit/View/Manage Cases → Track Progress (Open → Trial → Closed)
- **Legal Personnel:** Add Trial Proceedings → Add Judgments → Submit Analytics → Receive Notifications
- **Admin:** Add/Manage Users & Roles → Oversee Audit Logs & Activities → Send Notifications → Monitor Feedback & Support

Search & Retrieve Case Records → Audit & Monitoring → Automated Backup & Updates → End

3. Design :-

3.1 Data Dictionary :-

1. Table: members (Stores all system users)

Field Name	Description	Data Type	Constraints
user_id	Unique user identifier	INT(11)	Primary Key, Auto Increment
full_name	Full name of user	VARCHAR(50)	
Email	User's email ID	VARCHAR(100)	Unique (recommended)
Phone	Contact number	VARCHAR(15)	
password_hash	Encrypted password	VARCHAR(255)	Not Null
mfa_enabled	Multi-Factor Authentication flag	TINYINT(1)	Default: 0 or 1
last_login	Last login date and time	DATETIME	
Status	Account status (active/inactive)	VARCHAR(10)	
role_name	User role (Admin, Police, Legal)	VARCHAR(50)	

2. Table: cases (Stores case information)

Field Name	Description	Data Type	Constraints
case_id	Unique case identifier	INT(11)	Primary Key, Auto Increment
officer_id	Officer who created the case	INT(11)	Foreign Key → members(user_id)
Title	Case title/short name	VARCHAR(50)	
case_type	Case type (Criminal, Civil, Cyber)	VARCHAR(15)	
date_filed	Date the case was filed	DATE	
Status	Current status (Open/Closed/Trial)	VARCHAR(15)	
Description	Detailed description of the case	TEXT	
created_at	Case creation timestamp	TIMESTAMP	Default: Current Timestamp

3. Table: judgments

Field Name	Description	Data Type	Constraints
judgment_id	Unique judgment ID	INT(11)	Primary Key, Auto Increment
case_id	Related case	INT(11)	Foreign Key → cases(case_id)
user_id	Legal personnel who issued judgment	INT(11)	Foreign Key → members(user_id)
judgment_text	Text of the judgment	TEXT	Not Null
created_at	Date and time judgment recorded	DATETIME	Default: Current Timestamp

4. Table: accused (Details of accused persons linked to a case)

Field Name	Description	Data Type	Constraints
accused_id	Unique accused ID	INT(11)	Primary Key, Auto Increment
case_id	Linked case	INT(11)	Foreign Key → cases(case_id)
Name	Name of accused	VARCHAR(100)	
father_name	Father's name of accused	VARCHAR(100)	
Gender	Gender	VARCHAR(10)	
Birthdate	Date of Birth	DATE	
Height	Height in cm	DECIMAL(5,2)	
Weight	Weight in kg	DECIMAL(5,2)	
skin_colour	Skin complexion	VARCHAR(30)	
blood_group	Blood group	VARCHAR(5)	
current_charges	Current charges filed	TEXT	

5. Table: evidence (Stores uploaded case evidence)

Field Name	Description	Data Type	Constraints
evidence_id	Unique evidence ID	INT(11)	Primary Key, Auto Increment
case_id	Related case	INT(11)	Foreign Key → cases(case_id)
uploaded_by	Officer who uploaded evidence	INT(11)	Foreign Key → members(user_id)
file_name	File name	VARCHAR(255)	
file_path	Path where evidence is stored	TEXT	
file_type	File type (pdf, image, video)	VARCHAR(50)	
access_level	Access type (public, restricted)	VARCHAR(15)	
uploaded_at	Date and time of upload	DATETIME	

6. Table: trial_proceedings

Field Name	Description	Data Type	Constraints
tp_id	Trial proceeding ID	INT(11)	Primary Key, Auto Increment
case_id	Related case	INT(11)	Foreign Key → cases(case_id)
user_id	Legal/Officer updating proceedings	INT(11)	Foreign Key → members(user_id)
proceeding_note	Notes of the trial	TEXT	Not Null
created_at	Date and time note added	DATETIME	Default: Current Timestamp

7. Table: analytics (AI/Prediction results)

Field Name	Description	Data Type	Constraints
analytics_id	Analytics record ID	INT(11)	Primary Key, Auto Increment
case_id	Related case	INT(11)	Foreign Key → cases(case_id)
Prediction	Prediction or analysis result	TEXT	
generated_at	Date and time of prediction	DATETIME	

8. Table: similar_cases

Field Name	Description	Data Type	Constraints
Id	Unique record ID	INT(11)	Primary Key, Auto Increment
analytics_id	Related analytics record	INT(11)	Foreign Key → analytics(id)
similar_case_id	Case found similar	INT(11)	Foreign Key → cases(case_id)

9. Table: audit_log

Field Name	Description	Data Type	Constraints
log_id	Log record ID	INT(11)	Primary Key, Auto Increment
user_id	User who performed action	INT(11)	Foreign Key → members(user_id)
Action	Action performed	TEXT	
ip_address	IP address of the action	VARCHAR(45)	
Timestamp	Date and time of the action	DATETIME	

10. Table: notifications

Field Name	Description	Data Type	Constraints
ntf_id	Notification ID	INT(11)	Primary Key, Auto Increment
user_id	User receiving notification	INT(11)	Foreign Key → members(user_id)
Message	Notification message	TEXT	
Status	Notification status (read/unread)	ENUM	Default: unread
created_at	Date of notification	DATETIME	

11. Table: feedback

Field Name	Description	Data Type	Constraints
fdb_id	Feedback ID	INT(11)	Primary Key, Auto Increment
user_id	User who submitted feedback	INT(11)	Foreign Key → members(user_id)
Message	Feedback message	TEXT	
created_at	Date and time of feedback	DATETIME	

12. Table: support_requests

Field Name	Description	Data Type	Constraints
Id	Support request ID	INT(11)	Primary Key, Auto Increment
user_id	User who raised the issue	INT(11)	Foreign Key → members(user_id)
Name	Name of the requester	VARCHAR(150)	Not Null
Email	Email of requester	VARCHAR(200)	Not Null
Issue	Issue description	TEXT	Not Null
created_at	Date and time submitted	DATETIME	Default: Current Timestamp

13. Table: password_resets

Field Name	Description	Data Type	Constraints
Id	Reset request ID	INT(11)	Primary Key, Auto Increment
Email	User's email for password reset	VARCHAR(100)	Not Null
Otp	One-time password (reset code)	VARCHAR(10)	Not Null
expires_at	Expiry time for OTP	DATETIME	Not Null
created_at	Record creation time	TIMESTAMP	Default: Current Timestamp

3.2 Normalization :-

1. First Normal Form (1NF)
Rules:
- No repeating groups or arrays.
- Atomic values only.

In my schema:
- All tables have atomic columns (e.g., name, email, gender, etc.).
- Evidence files are stored with **file path**, not multiple file columns → satisfies atomicity.
- No multi-valued attributes.

Status: Achieved

2. Second Normal Form (2NF)
Rules:
- Must be in 1NF.
- No partial dependency (non-key attributes should depend on whole primary key).

In my schema:
- Tables like cases, members, accused, evidence have single-column primary keys → no chance of partial dependency.
- Composite key tables like similar_cases (analytics_id, similar_case_id) are correctly designed since both keys together identify uniqueness.

Status: Achieved

3. Third Normal Form (3NF)

Rules:

- Must be in 2NF.
- No transitive dependencies (non-key attributes should not depend on other non-key attributes).

In my schema:

- members table → role_name is directly related to user_id (fine).
- cases table → status, title, case_type are attributes of case only (no indirect dependency).
- audit_log → ip_address, action, timestamp depend only on log_id.
- evidence → file info depends only on evidence_id, no transitive dependency.

Status: Achieved

4. Boyce-Codd Normal Form (BCNF)

Rules:

- For every functional dependency (X → Y), X should be a superkey.

In my schema:

- Example: members.email → user_id (since email is unique).
- All FKs correctly reference PKs (no violation).
- No redundant functional dependencies.

Status: Achieved (BCNF compliant)

4. Implementation and testing :-

4.1 Screen layouts :-
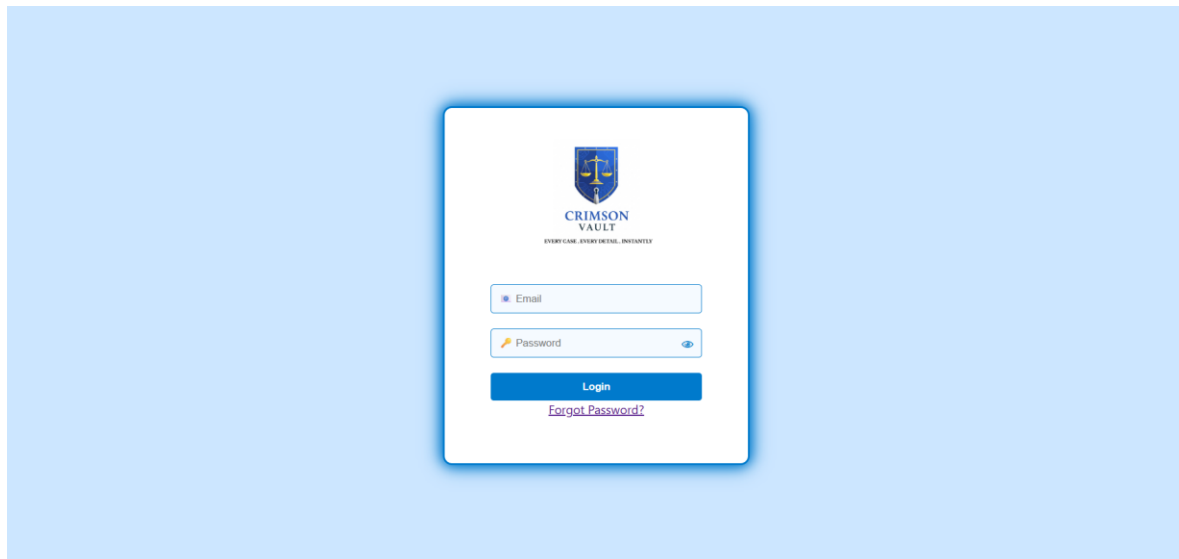
This is the main page (index,php) of the project which enables the user to login. Login is compulsory to use the website.

Now we will move forward to admin dashboard :-
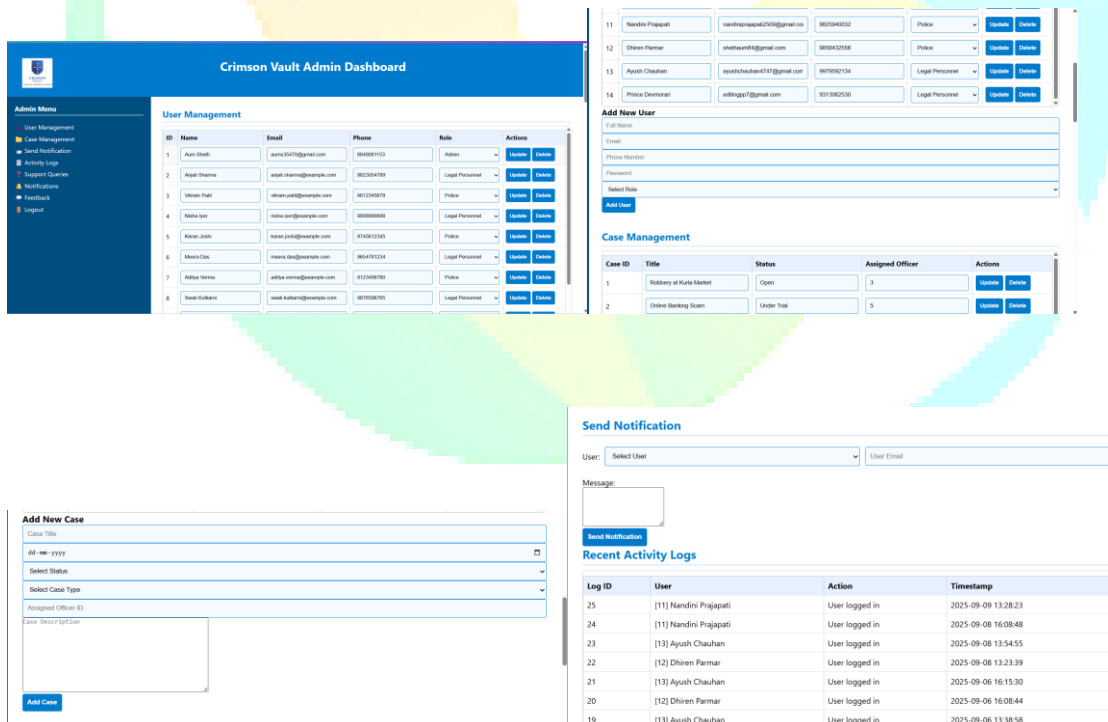
Support Queries
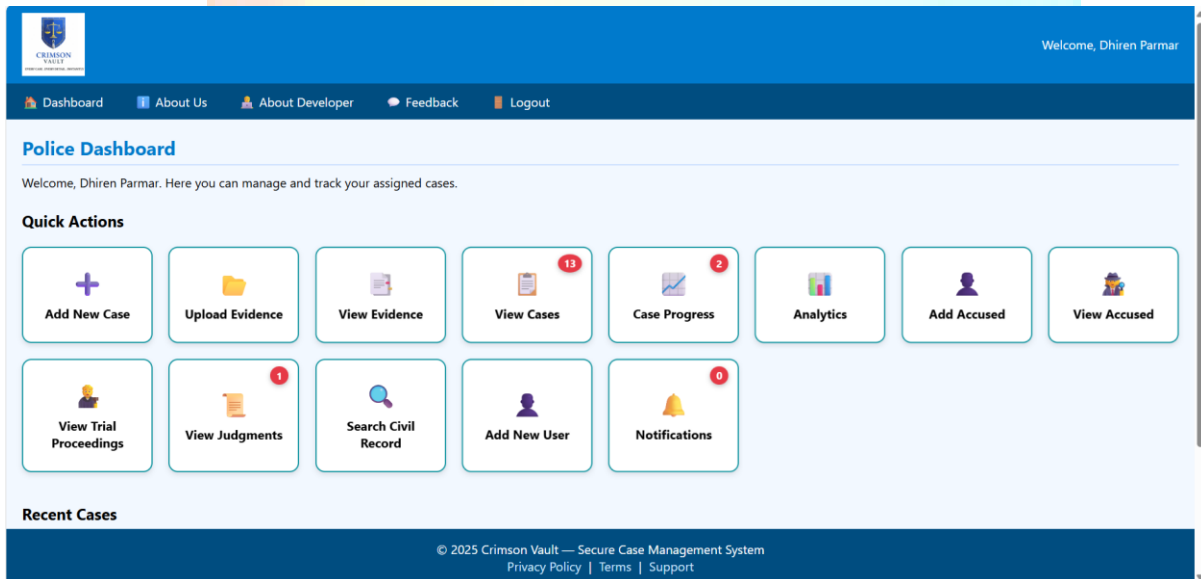
ID	Name	Email	Issue	Submitted On
2	Ayush Chauhan	ayushchauhan4747@gmail.com	Analytics is not working \r\n	2025-09-06 13:43:57
1	Dhiren Parmar	shethaum84@gmail.com	I can\'t upload the evidence properly	2025-08-26 21:35:46

User Feedback

- **Nandini Prajapati:** Very nice :) *(2025-08-26 13:17:52)*
- **Aum Sheth:** Good *(2025-08-26 13:11:11)*
- **Aum Sheth:** Nice *(2025-08-26 12:58:06)*
- **Aum Sheth:** Easy to use functionality *(2025-08-26 12:56:20)*
- **Aum Sheth:** Nice UI *(2025-08-26 12:54:58)*

The above are screenshots of admin panel (dashboard.php) which enables the admin to view users, add users, view cases, add cases, send notification to a user, view audit logs, view support queries and feedback of the website.

Now we will move forward to the police officer's perspective in this website :-

This is the main police dashboard (police_dashboard.php). In this dashboard the police officer will be able to :- Add new case, Upload Evidence,View Evidence, View Cases with total number of cases registered, Progress of assigned cases, View Analytics, Add Accused, View Accussed, View Trial Proceedings, View Judgements, Search Civil Record, Add new user and can view notifications sent by the admin along with recently assigned cases.

This is the add_case.php file where police officer can add case information like Title, Type of case, Date filed, Status of the case and description.

This upload_evidence.php where police officer can upload evidence by selecting the case.
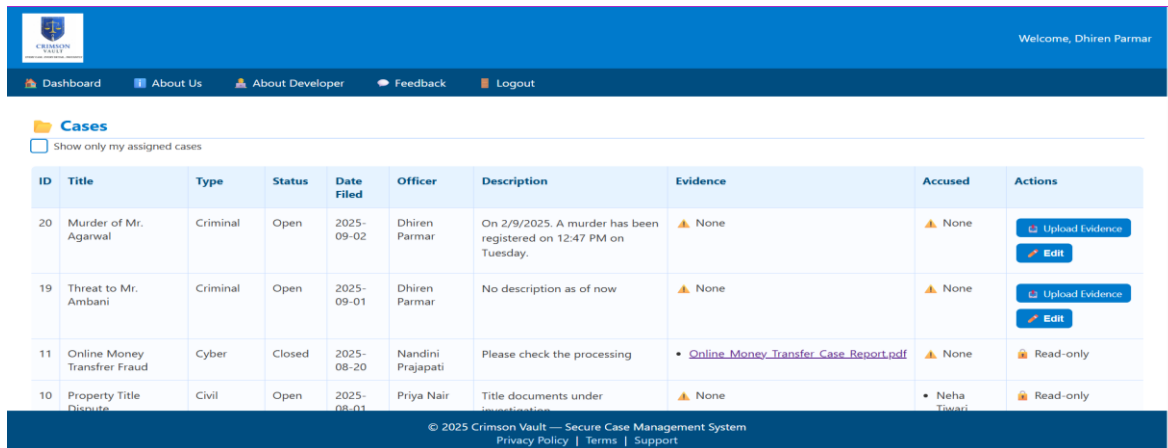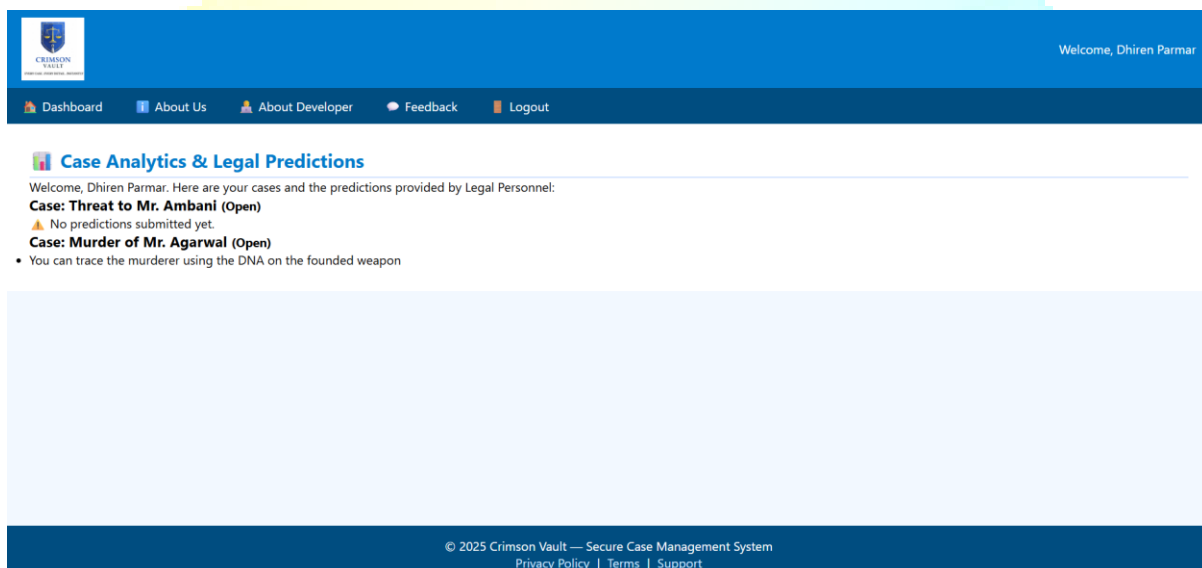
This view_evidence.php where officer can view/download their uploaded evidence of assigned case.

The previous page's last pic is the cases.php file where officer can view the cases of other police officers in read only mode, while own assigned cases can be edited, upload evidence and can view evidence uploaded of any case.
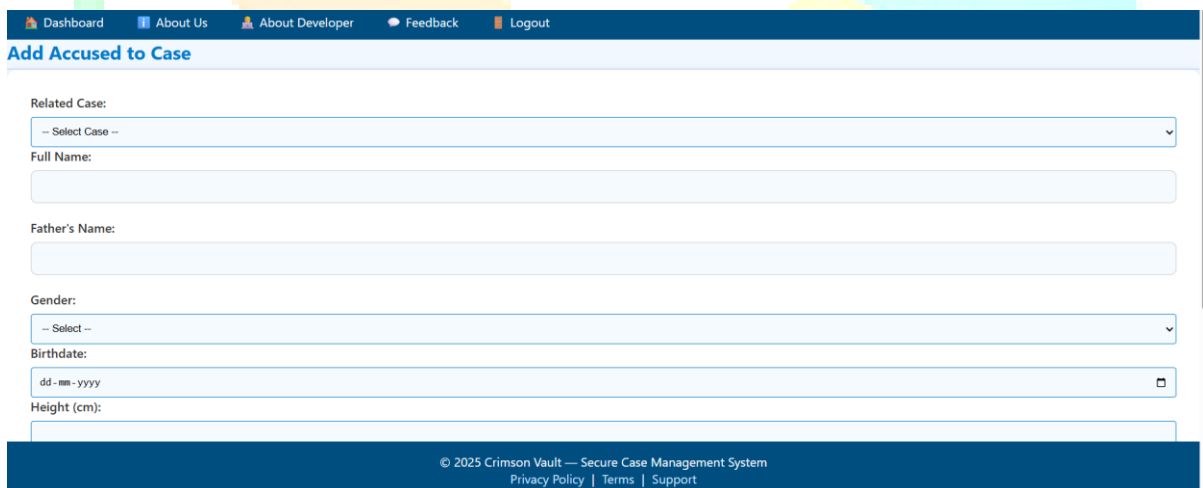
This cases analytics page where legal personnel can give their analysis and prediction and police officer can view the analysis and prediction in their own assigned cases only.
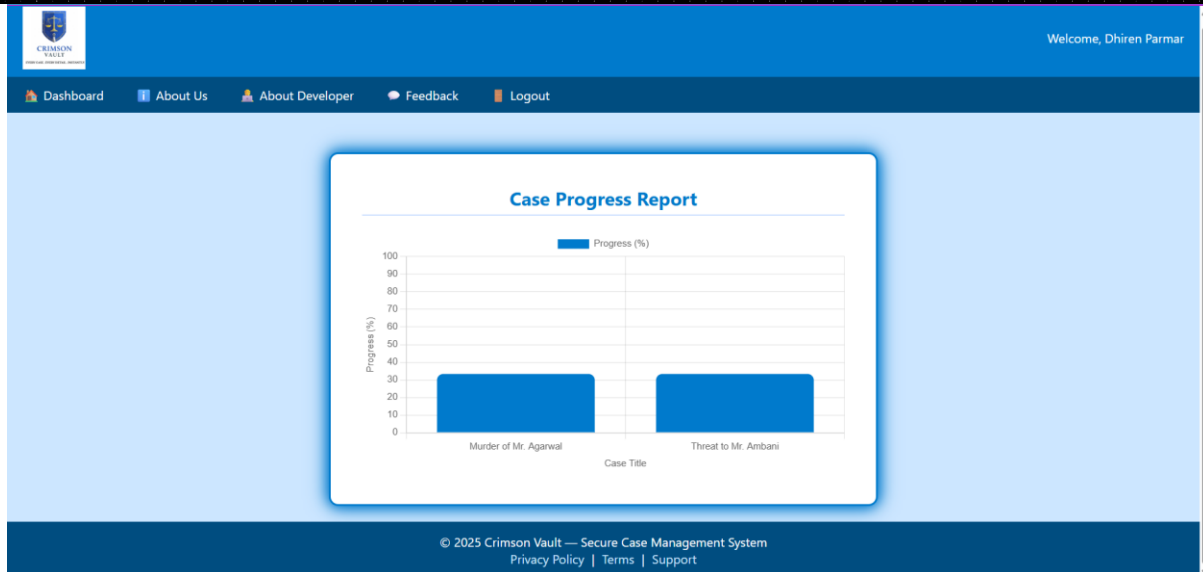
This is progress.php where the police officer can see the percentage of progress made in their assigned cases in form of a bar graph. Currently it is basic model so it have static progress percentage. It depends on the status of the case. The following are percentage of the status of an case :-

- Open – **33.33%**
- Under trial – **66.66%**
- Closed – **100%**

This is accussed.php file where officer can add accussed of the case with details as following :- Related case, Full name, Father's name, Gender, Birthdate, Height, Weight, Skin Color, Blood group and current charges.
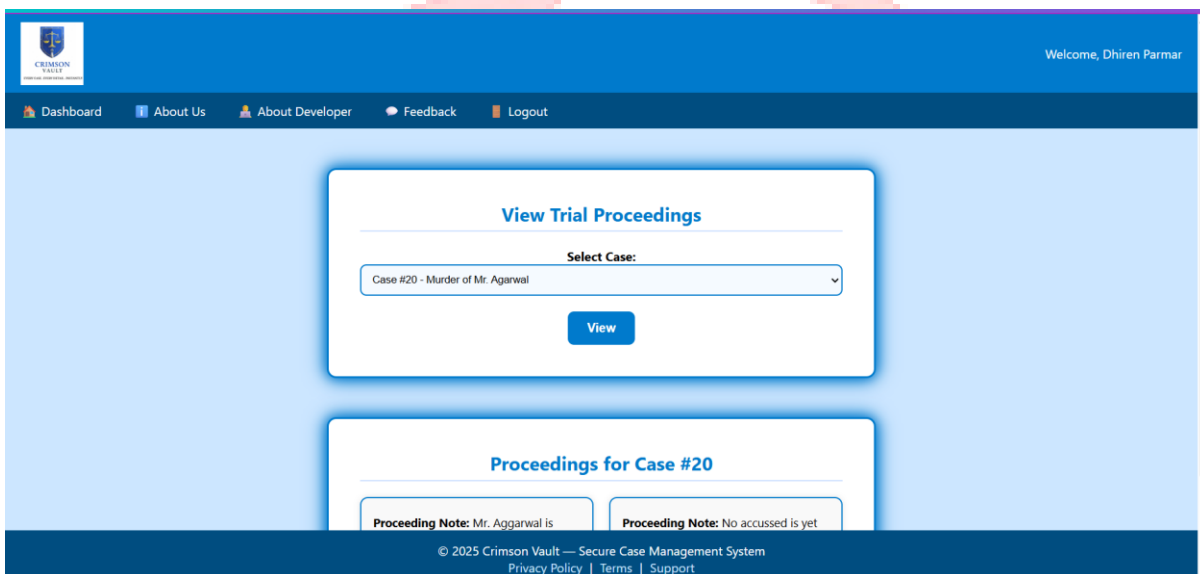
This is view_accussed.php where officer can view and edit their uploaded accused of assigned case.
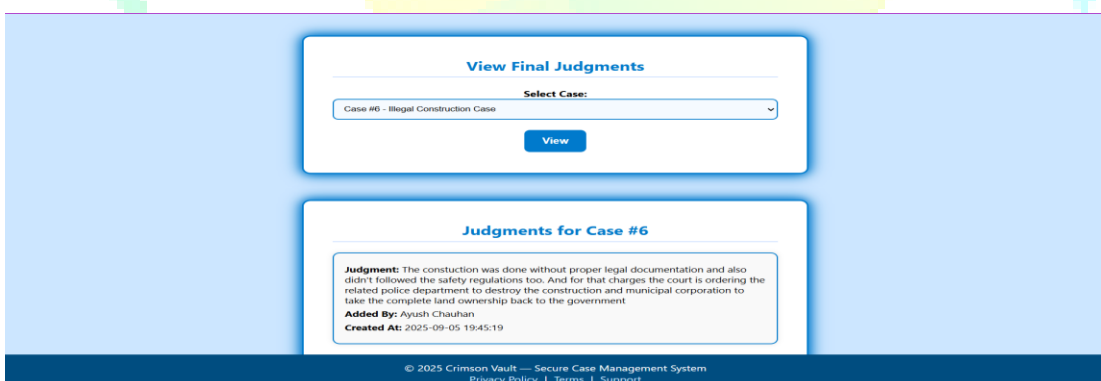
This is view_proceedings.php where officer can see the trial results of the case.

This is view_judgements.php where final judgement of a case can be viewed.

This is search_accussed.php file where officer can search if a person is accused to any particular case or not.

This is new_user.php file where police officer can add new user with following details :- Full name, email, phone number, password and role (police, legal personnel)

The previous page's last screenshot was notifications.php where user can see the notifications sent by the admin.
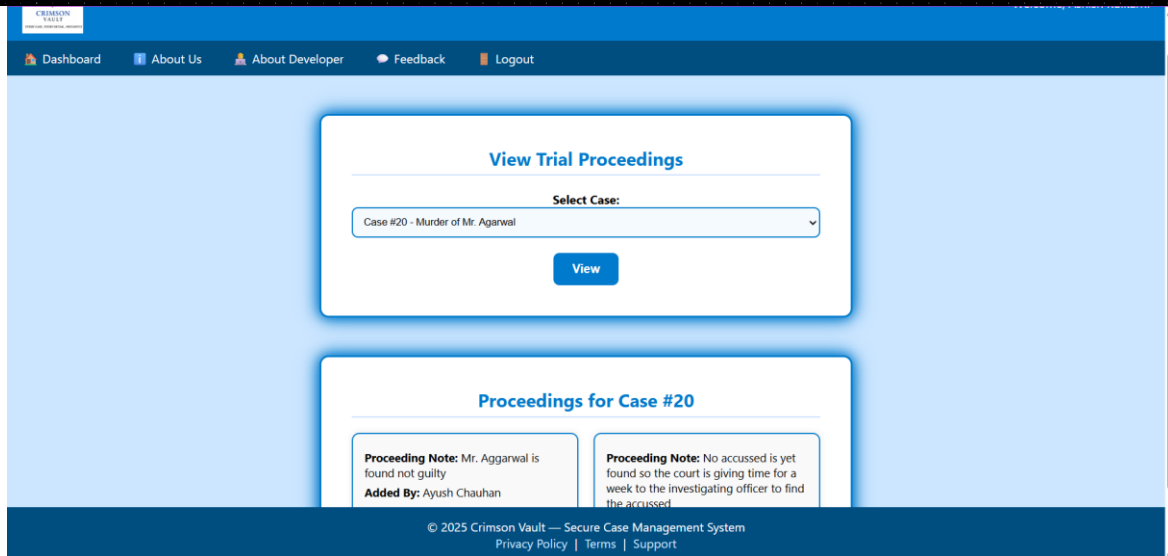
Now we will move forward to Legal Personnel's perspective of this website :-
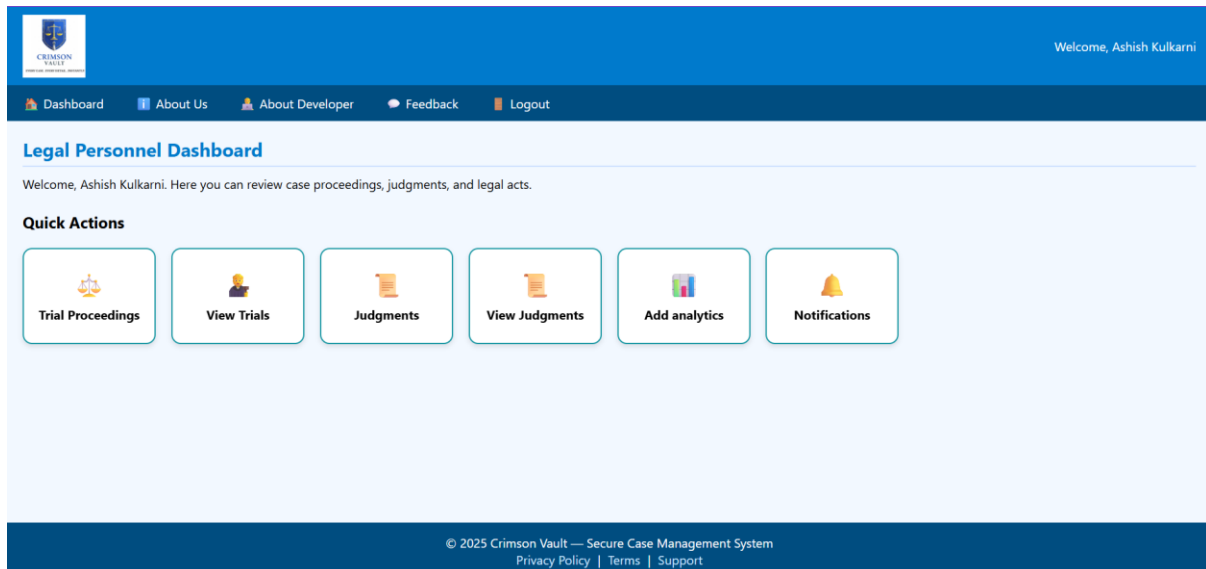
This is legal personnel's main dashboard (legal_dashboard.php) where legal personnel can access the functionalities of :- Adding trial proceedings, view trials, add final judgement, view final judgement, add analytics to a case, and notifications.
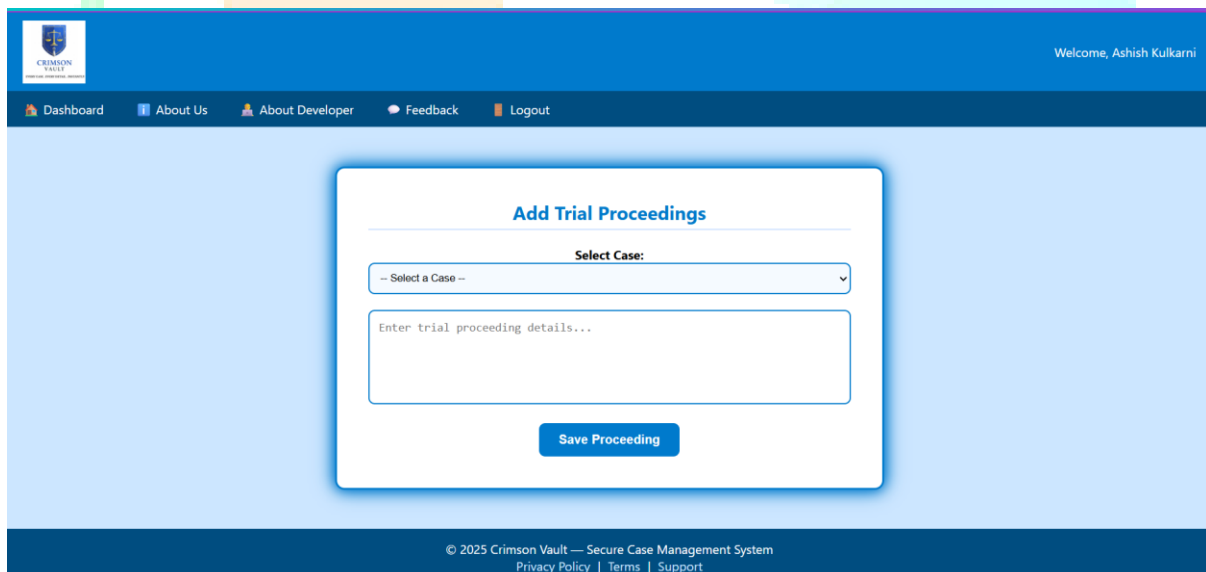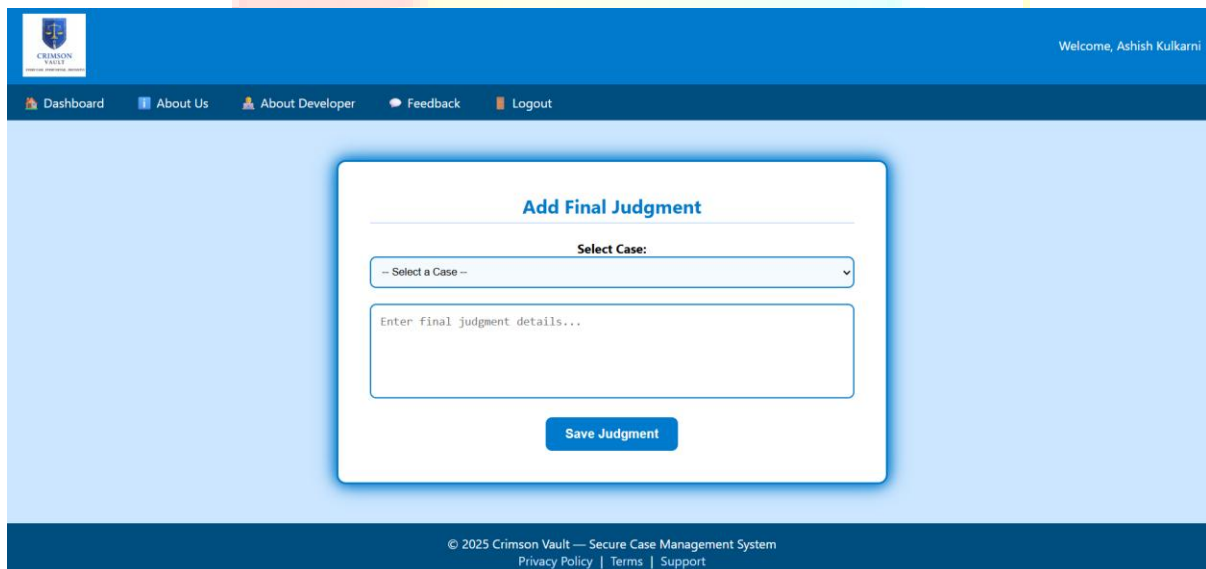
This is trial_proceedings.php where a legal personnel can add the trial update of each and every case.

This is again view_proceedings.php where legal personnel can see the trial results of cases.
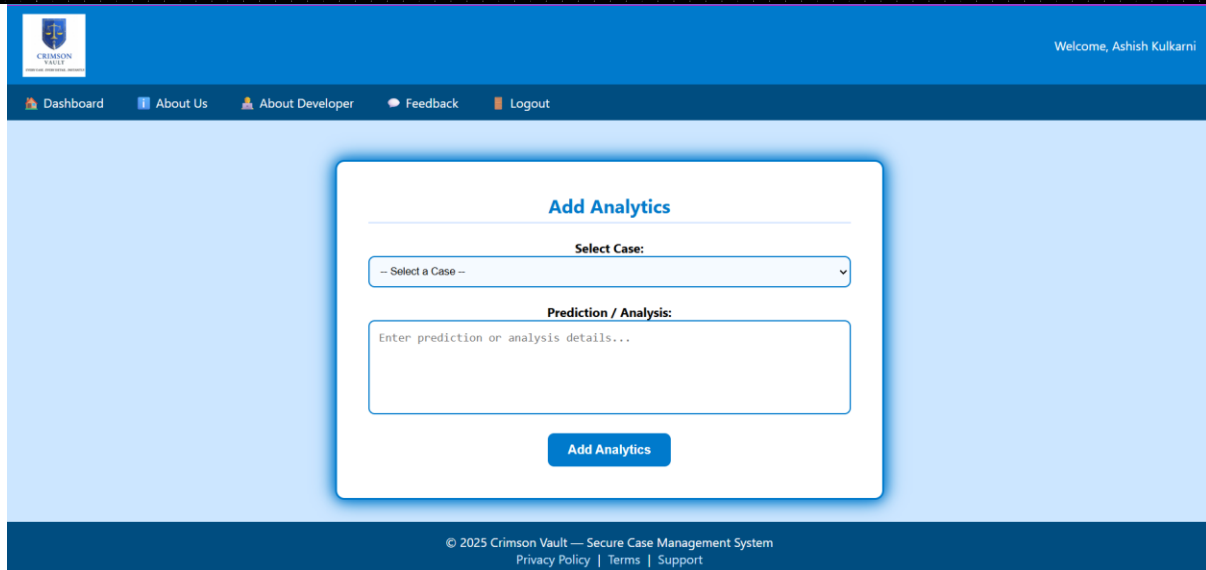
This is judgements.php where final judgement of the case can be added
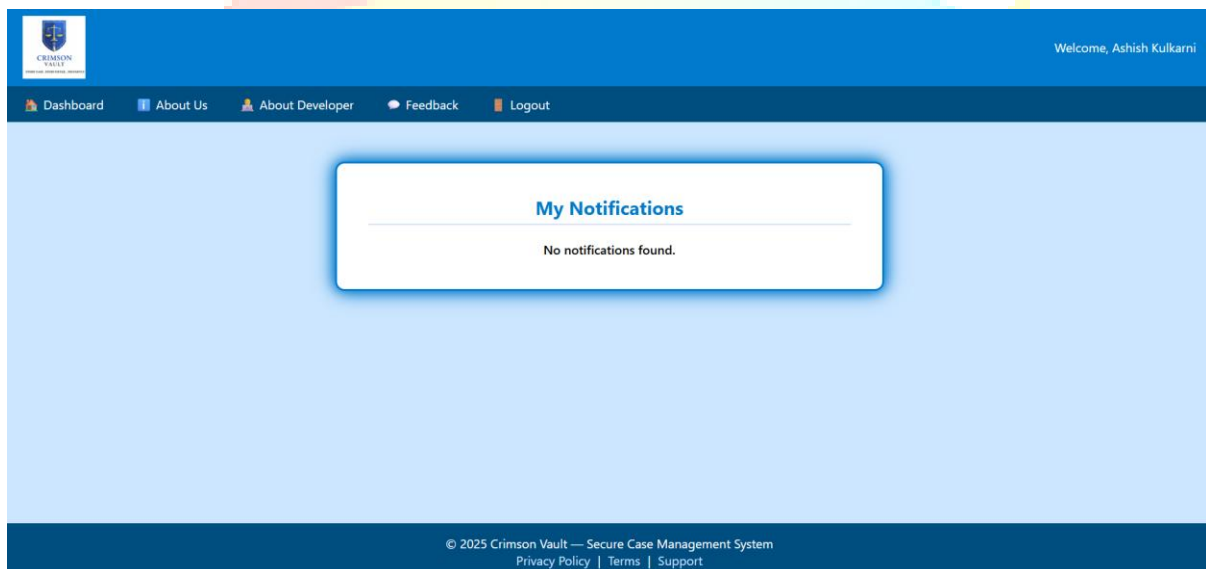
This is again view_judgements.php where final judgement of the case can be seen.

This is add_analytics.php where analysis or prediction of an case can be entered.

And this is again notifications.php where notifications from the admin can be seen

Now we will move forward to pages included in header and footer which is included in every page that can be accessed by police officer or legal personnel.
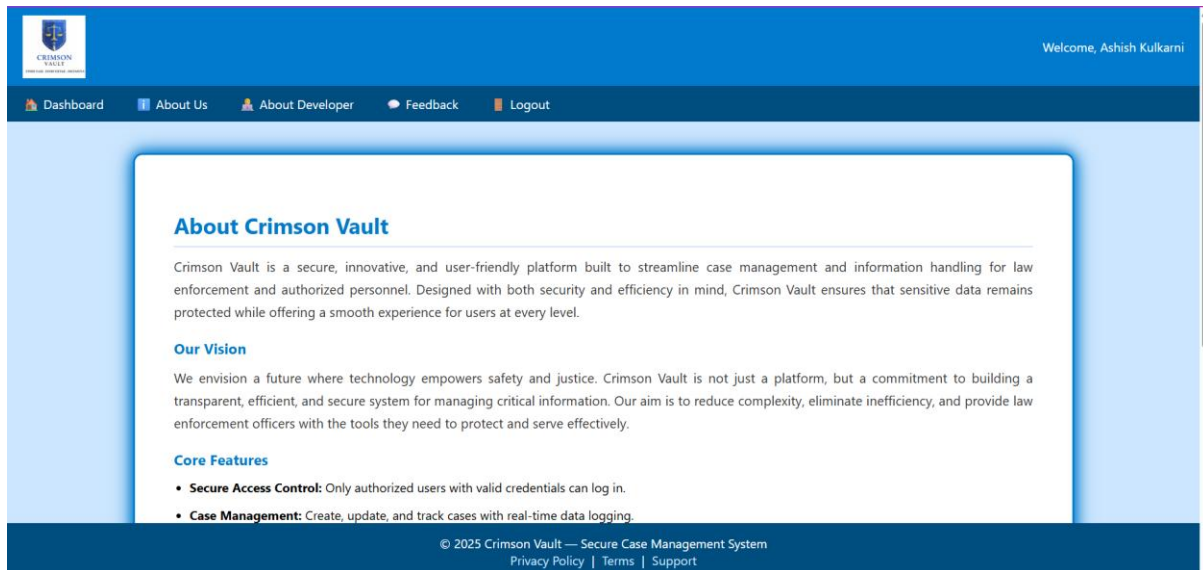
This is about.php where the motive, vision, core features, and commitment is shown.
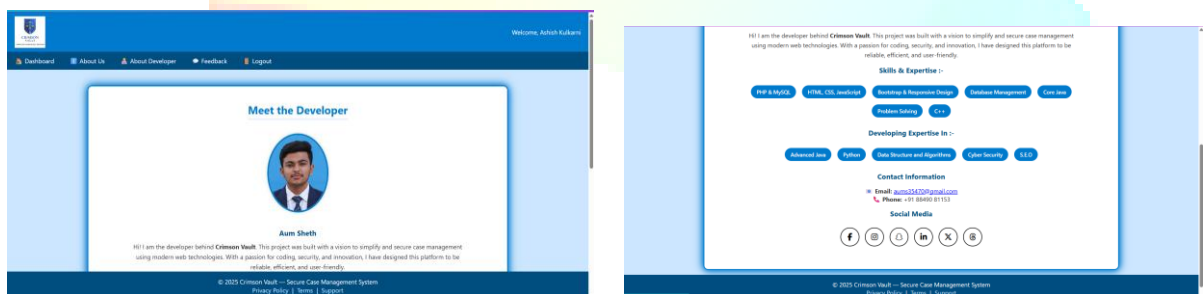
This is developer.php where a small introduction about developer, skiils & expertise, developing expertise, contact information and social media links is given.
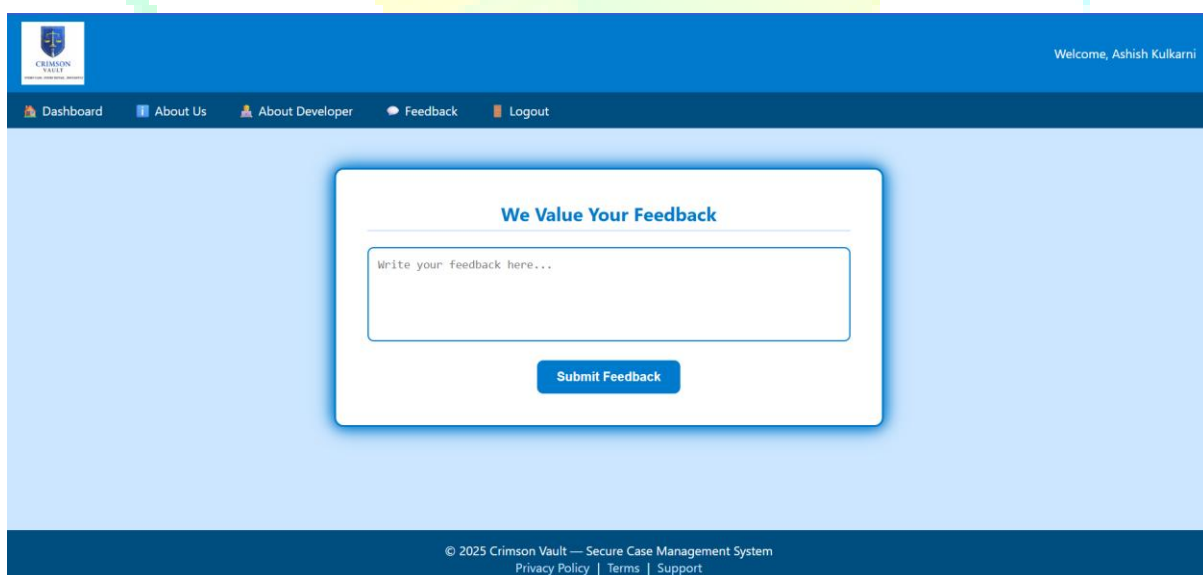
This is feedback.php for giving feedback of the website.

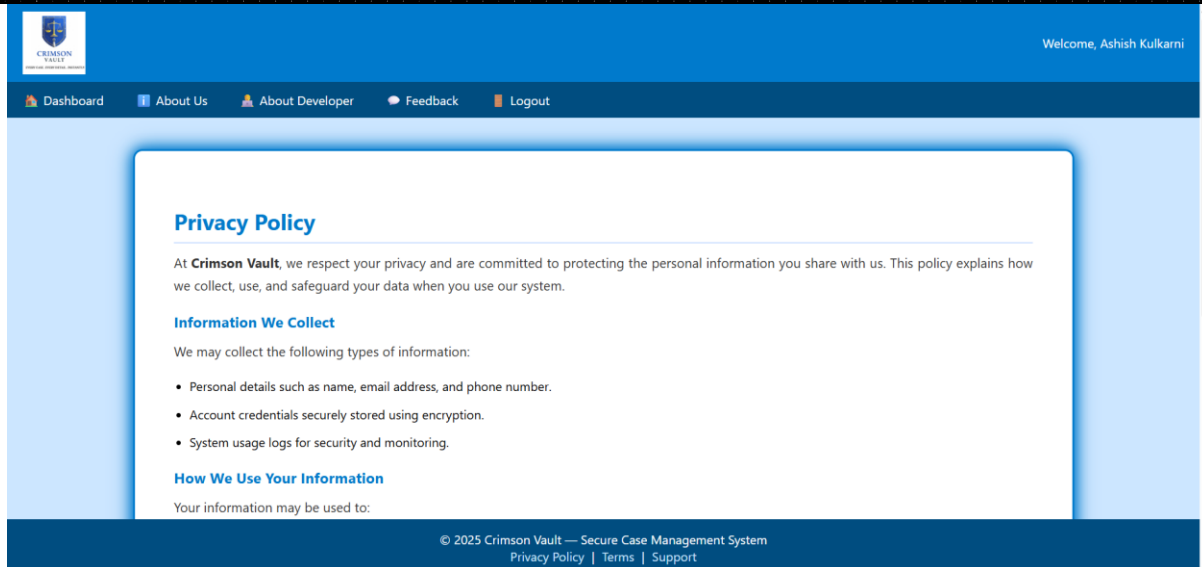This is privacy.php information collected, how information is used, data protection, user's rights, and changes of the policy condition is mentioned.

This is terms.php where acceptance of terms, user responsibilities, restrictions, limitation of liability, ammendments and contact is mentioned.

The last image of the previous page is support.php where user can submit the bug or problems faced in the website.

Now we will move forward to the forgot password pages :-

In this page user will verify the email and an otp will be sent to the entered email

This page is for verification of otp. The otp is valid only for 3 minutes.

Note :- This page will also appear while logging in as the website have Multi Factor Authentication feature so an OTP is sent both the times(login and forget password).
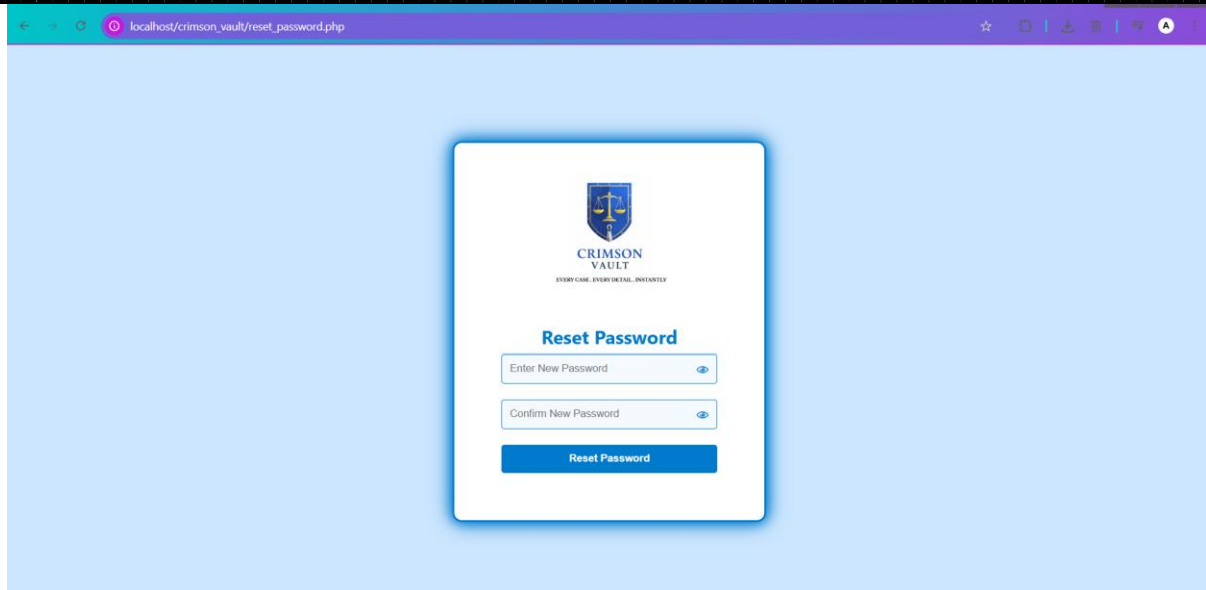
This page will enable the user to enter the new password.

4.2 Code Snippets with Explanation :-

A new feature has been implemented that allows police officers to monitor the progress percentage of their assigned cases.

```php
//1.Fetching case data
while ($row = $result->fetch_assoc()) {
    $caseTitles[] = $row['title'];
    //2.Mapping status to progress percentage
    // Map status → percentage
    switch (strtolower($row['status'])) {
        case 'open':
            $caseProgress[] = 33.33;
            break;
        case 'under trial':
            $caseProgress[] = 66.66;
            break;
        case 'closed':
            $caseProgress[] = 100;
            break;
        default:
            $caseProgress[] = 0;
    }
}
```

```
<script>
        const ctx =           //4.chart.js visualization
document.getElementById('caseProgressChart').getContext('2d');
        const chart = new Chart(ctx, {
            type: 'bar',
            data: {
                labels: <?php echo json_encode($caseTitles); ?>,
                datasets: [{
                    label: 'Progress (%)',
                    data: <?php echo json_encode($caseProgress); ?>,
                    backgroundColor: '#007acc',
                    borderRadius: 8
                }]
            },
            options: {    //5.Chart configuration
                responsive: true,
                scales: {
                    y: {
                        beginAtZero: true,
                        max: 100,
                        title: {
                            display: true,
                            text: 'Progress (%)'
                        }
                    },
                    x: {
                        title: {
                            display: true,
                            text: 'Case Title'
                        }
                    }
                }
            }
        });
    </script>
```

Explanation :-
This module generates a **visual progress report** of the most recent cases assigned to a police officer. It retrieves case information from the database, converts the **case status** into a **progress percentage**, and finally displays the data in a **bar chart** using Chart.js.

1. Fetching Case Data :-
- Loops through each case record fetched from the database.
- Stores the case titles in an array ($caseTitles) to be used as labels on the chart.

2. Mapping Status to Progress Percentage
- The status of each case is mapped to a corresponding progress percentage:
  - Open → 33.33% (case just started)
  - Under Trial → 66.66% (case in progress)
  - Closed → 100% (case completed)
- If the status is unknown, progress defaults to 0%.
- Percentages are stored in the $caseProgress array.

3. Passing Data to JavaScript
labels: <?php echo json_encode($caseTitles); ?>,
data: <?php echo json_encode($caseProgress); ?>,

- PHP arrays ($caseTitles and $caseProgress) are converted into JSON format so they can be used in JavaScript.
- This ensures server-side data is passed correctly to the client-side chart.

4. Chart.js Visualization
- Creates a bar chart using Chart.js.
- Each bar represents a case, labeled by its title.
- Bar height corresponds to the progress percentage.
- Style:
  o Blue color (#007acc) for bars.
  o Rounded edges (borderRadius: 8) for a modern look.

5. Chart Configuration
- Responsive Design → Chart adjusts to screen size.
- Y-axis (Progress %):
  o Starts at 0.
  o Ends at 100.
  o Label: *Progress (%)*.
- X-axis (Case Title):
  o Displays the case titles as labels.

4.3 Testing :-

Testing was carried out to ensure the *Crimson Vault Project* performs as expected. The testing strategy included both **white-box testing** (code-level verification) and **black-box testing** (functional verification).

The main modules tested were:
- **Authentication** (login, registration, password reset with OTP)
- **Case Management** (add, update, view, progress tracking)
- **Analytics and Dashboard** (progress charts, officer dashboards, notifications)
- **Evidence & Document Upload**
- **UI and Navigation** (links, forms, responsiveness)

White Box Testing :-

This testing was carried out by the project guide of mine **(Prof. Bhavesh K Hadiyal)**

Tests Performed:
1. **Login Verification (login.php, index.php)**
   o Tested SQL queries for fetching user by email and password.
   o Verified session variables ($_SESSION['officer_id']) are set after login.
   o Ensured incorrect credentials do not bypass authentication.
   Result: Working correctly, invalid users are blocked.
2. **Password Reset (reset_password.php, verify_otp.php, generate_otp.php)**
   o Traced OTP generation and deletion from password_resets table.
   o Verified password update logic uses prepared statements to avoid SQL injection.
   o Checked redirect after successful reset.
   Result: Secure update; redirection works.
3. **Case Progress (progress.php)**
   o Verified SQL query limits to **5 most recent cases** per officer.
   o Mapped case status → progress percentage (Open=33.33, Under Trial=66.66, Closed=100).

o Confirmed JSON encoding correctly passes data to Chart.js.
   Result: Chart displays accurate percentages.

4. **Evidence Upload (upload_evidence.php)**
   o Checked file upload path (/codes/uploads/) and file type validation.
   o Ensured file is linked with correct case_id.
   Result: Upload and retrieval successful.

Black Box Testing :-

This testing was carried out by a friend of mine named **Heet H. Doshi**

Tests Performed :

1. **Login Module**
   o (Worked) Entering a valid email and password successfully redirected the officer to the dashboard.
   o (Not Worked) Entering a wrong password displayed an error message and access was denied.
   o (Worked) Leaving fields blank showed a validation error.
   Result: Login module works correctly and prevents unauthorized access.

2. **Password Reset**
   o (Worked) Requesting an OTP with a registered email successfully delivered an OTP.
   o (Worked) Entering the correct OTP and matching new passwords updated the password and redirected to the login page.
   o (Not Worked) Entering an incorrect OTP resulted in an error and blocked the reset process.
   o (Not Worked) Entering mismatched passwords was rejected with an error message.
   Result: Password reset is secure and functions as expected.

3. **Case Management**
   o (Worked) Adding a new case with all fields filled correctly saved the case into the database.
   o (Worked) Editing case status from **Open → Under Trial → Closed** updated the status in the database.
   o (Worked) Only the 5 most recent cases were displayed when more than 5 existed.
   Result: Case management works reliably, enforcing proper status updates and limits.

4. **Analytics & Progress Chart**
   o (Worked) The chart displayed progress correctly for cases based on their statuses.
   o (Worked) Status-to-progress mapping was verified: **Open = 33.33%**, **Under Trial = 66.66%**, **Closed = 100%**.
   o (Worked) Chart resized correctly on different screen sizes, maintaining readability.
   Result: Progress chart is accurate and responsive.

5. **Evidence Upload**
   o (Worked) Uploading valid files (PDF/DOC) successfully stored them and linked to the correct case.
   o (Not Worked) Uploading unsupported file types (e.g., EXE) or oversized files was blocked with an error.
   Result: Evidence upload is secure and enforces file validation.

6. **UI and Navigation**
   - o (Worked) All dashboard links (Dashboard, Cases, Analytics, Logout) navigated to the correct pages.
   - o (Not Worked) CSS styling was consistent across pages with proper fonts, colors, and layouts.
   - o (Worked) Application displayed correctly on mobile screens with responsive design.
     Result: UI and navigation are user-friendly and responsive.

4.4 Bug Reports :-

1. Password Visibility Toggle Not Working in Reset Page
- **Bug ID:** BUG-001
- **Module:** Reset Password (reset_password.php)
- **Severity:** Medium
- **Description:** Password toggle (eye icon) did not work when entering a new password during reset.
- **Steps to Reproduce:**
  1. Open Reset Password page.
  2. Enter a new password.
  3. Click on the eye icon to toggle visibility.
- **Expected Result:** Password field should toggle between hidden (password) and visible (text).
- **Actual Result:** Nothing happens when clicking the icon.
- **Status:** Fixed
- **Fix Applied:** Added id attributes to input fields and updated togglePassword() function in script.js.

2. Officer ID Not Retrieved in Progress Chart
- **Bug ID:** BUG-002
- **Module:** Case Progress (progress.php)
- **Severity:** High
- **Description:** Case progress chart did not load because officer_id was not defined.
- **Steps to Reproduce:**
  1. Login as officer.
  2. Navigate to Progress page.
  3. Page shows no data.
- **Expected Result:** Last 5 cases of the logged-in officer should be displayed.
- **Actual Result:** Query returned empty because officer_id variable was missing.
- **Status:** Fixed
- **Fix Applied:** Used $_SESSION['officer_id'] to retrieve the logged-in officer's ID.

3. Back to Login Link Unnecessary After Reset
- **Bug ID:** BUG-003
- **Module:** Reset Password (reset_password.php)
- **Severity:** Low
- **Description:** After successful password reset, user was shown "Back to Login" link instead of being redirected automatically.
- **Steps to Reproduce:**
  1. Reset password with valid OTP.

2. Observe "Back to Login" link.
- **Expected Result:** User should be redirected automatically to index.php.
- **Actual Result:** Manual link had to be clicked.
- **Status:** Fixed
- **Fix Applied:** Added JavaScript redirection after successful reset.

4. Case Progress Percentage Mapping
- **Bug ID:** BUG-004
- **Module:** Progress Chart (progress.php)
- **Severity:** Medium
- **Description:** Progress percentages were initially mapped incorrectly (Open = 20%, Under Trial = 60%).
- **Steps to Reproduce:**
    1. View progress chart.
    2. Compare case status with percentage.
- **Expected Result:** Open = 33.33%, Under Trial = 66.66%, Closed = 100%.
- **Actual Result:** Chart showed wrong values.
- **Status:** Fixed
- **Fix Applied:** Adjusted mapping logic in PHP switch statement.

5. File Upload Validation
- **Bug ID:** BUG-005
- **Module:** Evidence Upload (upload_evidence.php)
- **Severity:** High
- **Description:** Users could attempt to upload unsupported file formats or oversized files.
- **Steps to Reproduce:**
    1. Upload .exe file.
    2. Upload very large file.
- **Expected Result:** File should be rejected with an error.
- **Actual Result:** System allowed attempt without proper validation.
- **Status:** Fixed
- **Fix Applied:** Implemented file type and size validation before saving.

Bug Report Summary
- **Total Bugs Found:** 5
- **Fixed:** 5
- **Pending:** 0
- **Critical Issues:** None remaining
- **System Stability:** Stable after fixed

4.5 Comparison with Objectives :-

Objective	Planned Outcome	Actual Outcome (After Testing & Fixes)	Status
1. Secure Login & Authentication	Provide secure login system with support for MFA (OTP Email).	Implemented successfully. Email OTP works reliably.	Achieved
2. Role-based Access (Police Officer / Admin)	Officers should access only their assigned cases, while admins have full access.	Session-based officer ID used for filtering cases. Officers can view only their cases. Admin access validated.	Achieved
3. Case Management System	Officers should be able to add, update, and edit case details.	Functionality works correctly after testing. Minor toggle issue in edit form fixed.	Achieved
4. Progress Monitoring (Chart Representation)	Officers should view the percentage progress of their cases (open, under trial, closed).	Chart implemented with correct percentage mapping and enlarged UI for better readability.	Achieved
5. Password Management	Users should be able to reset forgotten passwords securely.	Password reset implemented with session-based email validation. Toggle bug fixed. User is auto-redirected to login after reset.	Achieved
6. Evidence Handling (File Uploads)	System should allow uploading supporting documents/images securely.	File size & type validation added after testing. Malicious uploads blocked.	Achieved
7. System Usability (UI/UX)	Interface should be user-friendly, clear, and responsive.	CSS improved for charts, forms, and login pages. Tested across devices.	Achieved
8. Deployment & Local Testing (XAMPP)	System should run smoothly in local development environment.	Minor port conflict in XAMPP resolved by changing MySQL port. System stable afterwards.	Achieved

5. Conclusion and Future Work :-

5.1 Summary of work :-

The "***Crimson Vault***" is a web-based case management system designed to streamline the handling of legal and law enforcement processes. It provides secure, role-based dashboards for different stakeholders, including police officers, legal professionals, and administrators, ensuring efficient coordination and transparency in case proceedings.

The system was developed using **PHP, MySQL, HTML, CSS, and JavaScript**, with **PHPMailer** integration for secure email communication and OTP verification. The application architecture supports modular functionality, making it adaptable to different use cases in the justice and law enforcement sectors.

Key features include:

- **Case Management** – Add, update, and track cases, including related judgments, trial proceedings, and evidence records.
- **User Management** – Secure registration, login, password reset, and OTP-based verification for account safety.
- **Evidence Handling** – Upload and review case-related documents and reports in digital format.
- **Dashboards & Analytics** – Role-specific dashboards (police and legal) with analytics to monitor progress, track patterns, and improve decision-making.
- **Notifications & Support** – Inbuilt feedback, support, and notification modules to ensure smooth communication.
- **Data Security** – Structured SQL database for managing case data with privacy and terms modules for compliance.

The project also includes a sample database schema (crimson_vault.sql) and uploaded case documents (e.g., cyber fraud, property disputes, crime scene reports), which demonstrate the system's ability to handle real-world case files.

Overall, the Crimson Vault Project demonstrates the integration of secure web technologies with case management workflows, providing a reliable platform that can be expanded for wider adoption in digital law enforcement and judicial processes.

5.2 Challenges Faced :-

During the development of the "***Crimson Vault***", several challenges were encountered, each contributing to valuable learning outcomes:

1. **Database Design & Normalization**
   - Creating a structured database that could handle different case types, users, and evidence records without redundancy required multiple iterations. Balancing efficiency with flexibility was a key challenge.
2. **User Authentication & Security**
   - Implementing OTP verification and secure password reset functions using **PHPMailer** involved overcoming issues with email server configuration and ensuring that sensitive user data remained protected.
3. **File Uploads & Evidence Management**
   - Designing a secure system to handle document uploads (e.g., case reports, evidence files) required attention to file validation, storage, and access control to prevent misuse or corruption.

4. **Role-Based Dashboards**
   - Developing separate dashboards for police officers, legal professionals, and administrators demanded careful access control so that each role only viewed relevant information.
5. **UI/UX Consistency**
   - Ensuring a user-friendly and consistent interface across multiple modules (cases, judgments, proceedings, analytics) required extra testing and refinement of styles and layouts.
6. **Integration & Testing**
   - Bringing together different modules such as case management, evidence uploads, notifications, and analytics into a seamless system introduced integration challenges. Debugging and testing were essential to maintain stability.
7. **Scalability Considerations**
   - While building the prototype, ensuring that the system could be scaled to handle larger datasets and more users was an ongoing design challenge.

5.3 Future improvements :-

While the current version of the **Crimson Vault Project** provides a strong foundation for digital case management, there are several areas where enhancements can further increase its effectiveness and usability:

1. **Photo Integration for Accused Profiles**
   - Attach profile photos to accused records to aid in better identification and case referencing.
2. **Enhanced Progress Bar Reporting**
   - Introduce more logical and detailed progress tracking for each case, reflecting milestones such as investigation, trial, judgment, and closure.
3. **Categorization of Legal Personnel**
   - Add structured categories for legal professionals (e.g., judges, prosecutors, defense attorneys) to improve clarity in case roles and responsibilities.
4. **Advanced Search & Filtering**
   - Implement powerful search and filter options across cases, evidence, and judgments to make information retrieval faster and more precise.
5. **AI-Assisted Case Insights**
   - Integrate basic AI tools to analyze past cases and provide insights, such as predicting case durations or identifying patterns in criminal activity.
6. **Multilingual Support**
   - Expand accessibility by offering support for multiple languages, ensuring usability across diverse regions and communities.
7. **Integration with External Systems**
   - Enable secure data exchange with other government or judicial systems, such as police databases or court record systems, for smoother collaboration.
8. **OTPs on mobile number**
   - Currently OTPs are send only via email. In future updates OTPs on mobile number will be enabled.
9. **Collaborative Working on case :-**

   - In future the website will allow the police officer to work with collaboration by seeking the permission of assigned police officers.

5.4 References :-

1. Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley.
2. Ullman, L. (2019). *PHP for the Web: Visual QuickStart Guide* (5th ed.). Peachpit Press.
3. MySQL Documentation. (2023). *MySQL 8.0 Reference Manual*. Oracle. Retrieved from https://dev.mysql.com/doc/
4. Mozilla Developer Network (MDN). (2023). *HTML, CSS, and JavaScript Documentation*. Retrieved from https://developer.mozilla.org/
5. GitHub – PHPMailer. (2023). *PHPMailer Documentation*. Retrieved from https://github.com/PHPMailer/PHPMailer
6. National Institute of Standards and Technology (NIST). (2001). *FIPS 197: Advanced Encryption Standard (AES)* — specification for AES-128, AES-192, and **AES-256**. Retrieved from https://csrc.nist.gov/publications/detail/fips/197
7. OWASP Foundation. (2021). *OWASP Top Ten — The Ten Most Critical Web Application Security Risks (2021)*. Retrieved from https://owasp.org/www-project-top-ten/

5.5 Bibliography :-

- MySQL. (2023). *MySQL 8.0 Reference Manual*. Oracle. Retrieved from https://dev.mysql.com/doc/
- PHPMailer. (2023). *PHPMailer Documentation*. GitHub repository. Retrieved from https://github.com/PHPMailer/PHPMailer
- Google Chrome Developers. (2023). *Google Chrome Developer Documentation*. Retrieved from https://developer.chrome.com/
- Ullman, L. (2019). *PHP for the Web: Visual QuickStart Guide* (5th ed.). Berkeley, CA: Peachpit Press.
- Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development* (5th ed.). Boston, MA: Addison-Wesley.
- OpenAI. (2025). *Assistance in drafting documentation for Crimson Vault Project (ChatGPT – GPT-5)*.