A REPORT

ON

CASE CODE MAPPER: UNRAVELING LEGAL SECTIONS

The project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

By

Names of the student's	Registration No.
Jahnavi Ponaganti	AP21110011364
Ujwala Matha	AP21110011366
Varnitha Chinthala	AP21110011407
Chinni Yandapalli	AP21110011409

Under the Guidance of

Dr. Sanjay Kumar

SRM University-AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh - 522 240

May, 2024

Certificate

Date: 01-May-2024

This is to certify that the work present in this Project entitled "**CASE CODE MAPPER - UNRAVELLING THE LEGAL SECTION**" has been carried out by **Varnitha Chinthala, Chinni Yandapalli, Ujwala Matha, Jahnavi Ponaganti** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in the**School of Engineering and Sciences**.

Supervisor

Dr. Sanjay Kumar

Assistant Professor.

Acknowledgments

We would like to thank Dr.Sanjay Kumar, our Professor-in-charge for his supportand guidance in completing our Project on the topic Case Code Mapper - Unravelling the Legal Codes. It was a great learning experience.

We would like to take this opportunity to express gratitude to all of our group members with one another. The project would not have been successful without the group's cooperation and input.

Finally, I would like to thank our Professor Dr.Sanjay Kumar for supporting the project and giving his valuable insights. We are also thankful for the resources andreferences used which finally made us complete the project.

Table of Contents

Certificate……i

Acknowledgments……..ii

Table of Contents……...iii-iv

Abbreviations…….….v

List of Tables…….….vi

List of Figures…….…vii

Abstract……...viii

1. Introduction……1

1.1 Description of the Project……...2-3

2. Methodology……4

2.1 SRS Documentation……4-19

2.2 Functional and Non-Functional Requirements……...19-22

2.2 Technology used……22

3. Designing……23-50

3.1 Data Flow Diagrams……....23-29

3.1.1 Level 0 DFD……23-24

3.1.2 Level 1 DFD……25-26

3.1.3 Level 2 DFD……27-29

3.2 Data Dictionary……29-30

3.3 Use Case Diagram……31-34

3.4 Entity Relationship Diagram……34-35

3.5 Class Diagram……..36-37

3.6 Activity Diagram……..38-39

3.7 State Diagram……...39-40

3.8 Collaboration Diagram……40-42

3.9 Description of Frontend and Backend Development……...42-43

3.10 Interfaces of Frontend and Backend…….....44-50

4. Testing…….....51-59

4.1 Functional Testing……...51-52

4.1.1 Login Module…….....51

4.1.2 Result Module……..51

4.1.3 Main Module……52

4.2 Path Coverage Testing…….53-59

4.2.1 Start Application…….53

4.2.2 Navigate to Registration Form……53

4.2.3 Access Registration Form……53

4.2.4 Have Account……...54

4.2.5 Create Account Form……54

4.2.6 Enter Email and Password……54

4.2.7 Enter Data……...55

4.2.8 Save Data……55

4.2.9 Process Login…….55

4.2.10 Validate User Access……..56

4.2.11 Select Below one(blogs/prompts) ……56

4.2.12 Enter Data…….57

4.2.13 Submitting Data……...57

4.2.14 Save Data……..58

4.2.15 Display Query……..58

4.2.16 Stop……...59

5. Future Work……60

6. Limitations……..61

7. Conclusion……..62

References……..63

Abbreviations

SRS: Software Requirement Specification

DFD: Data Flow Diagram

ERD: Entity Relationship Diagram

IPC: Indian Penal Code

CRPC: Code of Criminal Procedure

FIR: First Information Report

IPC Sections: Specific sections of the Indian Penal Code, often referred to by their section numbers (e.g., IPC Section 302 for murder)

POCSO: Protection of Children from Sexual Offences Act

Dowry Prohibition Act: Dowry Prohibition Act, which addresses dowry-related offenses

NCRB: National Crime Records Bureau

CCTV: Closed-circuit television, often used for surveillance and crime prevention **DNA:** Deoxyribonucleic acid, used in forensic investigations for identifying individuals**POTA:** Prevention of Terrorism Act

NIA: National Investigation Agency

CBI: Central Bureau of Investigation

MO: Modus Operandi, referring to the method of operation used by criminals in committing crimes

MCOCA: Maharashtra Control of Organised Crime Act

IPC Amendment Acts: Acts that amend or update specific sections of the Indian Penal Code, such as the Criminal Law (Amendment) Act, 2013.

These acronyms and abbreviations are commonly used in legal and law enforcement contextsin India and there are many more that cannot able to derived from all.

List of Tables

Table 1. Data Dictionary for DFD……..30

Table 2. Output 1 of the IPC Sections for the input 1 problem statement……..50

Table 3: Output 2 of the IPC Sections for the input 2 problem statement……..50

List of Figures

Figure 1: Level 0 Data Flow Diagram…….23
Figure 2: Level 1 Data Flow Diagram…….25
Figure 3: Level 2 Data Flow Diagram…….27
Figure 4: Use case Diagram…….31
Figure 5: Entity Relationship Diagram……34
Figure 6: Class Diagram……..36
Figure 6.1: Activity Diagram……...38
Figure 7: State Chart Diagram……..39
Figure 8: Collaboration Diagram……..40
Figure 9: Start application of the control flow diagram……53
Figure 10: Navigate to the Registration Form of the control flow diagram……..53
Figure 11: Access the Registration Form of the control flow diagram…….53
Figure 12: Have an Account of the control flow diagram…….54
Figure 13: Create an Account Form of a control flow diagram…….54
Figure 14: Enter the Email and Password of the control flow diagram…….54
Figure 15: Enter the Data of the control flow diagram……..55
Figure 16: Save Data of control flow diagram……...55
Figure 17: Process Login of Control Flow Diagram……..55
Figure 18: Validate user Access to the control flow diagram……....56
Figure 19: Blogs/prompts of control flow diagram……56
Figure 20: Enter Data of control flow diagram……..57
Figure 21: Submitting data of control flow diagram……..57
Figure 22: Save Data of control flow diagram……...58
Figure 23: Display Query of control flow diagram……58
Figure 24: Stop of Control flow diagram……...59

Abstract

The Full Stack Project, titled “Case Code Mapper: Unraveling Legal Section” offers a comprehensive solution to facilitate the process of identifying the legal articles or sections applicable to the problems or cases stated. Through a user-friendly interface, the tool allows individuals to articulate legal problems in plain language subsequently suggesting relevant legal sections. In the backend to produce the legal articles we used matching algorithms according to the text given by the client.

Our project is a dynamic web-based solution crafted to simplify access to legal information. It endeavors to bridge the gap between complex legal texts and users by swiftly processing problem descriptions and suggesting relevant legal sections. Employing front-end technologies for user interaction, and backend systems for data processing this project ensures efficient legal data handling. Users can input details of court cases or relevant keywords, and the system retrieves and presents pertinent legal sections promptly, empowering users to make informed decisions. Our project seeks to revolutionize legal research, fostering greater accessibility and understanding within the legal landscape.

1. Introduction

Introducing the "Case Code Mapper: Unraveling Legal Section," a groundbreaking Full Stack Project poised to redefine the landscape of legal research. This transformative web-based application marries cutting-edge technology with legal expertise to provide users with unparalleled accessibility and comprehension within the legal domain. At its core, the project aims to simplify the often convoluted process of legal research by empowering users to articulate complex legal issues in plain language. By leveraging sophisticated AI generative models and dynamic matching algorithms, the system offers prompt and precise suggestions of applicable legal sections, effectively bridging the gap between intricate legal texts and end-users. The "Case Code Mapper" functions as a comprehensive and unified system, seamlessly integrating interconnected components to deliver a seamless user experience. Its intuitive web-based platform not only provides access to relevant legal sections but also supports additional functionalities such as viewing incident reports and displaying IPC section codes, thereby extending its utility across various legal scenarios.

In terms of infrastructure, the project encompasses a robust hardware setup, including servers, databases, user interfaces, and communication links, all working in harmony to facilitate seamless interaction with the backend systems. The system prioritizes data security, employing secure networking protocols to ensure the confidentiality and integrity of legal data throughout the communication process. Furthermore, the project boasts impressive hardware specifications, with ample memory and storage components designed to support the intensive processing and storage requirements of legal information. This scalability ensures the system's adaptability and efficiency as it continues to grow and evolve. In essence, the "Case Code Mapper: Unraveling Legal Section" represents a paradigm shift in legal research and decision-making, ushering in a new era of efficiency, precision, and accessibility within the legal domain. By democratizing access to legal knowledge and expertise, this project promises to revolutionize the way legal professionals and laypersons alike navigate the complexities of the legal landscape.

1.1 Description of the project

The "Case Code Mapper: Unraveling Legal Section" is an ambitious web-based application that revolutionizes the landscape of legal research and decision-making. Designed as a comprehensive and interconnected system, it empowers users to articulate legal issues in plain language, eliminating the need for direct consultation with legal professionals.

1. Comprehensive Legal Assistance: Beyond merely suggesting relevant legal sections, the platform offers a wide array of functionalities to cater to diverse legal scenarios. Users can access incident reports, switch between different categories of law, and even generate receipts for their interactions, ensuring a holistic approach to legal assistance.

2. User-Centric Design: The platform's user interface is carefully crafted to accommodate users with varying levels of legal expertise. From seasoned legal professionals to laypersons seeking basic legal information, the system offers tailored features and a user-friendly interface to cater to their specific needs.

3. Empowerment Through Technology: By leveraging advanced algorithms and AI generative models, the platform empowers users to navigate complex legal landscapes with ease. It acts as a virtual legal assistant, providing prompt and accurate legal section recommendations based on user inputs, thereby reducing reliance on costly legal consultations.

4. Enhanced Accessibility: The web-based nature of the platform ensures accessibility from anywhere with an internet connection. This means that individuals in remote areas or those with limited access to legal resources can now avail themselves of comprehensive legal assistance at their convenience.

5. Scalable Infrastructure: The project's robust hardware infrastructure is designed to accommodate a substantial volume of legal data and support the growing needs of users over time. This scalability ensures that the platform remains efficient and responsive even as the user base expands.

6. Data Security and Compliance: Given the sensitive nature of legal information, the platform prioritizes data security and compliance with regulatory standards. Secure networking protocols are employed to ensure the confidentiality and integrity of legal data, instilling trust and confidence among users.

7. Continuous Improvement: The project operates on the principle of continuous improvement, with a dedicated team of system administrators responsible for maintaining and enhancing the platform's functionality. User feedback is actively solicited and incorporated into future updates, ensuring that the platform remains relevant and effective.

In summary, the "Case Code Mapper: Unraveling Legal Section" project represents a paradigm shift in the field of legal research, offering a comprehensive, user-centric, and technologically advanced solution to empower individuals and organizations in navigating the complexities of the legal domain.

2. Methodology

2.1 SRS Documentation

SYSTEM REQUIREMENT SPECIFICATION DOCUMENT

FOR

CASE CODE MAPPER: UNRAVELING LEGAL SECTION

Table of Contents

1. Introduction……..7-9

1.1 Purpose …….7

1.2 Scope ……7

1.3 Definitions, Acronyms, and Abbreviations ……..8-9

1.4 References ……9

1.5 Overview ……....9

2. The Overall Description……10-12

2.1 Product Perspective……..10

2.2 Product Functions…….10-11

2.3 User Characteristics……..11-12

2.4 Constraints…….12

2.5 Assumptions and Dependencies…….12

3. External Interface Requirements…….13-16

3.1 User Interfaces……..13-14

3.2 Hardware Interfaces……..14-15

3.3 Software Interfaces…….15

3.4 Communications Interfaces……16

4. System Features……..16-17

4.1 User Authentication……16

4.2 Legal Problem Input……...16

4.3 Matching Algorithm……17

4.4 Legal Section Suggestions & Database Management…….17

4.5 Reporting & Analytics…….17

5. Other Non-Functional Requirements……18-19

5.1 Performance Requirements…….18

5.2 Safety requirements…….18

5.3 Security requirements……..18-19

5.4 Software quality attributes……...19

1. Introduction

The "Case Code Mapper: Unraveling Legal Section" is a transformative Full Stack Project designed to simplify legal research. With its dynamic web-based platform and sophisticated AI generative models, this tool empowers users to articulate complex legal problems in plain language, offering prompt and accurate suggestions of applicable legal sections. Bridging the gap between intricate legal texts and end-users, the project aims to revolutionize accessibility and comprehension within the legal domain, ushering in a new era of efficiency and precision in legal research and decision-making.

1.1 Purpose

This SRS documentation defines the External Interface, Performance, and Software System Attributes requirements of Case Code Mapper: Unraveling Legal Section. This project is specified for the following ones:

- This innovative solution is a vital resource for law students, enabling them to navigate and understand legal landscapes effortlessly.
- It proves invaluable for law enforcement personnel by providing quick access to relevant IPC section codes and facilitating the interpretation of crime reports, ultimately fostering efficiency and informed decision-making in the legal domain.
- The Police Officers get easy access to IPC to file the FIR.

1.2 Scope

The "Case Code Mapper: Unraveling Legal Section" project has a broad scope, aiming to revolutionize the accessibility and understanding of legal information. The key functionalities include:

- Dynamic Display of IPC Section Codes: The system will dynamically display relevant Indian Penal Code (IPC) section codes based on user-inputted case details, providing quick and accurate references for legal practitioners and law students.
- Incident Report Access: Users, including law enforcement officers, can access detailed reports for stored incidents in the database. This feature enhances the tool's utility for crime investigation and legal analysis.

- User Accessibility: The application is designed to cater to a diverse user base, including students, legal officers, and common citizens. Different user profiles will be created, ensuring tailored access levels and functionalities for each category of users.
- Simplified Constitutional Data Study: Recognizing the complexity of constitutional data, the project aims to simplify the study of legal frameworks. The tool assists law students in comprehending intricate constitutional details by providing easy access to relevant sections and articles.
- User Profiles: The system will implement user profiles with varying levels of access and functionality. Students can benefit from educational resources, legal officers can efficiently retrieve IPC section codes, and common citizens can gain insights into legal aspects relevant to their queries.

By encompassing these functionalities, the "Case Code Mapper" project extends its scope to cater to a wide audience, making legal information more accessible and user-friendly across different user categories. The software prototype is expected to be completed in duration of 2 to 3 months of period.

1.3 Definitions, Acronyms, and Abbreviations

Case Code Mapper: The name of the project, refers to the software tool developed to assist in identifying relevant legal sections or articles based on problem descriptions.

Legal Sections: Specific sections or articles within legal documents or statutes that pertain to particular legal issues or topics.

Matching Algorithms: Algorithms utilized in the backend of the system to identify and suggest relevant legal sections based on input text.

IPC: Indian Penal Code

CRPC: Code of Criminal Procedure

FIR: First Information Report

IPC Sections: Specific sections of the Indian Penal Code, often referred to by their section numbers (e.g., IPC Section 302 for murder)

POCSO: Protection of Children from Sexual Offences Act

Dowry Prohibition Act: Dowry Prohibition Act, which addresses dowry-related offenses

NCRB: National Crime Records Bureau

CCTV: Closed-circuit television, often used for surveillance and crime prevention **DNA:**

Deoxyribonucleic acid, used in forensic investigations for identifying individuals**POTA:**

Prevention of Terrorism Act

NIA: National Investigation Agency

CBI: Central Bureau of Investigation

MO: Modus Operandi, referring to the method of operation used by criminals in committing crimes

MCOCA: Maharashtra Control of Organised Crime Act

IPC Amendment Acts: Acts that amend or update specific sections of the Indian Penal Code, such as the Criminal Law (Amendment) Act, 2013.

These acronyms and abbreviations are commonly used in legal and law enforcement contexts in India and there are many more that cannot able to derive all.

1.3 References

The references used to complete the SRS Documentation are as follows:

1.https://en.wikipedia.org/wiki/Indian_Penal_Code#:~:text=Sections%20120B%20(criminal%20conspiracy)%2C,kidnapping%20for%20ransom)%2C%20396%20

2. http://www.google.co.in

3.https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

1.5 Overview

1.1 Purpose: Identifies target users and key functionalities of the project.

1.2 Scope: Outlines the breadth of features and objectives covered by the project.

1.3 Definitions, Acronyms, and Abbreviations: Clarifies terminology used within the document.

1.4 References: Lists the sources consulted for the SRS documentation.

2. The Overall Description

2.1 Product Perspective

- The "Case Code Mapper: Unraveling Legal Section" is a comprehensive web-based application, functioning as a unified system with interconnected components.
- This software empowers users to articulate legal problems in plain language and suggests relevant legal sections, eliminating the need for direct interaction with legal professionals.
- Beyond accessing legal sections, the system supports additional functionalities like viewing incident reports and displaying IPC section codes, extending its utility to various legal scenarios.
- Hardware components include servers, databases, user interfaces, and communication links, enabling seamless interaction with the backend systems.
- The system utilizes matching algorithms and an AI generative model to process user inputs, providing accurate and dynamic legal section recommendations.
- Communication with the backend server occurs through secure networking protocols, ensuring the confidentiality and integrity of legal data.
- Expected hardware specifications include sufficient memory (e.g., 50MB) to support the matching algorithms and AI generative model, ensuring efficient processing of legal information.
- Storage components, such as databases and disk space, are designed to accommodate a substantial volume of legal data, ensuring scalability as the system grows.

2.2 Product Functions

The key functions of the Case Code Mapper: Unraveling Legal Section are outlined below:

- Legal Problem Articulation: Users can input problem descriptions or case details, enabling the system to process the information using advanced algorithms.
- Legal Section Suggestions: The system retrieves and presents relevant legal sections based on user-provided information, aiding in understanding and applying the law effectively.
- Incident Report Viewing: Users have the functionality to view detailed incident reports related to legal cases stored in the system's database.

- Language Switching: Flexibility is provided for users to switch between different sections or categories of law, ensuring a comprehensive exploration of legal information.
- User Profiles: The system supports different user profiles tailored to students, legal officers, and common citizens, offering customized features for each category.
- Receipt Generation: Every transaction or legal suggestion is recorded and can be printed as a receipt, providing a tangible record of the user's legal interactions.
- Cancellation and Regenerating: Users can cancel specific legal section suggestions or the entire session, ensuring flexibility and user control.
- Interactive Legal Maps: The system displays maps marking the locations of other legal databases or resources, facilitating comprehensive legal research. The main and whole motive functions are:
    1. Let users type in their legal problems or case details
    2. Uses smart algorithms to understand what the users are saying
    3. Find and show the right legal rules that match the user's problem
    4. Gives an easy way for users to interact with the system

2.3 User Characteristics

The Case Code Mapper is designed to be user-friendly for different types of users, each with their unique needs and expertise. Here's a breakdown of the user characteristics:

- **Legal Professionals:**
    - **Description:** Experienced legal practitioners such as lawyers and legal advisors.
    - **Characteristics:** Have in-depth legal knowledge, familiar with complex legal terms, and use the tool for quick referencing and research. Expected to be proficient in using computers.
- **Law Students:**
    - **Description:** Individuals studying law and seeking assistance in understanding legal sections and codes.
    - **Characteristics:** Developing legal knowledge, may not be fully familiar with intricate legal language. Needs a Basic understanding of computer usage.

- **General Public/Citizens:**
    - **Description:** Everyday individual seeking legal information for personal understanding.
    - **Characteristics:** Limited legal knowledge, may not be familiar with complex legal terminology. Needs basic computer skills and a user-friendly interface is crucial.
- **System Administrators:**
    - **Description:** Individuals responsible for maintaining and managing the Case Code Mapper system.
    - **Characteristics:** Technical expertise to ensure smooth operation and troubleshooting. Having Advanced computer skills, and knowledge of system administration.

2.4 Constraints

There are a few things that might limit how we make the Case Code Mapper:

- We need to finish making it and testing it in a certain amount of time
- We have a limit on the people working on it
- We need to follow the rules about using and sharing legal information

2.5 Assumptions and Dependencies

When we make the Case Code Mapper, we assume a few things will happen:

- People using the tool will tell us the right information about their legal problems
- The legal documents we use to make the tool are correct and up-to-date
- We'll host the tool on a reliable web server that keeps it running smoothly
- We assume that users will actively engage with the tool and provide accurate and relevant information about their legal problems.
- An assumption is made that there will be a continuous feedback loop with users. It is assumed that users, especially law students and legal professionals, have a basic understanding of legal terminology.

3. External Interface Requirements

3.1 User Interface Requirements

For the "Case Code Mapper: Unraveling Legal Section" project, the user interface will be web-based and includes as follows:

- **Registration/Login Page:**
    - Input fields for creating an account or logging in with different user profiles.
    - Interactive help options are available for each service listed.
- **Home Page:**
    - Upon logging in, users will land on the home page featuring sections such as "Recent Crimes," "Posts," and "Contact."
    - Text input area for entering case details or uploading image URLs.
- **Case Details and IPC Display:**
    - Upon entering case details, a navigational interface to display relevant IPC sections.
    - Can get similar incident visuals or articles if present in the database
    - Comments section for users to discuss and suggest improvements on displayed IPCs.
    - Users can view and comment on suggested IPC sections, providing a collaborative platform for legal discussions.
- **Feedback Section:**
    - A dedicated page for users to provide feedback on the tool's functionality and suggestions for improvement.
    - The system may generate reports on feedback for continuous enhancement.

The project will generate detailed reports after each user session to ensure transparency and enhance system functionality. The reports include:

User Interaction Logs: Capture login and logout times, along with user identifiers, for auditing purposes.

User Profile and Access Logs: Document user profiles, access permissions, and any profile modifications for administrative tracking.

Case Entry and Modification Logs: Track the creation and modification of legal case entries, ensuring accountability and transparency.

IPC Section Suggestions Logs: Record suggestions provided by the system for relevant IPC sections, including user feedback.

Feedback and Improvement Logs: Store user feedback and suggestions for system improvements, supporting continuous refinement.

3.2 Hardware Interface Requirements

The Case Code Mapper is designed to be accessible on standard computer and mobile devices. The hardware interface requirements are as follows:

- **Device Compatibility:** The system should be accessible on common computing devices, including desktop computers, laptops, tablets, and mobile phones. Usersshould be able to access the Case Code Mapper from a variety of devices without encountering compatibility issues.
- **Internet Access:** Users must have access to the internet to use the Case Code Mapper. This is essential for connecting to the web-based application, retrieving legal information, and ensuring real-time functionality.
- **Web Browser:** The Case Code Mapper is accessed through a web browser. The system should be compatible with widely used browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Ensuring compatibility with various browsers enhances the user experience and accessibility.
- **Responsive Design:** The system should adapt to different screen sizes and resolutions. Whether users access the Case Code Mapper on a large desktop monitor or a small mobile screen, the interface should be responsive, providing a seamless experience across devices.

- **Input Devices:** Standard input devices, such as keyboards and touchscreens, should be supported. Users should be able to interact with the system using conventional input methods for typing case details, navigating the interface, and submitting queries.
- **Minimal Hardware Requirements:** The hardware requirements for using the Case Code Mapper should be minimal to ensure accessibility for a broad user base. The application should run smoothly on devices with average computing capabilities.

These hardware interface requirements aim to make the Case Code Mapper user-friendly and accessible across a range of devices, facilitating ease of use for individuals with varying technological setups and preferences.

3.3 Software Interfaces

The Case Code Mapper will interact with various software components to ensure smooth operation and efficient processing. The key software interface requirements include:

- **User Interface (UI):** The system will have a user-friendly web-based interface accessible via standard web browsers. The UI will allow users to input case details or problem descriptions and interact with the system effortlessly.
- **Backend Processing:** The backend of the Case Code Mapper will be implemented using Python with the Django framework. This includes the processing of user inputs, application logic, and the integration of AI generative models for legal section suggestions.
- **Database Management:** Legal data will be stored and retrieved using a SQL database. This ensures efficient data management, allowing the system to store and retrieve relevant legal sections promptly.
- **AI Generative Model Integration:** The system will integrate AI generative models to enhance the accuracy of legal section suggestions. The integration will involve communication between the backend processing and the AI component to dynamically generate relevant legal articles.
- **Web Server:** The Case Code Mapper will be hosted on a reliable web server to ensure continuous availability. This includes considerations for server specifications, maintenance, and scalability to accommodate growing user traffic.

3.4 Communication Interface Requirements

The Case Code Mapper will communicate with external entities via standard web protocols, including:

- HTTP for transferring data between the client and server
- JSON for exchanging data in a lightweight format
- The communication interface will follow REST (Representational State Transfer) principles, providing a structured and scalable architecture for interactions.

4. System Features

4.1 User Authentication

- Description: This feature ensures secure access to the Project by implementing user authentication. Users, based on their roles, will have personalized access to the system's functionalities.
- Functionality:
    - User registration and login
    - Role-based access control (e.g., students, legal officers)
    - Password protection and security measures

4.2 Legal Problem Input

- Description: Allows users to input legal problems or case details in plain language or through image URLs, serving as the foundation for generating legal section suggestions.
- Functionality:
    - User-friendly interface for problem description input
    - Support for various input formats (text, keywords, jpg, jpeg…)
    - Validation checks to ensure accurate input

4.3 Matching Algorithm

- Description: The core functionality that processes user inputs using advanced matching algorithms to identify and understand the legal context of the problem description.
- Functionality:
    - Natural language processing for understanding user inputs
    - Matching algorithms to correlate input with legal documents
    - Regular updates to improve matching accuracy

4.4 Legal Section Suggestions & Database Management

- Description: Generates and presents relevant legal sections based on the processed user inputs. Manages the legal database efficiently.
- Functionality:
    - Dynamic suggestion of applicable legal sections
    - Database management for storing and retrieving legal information
    - Integration with AI generative models for enhanced suggestions

4.5 Reporting & Analytics

- Description: Provides reporting features and analytical tools for users to gain insights into legal data and track system usage.
- Functionality:
    - View incident reports linked to legal sections
    - Analytics dashboard for user activity and system performance
    - Reporting tools for user feedback and system improvements

These system features collectively contribute to the Case Code Mapper's functionality, offering a comprehensive solution for users to navigate legal information efficiently and make informed decisions.

5. Other Non-Functional Requirements

5.1 Performance Requirements

The performance requirements of the project are as follows below in brief:

- **Capacity:** The Application provides the capacity of the multiple users without any performance issues
- **Dynamic Requirements:** The application should dynamically adapt to changing user demands and system loads.
- **Quality:** The system must maintain high-quality standards in terms of accuracy, relevance, and responsiveness.
- **Response Time:** The system should respond to user queries and inputs promptly, aiming for minimal response times.
- **Scalability:** The architecture must be scalable to accommodate an increasing number of users and data volumes without compromising performance.

5.2 Safety requirements

The Case Code Mapper primarily deals with legal information and does not have direct safety implications. However, the safety of user data and privacy is a paramount concern. The system must adhere to data protection laws and regulations to ensure the safety and confidentiality of user information. An important aspect is validation of the users is required in a more authorized way to not misuse the application.

5.3 Security requirements

The security requirements are necessary for the project which they include sensitive, confidential data. So, we need the following security requirements as follows:

Data Encryption: All sensitive data, including user credentials and legal information, must be encrypted during transmission and storage.

Access Control: Implement role-based access control to ensure that users only have access to information relevant to their roles.

Authentication and Authorization: Strong user authentication and authorization mechanisms must be in place to prevent unauthorized access.

Regular Security Audits: Conduct regular security audits to identify and address potential vulnerabilities.

5.4 Software quality attributes

Reliability: The system should consistently provide accurate and relevant legal section suggestions.

Maintainability: The software should be designed in a modular and maintainable manner to facilitate future updates and improvements.

Usability: The user interface must be intuitive and user-friendly, promoting ease of interaction for users of varying technical backgrounds.

Accuracy: The matching algorithms must be precise and continuously improved to enhance the accuracy of legal section suggestions.

2.2 Functional and Non-Functional Requirements

Functional Requirements

1. User Registration:

- Users should be able to register for an account by providing the necessary information.
- The registration process should include fields for username, email, and password.

2. User Authentication:

- Registered users should be able to log in securely using their credentials.
- Passwords should be encrypted before storing them in the database.

3. Profile Management:

- Users should have the ability to update their profile information, including name, email, and password.
- They should also be able to upload a profile picture.

4. Content Creation:

- Users should be able to create new content, such as articles or posts.
- The content creation process should include fields for title, body, and tags.

5. Content Editing:

- Users should be able to edit their existing content, including updating the title, body, or tags.

6. Content Deletion:

- Users should have the option to delete their own content if needed.

7. Content Viewing:

- Users should be able to view content created by themselves and others.
- They should also have the ability to filter and search for specific content.

8. User Roles and Permissions:

- The system should support different user roles, such as regular users and administrators.
- Administrators should have additional privileges, such as the ability to delete or moderate content.

9. User Interaction:

- Users should be able to like, comment on, and share content.
- Content creators should receive notifications when their content is interacted with.

10. User Feedback:

- Users should be able to provide feedback on the platform, such as reporting bugs or suggesting improvements.
- There should be a mechanism for administrators to review and address user feedback.

Non-Functional Requirements:

1. Usability:

- The application should have an intuitive user interface to ensure ease of navigation and usage.
- Forms and input fields should be well-labeled and organized for user convenience.

2. Security:

- User data should be encrypted during transmission and storage to prevent unauthorized access.
- Passwords should be securely hashed before storage to protect user privacy.

3. Performance:

- The application should respond promptly to user interactions without significant delays.
- Database queries and data retrieval processes should be optimized for efficiency.

4. Reliability:

- The application should be stable and reliable, with minimal downtime or system failures.
- Error handling mechanisms should be in place to gracefully handle unexpected errors or exceptions.

5. Scalability:

- The application architecture should be designed to accommodate future scalability requirements.
- It should be able to handle an increasing number of users and data without compromising performance.

6. Accessibility:

- The application should be accessible to users with disabilities, following accessibility standards such as WCAG (Web Content Accessibility Guidelines).

- Alternative text should be provided for images, and keyboard navigation should be supported.

7. Data Integrity:

- Measures should be implemented to ensure the integrity and consistency of user data stored in the database.
- Data validation rules should be enforced to prevent the entry of incorrect or invalid information.

8. Auditability:

- The application should maintain logs of user activities and system events for auditing purposes.
- Admins should have access to view and analyze audit logs for security and compliance purposes.

9. Interoperability:

- The application should be compatible with various devices, browsers, and operating systems to ensure broad accessibility.
- It should support standard protocols and APIs for integration with third-party services or systems.

10. Compliance:

- The application should comply with relevant legal and regulatory requirements, such as data protection laws (e.g., GDPR, CCPA).
- It should also adhere to industry standards and best practices in software development and security.

2.3 Technology Used

Front End: HTML, CSS, JavaScript

Backend: Python,flask, PHP,Xampp

3. Designing

3.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system.It shows how data enters and leaves the system, what changes the information, and where data is stored. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

3.1.1 Context Level or Level 0 DFD:

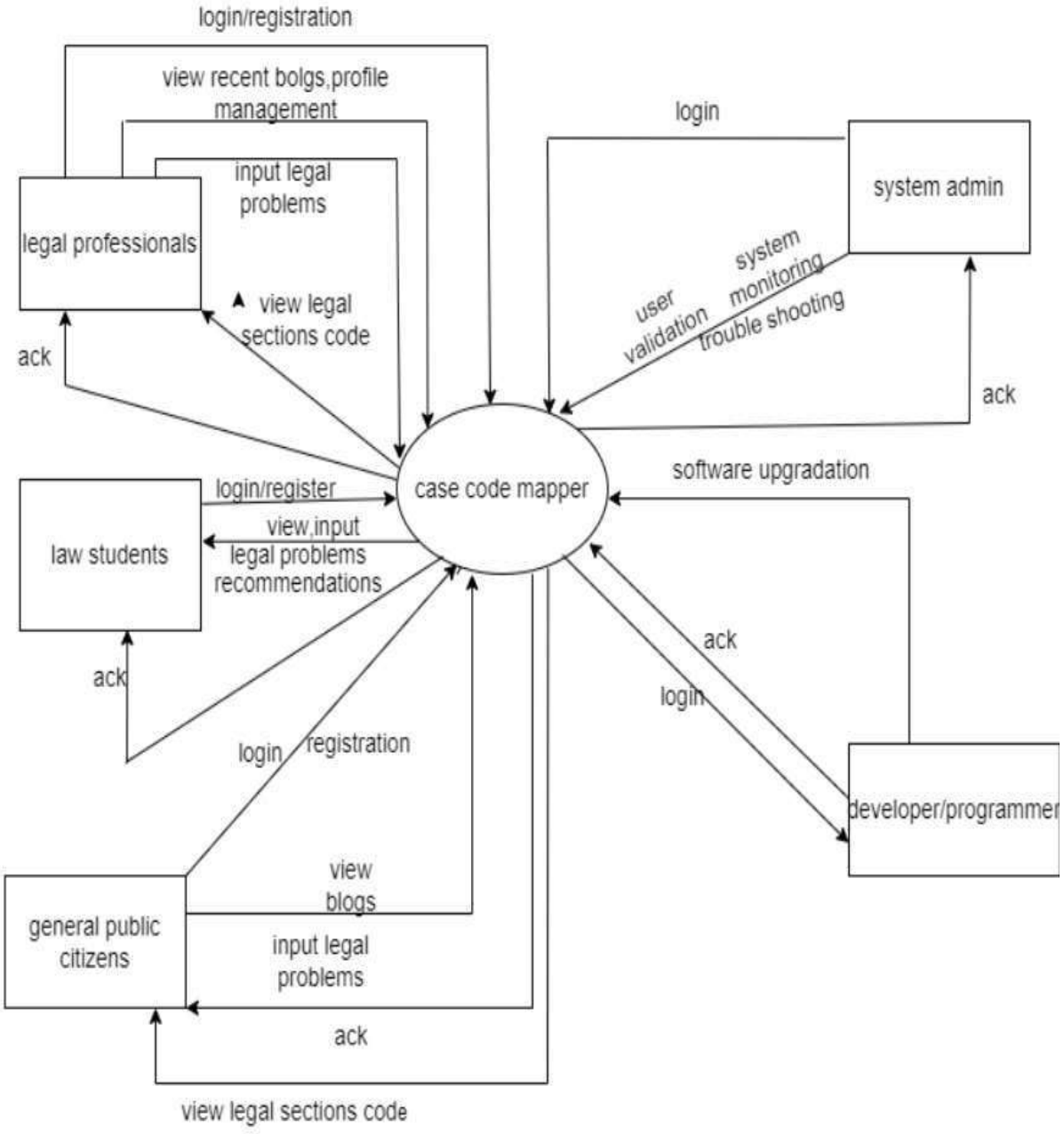


Figure 1: Level 0 Data Flow Diagram

Level 0 Data Flow Diagram (DFD)

- The Level 0 Data Flow Diagram (DFD) illustrates the fundamental processes, data flows, data stores, and external entities within the system. The system encompasses various user roles, including legal professionals, law students, general public citizens, system administrators, and developers/programmers. It facilitates functionalities such as viewing recent blogs, managing user profiles, inputting legal problems or cases, monitoring system health, troubleshooting, mapping legal cases to respective codes, and upgrading software. The central data store in the system is the Legal Sections Code Database, housing codes for legal sections for reference. Data flows within the system include retrieving and displaying recent blogs, users submitting legal problems, accessing legal sections codes, providing recommendations for legal problems, and mapping legal cases to codes.

- At the core of the system are processes aimed at facilitating user interaction with legal content and system management. Users can access recent blogs, submit legal problems, and receive recommendations based on their input. The system also supports administrative functions such as profile management and software upgrades. External entities, including legal professionals, law students, and general citizens, interact with the system to fulfill their respective needs. System administrators and developers/programmers play vital roles in maintaining system integrity, ensuring smooth operation, and implementing necessary upgrades. The Level 0 DFD offers a holistic view of the system's architecture, highlighting key processes, data flows, and external entities involved in its operation.

3.1.2 Level 1 DFD:

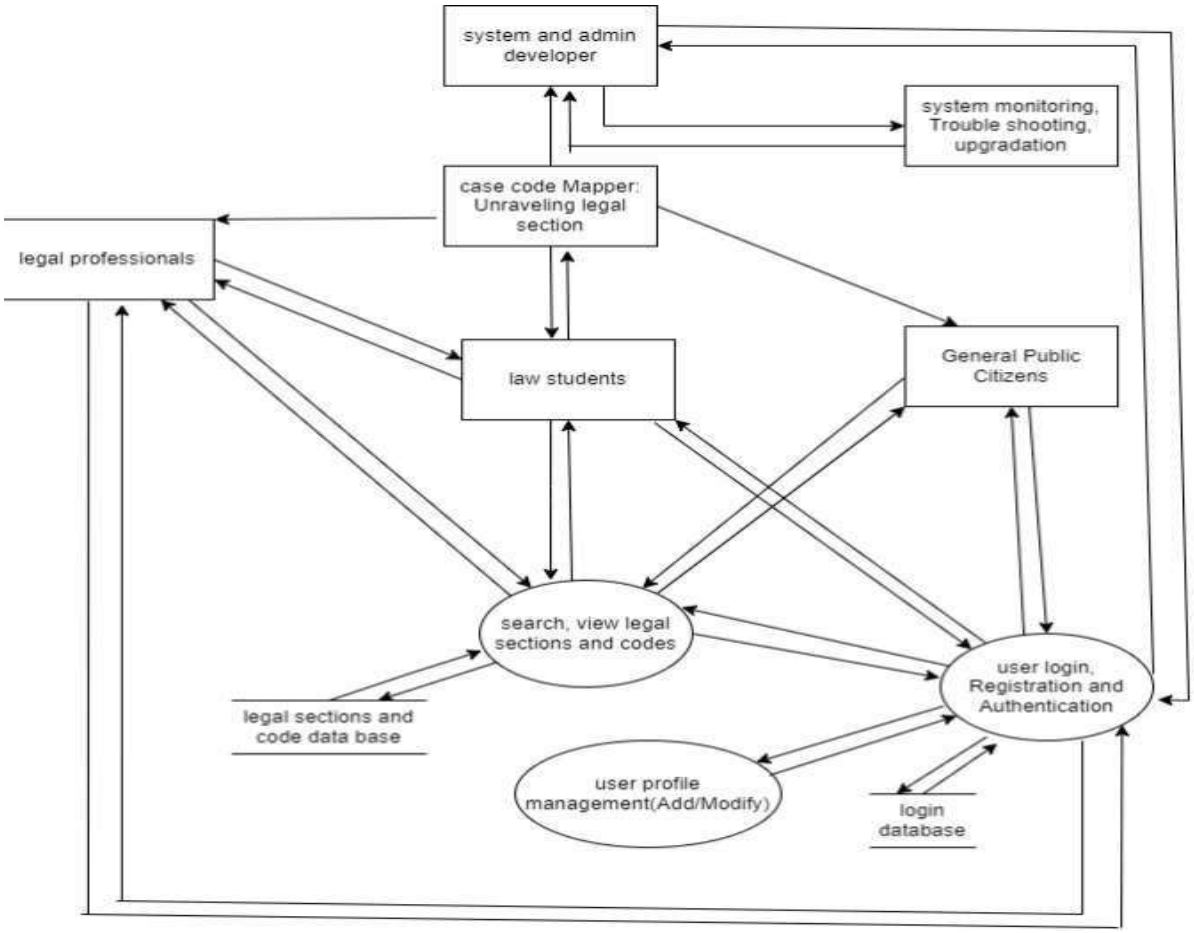


Figure 2: Level 1 Data Flow Diagram

Level 1 Data Flow Diagram (DFD)

Processes:

- System Monitoring and Troubleshooting
- Legal Section Upgradation
- User Profile Management
- User Authentication
- Legal Section Search

Data Stores:

- Legal Sections and Codes Database
- Login Database

External Entities:

- System and Admin Developer
- Legal Professionals
- Law Students
- General Public Citizens

Data Flows:

- From System and Admin Developer to Legal Sections and Codes Database: System Upgrades
- From Legal Professionals to Legal Sections and Codes Database: Legal Section Edits
- From Law Students to Legal Sections and Codes Database: Legal Section Views
- From General Public Citizens to Legal Sections and Codes Database: Legal Section Searches
- From Login Database to User Authentication: User Login Details
- From User Authentication to Legal Sections and Codes Database: User Authentication Check
- From Legal Sections and Codes Database to Legal Professionals: Legal Section Retrieval
- From Legal Sections and Codes Database to Law Students: Legal Section Retrieval
- From Legal Sections and Codes Database to General Public Citizens: Legal Section Retrieval.

3.1.3 Level 2 DFD:

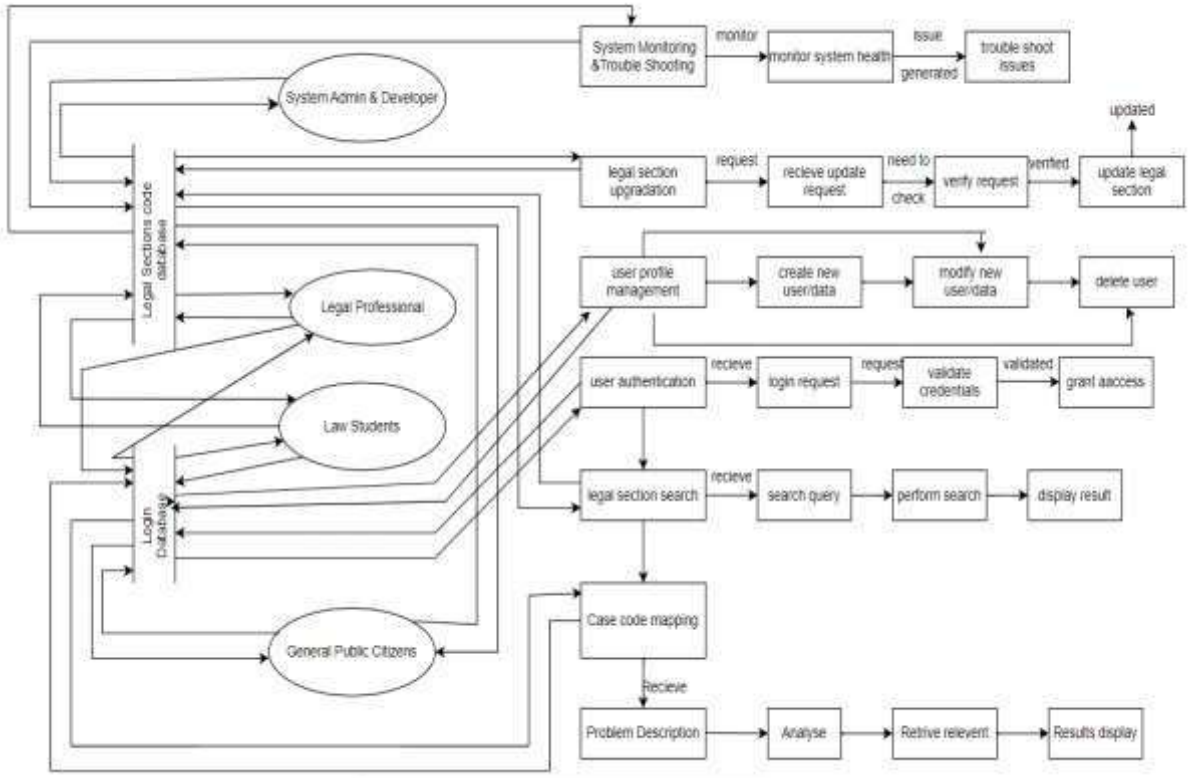


Figure 3: Level 2 Data Flow Diagram

Let us see in detail all the processes, data stores entities and data flows.

Processes:

1. System Monitoring and Troubleshooting:

Subprocess 1.1: Monitor System Health

Subprocess 1.2: Troubleshoot Issues

2. Legal Section Upgradation:

Subprocess 2.1: Receive Update Request

Subprocess 2.2: Verify Request

Subprocess 2.3: Update Legal Section

3. User Profile Management:

Subprocess 3.1: Create a New User Profile

Subprocess 3.2: Modify User Profile

Subprocess 3.3: Delete User Profile

4. User Authentication:

Subprocess 4.1: Receive Login Request

Subprocess 4.2: Validate User Credentials

Subprocess 4.3: Grant Access

5. Legal Section Search:

Subprocess 5.1: Receive Search Query

Subprocess 5.2: Perform Search

Subprocess 5.3: Display Results

6. Case Code Mapping:

Subprocess 6.1: Receive Problem Description

Subprocess 6.2: Analyze Problem

Subprocess 6.3: Retrieve Relevant Legal Section Codes

Subprocess 6.4: Display Results

Data Stores:

1. Legal Sections and Codes Database:

Substore 1.1: Legal Section Details

Substore 1.2: Updated Legal Sections

Substore 1.3: Case Code Mapping Data

2. Login Database:

Substore 2.1: User Credentials

Substore 2.2: User Profiles

External Entities:

System and Admin Developer

Legal Professionals

Law Students

General Public Citizens

Data Flows:

1. From System and Admin Developer to Legal Sections and Codes Database:

Subflow 1.1: System Upgrades

Subflow 1.2: Updated Legal Sections

2. From Legal Professionals to Legal Sections and Codes Database:

Subflow 2.1: Legal Section Edits

Subflow 2.2: Updated Legal Sections

3. From Law Students to Legal Sections and Codes Database:

Subflow 3.1: Legal Section Views

Subflow 3.2: Case Code Mapping Data

4. From General Public Citizens to Legal Sections and Codes Database:

Subflow 4.1: Legal Section Searches

Subflow 4.2: Case Code Mapping Data

5. From Login Database to User Authentication:

Subflow 5.1: User Login Details

6. From User Authentication to Legal Sections and Codes Database:

Subflow 6.1: User Authentication Check

Subflow 6.2: Case Code Mapping Data

7. From Legal Sections and Codes Databases to Legal Professionals, Law Students, and General Public Citizens:

Subflow7.1: LegalSectionRetrieval

Subflow 7.2: Case Code Mapping Data

3.2 Data Dictionary for the DFD:

Data Item	Description	Type	Format	Example
System Upgrades	Updates and enhancements to the system.	Text	N/A	System Upgrade v2.1.3
Legal Section Edits	Changes made to legal sections by professionals.	Text	N/A	Added new subsection
Legal Section Views	Accessing legal sections by law students.	Text	N/A	Viewed section on contracts
Legal Section Searches	Searching for legal sections by the public.	Text	N/A	Search for labor laws
User Login Details	Information is provided during user login.	Structured	Username, Password	Username: user1, Password: *****
User Authentication Check	Verification of user credentials.	Boolean	True/False	True

Legal Section Retrieval	Accessing legal sections by professionals, law students, and the public.	Text	N/A	Retrieved section on taxes
Case Code Mapping Data	Data related to the mapping of problem descriptions to legal section codes.	Structured	Problem Description, Legal Code	"Injury due to negligence" -> "Section 12A"
User Profiles	Information about users' profiles.	Structured	Username, Email, Role	"user1", "user1@example.com", "Admin"
Legal Sections and Codes Database	Database containing legal section details and codes.	Database	SQL Database	Table: LegalSections, Columns: SectionID, SectionTitle, SectionCode
Login Database	Database storing user login information.	Database	SQL Database	Table: Users, Columns: UserID, Username, Password, Role
System and Admin Developer	Developers and administrators of the system.	Text	N/A	A Person
Legal Professionals	Professionals who interact with legal sections.	Text	N/A	Officers
Law Students	Students studying law.	Text	N/A	Law Student
General Public Citizens	Members of the public accessing legal information.	Text	N/A	All citizens

Table 1 Data Dictionary for DFD

3.3 Use Case Diagram

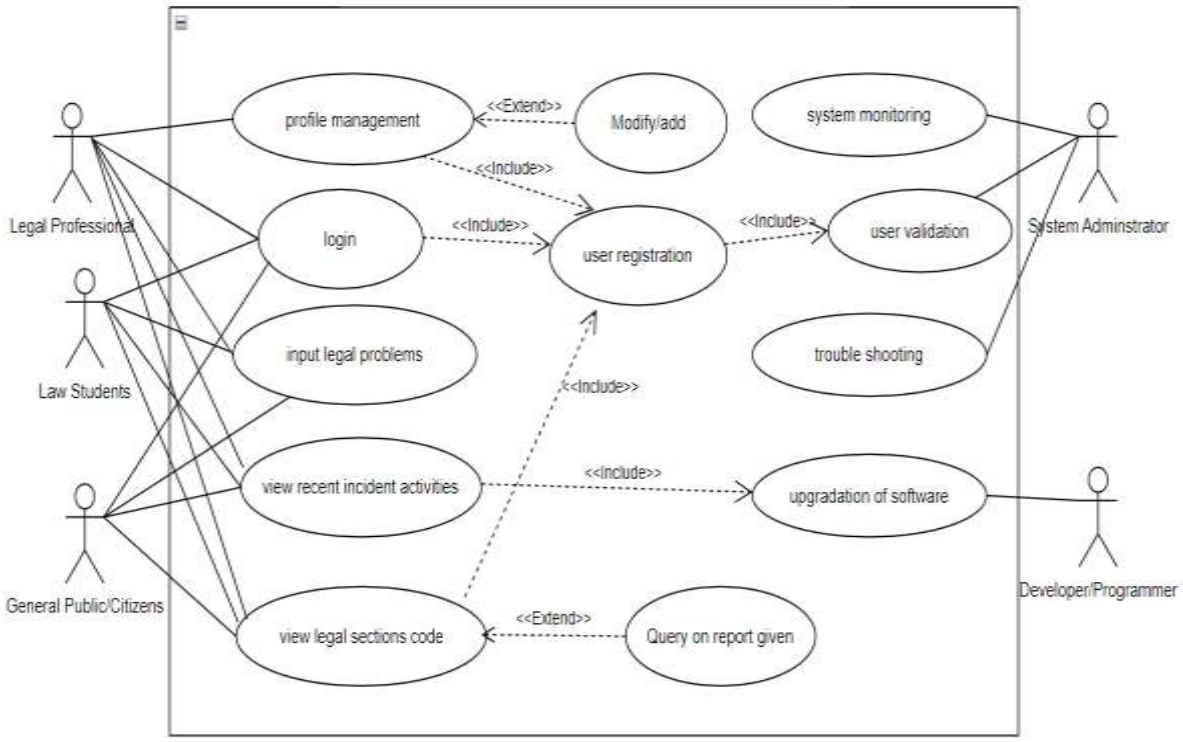


Figure 4: Use case Diagram

Description

Profile Management (Legal Professional):

Description: This use case involves the functionality for Legal Professionals to manage their user profiles within the system. It includes options to modify existing profile information and add new details as needed.

Actors: Legal Professional

Preconditions: The Legal Professional must be logged into the system.

Postconditions: Changes made to the profile are successfully saved in the system.

System Monitoring (System Administrator):

Description: This use case pertains to the capability of the System Administrator to monitor the system for any irregularities, errors, or performance issues. It includes features for real-time monitoring and generating alerts if any issues are detected.

Actors: System Administrator

Preconditions: The System Administrator must have access to the system monitoring tools.

Postconditions: Any anomalies detected in the system are appropriately addressed or escalated.

User Registration (General Public/Citizens, Law Students, Legal Professionals):

Description: This use case enables members of the general public or citizens, Law Students, and Legal Professionals to register for access to the system. It involves providing necessary personal information and creating login credentials.

Actors: General Public/Citizens, Law Students, Legal Professionals.

Preconditions: The user must have access to the registration interface.

Postconditions: Upon successful registration, the user gains access to the system with their provided credentials.

Login (General Public/Citizens, Law Students, Legal Professionals):

Description: This use case allows registered users (General Public/Citizens) to log into the system using their previously created credentials.

Actors: General Public/Citizens, Law Students, Legal Professionals.

Preconditions: The user must have a registered account.

Postconditions: Upon successful login, the user gains access to the system's functionalities and resources.

Troubleshooting (System Administrator):

Description: This use case involves providing support and guidance to users (General Public/Citizens) who encounter issues or difficulties while using the system. It includes steps for identifying, diagnosing, and resolving common problems.

Actors: General Public/Citizens

Preconditions: The user must be experiencing a technical issue with the system.

Postconditions: The user's issue is addressed satisfactorily, and they can continue using the system without further hindrance.

User Validation (System Administrator):

Description: This use case involves the process of validating users within the system, particularly aimed at ensuring the authenticity and credentials of System Admin. It includes verification procedures and checks to maintain the integrity of user accounts.

Actors: System Administrator

Preconditions: The user validation process is initiated by the system administrator or designated authority.

Postconditions: Verified users gain access to their respective privileges and functionalities within the system.

Input Legal Problems (General Public/Citizens, Law Students, Legal Professionals):

Description: This use case enables the General Public/Citizens, Law Students, Legal Professionals to input legal problems or cases into the system for further processing and management. It involves capturing relevant details, categorizing the issues, and assigning them to appropriate entities for resolution.

Actors: General Public/Citizens, Law Students, Legal Professionals.

Preconditions: The System Administrator must have appropriate permissions to input legal problems.

Postconditions: Legal problems are successfully recorded in the system and made available for further action.

View Recent Incident Activities (Law Students, General Public/Citizens, Law Students, Legal Professionals):

Description: This use case allows Law Students, General Public/Citizens, Law Students, and Legal Professionals to view recent incident activities recorded within the system. It provides insights into ongoing legal matters, recent cases, and related updates.

Actors: Law Students, General Public/Citizens, Law Students, Legal Professionals.

Preconditions: Law Students must have access rights to view incident activities.

Postconditions: Law Students gain knowledge and awareness of recent legal incidents and activities.

Upgradation of Software:

Description: This use case involves updating or upgrading the software system to newer versions or patches. It includes procedures for downloading, installing, and configuring the updates to ensure the system remains up-to-date and secure. Users Perform this action when upgradation is done by the admin.

Actors: System Administrator

Preconditions: The user must have appropriate permissions to perform software upgrades.

Postconditions: The software system is successfully upgraded to the latest version, with improved features and security enhancements.

View Legal Sections Code (General Public/Citizens, Law Students, Legal Professionals):

Description: This use case allows members of the General Public/Citizens, Law Students, and Legal Professionals to view legal sections and codes within the system. It provides access to relevant legal information, statutes, and regulations.

Actors: General Public/Citizens**,** Law Students, Legal Professionals

Preconditions: The user must have access rights to view legal sections and codes.

Postconditions: Users gain access to the desired legal information for reference and understanding.

Developer/Programmer:

Description: This use case provides a detailed description template for developers or programmers to create new use cases within the system. It includes guidelines, format, and required information to ensure consistency and completeness in documenting use cases.

Actors: Developer/Programmer

Preconditions: The developer or programmer must be tasked with creating new use cases.

Postconditions: A well-defined use case description is generated, facilitating the development process and system understanding.

3.4 Entity Relationship Diagram

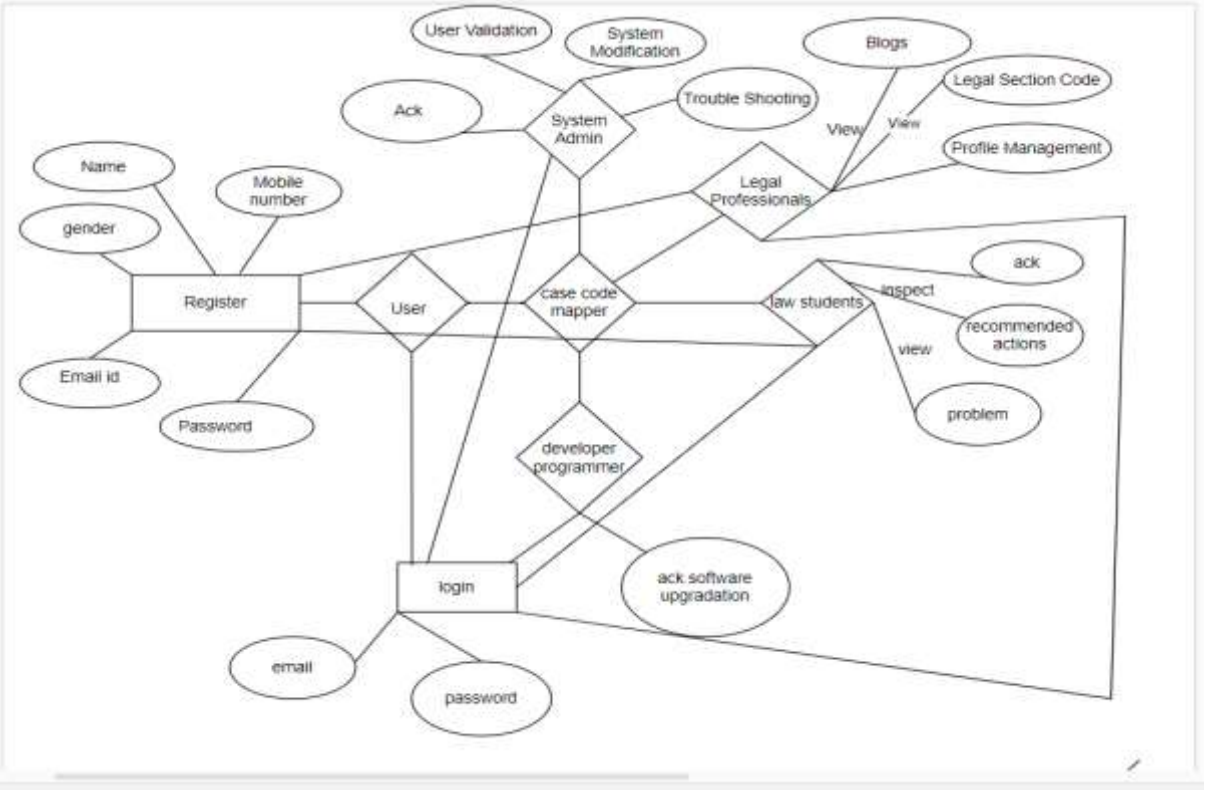


Figure 5: Entity Relationship Diagram

Entities:

User: Represents users of the system, including system administrators, legal professionals, law students, and general public citizens.

Attributes: UserID (Primary Key), Username, Password, Email, Role

Legal Section: Represents sections of legal documents stored in the system.

Attributes: SectionID (Primary Key), SectionTitle, SectionContent

Relationships:

User Authentication: Connects the User entity to the Login Database. It's a one-to-one relationship based on the UserID.

Legal Section Access: Connects the User entity to the Legal Section entity. It's a many-to-many relationship, indicating that users can access multiple legal sections, and legal sections can be accessed by multiple users.

Legal Section Upgradation: Represents the process where Legal Professionals edit and update legal sections. It's a one-to-many relationship between the Legal Professional and Legal Section entities, indicating that a Legal Professional can upgrade multiple legal sections.

Case Code Mapping: Represents the mapping between problem descriptions and legal section codes. It's a one-to-many relationship between the Problem Description and Legal Section Code entities, indicating that a problem description can map to multiple legal section codes.

Attributes:

User Attributes: UserID, Username, Password, Email, Role

Legal Section Attributes: SectionID, SectionTitle, SectionContent

Problem Description Attributes: DescriptionID, Description

Legal Section Code Attributes: CodeID, Code

3.5 Class Diagram

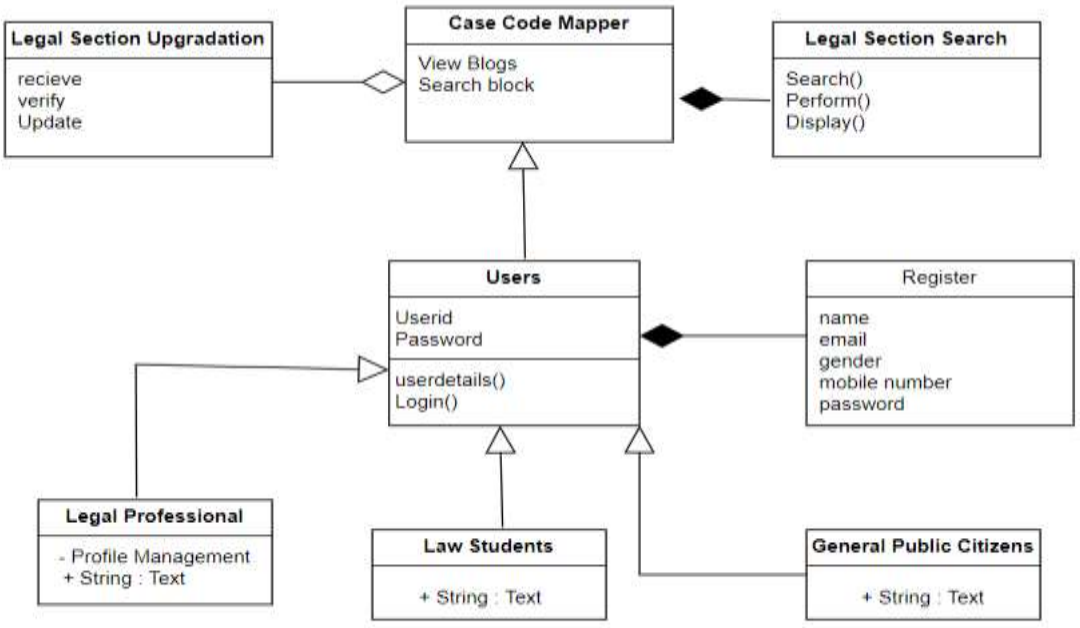


Figure 6: Class Diagram

Description:

Classes:

1. **Users:**
    - Attributes: UserID, Password
    - Methods: UserDetails(), Login()
2. **LegalProfessional:**
    - Attributes: UserID, Password, ProfileManagement (String)
    - Methods: UserDetails(), Login()
3. **LegalSection:**
    - Attributes: -
    - Methods: Upgradation(), VerifyUpdate()
4. **CaseCodeMapper:**
    - Attributes: -
    - Methods: ReceiveVerifyUpdate()

5. **ViewBlogsSearchBlock:**
    - Attributes: -
    - Methods: Search(), Perform(), Display()
6. **Register:**
    - Attributes: Name, Email, Gender, MobileNumber, Password
    - Methods: Description()

Relationships:

1. **Users-LegalProfessional:**
    - Association: Users can have roles as Legal Professionals.
    - Multiplicity: One-to-One
2. **Users-Register:**
    - Association: Users can register with the system.
    - Multiplicity: One-to-One
3. **LegalProfessional-LegalSection:**
    - Association: Legal Professionals can perform operations on legal sections.
    - Multiplicity: One-to-Many
4. **LegalSection-CaseCodeMapper:**
    - Association: Legal sections are used in the Case Code Mapper.
    - Multiplicity: One-to-Many
5. **CaseCodeMapper-ViewBlogsSearchBlock:**
    - Association: The Case Code Mapper interacts with the View Blogs Search Block.
    - Multiplicity: One-to-One

3.6 Activity Diagram

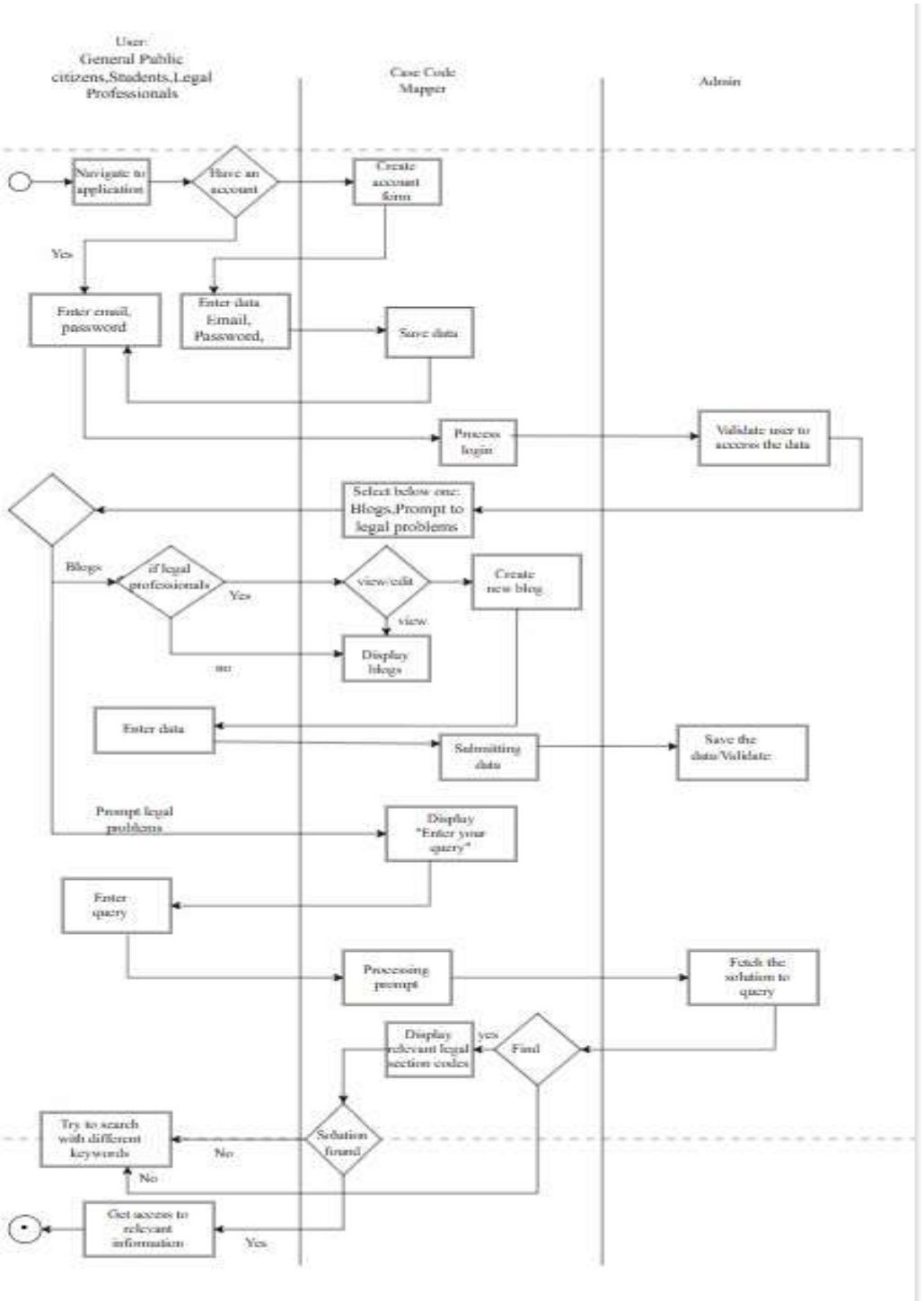


Figure 6.1: Activity Diagram

Description:

1. Start Application: The User/General Public starts the application.
2. Navigate to Registration Form: User/General Public navigates to the registration form.
3. Access Registration Form: Case Code Mapper provides access to the registration form.

4. Have Account: Case Code Mapper checks if the user already has an account.

5. Create Account Form: If the user does not have an account, Case Code Mapper presents the create account form.

6. Enter Email and Password: User/General Public enters their email and password.

7. Enter Data: User/General Public enters necessary profile information.

8. Save Data: Case Code Mapper saves the entered data.

9. Process Login: Admin processes the login information.

10. Validate User Access: After login, the system validates whether the user is a Legal Professional.

11. Select Below One (Blogs/Prompts): Based on user validation, if the user is a Legal Professional, they are given options to view, edit, and create content. Otherwise, blogs are displayed.

12. Enter Data: User/General Public enters the required information, either for creating/editing content or submitting queries.

13. Submitting Data: User/General Public submits the entered data.

14. Save Data: Case Code Mapper saves the submitted data.

15. Display Query: The system displays the submitted query or the created/edited content, depending on the user's action.

16. Stop: The activity within the application concludes.

3.7 State Diagram

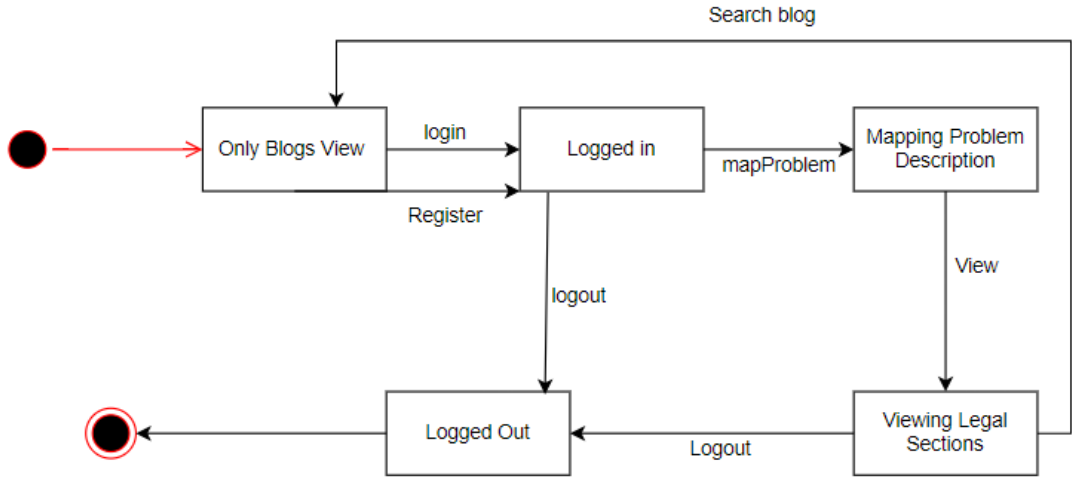


Figure 7: State Chart Diagram

Description:

A state chart diagram, also known as a state machine diagram, depicts the various states that an object or system can transition through in response to events.

States:

Logged Out

Logged In

Mapping Problem Description

Viewing Legal Section, Searching Blogs

Transitions:

User logs in (from Logged Out to Logged In)

User logs out (from Logged In to Logged Out)

User starts mapping a problem description (from Logged In to Mapping Problem Description)

User completes mapping (from Mapping Problem Description to Logged In)

User views a legal section (from Logged In to Viewing Legal Section)

User searches for blogs (from Logged In to Searching Blogs)

3.8 Collaboration Diagram

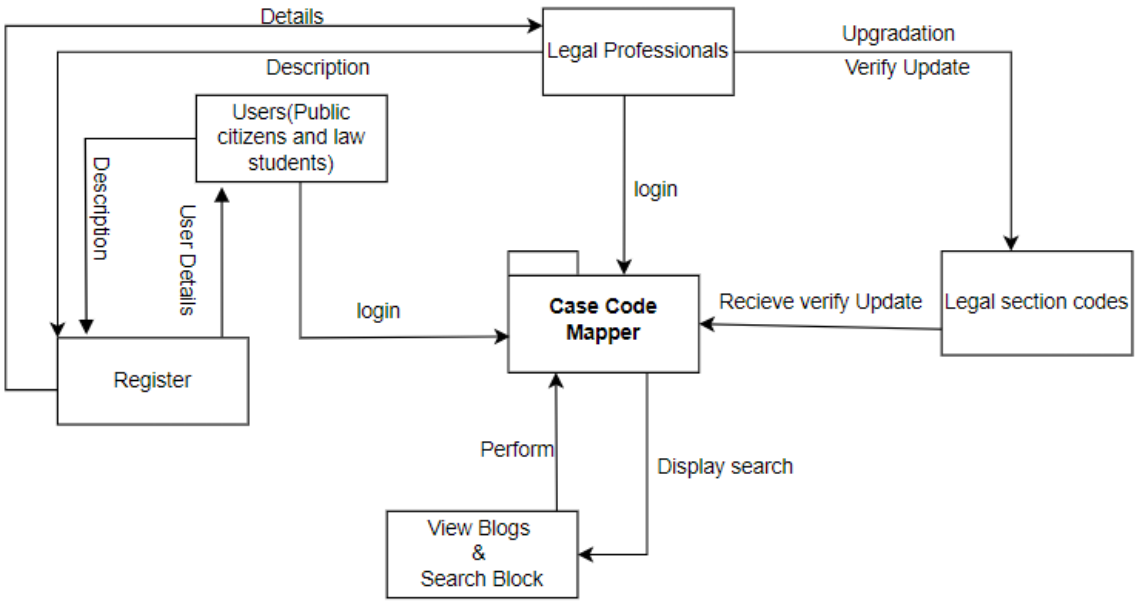


Figure 8: Collaboration Diagram

Description:

A collaboration diagram, also known as a communication diagram, illustrates the interactions between objects in a system to achieve a particular functionality.

Objects:

1. User
2. LegalProfessional
3. LegalSection
4. CaseCodeMapper
5. ViewBlogsSearchBlock
6. Register

Interactions:

1. User requests to log in.
2. LegalProfessional requests to log in.
3. User registers in the system.
4. LegalProfessional manages legal sections (upgrades, verifies updates).
5. CaseCodeMapper receives and verifies updates.
6. CaseCodeMapper interacts with ViewBlogsSearchBlock for searching blogs.
7. ViewBlogsSearchBlock performs the search and displays the results.

Message Sequence:

1. User -> CaseCodeMapper: Login()
2. LegalProfessional -> CaseCodeMapper: Login()
3. User -> Register: Description()
4. LegalProfessional -> LegalSection: Upgradation()
5. LegalProfessional -> LegalSection: VerifyUpdate()
6. LegalSection -> CaseCodeMapper: ReceiveVerifyUpdate()
7. CaseCodeMapper -> ViewBlogsSearchBlock: Search()

8. ViewBlogsSearchBlock -> CaseCodeMapper: Perform()
9. CaseCodeMapper -> ViewBlogsSearchBlock: Display()

This message sequence illustrates the flow of interactions between the objects involved in the Case Code Mapper system. Each message represents a method call or interaction between objects, facilitating the functionality of the system.

3.9 Description of Front end and Backend Development

Front-end Technologies:

Front-end technologies are responsible for creating the user interface and facilitating user interactions.

HTML (HyperText Markup Language):

HTML is used to structure the content of the web pages in Case Code Mapper.

It defines the layout of various elements such as navigation bars, forms, buttons, and text fields.

HTML provides the foundation upon which the visual elements of the application are built.

CSS (Cascading Style Sheets):

CSS is utilized to style the HTML elements and enhance the visual appearance of the website.

It defines the colors, fonts, spacing, and layout of the elements to ensure a cohesive and visually appealing design.

CSS plays a crucial role in creating a user-friendly and aesthetically pleasing interface for legal professionals interacting with the application.

JavaScript:

JavaScript is employed to add interactivity and dynamic functionality to the Case Code Mapper website.

It handles client-side interactions such as form validation, asynchronous data fetching, and DOM manipulation.

JavaScript enhances the user experience by providing real-time updates, interactive features, and responsiveness to user actions.

Back-end Technologies:

Back-end technologies are responsible for server-side logic, data processing, and communication with the database.

Python with Flask:

Python, a versatile programming language, is used in conjunction with Flask, a micro web framework, for back-end development.Flask facilitates the creation of web applications by providing tools and libraries for routing, request handling, and response generation.Python with Flask is employed to implement server-side logic, handle HTTP requests from the front end, and generate dynamic content.Flask interacts with the database to retrieve, store, and manipulate data related to case codes, legal resources, and user accounts.

XAMPP (with PHP):

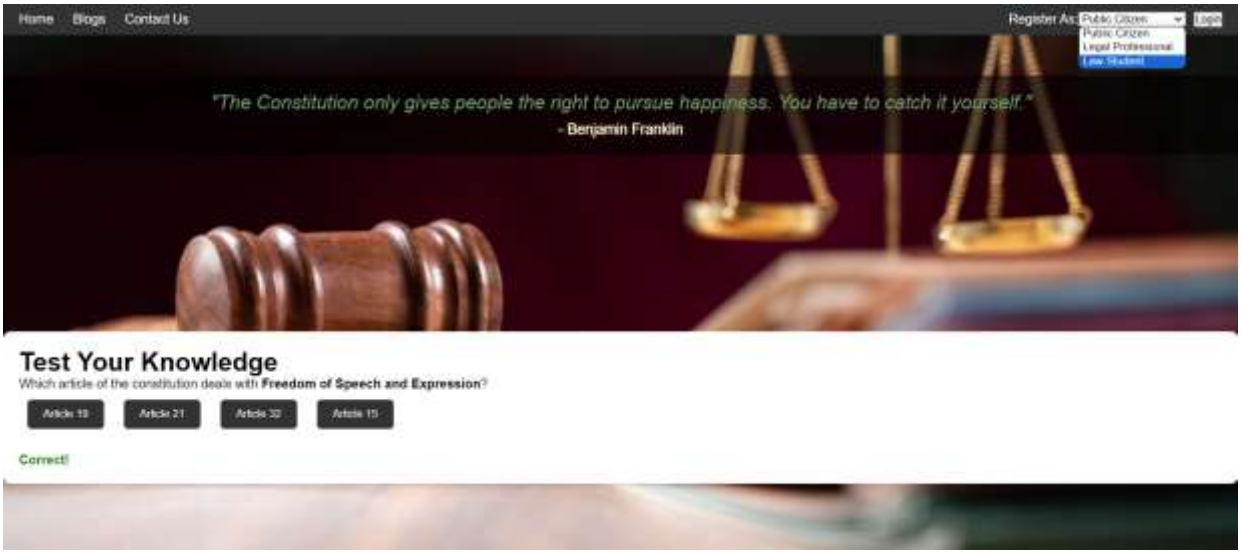
XAMPP is a web server solution stack that includes Apache, MySQL, PHP, and Perl.In the context of Case Code Mapper, XAMPP is used to set up a local development environment for testing and debugging PHP scripts.PHP scripts, integrated with XAMPP, can handle server-side tasks such as user authentication, session management, and database interactions.XAMPP provides a convenient platform for hosting and running PHP-based back-end components of the application.

By leveraging front-end technologies (HTML, CSS, JavaScript) for creating a visually appealing and interactive user interface, and back-end technologies (Python with Flask, PHP with XAMPP) for implementing server-side logic and database functionality, Case Code Mapper can deliver a comprehensive solution that meets the needs of legal professionals in managing case codes and accessing legal resources.

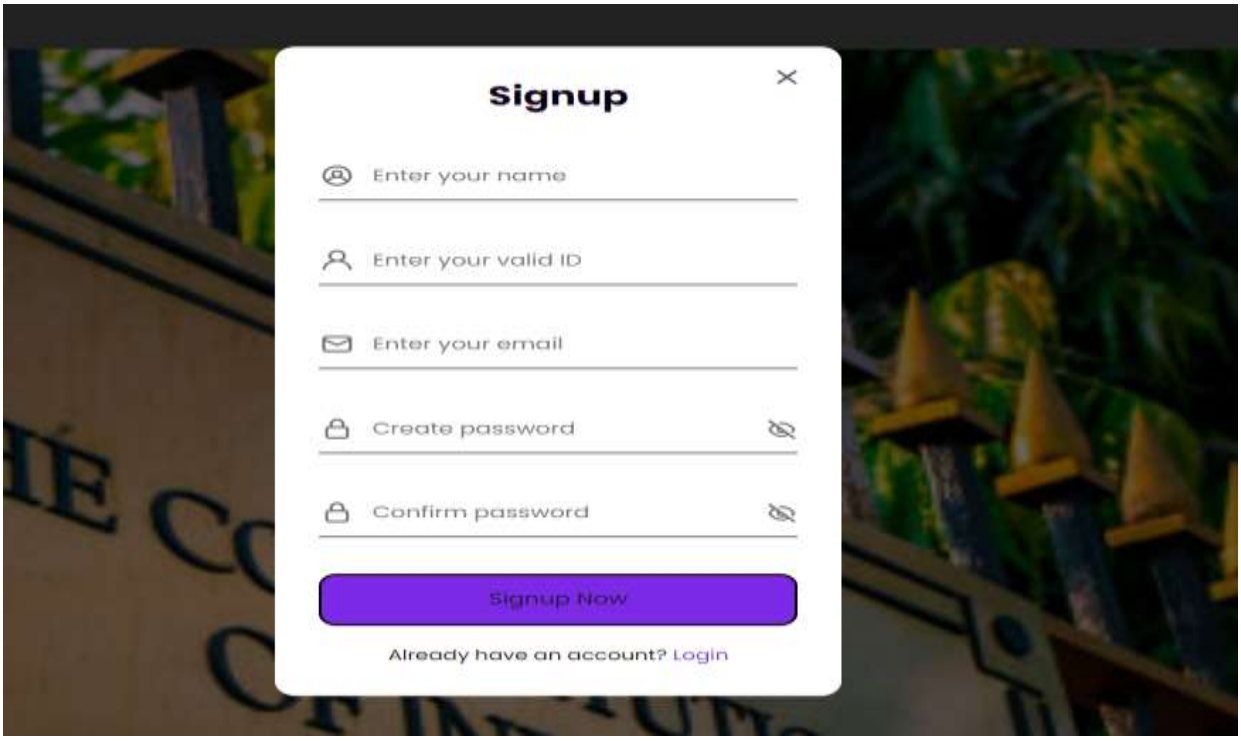
3.10 Interfaces and output of Front end and Backend

Frontend Interfaces

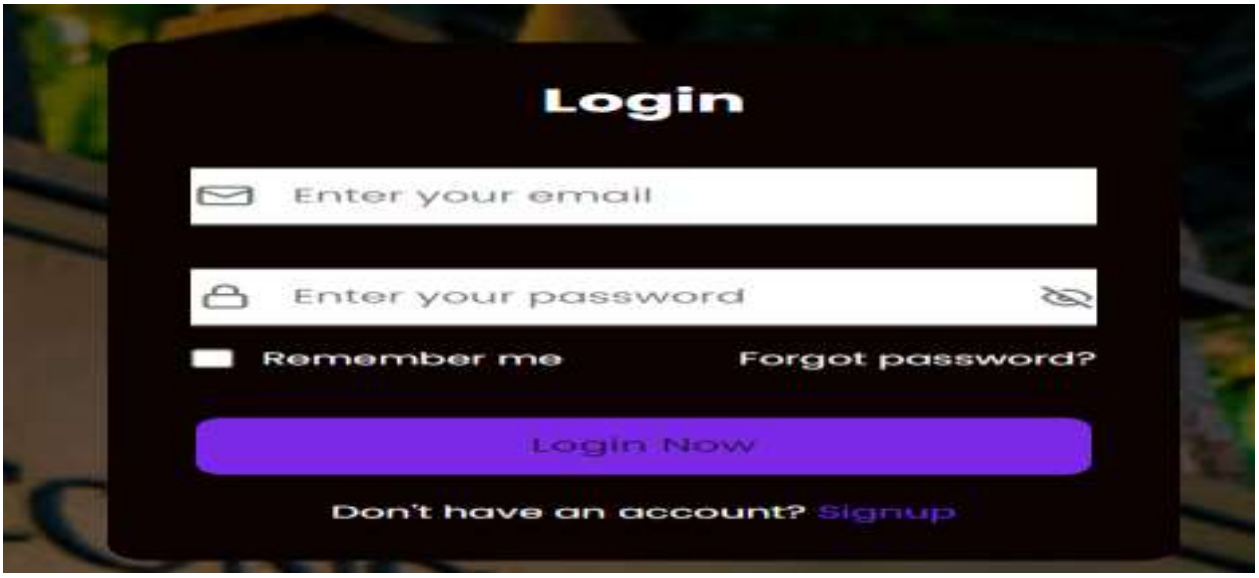
1. The page where User to describe whether he is login as public citizen or legal professional or law student.



2. Signup Page for Legal Professionals and Law Students (Similar as they both have to enter their respective ids)

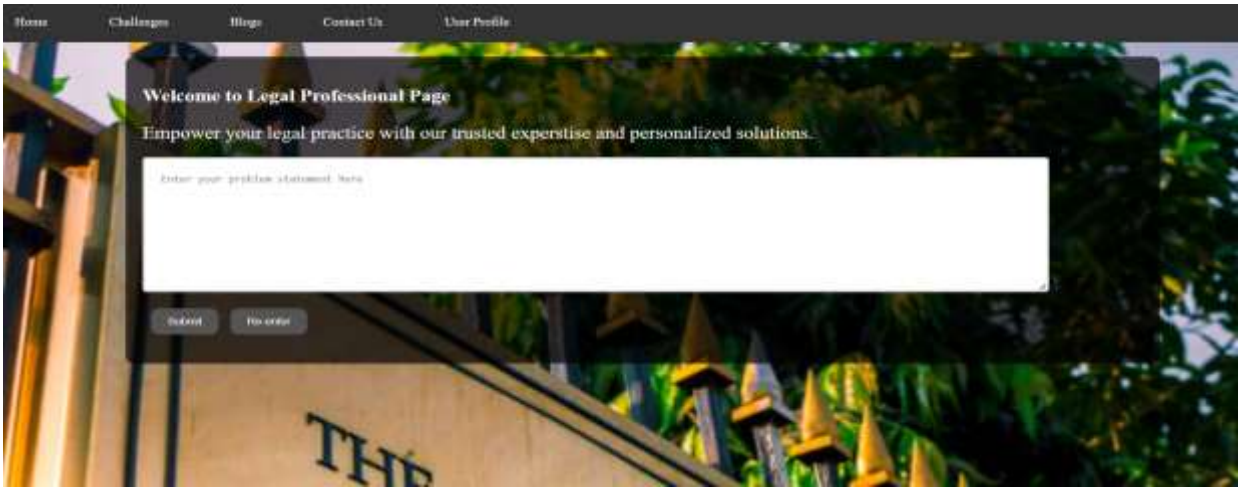


3. The Login Page looks like below for all types of users.

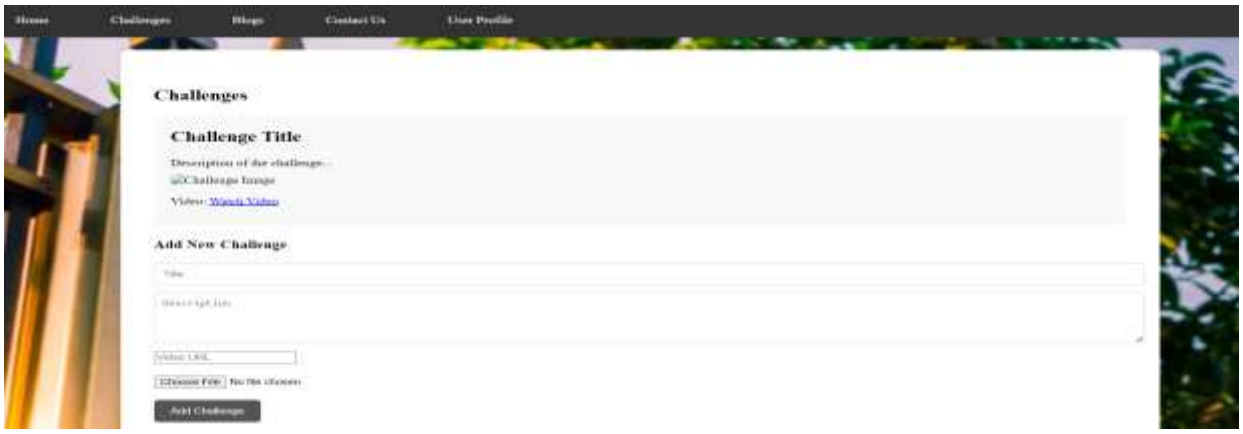


4. Database and the tables for legal professionals(legal), Law Students(law), Public Citizens(users)

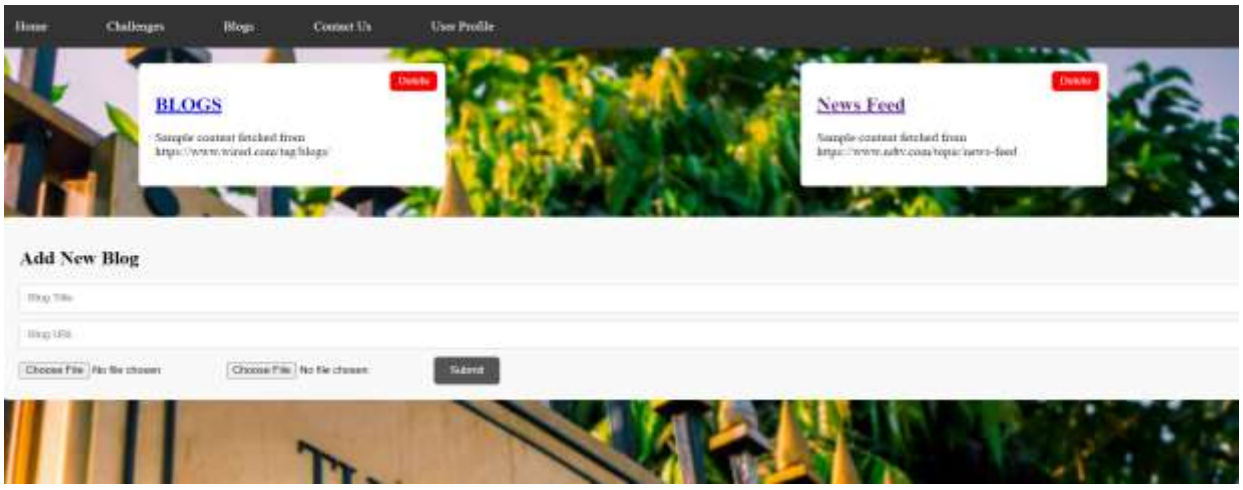
5. Legal Professional Home Page



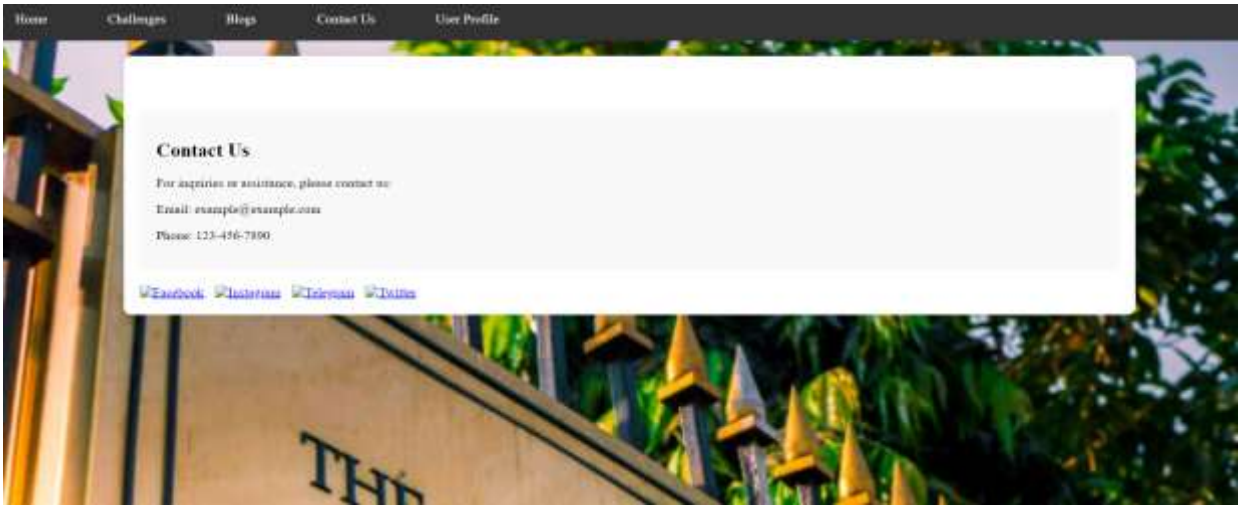
6. Legal Professional Challenge Page as they have access to add challenges like research, and case studies on crime incidents.



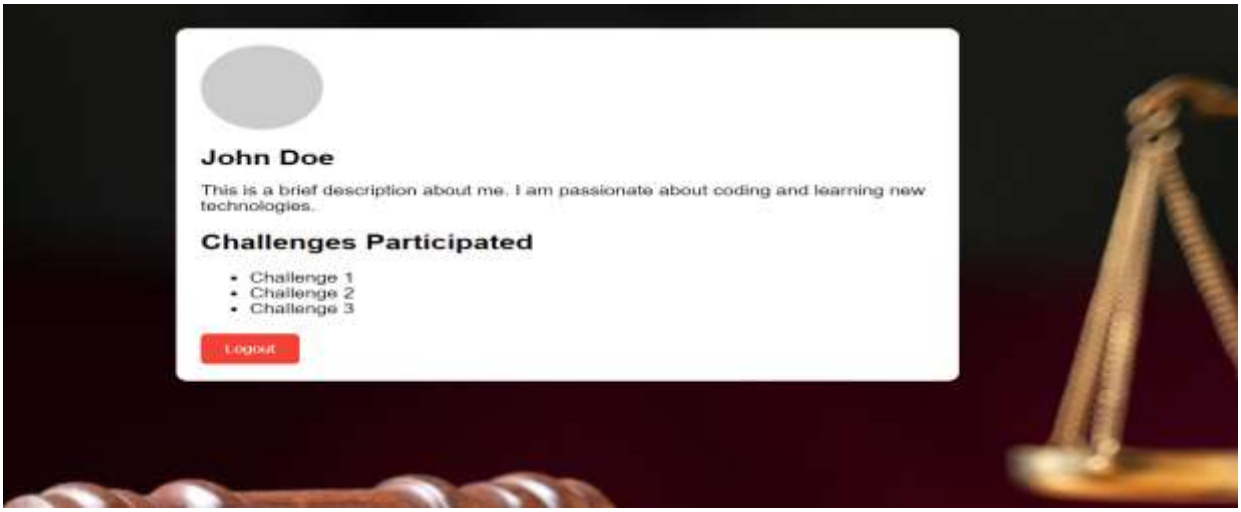
7. The Legal Professional Blogs Page Looks like below and note that they can also add the reports or their blogs which have special access to legal professionals, law students and some public citizens.



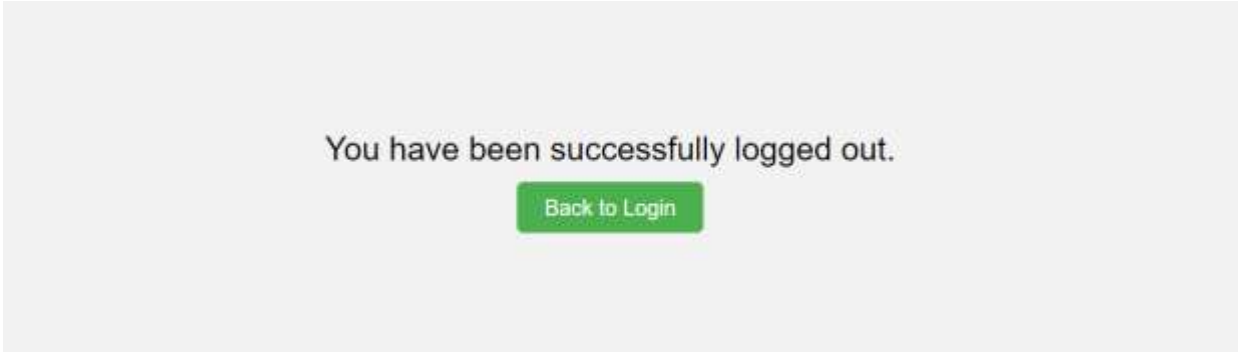
8. Contact Us (same for all types of users)



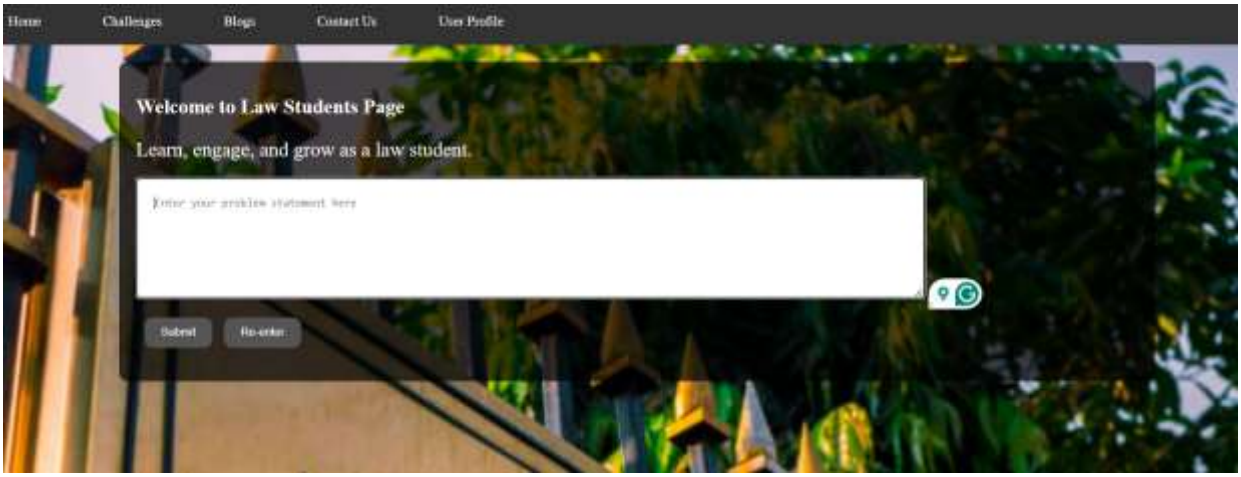
9. Sample User Profile



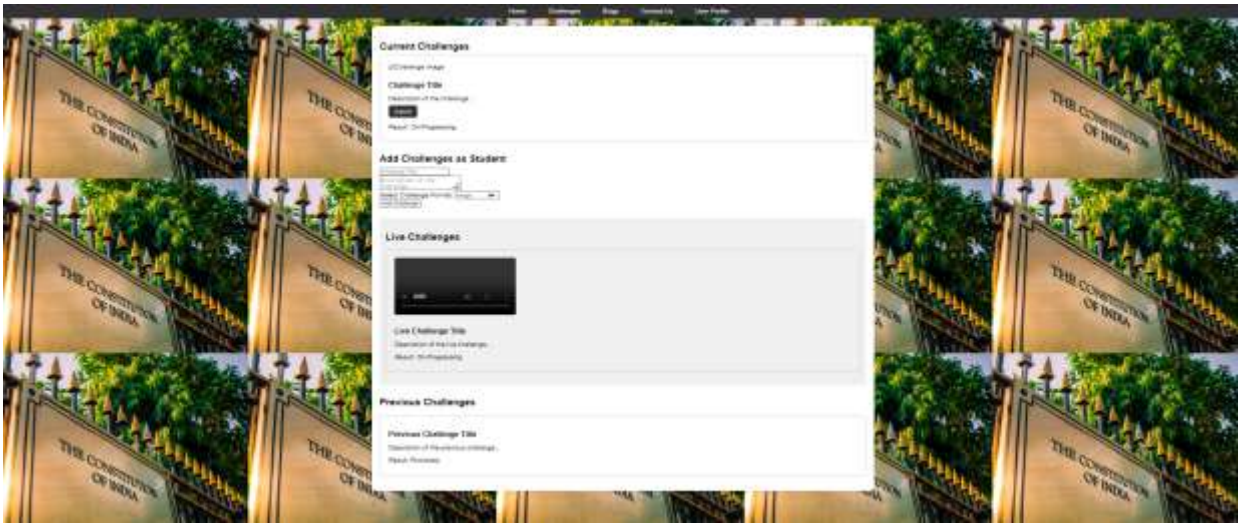
10. Logout Page (By clicking on back to login you navigate to the home page)



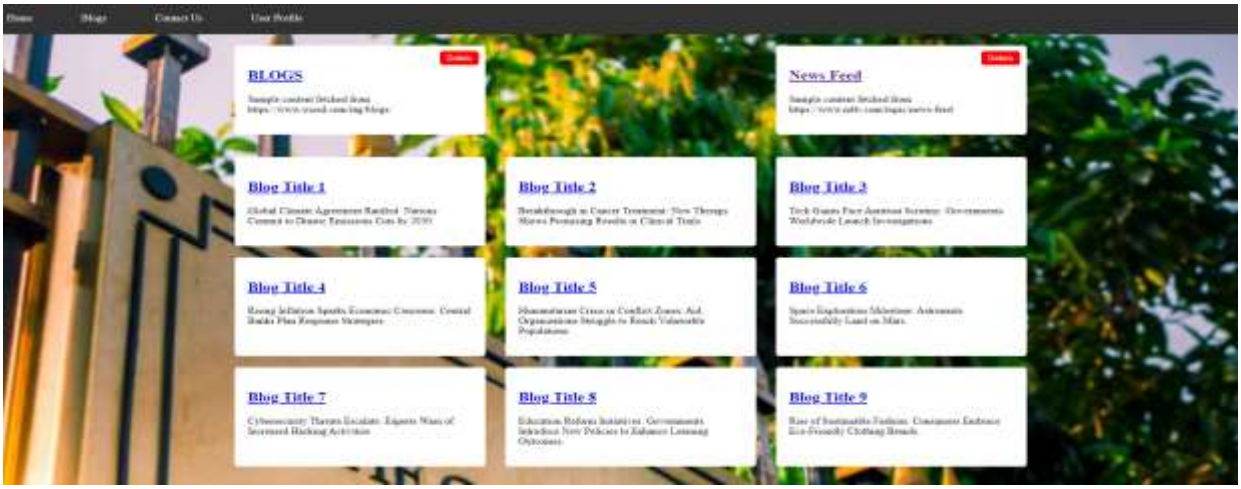
11. Law student's home page



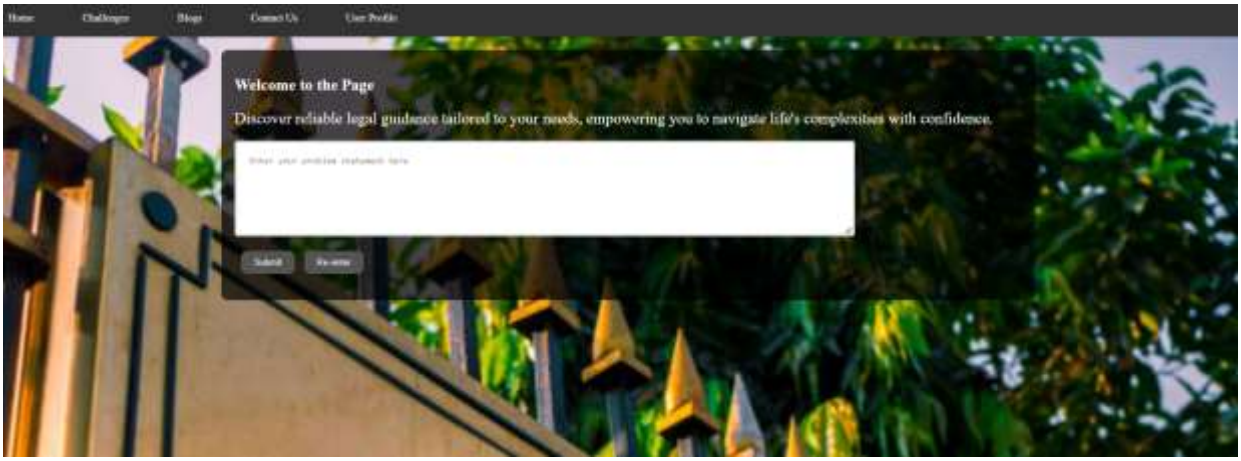
12. Law student challenges page where they can add challenges as students and participate in the challenges assigned by their professors, legal professionals, and law students.



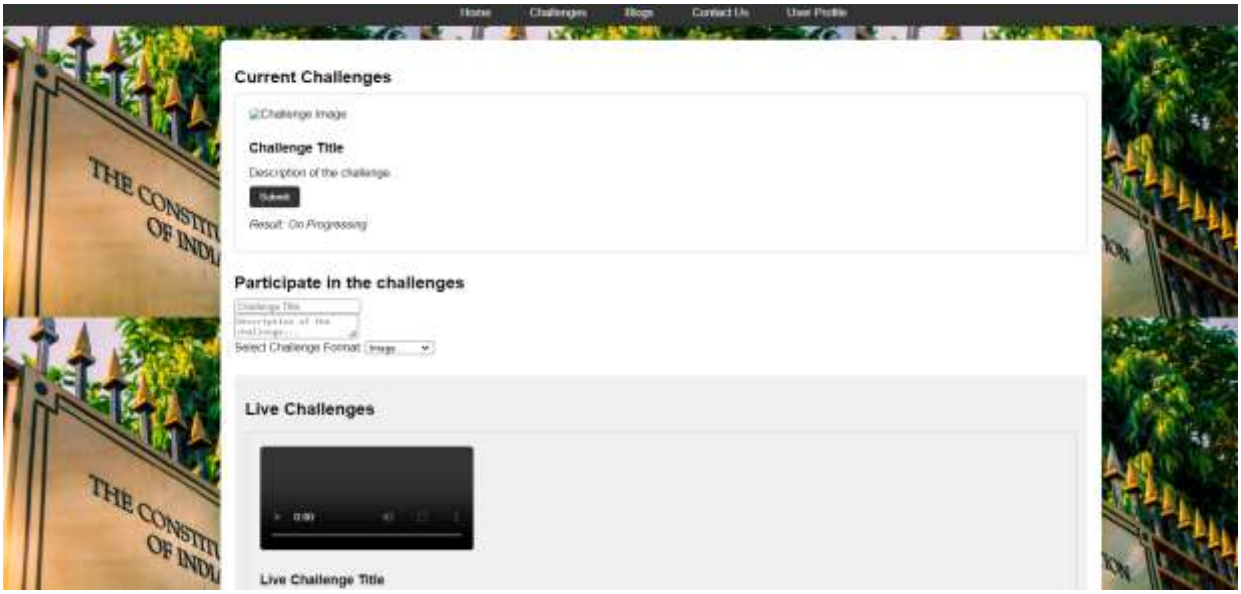
13. Law Students Blog Page



14. Public Citizens Home Page



15. Public Citizens Challenges Page (Only participate)



Backend Interfaces

16. Enter the problem statement input 1

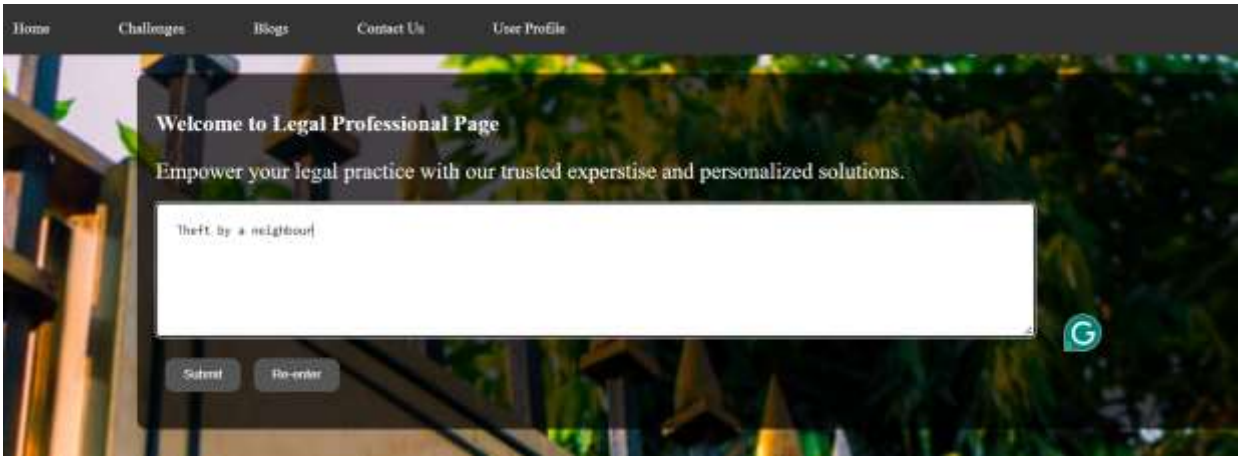
17. Output 1 for the input 1 problem statement (respective ipc sections)

Suggested IPC Sections:

Section	Offense	Punishment
IPC_302	Murder	Death or Imprisonment for Life + Fine
IPC_307	Attempt to murder	10 Years + Fine
IPC_396	Murder in Dacoity	Death, Imprisonment for Life, or Rigorous Imprisonment for 10 Years + Fine
IPC_366B	Importation of girl from foreign country	10 Years + Fine
IPC_364	Kidnapping or abducting in order to murder	Imprisonment for Life or Rigorous Imprisonment for 10 Years + Fine
IPC_303	Murder by a person under Sentence of imprisonment for life	Death
IPC_402	Being one of five or more persons assembled for the purpose of committing Dacoity	Rigorous Imprisonment for 7 Years + Fine
IPC_198	Using as a true certificate one known to be false in a material point	As for False Evidence
IPC_189	Threatening a public servant with injury to him or one in whom he is interested, to induce him to do or forbear to do any official act	2 Years or Fine or Both

Table 2: Output 1 of the IPC Sections for the input 1 problem statement

18. For another problem statement i.e., input 2



19. Output 2 for the input 2

Suggested IPC Sections:

Section	Offense	Punishment
IPC_379	Theft	3 Years or Fine or Both
IPC_314	Death caused by an act done with intent to cause miscarriage	10 Years + Fine
IPC_380	Theft in a building, tent or vessel	7 Years + Fine
IPC_382	Theft, after preparation having been made for causing death, or hurt, or restraint or fear of death, or of hurt or of restraint, in order to the committing of such theft, or to retiring after committing it, or to retaining property taken by it	Rigorous Imprisonment for 10 Years + Fine
IPC_439	Running vessel ashore with intent to commit theft, etc.	10 Years + Fine
IPC_381	Theft by clerk or servant of property in possession of master or employer	7 Years + Fine
IPC_315	Act done with intent to prevent a child being born alive, or to cause it to die after its birth	10 Years or Fine or Both
IPC_304	Culpable homicide not amounting to murder, If act by which the death is caused is done with intention of causing death, etc.	Imprisonment for Life or 10 Years + Fine
IPC_356	Assault or criminal force in attempt to commit theft of property worn or carried by a person	2 Years or Fine or Both

Table 3: Output 2 of the IPC Sections for the input 2 problem statement

4. Testing

4.1 Functional Testing

4.1.1 Login Module:

Functional testing verifies that software functions correctly according to requirements. Test cases are derived from functional specifications and executed to validate each feature. It ensures that the software meets user needs and performs as expected. This module involves testing the login and user redirection based on the user type selected.

Test Cases:

- User Type Selection: Verify that the dropdown for user type selection correctly lists "Public Citizen," "Legal Professional," and "Law Student."
- Redirection on Login: Test that clicking the "Login" button redirects users to the correct pages based on their selected user type. This involves:
- - Public Citizen: Redirect to a public login page.
- - Legal Professional: Redirect to a professional login page.
- - Law Student: Redirect to lawstudents.html.
- Navigation Links: Ensure that all navigation links ("Home," "Blogs," "Contact Us") are clickable and redirect to the correct pages or sections.

4.1.2 Result Module

This module is designed to display legal sections, offenses, and their corresponding punishments.

Test Cases:

- Table Display: Verify that the table is properly formatted with headings for Section, Offense, and Punishment.
- Data Population: Ensure that the table populates dynamically with data for each section, offense, and punishment. This may require integration testing with backend services to confirm data retrieval and display.
- Visual Inspection: Check for any CSS issues that might affect readability and accessibility, such as text color, font size, and background contrast.

4.1.3 Main Module

This main module page is targeted at law students, public citizens, and legal professionals allowing them to submit problem statements and interact with educational content.

Test Cases: Form Submission

- Text Entry: Test that users can enter text into the problem statement text area.
- Form Submission: Ensure that the form submits data correctly and interacts with backend services to display results or confirmation messages.
- Re-enter Button: Verify the functionality of the "Reenter" button to clear the text area field.
- Navigation and Links: Confirm that all links in the navigation bar (e.g., "Home," "Challenges," "Blogs," "Contact Us") work correctly and lead to the intended pages or sections.
- Background and Style: Check that the background image loads correctly and that all text is readable over the background.
- Problem Statement Handling: Ensure the JavaScript fetch operation handles errors gracefully and displays appropriate messages to the user.

4.2 Path Coverage Testing

Path testing involves testing different paths or scenarios within a software application to ensure comprehensive coverage. Here's how path testing could be applied to the described application flow.

Cyclomatic Complexity = E-V+2

E: Number of edges

V: Number of vertices

4.2.1 Start Application

Pseudocode: 1. Start the application

Control Flow Diagram:

Figure 9: Start application of the control flow diagram

4.2.2 Navigate to Registration Form(2)

Pseudocode: 2. Navigate to Registration Form.

Control Flow Diagram:

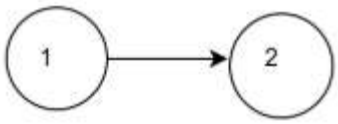


Figure 10: Navigate to the Registration Form of the control flow diagram

Complexity = 1-2+2=1

4.2.3 Access Registration Form(3)

Pseudocode: 3. Access Registration Form

Control Flow Diagram:

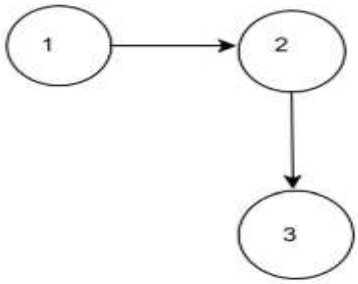


Figure 11: Access the Registration Form of the control flow diagram

Complexity = 2-3+2=1

4.2.4 Have Account (4)

Pseudocode: 4. Check if the user already has an account

Control Flow Diagram

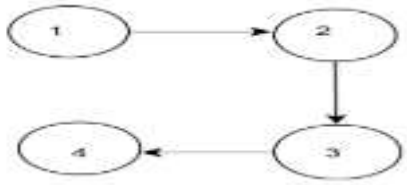


Figure 12: Have an Account of the control flow diagram

Complexity = 3-4+2=1

4.2.5 Create Account Form

Pseudocode: 5. If the user does not have an account: Present the create account form

Control Flow Diagram:

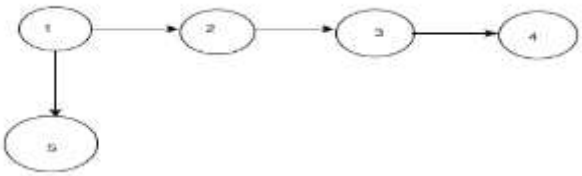


Figure 13: Create an Account Form of control flow diagram

Complexity = 4-5+2 = 1

4.2.6 Enter Email and Password

Pseudocode: 6. User enters their email and password

Control Flow Diagram:

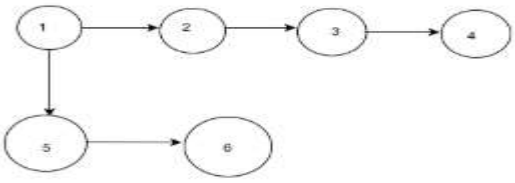


Figure 14: Enter the Email and Password of the control flow diagram

Complexity = 5-6+2 = 1

4.2.7 Enter Data

Pseudocode: 7. User enters necessary profile information

Control Flow Diagram:

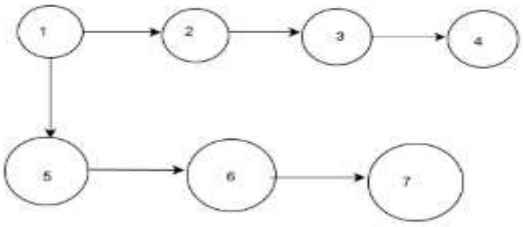


Figure 15: Enter the Data of the control flow diagram

Complexity = 6-7+2 = 1

4.2.8 Save Data

Pseudocode: 8. Save the entered data

Control Flow Diagram:

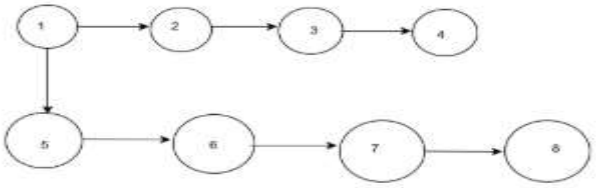


Figure 16: Save Data of control flow diagram

Complexity = 7-8+2 = 1

4.2.9 Process Login

Pseudocode: 9. Admin processes the login information

Control Flow Diagram:

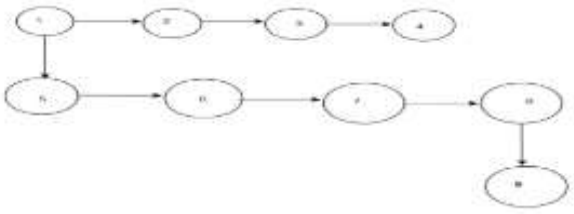


Figure 17: Process Login of Control Flow Diagram

Complexity = 8-9+2 = 1

4.2.10 Validate User Access

Pseudocode: 10. Validate whether the user is a Legal Professional

Control Flow Diagram:

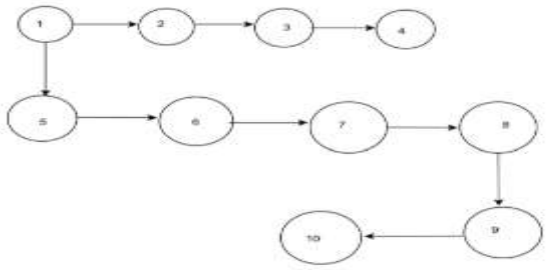


Figure 18: Validate user Access to the control flow diagram

Complexity = 9-10+2=1

4.2.11 Select Below One (Blogs/Prompts)

Pseudocode: 11. Based on user validation:

11.1. If the user is a Legal Professional:

11.1.1 Provide options to view, edit, and create content

11.2. Otherwise:

11.2.1 Display blogs

Control Flow Diagram

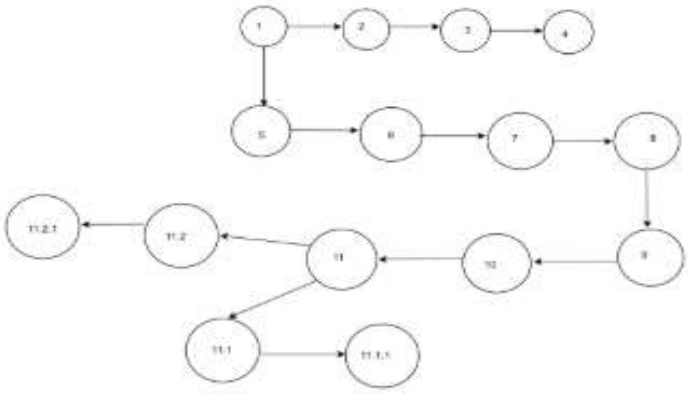


Figure 19: Blogs/prompts of control flow diagram

Complexity = 14 - 15+2=1

4.2.12 Enter Data

Pseudocode: 12. User enters the required information for creating/editing content or submitting queries

Control Flow Diagram:

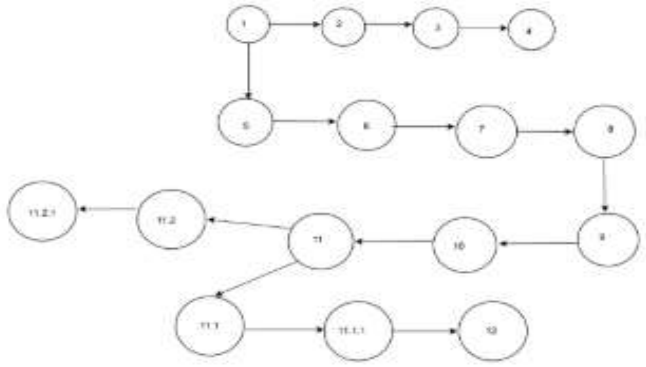


Figure 20: Enter Data of control flow diagram

Complexity = 15-16+2=1

4.2.13 Submitting Data

Pseudocode: 13. User submits the entered data

Control Flow Diagram:

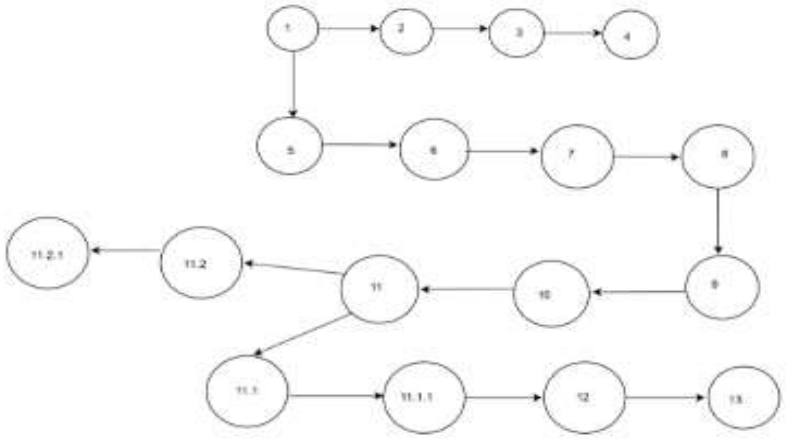


Figure 21: Submitting data of control flow diagram

Complexity = 16-17+2=1

4.2.14 Save Data

Pseudocode: 14. Save the submitted data

Control Flow Diagram:

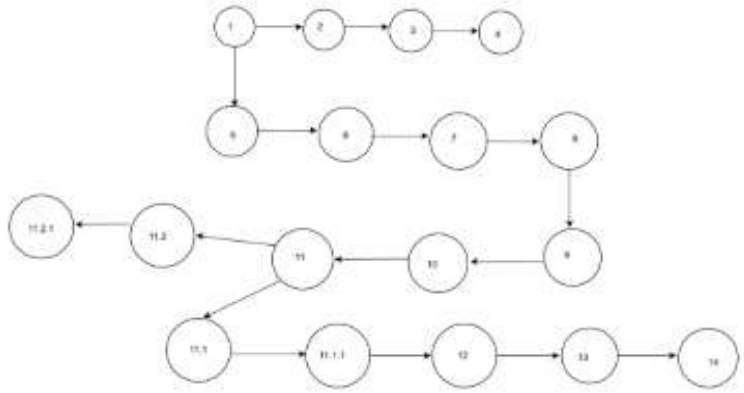


Figure 22: Save Data of control flow diagram

Complexity = 17-18+2=1

4.2.15 Display Query

Pseudocode: 15. Display the submitted query or the created/edited content, depending on the user's action

Control Flow Diagram:

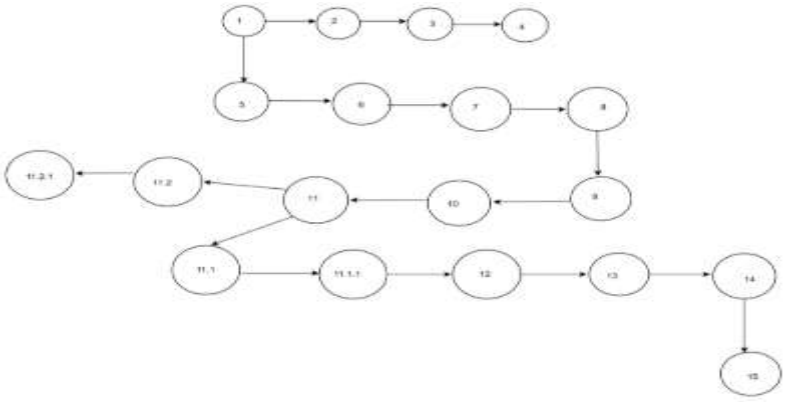


Figure 23: Display Query of control flow diagram

Complexity = 18-19+2=1

4.2.16 Stop

Pseudocode: 16. Display the submitted query or the created/edited content, depending on the user's action

Control Flow Diagram:

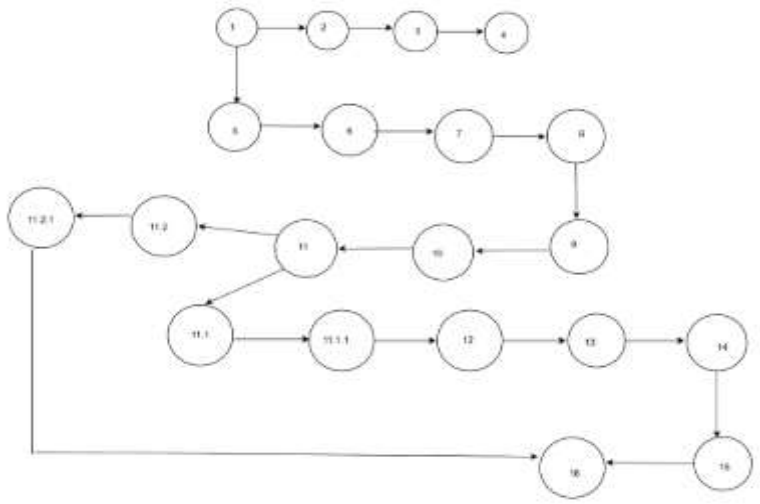


Figure 24: Stop of Control flow diagram

Complexity = 20-20+2=2

5. Future Work

In the future, the "Case Code Mapper: Unraveling Legal Section" project could focus on several key areas to further enhance its capabilities and user experience. This includes expanding its legal database by continuously updating legal sections and cases to ensure that the platform remains current and comprehensive. Integration of advanced artificial intelligence features could also be explored to improve the platform's analytical capabilities, enabling more accurate legal section recommendations to have 100 percent accuracy.

Additionally, efforts could be made to enhance the user interface, making it even more intuitive and accessible to users of all levels of legal expertise. The development of a mobile application version could further increase accessibility, allowing users to access legal assistance conveniently from their smartphones. Multilingual support could be added to cater to users from diverse linguistic backgrounds, while community features such as discussion forums could foster collaboration and knowledge-sharing among users. Implementing analytics tools would provide valuable insights into user interactions and behaviors, guiding future enhancements. Collaborations with legal institutions and continuous updates to ensure regulatory compliance would also be crucial for the project's success. Overall, these future endeavors aim to further empower users and solidify the platform's position as a valuable tool in the field of legal research and decision-making.

6. Limitations

Our current project has certain limitations as we can upgrade it in the future. The limitations are as follows:

- The User needs to give some similar, relevant, or meaningful words in the sentences to identify the correct IPC Sections.
- The current project works on only web browsers that need to be built for mobile applications.
- Text description is only subjected to the English language.

7. Conclusion

The "Case Code Mapper: Unraveling Legal Section" project is like having a smart legal assistant at your fingertips. It helps people understand legal issues easily, without needing to speak with a lawyer directly. The platform is designed to be user-friendly, catering to everyone from legal experts to those who are new to the legal world.

By using advanced technology like AI algorithms, and NLP it quickly provides accurate legal advice based on what users need. Plus, it's accessible from anywhere with an internet connection, making legal help available to more people, even in remote areas. Security is a big focus too, ensuring that all legal information remains safe and confidential. And the project is always evolving based on user feedback, so it stays relevant and helpful over time. In short, it's a game-changer for legal research, making it easier and more accessible for everyone involved.

References

1. Srivastav, A., & Prajapat, S. (2022). Text similarity algorithms to determine Indian Penal Code sections for offence report. IAES International Journal of Artificial Intelligence (IJ-AI), 11(1), 34–40. DOI: 10.11591/ijai.v11.i1.pp34-40[1].

2. Srivastav, A., Prajapat, S. (2024). Investigation of Decision Support System for Indian Penal Code Section Using Similarity Algorithm and Fuzzy Logic. In: Naik, N., Jenkins, P., Grace, P., Yang, L., Prajapat, S. (eds) Advances in Computational Intelligence Systems. UKCI 2023. Advances in Intelligent Systems and Computing, vol 1453. Springer, Cham. https://doi.org/10.1007/978-3-031-47508-5_50[2].

3. Indian-penal-code-ipc-sections-information_dataset[3].

4. Indian Penal Code,1860 | IPC Sections | lawrato.com[4].

5. Code of Criminal Procedure | CrPC Sections | lawrato.com[5].

6. List of All Articles (1 to 395) And Parts (1 to 22) Of The Indian Constitution[6].

7. Github Code Repository Case-Code-Mapper-Unraveling-Legal-Sections[7].