LAW ACT FINDER

Submitted By:

Choudhary Ameer Hamza

Malik Daniyal

Supervisor

Mam Fatima Yousaf

DEPARTMENT OF INFORMATION TECHNOLOGY INSITUTE OF SOUTHEREN PUMJAB MULTAN

LAW ACT FINDER

Chaudhary Ameer Hamza

BSIT-023R20-29

Malik Daniyal

BSIT-023R20-48

DEPARTMENT OF CS/IT

INSTITUTE OF SOUTHERN PUNJAB (ISP)

Table of Contents

Chapter 1

1. Introduction:

Legal cases are disputes or conflicts between parties that are brought before a court or other legal tribunal for resolution. In a legal case, one party (the plaintiff) typically brings a claim or lawsuit against another party (the defendant) to seek redress for some harm or injury that has been suffered. Legal cases can arise in a wide range of contexts, including criminal cases, Health lawsuits, family law disputes, and administrative proceedings. The parties to a legal case are typically represented by attorneys who argue their positions before a judge or jury, and the outcome of the case is determined by the legal rules and procedures that govern the particular type of case involved. Legal cases are managed through a series of procedures and steps that are designed to ensure that the case proceeds fairly and efficiently. The precise management of a legal case can vary depending on the type of case involved and the jurisdiction in which it is being heard.

Law is an associate which is dealing with cases like criminal cases, Health cases, family cases Labour cases, Property cases Environmental cases, financial cases etc. We are working for them. We are developing legal case management software for law associates.

1.1 Current System:

Functions	**Current system**	**New system**	**App4legal software**	**Case management system**
Login page		✓		✓
Admin portal		✓	✓	✓
Lawyer portal		✓		
Client portal		✓	✓	✓
Document management	✓	✓	✓	✓

Calendar and task management		✓	✓	✓
Client billing platform		✓	✓	
Reminders & Notifications		✓	✓	
Litigation case management		✓	✓	
Contracts		✓	✓	
Matters		✓	✓	

1.1.1 Information engineering (Questionnaires):

I have used the method of questions.

Questions:

Q. Who is the owner of law associate?

Ans. Akmal LAW Associate

Q. How many members are associated with this firm?

Ans. 100 to 150 members are associated with this firm.

Q. which types of cases law associate deal?

Ans. law associate deals with all types of legal cases including criminal cases, civil cases, family cases, banking cases, FIA cases etc.

Q. in which region do they work?

Ans. In all Punjab.

Q. where the members of law located?

Ans. Basically they are located at the main office in Multan but in their region also.

Q. In which court law associate deal cases?

Ans. In session court, high court and some cases in Supreme Court.

Q. Which system is being used currently by law associate?

Ans. Manual system is currently being used.

Q. How the members of law deal cases?

Ans. They deal with their cases individually also and with law associates also.

Q. How is the information of cases being stored?

Ans. The cases information is being stored manually

1.1.2 Basic Diagram:

Use case:

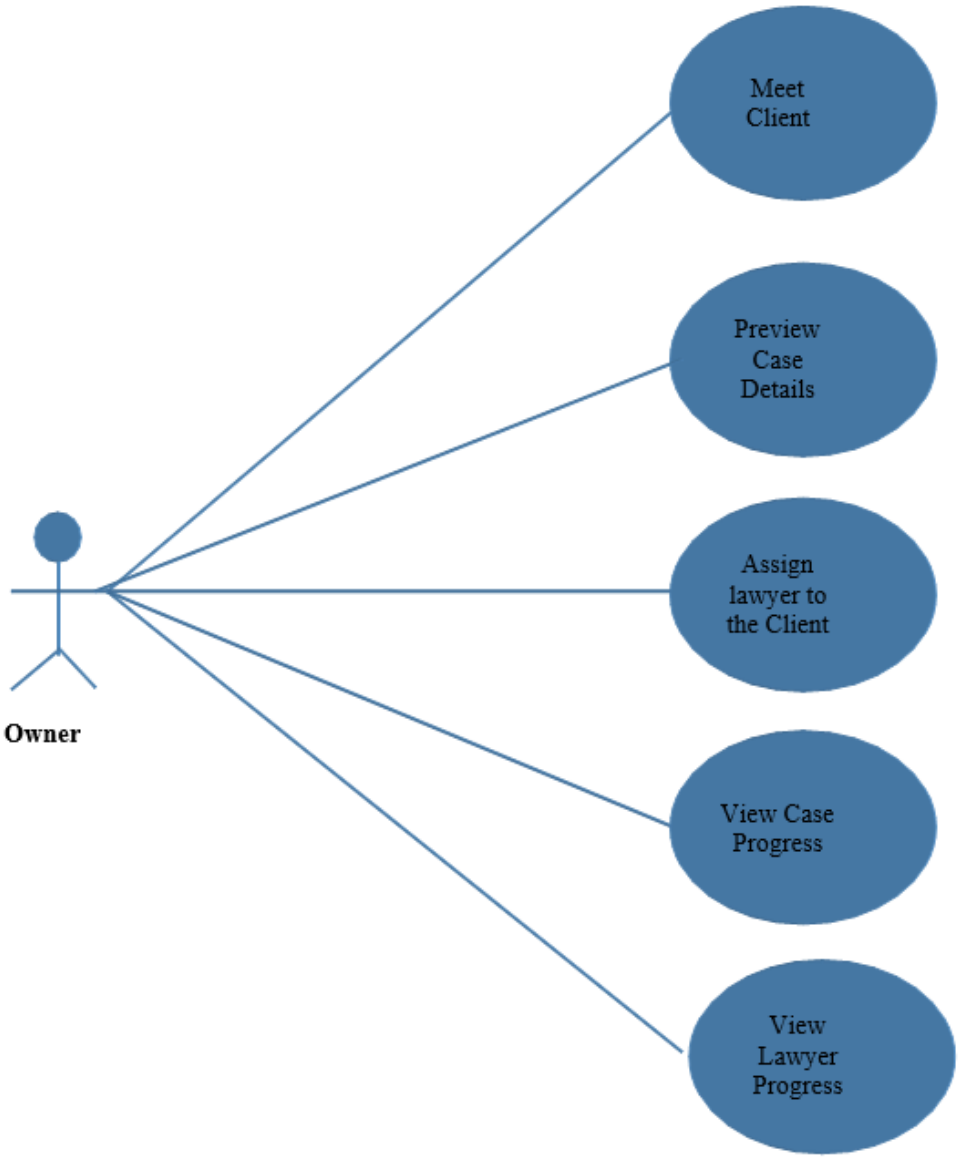


Figure 1.1 use case diagram

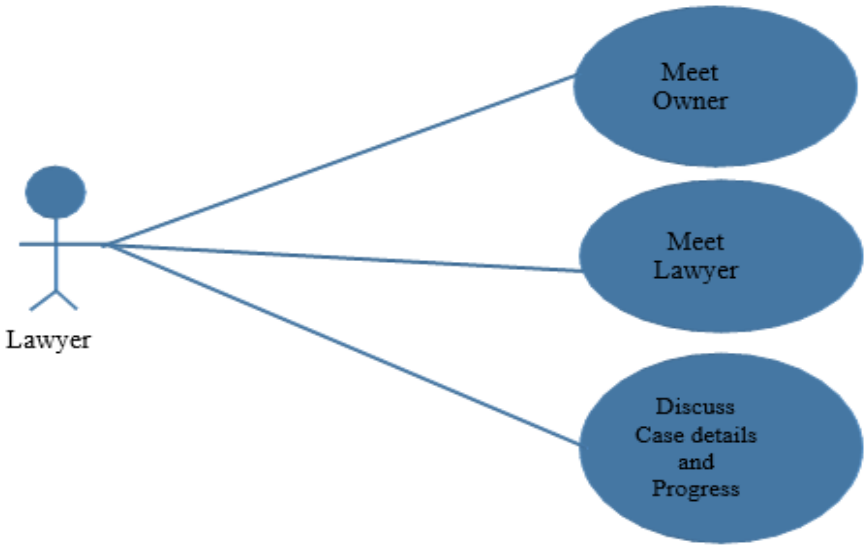


Figure 1.2 lawyer use case

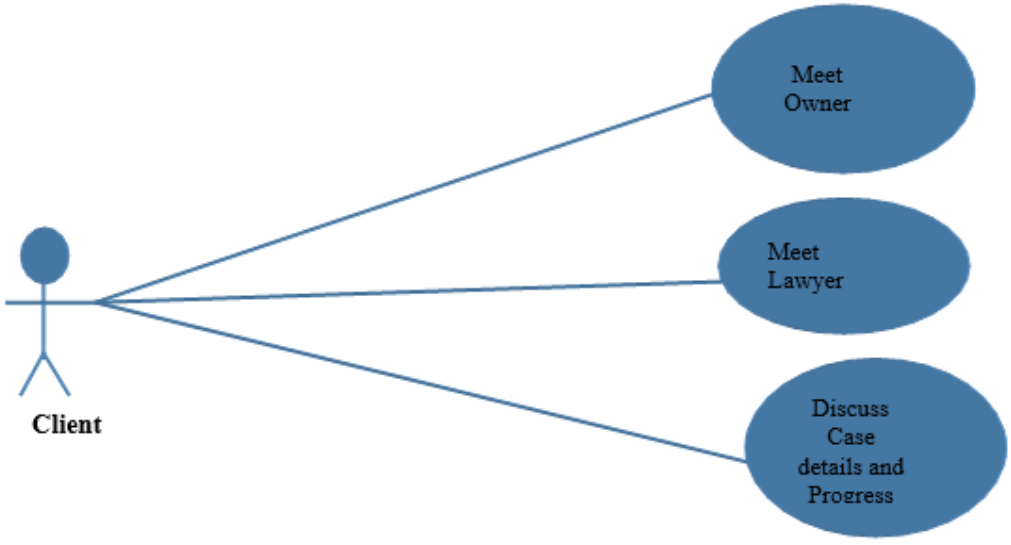


Figure 1.3 client use case diagram

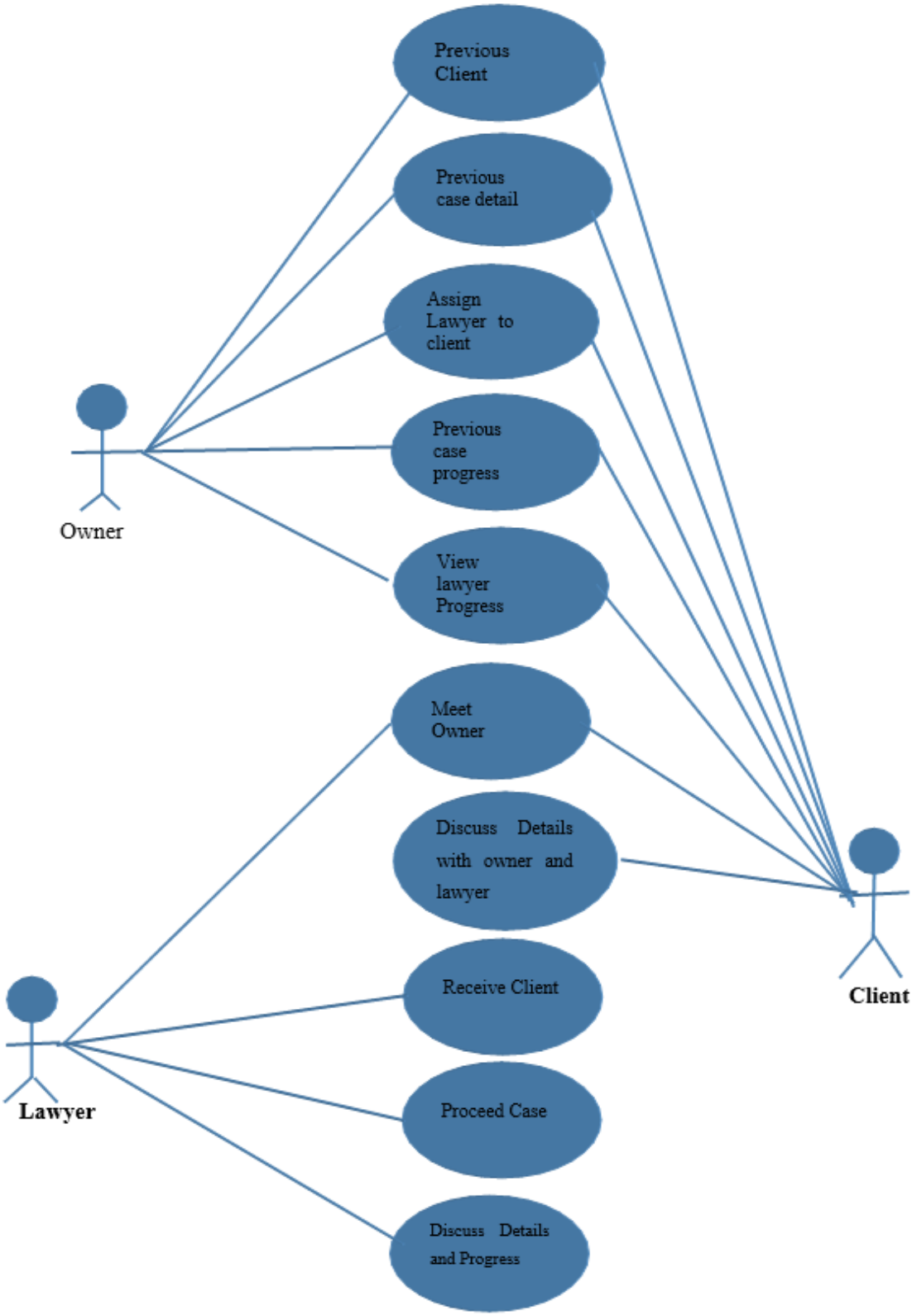


Figure1.4 overall use case

1.1.2 Activity diagram:

Figure1.5 activity diagram

Figure1.6 lawyer activity diagram

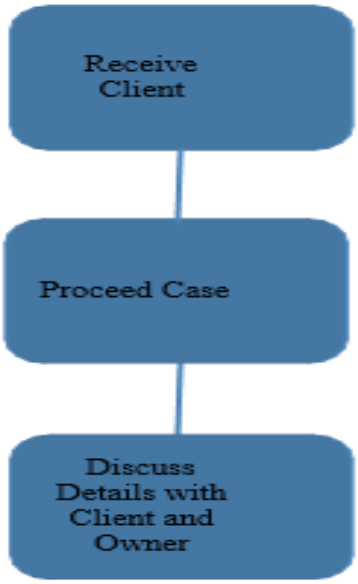


Figure1.7 client activity diagram

1.1.2 DFD:

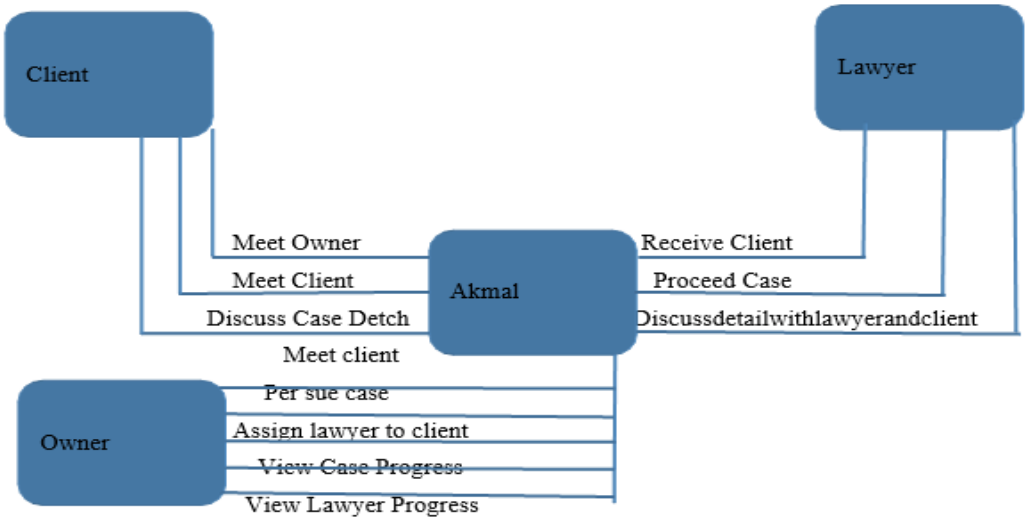


Figure1.8 Data flow diagram

1.1.3 Problem within current system:

The current system is working manually it mean law associate doesn't have any easy way to record their files, to check the progress. And there is also no easy way to chat with clients, to discuss details with clients. There is no easy way to get notified for the case. There is no easy way for the owner to check which client should visit which lawyer of the chamber. Manual processes can be time-consuming and require a lot of effort, resulting in slower progress and decreased productivity. Manual processes are prone to human error, which can result in mistakes such as incorrect data entry or missed deadlines. Without software tools, it can be challenging to keep track of important information and stay organized, which can lead to missed deadlines or lost data. Without digital tools, it can be challenging to share information or work collaboratively with colleagues or clients, which can result in communication breakdowns. Manual processes can be less efficient than automated processes, resulting in decreased productivity and higher costs. Overall, using software tools can help Law Associate to improve efficiency, reduce errors, increase productivity, and enhance collaboration and communication.

1.2 Comparison Table:

Functions	**Current system**	**New system**	**App4legal software**	**Case management system**
Login page		✓		✓
Admin portal		✓	✓	✓
Lawyer portal		✓		
Client portal		✓	✓	✓
Document management	✓	✓	✓	✓
Calendar and task management		✓	✓	✓
Client billing platform		✓	✓	
Reminders & Notifications		✓	✓	
Litigation case management		✓	✓	
Contracts		✓	✓	
Matters		✓	✓	

1.3 Basic Constraints:

- In this system, a lawyer is being assigned.
- In this system all, the functions are processed manually.
- In this system, the client cannot see the past performance of the lawyer.

- In this system, tasks are managed manually.
- In this system, contracts are managed manually.
- In this system, the owner tracks the performance of the lawyer manually.

1.4 Project proposal:

I will build a system that will automatically maintain the records of the Akmal law associate. My system will be able to store all past records of the lawyer so that a client should see the progress of the lawyer. In my system owner can manage the lawyers, can manage the client, can manage case info, can manage profiles, can manage user, can view and manage calendar. Owner can also track the performance of the lawyer. In my system owner will be able to manage client detail, can manage case detail, can manage case history, can manage profile. In my system client will be able to see case detail and upcoming calendar. My system will generate the signal as an alert for case hiring. My system will manage all the documents.

1.4.1 Methodology:

1.4.2 Law Management System Methodology

This methodology outlines the development process for a Law Management System (LMS):

1.4.3 Requirement Gathering:

Conduct interviews and workshops with lawyers, paralegals, and administrative staff to understand their needs, workflows, and pain points.

Identify key functionalities and desired system integrations.

1.4.4 System Design:

Develop a system architecture that addresses user requirements and data security.

Design user interfaces (UI) that are intuitive and user-friendly for legal professionals.

1.4.5 Data Modeling:

Define the data entities (e.g., clients, cases, documents) and their relationships within the system.

Establish data security protocols for client information and sensitive case data.

1.4.6 Development:

Develop the LMS features using chosen programming languages and frameworks.

Ensure proper integration with existing legal practice management systems or document management platforms (optional).

1.4.7 Testing and Quality Assurance:

Conduct rigorous testing to identify and fix bugs in functionality and user interface.

Perform security testing to ensure data integrity and system protection.

1.4.8 Deployment and Training:

Deploy the LMS in a production environment, ensuring secure access for authorized users.

Provide comprehensive training sessions for legal professionals on LMS features and workflows**.**

1.4.9 User Support:

Establish a user support system (e.g., knowledge base, ticketing system) to address user inquiries and resolve issues.

Offer ongoing training and support to ensure user adoption and maximize system benefits.

1.4.10 Feedback and Improvement:

Gather user feedback through surveys, interviews, and support tickets.

Utilize feedback for continuous improvement and development of new features and functionalities for the LMS.

1.4.11 Maintenance and Updates:

Implement a regular maintenance schedule to address bugs, security patches, and legal industry compliance updates.

Ensure ongoing system enhancements based on user feedback and evolving legal needs.This 9-step methodology provides a structured approach for developing a Law Management System that meets the specific needs of a legal practice while prioritizing functionality, data security, user experience, and continuous improvement

Chapter 2

2. Overall description:

2.1 Proposed system environment:

We are building a system that will manage all records of law associate. This system will properly manage all the documents and progress of every case. This system will provide extra facility to owner, client and lawyers.

2.2 Scope:

The scope of legal case management system can vary depending on the specific need of the law or legal department that is using it. Legal case management system is used to provide lawyers and legal professionals with a comprehensive tool for managing cases and workflow efficiently, reducing, administrative tasks and improving overall productivity and profitability.

It manages Lawyers detail.

- It manages Client detail.
- It manages Lawyer progress.
- It manages Case progress.
- It manages Case details.
- It manages task.
- It manages schedule.

2.3 Functional Requirements:

Functional requirement of the system is as below.

Admin Functional Requirement

1	Admin will login with username and password
2	Admin will manage Lawyers
3	Admin will manage client
4	Admin will check performance of lawyers

5	Admin will manage profile
6	Admin will manage user
7	Admin will view and manage calendar

Table 2.1 Admin Functional Requirement

Lawyer Functional Requirement

1	Lawyer will manage profile
2	Lawyer will manage client detail
3	Lawyer will manage case detail
4	Lawyer will manage case history
5	Lawyer will manage daily task calendar

Table 2.1 Lawyer Functional Requirement

Client Functional Requirement

1	Client will view case detail
2	Client will view case upcoming calendar

Table 2.1 Client Functional Requirement 2.4

2.4 Use case:

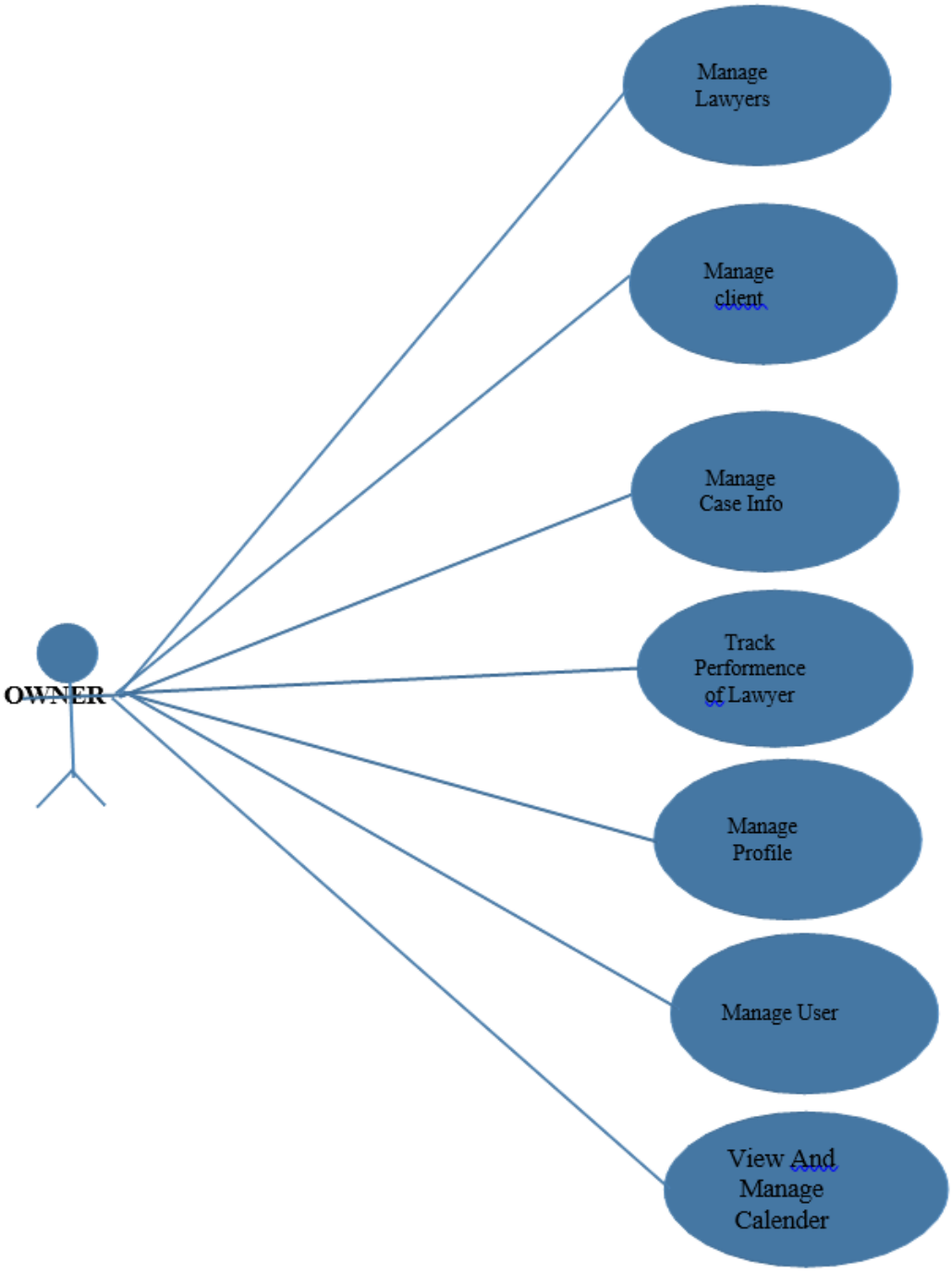


Figure 2.1 owner use case diagram

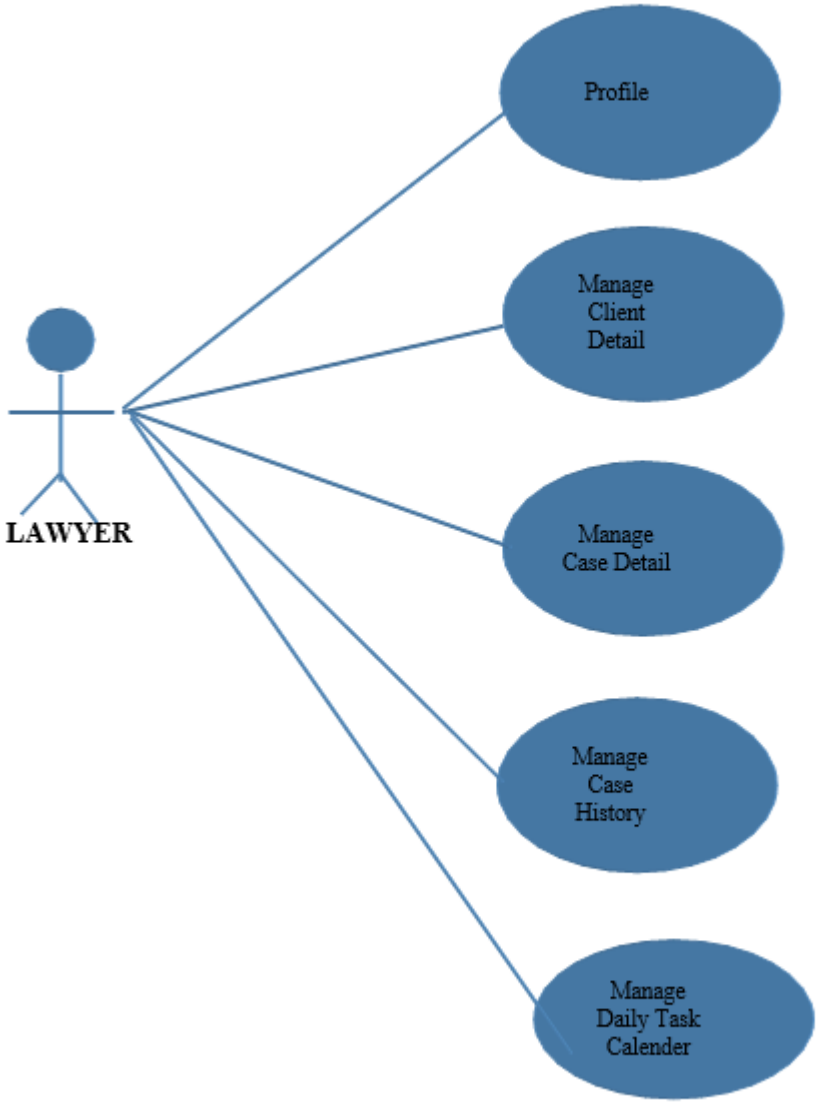


Figure2.2 lawyer use case diagram

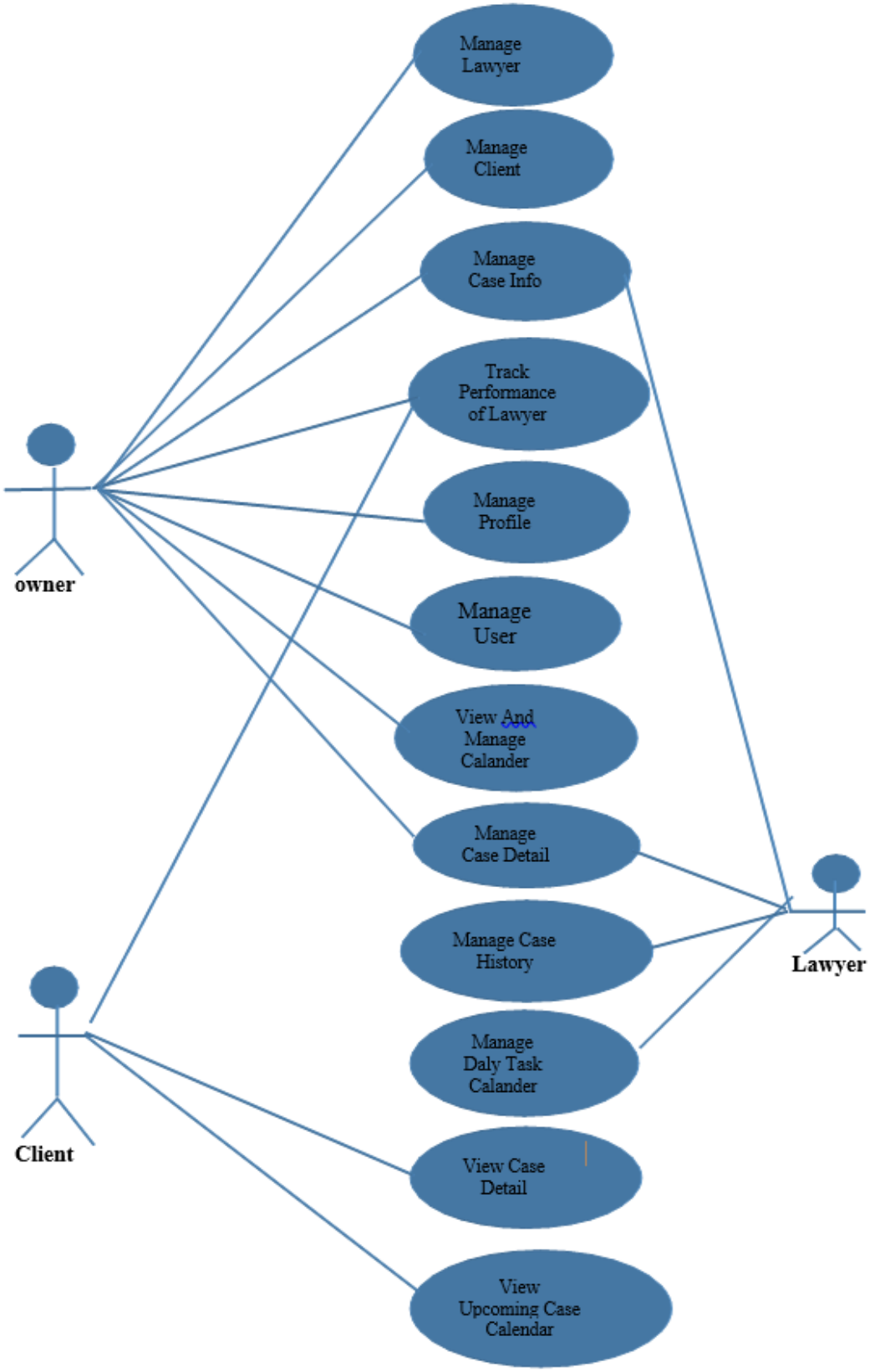


Figure2.3 overall use case

2.5 Activity diagram:

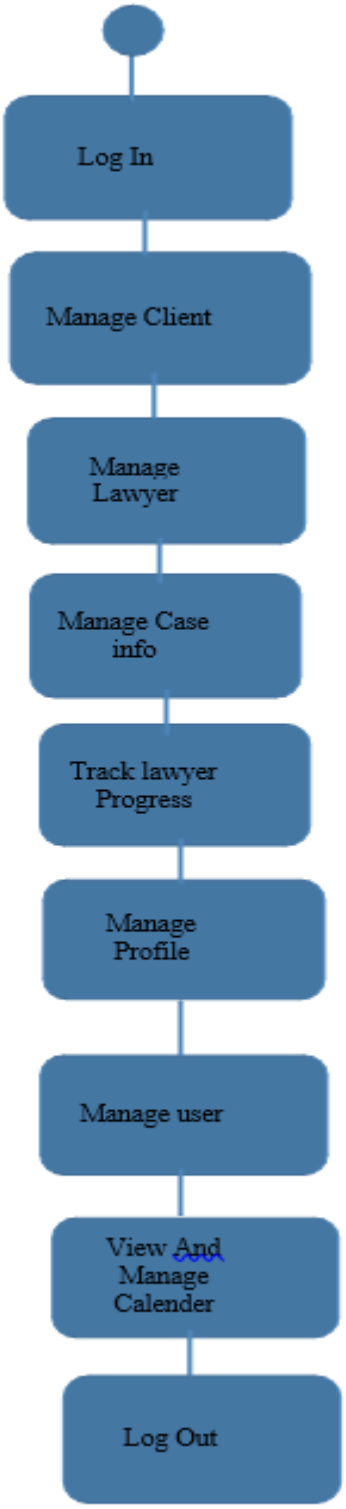


Figure2.4 owner activity diagram

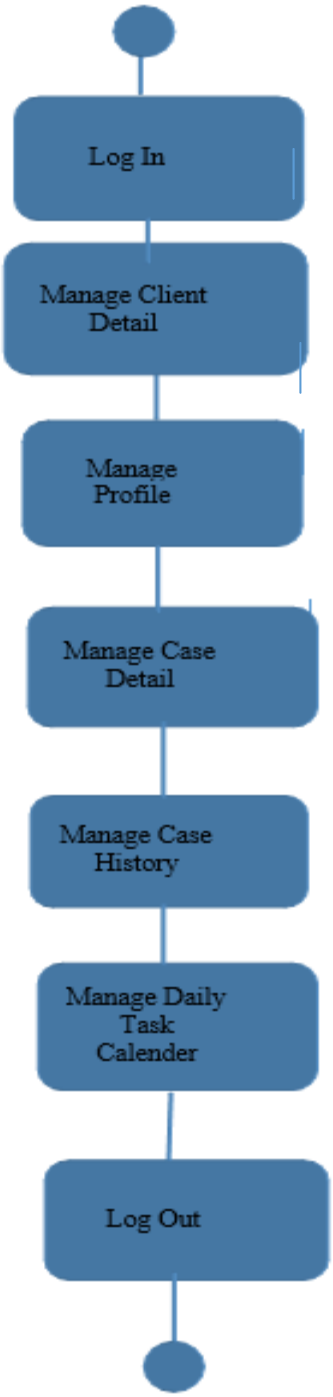


Figure 2.5 lawyer activity diagram

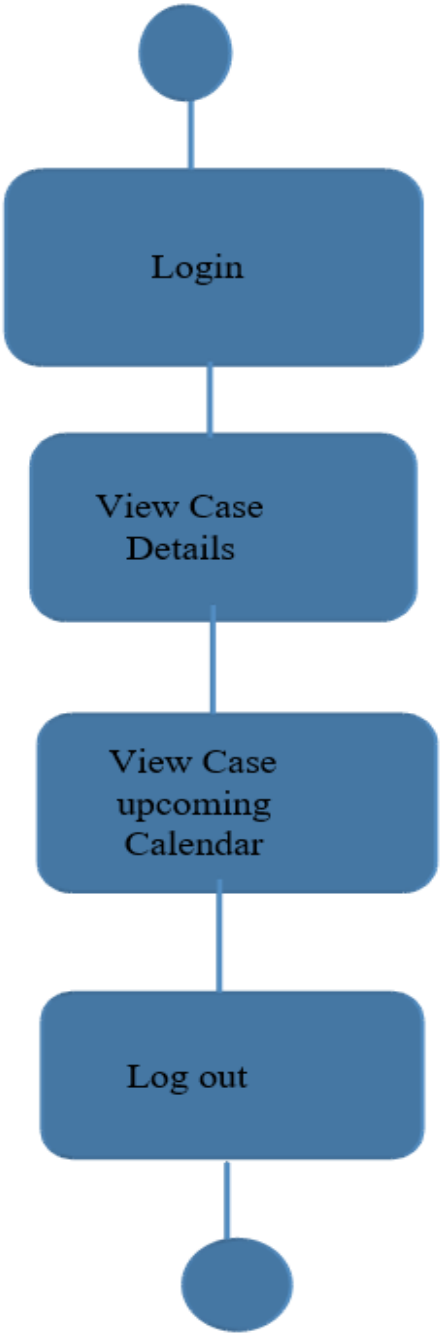


Figure2.6 client activity diagram

2.6 Sequence diagram:

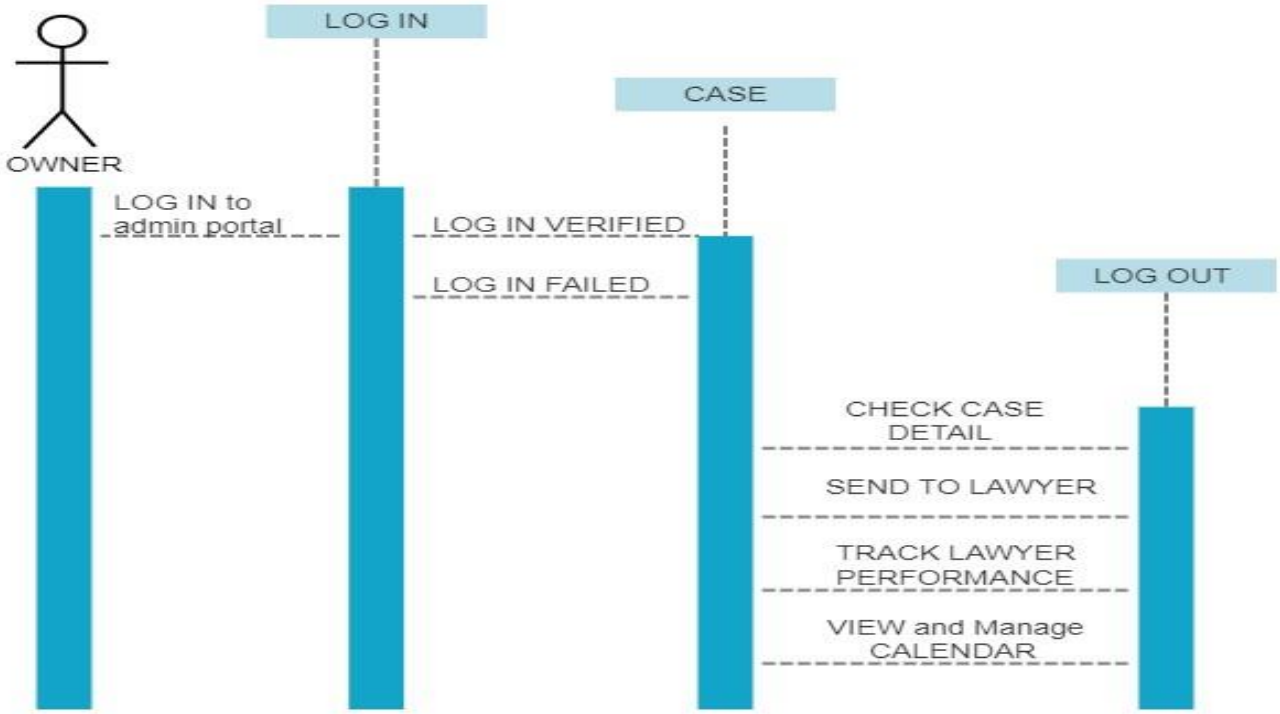


Figure 2.7 owner sequence diagram

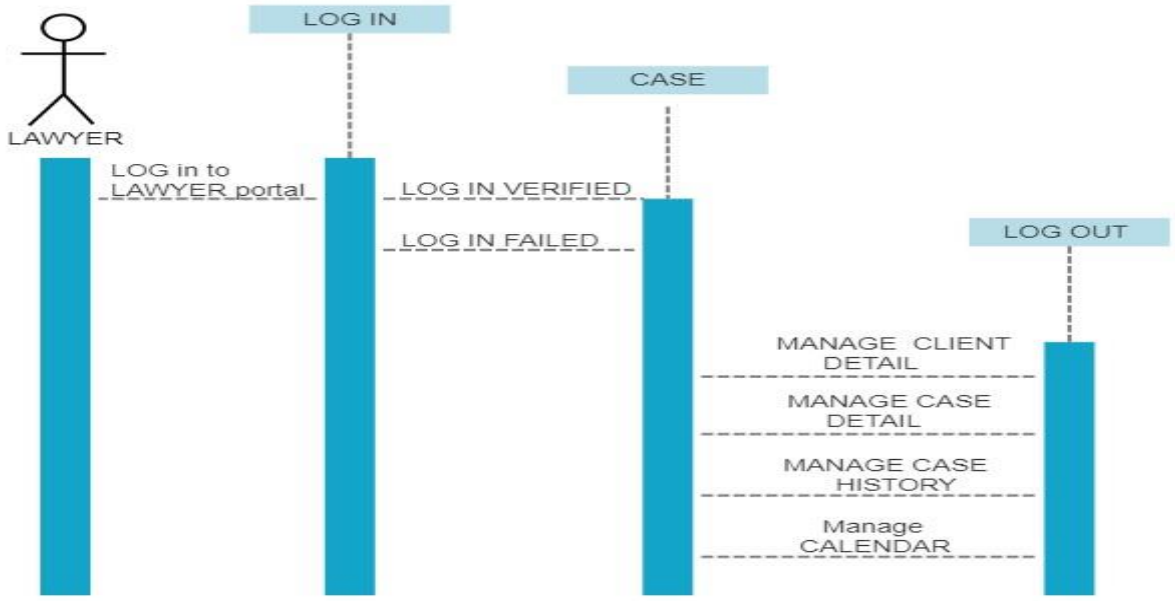


Figure 2.8 lawyer sequence diagram

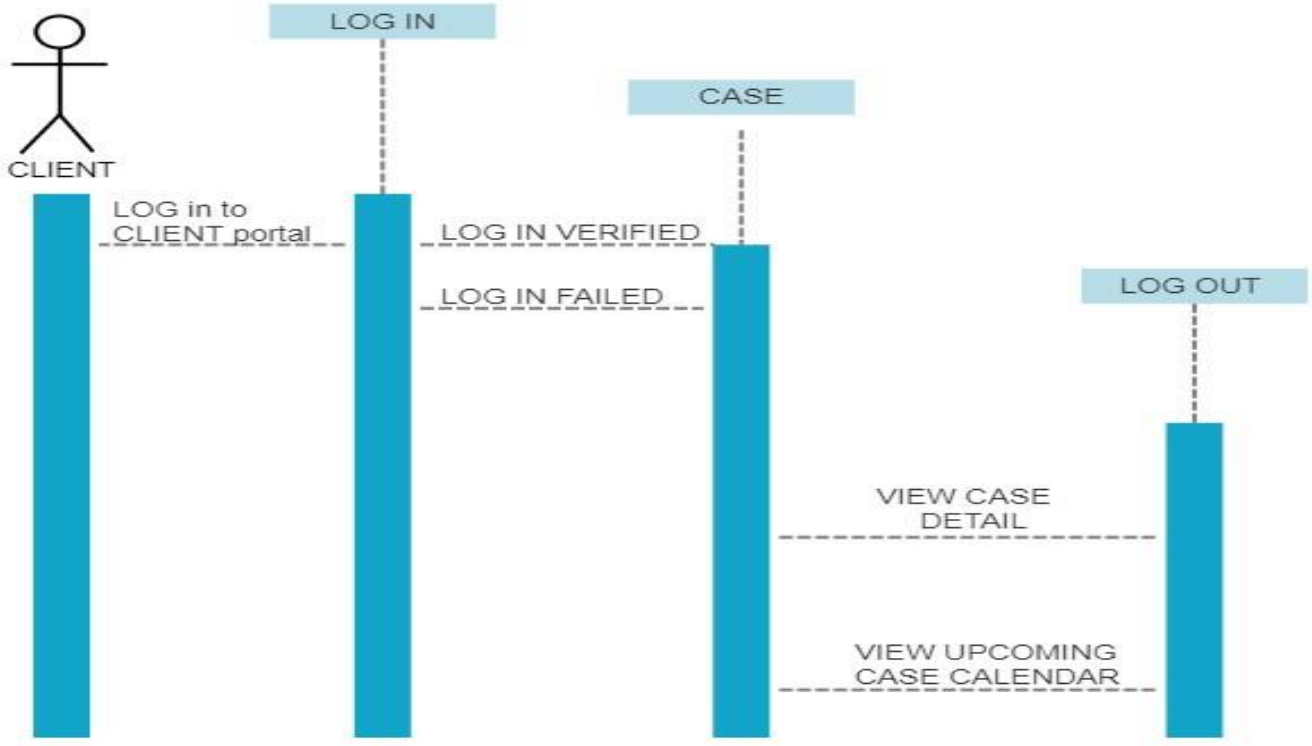


Figure 2.9 client sequence diagram

2.7 DFD (Data Flow Diagram):

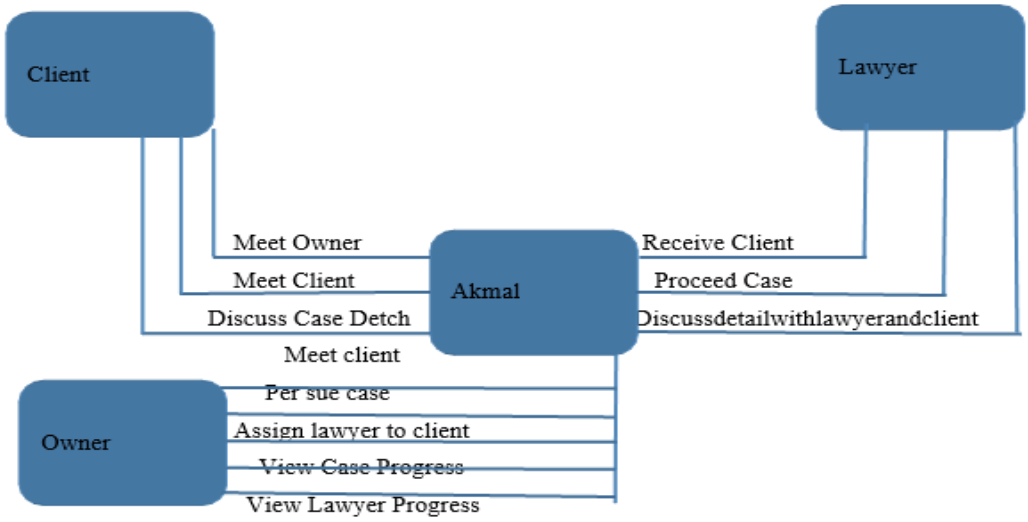


Figure 2.10 data flow diagram

2.8 ER Diagram:

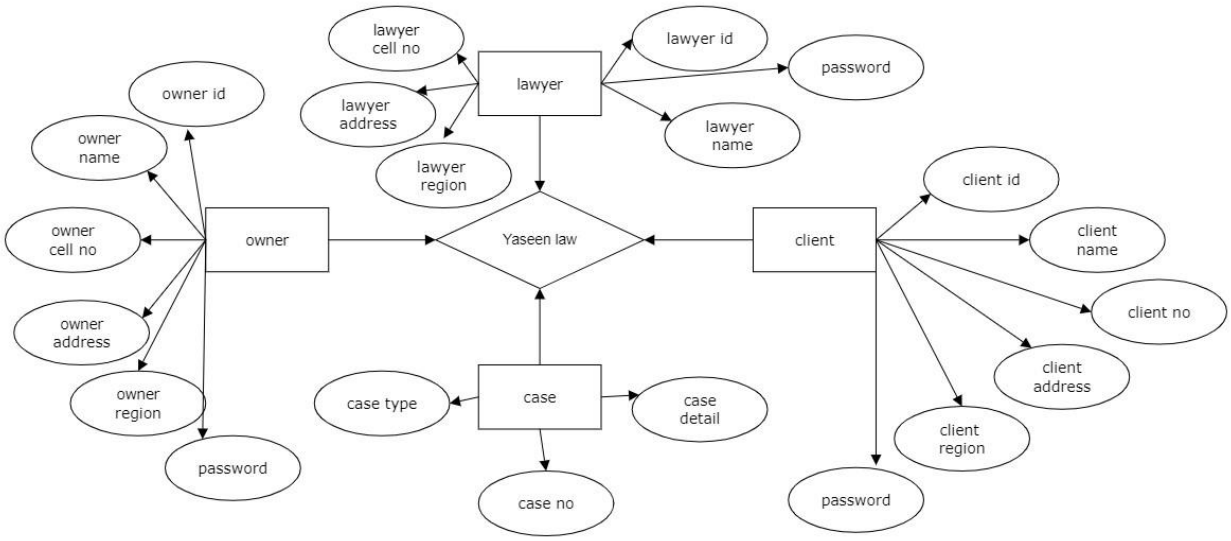


Figure 2.11 ER diagram

2.9 Schema diagram:

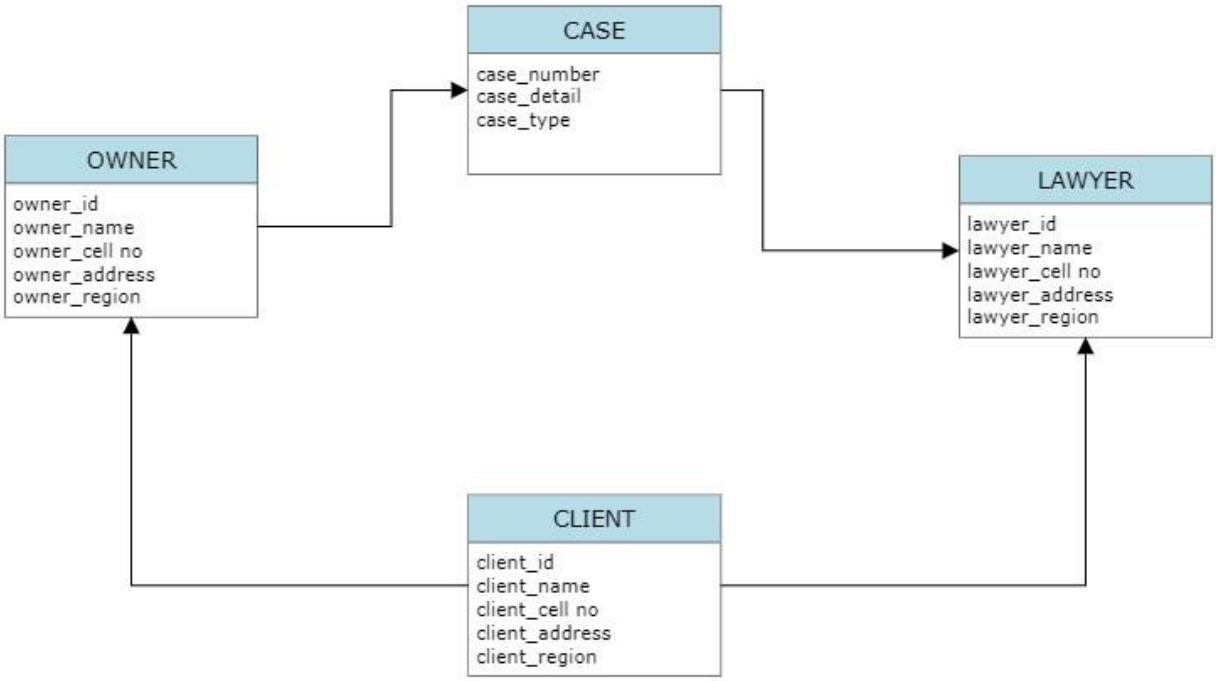


Figure 2.12 schema diagram

2.10 Nonfunctional Requirement:

1. Security: The system should have robust security measures in place to protect sensitive legal information from unauthorized access, alteration, or deletion.

2. Performance: The system should be able to handle a large volume of data and user requests without compromising its speed and responsiveness.

3. Reliability: The system should be available and accessible to users at all times, with minimal downtime or disruptions.

4. Scalability: The system should be able to scale up or down to accommodate changes in user demand or business needs.

5. Compatibility: The system should be compatible with various devices and platforms, including desktops, laptops, tablets, and smartphones.

6. Usability: The system should be user-friendly and easy to navigate, with clear and concise instructions and guidelines.

7. Accessibility: The system should be accessible to users with disabilities, complying with relevant accessibility standards and guidelines.

8. Maintainability: The system should be easy to maintain and update, with minimal downtime or disruptions during maintenance activities.

9. Data integrity: The system should ensure the accuracy, consistency, and completeness of data, with appropriate data validation and verification mechanisms.

10. Auditability: The system should maintain a detailed audit trail of user activities and system events, facilitating compliance with legal and regulatory requirements.

2.10.1 Overall constraints:

1. In this system, online data is managed.
2. This system has an owner portal.
3. This system has a client portal.
4. This system has a lawyer portal.
5. In this system, documents are managed.
6. In this system case details are managed.

Chapter 3

3. System features and requirements:

- This system manages a document.
- This system manages calendar.
- This system stores all cases performance and details.
- This system shows lawyer performance

3.1 Hardware Requirement:

This application will be store 5000 product data.

3.2 Software Requirement:

This software requires the updated version of any browser. This software requires MYSQL for database. This software requires asp.net. This software requires VS CODE and sublime text.

3.3 System requirements:

Hardware Requirements :-

Processor	**:**	Dual Core x86, 1.6 GHz or higher
RAM	**:**	4 GB or more
Hard Disk	**:**	32GB free space of hard-disk

Software Requirements :-

Database	**:**	Microsoft SQL Server
Operating System	**:**	WINDOWS 10 64 bit or above
Software	**:**	**VS CODE and sublime text**

Chapter 4

Implementation:

Implementation Report

1. Physical Design

This section outlines the physical design of the web application, focusing on the form names, their connections with the database tables or datasets, basic operations for each form, and how these forms meet the functional requirements.

Forms and Their Connections

1.1. Admin Portal

- **Form Name:** Admin Dashboard
- **Connection with Tables:** Users, Laws
- **Basic Operations:**
  - View all registered users
  - Add new law
  - Update existing law
  - Delete law
- **Functional Requirements:** Allows administrators to manage users and laws in the database.

1.2. Login Page

- **Form Name:** User Login
- **Connection with Tables:** Users
- **Basic Operations:**
  - User authentication
  - Password validation
- **Functional Requirements:** Allows users to log in to the application securely.

1.3. Registration Page

- **Form Name:** User Registration
- **Connection with Tables:** Users
- **Basic Operations:**
  - User registration
  - Data validation
  - Insertion of new user data into the Users table
- **Functional Requirements:** Enables new users to register for the application.

1.4. Law Act Finder

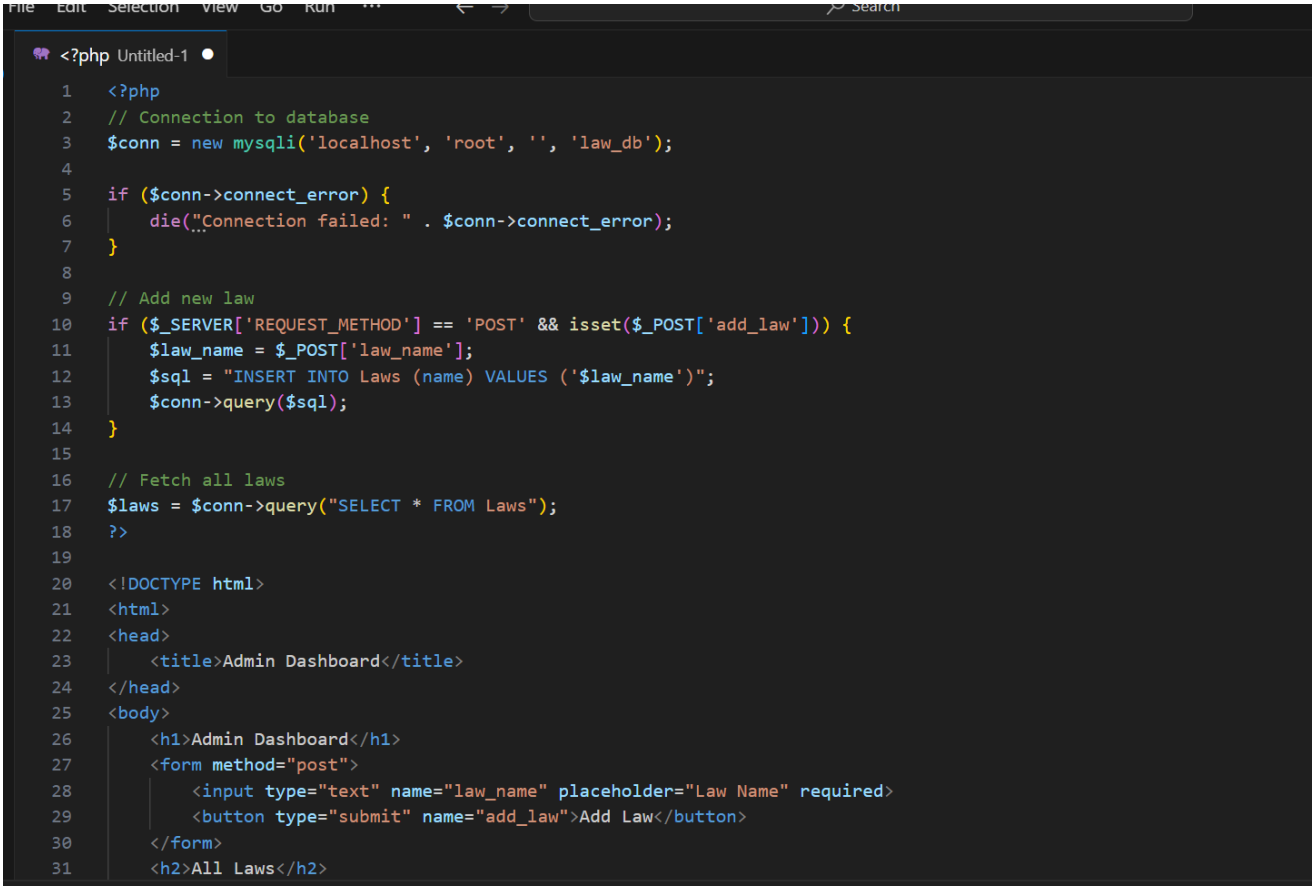
- **Form Name:** Law Search
- **Connection with Tables:** Laws, Chapters, Sections
- **Basic Operations:**
  - Display list of laws
  - Search for laws by keywords
  - Filter laws by chapters and sections
  - Display detailed view of selected law
- **Functional Requirements:** Allows users to search and view laws and their details.

2. Coding

This section provides code snippets for each form, illustrating how they interact with the front-end and the database.

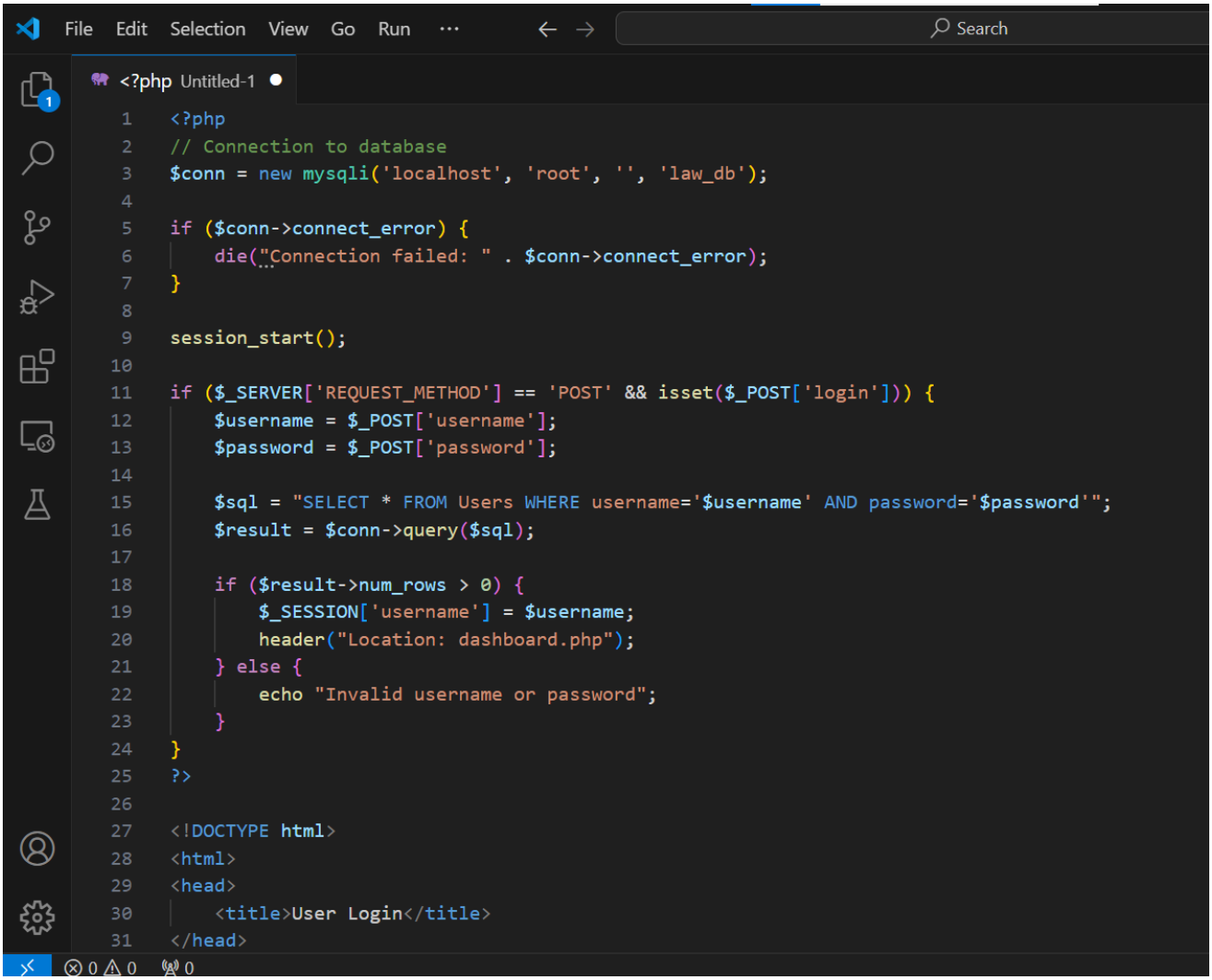
2.1. Admin Portal

Admin Dashboard Code:



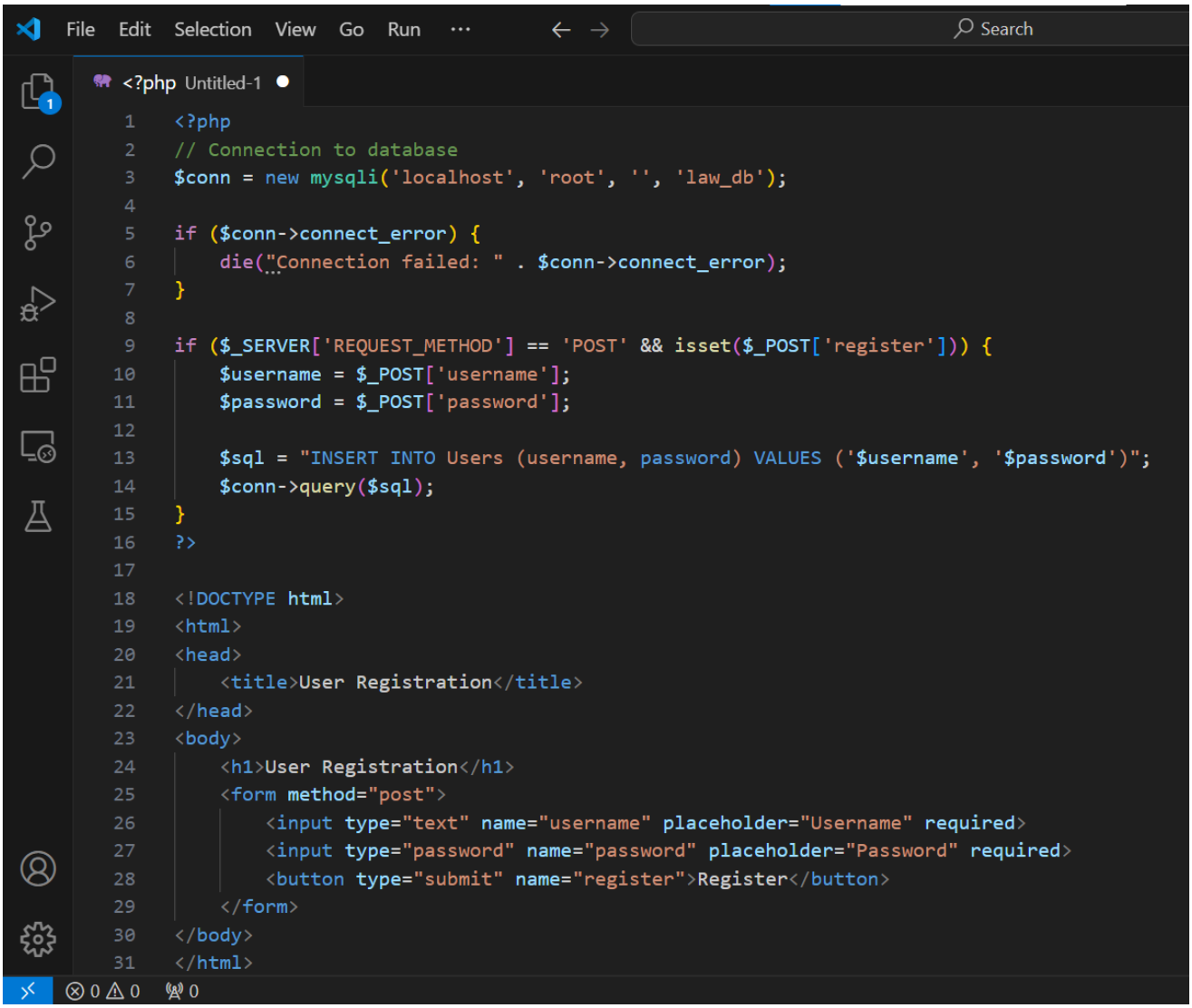
2.2. Login Page

User Login Code:



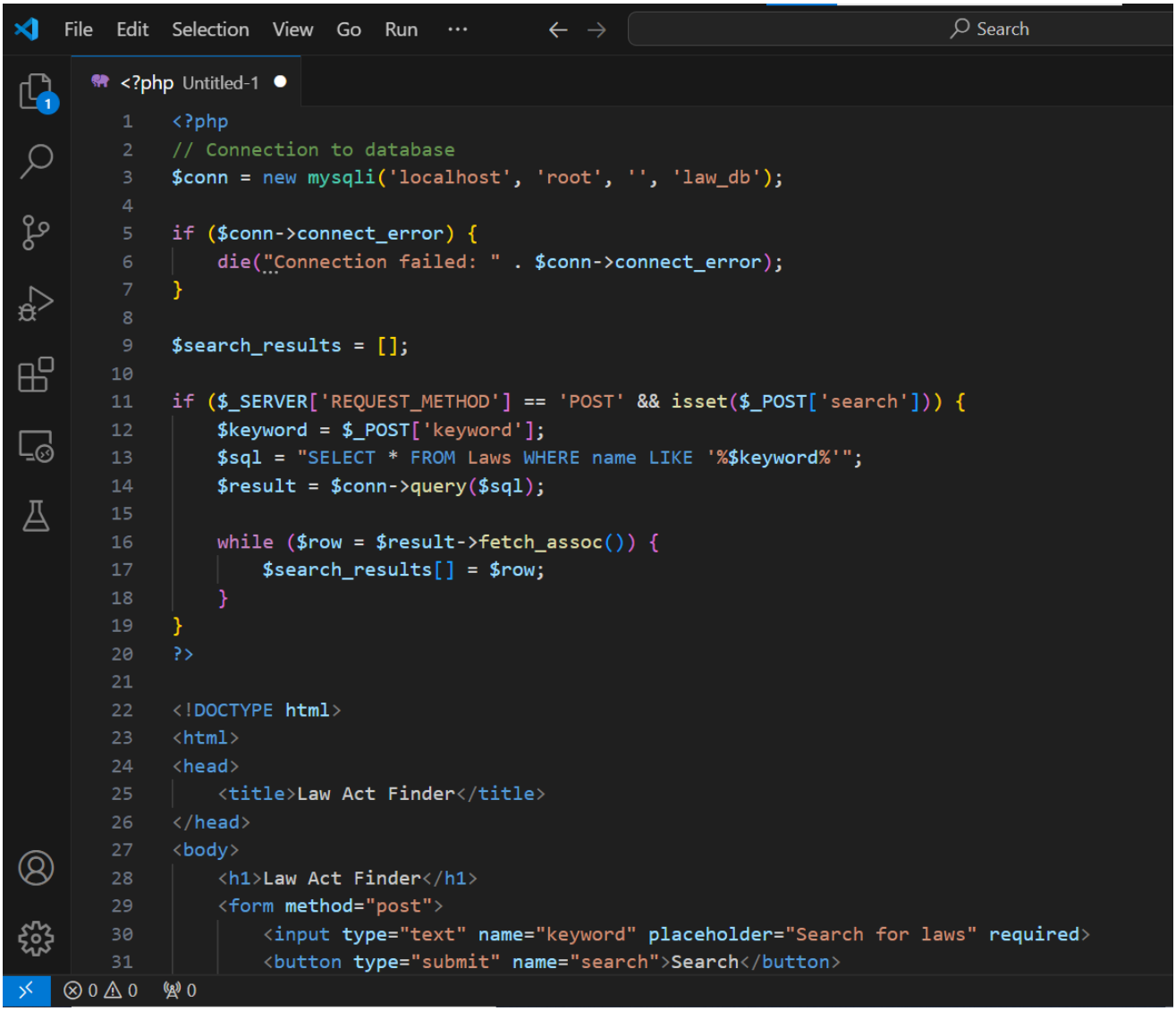
2.3. Registration Page

User Registration Code:



2.4. Law Act Finder

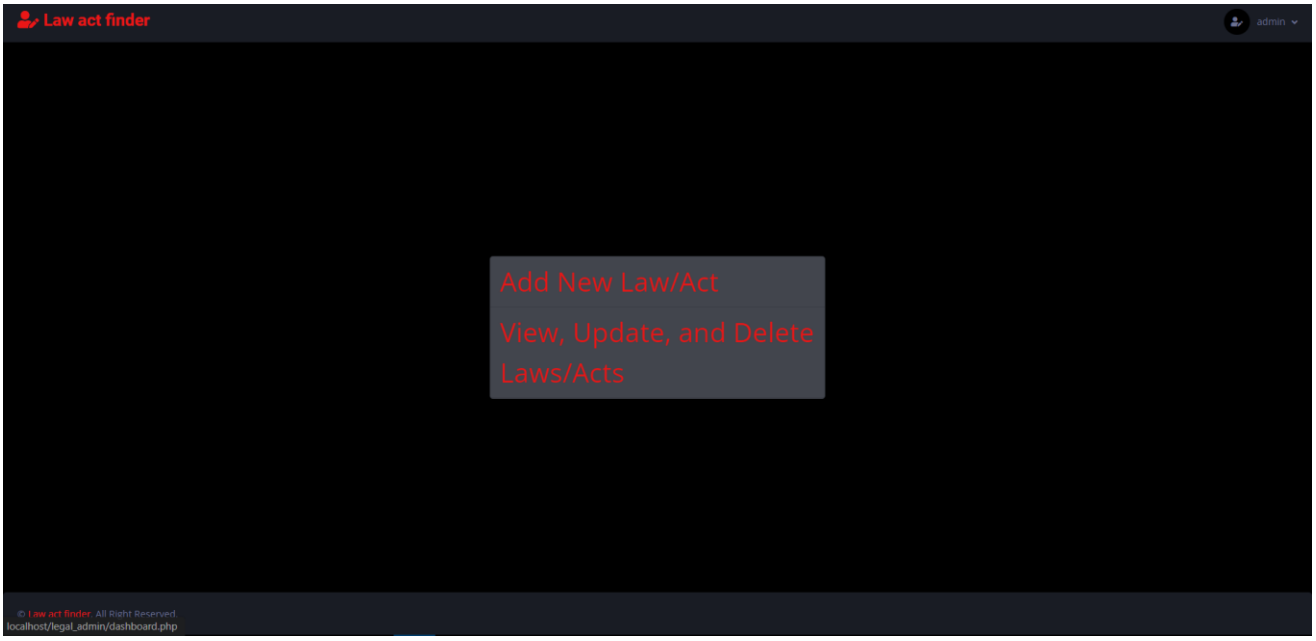
Law Search Code:

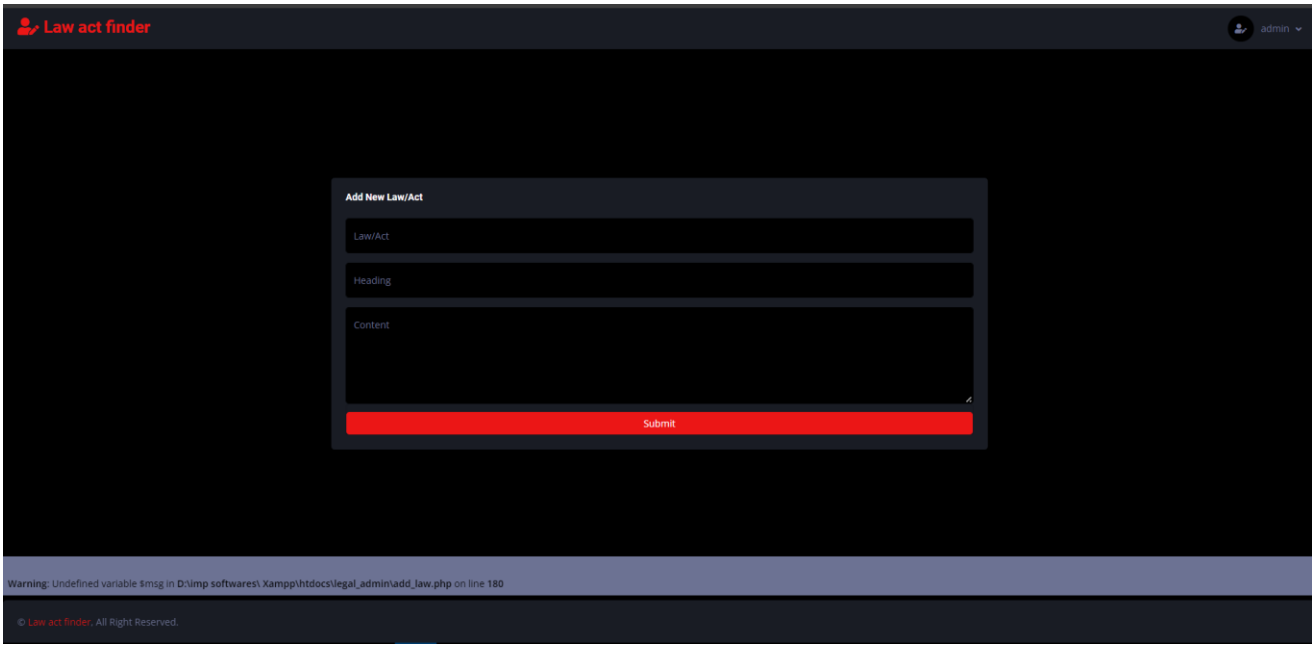


3. Front-end / GUI

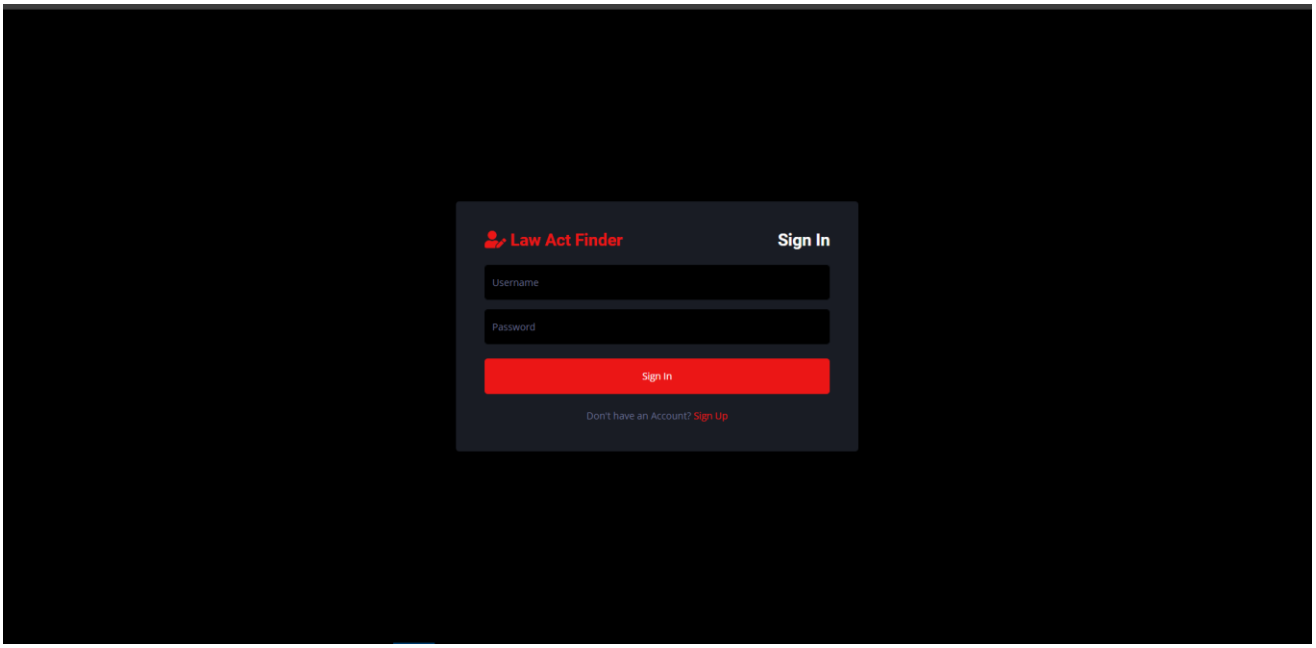
The front-end is developed using basic HTML and CSS to provide a simple and user-friendly interface.

- **Admin Dashboard:** Allows administrators to manage users and laws.

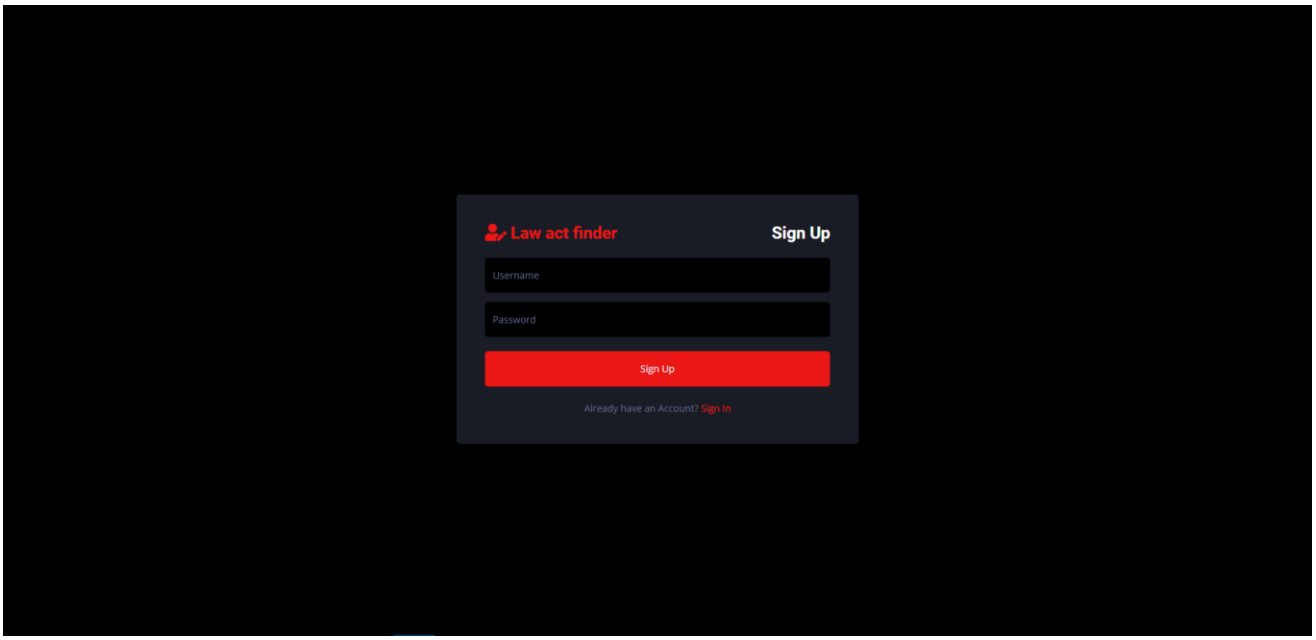




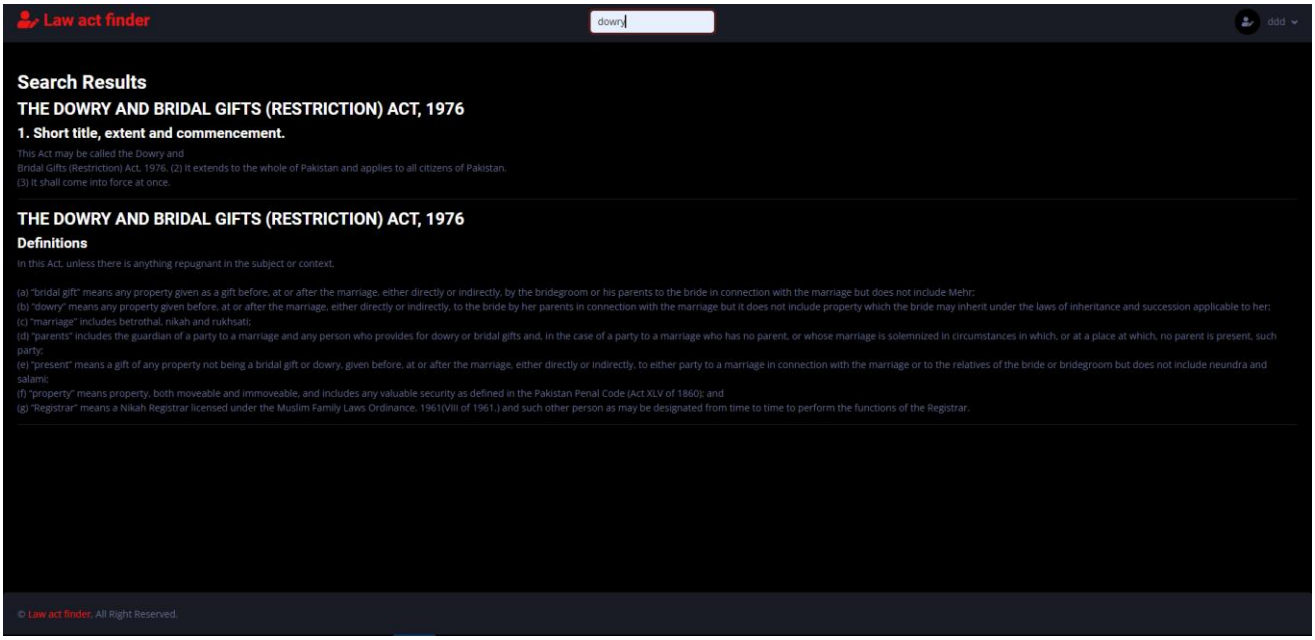
- **User Login:** Provides a secure login interface for users.



- **User Registration:** Enables new users to register.



- **Law Act Finder:** Allows users to search for laws and view their details.



Chapter 5:

Testing

The project focused on designing and implementing a LAW ACT FINDER web application using PHP, XAMPP, HTML, and Bootstrap. The application allows administrators to input laws via an admin panel and enables users to search for specific words within these laws. The output displays all occurrences of the searched word along with any associated headings and the specific law/act. During the completion of this project, efforts were made to consider all relevant principles of web application development and user experience. The following tests were conducted to ensure the system works properly.

5.1 Testing of a system

During the development of the LAW ACT FINDER, several types of testing were implemented to ensure the system's quality and reliability. Here are the key types of testing conducted during the development process:

System Testing

Type of Testing: System Testing
Description: System testing involved testing the entire system as a whole to ensure it complied with both functional and non-functional requirements. This comprehensive testing evaluated the system's overall behavior, data flow, and user interactions.

Acceptance Testing

Type of Testing: Acceptance Testing
Description: Acceptance testing was conducted by end-users to validate whether the system met their business needs and requirements. This testing phase aimed to gain confidence in the system's readiness for deployment.

Regression Testing

Type of Testing: Regression Testing
Description: Regression testing was performed to ensure that new changes or updates to the system did not introduce new defects or negatively impact existing functionalities. It involved retesting the system's core functionalities after each modification.

Performance Testing

Type of Testing: Performance Testing
Description: Performance testing evaluated the integrated system's responsiveness, scalability, and stability under various workloads and stress conditions. This testing helped identify potential performance bottlenecks and areas for optimization.

By implementing these various types of testing throughout the development process, the system's quality, functionality, and user experience were thoroughly evaluated. Test results were used to identify and rectify defects, ensuring a reliable, robust, and user-friendly system.

5.2 Test cases

Below are some detailed test cases for the proposed LAW ACT FINDER system.

Test Case 1: Search for a Specific Word
Input: Search for the word "contract"
Expected Output: All text containing the word "contract" is displayed, along with associated headings and the specific law/act.

Test Case 2: Search for a Common Term
Input: Search for the word "agreement"
Expected Output: All text containing the word "agreement" is displayed, along with associated headings and the specific law/act.

Test Case 3: Admin Panel Input
Input: Input a new law titled "Contract Law" with the content "All contracts must be in writing."
Expected Output: The new law is successfully added and can be searched by users.

Test Case 4: Search for a Rare Term
Input: Search for the word "arbitration"
Expected Output: All text containing the word "arbitration" is displayed, along with associated headings and the specific law/act.

Test Case 5: Search for a Word with Multiple Occurrences
Input: Search for the word "dispute"
Expected Output: All occurrences of the word "dispute" are displayed, along with associated headings and the specific laws/acts.

Test Case 6: Performance Under Load
Input: Perform multiple searches simultaneously for words like "contract," "agreement," "arbitration," and "dispute."
Expected Output: The system responds promptly without significant delay, displaying the correct results for each search query.

By conducting these test cases, the functionality and performance of the LAW ACT FINDER web application were thoroughly validated, ensuring a robust and user-friendly experience for both administrators and end-users.

5.2 Test cases tables

Table 5.1: Test Case 1 LAW ACT FINDER

Test Case Attribute	Description
Test Case ID	1
Test Item	LAW ACT FINDER
Test Type	Manual
Test Case Name	Search for "contract"
Input	contract
Test Case Description Data	User searches for the word "contract"
Operational Procedure	Single Click
Pre-Condition	Homepage
Post-Condition	Search Results Page
Expected Output	All occurrences of "contract" with associated headings and laws/acts are shown
Actual Output	All occurrences of "contract" with associated headings and laws/acts are successfully

	shown

Table 5.2: Test Case 2 LAW ACT FINDER

Test Case Attribute	Description
Test Case ID	2
Test Item	LAW ACT FINDER
Test Type	Manual
Test Case Name	Search for "agreement"
Input	agreement
Test Case Description Data	User searches for the word "agreement"
Operational Procedure	Single Click
Pre-Condition	Homepage
Post-Condition	Search Results Page
Expected Output	All occurrences of "agreement" with associated headings and laws/acts are shown
Actual Output	All occurrences of "agreement" with associated headings and laws/acts are successfully shown

Table 5.3: Test Case 3 LAW ACT FINDER (Formatted for Readability)

Test Case Attribute	Description
Test Case ID	3
Test Item	LAW ACT FINDER
Test Type	Manual
Test Case Name	Admin Input New Law
Input	New Law
Test Case Description Data	Admin inputs a new law titled "Contract Law"
Operational Procedure	Form Submission
Pre-Condition	Admin Panel
Post-Condition	Law Added Confirmation
Expected Output	New law is successfully added and searchable
Actual Output	New law is successfully added and can be searched by users

Table 5.4: Test Case 4 LAW ACT FINDER (Formatted for Readability)

Test Case Attribute	Description
Test Case ID	4
Test Item	LAW ACT FINDER
Test Type	Manual
Test Case Name	Search for "arbitration"
Input	arbitration
Test Case Description Data	User searches for the word "arbitration"
Operational Procedure	Single Click
Pre-Condition	Homepage
Post-Condition	Search Results Page
Expected Output	All occurrences of "arbitration" with associated headings and laws/acts are shown
Actual Output	All occurrences of "arbitration" with associated headings and laws/acts are successfully shown

Table 5.5: Test Case 5 LAW ACT FINDER (Formatted for Readability)

Test Case Attribute	Description
Test Case ID	5
Test Item	LAW ACT FINDER
Test Type	Manual
Test Case Name	Search for "dispute"
Input	dispute
Test Case Description Data	User searches for the word "dispute"
Operational Procedure	Single Click
Pre-Condition	Homepage
Post-Condition	Search Results Page
Expected Output	All occurrences of "dispute" with associated headings and laws/acts are shown
Actual Output	All occurrences of "dispute" with associated headings and laws/acts are successfully shown

5.3 Conclusion / Future Work

Conclusion

In conclusion, the LAW ACT FINDER project successfully utilizes PHP, XAMPP, HTML, and Bootstrap to create a user-friendly web application for searching and managing legal texts. The system allows administrators to input laws through an admin panel, and users can search for specific words within these laws. The application displays all occurrences of the searched word along with any associated headings and the specific law/act. This project demonstrates the importance of efficient data handling and user-centric design in legal information systems. By providing a robust search functionality and a simple admin interface, the LAW ACT FINDER aids in legal research and information retrieval.

Future Work

Future work for this project involves integrating advanced search capabilities, such as natural language processing (NLP) to understand the context of search queries better. Additionally, implementing AI and machine learning algorithms can enhance the system by providing predictive analytics and automated suggestions based on user search patterns. Another potential improvement is to expand the system to support multiple languages, allowing users to search for legal texts in various languages. These enhancements aim to make the LAW ACT FINDER more intelligent, versatile, and accessible to a broader audience, thereby improving its usefulness and efficiency in legal research and information management.