ONLINE TRAFFIC LAW CONSULTATION SYSTEM

DHEEPAK B 23MX203

MARIAN A 23MX215

23MX18 WEB APPLICATION DEVELOPMENT

REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENT FOR THE DEGREE OF

MASTER OF COMPUTER APPLICATIONS

ANNA UNIVERSITY

DECEMBER 2023

DEPARTMENT OF COMPUTER APPLICATIONS

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE - 641 004

ONLINE TRAFFIC LAW CONSULTATION SYSTEM

Bonafide record of work done by

DHEEPAK B 23MX203

MARIAN A 23MX215

23MX18 WEB APPLICATION DEVELOPMENT

Report submitted in partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER APPLICATIONS

ANNA UNIVERSITY

DECEMBER 2023

Faculty Guide

……

Mrs. A.Kalyani

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We immensely take this opportunity to express our sincere gratitude to **Dr. K. Prakasan**, Principal, PSG College of Technology, for providing us all the facilities within the campus for the completion of the project.

We profoundly thank **Dr. A. Chitra**, Professor and Head, Department of Computer Applications, PSG College of Technology, for her moral support and guidance.

We owe an extremely unbound gratitude and extend our thanks to our Programme Coordinator, **Dr.R .Manavalan**, Associate Professor, Department of Computer Applications, PSG College of Technology, whose motivation and support encouraged us in taking up and completing this project work.

We are overwhelmed in all humbleness and gratefulness in acknowledging our subject Coordinator and Project Guide **Mrs. A. Kalyani**, Assistant Professor, Department of Computer Applications, PSG College of Technology, for her priceless suggestions and unrelenting support in all our efforts to improve our project and for piloting the right way for the successful completion of our project.

We also express our sincere thanks to all the staff members of the Department of Computer Applications for their encouragement. We also thank our parents and all the hands that helped us.

SYNOPSIS

The Online Traffic Law Consultation System is a web-based platform that aims to provide a convenient and efficient way for individuals to seek legal advice and consultation regarding traffic law matters. It is an innovative digital platform aimed at revolutionizing the way individuals' access and receive legal advice pertaining to traffic laws and regulations. In an era marked by rapid urbanization and increasing vehicular traffic, understanding and adhering to traffic laws are paramount to ensuring road safety and compliance with legal requirements by providing a user- friendly interface and convenient features, it offers an efficient alternative to traditional methods of accessing legal services. The existing system is a paper-based system that involves a lot of manual work that is prone to careless handling of data. The existing system also provides Sharing sensitive legal information online can raise security and privacy concerns. Drawbacks of the above-mentioned existing system can be resolved by the proposed system which automates the existing manual system by a web application that allows users to operate through dedicated websites or mobile applications on these platforms serve as the primary interface for users to access legal services.

CHAPTER 1

INTRODUCTION

Online Traffic Law Consultation System – user go-to destination for simplified access to traffic laws. Here, users can effortlessly search and explore a comprehensive database of traffic regulations, making legal information easily accessible. Say goodbye to confusion and uncertainty – with just a few clicks, discover the answers you need to navigate the roads with confidence. Welcome to a user-friendly platform where traffic laws are just a search away.

1.1 Project Overview

Online Traffic Law Consultation System, where users and administrators can connect seamlessly to change the way traffic law administration is done. This cutting-edge tool is an accessible hub that makes it simple for users to search, find, and keep updated about traffic rules. On the other side, administrators have strong tools at their disposal to add, amend, and remove laws instantly, guaranteeing that the system is a trustworthy source of up-to-date legal knowledge.

This technology provides an easy-to-use dashboard for administrators to handle laws with ease. Simplifying the process of adding, amending, or removing laws enables quick adjustments to the always shifting legal environment. Users will always have access to the most recent information thanks to real-time updates, and thorough analytics offer insights into user behaviour.

1.2 Tools and Technology

HTML5 (HyperText Markup Language 5)

HTML is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet. It is the fifth revision of the HTML standard. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices.

Many reasons could make use of HTML, and here are the most five reasons to mention:

- **It is the future**

With all the big players embracing HTML already, it looks a safe bet that it will be the future. How it will exactly pan out, no one quite knows (or when), but at some point, it will land as a standard and is the next logical step.

- **It can help to correctly index the content in search engines**

This may not be the case now, certain tags will help such as Microformats, but Google search engines are getting smarter and smarter.

- **It can help accessibility**

As with Web Standards, just by producing a site that uses CSS for presentation, it does not mean it is accessible. HTML is the same story, just by coding in HTML doesn't mean the site is more accessible. Instead, it is another arrow in the bow to help make the site become more accessible.

- **It has bundles of new features**

HTMLdoes not just allow users to mark-up documents with meaningful semantic tags, it also offers new functionality. These are not all in HTMLspec per se but part of the new movement towards funky new web applications, hence why they are mentioned. From watching videos without having to have a plug in, native form features, the 39 canvas, application caches so information can be stored offline to geolocation. Whilst these are in no way the finished article, there is no harm being ready to embrace them

CSS3 (Cascading Style Sheets 3):

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVVG, MathML, XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. HTML largely determines textual content, whereas CSS determines visual structure, layout, and aesthetics.

There are number of benefits of CSS, including:

- **Pages load faster**

CSS does not require the writing of HTML tag attributes every time. There is the writing of rule

just once for a tag, which can be applied to all the occurrences of the corresponding tag. So using CSS, there is less code, which means faster downloading.

- **Easier website maintenance**

CSS makes the maintenance of the website easier. If users require a global change in the file, it can be simply done by changing the style by which all the elements on the web page will update automatically. The CSS file provides a flexible look to the website, which can be altered in a convenient way. It also makes HTML formatting and modification of corresponding data elements easier.

- **Saves a lot of time**

CSS style definitions are saved in external CSS files, so it is possible to change the entire website by changing just one file. Write CSS once and then reuse the same sheet in multiple HTML pages. Then define a style for each HTML element and apply it to as many web pages as users want.

- **Platform independence**

The Script offers consistent platform independence and can support the latest browsers as well.

- **Multiple device compatibility**

CSS is compatible with the older language versions so that users can use CSS with the earlier language versions. Because of this, if the CSS application is developed with the older programming language versions and if the developer combines the same with new improvements, then CSS can be easily implemented with the corresponding changes so the developer can update the existing code successfully. CSS allows the content to be optimized for more than one type of device.

Javascript:ECMAScript 2021 (ES12)

JavaScript serves as the dynamic force behind Online Traffic law Consultation System, injecting vitality into the platform with its capabilities as a client-side scripting language. As the backbone of real-time interactivity, JavaScript breathes life into dynamic dashboards for learners, fostering seamless and responsive user interactions. One of its pivotal roles is in form validation, where it ensures accuracy and reduces errors by validating user inputs in real time. Beyond this, JavaScript showcases its versatility by contributing to personalized user experiences. The multifaceted nature of JavaScript, encompassing real-time functionality, validation prowess, and personalization features, establishes it as an indispensable element in shaping an

interactive and learner-focused experience on Online Traffic law Consultation System.

\Features of JavaScript in Online Traffic law Consultation System

- **Real-time Interactivity**

  JavaScript enables real-time updates and interactions, creating a dynamic and responsive user interface.

- **Seamless User Interactions**

  JavaScript ensures smooth and seamless user interactions, enhancing the overall learning experience on Online Traffic law Consultation System.

- **Form Validation**

  Accurate form validation is achieved through JavaScript, reducing errors and improving data integrity by validating user inputs in real time.

- **Versatility**

  JavaScript's versatility is showcased in its ability to handle a variety of tasks, from real-time updates to personalized user experiences.

  This strategic use of JavaScript in Online Traffic law Consultation System contributes to a dynamic, engaging, and user-centric platform, ensuring a seamless and responsive experience. The below Fig1.1 shows the diagrammatic representation of technologies used.

Fig. 1.1 Technology structure

MongoDB (Version-7.0.4):

MongoDB is a popular NoSQL database management system that uses a document-oriented data model. It is designed to store, query, and process large amounts of data in a flexible, schema-free format.

MongoDB is classified as a NoSQL database because it doesn't rely on the traditional relational database structure.The features of MongoDB are:

- **Document-Oriented:** Data is stored in BSON (Binary JSON) documents, which are similar to JSON objects.
- **Scalability**: MongoDB is horizontally scalable, meaning users can add more servers to handle increased load.
- **Flexible Schema:** Unlike traditional relational databases, MongoDB doesn't require a predefined schema. Fields can vary in each document.
- **Indexing:** Supports various types of indexes to improve query performance.
- **Aggregation Framework:** Provides powerful and flexible aggregation capabilities for data processing.

NodeJS (Version: 18.12.0):

Node.js is a runtime environment that allows users to execute JavaScript code outside of a web browser. It is built on the V8 JavaScript runtime and enables server-side development using JavaScript. Node.js uses an event-driven, non-blocking I/O model, making it efficient for building scalable network applications.The features of Node.js are:

- **Asynchronous I/O**: Node.js is designed to handle asynchronous operations, allowing it to efficiently manage many connections simultaneously.
- **Single-Threaded:** Although the core of Node.js is single-threaded, it uses an event loop and non-blocking I/O to handle concurrent requests.
- **npm (Node Package Manager):** npm is a package manager for Node.js that simplifies the process of installing, sharing, and managing third-party libraries.
- **Cross-Platform:** Node.js is cross-platform and can run on various operating systems, including Windows, macOS, and Linux.

EJS:

EJS (Embedded JavaScript) is an integral component in web development, seamlessly merging JavaScript with HTML to enhance the dynamism and interactivity of web applications. Operating as a templating engine, EJS enables the embedding of JavaScript code directly into HTML templates, facilitating the generation of dynamic content on the server side. Through special tags like <%= %>, EJS empowers developers to inject server-side data into HTML,

fostering a more personalized and engaging user experience. This templating engine supports JavaScript control flow structures, allowing the integration of loops and conditionals within

HTML templates. Additionally, EJS facilitates the passage of variables from the server to the template, enabling the rendering of dynamic content. The concept of partial views further enhances code organization and maintainability by promoting the creation and reuse of modular

components across multiple pages. In summary, EJS plays a pivotal role in the development of flexible, data-driven web applications by seamlessly combining JavaScript with HTML and facilitating dynamic content generation.

The MVC framework includes the following 3 components:

1.Controller

2.Model

3.View

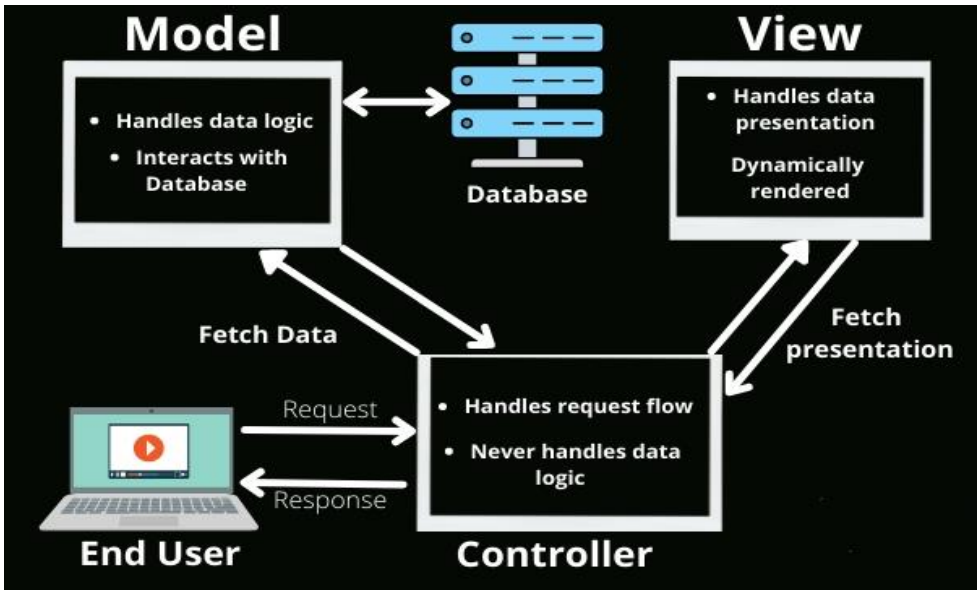


Fig. 1.2 MVC ARCHITECTURE

MVC Architecture Design

Controller:

The controller is the component that enables the interconnection between the views and the model so it acts as an intermediary. The controller doesn't have to worry about handling data logic, it just tells the model what to do. It processes all the business logic and incoming requests, manipulates data using the Model component, and interact with the View to render the final output.

View:

The View component is used for all the UI logic of the application. It generates a user interface for the user. Views are created by the data which is collected by the model component but these data aren't taken directly but through the controller. It only interacts with the controller.

Model:

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. It can add or retrieve data from the database. It responds to the controller's request because the controller can't interact with the database by itself. The model interacts with the database and gives the required data back to the controller.

Working of the MVC framework with an example:

Let's imagine an end-user sends a request to a server to get a list of students studying in a class. The server would then send that request to that particular controller that handles students. That controller would then request the model that handles students to return a list of all students studying in a class.

CHAPTER 2

SYSTEM ANALYSIS

This chapter discusses the analysis of the system design. Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces, and data for a system based on the specified requirements.

2.1 Existing System

There is no particular existing system for Online Traffic Law Consultation system. There is an existing system for overall laws. In that system the data entry is manual and maintaining the records of the laws is very difficult when it comes to storing the records physically. There may be a chance the data might get lost. And when a user requests a particular law one has to manually check over the books. Users get irrelevant and unorganised data when they search through online.

2.2 Proposed System

In this system we are making a particular section covering only Traffic related laws. Here we have an Admin panel, where the admin can add, update and delete laws. The user may search for a particular keyword or type the text to find the related laws in our website. Here we provide security for the user's data like user details. Then if the user wants to file a complaint and doesn't know where to file it we are making it simple through our website. We collect feedback and ratings from the users for constantly updating our services thus solving the user's problems more efficiently, user friendly and legally is our motto.

2.3 Functional Requirements

Sign In Module:

The Sign-In functional module is designed to manage user authentication and access control within the Online Traffic law system. Its primary purpose is to verify the identity of users and grant them appropriate access based on their credentials. The module ensures that only authorized users can log in to the application, and they Sign-In to the system.

Admin Module:

The admin module of our Online Traffic Law Consultation System empowers admin with effortless control. The admin can seamlessly add, update, and delete traffic laws in real-time. This feature ensures the platform stays current, adapting to the evolving legal landscape promptly. The user-friendly interface guarantees efficient law management, making it a powerful tool for administrators to maintain an up-to-date and reliable source of information. Welcome to a streamlined approach to traffic law administration.

Search Module:

Our Search Module offers users flexibility with both keyword and full-text searches. Simply type a keyword to quickly retrieve specific laws or enter entire text for a comprehensive search experience. This user-friendly functionality ensures efficient access to relevant traffic laws, tailored to individual Needs. Welcome to a hassle-free way of navigating the traffic regulations.

Complaint and Feedback Module:

Our Complaint Register and Feedback Module allow users to easily register complaints through our website. User feedback is the driving force behind continuous improvements, ensuring our website Evolves based on real user experiences. Enhancing our platform through valuable feedback.

2.4 Non-Functional Requirements

Performance:

This Online Traffic law consultation system responds quickly to user interactions, loads laws efficiently, and provides smooth navigation. This application is optimized to handle concurrent users without significant slowdowns or crashes for any number of users.

Security:

Robust security measures are necessary to protect user data, prevent unauthorized access,and secure communication between the app and the server. This Online traffic law consultation system ensures the data security provided by the users.

Usability:

This app contains an intuitive and user-friendly interface. Clear instructions,well-organized content, and easy-to-navigate menus contribute to a positive user experience,which was provided by this Online Traffic Law Consultation

Compatibility:

This functions smoothly on various devices, operating systems and Screen sizes.

CHAPTER 3

SYSTEM DESIGN

This Chapter discusses system design. System design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements.

The design activities are of main importance in this phase, because in these activities decisions ultimately affect the success of the software implementation and its ease of maintenance is made. This decision has the final bearing on the reliability and maintainability of a system. Design is the only way to accurately transfer the requirements into finished software or systems that address the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

3.1 Use Case Diagram

A use case diagram is a visual representation in software engineering that depicts the interactions between users (actors) and a system to illustrate the functionality and behaviour of the system from a user's perspective. It is a powerful tool for capturing and documenting the requirements of a system and understanding the various use cases or scenarios in which the system will be utilized.

Actors represent the external entities interacting with the system, such as users, external systems, or devices. Use cases represent the specific functionalities or tasks that the system provides to the actors. The relationships between actors and use cases show the associations and interactions between them. Use case diagrams help to identify system requirements, define
system boundaries, and clarify the expected behavior of the system. Figure 3.1 shows the use case diagram of Online Traffic Law Consultation system.

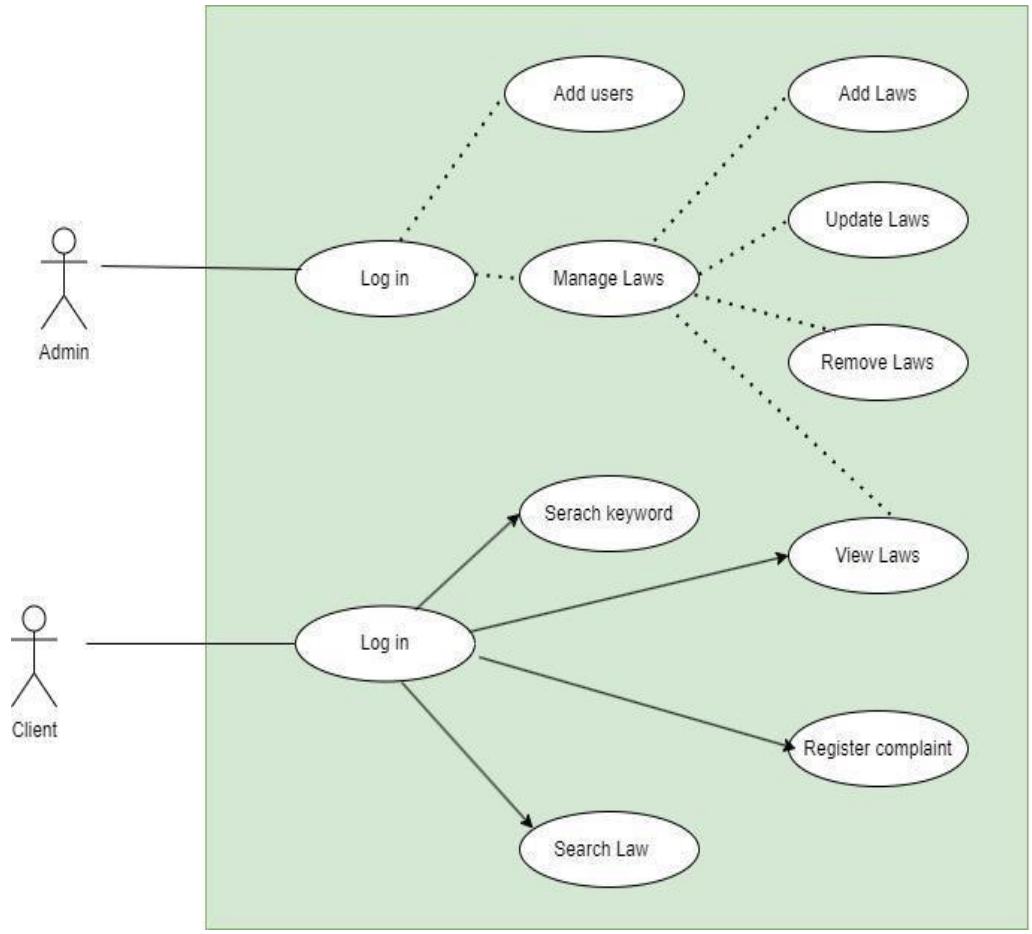


Fig 3.1 Use Case Diagram of Online Traffic Law Consultation System

3.2 System Flow Diagram

A system flow diagram, also known as a flowchart, is a graphical representation of the logical steps or flow of a process or algorithm in software engineering. It is a visual tool used to understand, analyze , and communicate the sequence of activities or decisions involved in a software system. Flow diagrams help in visualizing the overall structure and logic of a system, making it easier to identify bottlenecks, potential issues, and areas for optimization. Figure 3.2 shows the system flow of the Online Traffic Law Consultation System.

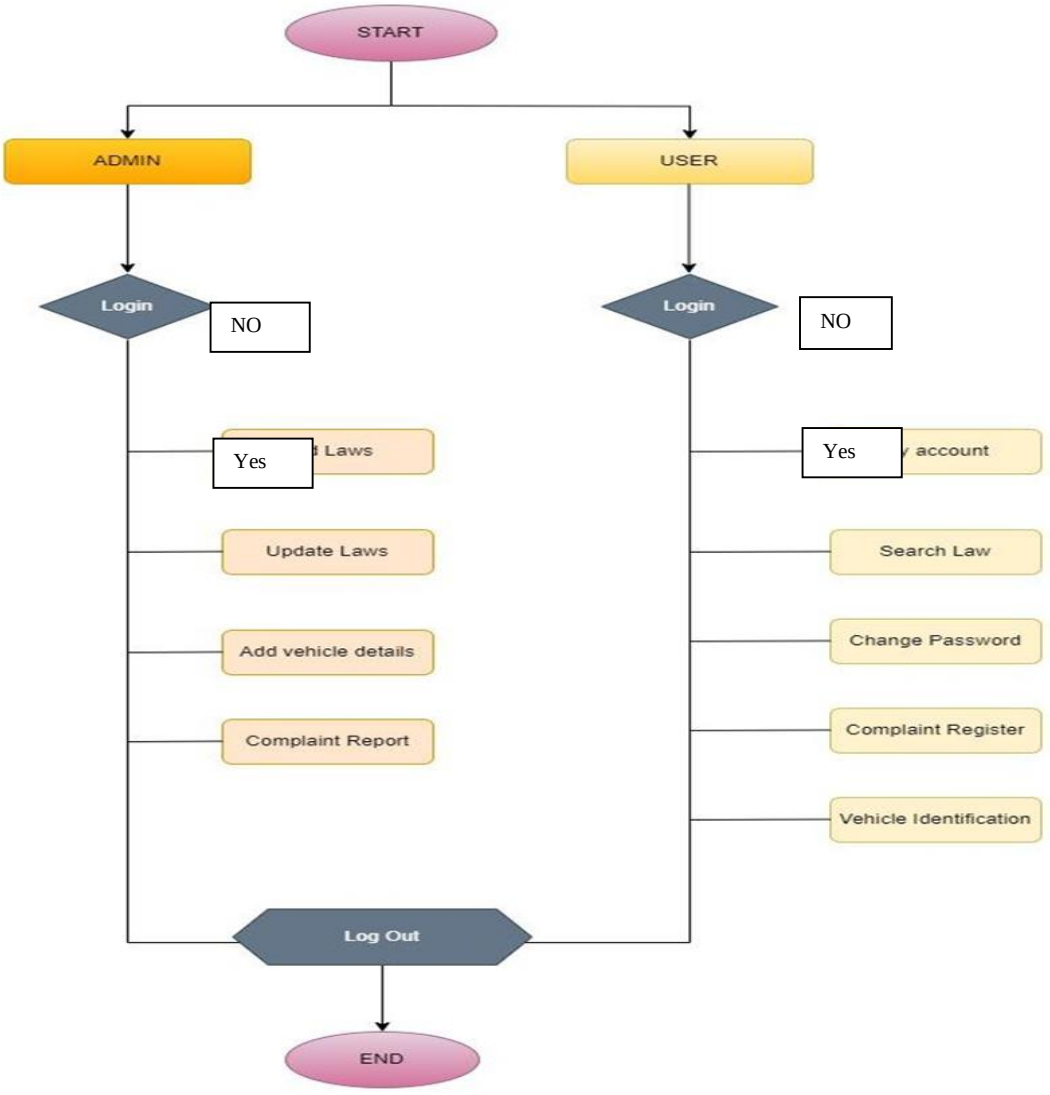


Fig 3.2 System Flow Diagram of Online Traffic Law Consultation System

3.3Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

Collections used in Application

A collection is a grouping of documents. Documents within a collection can have different fields. A collection is the equivalent of a table in a relational database system. A collection exists within a single database.

Login

```
[
{
"email" : "23mx215@psgtech.ac.in",
"password" : "23mx215",
}
]
```

Registration

```
[
{
"name" : "Marian A",
"email" : "23mx215@psgtech.ac.in",
"password" : "23mx215",
}
]
```

Law Details

```
[
{
"keyword" : "search keyword",
"law" : "actual elaborated law",
}
]
```

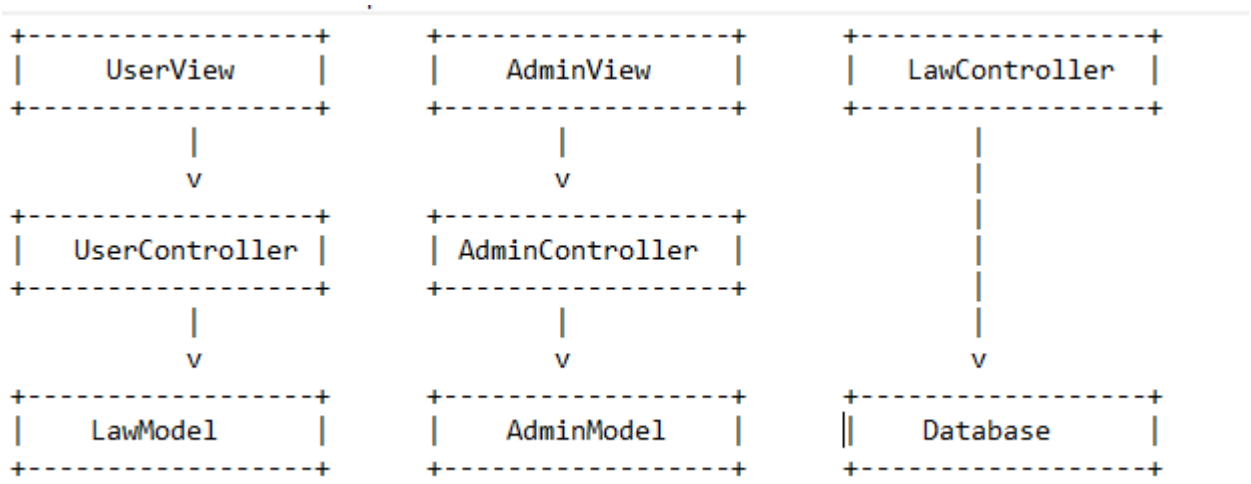
Feedback

```
[
{
"feedback" : "feedback from the user",
}
]
```

CHAPTER 4

SYSTEM IMPLEMENTATION

This chapter discusses the system implementation. System implementation is a set of procedures performed to complete the design contained in the systems design document.



4.1 Sign-in Module

The Sign-In functional module is designed to manage user authentication and access control within the Online Traffic law system. Its primary purpose is to verify the identity of users and grant them appropriate access based on their credentials. The module ensures that only authorized users can log in to the application, and they Sign-In to the system. Figure 4.1 shows the Sign up Module

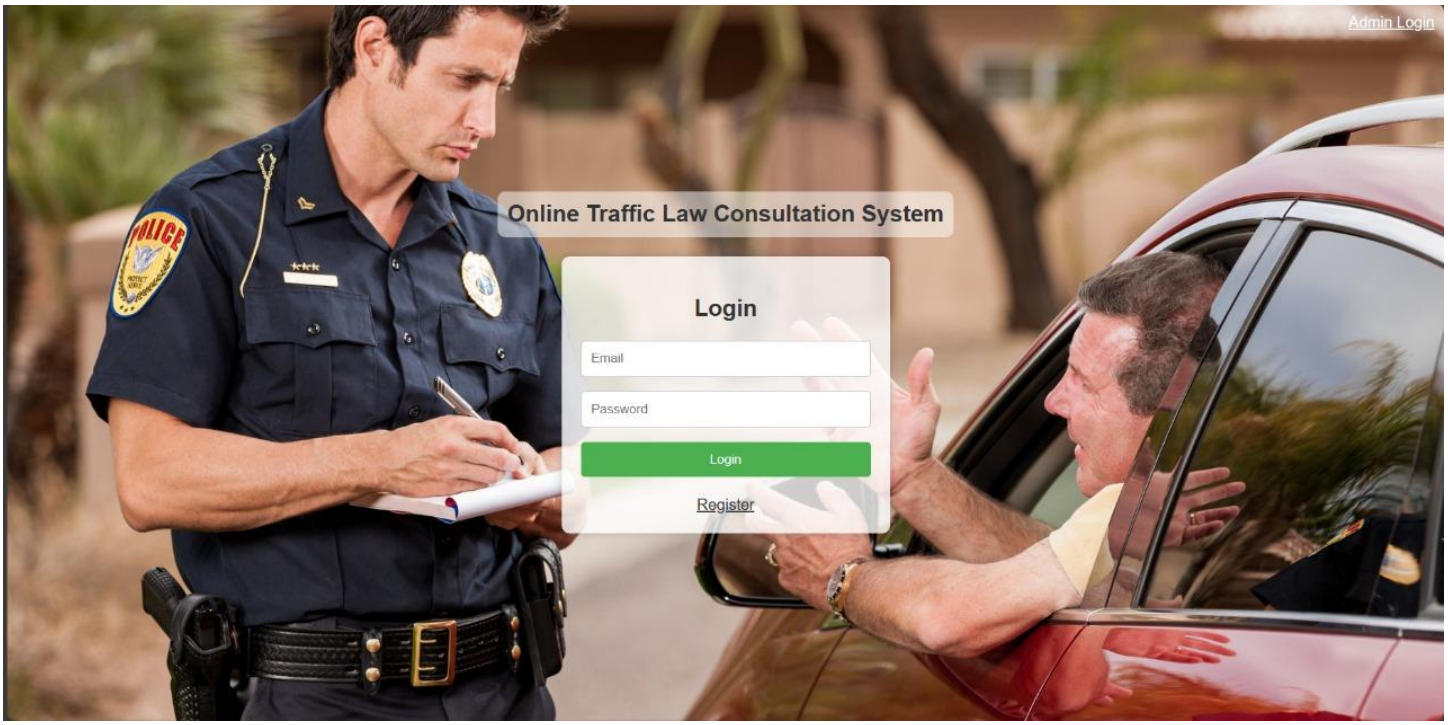


Fig 4.1 – User Login

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Login Page</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            font-family: Arial, sans-serif;
            background-image:
url('https://www.findlawimages.com/content/original-images/Police-
officer-writing-a-ticket-to-combative-driver.jpg'); /* Replace with
your image URL */
            background-size: cover;
            background-position: center;
            height: 100vh;
            overflow: hidden;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
        }
```

```
.title-box {
    background-color: rgba(255, 255, 255, 0.5);
    padding: 10px;
    border-radius: 8px;
    margin-bottom: 20px;
}

.title {
    color: #333;
    font-size: 24px;
    margin: 0;
}
.login-container {
    background-color: rgba(255, 255, 255, 0.8);
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
    text-align: center;
    width: 300px;
}

.login-container h2 {
    color: #333;
}

.login-form {
    display: flex;
    flex-direction: column;
    margin-top: 20px;
}

.login-form input {
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

.login-form button {
    background-color: #4caf50;
    color: #fff;
    padding: 10px;
    border: none;
```

```
            border-radius: 4px;
            cursor: pointer;
        }

        .login-form button:hover {
            background-color: #45a049;
        }

        .register-link {
            margin-top: 20px;
            color: #333;
            text-decoration: underline;
            cursor: pointer;
        }
        .admin-login {
            position: absolute;
            top: 10px;
            right: 10px;
            color: #fff;
            cursor: pointer;
            text-decoration: underline;
        }
    </style>
</head>
<body>

  <div class="title-box">
    <h1 class="title">Online Traffic Law Consultation System</h1>
</div>
<div class="admin-login" onclick="redirectToAdminLogin()">Admin
Login</div>
    <div class="login-container">
        <h2>Login</h2>
        <form class="login-form"action="/index"method="post">
            <input type="email"
placeholder="Email"name="email"id="email" required>
            <input type="password"
placeholder="Password"name="password"id="password" required>
            <button type="submit">Login</button>
        </form>
        <div class="register-link"
onclick="redirectToRegister()">Register</div>
    </div>

    <script>
```

```
        function redirectToRegister() {
            window.location.href = "/register"; // Replace with your
registration page URL
        }
        function redirectToAdminLogin() {
            window.location.href = "/admin"; // Replace with your admin
login page URL
        }
    </script>

</body>
</html>
```

4.2 Admin Module

The admin module of our Online Traffic Law Consultation System empowers admin with effortless control. The admin can seamlessly add, update, and delete traffic laws in real-time. This feature ensures the platform stays current, adapting to the evolving legal landscape promptly. The user-friendly interface guarantees efficient law management, making it a powerful tool for administrators to maintain an up-to-date and reliable source of information. Welcome to a streamlined approach to traffic law administration.Fig 4.2 shows the Admin Module.

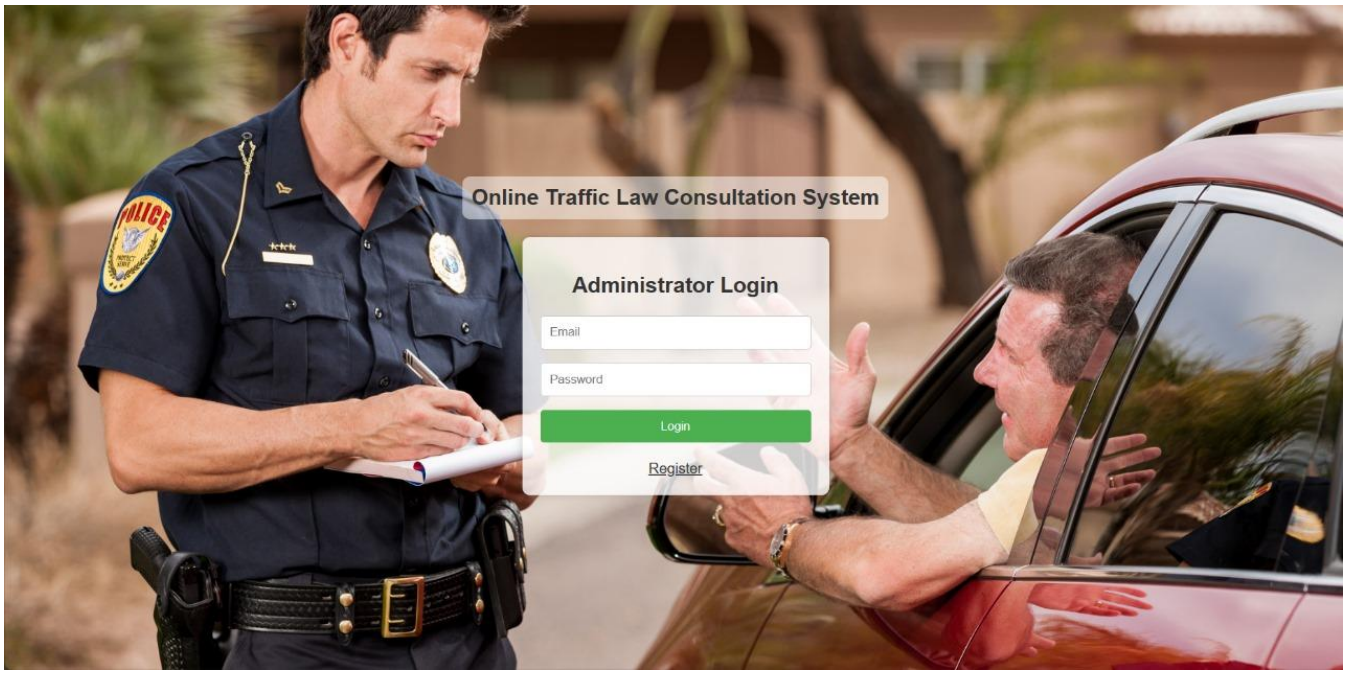


Fig 4.2 - Admin Login

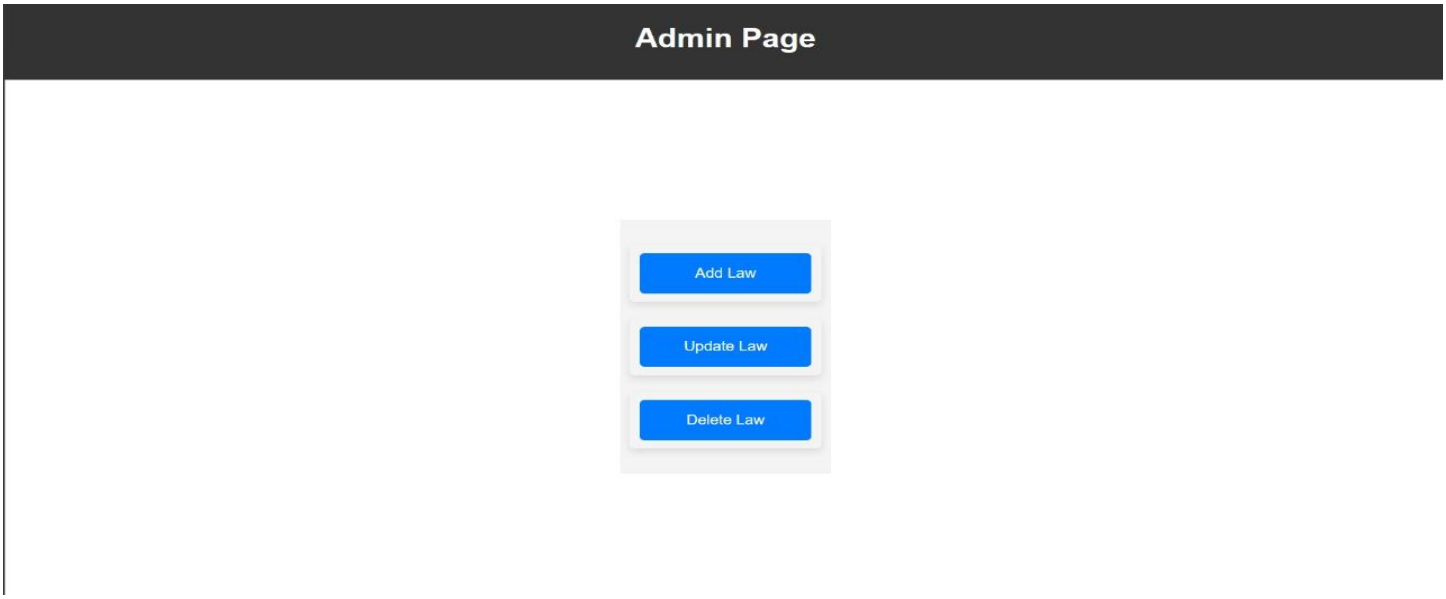
// ADMIN ADDING LAW

```
  app.post('/addlaw', async (req, res) => {
        try {
            const { keyword,law } = req.body;
            console.log('Received Form Data:', req.body);
            const client = await MongoClient.connect(mongoURI, {
useNewUrlParser: true, useUnifiedTopology: true });
            const db = client.db();

            const addlaw = {
                keyword,
                law,
            };

            await db.collection('law_details').insertOne(addlaw);

            client.close();

            res.json({ message: 'law added successfully' });
        } catch (error) {
            console.error(error);
            res.status(500).json({ message: 'Internal Server Error'
});
        }
    });
```

Fig 4.2 Home Page Module

Fig 4.2 Home Page Module

Fig 4.2 Admin Module



The figure 4.2 shows the admin module

Fig 4.2 Admin Add Law Module

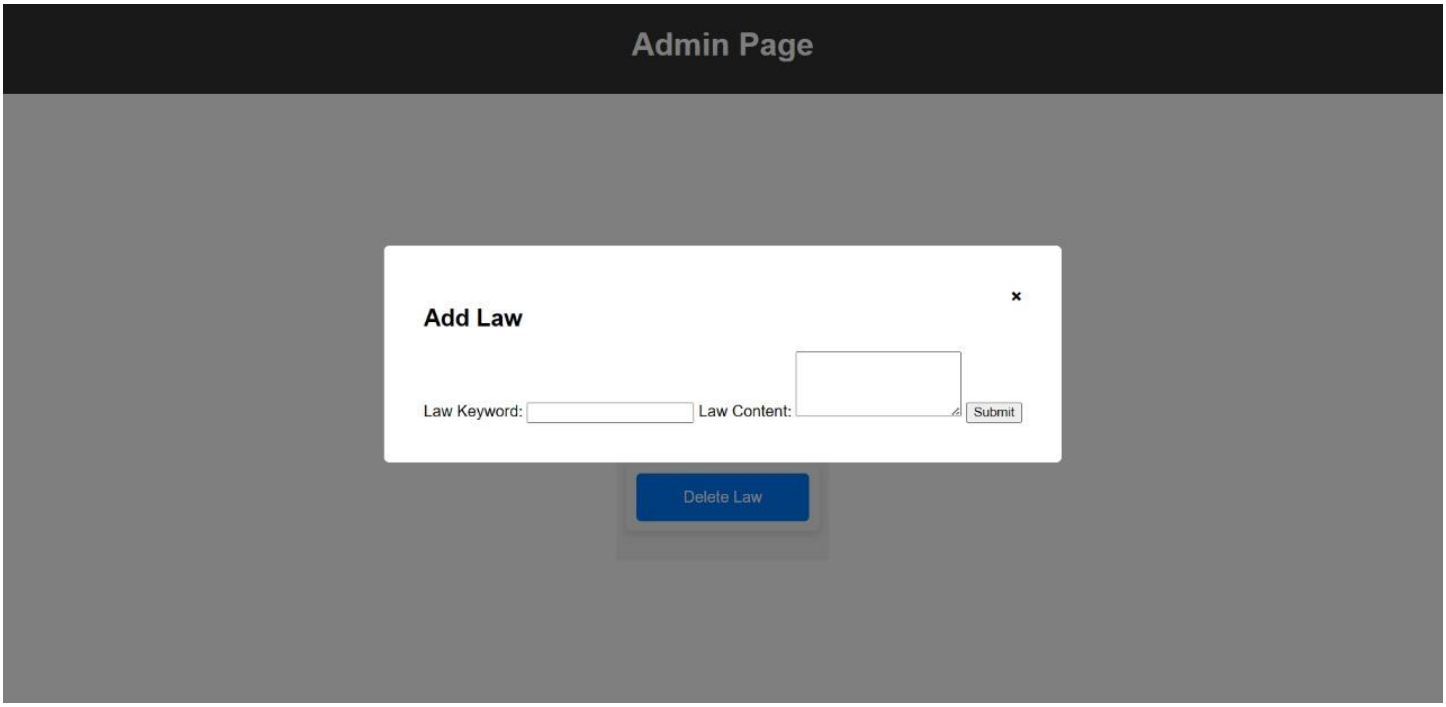


Fig 4.2 Admin Update Law Module

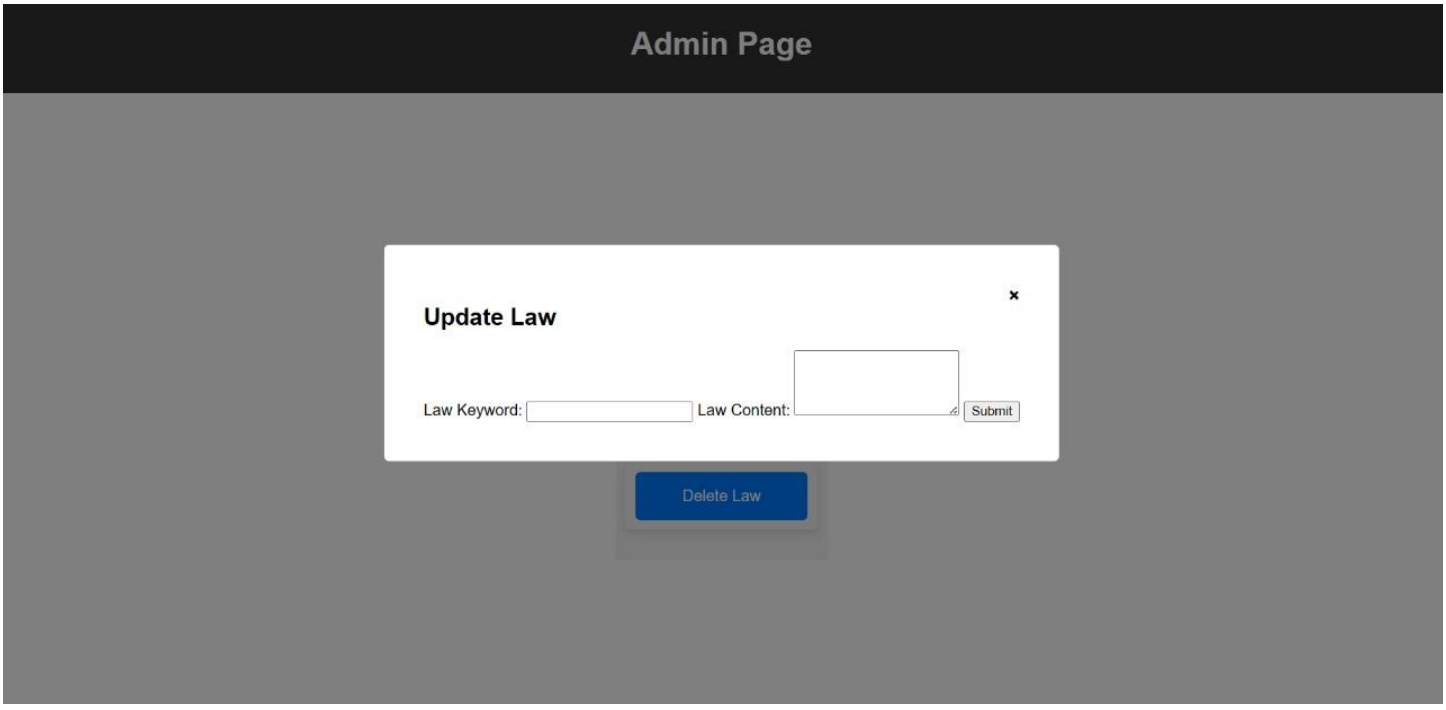


Fig 4.2 Admin Delete Law Module

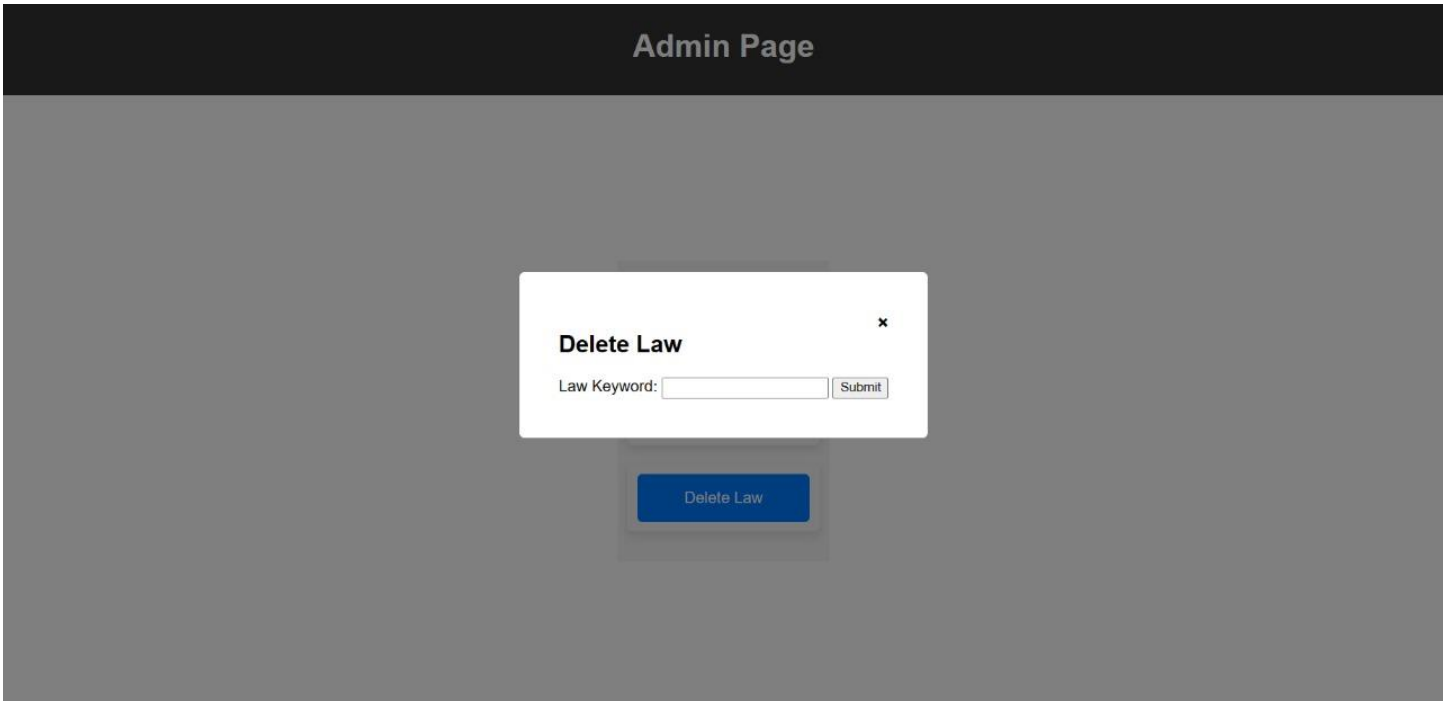


Fig 4.3 Search Module

Our Search Module offers users flexibility with both keyword and full-text searches. Simply type a keyword to quickly retrieve specific laws or enter entire text for a comprehensive search experience. This user-friendly functionality ensures efficient access to relevant traffic laws, tailored to individual Needs. Welcome to a hassle-free way of navigating the traffic regulations. Fig 4.2 shows the Search Module.

Fig 4.3 Search Module

//USER SEARCH

```
app.post('/search', async (req, res) => {
      try {
          const { keyword } = req.body;
          console.log('Received Form Data:', req.body);

          const client = await MongoClient.connect(mongoURI, {
              useNewUrlParser: true,
              useUnifiedTopology: true
          });

          const db = client.db();

             // Use find() for multiple results or findOne() for a single
 result
             const lawDetailsCursor = db.collection('law_details').find({
 keyword });

          console.log('Law Details Cursor:', lawDetailsCursor);

          const lawDetails = [];
          await lawDetailsCursor.forEach(doc => {
              lawDetails.push(doc);
          });

          console.log('Law Details from Database:', lawDetails);

          if (lawDetails.length === 0) {
                return res.status(404).json({ message: 'Search details not
 found' });
             }
```

```
            res.render('searchresults', { lawDetails });
        } catch (error) {
            console.error(error);
            res.status(500).json({ message: 'Internal Server Error' });
        }
    });
```

Fig 4.4 Feedback Module

Our Complaint Register and Feedback Module allow users to easily register complaints through us website. User feedback is the driving force behind continuous improvements, ensuring our website Evolves based on real user experiences. Enhancing our platform through valuable feedback. Fig 4.4 shows the Feedback Module.

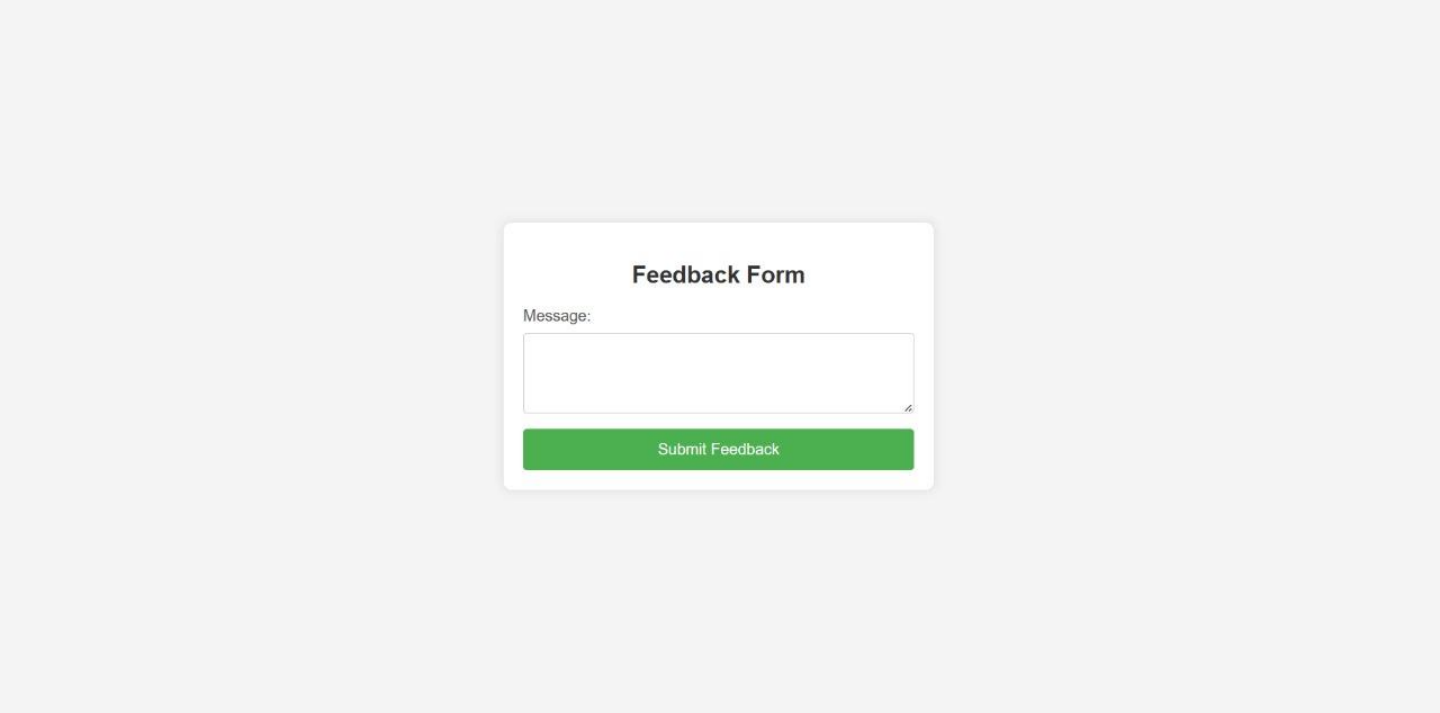


Fig 4.4 Feedback Module

CHAPTER 5

TESTING

This chapter delves into the critical phase of system testing, elucidating diverse testing methodologies employed throughout the application's development lifecycle. Rigorous testing was conducted at various stages to detect and rectify bugs promptly, ensuring the delivery of flawless end products. The primary goal was to validate that the expected outcomes align seamlessly with the defined inputs.

5.1 Testing Techniques

The testing process intricately examines the software's logical internals, affirming the completeness of code execution and scrutinizing functional aspects. It guarantees that specified inputs yield results consistent with the anticipated outcomes. Testing is integral to the development cycle, its extent contingent upon the application's size and complexity. This chapter elucidates the diverse testing strategies embraced in this project.

User Interface Testing

GUI Testing is a software testing type that checks the Graphical User Interface of the Software. The purpose of Graphical User Interface (GUI) Testing is to ensure the functionalities of software application work as per specifications by checking screens and controls like menus, buttons, icons, etc. The following are checklist to ensure the user interface testing,

- Check all the GUI elements for size, position, width, length, and acceptance of characters or numbers
- Check the intended functionality of the application using the GUI.
- Check Error Messages are displayed correctly.
- Check Font used in an application is readable.
- Check the alignment of the text is proper.

- Check the Colour of the font and warning messages is aesthetically pleasing.
- Check that the images are properly aligned.

5.2 Test Case Report

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to the customer. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques.

Below are the test cases for each module along with the expected outcomes:

Admin Module

1. Login Functionality:

Test Case 1: Verify that the admin can log in with valid credentials.
Expected Outcome: Successful login.

Test Case 2: Verify that the admin cannot log in with invalid credentials.
Expected Outcome: Unsuccessful login, display an error message.

2. Add Laws

Test Case 3: Verify that the admin can add a new Law with valid information.
Expected Outcome: Law is added successfully.

Test Case 4: Verify that the admin cannot add a Law with incomplete or invalid information.
Expected Outcome: Law addition fails, displays an error message.

3. Delete Laws

Test Case 5: Verify that the admin can delete a Law.
Expected Outcome: Law is deleted successfully.

Test Case 6: Verify that the admin cannot delete a non-existent Law.
Expected Outcome: Deletion fails, displays an error message.

4. Update Laws::

Test Case 7: Verify that the admin can update the information of a Law.

Expected Outcome: Law information is updated successfully.

Test Case 8: Verify that the admin cannot update the information of a non-existent Law.

Expected Outcome: Update fails, displays an error message.

User Module:

1. Register:

Test Case 9: Verify that a user can register with valid information.

Expected Outcome: User is registered successfully.

Test Case 10: Verify that a user cannot register with incomplete or invalid information.

Expected Outcome: Registration fails, display an error message.

2. Search Law:

Test Case 11: Verify that a user can Search a Law...

Expected Outcome: Relevant laws according to the result is displayed.

Test Case 12: Verify that a user cannot Search a Law that is not available.

Expected Outcome: Searching fails, display an error message.

3. Register Complaint:

Test Case 13: Verify that a user can register a complaint.

Expected Outcome: Redirect to the Government's official page.

4. Give Feedbacks;

Test Case 15: Verify that a user can give their feedback.

Expected Outcome: Feedback is received and updated to the admin.

Sign In Module:

1. Valid Sign In:

Test Case 16: Verify that a user can sign in with valid credentials.

Expected Outcome: Successful sign-in.

Test Case 17: Verify that the user cannot sign in with incorrect credentials.

Expected Outcome: Unsuccessful sign-in, display an error message.

2. Forgot Password:

Test Case 18: Verify that the user can reset the password using the "Forgot Password".

Expected Outcome: Password reset is successful, and the user can log in with the new password.

Test Case 19: Verify that the user cannot reset the password without providing valid information.

Expected Outcome: Password reset fails, display an error message.

3. Security Measures:

Test Case 20: Verify that the system has proper security measures in place to protect user credentials.

Expected Outcome: The system has secure measures, and user credentials are protected.

CHAPTER 6

CONCLUSION

Online Traffic Law Consultation System simplifies the understanding of traffic rules for everyone. It acts as a friendly guide, ensuring that both administrators updating details and users searching for information are well-supported. Experience a hassle-free approach to dealing with traffic rules, where obtaining accurate information is as simple as a quick search. Join us for a stress-free drive, and relish the simplicity of our online traffic law help system.

BIBLIOGRAPHY

BOOK REFERENCES

1. MongoDB: The Definitive Guide" by Kristina Chodorow and Michael Dirolf.
2. "Node.js Web Development" by David Herron
3. Web Development with Node and Express: Leveraging the JavaScript Stack by Ethan Brown

WEB REFERENCES

1. https://www.w3schools.com/
2. https://stackoverflow.com/
3. https://github.com/
4. https://nodejs.org/en
5. https://www.mongodb.com/