SIMATS SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES CHENNAI-602105

Cab Booking Management System

A CAPSTONE PROJECT REPORT

Submitted in the partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

Submitted by

T. SAISANTHOSH (192211860)

P. NAVEEN (192210649)

Under the Supervision of

M.VENKATRAMANA

JULY 2024

DECLARATION

We, **T.SAI SANTHOSH, P. NAVEEN** students of **Bachelor of Engineering in**, Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Cab Booking Management System** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

Date:

Place:

CERTIFICATE

This is to certify that the project entitled **“Cab Booking”** submitted by**T.SAI SANTHOSH,P.NAVEEN** has been carried out under my supervision. The project has been submitted as per the requirements in the current semester of B.E Computer Science.

Teacher-in-charge

Venkatraman

Table of Contents

ABSTRACT:

The Cab Booking Database Management System (CBMS) project aims to develop a robust application for managing cab booking information using Java for frontend development and MySQL for backend database management. This system facilitates efficient organization and retrieval of cab booking records, offering functionalities for booking, cancelling, tracking, and updating cab reservations. The database schema includes essential attributes such as BookingID, CustomerID, DriverID, PickupLocation, DropLocation, PickupDateTime, and Fare Amount, ensuring comprehensive data storage and management.

The application's frontend is implemented in Java, utilizing JDBC (Java Database Connectivity) to establish a seamless connection with MySQL. This setup enables a secure and reliable user interface and database backend interaction. The project emphasizes data integrity and security, ensuring that cab booking information is accurately maintained and accessible. Overall, the Cab Booking Database Management System project demonstrates the effective utilization of modern software development practices to provide a reliable solution for transportation service providers' booking management needs. The Car Booking System is an automated platform designed to streamline the process of renting cars for personal and business use. This system aims to enhance the customer experience by providing a user-friendly interface for booking vehicles, managing reservations, and handling payments. The platform offers a comprehensive database of available cars, including their specifications, availability status, and rental rates. Key features of the system include user authentication, real-time booking updates, vehicle availability tracking, and detailed booking history. Additionally, the system integrates with payment gateways to facilitate secure transactions. The Car Booking System is developed using a robust backend to ensure reliability and scalability, coupled with a responsive frontend for seamless user interaction. This system not only simplifies the booking process for customers but also provides car rental agencies with efficient tools for managing their fleet and reservations.

The Car Booking System is designed to automate and streamline the process of renting cars, offering a seamless experience for customers and car rental agencies. The system features a user-friendly interface with two main portals: the Customer Portal, which allows users to search for available cars, view details, make reservations, and process payments, and the Admin Portal, which enables car rental agencies to manage their fleet, track bookings, handle customer inquiries, and generate reports. Key functionalities include secure user authentication, robust search and filter options, real-time availability tracking, and comprehensive booking management. Integration with secure payment gateways ensures safe online transactions, while automated email and SMS notifications keep users informed about booking confirmations, reminders, and cancellations. Additionally, the system incorporates a feedback mechanism for customers to rate and review their rental experiences. The backend architecture supports reliable database management, ensuring data integrity and scalability to handle growing user demands. This Car Booking System not only simplifies the rental process for customers but also provides car rental agencies with efficient tools to optimize their operations

INTRODUCTION:

In today's digital age, efficient data management plays a critical role across diverse industries, including transportation services. The Cab Booking Management System (CBMS) project addresses this need by offering a comprehensive solution for organizing and managing cab booking information. This system utilizes Java for frontend development and MySQL for backend database management, providing a robust framework to facilitate seamless interaction and manipulation of booking records. Transportation service providers generate substantial data related to bookings, including customer details, trip information, and payment records. Effectively managing this data ensures smooth operations, enables informed decision-making, and enhances customer satisfaction. The CBMS project aims to streamline these processes by centralizing booking data into a structured MySQL database. This database employs a well-defined schema designed for efficient storage and retrieval, featuring essential fields such as BookingID, CustomerID, DriverID, PickupLocation, DropLocation, PickupDateTime, and Fare Amount.

These attributes collectively capture key aspects of booking transactions, facilitating comprehensive management and analysis. For frontend development, the CBMS leverages Java, renowned for its versatility and reliability in building robust applications. Java's JDBC (Java Database Connectivity) API enables seamless integration with MySQL, empowering functionalities such as booking management, reservation updates, trip tracking, and fare calculation. This frontend-backend synergy ensures that stakeholders, including customers and administrative personnel, can access and manage booking information through an intuitive and responsive interface.

1.Project Description Project Title:

Cab Booking Project Overview:

The Cab Booking Software project aims to create a comprehensive and user-friendly online platform tailored specifically for the transportation industry. This software will serve as an essential tool for cab service providers to manage their operations, interact with customers, and optimize their services. By integrating advanced booking capabilities, customizable features, and robust analytics, the platform will empower cab businesses to enhance their efficiency, drive sales, and build a loyal customer base.

Key Objectives:

1.1 Elegant and Customizable Design:

Develop a Range of Aesthetically Pleasing and Customizable Design Templates: Create a variety of templates that can be tailored to reflect the unique identity of each cab service provider.

Support High-Resolution Images and Video Content: Ensure the platform supports highquality images and videos for a visually engaging user experience.

1.2 E-commerce Integration:

User-Friendly Interface: Design a straightforward and intuitive interface that allows users to easily navigate and manage their bookings.

Accessibility and Responsiveness: Ensure the platform is accessible and responsive across all devices, including desktops, tablets, and smartphones.

Analytics and Insights: Integrate advanced analytics tools to track booking performance, customer behavior, and website traffic. Provide actionable insights to help businesses make data-driven decisions and optimize their operations.

Social Media Integration: Enable seamless integration with social media platforms to enhance online marketing efforts and community engagement. Allow users to share content, promote services, and connect with their audience through various social media channels.

Customer Engagement and Support: Incorporate features for customer reviews, feedback, and support to foster a strong relationship between cab service providers and their customers. Provide tools for personalized marketing and communication to enhance customer loyalty.

2.3 Project Scope:

Requirement Analysis:

Conduct a detailed analysis of the needs and preferences of cab service providers.

Gather requirements through market research, surveys, and consultations with industry experts.

Design and Development:

Create wireframes and design mockups based on gathered requirements.

Develop the website using modern web technologies and best practices for security, performance, and scalability.

Testing and Quality Assurance:

Conduct comprehensive testing to ensure the website is free of bugs and performs optimally.

Implement user testing sessions to gather feedback and make necessary adjustments.

Deployment and Launch:

Deploy the website on a reliable hosting platform.

Ensure a smooth launch with a marketing strategy to attract initial users.

Post-Launch Support and Maintenance:

Provide ongoing technical support and updates to ensure the website remains up-to-date and functional.

Gather user feedback for continuous improvement and feature enhancement

2.ProblemDescription Key

Problems to Address:

3.1 Complexity of Website Creation:

Technical Expertise: Cab service providers often lack the technical expertise required to create and maintain a professional-looking website.

Simplified Process: There is a need for a simplified process that allows users to build and customize their websites without extensive technical knowledge.

3.2 Visual Presentation:

Visual Appeal: Cab services rely heavily on visual appeal to attract customers.

High-Quality Media: The software must support high-quality images and videos, providing an engaging and immersive user experience.

3.3 E-commerce Integration:

Seamless E-commerce Features: Many cab service providers struggle with integrating ecommerce functionalities into their websites. The software should offer seamless e-commerce features, including service listings, booking management, and secure payment processing.

3.4 Responsive Design:

Cross-Device Compatibility: With a significant number of users accessing websites via mobile devices, the software must ensure a responsive design that works well across all devices.

3.5 Analytics and Insights:

Performance Tracking: Cab service providers need access to analytics to understand booking trends and customer behavior.

Data-Driven Decisions: The software should include tools for tracking performance and generating insights to inform business decisions.

3.6 Social Media Integration:

Online Marketing: Effective online marketing is crucial for cab service providers to reach a broader audience.

Easy Integration: The software should facilitate easy integration with social media platforms for marketing and engagement purposes.

Tool Description User

Interface:

4.1 Dashboard:

Central Hub: Central hub for managing all aspects of the website.

Overview of Key Metrics: Provides an overview of key metrics such as bookings, traffic, and customer activity.

Quick Access: Quick access to various tools and settings.

4.2 Drag-and-Drop Editor:

Customizable Layout: Allows users to easily customize their website layout without any coding knowledge.

Pre-built Sections: Includes pre-built sections (e.g., header, footer, booking forms) that can be dragged and dropped into place.

Real-Time Preview: Real-time preview feature to see changes as they are made.

4.3 Template Library:

Professionally Designed Templates: A collection of professionally designed, customizable templates tailored for cab service providers.

Categorized Templates: Templates are categorized based on different styles and themes to suit various brand identities.

Easy Switching: Easy switching between templates with content preservation.

4.4 Responsive Design:

Cross-Device Compatibility: Ensures the website looks and functions well on all devices, including desktops, tablets, and smartphones.

Preview and Adjust: Built-in tools to preview and adjust the website for different screen sizes.

4.5 Tools and Technologies:

HTML/CSS: For creating the structure and styling of the website.

JavaScript: For interactive elements and client-side logic.

Responsive Frameworks: For ensuring the website is mobile-friendly.

Google Analytics: For tracking website traffic and user behavior.

4.6 Marketing and SEO Tools:

SEO Optimization: Tools for optimizing website content for search engines, including meta tags, keywords, and sitemaps.

Email Marketing: Integration with email marketing services for newsletters and promotional campaigns.

Social Media Integration: Easy sharing of services and content on social media platforms, with tracking of engagement metrics.

Operations:

5.1 Customizations:

Template Selection and Customization: Users can choose and customize templates.

Drag-and-Drop Interface: JavaScript frameworks (like React, Vue, or Angular) can be used to build a user-friendly drag-and-drop interface for layout customization.

Real-Time Preview: Implementing real-time preview using WebSocket or AJAX for live updates.

5.2 E-commerce Integration:

Tools and Technologies:

Payment Gateways (e.g., Stripe, PayPal): For handling transactions.

Shopping Cart Libraries (e.g., Snipcart): For managing the shopping cart.

Operations:

Service Management: Using a backend (Node.js, Django, or Ruby on Rails) for CRUD operations on service listings.

Payment Processing: Integrating APIs from payment gateways to handle transactions securely.

Booking Management: Building backend logic to manage bookings, track rides, and handle cancellations.

5.3 Inventory Management:

Tools and Technologies:

Database Management (e.g., MongoDB, MySQL): For storing service and inventory data.

Server-Side Scripting (e.g., Node.js, Python): For handling inventory logic.

Operations:

Stock Level Monitoring: Implementing server-side scripts to monitor stock levels and trigger alerts for low inventory.

Automated Reordering: Setting up automated systems to reorder supplies when stock levels fall below a certain threshold.

5.4 Customer Engagement:

Tools and Technologies:

Email APIs (e.g., SendGrid, Mailchimp): For email marketing campaigns.

Social Media APIs (e.g., Facebook, Instagram): For social media integration.

Live Chat Tools (e.g., Intercom, Tawk.to): For real-time customer support.

Operations:

Email Campaigns: Using email APIs to send newsletters and promotional emails

Social Media Integration: Utilizing social media APIs to share services and content directly from the website.

Customer Reviews and Ratings: Implementing user feedback forms and displaying reviews using JavaScript and server-side scripting.

5.5 Analytics and Reporting:

Tools and Technologies:

Google Analytics: For tracking website traffic and user behavior.

Custom Analytics Dashboards: Built using JavaScript libraries like Chart.js or D3.js.

Operations:

Sales Analytics: Using server-side scripting to aggregate sales data and visualize it on custom dashboards.

Customer Behavior Tracking: Integrating Google Analytics to monitor user interactions and generate insights.

Performance Reports: Automated generation of reports on website performance, bookings, and customer engagement.

5.6 Security and Compliance:

Tools and Technologies:

SSL Certificates: For secure data transmission.

Data Protection Tools: For ensuring compliance with regulations like GDPR.

Operations:

User Authentication: Implementing secure login systems using OAuth or JWT.

Data Encryption: Using HTTPS and encrypting sensitive data in the database.

Compliance Monitoring: Regularly auditing the website for compliance with data protection regulations.

6. **Approach / Module Description / Functionalities:**

- ***Home*****:** Provide a home page that redirects to the Seller or Buyer sections.
- ***Login*****:** Display a login form for username and password. Provide an option to register if the user is new.

- ***Register*:** Allow new users to register by creating a username and password.
- ***Booking:*** Allow booking to enter their name, address, and a description of their service. Driver: driving register creating a user role.
- ***Payment***: We can select what method of transaction we want by EMI or any Card or Online payments.
- ***Feedback***: driver and user are share their experience

7. Implementation HTML Code

Login page

```
<!DOCTYPE html>
<html> <head>
   <title>Login Form</title>
   <style>        body {            font-
family: Arial, sans-serif;
background-color: #f0f0f0;
display: flex;            justify-content:
center;            align-items: center;
height: 100vh;            margin: 0;
       }
       .login-container {
background-color: #fff;
padding: 20px;            border-
radius: 5px;
           box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
       }        h1 {
text-align: center;
color: #333;
       }        label {
display: block;
margin-bottom: 5px;
color: #333;
       }
       input[type="text"],
input[type="password"] {
width: 100%;            padding:
10px;            margin-bottom:
20px;            border: 1px
solid #ccc;
           border-radius: 5px;
       }        button {            width:
100%;            padding: 10px;
background-color: #28a745;
color: #fff;            border: none;
border-radius: 5px;
           cursor: pointer;
       }
button:hover {
           background-color: #218838;
       }
       .forgot-password,
.signup {
           background-color: #dedcd7;
       }
```

```
        .forgot-password:hover,
.signup:hover {
            background-color: #332f5d;
        }
    </style>
</head>
<body>
    <div class="login-container">
        <h1>Cab Booking</h1>
        <form id="loginForm" action="login.php" method="post">
            <label for="username">Username</label>
            <input type="text" id="username" name="username" required>

            <label for="password">Password</label>
            <input type="password" id="password" name="password" required>

            <button type="submit">Login</button>
            <button type="button" class="forgot-password">Forgot Password</button>
            <button type="button" class="signup">Sign Up</button>
        </form>
    </div>
    <script>
        document.getElementById('loginForm').addEventListener('submit', function(event) {
var password = document.getElementById('password').value;

            if (password.length >= 6) {
                alert('Password must be less than 6 characters to login.');
event.preventDefault(); // Prevent form submission
            }
        });
    </script>
</body>
</html>
```

REGISTRATION

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cab Booking Registration Form</title>
    <style>        body {            font-
family: Arial, sans-serif;
background-color: #f2f2f2;
            margin: 0;
padding: 0;
```

```
        }
        .container {            max-
width: 600px;            margin:
50px auto;
background-color: #fff;
padding: 20px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            border-radius: 8px;
        }
        .container h2 {
text-align: center;
            margin-bottom: 20px;
        }
        .form-group {            margin-
bottom: 15px;
        }
        .form-group label {
display: block;
            margin-bottom: 5px;
        }
        .form-group input {
width: 100%;            padding:
8px;            box-sizing:
border-box;            border:
1px solid #ccc;            border-
radius: 4px;
        }
        .form-group button {
width: 100%;            padding: 10px;
background-color: #007BFF;
            color: white;
border: none;
border-radius: 4px;
cursor: pointer;
            font-size: 16px;
        }
        .form-group button:hover {
background-color: #0056b3; }
    </style>
</head>
<body>

<div class="container">
    <h2>Cab Booking Registration Form</h2>
    <form action="reg.php" method="post">
        <div class="form-group">
            <label for="first_name">First Name:</label>
```

```
        <input type="text" id="first_name" name="first_name" required>
</div>
      <div class="form-group">
        <label for="last_name">Last Name:</label>
        <input type="text" id="last_name" name="last_name" required>
</div>
      <div class="form-group">
        <label for="phone">Phone Number:</label>
        <input type="tel" id="phone" name="phone" required>
      </div>
      <div class="form-group">
        <label for="email">Email Address:</label>
        <input type="email" id="email" name="email" required>
      </div>
      <div class="form-group">
        <button type="submit">Register</button>
      </div>
   </form>
</div>

</body>
</html>
```

BOOKING

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Cab Booking</title>
</head>
<body>
   <h2>Cab Booking</h2>
   <form id="cabBookingForm">
      <div>
        <label for="pickupLocation">Pickup Location:</label>
        <input type="text" id="pickupLocation" required>
</div>
      <div>
        <label for="dropLocation">Drop Location:</label>
        <input type="text" id="dropLocation" required>
      </div>
      <div>
        <label for="pickupDate">Pickup Date:</label>
        <input type="date" id="pickupDate" required>
      </div>
      <div>
```

```
        <label for="pickupTime">Pickup Time:</label>
        <input type="time" id="pickupTime" required>
      </div>
      <button type="submit">Book Cab</button>
    </form>
</body>
</html>
```

DRIVER REGISTRATION

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Driver Registration</title>
</head>
<body>
    <h2>Driver Registration</h2>
    <form id="driverRegistrationForm">
        <div>
            <label for="driverName">Name:</label>
            <input type="text" id="driverName" required>
        </div>
        <div>
            <label for="licenseNumber">License Number:</label>
            <input type="text" id="licenseNumber" required>
        </div>
        <div>
            <label for="vehicleNumber">Vehicle Number:</label>
            <input type="text" id="vehicleNumber" required>
        </div>
        <div>
            <label for="phone">Phone Number:</label>
            <input type="tel" id="phone" required>
        </div>
        <div>
            <label for="email">Email:</label>
            <input type="email" id="email" required>
    </div>
        <button type="submit">Register</button>
    </form>
</body>
</html>
```

Admin Dashboard (Example)

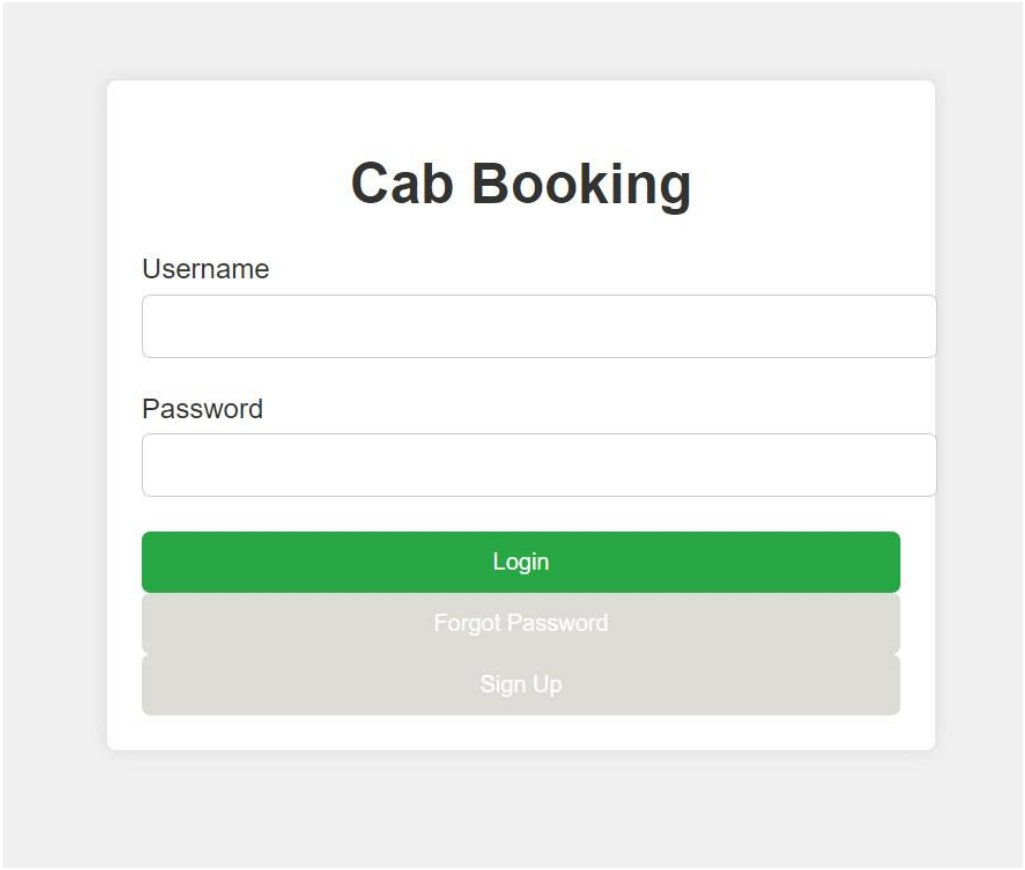
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Admin Dashboard</title>
</head>
<body>
   <h2>Admin Dashboard</h2>
   <div>
      <button onclick="viewUsers()">View Users</button>
      <button onclick="viewBookings()">View Bookings</button>
      <button onclick="viewDrivers()">View Drivers</button>
   </div>
   <div id="content">
      <!-- Dynamic content will be loaded here based on button clicks -->
</div>

   <script>
      function viewUsers() {
         // Implement the logic to view users
         document.getElementById('content').innerHTML = '<p>User List</p>';
      }

      function viewBookings() {
         // Implement the logic to view bookings
document.getElementById('content').innerHTML = '<p>Booking List</p>';
      }

      function viewDrivers() {
         // Implement the logic to view drivers
         document.getElementById('content').innerHTML = '<p>Driver List</p>';
      }
   </script>
</body>
</html>
```

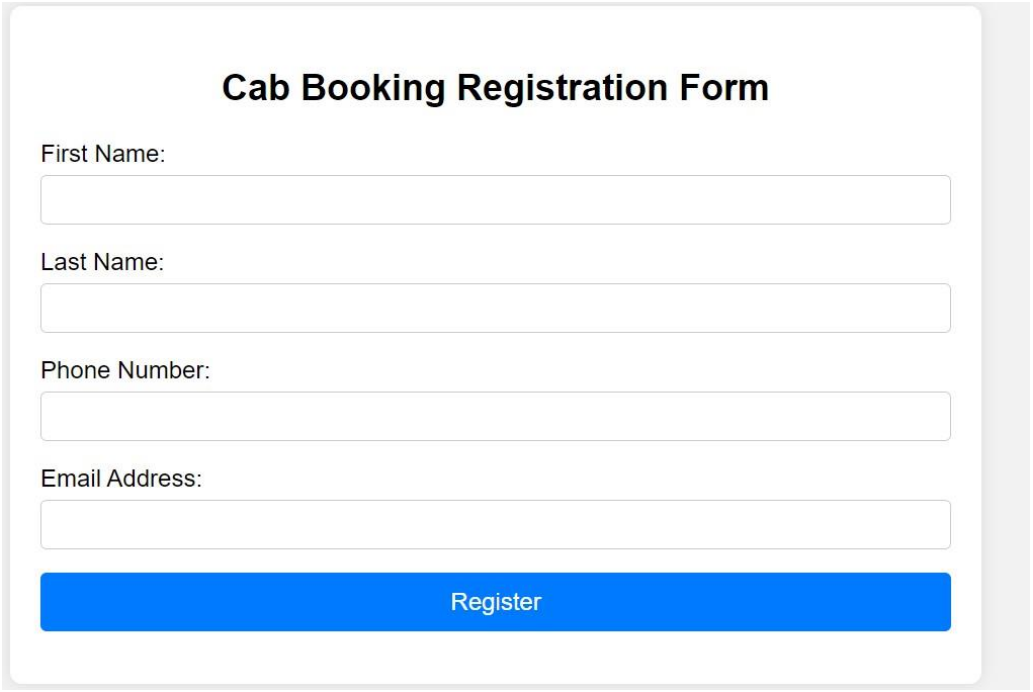
Booking

```
<!DOCTYPE html>
<html lang="en">
<head>
```

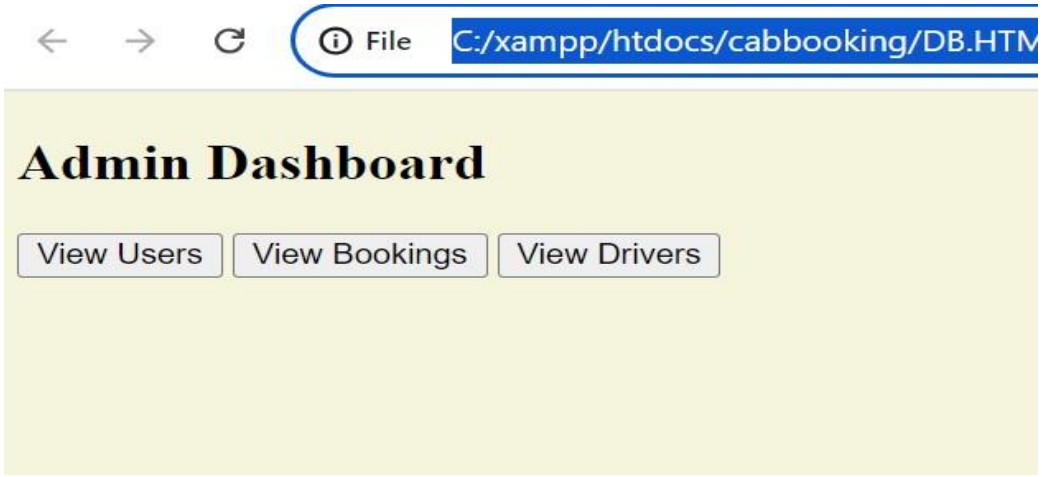
```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initialscale=1.0">
    <title>Cab Booking</title>
</head>
<body>
    <h2>Cab Booking</h2>
    <form id="cabBookingForm">
        <div>
            <label for="pickupLocation">Pickup Location:</label>
            <input type="text" id="pickupLocation" required>
        </div>
        <div>
            <label for="dropLocation">Drop Location:</label>
            <input type="text" id="dropLocation" required>
        </div>
        <div>
            <label for="pickupDate">Pickup Date:</label>
            <input type="date" id="pickupDate" required>
        </div>
        <div>
            <label for="pickupTime">Pickup Time:</label>
            <input type="time" id="pickupTime" requred>
        </div>
        <button type="submit">Book Cab</button>
    </form>
</body>
</html>
```

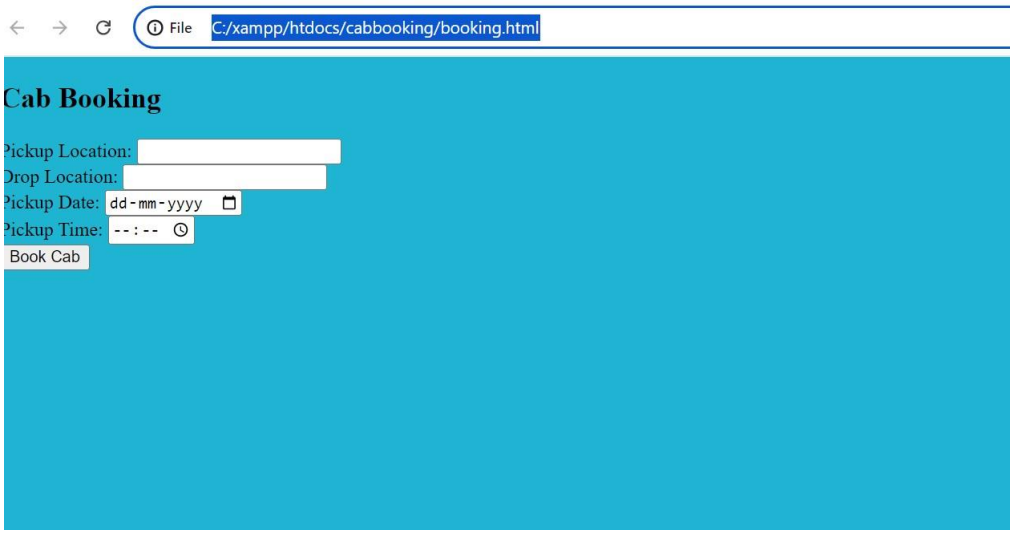
8.RESULT:


Cab Booking
Username
Password
Login
Forgot Password
Sign Up


Cab Booking Registration Form
First Name:
Last Name:
Phone Number:
Email Address:
Register


File
C:/xampp/htdocs/cabbooking/DR
Driver Registration
Name:
License Number:
Vehicle Number:
Phone Number:
Email:
Register


File
C:/xampp/htdocs/cabbooking/DB.HTM
Admin Dashboard
View Users
View Bookings
View Drivers


File
C:/xampp/htdocs/cabbooking/booking.html
Cab Booking
Pickup Location:
Drop Location:
Pickup Date:
dd-mm-yyyy
Pickup Time:
--:--
Book Cab

9.CONCLUSION:

The student management software provides a simple yet effective way to manage student information, including personal details, academic performance, and course registrations. The modular approach ensures that each function can be developed, tested, and maintained independently, leading to a robust and scalable solution.

Future Enhancement

Future enhancements could include:

- Implementing a graphical user interface (GUI) for a better user experience.
- Adding authentication and authorization features to secure the system.
- Including a database for persistent storage.
- Adding more features such as report generation, bulk data import/export, and notifications.

References

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts.* McGraw-Hill Education.
2. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley.
3. Sebesta, R. W. (2012). *Concepts of Programming Languages.* Pearson Education.
4. Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management.* Pearson Education.
5. Ramamurthy, S., & Sen, A. (2017). *Java Programming: From Problem Analysis to Program Design.* Cengage Learning.
6. Yalcinalp, U., & Osborne, J. (2016). *JavaFX: A Beginner's Guide.* McGraw-Hill Education.
7. Sommerville, I. (2016). *Software Engineering.* Pearson Education.
8. Raj, P., Suresh, S., & Shantharajah, S. P. (2016). *Software Testing.* Oxford University Press.
9. Dikaiakos, M. D., Katsaros, D., Mehra, P., Pallis, G., & Vakali, A. (2009). *Cloud Computing: Distributed Internet Computing for IT and Scientific Research.* IEEE Computer Society.
10. Wu, H., Tan, C. W., & Zeng, D. (2020). *Big Data Privacy and Security.* CRC Press.