

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**Bus Ticket Reservation System**

**A CAPSTONE PROJECT REPORT**

*Submitted in the partial fulfillment for the completion of the course*

**CSA4307 INTERNET PROGRAMMING FOR CLIENT SERVER MODEL**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**B.Sudharshan reddy**

**192311192**

**Under the Supervision of**

**L.REETHA**

**MAY 2025**

**DECLARATION**

I, **B.Sudharshanreddy**, students of **Bachelor of Engineering in the Department** of Computer Science and Engineering [Data science] Saveetha Institute of Medical and Technical Sciences, Saveetha School of Engineering, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Bus Ticket Reservation 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.

B.Sudharshanreddy

192311192

Date:

Place:

**CERTIFICATE**

This is to certify that the project entitledConventional support system for on-line retail services submitted by **B.Sudharshanreddy** 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 and Engineering.

Supervisor

L.Reetha

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **PAGE NO.** |
| 1 | **Abstract** | **5** |
| 2 | **Introduction** | **5** |
| 3 | **Project Description**  About your project | **6** |
| 4 | **Problem Description**  Program to build a simple Software for < > | **8** |
| 5 | **Tool Description**  ****1. Front-End Technologies**** ****2. Back-End Technologies**** | **9** |
| 6 | **Operations** ****1.User Operations (Passenger Side)********2.Administrator Operations (Admin Side)**** | **12** |
| 7 | **Approach / Module Description / Functionalities** 1.  ****User Authentication Module**** ****2 .Bus and Route Management Module (Admin)**** ****3.Schedule and Seat Management Module**** | **13** |
| 8 | **Implementation**  Coding | **16** |
| 9 | **Result**  Output with Screenshots | **27** |
| 10 | **Conclusion**  Future Enhancement  **References** | **30** |

## ****ABSTRACT****

The Bus Ticket Reservation System is a web-based application designed to streamline the process of booking bus tickets for passengers while providing bus operators with an efficient platform to manage schedules, seat availability, and passenger records. The traditional method of reserving tickets manually is often time-consuming, error-prone, and inconvenient for both customers and service providers. This project aims to digitize and automate the entire process to enhance user experience and operational efficiency.

Developed using technologies such as PHP, MySQL, HTML, CSS, and JavaScript, the system enables users to view available routes, check seat availability in real time, book or cancel tickets, and receive digital confirmation. Administrators have access to modules for managing bus schedules, seat allocations, and customer information securely. The platform incorporates features such as login authentication, booking history tracking, and responsive design for mobile accessibility.

This project demonstrates how software solutions can significantly improve service delivery in the transportation sector. By automating ticket reservation, it reduces human workload, minimizes booking errors, and provides users with a convenient and accessible method to plan their travel.

.

****INTRODUCTION****

In today’s fast-paced digital world, automation has become an essential aspect of service delivery across various industries, including transportation. Traditional methods of booking bus tickets—such as standing in queues at bus terminals or relying on manual record-keeping—are increasingly being replaced by digital systems that offer convenience, accuracy, and efficiency. The Bus Ticket Reservation System is a modern solution aimed at addressing the limitations of conventional ticket booking processes.

This project presents a web-based application that allows passengers to reserve bus tickets online from the comfort of their homes. The system provides key functionalities such as viewing available buses, selecting travel routes, checking seat availability, booking tickets, and receiving instant confirmations. For administrators and bus operators, the system includes features to manage bus schedules, monitor seat allocations, and maintain passenger data, all within a secure and user-friendly interface.

Developed using tools such as **PHP**, **MySQL**, **HTML**, **CSS**, and **JavaScript**, and deployed via **XAMPP** on a local server environment, the platform ensures smooth interaction between users and the system. The project also focuses on core system attributes like authentication, data validation, and responsive design to ensure usability and security.

Overall, the Bus Ticket Reservation System aims to simplify the ticketing process for both passengers and bus service providers, reducing human errors, saving time, and promoting a more efficient and modern travel experience.

### ****2. Project Description****

The **Bus Ticket Reservation System** is a web-based application designed to automate the process of booking bus tickets and managing transportation services. It serves two main user groups: **passengers** and **administrators**. Passengers can book tickets online, while administrators can manage buses, routes, schedules, and bookings. The system is built to eliminate the drawbacks of the traditional manual system, such as long queues, human error, and lack of real-time information.

#### ****2.1 Purpose of the Project****

The main purpose of this project is to provide a convenient, secure, and efficient platform for passengers to book bus tickets anytime and from anywhere, and for transport operators to manage and monitor their services digitally.

#### ****2.2 Features****

The system offers the following core features:

**User Registration and Login**  
Secure authentication system for passengers and admin users.

**Bus Route and Schedule Management**  
Admins can add, update, and delete routes, bus schedules, and seat configurations.

**Seat Selection and Real-Time Availability**  
Passengers can view seat layouts and book available seats.

**Online Ticket Booking and Cancellation**  
Users can book and cancel tickets with real-time updates.

**Booking History**  
Passengers can view their booking history and ticket details.

**Admin Dashboard**  
Admins can monitor all bookings, manage buses and routes, and generate basic reports.

#### ****2.3 Technologies Used****

**Front-End**: HTML, CSS, JavaScript

**Back-End**: PHP

**Database**: MySQL

**Server Environment**: XAMPP (Apache, MySQL, PHP, Perl)

#### ****2.4 Benefits****

Reduces manual work and errors in booking.

Saves time and effort for both passengers and bus operators.

Enables passengers to plan and book their travel remotely.

Helps operators manage resources more effectively through automation.

#### ****2.5 Target Users****

General public/passengers using bus transport.

Bus company staff/admins managing routes and bookings.

## ****PROBLEM DESCRIPTION****

In many regions, the process of booking bus tickets is still done manually, either through in-person counters or over the phone. This traditional approach is often inefficient, time-consuming, and prone to human error. Passengers are required to travel to booking offices, wait in long queues, and rely on handwritten receipts or verbal confirmations, which can lead to miscommunications, overbooking, and poor customer experience.

Bus operators, on the other hand, face challenges in managing passenger data, monitoring seat availability in real time, and maintaining accurate records of bookings, cancellations, and schedules. Without a centralized digital system, there is a higher risk of lost records, double bookings, and difficulty in tracking overall business performance.

### ****4. Project Objectives****

The main objective of the **Bus Ticket Reservation System** is to develop an efficient, user-friendly, and secure web-based application that simplifies the ticket booking process for passengers and provides administrative tools for managing bus operations. The system aims to bridge the gap between traditional booking methods and modern digital solutions.

#### ****4.1 General Objective****

To design and implement a web-based bus ticket reservation system that allows passengers to book tickets online and enables administrators to manage bus schedules, seat availability, and passenger data efficiently.

#### ****4.2 Specific Objectives****

To provide a **user registration and login** system for secure access.

To allow passengers to **search for buses**, view schedules, and check **seat availability** in real time.

To enable users to **book, view, and cancel tickets** through an interactive and responsive interface.

To develop an **admin dashboard** for managing routes, buses, schedules, and booking records.

To ensure **data security and integrity** through authentication and validation mechanisms.

To generate **booking history and reports** for both users and administrators.

To create a system that is **responsive and accessible** on multiple devices (e.g., desktops, tablets, and smartphones).

To reduce manual workload and eliminate common errors associated with traditional booking systems.

### ****5. Development Tools****

The development of the **Bus Ticket Reservation System** involved the use of various software tools and technologies to ensure a responsive, efficient, and secure web-based application. The tools were selected based on compatibility, ease of use, and effectiveness in achieving the project objectives.

#### ****5.1 Front-End Technologies****

**HTML5** – Used to structure the content and layout of web pages.

**CSS3** – Applied for styling the application, ensuring a modern and user-friendly interface.

**JavaScript** – Used for client-side interactivity and validation.

#### ****5.2 Back-End Technologies****

**PHP** – Server-side scripting language used for developing dynamic functionalities such as login, booking logic, and admin panel.

**MySQL** – Relational database management system used to store user data, booking records, bus schedules, and routes.

#### ****5.3 Development Environment****

**XAMPP** – An open-source platform that includes Apache, MySQL, PHP, and Perl, used to create a local server environment for developing and testing the application.

#### ****5.4 Code Editor****

**Visual Studio Code** – A lightweight and powerful source code editor used for writing and managing code files with support for PHP, HTML, CSS, and JavaScript.

#### ****5.5 Design and Testing Tools****

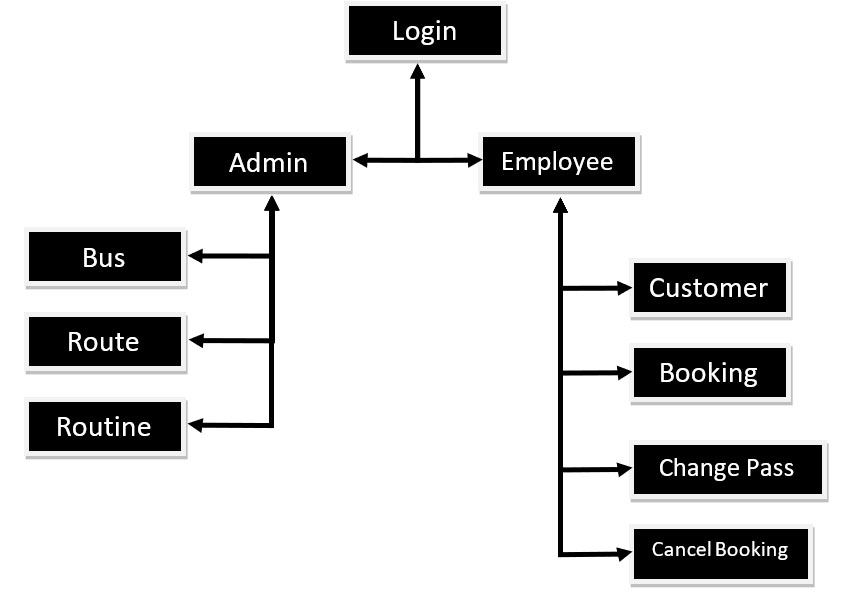
**Figma / Adobe XD (optional)** – Used for UI/UX design mockups and wireframes (if applicable).

**Browser Developer Tools** – Utilized for debugging, inspecting, and testing front-end behavior across different browsers.

#### ****5.6 Version Control (optional)****

**Git** – For tracking changes and collaborating during development (if used).

**BLOCK DIAGRAM:**



### ****6. Operations****

The **Bus Ticket Reservation System** performs a series of operations that facilitate seamless interaction between users (passengers and administrators) and the system. These operations are categorized into user-side functions and administrator-side functions, each playing a crucial role in ensuring efficient ticket booking and management.

#### ****6.1 User Operations (Passenger Side)****

**User Registration & Login**  
Users can create an account with personal details and securely log in to access the booking system.

**Search Buses and Routes**  
Passengers can search for available buses by selecting source, destination, and date of travel.

**View Schedule and Seat Availability**  
The system displays the bus schedule and current seat availability in real time.

**Book Tickets**  
Users can select preferred seats and confirm their booking. A digital ticket or booking ID is generated.

**Cancel Bookings**  
Users can cancel previously booked tickets, and the seat status is updated accordingly.

**View Booking History**  
Users can view their past and upcoming bookings, including ticket details and status.

#### ****6.2 Administrator Operations (Admin Side)****

**Admin Login**  
Secure login access is provided for system administrators.

**Manage Bus Information**  
Admins can add, update, or delete details of buses, including bus number, type, and capacity.

**Manage Routes and Schedules**  
Admins can define travel routes, set departure and arrival times, and assign buses to routes.

**Monitor Bookings and Seat Status**  
Real-time monitoring of booked and available seats for each bus is provided.

**Manage Users**  
Admins can view registered users, and if needed, suspend or delete accounts.

**Generate Reports**  
Admins can generate reports on ticket bookings, cancellations, and overall system usage.

### ****7. Module Description****

The **Bus Ticket Reservation System** is divided into several functional modules, each responsible for handling specific tasks within the system. These modules work together to deliver a seamless experience for both passengers and administrators.

#### ****7.1 User Authentication Module****

**Purpose**: Handles user registration and login functionalities.

**Functions**:

Register new users with basic personal details.

Validate login credentials.

Manage user sessions and logout.

#### ****7.2 Bus and Route Management Module (Admin)****

**Purpose**: Enables administrators to add and manage bus details and travel routes.

**Functions**:

Add/edit/delete buses.

Define routes with source, destination, and timing.

Assign buses to specific routes.

#### ****7.3 Schedule and Seat Management Module****

**Purpose**: Manages travel schedules and seat availability for each trip.

**Functions**:

Define depar ture and arrival times.

Show available seats in real time.

Update seat status upon booking or cancellation.

#### ****7.4 Ticket Booking Module****

**Purpose**: Allows users to book tickets based on selected route, date, and seat.

**Functions**:

Display list of available buses and seats.

Allow passengers to select a seat and confirm the booking.

Generate a booking ID or digital ticket.

#### ****7.5 Booking Cancellation Module****

**Purpose**: Enables users to cancel existing bookings.

**Functions**:

Show list of active bookings.

Allow users to cancel and release the seat for future bookings.

Update the booking status and reflect changes in the seat layout.

#### ****7.6 Booking History Module****

**Purpose**: Displays booking records for users and admins.

**Functions**:

Show history of completed, cancelled, and upcoming bookings.

Allow filtering by date or status.

Provide details like seat number, route, and fare.

#### ****7.7 Admin Dashboard Module****

**Purpose**: Provides a central control panel for the administrator.

**Functions**:

View system statistics (total users, bookings, routes).

Manage buses, routes, users, and schedules.

Generate simple reports on system usage.

**IMPLEMENTATION**

**auth.js:**

// auth.js

function saveUserDetails(user) {

  let users = JSON.parse(localStorage.getItem('users')) || [];

  users.push(user);

  localStorage.setItem('users', JSON.stringify(users));

}

function validateUser(loginDetails) {

  let users = JSON.parse(localStorage.getItem('users')) || [];

  return users.some(user => user.email === loginDetails.email && user.password === loginDetails.password);

}

function signup(event) {

  event.preventDefault();

  const name = document.getElementById('name').value;

  const email = document.getElementById('email').value;

  const password = document.getElementById('password').value;

  const user = { name, email, password };

  saveUserDetails(user);

  alert('Signup successful! You can now login.');

  window.location.href = 'login.html';

}

function login(event) {

  event.preventDefault();

  const email = document.getElementById('email').value;

  const password = document.getElementById('password').value;

  const loginDetails = { email, password };

  if (validateUser(loginDetails)) {

      alert('Login successful!');

      window.location.href = 'bus.html';

  } else {

      alert('Invalid login credentials!');

  }

}

**book-details.js**

function getBookingDetails() {

  return {

    from: localStorage.getItem('from'),

    to: localStorage.getItem('to'),

    date: localStorage.getItem('date'),

    busName: localStorage.getItem('busName'),

    startTime: localStorage.getItem('startTime'),

    endTime: localStorage.getItem('endTime'),

    price: localStorage.getItem('price') + 'rs',

    seats: JSON.parse(localStorage.getItem('selectedSeats')).join(', '),

    name: localStorage.getItem('name'),

    age: localStorage.getItem('age'),

    gender: localStorage.getItem('gender'),

    mobile: localStorage.getItem('mobile')

  };

}

function displayBookingDetails() {

  const details = getBookingDetails();

  const bookingDetailsDiv = document.getElementById('bookingDetails');

  bookingDetailsDiv.innerHTML = `

    <p><strong>From:</strong> ${details.from}</p>

    <p><strong>To:</strong> ${details.to}</p>

    <p><strong>Date:</strong> ${details.date}</p>

    <p><strong>Bus Name:</strong> ${details.busName}</p>

    <p><strong>Time:</strong> ${details.startTime} - ${details.endTime}</p>

    <p><strong>Price:</strong> ${details.price}</p>

    <p><strong>Seats:</strong> ${details.seats}</p>

    <p><strong>Name:</strong> ${details.name}</p>

    <p><strong>Age:</strong> ${details.age}</p>

    <p><strong>Gender:</strong> ${details.gender}</p>

    <p><strong>Mobile:</strong> ${details.mobile}</p>

  `;

}

// Display the booking details on page load

window.onload = displayBookingDetails;

book.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Booking Details</title>

    <link rel="stylesheet" href="style.css" />

    <script src="book-details.js" defer></script>

  </head>

  <body>

    <div class="content-container">

      <h1>Booking Details</h1>

      <div id="bookingDetails"></div>

    </div>

  </body>

</html>

bus.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Online Bus Booking - Home</title>

    <link rel="stylesheet" href="style.css" />

    <script src="scripts.js" defer></script>

  </head>

  <body>

    <form id="bookingForm">

      <h1>Online Bus Booking</h1>

      <div class="nam">

        <label style="font-weight: bold" for="from">From</label>

        <input type="text" name="from" id="from" required />

      </div>

      <div id="too">

        <label style="font-weight: bold" for="to">To</label>

        <input type="text" name="to" id="to" required />

      </div>

      <div id="dat">

        <label style="font-weight: bold" for="date">Date</label>

        <input type="date" name="date" id="date" required />

      </div>

      <button

        class="bus-booking-btn"

        type="button"

        onclick="navigateToNextPage()"

      >

        Book

      </button>

    </form>

    <main></main>

  </body>

</html>

login.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Login</title>

    <link rel="stylesheet" href="styles.css" />

    <script src="auth.js" defer></script>

  </head>

  <body>

    <form id="loginForm" onsubmit="login(event)">

      <h1>Login</h1>

      <label for="email">Email:</label>

      <input type="email" id="email" required />

      <label for="password">Password:</label>

      <input type="password" id="password" required />

      <button type="submit">Login</button>

      <p>Don't have an account? <a href="signup.html">Sign Up</a></p>

    </form>

  </body>

</html>

next page.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Select Bus</title>

    <link rel="stylesheet" href="style.css" />

    <script src="next-page.js" defer></script>

  </head>

  <body>

    <div class="content-container">

      <h1>Select Your Bus</h1>

      <div class="bus-options">

        <button

          class="bus-option"

          onclick="selectBus('SETC', '08:00', '12:00', '200')"

        >

          <div><h4>SETC</h4></div>

          <div>08:00 - 12:00</div>

          <div>Price: 200rs</div>

        </button>

        <button

          class="bus-option"

          onclick="selectBus('Amaravathi', '09:00', '13:00', '250')"

        >

          <div><h4>Amaravathi</h4></div>

          <div>09:00 - 13:00</div>

          <div>Price: 250rs</div>

        </button>

        <button

          class="bus-option"

          onclick="selectBus('Airavat', '10:00', '14:00', '300')"

        >

          <div><h4>Airavat</h4></div>

          <div>10:00 - 14:00</div>

          <div>Price: 300rs</div>

        </button>

        <button

          class="bus-option"

          onclick="selectBus('Nue GO', '11:00', '15:00', '350')"

        >

          <div><h4>Nue GO</h4></div>

          <div>11:00 - 15:00</div>

          <div>Price: 350rs</div>

        </button>

      </div>

    </div>

  </body>

</html>

next-page.js

function selectBus(name, startTime, endTime, price) {

  // Save bus details to localStorage

  localStorage.setItem('busName', name);

  localStorage.setItem('startTime', startTime);

  localStorage.setItem('endTime', endTime);

  localStorage.setItem('price', price);

  // Redirect to seat selection page

  window.location.href = 'seat-selection.html';

}

Scripts.js

function navigateToNextPage() {

    // Prevent form submission

    event.preventDefault();

    // Get form data

    const from = document.getElementById('from').value;

    const to = document.getElementById('to').value;

    const date = document.getElementById('date').value;

    // Save data to localStorage

    localStorage.setItem('from', from);

    localStorage.setItem('to', to);

    localStorage.setItem('date', date);

    // Redirect to next page

    window.location.href = 'next-page.html';

  }

seat-selection.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Select Seats</title>

    <link rel="stylesheet" href="style.css" />

    <script src="seat-selection.js" defer></script>

  </head>

  <body>

    <div class="content-container">

      <h1>Select Your Seats</h1>

      <div class="seats-container" id="seatsContainer">

        <!-- Seats will be generated here by JavaScript -->

      </div>

      <button class="bus-booking-btn" onclick="confirmSeats()">

        Confirm Seats

      </button>

    </div>

  </body>

</html>

seat-selection.js

document.addEventListener('DOMContentLoaded', () => {

    const seatsContainer = document.getElementById('seatsContainer');

    // Generate seats

    for (let row = 1; row < 10; row++) {

      const rowDiv = document.createElement('div');

      rowDiv.className = 'seat-row';

      const rowLetter = String.fromCharCode(64 + row); // Convert row number to letter (A, B, C, ...)

      for (let col = 1; col <= 4; col++) {

        const seatNumber = `${rowLetter}${col}`;

        const seat = document.createElement('button');

        seat.className = 'seat';

        seat.textContent = seatNumber;

        seat.onclick = () => toggleSeatSelection(seat);

        // Add seat to the row

        rowDiv.appendChild(seat);

        // Add a spacer between the two columns

        if (col === 2) {

          const spacer = document.createElement('div');

          spacer.className = 'seat-spacer';

          rowDiv.appendChild(spacer);

        }

      }

      // Add row to the container

      seatsContainer.appendChild(rowDiv);

    }

  });

  a

  function toggleSeatSelection(seat) {

    seat.classList.toggle('selected');

  }

  function confirmSeats() {

    const selectedSeats = [];

    document.querySelectorAll('.seat.selected').forEach(seat => {

      selectedSeats.push(seat.textContent);

    });

    if (selectedSeats.length > 0) {

      // Save selected seats to localStorage

      localStorage.setItem('selectedSeats', JSON.stringify(selectedSeats));

      // Redirect to user details page

      window.location.href = 'user-details.html';

    } else {

      alert('Please select at least one seat.');

    }

  }

Signup.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Sign Up</title>

    <link rel="stylesheet" href="styles.css" />

    <script src="auth.js" defer></script>

  </head>

  <body>

    <form id="signupForm" onsubmit="signup(event)">

      <h1>Sign Up</h1>

      <label for="name">Name:</label>

      <input type="text" id="name" required />

      <label for="email">Email:</label>

      <input type="email" id="email" required />

      <label for="password">Password:</label>

      <input type="password" id="password" required />

      <button type="submit">Sign Up</button>

    </form>

  </body>

</html>

style.css

/\* styles.css \*/

/\* General styles for the body \*/

body {

    background-image: url('aa.jpg');

    background-size: cover;

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-blend-mode: color-dodge;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

  }

  /\* Form container styles \*/

  form, .content-container {

    background-blend-mode: color-burn;

    padding: 20px;

    border-radius: 8px;

    box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);

    width: 300px;

    text-align: center;

  }

  /\* Header styles \*/

  h1 {

    text-align: center;

    color: #333;

    margin-bottom: 10px;

  }

  /\* Label styles \*/

  label {

    display: block;

    margin-bottom: 5px;

    color: #555;

  }

  /\* Input field styles \*/

  input[type="text"], input[type="date"] {

    width: calc(100% - 20px);

    padding: 8px 10px;

    margin-bottom: 15px;

    border: 1px solid #ccc;

    border-radius: 4px;

  }

  /\* Button styles \*/

  button, .bus-booking-btn {

    background-color: #007bff;

    color: #fff;

    border: none;

    padding: 10px 15px;

    border-radius: 4px;

    cursor: pointer;

    width: calc(100% - 20px);

    margin: 10px 0;

  }

  button:hover, .bus-booking-btn:hover {

    background-color: #0056b3;

  }

  /\* Styles for specific elements \*/

  .nam, #too, #dat {

    margin-bottom: 15px;

  }

  /\* Bus option button styles \*/

  .bus-option {

    display: block;

    background-color: bisque;

    padding: 3px;

    border: 1px solid #ccc;

    border-radius: 5px;

    margin-bottom: 10px;

    text-align: left;

    cursor: pointer;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

  }

  .bus-option:hover {

    background-color: #e9f5ff;

  }

  .bus-option div {

    color: black;

    margin-bottom: 5px;

  }

  .bus-options {

    display: flex;

    flex-direction: column;

    align-items: center;

  }

  #bookingDetails p {

    margin: 10px 0;

    font-weight: bold;

    color: black;

  }

  .content-container {

    text-align: center;

    margin-top: 20px;

  }

  .seats-container {

    display: inline-block;

  }

  .seat-row {

    display: flex;

    justify-content: center;

    margin-bottom: 10px;

  }

  .content-container {

    align-self: baseline;

    text-align: center;

    margin-top: 20px;

  }

  .seats-container {

    display: inline-block;

  }

  .seat-row {

    display: flex;

    justify-content: center;

    margin-bottom: 10px;

  }

  .seat {

    width: 40px;

    color: black;

    height: 40px;

    margin: 5px;

    background-color: #f1f1f1;

    border: 1px solid #ccc;

    border-radius: 5px;

    cursor: pointer;

  }

  .seat.selected {

    background-color: #4CAF50;

    color: white;

  }

  .seat-spacer {

    width: 40px;

    height: 40px;

    margin: 5px;

    background-color: transparent;

  }

  .form-group {

    margin-bottom: 15px;

  }

  .form-group label {

    color: black;

    display: block;

    font-weight: bold;

  }

  .form-group input,

  .form-group select {

    width: 100%;

    padding: 8px;

    margin-top: 5px;

    box-sizing: border-box;

  }

  .bus-booking-btn {

    padding: 10px 20px;

    background-color: #4CAF50;

    color: white;

    cursor: pointer;

  }

  .bus-booking-btn:hover {

    background-color: #45a049;

  }

styles.css

body {

  background-image: url('aa.jpg');

    background-size: cover;

  font-family: Arial, sans-serif;

  display: flex;

  justify-content: center;

  align-items: center;

  height: 100vh;

  margin: 0;

  background-color: #f0f0f0;

}

form {

  background: #fff;

  padding: 20px;

  border-radius: 5px;

  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1 {

  margin-bottom: 20px;

}

label {

  display: block;

  margin-bottom: 5px;

}

input {

  width: 100%;

  padding: 10px;

  margin-bottom: 10px;

  border: 1px solid #ccc;

  border-radius: 3px;

  box-sizing: border-box;

}

button {

  width: 100%;

  padding: 10px;

  background: #007bff;

  border: none;

  border-radius: 3px;

  color: white;

  font-size: 16px;

  cursor: pointer;

}

button:hover {

  background: #0056b3;

}

User-details.html

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>User Details</title>

    <link rel="stylesheet" href="style.css" />

    <script src="user-details.js" defer></script>

  </head>

  <body>

    <div class="content-container">

      <h1>Enter Your Details</h1>

      <form id="userDetailsForm">

        <div class="form-group">

          <label for="name">Name</label>

          <input type="text" id="name" name="name" required />

        </div>

        <div class="form-group">

          <label for="age">Age</label>

          <input type="number" id="age" name="age" required />

        </div>

        <div class="form-group">

          <label for="gender">Gender</label>

          <select id="gender" name="gender" required>

            <option value="">Select</option>

            <option value="Male">Male</option>

            <option value="Female">Female</option>

            <option value="Other">Other</option>

          </select>

        </div>

        <div class="form-group">

          <label for="mobile">Mobile Number</label>

          <input type="tel" id="mobile" name="mobile" required />

        </div>

        <button type="button" onclick="saveUserDetails()">Submit</button>

      </form>

    </div>

  </body>

</html>

User-details.js

function saveUserDetails() {

    const name = document.getElementById('name').value;

    const age = document.getElementById('age').value;

    const gender = document.getElementById('gender').value;

    const mobile = document.getElementById('mobile').value;

    // Save user details to localStorage

    localStorage.setItem('name', name);

    localStorage.setItem('age', age);

    localStorage.setItem('gender', gender);

    localStorage.setItem('mobile', mobile);

    // Redirect to booking details page

    window.location.href = 'book.html';

  }

**RESULT:**

**Figure 1:**

****

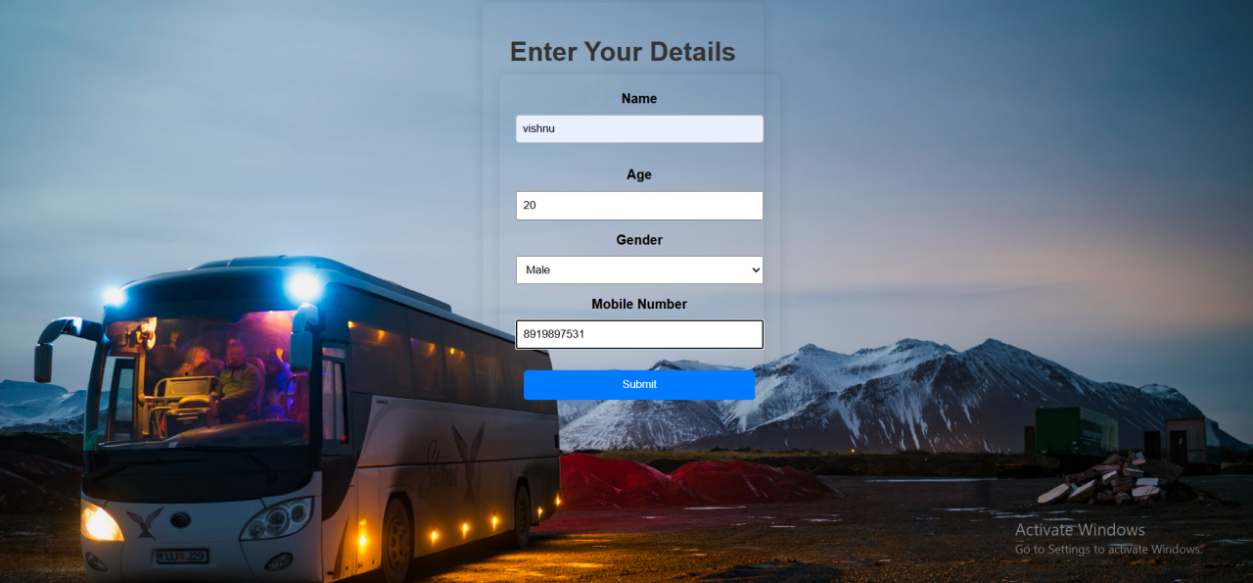
**Figure 2:**

****

**Figure 3:**

****

**Figure 4:**

****

**Figure 5:**

****

**CONCLUSION**

The **Bus Ticket Reservation System** project successfully demonstrates how technology can streamline and enhance the ticketing process for both passengers and bus operators. By automating the booking, cancellation, and management of bus tickets, the system provides a more efficient, transparent, and user-friendly alternative to traditional methods. Passengers can now book tickets from anywhere at any time, view real-time seat availability, and manage their travel plans with ease. For administrators, the system simplifies operations such as managing routes, schedules, and bookings, reducing the risk of errors and improving overall service efficiency.

The system has been developed with a focus on security, usability, and scalability, ensuring that it can accommodate future growth and evolving user needs. The use of a web-based interface allows for easy access from a variety of devices, ensuring maximum accessibility for users. Additionally, by integrating key features such as seat availability, booking history, and real-time updates, the system enhances the customer experience and reduces the operational load for bus operators.

While the system addresses the immediate needs of bus ticket reservation and management, it also lays the groundwork for future improvements. These could include the integration of payment gateways, mobile applications, and advanced analytics for route optimization and demand forecasting.

### ****8. References****

**1.W3Schools**. (n.d.). HTML, CSS, JavaScript, and PHP Tutorials. Retrieved from [https://www.w3schools.com](https://www.w3schools.com" \t "_new)

**2.PHP Manual**. (n.d.). PHP Documentation. Retrieved from [https://www.php.net/manual/en/](https://www.php.net/manual/en/" \t "_new)

**3.MySQL Documentation**. (n.d.). MySQL Reference Manual. Retrieved from [https://dev.mysql.com/doc/](https://dev.mysql.com/doc/" \t "_new)

**4.XAMPP**. (n.d.). XAMPP for Windows. Retrieved from https://www.apachefriends.org/index.html

**5.MDN Web Docs**. (n.d.). JavaScript and Web Development Resources. Retrieved from [https://developer.mozilla.org](https://developer.mozilla.org" \t "_new)

**6.Fritz, B.** (2020). Building Web Applications with PHP and MySQL: An Introduction. Wiley & Sons.

**7.Smith, J. & Taylor, R.** (2019). Web Development Essentials: Tools and Technologies for Modern Websites. Pearson Education.

**8.WebAIM**. (2021). Web Accessibility in Mind. Retrieved from [https://webaim.org](https://webaim.org" \t "_new)

**9.GitHub**. (n.d.). Version Control with Git. Retrieved from [https://github.com](https://github.com" \t "_new)

**10.Figma**. (n.d.). Design and Prototyping for Web and Mobile Applications. Retrieved from [https://www.figma.com](https://www.figma.com" \t "_new)