ANJALAI AMMAL MAHALINGAM
ENGINEERING COLLEGE,
KOVILVENNI-614 403, THIRUVARUR (DT)

ELECTRONICS AND COMMUNICATION ENGINEERING

BUS TICKET BOOKING SYSTEM

A PROJECT REPORT

Submitted by

DHARSHIKA.R **820421106010**

MANGAIYARKARASI.M **820421106028**

NM1050- SOFTWARE AS A SERVICE

BONAFIDE CERTIFICATE

Certified that this project report "**BUS TICKET BOOKING SYSTEM**" is the bonafide work of **DHARSHIKA.R & MANGAIYARKARASI.M** who carried out the project work under my supervision.

SIGNATURE

Dr. A. Bhavani Sankar,M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFFESSOR,

DEPARTMENT OF ECE,

Anjalai Ammal Mahalingam

Engineering College,

Kovilvenni-614 403,

Thiruvarur(Dt)

SIGNATURE

G.Murugesan M.E.,

SUPERVISOR

ASSOCIATE PROFFESSOR,

DEPARTMENT OF ECE,

Anjalai Ammal Mahalingam

Engineering College,

Kovilvenni-614 403,

Thiruvarur(Dt)

Examined on: ……...

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

S.No	Content	Page No
1.	Abstract	4
2.	Introduction	5
3.	About SAAS	6
4.	Project Description	8
5.	Coding	12
6.	Result	24
7.	Conclusion	25
8.	Reference	26

ABSTRACT

This project, Bus Ticket Booking System, is a web-based application designed to facilitate convenient and efficient booking of bus tickets for users. Built using Flask, a Python web framework, and MySQL as the backend database, the system enables users to book bus tickets, manage travel details, and view bookings.

The system consists of two main components

1. **Booking Interface:** A user-friendly form where users can input their personal details, select travel options (starting point, destination, departure, and return dates), and choose their preferred payment method. Successful bookings are stored in the database.
2. **View Bookings:** A page displaying a tabular view of all the bookings, fetched dynamically from the database. This feature is primarily designed for administrators or users who wish to review booking data.

Key functionalities include

- Secure data storage in a MySQL database with validation for user inputs.
- Dynamic rendering of templates using Jinja2, ensuring an interactive and responsive user experience.
- Seamless integration of front-end and back-end technologies to provide a robust and scalable solution.

The application is styled with a modern, intuitive UI and includes features for ease of navigation, such as links for switching between booking and viewing pages. This project serves as a foundational system for travel booking automation and can be further expanded to include features like user authentication, real-time seat availability, and payment gateway integration.

1.INTRODUCTION:

The **Bus Ticket Booking System** is a web-based application aimed at simplifying the process of booking bus tickets for travelers. This system bridges the gap between users and transport services by providing an easy-to-use platform for ticket reservations. It allows users to enter their personal details, select their journey options, and make payments efficiently, eliminating the need for traditional, time-consuming methods.

This application is built using **Flask**, a Python web framework, which handles the backend logic, and **MySQL** as the database for securely storing user and booking information. The front-end utilizes HTML, CSS, and responsive design principles to deliver a modern and intuitive user interface.

1.1.Key Features of the System:

1. **Booking Functionality**: Users can provide their details, choose starting and destination points, select travel dates, and pick a payment method through a well-structured form.
2. **Database Integration**: Bookings are stored in a **MySQL database**, ensuring data persistence and security.
3. **View Bookings**: A dedicated page displays a comprehensive table of all bookings, allowing users and administrators to review booking records.

This project not only streamlines the booking process for users but also serves as a foundational model for more advanced travel systems. It can be extended with additional features such as seat selection, user authentication, real-time seat availability, and payment gateway integration. The Bus Ticket Booking System aims to enhance user convenience while maintaining data accuracy and system reliability.

2.ABOUT SAAS:

Software as a Service (SaaS) is a cloud-based software delivery model where applications are hosted by a service provider or vendor and made available to users over the Internet. Instead of purchasing, installing, and maintaining software on individual computers or servers, users can access these applications through web browsers, paying for them on a subscription or usage basis.

2.1.Key Features of SaaS:

2.1.1.Cloud-Based Access:

SaaS applications are hosted on cloud servers, enabling users to access them from anywhere with an internet connection.No need for local installations or complex IT setups.

2.1.2.Subscription Model:

Users typically pay a recurring fee (monthly or yearly), making it a cost-effective solution.Reduces upfront costs associated with traditional software licensing.

2.1.3.Scalability:

SaaS solutions can scale easily to accommodate growing business needs.Service providers manage the infrastructure, ensuring performance and reliability.

2.1.4.Automatic Updates:

Software updates and patches are handled by the provider, ensuring users always have the latest version.

2.1.5.Multi-Tenancy:

Multiple users or organizations (tenants) share the same infrastructure, but their data and configurations are securely isolated.

2.2.Advantages of SaaS:

1.Cost Savings: Eliminates the need for purchasing hardware or maintaining infrastructure.

2.Ease of Use: Minimal setup required, with user-friendly interfaces designed for quick adoption.

3.Flexibility: Accessible from multiple devices, supporting remote work and collaboration.

4.Maintenance-Free: Service providers handle backend operations like updates, security, and performance optimization.

2.3.Common Use Cases of SaaS:

1.Business Applications: CRM (e.g., Salesforce), HR management (e.g., Workday), and project management (e.g., Asana).

2.Productivity Tools: Google Workspace (Docs, Sheets) and Microsoft 365.

3.E-Commerce: Shopify, BigCommerce.

4.Streaming Services: Netflix, Spotify.

5.Communication Platforms: Zoom, Slack.

2.4.Challenges of SaaS:

1.Data Security: Reliance on third-party vendors raises concerns about data privacy and security.

2.Internet Dependency: Requires a stable internet connection to access applications.

3.Limited Customization: Off-the-shelf solutions may not meet highly specific business needs.

SaaS has become a dominant model in the software industry, enabling businesses and individuals to focus on leveraging technology without the overhead of managing complex IT infrastructure. Its rapid adoption is driven by its efficiency, cost-effectiveness, and adaptability in today's digital-first world.

3.PROJECT DESCRIPTION:

BUS TICKET BOOKING SYSTEM

3.1.Overview:

The Bus Ticket Booking System is a web-based application designed to simplify and streamline the process of booking bus tickets online. This project leverages the Flask web framework and MySQL database to create an efficient platform for managing ticket reservations and viewing existing bookings. The system allows users to book tickets by entering relevant travel details and provides an intuitive interface for both booking and viewing reservations.

3.2.Key Features:

3.2.1.User-Friendly Interface:

A responsive and visually appealing booking form for users to enter travel details such as:

1.Name

2.Email

3.Starting Point

4.Destination

5.Departure and Return Dates

6.Payment Method

3.2.2.Data Storage and Management:

All booking information is securely stored in a MySQL database with clearly defined fields for efficient data retrieval and management.

3.2.3.View Bookings:

A dedicated page to display all bookings in a structured tabular format, allowing users to review previously made reservations.

3.2.4.Dynamic Feedback:

Upon successful booking, users receive a confirmation message, ensuring clear communication and usability.

3.2.5.Navigation Links:

Links for navigating between the booking form and the bookings view page, ensuring a seamless user experience.

3.3.Technical Components:

3.3.1.Frontend:

1.Developed using HTML, CSS, and Jinja templates for dynamic rendering of booking forms and database results.

2.Responsive design elements ensure compatibility across devices.

3.3.2.Backend:

1.Built with Python and Flask to handle HTTP requests, form submissions, and dynamic rendering of templates.

2.Routes for handling booking submissions and viewing all reservations.

3.3.2.Database:

1.MySQL is used to store user information and booking details.

2.The bookings table includes fields such as:

1.ID (Primary Key)

2.Name

3.Email

4.Starting Point

5.Destination

6.Departure Date

7.Return Date

8.Payment Method

3.4.Workflow:

3.4.1.Booking a Ticket:

1.Users access the main page, fill out the booking form, and submit their details.

2.Data is validated and stored in the database.

3.A confirmation message is displayed.

3.4.2.Viewing Bookings:

1.Users navigate to the "View All Bookings" page to see a list of all bookings stored in the database.

2.Each booking is displayed with details such as ID, Name, Email, and Travel Information.

3.5.Potential Use Cases:

1.Individual Users: Travelers can book bus tickets online without visiting physical counters.

2.Bus Operators: Operators can use the system to manage reservations and analyze travel trends.

3.Integration: Can be expanded to include additional features like seat selection, ticket cancellation, or integration with payment gateways.

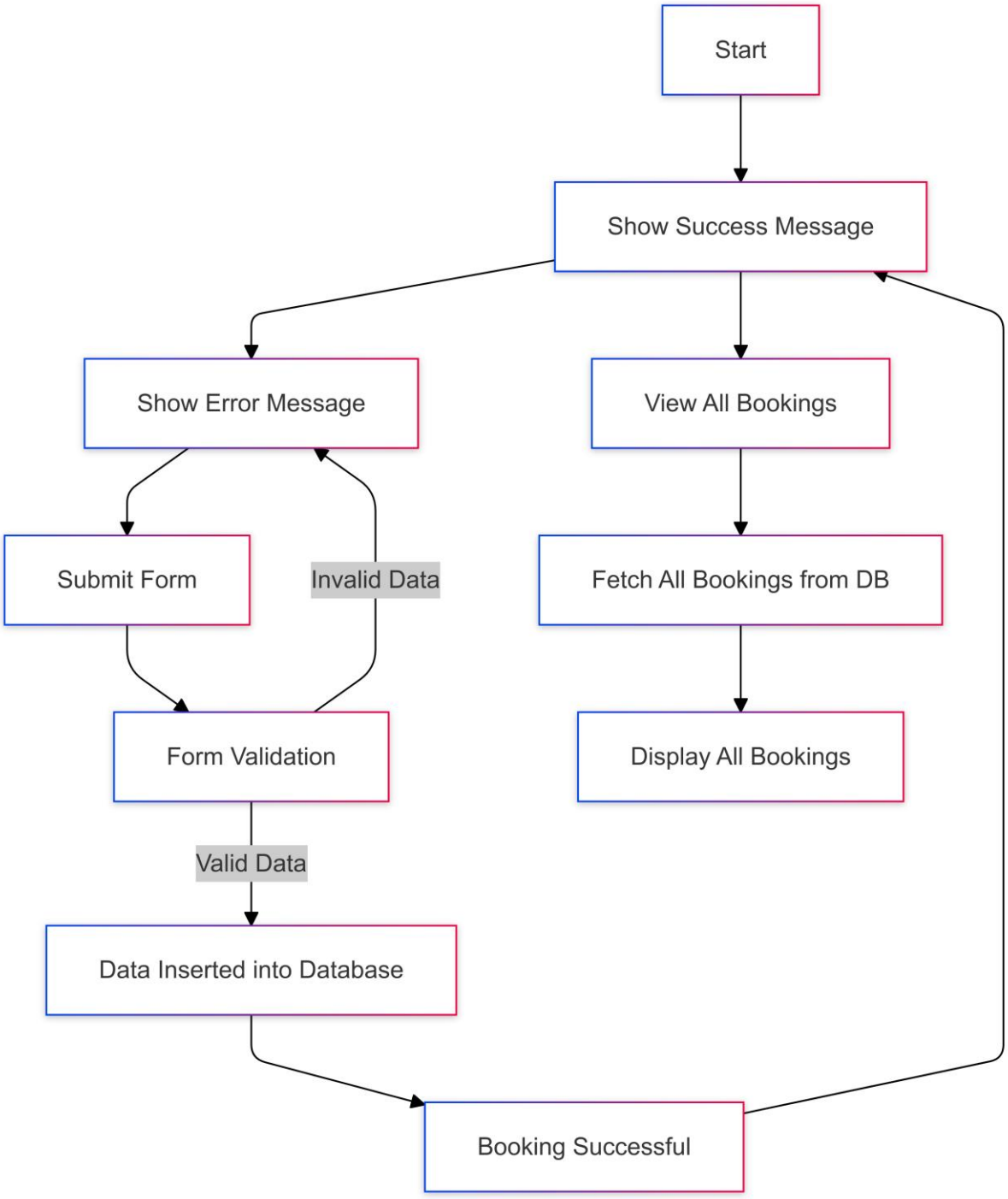
3.6.Benefits:

1.Simplifies the booking process for travelers.

2.Reduces manual errors in data entry and reservation handling.

3.Provides centralized data management for bus operators.

3.7.FLOW CHART:



3.8.Future Enhancements:

1. Integration with third-party payment gateways for secure online payments.
2. Adding seat selection and availability checking functionality.
3. User authentication to enable personalized booking history.
4. Mobile-friendly app development for broader accessibility.

This project serves as an essential tool for modernizing traditional bus ticket booking systems, offering convenience to travelers and efficiency to bus operators.

4.CODING:

4.1.Project Structure:

NM project/

```
├── templates/
        └── booking_form.html
        └── view_bookings.html
├── static/
        └── style.css
        └── background image
├── app.py
```

4.2.Templates:

4.2.1.Booking_Form.html

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bus Ticket Booking</title>
  <link rel="stylesheet" href="../static/style.css">
</head>
<body>
    <div class="background">
        <div class="booking-form">
            <h2>Bus Ticket Booking</h2>
            <form action="/" method="post">
                <label for="name" >Name:</label>
                <input type="text" name="name" id="name" required
placeholder="Enter Your Name" >

                <label for="email">Email:</label>
                <input type="email" name="email" id="email" required
placeholder="Enter Your Email">

                <label for="starting-point">Starting Point:</label>
                <select name="starting-point" id="starting-point" required>
                    <option value="" disabled selected>Select your starting
point</option>
                    <option value="Thiruvarur">Thiruvarur</option>
```

```
            <option value="Thanjavur">Thanjavur</option>
            <option value="Trichy">Trichy</option>
            <option value="Nagapattinam">Nagapattinam</option>
            <option value="Madurai">Madurai</option>
        </select>

        <label for="destination">Destination:</label>
        <select name="destination" id="destination" required>
            <option value="" disabled selected>Select your
destination</option>
            <option value="Chennai">Chennai</option>
            <option value="Coimbatore">Coimbatore</option>
            <option value="Bangalore">Bangalore</option>
            <option value="Kerala">Kerala</option>
            <option value="Hyderabad">Hyderabad</option>
        </select>

        <label for="departure-date">Departure Date:</label>
        <input type="date" name="departure-date" id="departure-date"
required>

        <label for="return-date">Return Date:</label>
        <input type="date" name="return-date" id="return-date" required>

        <label for="payment-method">Payment Method:</label>
```

```
            <select name="payment-method" id="payment-method" required>
                <option value="" disabled selected>Select your payment method</option>
                <option value="Credit Card">Credit Card</option>
                <option value="Debit Card">Debit Card</option>
                <option value="Net Banking">Net Banking</option>
                <option value="UPI">UPI</option>
                <option value="Cash">Cash</option>
            </select>

            <button type="submit">Book Now</button>
        </form>
        {% if message %}
            <p>{{ message }}</p>
        {% endif %}
    </div>
</div>
<center> <a href="/view_bookings" style="color: aliceblue;">View All Bookings</a></center>
</body>
</html>
```

4.2.2. view_bookings.html:

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

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>View Bookings</title>
</head>
<body>
    <div class="background">
        <div class="booking-form">
            <h2>All Bookings</h2>
            <table>
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Name</th>
                        <th>Email</th>
                        <th>Starting Point</th>
                        <th>Destination</th>
                        <th>Departure Date</th>
                        <th>Return Date</th>
                        <th>Payment Method</th>
                    </tr>
                </thead>
                <tbody>
                    {% for booking in bookings %}
```

```
                <tr>
                    <td>{{ booking[0] }}</td>
                    <td>{{ booking[1] }}</td>
                    <td>{{ booking[2] }}</td>
                    <td>{{ booking[3] }}</td>
                    <td>{{ booking[4] }}</td>
                    <td>{{ booking[5] }}</td>
                    <td>{{ booking[6] }}</td>
                    <td>{{ booking[7] }}</td>
                </tr>
                {% endfor %}
            </tbody>
        </table>

        <a href="/">Go Back to Booking</a>
    </div>
  </div>
</body>
</html>
```

4.3. Static:

4.3.1.Style.css:

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
```

```
/* General Reset and Body Styling */
body, html {
    height: 100%;
    margin: 0;
    font-family: 'Poppins', sans-serif;
    color: black;
    background-image: url(backimg.jpeg);
}

/* Background Styling */
.background {
    background-image: url('bg.jpg');
    background-size: cover;
    background-position: center;
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
}

/* Booking Form Styling */
.booking-form {
    background-color: rgba(219, 201, 142, 0.7);
    padding: 20px;
```

```
    border-radius: 10px;
    max-width: 500px;
    width: 80%;
    margin: 0 auto;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* Subtle shadow for better design
*/
}

.booking-form h2 {
    text-align: center;
    font-size: 1.8rem;
    font-weight: bold;
    margin-bottom: 20px;
}

/* Form Structure */
.booking-form form {
    display: flex;
    flex-direction: column;
}

/* Input and Select Styling */
.booking-form input,
.booking-form select {
    padding: 10px;
```

```
    font-size: 1rem;
    border: 1px solid #ccc;
    border-radius: 5px;
    margin-bottom: 20px;
    width: 100%;
    background-color: #f9f9f9;
    transition: border-color 0.3s ease, box-shadow 0.3s ease;
}

.booking-form input:focus,
.booking-form select:focus {
    border-color: #007bff;
    outline: none;
    box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);
}

/* Submit Button Styling */
.booking-form button {
    margin-top: 10px;
    padding: 12px;
    font-size: 1rem;
    background-color: #3bbd07;
    color: white;
    border: none;
```

```
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease, transform 0.2s ease;
    font-weight: bold;
}

.booking-form button:hover {
    background-color: #45a049;
    transform: translateY(-2px);
}
```

4.4.App.py:

```
from flask import Flask, render_template, request, redirect, url_for
from flask_mysqldb import MySQL

app = Flask(__name__)

# MySQL Configuration
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'monisha@3009'  # Update with your MySQL password
app.config['MYSQL_DB'] = 'travel_bookings'

mysql = MySQL(app)
```

```
@app.route('/', methods=['GET', 'POST'])
def index():
    message = ''
    if request.method == 'POST':
        # Fetch form data
        name = request.form['name']
        email = request.form['email']
        starting_point = request.form['starting-point']
        destination = request.form['destination']
        departure_date = request.form['departure-date']
        return_date = request.form['return-date']
        payment_method = request.form['payment-method']

        # Insert data into the database
        cursor = mysql.connection.cursor()
        cursor.execute(
            'INSERT INTO bookings (name, email, starting_point, destination, departure_date, return_date, payment_method) '
            'VALUES (%s, %s, %s, %s, %s, %s, %s)',
            (name, email, starting_point, destination, departure_date, return_date, payment_method)
        )
        mysql.connection.commit()
        cursor.close()
```

```
        message = 'Booking Successful!'

    return render_template('booking_form.html', message=message)

@app.route('/view_bookings')
def view_bookings():
    # Fetch all bookings from the database
    cursor = mysql.connection.cursor()
    cursor.execute('SELECT * FROM bookings')
    bookings = cursor.fetchall()
    cursor.close()

    return render_template('view_bookings.html', bookings=bookings)

if __name__ == '__main__':
    app.run(debug=True)
```

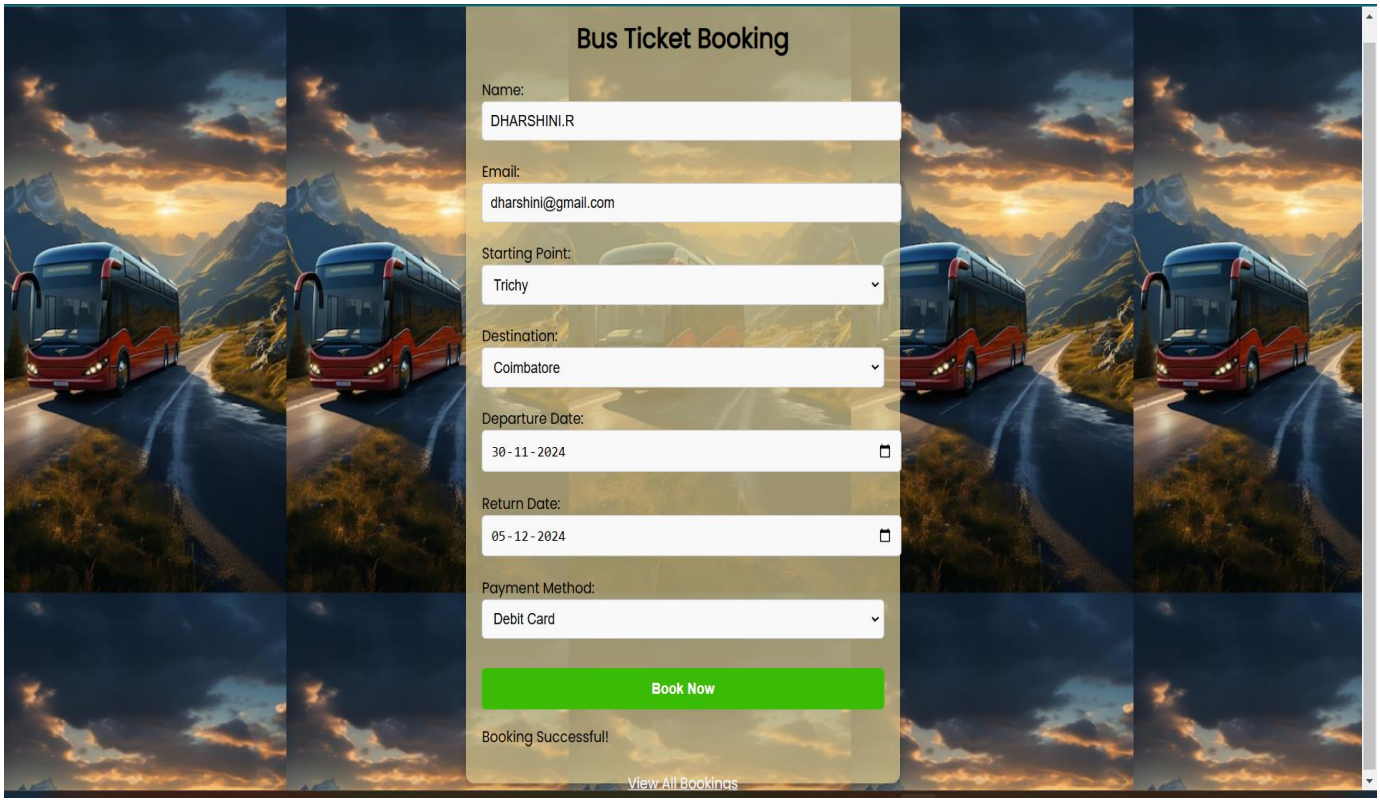
4.5. MySQL Database:

```
CREATE DATABASE travel_bookings;
USE travel_bookings;
CREATE TABLE bookings (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
```

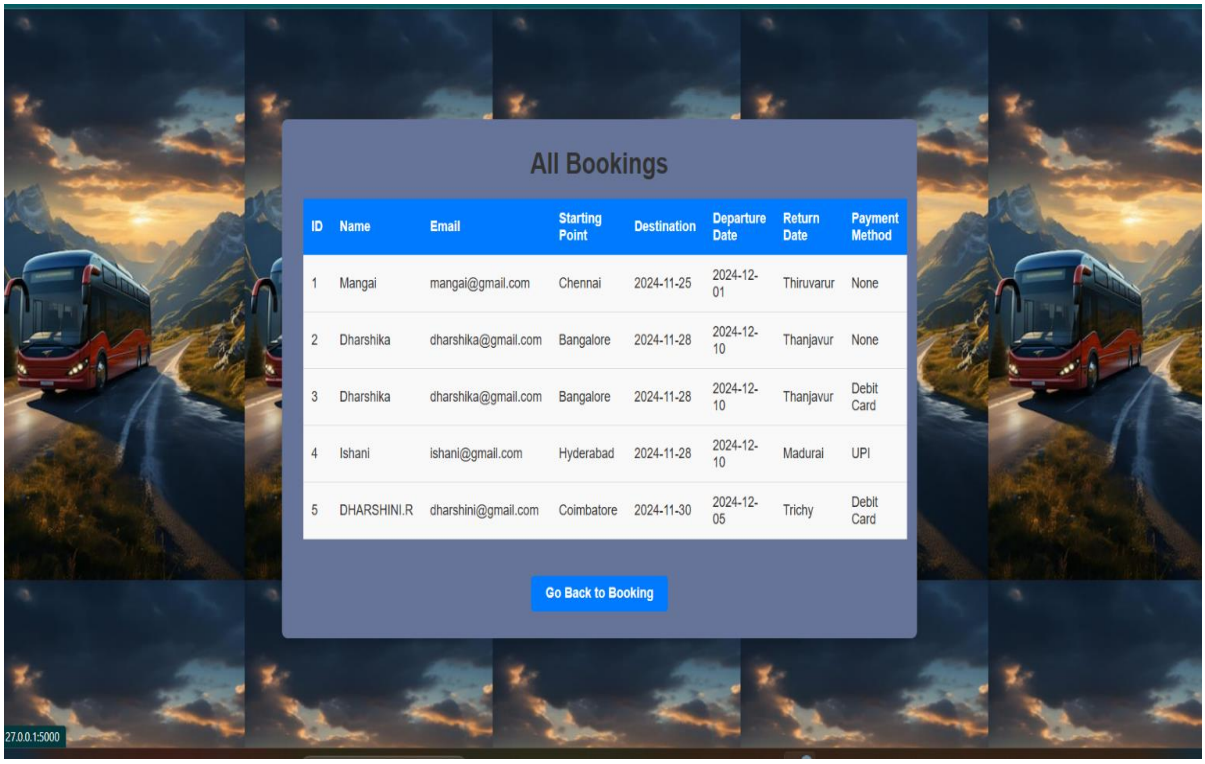
```
    starting_point VARCHAR(255) NOT NULL,
    destination VARCHAR(255) NOT NULL,
    departure_date DATE NOT NULL,
    return_date DATE NOT NULL,
    payment_method VARCHAR(255) NOT NULL
);
```

5.RESULT:

5.1.Booking Form:



5.2.View Page:



6. CONCLUSION:

The **Bus Ticket Booking System** is a user-friendly web application that simplifies the process of booking bus tickets. It is built using the **Flask** web framework for the backend, **HTML/CSS** for the frontend, and **MySQL** for data storage. This project serves as an excellent example of how to integrate a simple yet functional web application using modern development practices. It is ideal for beginners who want to learn full-stack development and database integration. The system can be expanded with additional features like ticket cancellation, booking history, and email notifications to make it a comprehensive ticket booking platform.

7.REFERENCE:

1. Flask:
Flask is the web framework used to develop the application.
Official Documentation: https://flask.palletsprojects.com/

2. Flask-MySQLdb:
Flask-MySQLdb is used to connect Flask with the MySQL database.
Documentation: https://flask-mysqldb.readthedocs.io/

3.MySQL:
MySQL is the relational database management system used for storing the booking data.
Documentation: https://dev.mysql.com/doc/

4. HTML & CSS:
HTML is used for structuring the web pages, and CSS is used for styling them.
Reference: https://developer.mozilla.org/en-US/docs/Web