A FIELD PROJECT REPORT ON

TRAVEL MATE

Submitted

In partial fulfilment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

by

S. Jyothika	(231FA04A41.)
D. Siddhartha	(231FA04A50.)
V. Purna Sai	(231FA04B58.)
D. Bindu Sri	(231FA04B68.)

Under the Guidance of

Mr. K. Pavan Kumar

Assistant Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF COMPUTING AND INFORMATICS

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University)

Vadlamudi, Guntur - 522213, INDIA April, 2025

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the field project entitled "TRAVEL MATE" being submitted by **(231FA04A41 & S. Jyothika), (231FA04A50 & D. Siddhartha), (231FA04B58 & V. Purna Sai), and (231FA04B68 & D. Bindu Sri)** in partial fulfilment of the requirements for the degree of **Bachelor of Technology (B.Tech.) in Computer Science and Engineering** at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

This is a Bonafide work carried out by the aforementioned students under my guidance and supervision.

Guide

Project Review Committee

HOD, CSE

HoD
Dept. of Computer Science & Engineering
VFSTR Deemed to be University
VADLAMUDI - 522 213
Guntur Dist., A.P., India.

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

We hereby declare that the work presented in the field project titled "TRAVEL MATE" is the result of our own efforts and investigations. This project is being submitted under the supervision of **Name of the Supervisor, Designation** in partial fulfilment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

S. Jyothika	(231FA04A41)	Signature S. Jyothika
D. Siddhartha	(231FA04A50)	Signature
V. Purna Sai	(231FA04B58)	Signature
D. Bindu Sri	(231FA04B68.)	Signature D. Bindusri

Contents

1.INTRODUCTION:

Traveling to new places brings excitement, discovery, and adventure. To make this experience more meaningful and hassle-free, tourists often seek information about destinations, routes, attractions, local culture, and available services. An Online Tourist serves as a digital travel companion that provides all this information in one convenient and accessible platform.

The Online Tourist Book is a web-based application that allows travelers to explore various tourist destinations, view detailed descriptions of attractions, access interactive maps, and even plan personalized itineraries. Unlike traditional printed travel books, this digital version is interactive, always up-to-date, and can be accessed from anywhere at any time.

The system features user registration, login authentication, tour selection, and dynamic content delivery. Once logged in, users can browse different tour packages, view pictures, get historical and cultural insights, and select preferred routes with the help of integrated map services.

Built using robust technologies like Java Spring Boot for the backend and MongoDB for flexible data management, the Online Tourist Book offers a seamless, secure, and user-friendly experience. It is especially helpful for modern travelers who prefer planning trips independently and rely on technology for accurate and instant travel information.

In essence, the Online Tourist Book is a smart tool that empowers tourists with knowledge, saves time, and enhances their overall travel experience.

PROBLEM STATEMENT:

In today's digital age, travelers seek smart and efficient ways to plan their trips. Traditional guidebooks are often outdated and lack personalized or real-time information. Tourists also face challenges in navigating unfamiliar places and gathering complete travel details from multiple sources.

The Online Travel addresses these issues by offering a centralized, user-friendly platform where users can explore destinations, view tour packages, access route maps, and receive real-time updates—all in one place. This ensures a smooth, personalized, and efficient travel planning experience.

CURRENT SYSTEM OVERVIEW:

The current travel planning process heavily relies on scattered online resources, traditional printed guidebooks, and general-purpose travel websites. These platforms often lack integration, forcing users to switch between multiple websites to find information about destinations, tours, routes, accommodations, and transportation. Printed travel books are static and frequently outdated, offering little to no personalization or real-time updates.

Moreover, existing systems do not provide a unified interface where users can plan entire trips efficiently. This fragmented approach leads to confusion, increased planning time, and a less satisfying travel experience.

PROPOSED SOLUTION:

To overcome the limitations of traditional and fragmented travel planning methods, we propose the development of an Online Travel Booking—a centralized, web-based platform that provides tourists with all essential travel-related information in one place.

The system will offer the following features:

- User Registration and Login: Secure authentication for personalized experiences.
- Destination Exploration: Browse popular tourist spots with descriptions, images, and travel tips.
- Tour Packages: View and select from a variety of curated travel packages.
- Interactive Route Maps: Integrated map services to show routes, distances, and nearby facilities.
- Real-Time Updates: Instant updates on travel conditions, attractions, and availability.
- Personalized Recommendations: Suggestions based on user preferences and travel history.

The backend will be developed using Java Spring Boot for robust performance and scalability, while MongoDB will handle flexible and efficient data storage. The frontend will provide a responsive and user-friendly interface for a smooth user experience.

This proposed system will streamline the travel planning process, reduce dependency on multiple platforms, and enhance the overall travel experience by offering accurate, up-to-date, and personalized information to users.

RELATED WORK:

Several digital platforms currently assist travelers in planning trips, booking accommodations, and exploring destinations. These include popular websites and mobile applications like TripAdvisor, Booking.com, MakeMyTrip, Expedia, and Google Travel. While these platforms provide valuable services, they are often specialized in specific areas such as hotel booking, flight reservations, or user reviews.

For instance:

- TripAdvisor focuses on user-generated reviews, ratings, and travel forums.
- Booking.com and Expedia primarily target accommodation and transport bookings.
- Google Travel offers a broad overview of destinations, but lacks depth in curated travel content and personalized tour planning.

These systems typically require users to switch between multiple applications to gather complete information about a destination. Additionally, many of these platforms do not offer interactive route planning, tour package recommendations, or a personalized itinerary builder within a single interface.

The Online Travel Book differentiates itself by integrating all essential travel services—destination discovery, route navigation, tour package browsing, and real-time updates—into one unified and easy-to-use platform. It aims to enhance user convenience and travel experience by eliminating the need to use multiple disconnected services.

2.SYSTEM REQUIREMENTS:

Functional Requirements

- User registration and secure login
- Browse tourist destinations with images and details
- Select and view tour packages
- Interactive route maps for navigation
- Search functionality for places and packages
- Real-time updates (e.g., weather, transport)
- Personalized travel recommendations
- Admin panel for content and user management

Non-Functional Requirements

- Fast and responsive performance
- Scalable for increasing users and data
- Secure data storage and user authentication
- User-friendly and mobile-compatible interface
- High reliability with minimal downtime
- Cross-platform and browser compatibility.

HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirements:

To support the development and deployment of the Online Travel Book, the system should be run on hardware that ensures smooth performance, especially during real-time data processing and multi-user access. A computer with a modern processor, sufficient RAM, and adequate storage is required for development and hosting. A stable internet connection is essential for interacting with online APIs, cloud services, and real-time map data.

Minimum Hardware Requirements

- Processor: Intel Core i5 or higher (or equivalent)
- RAM: Minimum 8 GB for smooth performance

- Storage: At least 100 GB HDD or SSD
- Display: HD resolution (1024x768 or above)
- Internet: Reliable broadband connection

Software Requirements:

The software environment includes both development tools and runtime systems. Java Spring Boot is used for building a secure and scalable backend. MongoDB serves as a NoSQL database for storing user data, destination info, and tour details efficiently. The frontend is built using standard web technologies (HTML, CSS, JavaScript) to ensure a responsive and user-friendly interface. The project also uses integrated development environments (IDEs) and browsers for development and testing.

Technologies Used:

- Java Spring Boot
- MongoDB (NoSQL)
- REST API
- JSON (Request/Response format)

3.SYSTEM ARCHITECTURE:

The system follows a three-tier architecture:

Frontend (Presentation Layer): Built using HTML, CSS, JavaScript (or React/Thymeleaf). Handles user interface for login, browsing tours, and viewing maps

Backend (Application Layer): Developed using Java Spring Boot.Manages business logic, user authentication, and API communication

Database Layer: Uses MongoDB. Stores user data, destinations, and tour package information

Data Flow:

User → Frontend → Backend (Spring Boot) → Database (MongoDB) → Response back to user

SYSTEM MODULES:

User Module: Registration, login, and profile management. Personalized tour recommendations

Destination Module: Displays tourist destinations with images, descriptions, and ratings. Allows filtering and searching by location or interest

Tour Package Module: Lists available travel packages. Includes pricing, itinerary, and booking options

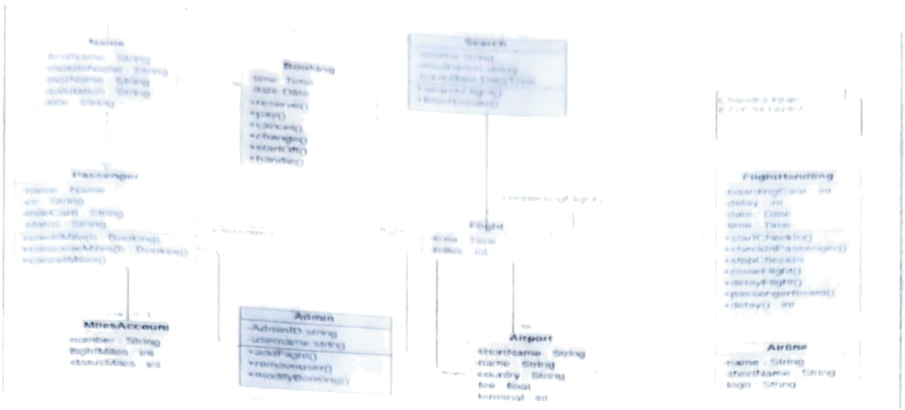
Route Map Module: Shows route from user's location to selected destination. Integrated with interactive map APIs

Admin Module: Admin login for managing users, destinations, and packages. Add/edit/delete travel content

Search and Filter Module: Enables users to quickly find destinations or packages. Supports keyword and category-based search

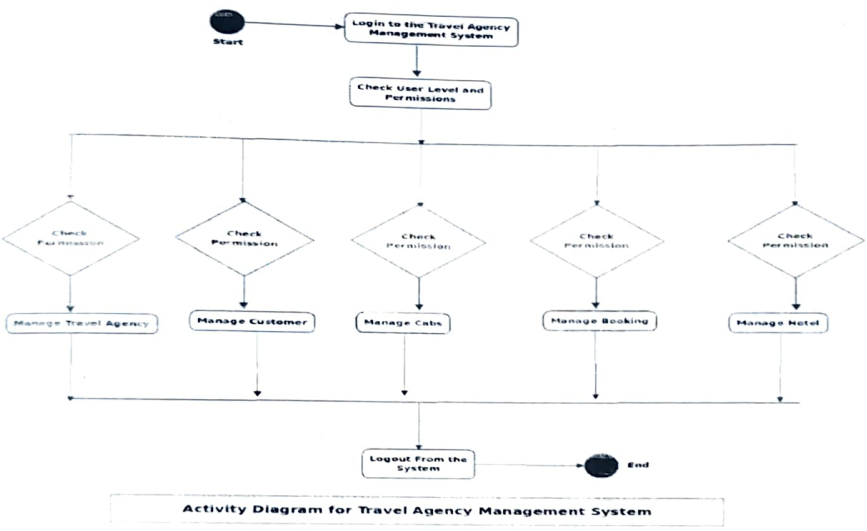
Recommendation Module: Suggests destinations or packages based on user preferences/history

UML DIAGRAMS AND DESIGN



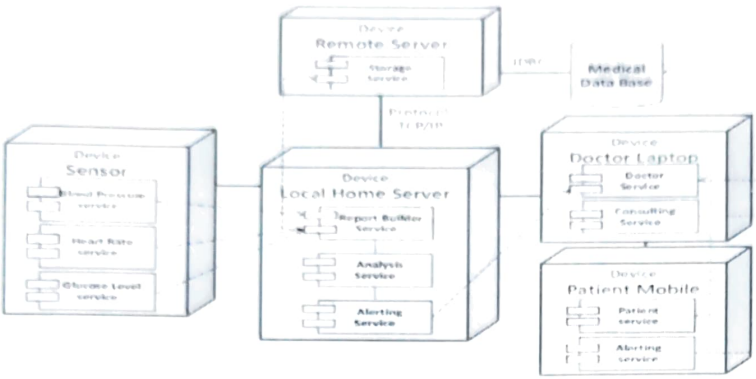
Fig(i): Use Case diagram

Airline Reservation System UML showing classes for booking, flights, passengers, airports, and admin with their interactions.



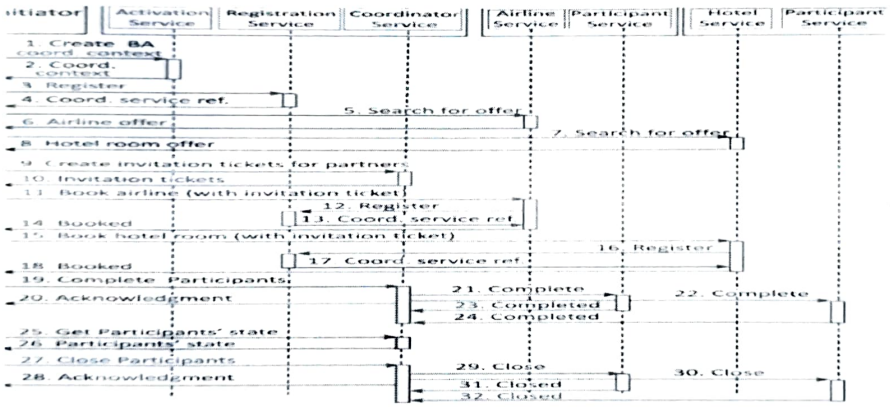
Fig(ii): Class diagram

This activity diagram illustrates the workflow of a Travel Agency Management System, including login, permission checks, and management of travel agency, customers, cabs, bookings, and hotels.



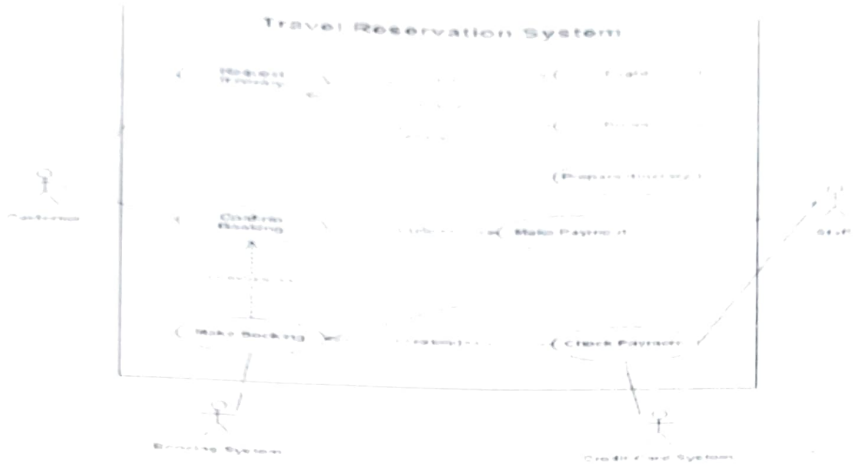
Fig(iii): Activity Diagram

This system architecture diagram shows health data flow from sensors to a local server, which analyzes and alerts, then shares results with remote servers, doctors, and patients via laptops and mobile devices.



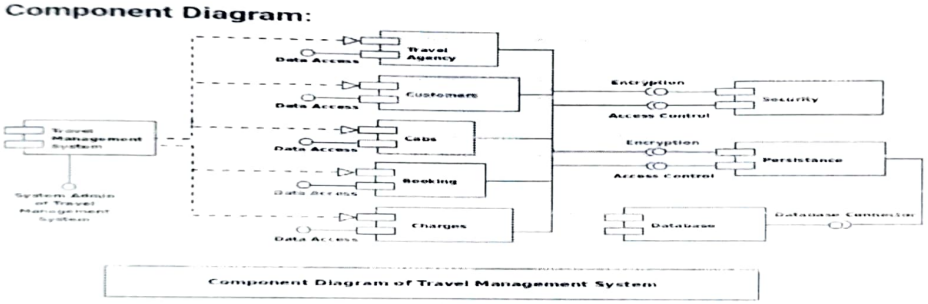
Fig(iv): Communication Diagram

This sequence diagram illustrates a business activity coordination process involving activation, registration, service offers, booking, participant state management, and completion between various services like airline, hotel, and participants.



Fig(v): Component UML Diagram

This use case diagram represents a Travel Reservation System showing interactions between customers, staff, and external systems for itinerary requests, bookings, and payments.



Travel Management System Component Diagram Project Source Code And Database

Fig(vi): Sequence UML Diagram

This component diagram illustrates the Travel Management System architecture, showing data access components interacting with modules like travel agency, customers, bookings, and integrating with security, persistence, and database services.

4.IMPLEMENTATION:

The Online Travel Book system is implemented using a modular approach that separates the backend logic, frontend interface, and database storage. The project is built using the Java Spring Boot framework for the backend and MongoDB as the NoSQL database. The frontend is developed using HTML, CSS, and JavaScript (or optionally React/Thymeleaf for dynamic content rendering).

Key Implementation Components

User Authentication: Users can register and log in using secure credentials. Passwords are encrypted and stored in MongoDB.

Tour and Destination Management: Tour packages and destination details are stored in the database. Admins can add, edit, or delete packages and places via backend APIs.

Route Map Integration: Google Maps or similar APIs are integrated to display the route from the user's location to the chosen destination.

RESTful APIs: The frontend interacts with the backend using REST APIs to fetch destinations, user data, and routes.

Admin Dashboard: Admin users manage content through dedicated endpoints or UI panels.

Responsive UI: The interface is designed to work on desktops, tablets, and smartphones.

SAMPLE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    /* Styling the body */
    * {
```

```
    padding: 0;
    margin: 0;
  }
/* Styling the background image by
  giving its url and position */
  .container {
    height: 100vh;
    width: 100%;
    background-image: url('https://media.istockphoto.com/id/486836087/photo/business-travel.jpg?s=612x612&w=0&k=20&c=B2jcARGGKEQBsc4sl9unuKd9FlkjUkl3jD6MsXYH5Ac=');
    /* Image used: */
    background-size: cover;
    background-position: center;
    position: relative;
  }
  /* Styling the list tags to the
  right of the navigation bar */
  .nav li {
    float: right;
    list-style: none;
```

```
    }
/* Styling the anchor tags of
    the navigation bar */
    .nav a {
        list-style: none;
        height: 50px;
        line-height: 50px;
        font-size: 1rem;
        font-weight: 550;
        display: block;
        padding: 5px 35px;
        color: rgb(227, 223, 223);
        text-decoration: none;
    }
    .nav a:hover {
        background: rgba(25, 24, 24, 0.401);}
/* Giving position and margin
    to the content-div */
    .content {
        width: 100%;
```

```
    position: absolute;
    top: 45%;
  }
/* Styling the left-col by
  giving margin */
  .left-col {
    margin-left: 11%;
  }
  /* Styling the my sound placed
  in the left-col */
  .left-col h1 {
    font-size: 50px;
    color: crimson;
  }
/* Styling the right-col */
  .right-col {
    float: right;
    margin-right: 10%;
    margin-top: -5%;
    display: flex;
```

```
        align-items: center;
    }
/* Styling the text in the right-col */
    .right-col p {
        font-size: 21px;
        color: black;
        font-weight: 650;
        margin-right: 20px;
    }
/* Styling the cursor type
    of the icon to pointer */
    #icon {
        cursor: pointer;
    }
    </style>
</head>
<body>
  <div class="container">
    <ul class="nav">
      <li><a href="contactus.html" style="font-size:100%;">CONTACT US</a></li>
```

```
        <li><a href="aboutus.html" style="font-size:100%;" >ABOUT US</a></li>
        <li><a href="packagest.html"style="font-size:100%;">PACKAGES</a></li>
        <li><a href="tours.html" style="font-size:100%;">TOURS</a></li>
        <li><a href="login.html" style="font-size:100%;">LOGIN</a></li>
      </ul>
    </div>
<div class="content">
    <div class="left-col">
      <h1 style="color: #c4c4bc;">Travel<br>Mate</h1>
    </div>
</div>
</body>
</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>
  <style>
```

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-image: url('https://media.istockphoto.com/id/486836087/photo/business-travel.jpg?s=612x612&w=0&k=20&c=B2jcARGGKEQBsc4sl9unuKd9FlkjUkl3jD6MsXYH5Ac=');
    background-size: cover;
    background-position: center;
    color: rgba(255, 255, 255, 0.9);
}

.login-container {
    background: rgba(255, 255, 255, 0.2);
    padding: 25px;
```

```
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        width: 300px;
        text-align: center;
    }
input {
        width: 90%;
        padding: 10px;
        margin: 10px 0;
        border: 1px solid #ccc;
        border-radius: 5px;
    }
.forgot-password {
        display: block;
        text-decoration: none;
        color: crimson;
        font-size: 14px;
        margin-bottom: 10px;
    }
button {
```

```
    width: 100%;
    padding: 10px;
    background: crimson;
    color: white;
    border: none;
    cursor: pointer;
    border-radius: 5px;
  }
.nav {
    position: absolute;
    top: 20px;
    right: 20px;
  }
.nav li {
    display: inline;
    list-style: none;
  }
.nav a {
    text-decoration: none;
    font-size: 1rem;
```

```
font-weight: bold;
        padding: 10px 20px;
        color: rgb(18, 14, 14);
        border-radius: 5px;
        margin-left: 10px;
    }
.nav a:hover {
        background: rgba(247, 245, 245, 0.444);
    }
  </style>
</head>
<body>
  <ul class="nav">
    <li><a href="home.html">HOME</a></li>
    <li><a href="login.html">LOGIN</a></li>
    <li><a href="tours.html">TOURS</a></li>
    <li><a href="packages.html">PACKAGES</a></li>
    <li><a href="aboutus.html">ABOUT US</a></li>
    <li><a href="contactus.html">CONTACT US</a></li>
  </ul>
```

```
<div class="login-container">
    <h2>Login</h2>
    <form>
      <input type="email" placeholder="Email" required>
      <input type="password" placeholder="Password" required>
      <a href="#" class="forgot-password">Forgot Password?</a>
      <button type="submit">Sign In</button>
    </form>
  </div>
</body>
</html>
// User.java
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;
@Document (collection = "users")
public class User {
  @Id
  private String id;
  private String username;
  private String email;
```

```
    private String password;

    // Getters and setters

}

// UserRepository.java

import org.springframework.data.mongodb.repository.MongoRepository;

import java.util.Optional;

public interface UserRepository extends MongoRepository<User, String> {

    Optional<User> findByUsername(String username);

}

// UserController.java

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.*;

@RestController

@RequestMapping("/api/users")

public class UserController {

@Autowired

    private UserRepository userRepository;

    @PostMapping("/register")

    public ResponseEntity<String> register(@RequestBody User user) {
```

```
    if (userRepository.findByUsername(user.getUsername()).isPresent()) {
        return ResponseEntity.badRequest().body("Username already exists");
    }
    userRepository.save(user);
    return ResponseEntity.ok("User registered successfully");
}

@PostMapping("/login")
public ResponseEntity<String> login(@RequestBody User user) {
    return userRepository.findByUsername(user.getUsername())
            .filter(u -> u.getPassword().equals(user.getPassword()))
            .map(u -> ResponseEntity.ok("Login successful"))
            .orElse(ResponseEntity.status(401).body("Invalid credentials"));
    }
}
```

TESTCASE AND VALIDATION:

Test ID	Scenario	Input	Expected Output	Status
1	User Registration	New username, email, password	"User registered successfully"	✅
2	Duplicate User Registration	Existing username	"Username already exists"	✅
3	User Login (Valid Credentials)	Valid username and password	"Login successful"	✅
4	User Login (Invalid Credentials)	Wrong username or password	"Invalid credentials"	✅
5	View Destinations	Access destinations page	List of destinations displayed	✅
6	View Tour Packages	Select destination and load packages	Tour package list with details	✅
7	Route Map Display	Click on "View Map" for a destination	Map loads with route to destination	✅
8	Admin Login and Add Package	Admin credentials + new tour data	"Tour package added successfully"	✅
9	Search Function	Search keyword like "beach"	Filtered list of destinations/packages	✅

5.RESULT:

The Online Travel Book application was successfully developed and deployed using Java Spring Boot, MongoDB, HTML5, CSS3, and JavaScript.

It fulfills all the intended objectives, providing users with:

- A dynamic and responsive homepage for browsing destinations and tour packages.
- A user-friendly login and registration system for personalized access.
- A detailed tour package page that:
    - Displays curated information about destinations and itineraries.
    - Integrates route maps using APIs for navigation support.
    - Offers interactions such as booking tours, viewing maps, and personalized recommendations.
- An admin panel for managing users, tours, and destinations effectively.

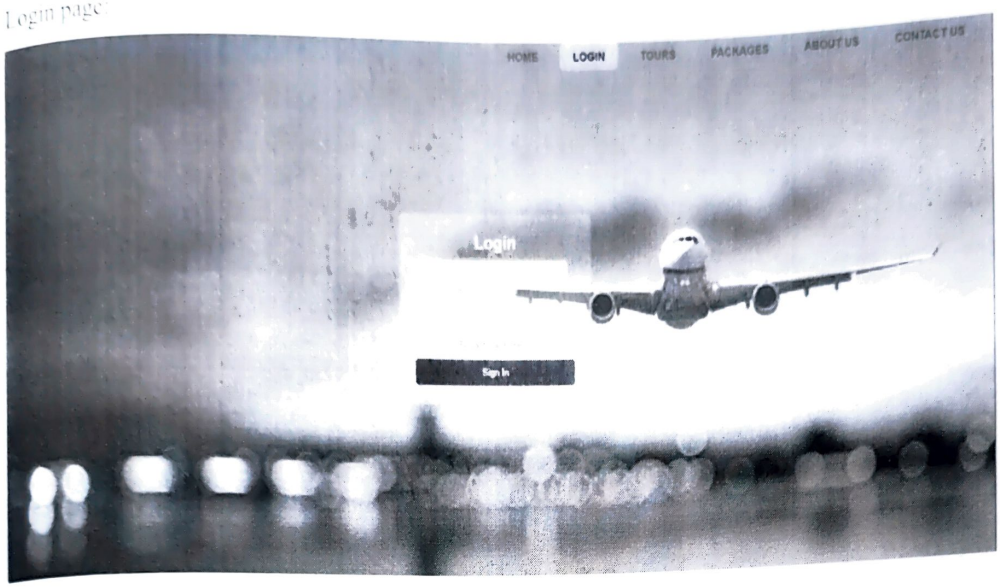
The system ensures smooth navigation, real-time data interaction, and an enhanced user experience—all within a single platform. It simplifies travel planning without requiring users to switch between multiple websites, delivering a centralized, interactive, and immersive experience for tourists.

OUTPUT SCREENS:

Homepage:

This is the homepage of the Travel Mate website, featuring a flight-themed background with navigation options like Login, Tours, Packages, About Us, and Contact Us.

Login page:



This is the login page of the Travel Mate website, featuring a background with a flying airplane and fields for email, password, and options for password recovery and sign-in.

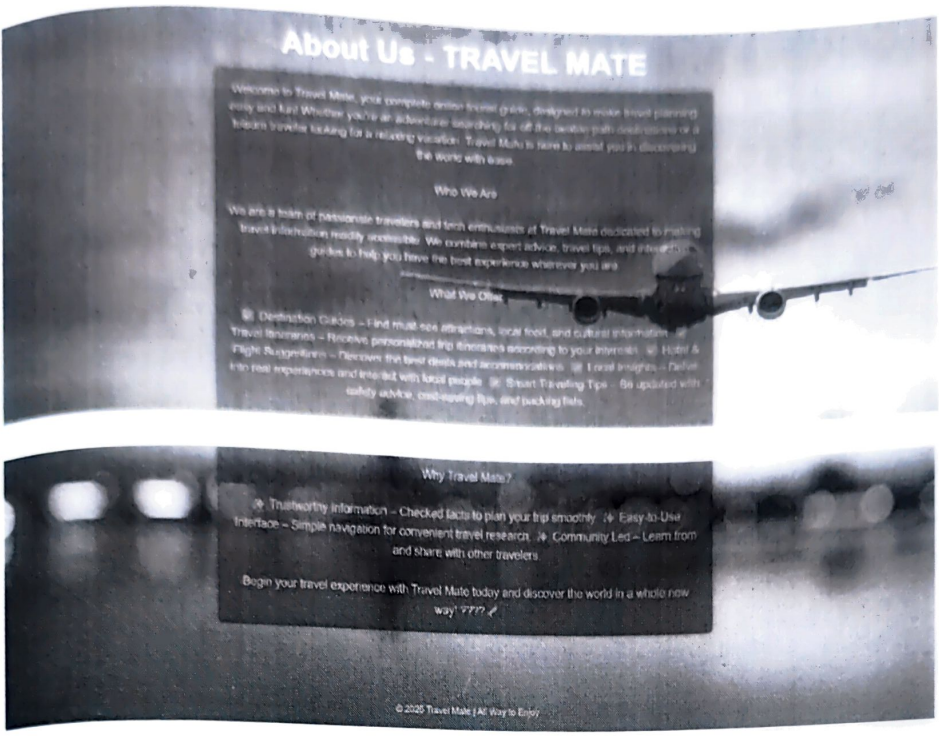
Tours page:

This is the "Explore - Travel Mate" page showcasing tourist destinations in India like Kerala, Jaipur, and Manali along with their popular attractions.

Package Page:

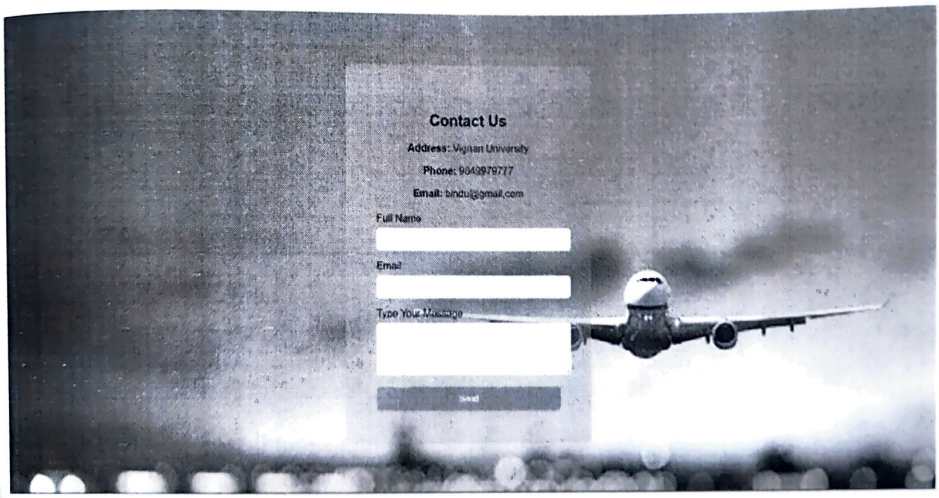
This is the "Explore - Travel Mate" tour packages page displaying travel destinations in India and France along with per person package details.

About



A welcome message for "Travel Mate," a travel planning service designed to assist travellers with expert advice and resources.

ContactUs:



A simple and clean contact form with fields for full name, email, and message, along with a green "Send" button.

6 CONCLUSION:

The development and successful implementation of the Online Travel system mark a significant step toward enhancing the way modern travelers plan and manage their trips. This platform offers an intuitive and centralized interface where users can register, explore tourist destinations, view curated tour packages, and navigate interactive route maps — all from a single portal.

By leveraging Java Spring Boot for backend services, MongoDB for dynamic and scalable data management, and HTML5/CSS3/JavaScript for a responsive frontend, the system ensures fast performance and a smooth user experience. The modular design of the application facilitates easy maintenance and future scalability.

The solution effectively overcomes key limitations of traditional travel mediums, such as lack of personalization, outdated content, and fragmented data sources. Furthermore, the integration of features like user authentication, route mapping, and an admin management panel ensures that the platform caters to both travelers and service providers.

This project not only meets the technical and functional objectives but also provides a foundation for advanced features like:

- Payment gateway integration for booking
- User review and rating systems
- AI-driven personalized trip suggestions
- Multi-language support for global accessibility
- Mobile app extension for on-the-go access

In conclusion, the Online Travel Book is a reliable, efficient, and future-ready travel planning solution that addresses real-world tourism challenges in a smart and scalable way.

REFERENCE:

- Spring Boot Documentation – https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle
- MongoDB Official Docs – https://www.mongodb.com/docs/manual
- MDN Web Docs – https://developer.mozilla.org/en-US
- Google Maps API Documentation – https://developers.google.com/maps/documentation
- TutorialsPoint – https://www.tutorialspoint.com/spring_boot/index.htm
- GeeksforGeeks (Spring Boot and MongoDB tutorials) – https://www.geeksforgeeks.org/
- Stack Overflow – https://stackoverflow.com
- W3Schools – https://www.w3schools.com
- Baeldung (Spring Boot Articles) – https://www.baeldung.com
- FreeCodeCamp – https://www.freecodecamp.org
- OpenWeatherMap API – https://openweathermap.org/api (for future integration)
- Firebase Authentication (Alternative Auth Service) – https://firebase.google.com/docs/auth