

WANDERLUXE – TRAVEL BOOKING WEB APPLICATION

PROJECT REPORT

1. Project Overview

WanderLuxe is a modern and user-friendly travel booking web application developed using **React (Vite)** and **JSON Server**. The application allows users to browse destinations, explore curated travel packages, view detailed package information, and book their trips through a simple and aesthetically pleasing interface.

This project demonstrates strong understanding of frontend development, routing, user interface design, and API integration.

2. Objectives

- To create a visually appealing and responsive travel booking interface.
 - To implement multi-page navigation using **React Router**.
 - To display dynamic package details using route parameters.
 - To build a functional booking form connected to a backend API (JSON Server).
 - To organize code into modular, reusable components.
 - To deliver a clean and professional user experience.
-

3. Tools & Technologies Used

Frontend Technologies

- React.js
- Vite
- JavaScript (ES6+)
- React Router DOM
- CSS (Custom styling + inline styling)

Backend Technologies

- JSON Server
- db.json (mock database)

Development Tools

- Visual Studio Code
- GitHub
- Google Fonts
- Unsplash Images

4. System Architecture

User Interface (React)

- React Router
- API Calls to JSON Server (<http://localhost:4000>)
- Data Storage in db.json

The architecture follows a clean separation between UI, routing, and backend simulation using JSON Server.

5. Project Modules Explained

5.1 Home Page

- Contains a hero banner.
- Displays popular destinations like Goa, Kerala, Andaman, Himalayas, etc.
- Smooth navigation to packages page.

5.2 Packages Page

- Shows an attractive grid of travel packages.
- Each card displays:
 - Package image
 - Destination title
 - Price
 - Trip duration
 - “View Details” button

5.3 Package Details Page

- Displays **full description** of the selected package.
- Landscape-style optimized image (fixed height for better UI).
- Includes package duration, price, and a **Book Now** button.
- Fetches data dynamically using route parameters.

5.4 Booking Form

- Users can enter:

- Full name
- Email
- Number of travellers
- Travel date
- Data is submitted to JSON server using:
POST → <http://localhost:4000/bookings>
- Uses a custom React hook for API submission.

5.5 Contact Page

- Contains support email and phone number.
- Styled using inline CSS.
- Simple and clean layout.

5.6 About Page

- Explains what WanderLuxe is.
- Includes attribution: “Created by Amirtha Varshini S”.
- Clean minimal aesthetic design.

5.7 Footer

- Fixed at bottom of every page.
 - Shows:
© 2025 WANDERLUXE by Amirtha Varshini S
 - Center-aligned.
-

6. JSON Server Endpoints

Fetch All Packages

GET <http://localhost:4000/packages>

Fetch Single Package

GET <http://localhost:4000/packages/:id>

Store Booking Data

POST <http://localhost:4000/bookings>

View All Bookings

GET <http://localhost:4000/bookings>

7. Folder Structure Overview

```
src/
  components/
    Navbar.jsx
    Footer.jsx
    DestinationCard.jsx
    PackageCard.jsx
    RouteTransition.jsx
  context/
    BookingContext.jsx
  hooks/
    useSubmitBooking.js
  routes/
    Home.jsx
    Packages.jsx
    PackageDetails.jsx
    BookingForm.jsx
    Contact.jsx
    About.jsx
  App.jsx
  main.jsx

public/
```

favicon.svg (custom pink plane icon)

db.json
package.json
vite.config.js
README.md

8. Conclusion

WanderLuxe successfully demonstrates the complete implementation of a modern travel booking application using React. The project highlights strong skills in:

- Component-based architecture
- Routing and navigation
- API handling with JSON Server
- UI/UX design
- State management through React Context

This application can be extended further by adding:

- User authentication
- Payment integration
- Admin dashboard
- Real travel APIs

This project is suitable for academic assessments, training evaluations, and frontend development showcases.

9. Student Info :

Name: Amirtha Varshini S

Project Title: WanderLuxe – Travel Booking Web Application

Technologies Used: React, Vite, JSON Server
