**Ease My Booking**

Problem Statement:   
All of us like to visit famous places in any location. But to enter any place like a zoo,   
museum etc. we always need to buy tickets to enter and in the booking counter you   
always need to stand in a long queue.   
To overcome this problem, we can create an online platform where all the places can   
register themselves and people can book tickets online to avoid the queue.

A full-stack tourism and ticket booking platform for discovering, viewing, and booking local experiences (e.g., zoos, events, adventure sports) across multiple cities in India. Users can browse by categories, view packages, select slots & prices, and make bookings. Admins can manage places, packages, slots, and prices through a dashboard.

**User Functionality**

Search bar with live suggestions for places (by name).

Browse by city and category (e.g., Mysore → Zoo).

Browse by popular places

View package details, slot timings, and pricing

Select slot date

Book tickets, generating a unique transaction ID.

Login / Register (with JWT Bearer authentication).

View "My Bookings" dashboard for logged-in users.

If user as two rule then it will given option login based on role

**Organiser Functionality**

Multi-step modal for:

Adding a Place

Adding Packages

Adding Slots (time/date)

Adding Price per slot

View, Edit, Delete Place, Package ,slot and price

**Admin dashboard**

Admin can verify place which is added by organiser and has an option to reject

Admin dashboard contains the details of all places

**Tech Stack-Frontent**

Vite+React.js

React Router DOM for routing.

Redux Toolkit for state management (slicer, thunk, services pattern).

Material UI (MUI) for responsive UI and components.

Axios for API requests.

Jspdf html2canvas for downloading ticket

React bootstrap, toastify

Cypress for front end testing and started with front end testing with component testing approach

We will be using cloudinary for storing images in cloud

**Authentication**

JWT Token-based login system.

Token stored in localStorage and passed as Bearer token.

Authenticated endpoints for booking and dashboard access.

**Backend -ASP.NET Core Web API**

We have created 5 microservice ,a JWTExtension and a gateway

SQL Server used for Relational Database as code first approach

AutoMapper used for DTO to entity mapping

JWT Bearer Authentication for Token-based authentication for securing APIs

MemoryCache as to be implemented

API Gateway (Ocelot) is used for single entry point for frontend calls routing to microservices

Swagger API documentation and testing

DTOs Data transfer objects for structured request/response

We will be working on NUnit test for backend testing

**Database-**

SQL server is used for storing data in database

**Deployment**

We will use docker and azure for deployment