HACKATHON PHASE-I

COLLEGE CODE :9530

COLLEGE NAME: ST.MOTHER THERESA ENGINEERING COLLEGE

DEPARTMENT : COMPUTER SCIENCE ENGINEERING

STUDENT NM-ID:7E218C3BFA487AFB48C4D7838988A650

ROLL NO :953023104082

DATE :06/10/2005

TECHNOLOGY : Front End/Node JS

PROJECT NAME : RESTAURANT TABLE BOOKING

SUBMITTED BY: (Team members name, mention TL)

TL Name :Nithish kumar.R

Mobile No:801595811

Team members:

1.navin.k[f6ec978c3ee75729d534ddc7f40e69c1]

Mobile No:9791413275

2.muthu Nagaraj.K[B71A418B6B70BC7BD1C42FE568E05544]

Mobile No:8838036512

3.parthiban.C[F8916AB947A382CB7194F395BD6B7947]

Mobile NO:8754731371

4.Muthu kumar.V[BEBFD5F37F974CCAE794C7A88EC1D1CD]

Mobile:8056847389

Project Overview & Objectives

Problem Statement:

Customers often face delays or uncertainty when visiting restaurants without knowing table availability. Restaurants, too, need a simple way to manage table reservations efficiently.

Objective:

Develop a Restaurant Table Booking web application that allows users to:

View available tables in real-time.

Book a table for a specific date/time.

Receive confirmation instantly.

Allow restaurants to manage bookings and view daily reservations.

Expected Outcome:

A responsive, user-friendly booking platform that reduces waiting time, improves restaurant efficiency, and enhances customer satisfaction.

Technology Stack & Environment Setup

Frontend:

React JS / Angular / HTML + CSS + JavaScript

Tailwind CSS or Bootstrap for styling

Backend:

Node.js with Express.js

Database:

MongoDB (Cloud via MongoDB Atlas) or MySQL

Tools & Environment:

Visual Studio Code

Git + GitHub for version control

Postman (for API testing)

Netlify / Vercel (for deployment)

API Design & Data Model

Planned REST Endpoints:

Method	Endpoint	Description
POST	/api/users/register	Register new user
POST	/api/users/login	User login
GET	/api/tables	Get available tables
POST	/api/bookings	Create new booking
GET	/api/bookings/:id	View booking details
DELETE	/api/bookings/:id	Cancel booking

Request/Response Format (Example):

POST /api/bookings

```
{
  "userId": "u123",
  "tableId": "t5",
  "date": "2025-10-10",
  "time": "19:00",
  "guests": 4
}
```

Response:

```
{
```

"message": "Table booked successfully",

"bookingId": "b678"

}

Database Schema (Example):

Users: { userId, name, email, password }

Tables: { tableId, capacity, status }

Bookings: { bookingId, userId, tableId, date, time, guests, status }

Front-End UI/UX Plan

Wireframes & Navigation Flow:

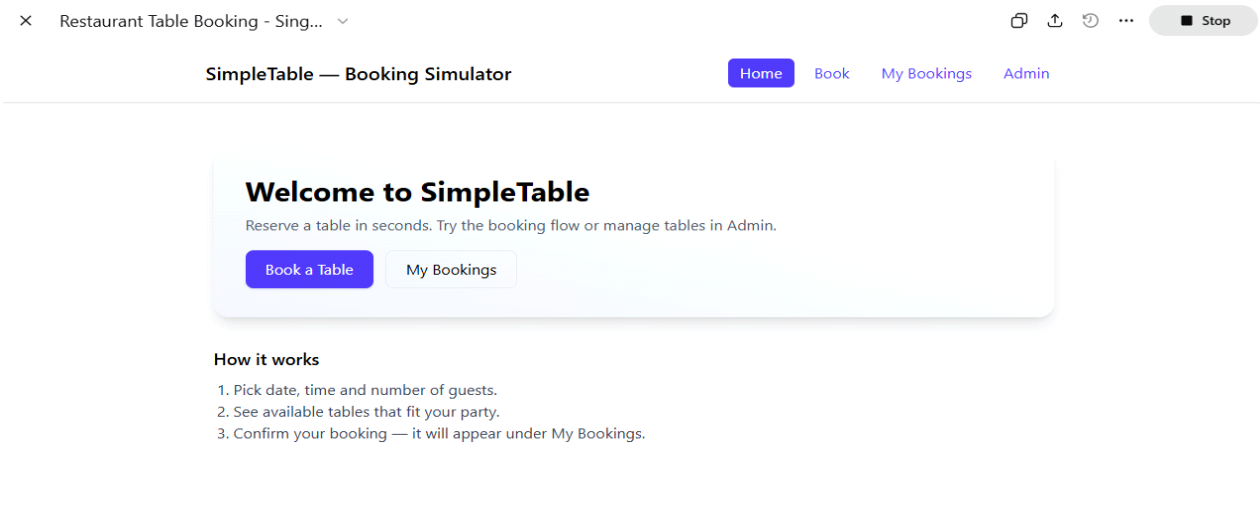
Home Page → Welcome + “Book a Table” CTA

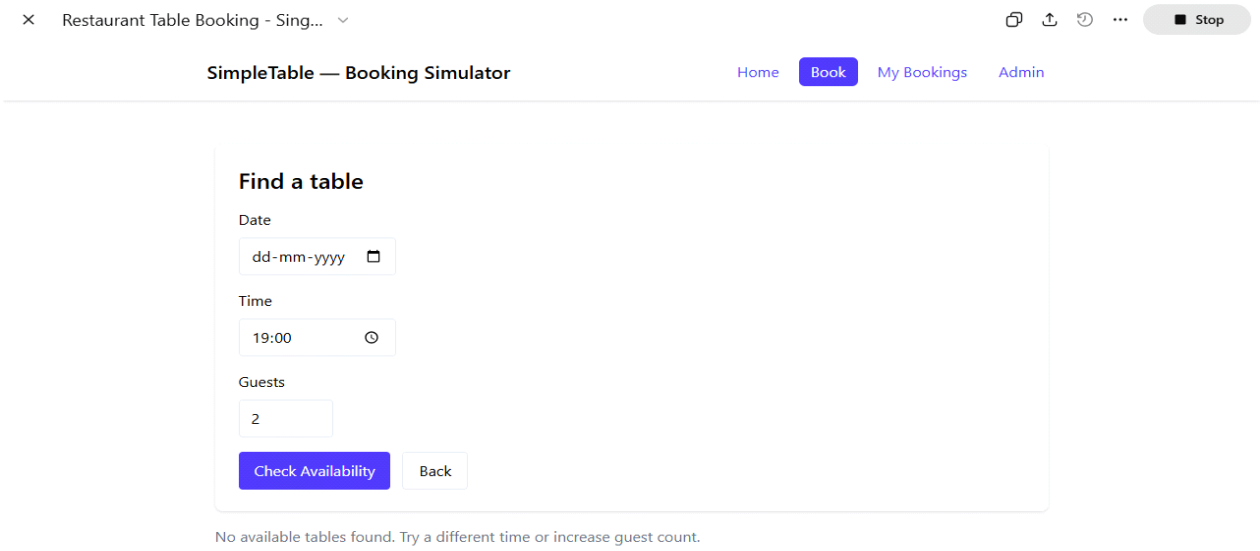
Booking Page → Date + Time + Guest Selection → Available Tables List

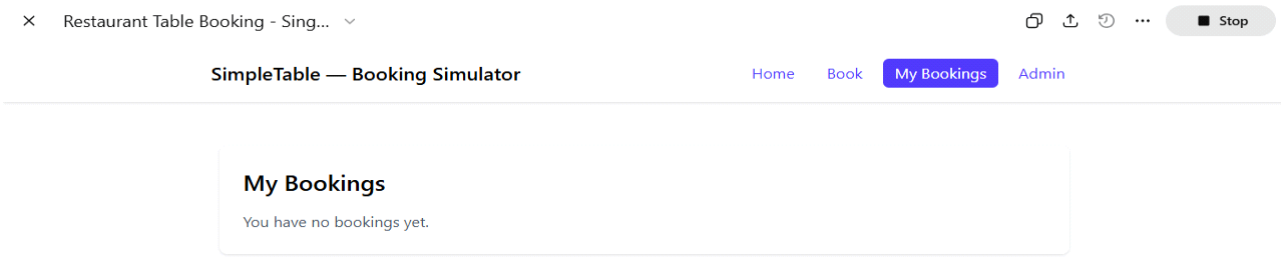
Confirmation Page → Booking Details + Success Message

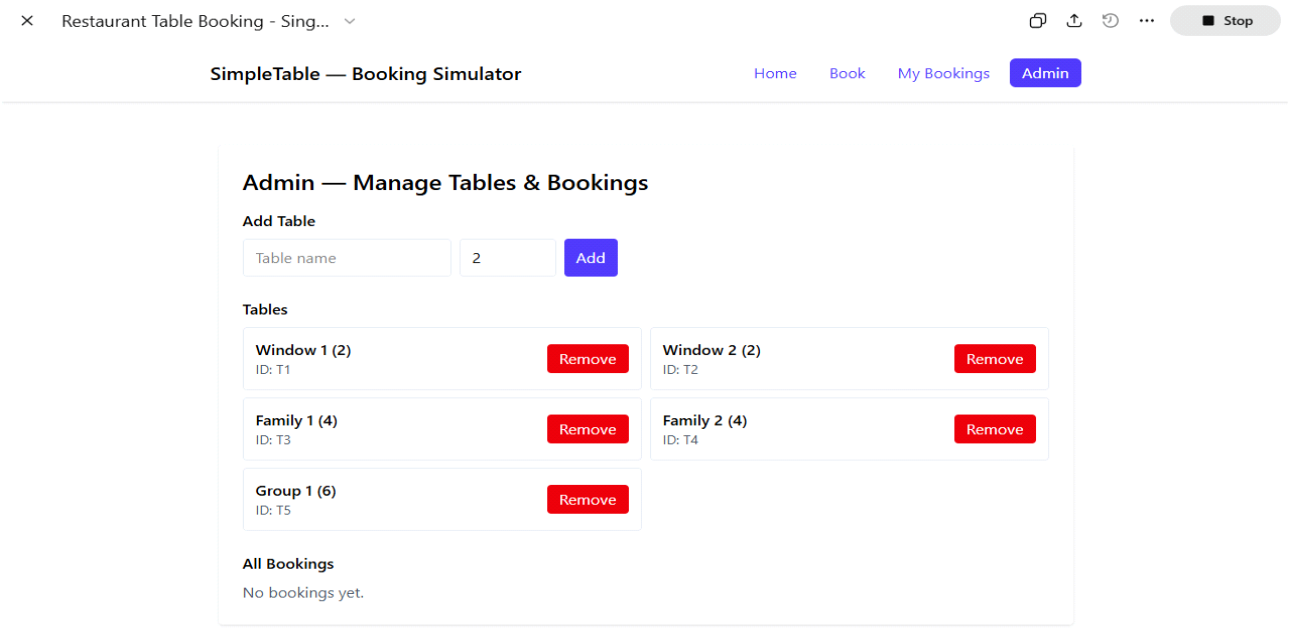
My Bookings → User Dashboard to View or Cancel Bookings

Admin Page (optional) → Manage Tables & Bookings









State Management Approach:

Local state (using React Hooks / Context API).

API integration for real-time data.

Form validation before submission.

Development & Deployment Plan

Team Roles:

Frontend Developer (UI Design & API integration)

Backend Developer (API creation & database setup)

Tester (QA & bug tracking)

Git Workflow:

main → stable builds

dev → feature branches (feature/booking, feature/auth)

Testing Approach:

Unit testing for API routes

UI testing for form validation and booking flow

Integration testing between frontend and backend

Hosting / Deployment Strategy:

Backend on Render / Railway / Vercel (Serverless Functions)

Frontend on Netlify / Vercel

Environment variables for API keys and DB URIs