

# Project Journal

## BookTable - Restaurant Reservation System

**Sprint Duration:** March 23 – May 11, 2025

**Team Members:** Syedanida, Rutuja, Aishly, Harsha

### Sprint 1: March 23 – March 29

#### Tasks Completed

- Syedanida: Proposed initial user stories and feature list
- Rutuja: Evaluated and finalized tech stack (React, FastAPI, PostgreSQL)
- Aishly: Created GitHub repository and set up team permissions
- Harsha: Designed initial UI wireframes for Search, Booking, Admin, and Manager screens

#### Next Tasks

- Initialize project codebases for frontend and backend
- Set up database schema
- Begin building core page routes

#### Blocked Tasks

- Delay in confirming UI layout direction from all team members
- Final decision on database deployment (local vs cloud)

## **Sprint 2: March 30 – April 5**

### **Tasks Completed**

- Syedanida: Developed UI for Login and Register pages
- Rutuja: Set up AuthContext and JWT token handling in frontend
- Aishly: Implemented backend login and signup endpoints using FastAPI
- Harsha: Built Home page with form for searching restaurants

### **Next Tasks**

- Implement Booking page functionality
- Create database models for users, restaurants, and reservations
- Begin integrating Google Maps API

### **Blocked Tasks**

- Backend schema not finalized for reservations
- Role definitions for user types not fully agreed on

## **Sprint 3: April 6 – April 12**

### **Tasks Completed**

- Syedanida: Built Booking page and reservation confirmation flow
- Rutuja: Connected React frontend to FastAPI endpoints for auth

- Aishly: Completed review and reservation models using SQLAlchemy
- Harsha: Set up Google Maps integration for displaying locations

#### **Next Tasks**

- Build restaurant listing API
- Implement Manager Dashboard features
- Connect SendGrid/Twilio for confirmations

#### **Blocked Tasks**

- Final definition of availability logic
- Issues with Google Maps API quota limits during testing

## **Sprint 4: April 13 – April 19**

#### **Tasks Completed**

- Syedanida: Built Manager Dashboard UI for adding and editing listings
- Rutuja: Developed backend logic for managing restaurant availability
- Aishly: Created APIs for writing and retrieving restaurant reviews
- Harsha: Finalized models for availability and tables

#### **Next Tasks**

- Build Admin dashboard for approvals and analytics
- Complete integration of review features
- Validate role-based access control

### **Blocked Tasks**

- None

## **Sprint 5: April 20 – April 26**

### **Tasks Completed**

- Syedanida: Integrated booking search with real-time availability
- Rutuja: Enabled advanced filters (cuisine, ratings, cost)
- Aishly: Developed approval/rejection logic for Admin
- Harsha: Completed Review form with submission and validation

### **Next Tasks**

- Test Admin dashboard and analytics
- Start backend deployment setup on EC2
- Finalize email/SMS notification flows

### **Blocked Tasks**

- Delay in acquiring SendGrid API credentials
- Admin dashboard charts not rendering properly in production build

## **Sprint 6: April 27 – May 3**

### **Tasks Completed**

- Syedanida: Polished Admin and Manager dashboards
- Rutuja: Refined reservation cancellation logic and UI
- Aishly: Wrote integration tests for role-based access and API workflows

- Harsha: Finalized map integration and testing for booking confirmation page

#### **Next Tasks**

- Deploy backend to AWS EC2 with Load Balancer
- Containerize and deploy database
- Begin final demo preparation

#### **Blocked Tasks**

- EC2 scaling configuration pending for testing
- PostgreSQL container connection issues during staging deployment

## **Sprint 7: May 4 – May 11**

#### **Tasks Completed**

- Syedanida: Deployed React frontend to EC2 and verified production routes
- Rutuja: Deployed backend with PostgreSQL and completed environment setup
- Aishly: Performed full application testing (booking, admin, manager workflows)
- Harsha: Finalized diagrams (component, deployment, and wireframes) and packaged documentation

#### **Next Tasks**

- Conduct live team demo
- Submit project repository and final report

#### **Blocked Tasks**

- None