FLIGHTFINDER PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

The project titled **"FlightFinder – Smart Flight Booking System"** aims to simplify the process of searching and booking flights through an online platform. The system allows users to register, search flights by source, destination, and date, book tickets, and manage their reservations efficiently.

Administrators can manage flight details, schedules, and seat availability through a dedicated dashboard.

This system eliminates the need to check multiple airline websites, thereby saving time and improving booking efficiency.

1.2 Purpose

The purpose of this project is to provide a user-friendly digital solution that connects travellers with flight services efficiently. It ensures:

- Easy and fast flight search
- Reduced booking errors
- Better seat availability management
- Secure and centralized booking system
- Improved travel planning experience

2. IDEATION PHASE

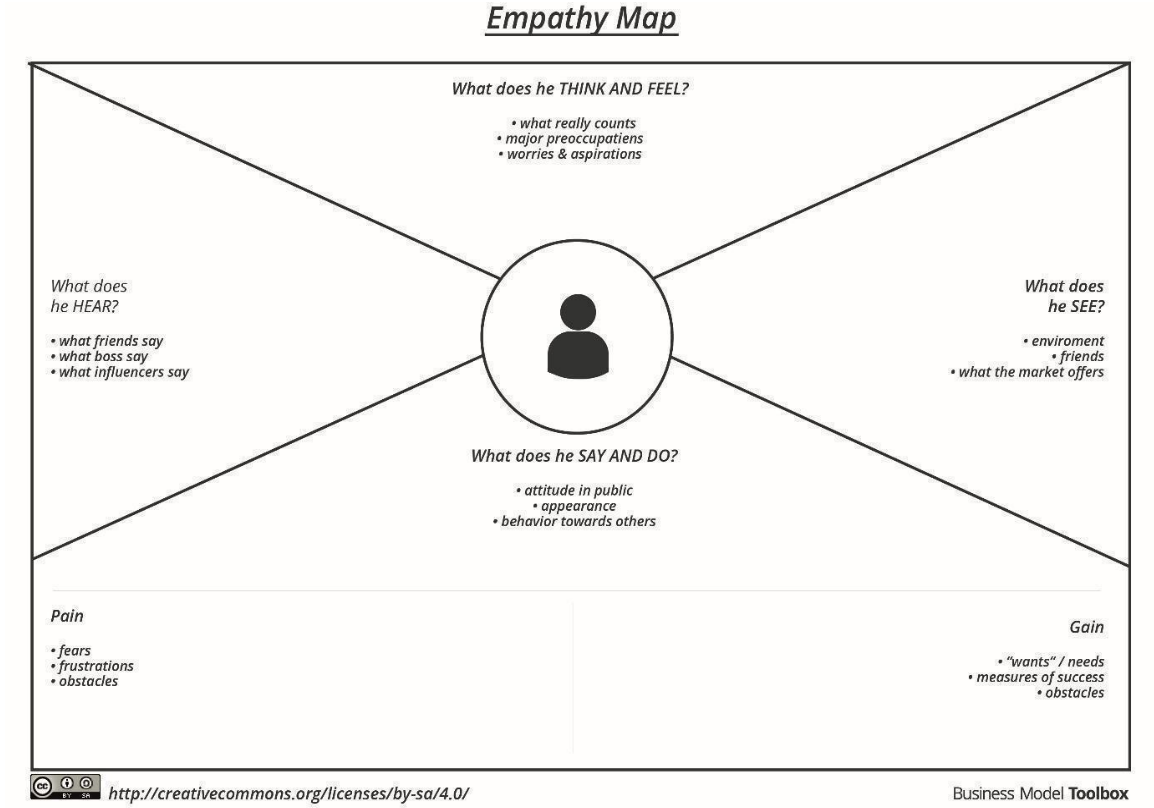
2.1 Problem Statement

Travelers face difficulties when searching for flights due to scattered information across platforms, inconsistent pricing, and lack of real-time seat availability.

How Might We:

How might we allow users to search and book flights quickly and securely from one centralized platform?

2.2 Empathy Map Canvas



http://creativecommons.org/licenses/by-sa/4.0/ Business Model Toolbox

Says:

“I wish I could compare flights easily in one place.” Thinks:

“Why do I need to visit multiple websites to find the best price?” Feels:

“Confused, frustrated, worried about overbooking”

Does:

“Checks multiple airline websites, compares manually, worries about seat availability”

2.3 Brainstorming

During brainstorming, the team explored ideas such as:

- Real-time seat availability updates
- Price filtering and sorting
- Role-based admin dashboard
- Booking history management

- Future payment gateway integration
- Mobile-first responsive design

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The customer journey includes:

- Awareness – User learns about the platform
- Registration/Login – User creates account or logs in
- Flight Search – Filters by source, destination, and date
- Booking – Selects flight and confirms seat booking
- Confirmation – Booking confirmation displayed
- Booking Management – View or cancel bookings

3.2 Solution Requirement

Functional Requirements

- User Registration and Login
- Flight Search by Filters
- Book/Cancel Flights
- View Booking History
- Admin Dashboard for Flight Management ● Seat Availability Management

Non-Functional Requirements

- Fast loading time (<2 seconds)
- Secure JWT-based authentication
- Mobile-responsive design
- High availability and scalability ● Secure database storage

3.3 Data Flow Diagram Level

0 DFD:

External Entities:
User, Admin

Processes:
Registration, Login, Flight Search, Booking, Seat Update

Data Stores:
Users DB, Flights DB, Bookings DB

3.4 Technology Stack

- Frontend: React.js
- Backend: Node.js + Express.js
- Database: MongoDB
- Authentication: JWT
- Hosting: MongoDB Atlas / Cloud Deployment

4. PROJECT DESIGN

4.1 Problem-Solution Fit

The proposed solution addresses the problem of scattered and inefficient flight booking systems by providing a centralized, secure, and scalable booking platform accessible from any device.

4.2 Proposed Solution

A web application where:

- Users can search, book, and manage flights
- Admins can create, update, and delete flights
- Seat availability updates automatically after booking
- Secure login and role-based access control

4.3 Solution Architecture

Architecture Type: **Client-Server Model (3-Tier Architecture)**

Layers:

- Presentation Layer – React-based UI
- Business Logic Layer – RESTful APIs (Node.js & Express)
- Data Layer – MongoDB Database

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Agile Methodology was used with sprints of 6 days each:

Sprint-1: User Authentication & Database Setup
Sprint-2: Flight Search & Listing

Sprint-3: Booking System Implementation
Sprint-4: Admin Dashboard & Testing

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing Tools

Used:

- Postman for API testing ● JMeter (optional) for load testing **Test Scenarios:**

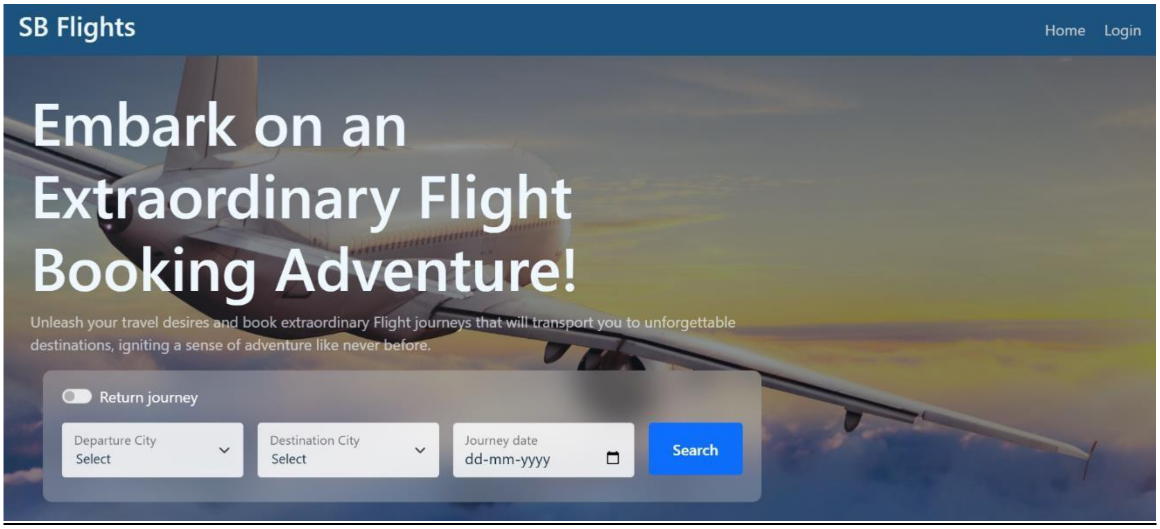
- Simulated multiple concurrent users
- Measured API response time (<1.5 sec)
- Verified seat update consistency

7. RESULTS

- Application handled concurrent bookings smoothly
- Average response time: ~1.2 seconds ● No major crashes or data inconsistencies

7.1 Output Screenshots:

HOME PAGE :



Embark on an Extraordinary Flight Booking Adventure!

Unleash your travel desires and book extraordinary Flight journeys that will transport you to unforgettable destinations, igniting a sense of adventure like never before.

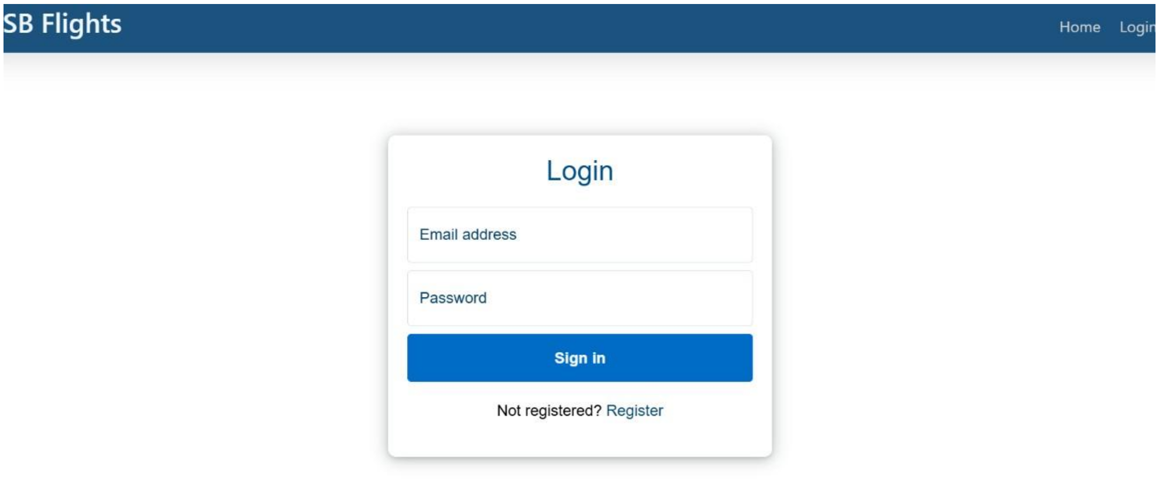
Return journey

Departure City: Select

Destination City: Select

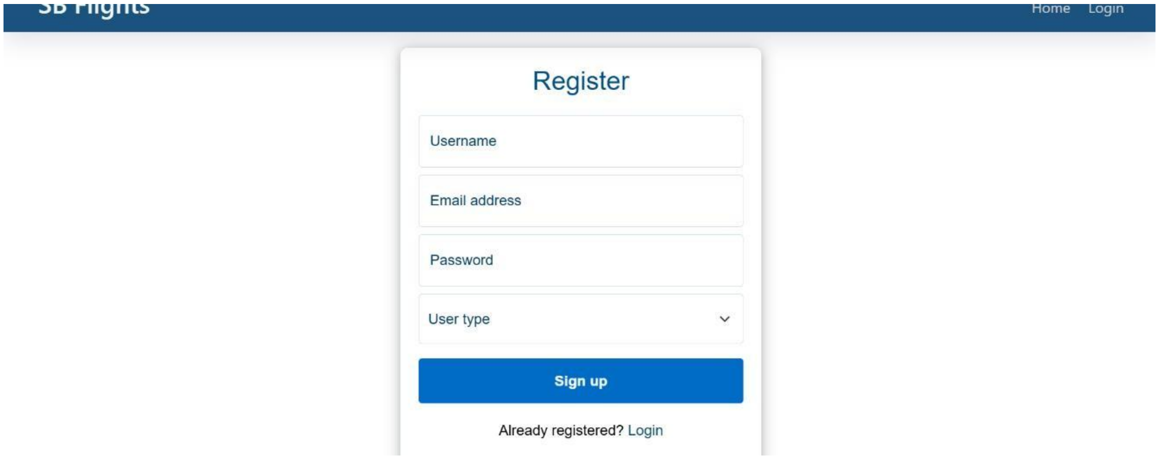
Journey date: dd-mm-yyyy

LOGIN PAGE:



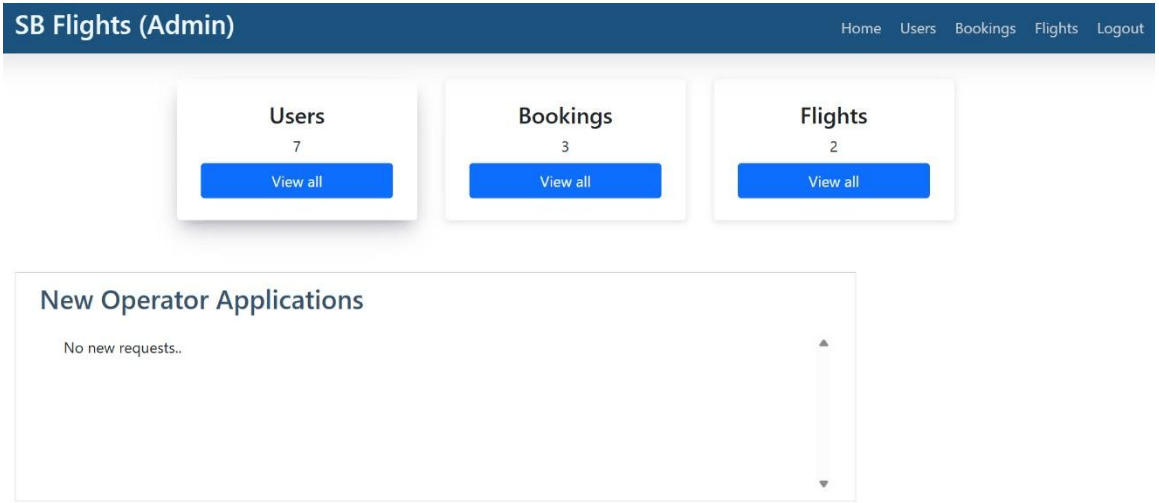
Not registered? Register

REGISTER PAGE :



Already registered? Login

DASHBOARD:



Users	Bookings	Flights
7	3	2

New Operator Applications

No new requests..

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Saves time
- Centralized flight comparison
- Secure authentication
- Easy booking management ● Scalable architecture Disadvantages:

- Requires internet connection
- Payment integration not yet implemented
- Admin management required for flight updates

9. CONCLUSION

The FlightFinder project successfully provides a digital solution for searching and booking flights efficiently. By leveraging the MERN stack and Agile development practices, the system improves user convenience, reduces booking complexity, and ensures secure and scalable travel management.

10. FUTURE SCOPE

- Payment gateway integration
- Email/SMS notifications
- Real-time seat updates using WebSockets
- Hotel and travel package booking
- Mobile application development
- Multi-language and multi-currency support

11. APPENDIX Source

Code:

https://github.com/siva0523ka/Flight_finder