**TravelSphere - Project Report**

**1. INTRODUCTION**

**1.1 Project Overview**

TravelSphere is a group travel booking and planning system that simplifies trip coordination by allowing users to create, manage, and collaborate on travel plans efficiently. It leverages modern web technologies and a secure backend to provide a seamless user experience.

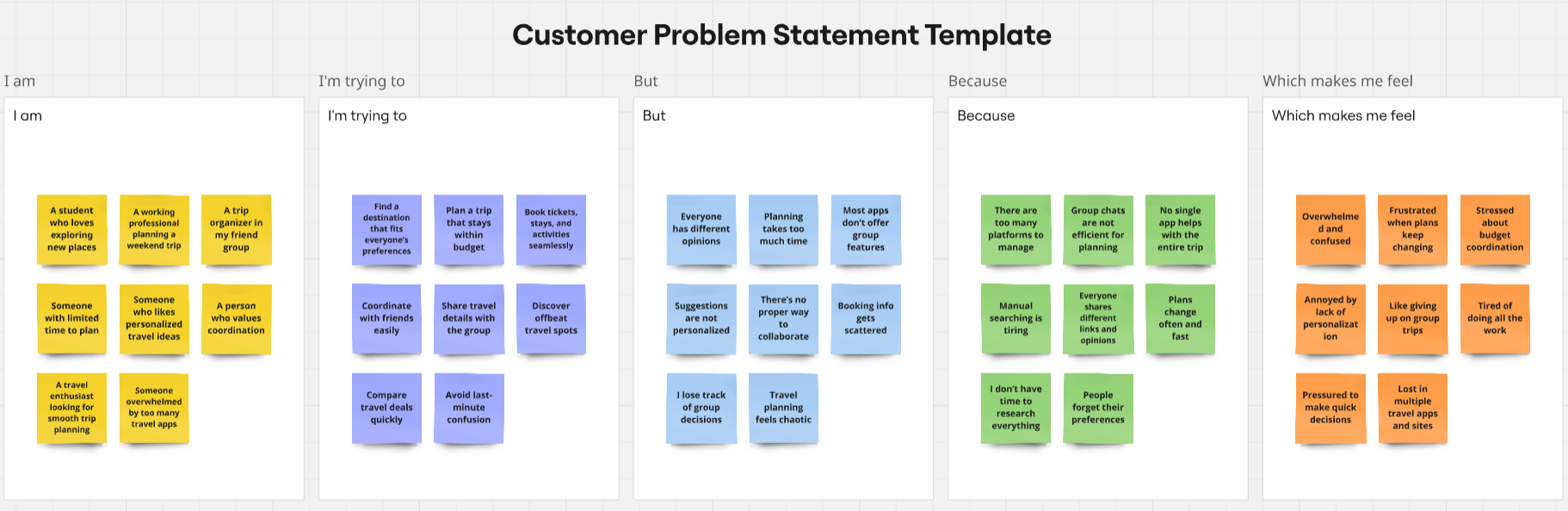
**1.2 Purpose**

The purpose of TravelSphere is to streamline group travel planning by solving common coordination issues, such as itinerary management, budget tracking, and trip approvals. It enables users to manage every aspect of a trip, from choosing destinations to organizing shared expenses.

**2. IDEATION PHASE**

**2.1 Problem Statement**

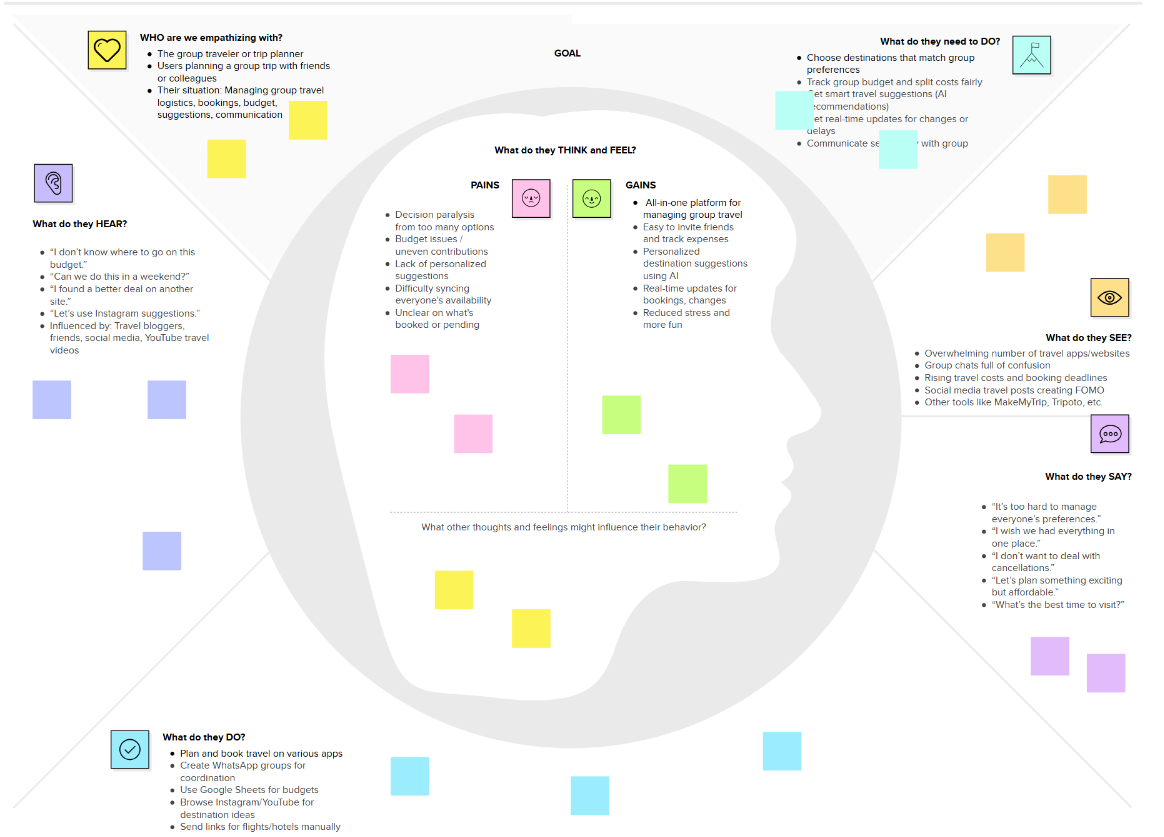
Planning group trips is often challenging due to miscommunication, scheduling conflicts, and budget constraints. TravelSphere aims to resolve these issues by providing a collaborative platform for easy trip planning and real-time coordination.



Link: <https://miro.com/app/board/uXjVIG-A1lw=/?share_link_id=66224704791>

**2.2 Empathy Map Canvas**

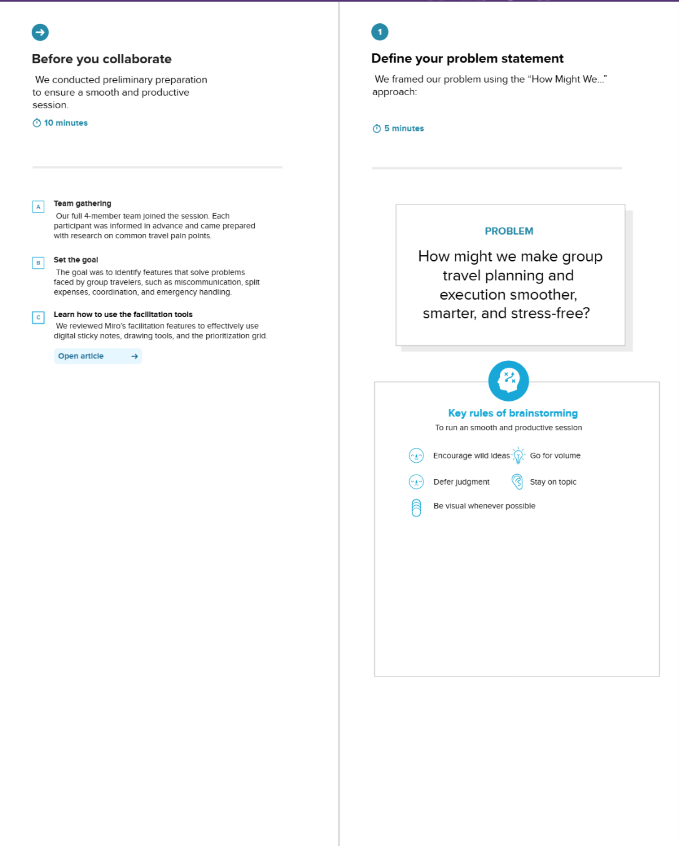
Understanding user pain points, needs, and motivations was key in designing TravelSphere. The empathy map helped identify issues such as difficulty in reaching a consensus, budget conflicts, and lack of real-time updates.

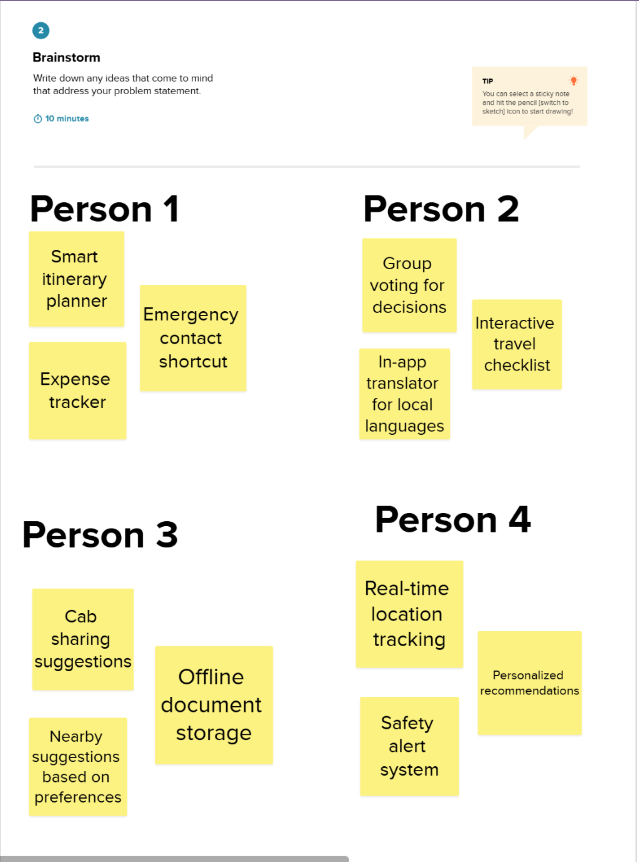


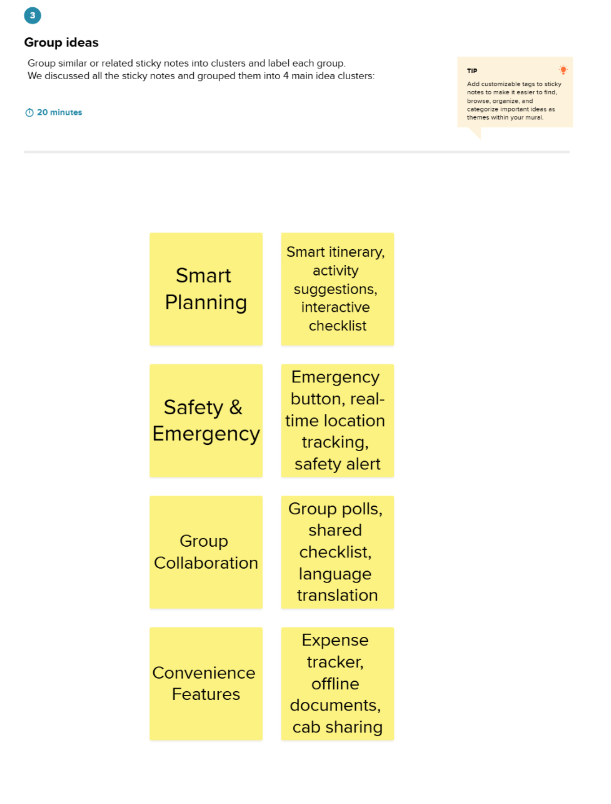
Link: <https://app.mural.co/t/travelgram7894/m/travelgram7894/1743874709113/b621784a8887f7365db647b7df481d30465e0d0d?sender=u1e90317ebed16c4cd93d8252>

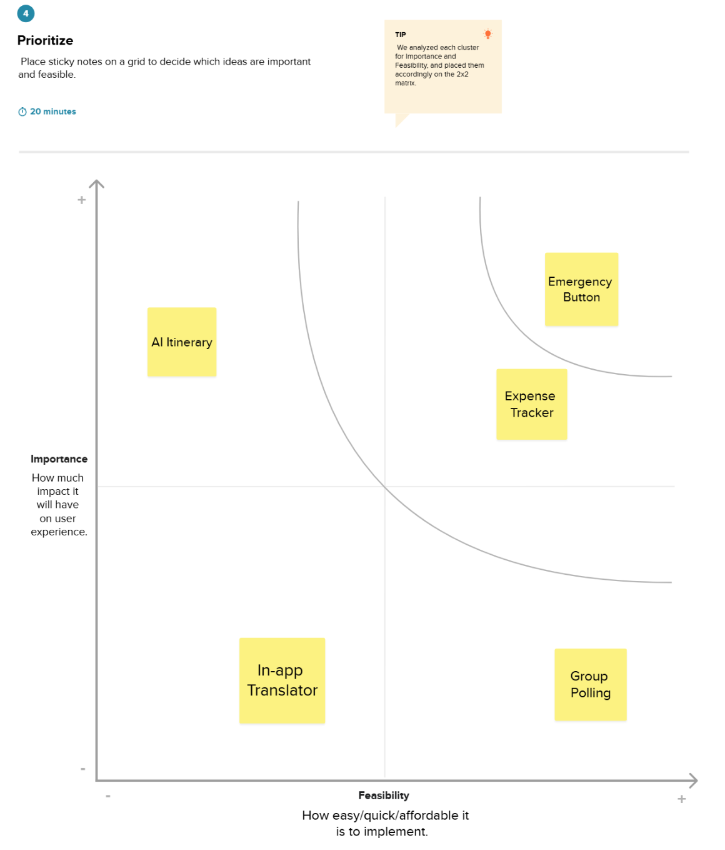
**2.3 Brainstorming**

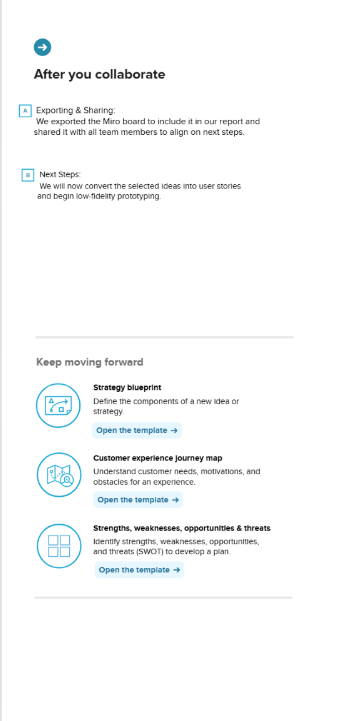
The brainstorming phase led to key feature decisions, including real-time chat, expense tracking, shared itineraries, and notification alerts.











<https://app.mural.co/t/travelgram7894/m/travelgram7894/1743877900278/751a3f6ee41620ef47143f24156c49249234ca59?sender=u1e90317ebed16c4cd93d8252>

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

Mapping out the typical user journey ensured a smooth experience from sign-up to trip execution. The stages include registration, trip creation, invitation of friends, itinerary building, and execution.

**3.2 Solution Requirements**

**Functional Requirements**

* User authentication (signup/login)
* Trip creation and management
* Expense tracking and budget management
* Shared itineraries and notifications

**Non-Functional Requirements**

* High availability and performance
* Secure data handling and privacy measures
* Scalability to handle multiple users concurrently

**3.3 Data Flow Diagram (DFD)**

The system follows a structured data flow where users interact with the frontend, which sends requests to the backend, and data is fetched from the database.

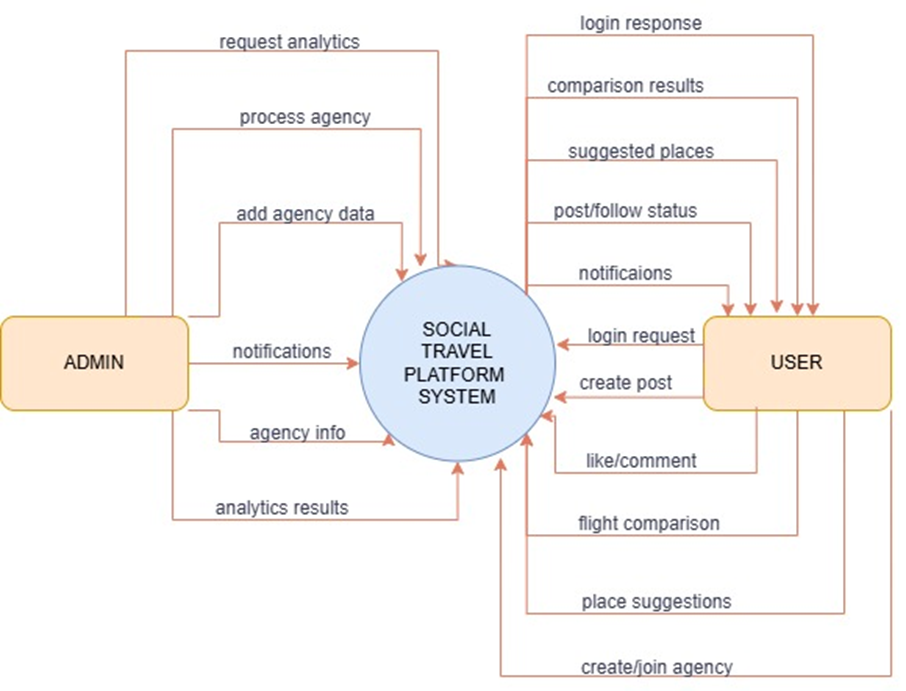


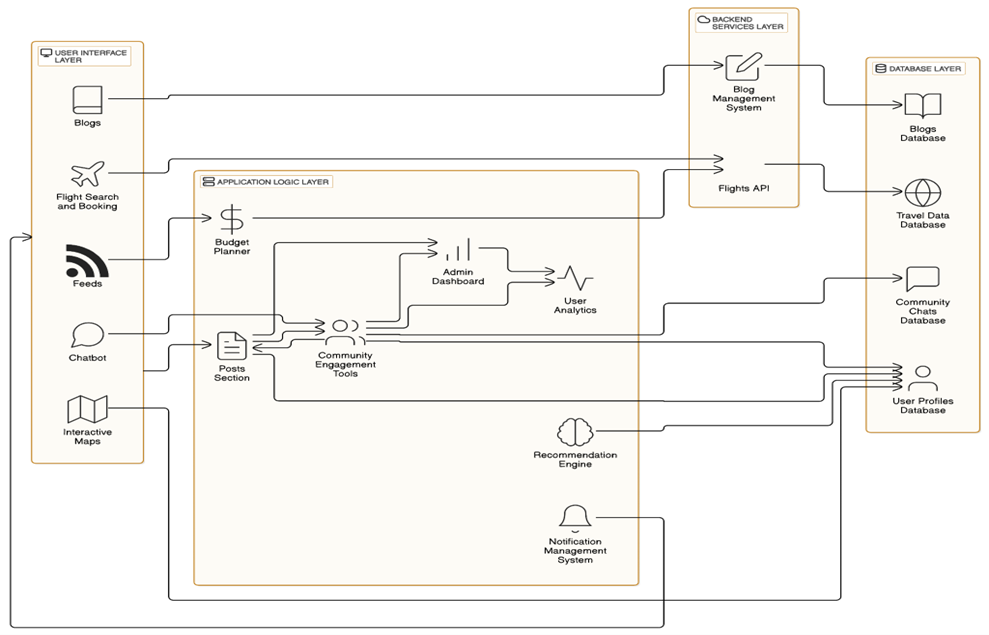
Fig: Data Flow Diagram

**3.4 Technology Stack**

Frontend: React.js, Redux, Material-UI, Axios

Backend: Spring Boot, MongoDB, Redis

Communication: WebSockets for real-time updates

****

**Fig: Layered Architecture of TravelSphere Application**

**4. PROJECT DESIGN**

**4.1 Problem-Solution Fit**

TravelSphere directly addresses the challenges of group trip coordination by providing a structured platform for shared itineraries, budget tracking, and collaborative decision-making.

**4.2 Proposed Solution**

A web-based platform with the following features:

* Trip creation & invitations
* Expense splitting & tracking
* Real-time collaboration & notifications

**4.3 Solution Architecture**

A three-tier architecture consisting of:

1. **Frontend** (React.js) - User interface and interactions
2. **Backend** (Spring Boot) - Business logic & API management
3. **Database** (MongoDB) - Stores user data, trips, and expenses

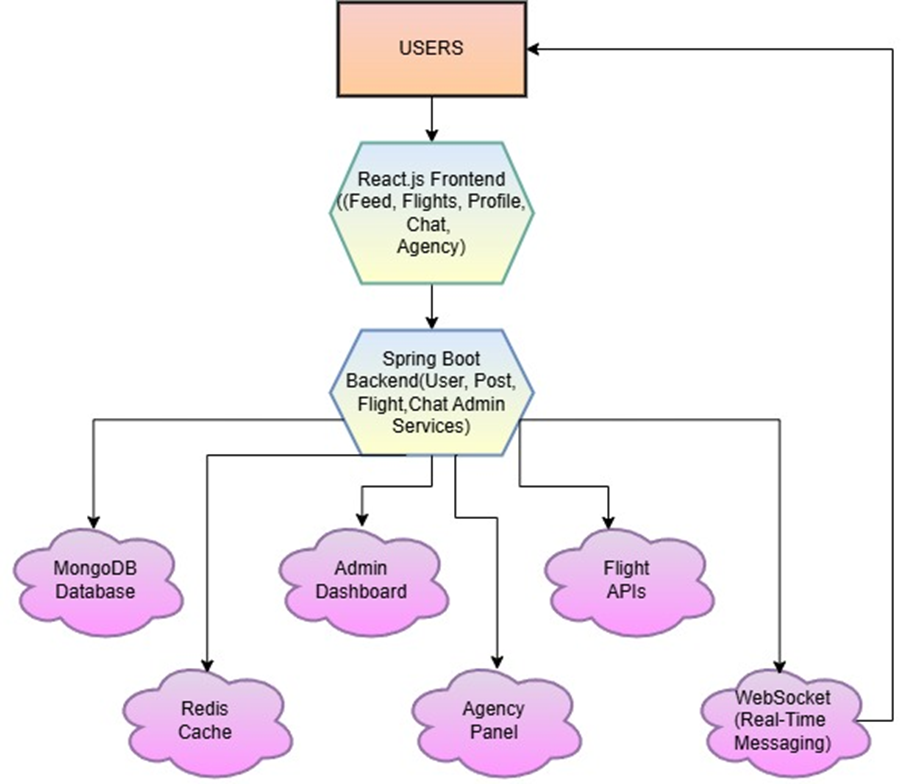


Fig: System Architecture Diagram

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

The project followed an agile methodology with iterative development cycles. Milestones included:

* Week 1-2: Requirement gathering & wireframing
* Week 3-4: Backend and frontend setup
* Week 5-6: Feature development & integration
* Week 7-8: Testing and deployment

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

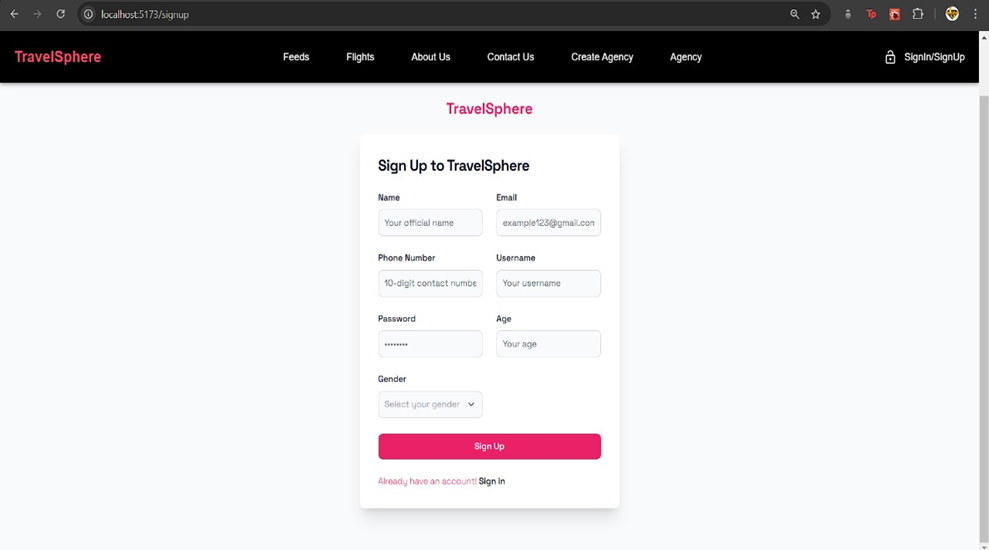
The system was tested for:

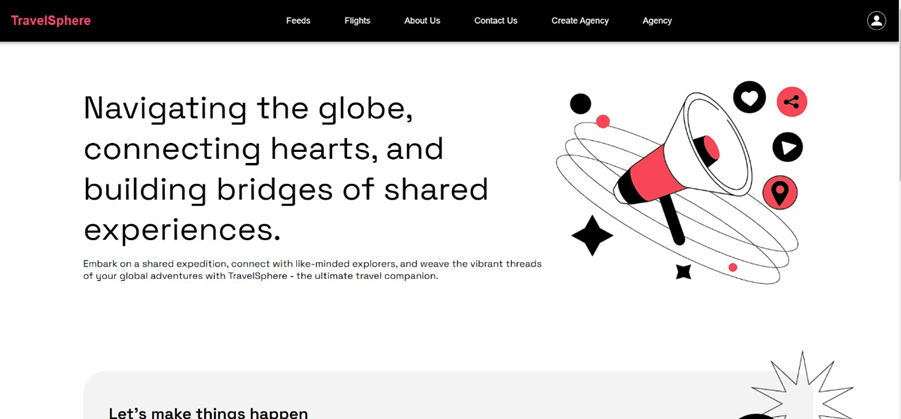
* Load Handling: Checking system response under concurrent users
* Response Time: Optimizing API calls for quick responses
* Data Security: Ensuring encrypted data transfer and JWT-based authentication

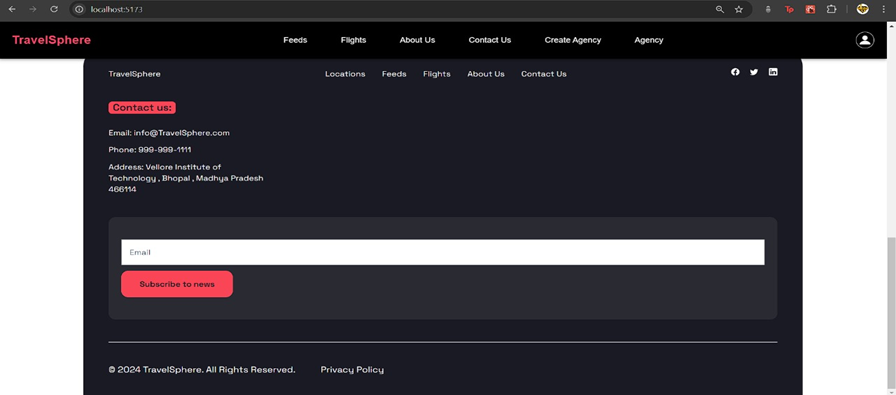
**7. RESULTS**

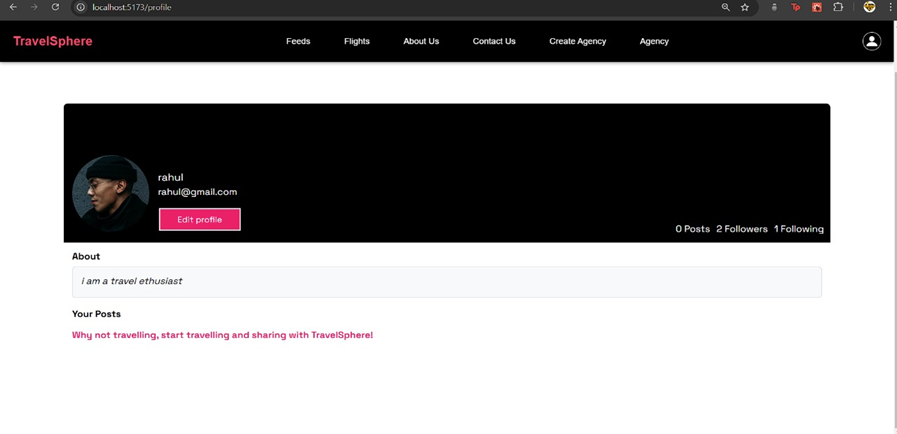
**7.1 Output Screenshots**

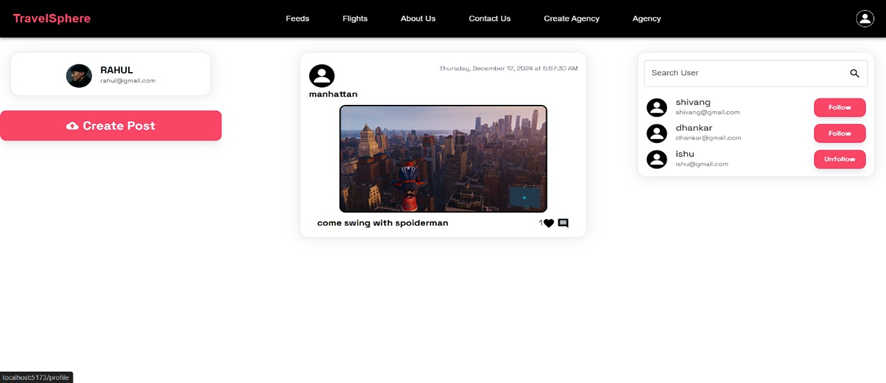
* User registration & login pages
* Trip creation interface
* Expense tracking dashboard
* Real-time chat & notifications

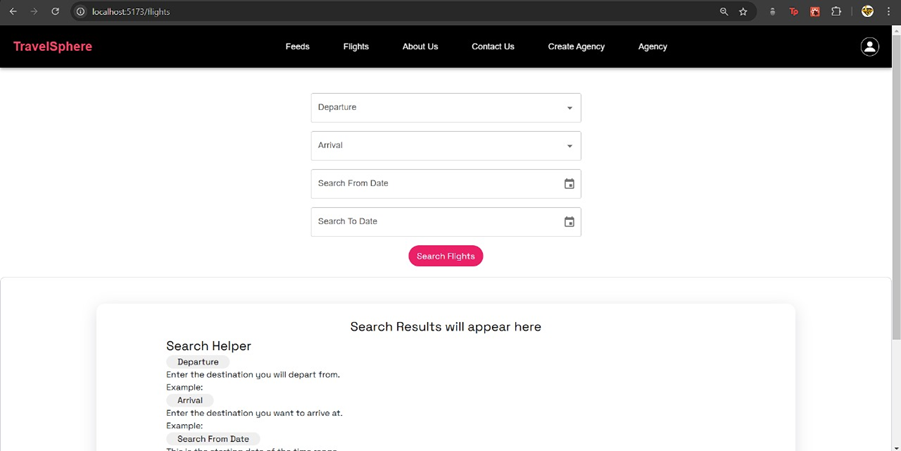
****

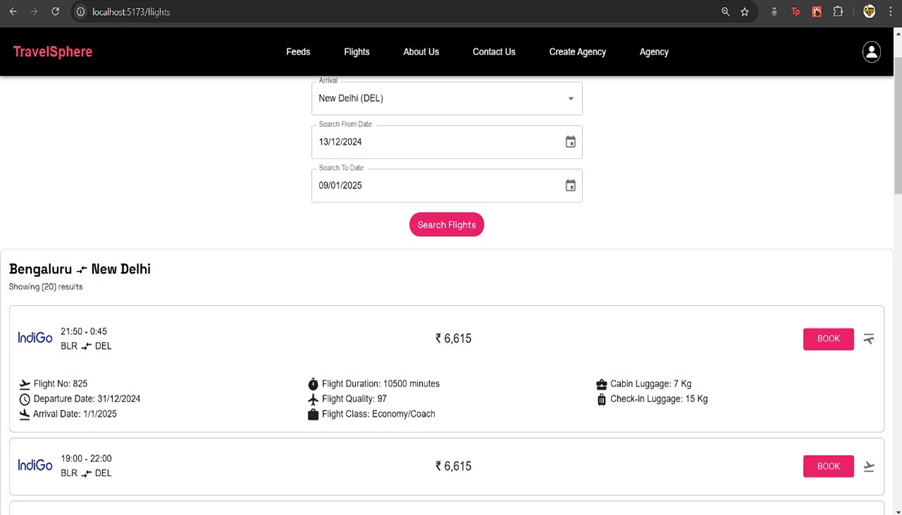
****

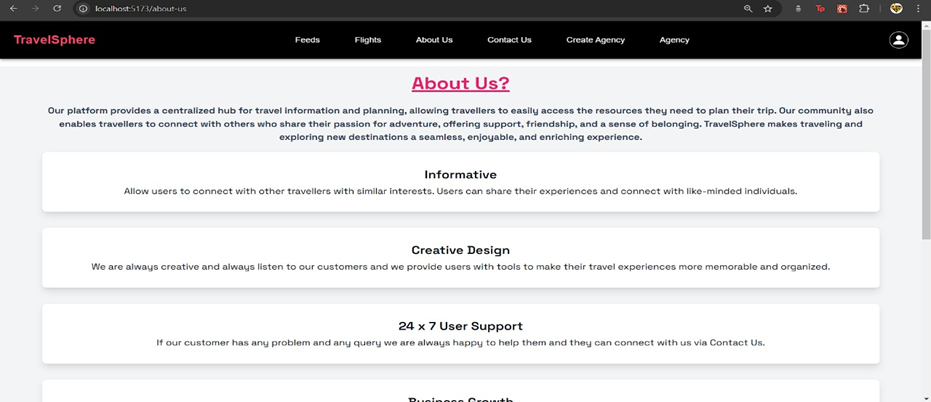
****

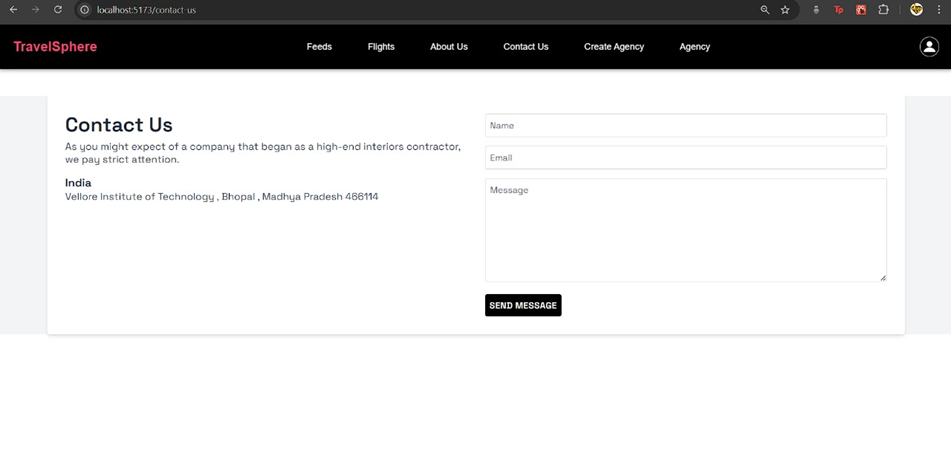
****

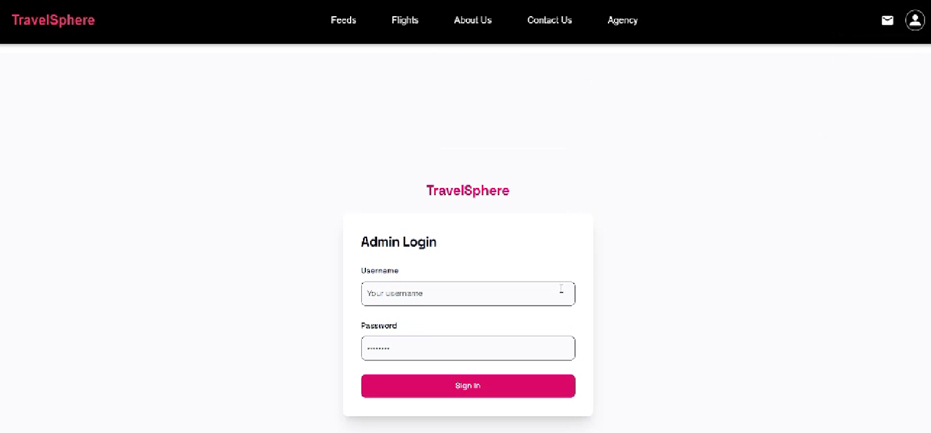
****

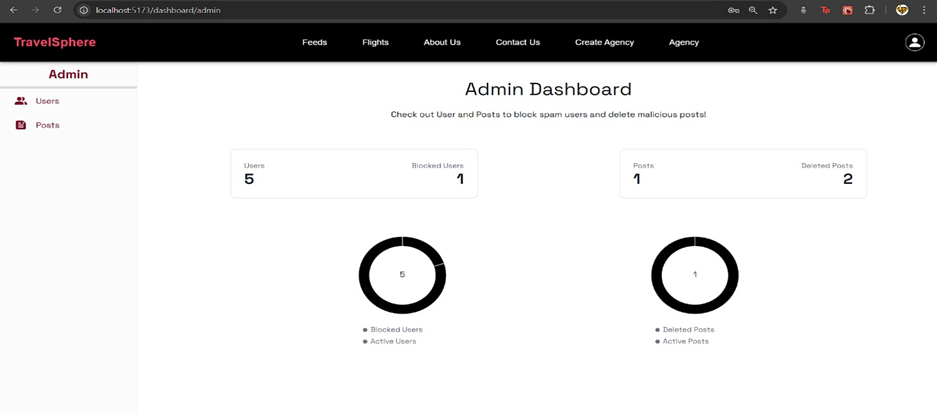
****

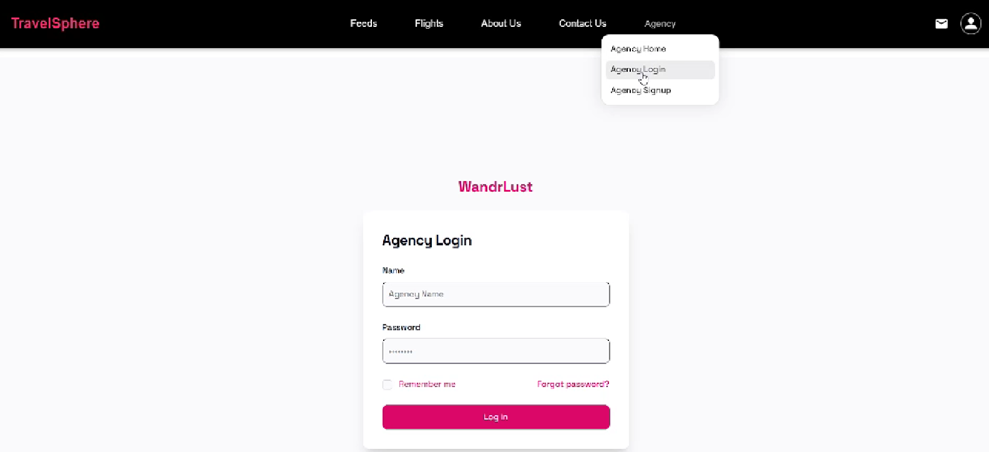
****

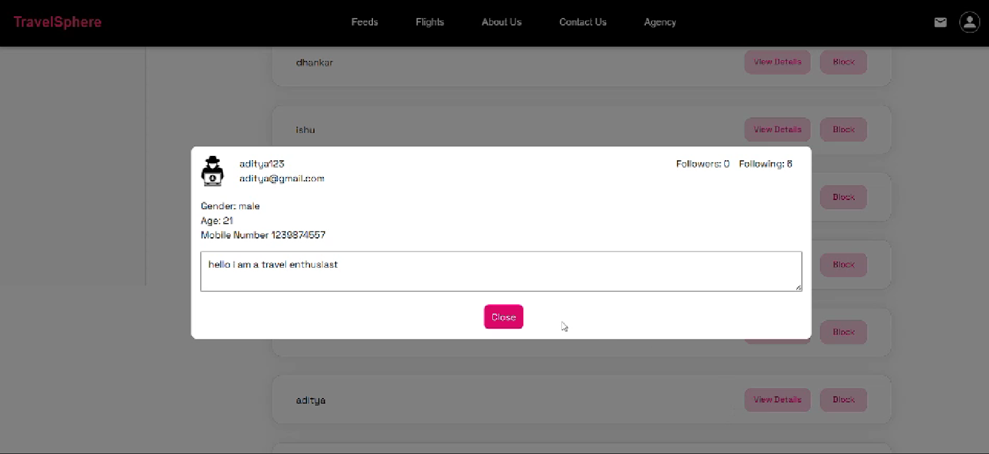
****

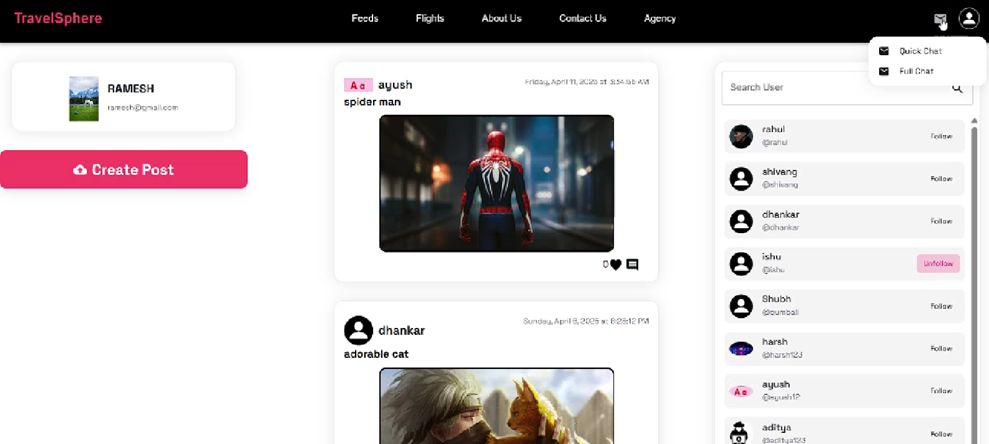
****

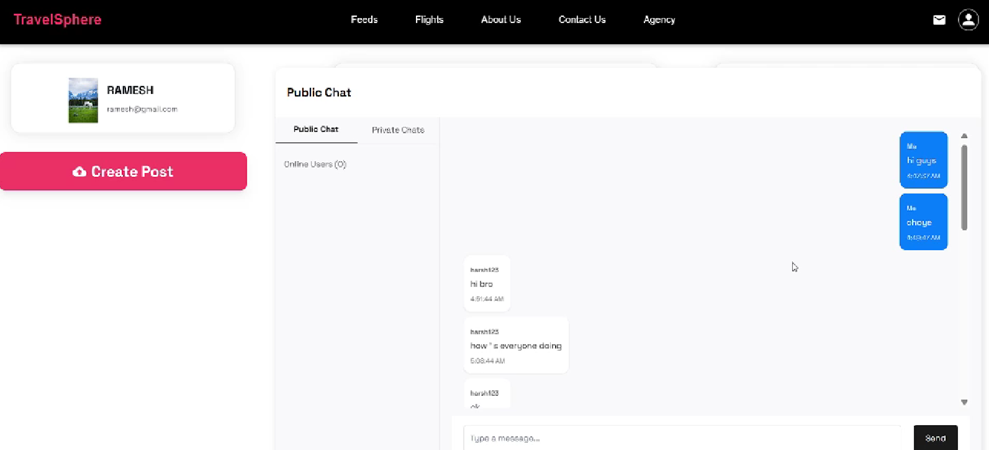
****

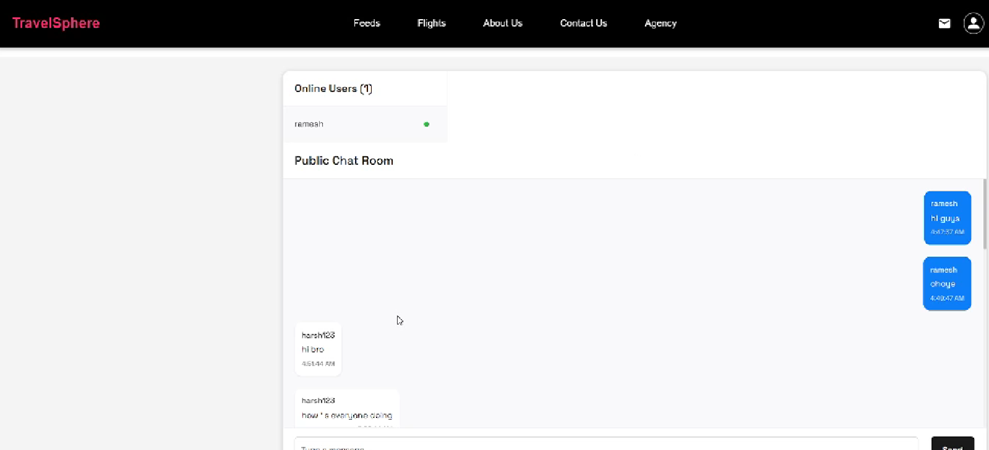
****

****

****

****

****

****

**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

* Seamless trip planning & collaboration
* Real-time updates for better coordination
* Secure & scalable backend infrastructure

**Disadvantages**

* Dependent on internet connectivity
* Requires user adoption for full efficiency

**9. CONCLUSION**

TravelSphere successfully provides a user-friendly, efficient, and scalable solution for group trip planning. It simplifies travel coordination by offering a single platform for budgeting, scheduling, and communication.

**10. FUTURE SCOPE**

* Integration with third-party booking APIs (hotels, flights, etc.)
* Mobile app development for a better user experience
* Enhanced analytics for personalized travel suggestions

**11. APPENDIX**

**GitHub & Project Demo Link**

* Live Project Demo: [Link]
* **GitHub Link: https://github.com/WebSorcerer10/travelsphere2.0**