

A Project Report on
FLIGHT FINDER: NAVIGATING YOUR AIR TRAVEL OPTIONS APPLICATION
USING FULLSTACK DEVELOPMENT WITH MONGODB
Industrial Internship Project report submitted in partial fulfillment of
the Requirements for the award of the degree in
BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING
TEAM MEMBERS

B BHAVANI	218X1A0449
K SUSHMA	218X1A0441
G AYYAPPA	218X1A0453
K HEMANTH KUMAR	218X1A0458
B LAVA KUMAR	218X1A0405

Under the Esteemed Guidance of :

K SRINIVASA RAO M.TECH

Assistant Professor



ELECTRONICS AND COMMUNICATION ENGINEERING
KALLAM HARANADHAREDDY INSTITUTE TECHNOLOGY
(AUTONOMOUS)

ACCREDITED BY NAAC WITH 'A' GRADE

(APPROVED BY AICTE, AFFILIATED TO JNTUK, KAKINADA)

NH-5, CHOWDAVARAM, GUNTUR – 522019.

2021 - 2025

**KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)**

ACCREDITED BY NAAC WITH 'A' GRADE

(APPROVED BY AICTE, AFFILIATED TO JNTUK, KAKINADA)

NH-5, CHOWDAVARAM, GUNTUR - 522019

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



CERTIFICATE

This is to certify that the Industrial Internship Project work entitled <**PROJECT TITLE**> being submitted by

B . BHAVANI	218X1A0449
K . SUSHMA	218X1A0441
G . AYYAPPA	218X1A0453
K . HEMANTH KUMAR	218X1A0458
B . LAVA KUMAR	218X1A0405

in partial fulfillment for the award of the Degree of Bachelor of Technology in Electronic and communication engineering in the Kallam Haranadhareddy Institute of Technology is a record of bonafide work carried out by them.

Internal Guide

K SRINIVASA RAO

Assistant Professor

Head of the Department

DR S SURYANARAYANA

Professor & HOD

External Examiner

CERTIFICATE FROM INTERN ORGANIZATION

This is to certify that **B.BHAVANI, K.SUSHMA, G.AYYAPPA, K HEMANTH KUMAR** and **B LAVA KUMAR** Reg.no **218X1A0449, 218X1A0441, 218X1A0453, 218X1A0458, 218X1A0405** of **Kallam Haranadhareddy Institute of Technology** underwent industrial internship in **SMART INTERNZ** (Name of the Intern Organization) from **23- 12-2024** to **7-03-2025**. The overall performance of the intern during his/her internship is found to be **Satisfactory**.

B BHAVANI

218X1A0449

K SUSHMA

218X1A0441

G AYYAPPA

218X1A0453

K HEMANTH KUMAR

218X1A0458

B LAVA KUMAR

218X1A0405

Authorized Signatory with Date and Seal

DECLARATION

We, **B.BHAVANI (218X1A0449), K.SUSHMA (218X1A0441), G.AYYAPPA (218X1A0453), K.HEMANTH KUMAR (218X1A0458), B.LAVA KUMAR (218X1A0405)**, hereby declare that the project report titled “**FLIGHT FINDER: NAVIGATING YOUR AIR TRAVEL OPTIONS APPLICATION USING FULLSTACK WITH MONGODB**” under the guidance of **K.SRINIVASA RAO M.TECH** is submitted in partial fulfilment of the requirements for the award of the degree of bachelor of technology in Computer Science and Engineering. This is a regard of bonafide work carried out by us and the results embodied in this project have not been submitted to any other university for the award of any other degree.

B . BHAVANI	218X1A0449
K . SUSHMA	218X1A0441
G . AYYAPPA	218X5A0453
K . HEMANTH KUMAR	218X1A0458
B . LAVA KUMAR	218X1A0405

ACKNOWLEDGEMENT

We profoundly grateful to express our deep sense of gratitude and respect towards **Sri. KALLAM MOHAN REDDY, Chairman, KHIT, Guntur** for his precious support in the college.

We are thankful to **Dr. M. UMA SANKAR REDDY**, Director, KHIT, Guntur for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B. SIVA BASIVI REDDY**, Principal, KHIT, Guntur for his support during and until the completion of the project.

We are greatly indebted to **Dr. SURYANARAYANA, M.Tech, Ph.D** Professor & Head of the department, Computer Science and Engineering, KHIT, Guntur for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college during Industry Internship.

We are also thankful to our Project Coordinator **Mr. G MANTHRU NAIK** who helped us in each step of our Project.

We extend our deep sense of gratitude to our Internal Guide **K. SRINIVASA RAO M.TECH**, Assistant Professor and other Faculty Members & Support staff for their valuable suggestions, guidance and constructive ideas in each and every step, which was indeed of great help towards the successful completion of our project.

ABSTRACT

This project presents a comprehensive flight booking management system designed to streamline the process of booking flights for users, flight management for companies, and administrative oversight. The system incorporates three primary roles: User, Company, and Admin. Users can register, login, search for flights, book or cancel reservations, view their profile and bookings, and log out. Companies are provided functionalities to register, login, manage flights through CRUD operations, view bookings, and log out. The Admin role oversees the approval of companies by accepting or rejecting requests, manages users and flight details, and ensures smooth operation of the system. This platform is designed to enhance the efficiency of flight booking processes while maintaining administrative control, making it a robust solution for both customers and service providers.

Keywords:Flight booking, User management, CRUD operations, Company registration, Admin control

TABLE OF CONTENT

Title	Page No
Abstract	9
Chapter-1 INTRODUCTION	13
1.1 Project Introduction	14
1.1 Problem Statement.....	14
1.2 Objective of the Project.....	14
1.3 Motivation	14
1.4 Scope.....	15
Chapter-2 LITERATURE SURVEY.....	16
2.1 Related Work.....	17
Chapter-3 SYSTEM ANALYSIS.....	19
3.1 Existing System.....	20
3.2 Disadvantages.....	20
3.3 Proposed System	21
3.4 Advantages	21
3.5 work Flow of Proposed system	22
Chapter-4 REQUIREMENT ANALYSIS	23
4.1 Function and nonfunctional requirements.....	24
4.2 Hardware Requirements.....	27
4.3 Software Requirements	27
4.4 Architecture.....	27
Chapter-5 SYSTEM DESIGN.....	28
5.1 Introduction of Input design.....	29

5.2 UML Diagram(class, use case, sequence, collaborative, deployment, activity, ER diagram and Component diagram).....	30
5.3 Data Flow Diagram.....	37
Chapter-6 IMPLEMENTATION.....	38
6.1 Modules.....	39
6.2 SourceCode	40
Chapter-7 TESTING AND RESULTS	42
7.1 System Study and Result	43
7.1.1 Feasibility study.....	43
7.1.2 Types of test	44
7.1.3 Test Case	47
7.2 Result.....	51
Chapter-8 CONCLUSION	62
Chapter-9 REFERENCES	64

List of Figures

Fig : Work Flow Of Proposed System.....	22
Fig :4 .4 Architecture.....	27
Fig : 5.2.1 Use Case Diagram... ..	31
Fig : 5.2.2 Class Diagram... ..	32
Fig : 5.2.3 Sequence Diagram... ..	33
Fig : 5.2.5 Deployment Diagram... ..	34
Fig : 5.2.6 Activity Diagram... ..	35
Fig : 5.2.7 Component Diagram... ..	36
Fig : 5.2.8 ER Diagram.....	37
Fig :7.2.1 Home Page	51
Fig : 7.2.2 Customer registration Page	52
Fig: 7.2.3 Login Page	53
Fig : 7.2.4 Admin Page.....	53
Fig: 7.2.5 View Registered Customer details... ..	54
Fig : 7.2.6 View Flight Details... ..	54
Fig : 7.2.7 Company Registration Page.....	55
Fig: 7.2.8 Company Login Page	55
Fig : 7.2.9 Flight Adding Page.....	56
Fig : 7.2.10 View Added Flight Details.....	56
Fig : 7.2.11 View Booked Flight Details... ..	57
Fig : 7.2.12Booked Customer Details in PDF	57
Fig : 7.2.13 After Customer Login... ..	58
Fig : 7.2.14 After Booking.....	58
Fig: 7.2.15 Profile Page and Change Password Page.....	59
Fig : 7.2.16 Price Fluctuation Notification... ..	59
Fig : 7.2.17 Alternate Flight Suggestions... ..	60
Fig : 7.2.18 Ticket Refund Charges.....	61
Fig : 7.2.19 Seat Allocation during Booking... ..	61

CHAPTER – 1

INTRODUCTION

INTRODUCTION

The Flight Booking Management System is an innovative platform designed to simplify and enhance the flight booking experience for users while providing effective management tools for airlines and administrative oversight. This system is structured around three key roles: User, Company, and Admin.

Users have the ability to register and log in, enabling them to search for flights, make bookings, cancel reservations, view their profiles, and manage their booking history. Companies can register, log in, and utilize various functionalities, including managing flights through Create, Read, Update, and Delete (CRUD) operations and viewing their booking records.

The Admin role is essential for ensuring the system's integrity and smooth operation. Admins oversee the approval process for new companies, manage user accounts, and maintain flight details, thereby ensuring a seamless experience for all parties involved.

PROBLEM STATEMENT

The project aims to develop an efficient flight booking system featuring three roles: User, Company, and Admin. Users can search, book, and manage flights, while companies handle flight details and bookings. Admins oversee system management by accepting or rejecting companies, managing users, and monitoring flight information to ensure smooth operations.

MOTIVATION:

The project aims to simplify flight booking processes by creating an intuitive platform for users, companies, and administrators. Users can efficiently manage their bookings, while companies can handle flights seamlessly. Admins maintain platform integrity by overseeing company registrations and user activities. This system enhances the overall user experience, reduces operational complexity, and fosters a streamlined, reliable booking system.

OBJECTIVE OF THE PROJECT:

The objective of the project is to develop a comprehensive flight booking system with distinct roles for users, companies, and administrators. Users can search, book, and manage flights, while

companies handle flight listings and bookings. Administrators oversee company approvals, user management, and flight monitoring, ensuring an efficient and secure platform for all stakeholders.

SCOPE:

The scope of this project is to develop a comprehensive flight booking system with three roles: User, Company, and Admin. Users can manage flight bookings and profiles. Companies have CRUD functionality for flights and can view bookings. Admins oversee platform operations, including approving or rejecting companies, viewing users, and managing flights. The system ensures streamlined flight management and user interaction.

CHAPTER - 2

LITERATURESURVEY

2. LITERATURE SURVEY

2.1 Related Work

1. Author: Kumar, A., & Sharma, S.

Title: Emerging Trends in Flight Booking Systems: A Review

Outcome: AI integration has improved personalized customer experiences, with mobile-first platforms providing easier access to flight bookings.

Disadvantages: High costs of initial setup and resistance to fully automated systems. Privacy and algorithm bias are also concerns.

2. Author: Zhang, L., Xu, S., & Huang, W.

Title: Blockchain Applications in Airline Ticketing Systems: Enhancing Transparency and Security

Outcome: Blockchain improves the transparency and security of booking transactions, reducing fraud and ensuring tamper-proof records.

Disadvantages: Regulatory issues and complex implementation slow down adoption, with interoperability between airlines being a challenge.

3. Author: Zhang, L., Xu, S., & Huang, W.

Title: Blockchain Applications in Airline Ticketing Systems: Enhancing Transparency and Security

Outcome: Blockchain improves the transparency and security of booking transactions, reducing fraud and ensuring tamper-proof records.

Disadvantages: Regulatory issues and complex implementation slow down adoption, with interoperability between airlines being a challenge.

4. Author: Patel, M., & Gupta, R.

Title: Artificial Intelligence-Powered Customer Service in Airlines: Enhancing Booking and Post-Booking Experiences

Outcome: AI-powered systems, like chatbots, improve customer engagement and enhance the booking and post-booking experience.

Disadvantages: Loss of human interaction can alienate customers, and over-reliance on AI systems poses a risk during technical issues, leading to service disruptions.

5. Author: Smith, P., & Johnson, T.

Title: The Impact of Mobile Applications on Airline Booking Systems: Convenience vs. Data Security

Outcome: Mobile apps have streamlined the flight booking process, offering enhanced convenience and engagement for travelers.

Disadvantages: Concerns around data privacy and security, as mobile apps handle large volumes of sensitive personal information, increasing the risk of breaches.

CHAPTER - 3

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

Existing System:

The existing system for flight management lacks an integrated platform where users, companies, and administrators can seamlessly interact. Users face difficulties in booking, canceling flights, and managing profiles, while companies struggle to efficiently manage flights and view bookings. Admins lack proper control over accepting/rejecting companies and monitoring user activities, leading to inefficiencies and scattered functionalities.

Disadvantages:

- 1. Limited User Control for Canceling Flights:** Users may face issues with restrictive cancellation policies, or lack of flexibility in canceling or modifying bookings without proper support, leading to dissatisfaction.
- 2. Admin Overload in Company Verification:** The responsibility of accepting or rejecting companies solely lies with the admin, which could lead to bottlenecks, especially if there are many company registration requests to process.
- 3. Scalability Challenges for CRUD Operations:** As the number of companies, users, and flights increases, managing flight CRUD operations (Create, Read, Update, Delete) may become complex without an efficient database management system, affecting performance.
- 4. Security Risks in Authentication:** Without multifactor authentication or strong encryption for sensitive operations like bookings and profile updates, the system might be vulnerable to security breaches, leading to compromised user or company data.
- 5. No Automated User Support:** There is no mention of user support mechanisms, such as a help desk or chatbot. Users and companies may struggle with issues that require admin intervention, affecting user experience and efficiency.

Proposed System:

The proposed system is an online flight management platform offering functionalities for users, companies, and an admin. Users can register, login, search and book flights, cancel bookings, view profiles, and bookings. Companies can manage flights using CRUD operations and view bookings. Admins oversee the platform by accepting/rejecting companies, viewing users and flights, ensuring smooth system management.

Price Fluctuations: Whenever the companies tend to change the flight ticket prices, a Notification is displayed to User

- **Ticket Refund Charges:** Calculate charges for users if a flight is canceled or delayed, ensuring transparent refund processing.
- **Alternate Flight Suggestions:** Provide alternative flights to users if their booked flight is canceled or delayed, ensuring customer convenience.
- **Seat Allocation During Booking:** Display available seats for passengers to select while booking their tickets, offering a customizable experience.



Advantages:

1. User Convenience: The system offers a seamless experience for users, allowing them to register, log in, search for flights, make bookings, cancel them, and view their profiles and bookings easily, which enhances user satisfaction.

2. Efficient Flight Management: Companies have the ability to manage flights using the CRUD functionality (Create, Read, Update, Delete), ensuring that they can keep their flight information accurate and up to date, improving operational efficiency.

3. Admin Control and Monitoring: Admins have the power to accept or reject companies and manage flights and users, ensuring a secure and wellregulated platform that can minimize misuse and improve trust among users.

4. Scalability: With separate modules for users, companies, and admins, the system can be expanded to include more features and functionalities without disrupting the core operations, ensuring future scalability.

5. Booking Transparency: Users and companies can view their bookings, providing clarity and reducing confusion about upcoming or past reservations, which helps in maintaining trust and reliability in the system.

Work Flow of Proposed system:

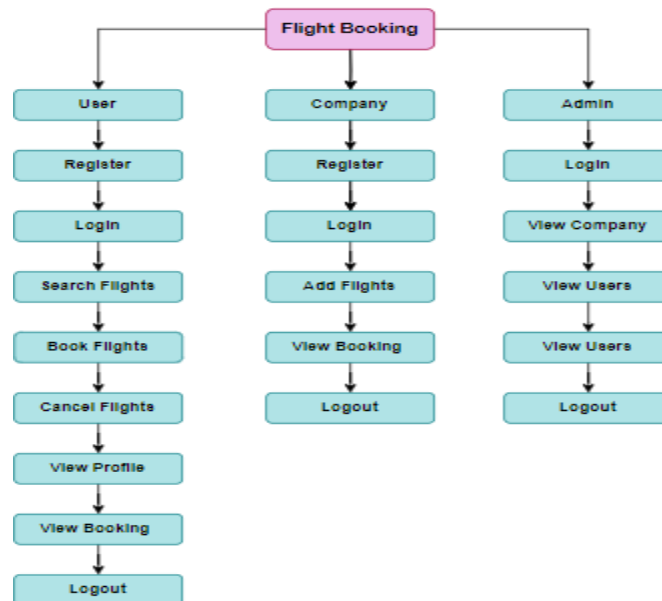


Fig : Work Flow Of Proposed System

CHAPTER - 4

REQUIREMENT ANALYSIS

4. REQUIREMENT ANALYSIS

4.1 Functional and Non-Functional Requirements for the Flight Booking System

Functional Requirement

User Module:

Company Module:

1. Company Registration:

Companies can register with the system by providing necessary business details.

2. Login:

Companies can securely log in using their credentials to access their dashboard.

3. Add Flights (CRUD):

Companies can create, read, update, and delete flights from their inventory.

4. View Bookings:

Companies can view bookings made by users for their flights.

5. Logout:

The system should securely log out the company user and end their session.

Admin Module:

1. Admin Login:

Admins should log in securely to access administrative functionalities.

2. View Company:

Admins can view company details and either accept or reject company registration.

3. View Users:

Admins can view details of all registered users.

4. View Flights:

Admins should be able to view all flights registered by the companies.

5. Logout:

The system should securely log out the admin and end the session.

Non-Functional Requirements

1. Security:

The system should ensure secure user authentication (e.g., using SSL encryption and password hashing).

Role-based access control should be in place to ensure that users, companies, and admins can only access permitted functionalities.

Sensitive user information such as passwords should be encrypted in the database.

2. Performance:

The system should respond to user actions (such as search and booking) within a reasonable time (e.g., less than 2 seconds).

The system should handle high volumes of concurrent users, especially during peak times (scalability).

3. Usability:

The user interface should be intuitive and easy to navigate for both users and companies.

Error messages should be user-friendly and offer guidance to help users resolve issues.

4. Reliability:

The system should ensure high availability (99.9% uptime) to ensure users and companies can access the system at all times.

Regular backups should be implemented to avoid data loss.

5. Data Integrity:

The system must ensure the integrity of data, especially for bookings and transactions, to prevent inconsistencies.

Transactional integrity should be maintained (e.g., ACID properties in the database).

6. Scalability:

The system should be scalable to handle increasing numbers of users, flights, and bookings without performance degradation.

7. Maintainability:

The system should be built using modular architecture to allow easy updates and bug fixes without affecting other components.

8. Accessibility:

The system should be accessible across different devices and platforms, including desktop, tablet, and mobile.

9. Auditability:

The system should keep logs of all user actions, especially bookings, cancellations, and updates, for audit and tracking purposes.

4.2 HARDWARE REQUIREMENTS:

HARDWARE REQUIREMENTS:

- Processor I3/Intel Processor
- RAM 4GB (min)
- Hard Disk 160GB
-

4.3 SOFTWARE REQUIREMENTS:

SOFTWARE SYSTEM CONFIGURATION:

- Operating System : Windows 7/8/10
- Server side Script : Express js
- Programming Language : TypeScript
- IDE/Workbench : VS Code
- Database : Mongodb
- Clint Side : React js

4.4 Architecture:

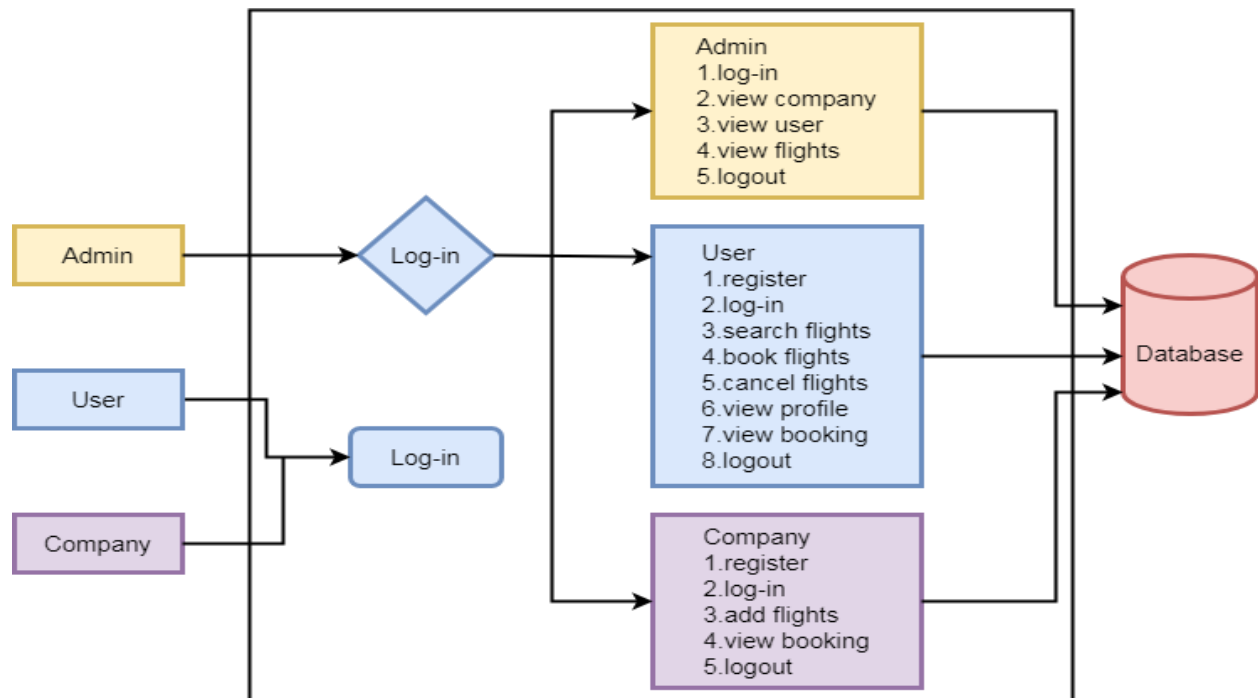


Fig :4 .4 Architecture

CHAPTER – 5

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 Introduction of Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well designed input forms and screens have following properties –

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding –
 - What are the inputs needed for the system?
 - How end users respond to different elements of forms and screens.

Objectives for Input Design:

The objectives of input design are –

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

Output Design: The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design:

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

5.2 UML Diagrams:

5.2.1 Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Usecase analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Fig : 5.2.1 Use Case Diagram

5.2.2 Class Diagram:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

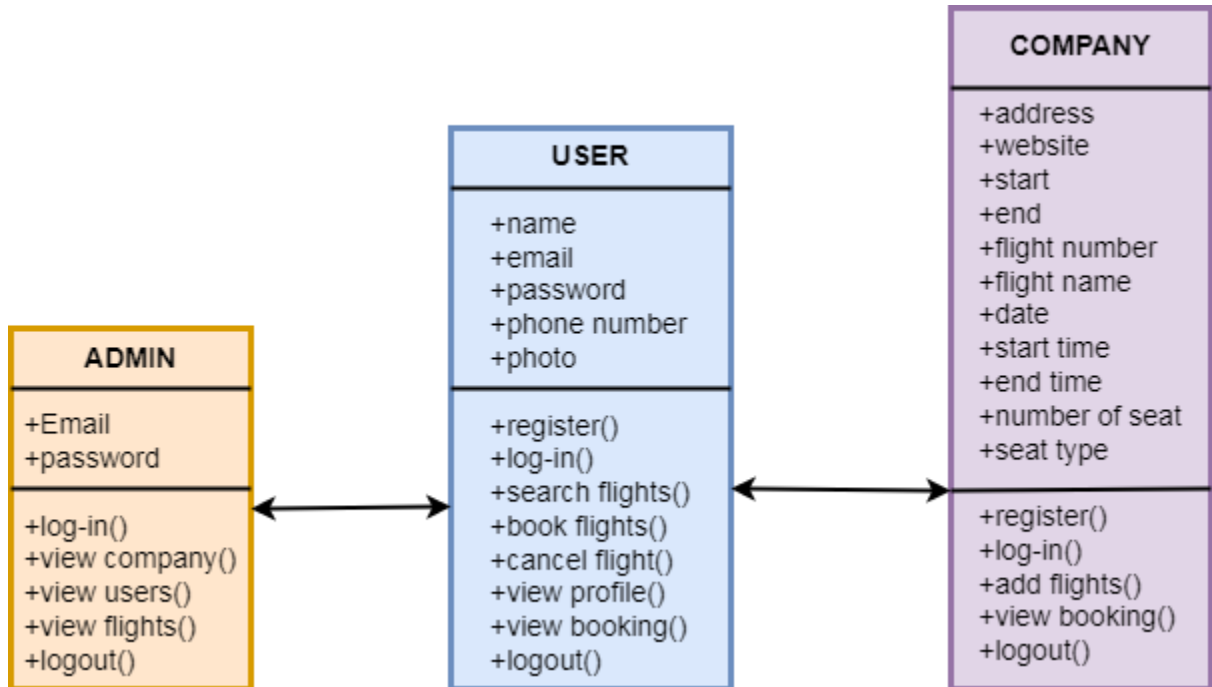


Fig : 5.2.2 Class Diagram

5.2.3 Sequence Diagram:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

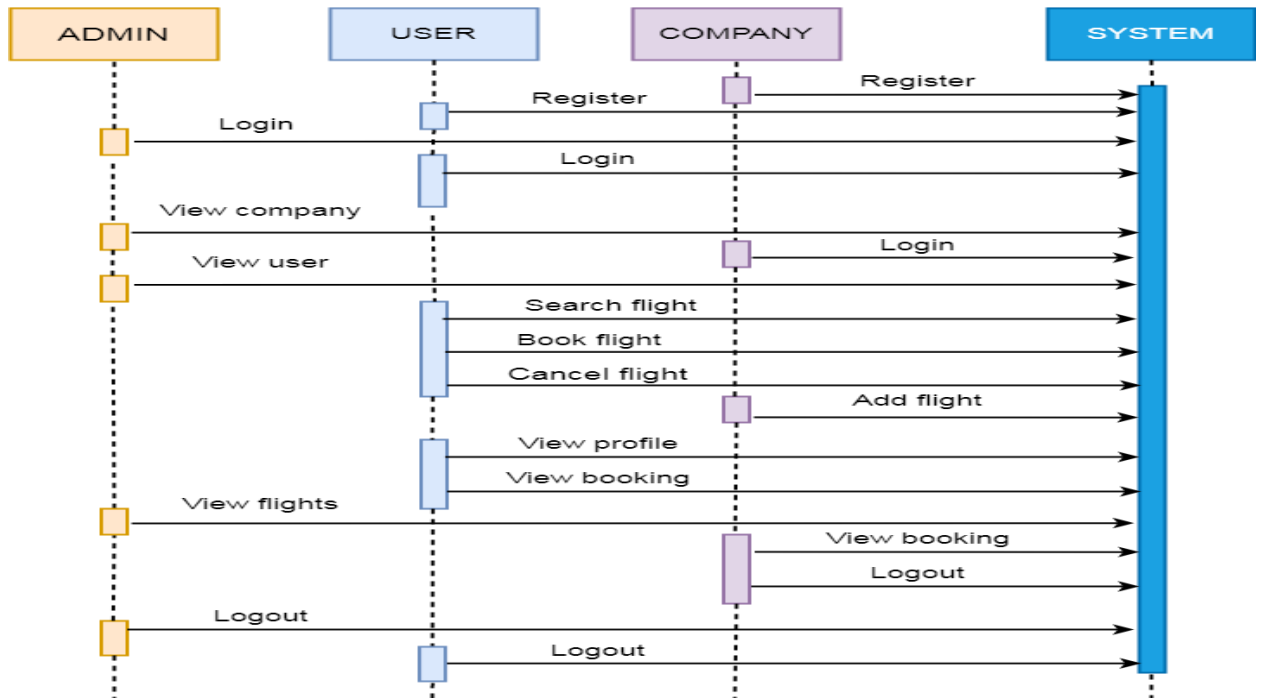


Fig : 5.2.3 Sequence Diagram

5.2.4 Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

5.2.5 Deployment Diagram

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

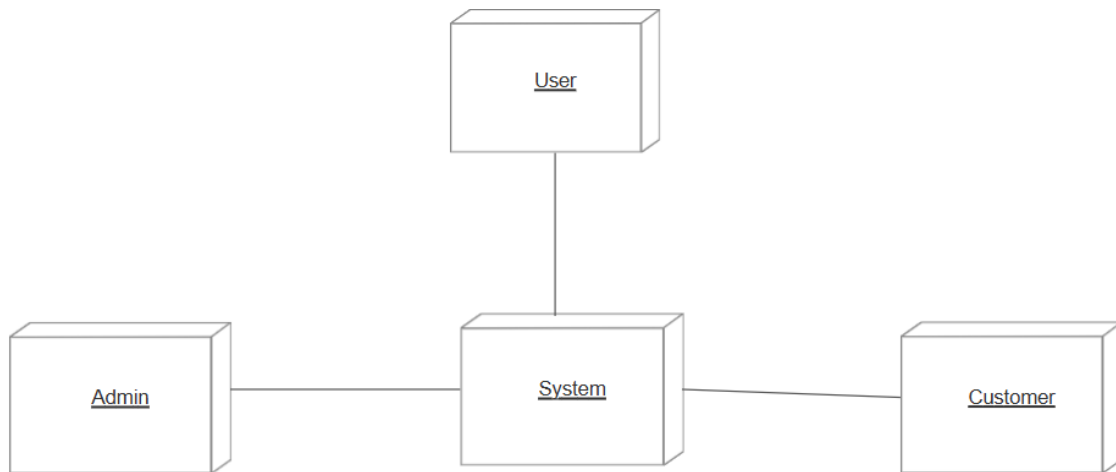


Fig : 5.2.5 Deployment Diagram

5.2.6 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational stepbystep workflows of components in a system. An activity diagram shows the overall flow of control.



Fig : 5.2.6 Activity Diagram

5.2.7 Component Diagram:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and doublecheck that every aspect of the system's required functions is covered by planned development.

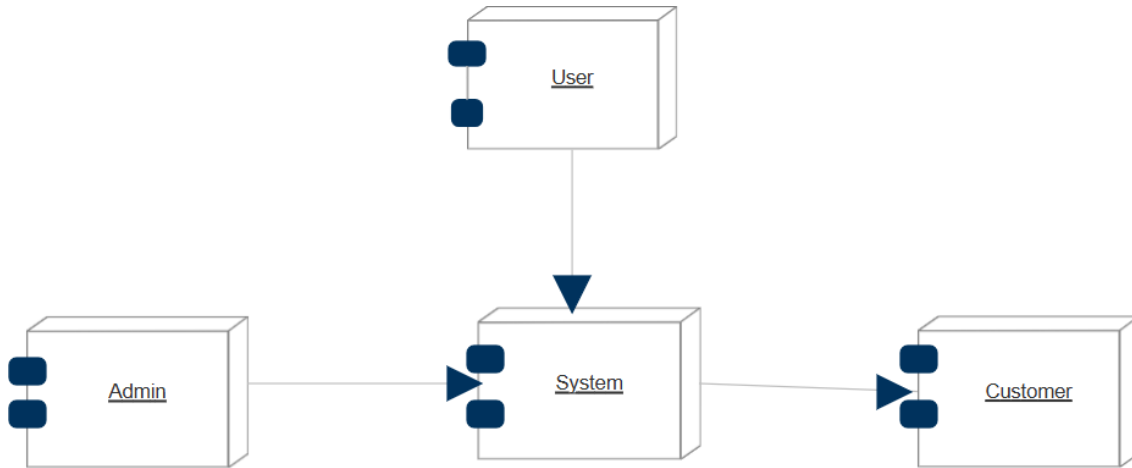


Fig : 5.2.7 Component Diagram

5.2.8 ER Diagram:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of ER model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

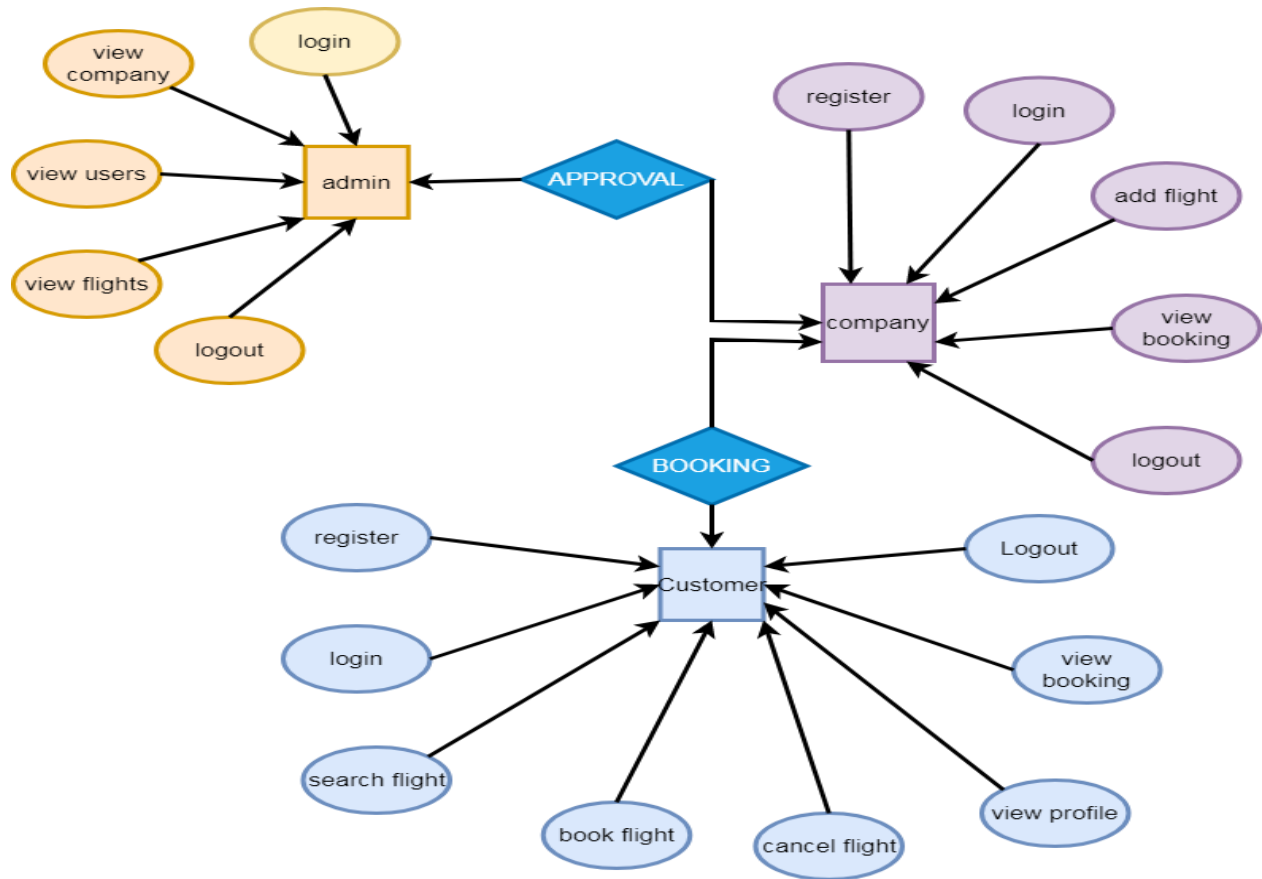


Fig : 5.2.8 ER Diagram

CHAPTER – 6

IMPLEMENTATION

6. IMPLEMENTATION

6.1. Modules:

1. User Module:

Register: User registration with personal details.

Login: Secure login for users to access their account.

Search Flights: Users can search for available flights.

Book Flights: Users can book available flights.

Cancel Flights: Users can cancel their booked flights.

View Profile: Users can view and edit their personal details.

View Bookings: Users can view their flight booking history.

Logout: Secure logout to end the session.

2. Company Module:

Register: Company registration with necessary business details.

Login: Secure login for companies to access their dashboard.

Add Flights (CRUD): Companies can create, read, update, and delete flights.

View Bookings: Companies can view all bookings made for their flights.

Logout: Secure logout for the company.

3. Admin Module:

Login: Admin access to manage the system.

View Company: Admin can view company details and accept/reject company registration.

View Users: Admin can view registered users and their details.

View Flights: Admin can view all flights registered by companies.

Logout: Secure logout for the admin.

6.2 Source Code:

```
import mongoose from "mongoose";

import express from "express";

import cors from "cors";

import bodyParser from "body-parser";

import Admin from "../Controller/Admin";

import Company from "../Controller/Company";

import Customer from "../Controller/Customer";

import path from "path";

const app = express();

app.use(

  cors({

    origin: "http://192.168.1.130:5175/",

    allowedHeaders: ["Content-Type", "Authorization"],

    methods: ["GET", "POST", "PUT", "DELETE"],

  })

);

app.use(express.json());
```

```
app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.use("/uploads", express.static(path.join(__dirname, "uploads")));

// Connect to MongoDB

const uri = "mongodb://localhost:27017/flightnew";

const port = 4000;

const server = async () => {

  try {

    const db = await mongoose.connect(uri);

    if (db) {

      app.listen(port, () => {

        console.log(`Server is running on port ${port}`);

      });

      console.log("Connected to MongoDB");

    }

  } catch (error) {

    console.log("Error connecting to MongoDB", error);

  }

};

server();

// Routes

app.use("/api/admin", Admin);

app.use("/api/company", Company);

app.use("/api/customer", Customer);
```

CHAPTER – 7

TESTING AND RESULT

7.1 SYSTEM STUDY AND TESTING

7.1.1 Feasibility Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ Economic feasibility
- ◆ Technical feasibility
- ◆ Social feasibility

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.1.2 Types of Tests

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and rsesponses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

7.1.3 TEST CASES:

User Module:

Test Case ID	Test Scenario	Precondition	Test Steps	Expected Result
TC-01	User Registration	User is on the registration page.	1. Open registration page. 2. Enter valid details. 3. Click on "Register."	User account should be successfully created, and a confirmation message should be displayed.
TC-02	User Login	User has registered and is on the login page.	1. Open login page. 2. Enter valid email and password. 3. Click on "Login."	User should be logged in successfully and redirected to the dashboard.
TC-03	Search Flights	User is logged in.	1. Navigate to the flight search page. 2. Enter flight search criteria. 3. Click on "Search."	A list of available flights matching the criteria should be displayed.
TC-04	Book Flight	User has searched for available flights.	1. Select a flight. 2. Click on "Book." 3. Confirm the booking.	Flight booking should be confirmed, and the user should receive a booking confirmation message.
TC-05	Cancel Flight Booking	User has an active booking.	1. Navigate to "My Bookings" page. 2. Select a booked flight.	The flight should be canceled, and a cancellation confirmation message should be displayed.

			3. Click on "Cancel."	
TC-06	View Profile	User is logged in.	1. Navigate to the "Profile" page. 2. View or edit personal details. 3. Click on "Save Changes."	Profile details should be updated and saved successfully.
TC-07	View Bookings	User has at least one booked flight.	1. Navigate to "My Bookings" page.	All previous and current bookings should be displayed.
TC-08	Logout	User is logged in.	1. Click on "Logout."	User should be logged out, and the session should be terminated.

Company Module:

Test Case ID	Test Scenario	Precondition	Test Steps	Expected Result
TC-09	Company Registration	Company is on the registration page.	1. Open registration page. 2. Enter valid company details. 3. Click on "Register."	Company account should be successfully created, and a confirmation message should be displayed.
TC-10	Company Login	Company is registered and is on the login page.	1. Open login page. 2. Enter valid credentials. 3. Click on "Login."	Company should be logged in successfully and redirected to the company dashboard.
TC-11	Add Flight (CRUD)	Company is logged in.	1. Navigate to "Add Flight" page. 2. Enter flight details.	Flight should be added successfully, and a confirmation message should be displayed.

			3. Click on "Add Flight."	
TC-12	View Bookings (Company)	Company has bookings.	1. Navigate to "View Bookings" page. 2. Review the list of bookings made by users.	All bookings for the company's flights should be displayed.
TC-13	Update Flight	Company has added flights.	1. Navigate to "Manage Flights" page. 2. Select a flight to update. 3. Modify details. 4. Save.	Flight details should be updated successfully.
TC-14	Delete Flight	Company has flights listed.	1. Navigate to "Manage Flights" page. 2. Select a flight to delete. 3. Click on "Delete."	The selected flight should be deleted, and the flight list should update without the deleted entry.
TC-15	Company Logout	Company is logged in.	1. Click on "Logout."	Company should be logged out, and the session should be terminated.

Admin Module:

Test Case ID	Test Scenario	Precondition	Test Steps	Expected Result
TC-16	Admin Login	Admin is on the login page.	1. Enter valid admin credentials. 2. Click "Login."	Admin should be logged in and redirected to the admin dashboard.
TC-17	View Company List	Admin is logged in.	1. Navigate to the "View Companies" page. 2. Review the list of registered companies.	Admin should see all registered companies and be able to approve or reject company registrations.

			3. Approve or reject company registration.	
TC-18	View Users	Admin is logged in.	1. Navigate to the "View Users" page. 2. View the list of all registered users.	Admin should see all user details, including name, email, and booking history.
TC-19	View All Flights	Admin is logged in.	1. Navigate to the "View Flights" page. 2. View the list of all flights added by companies.	A complete list of all registered flights should be displayed, with details like flight number and time.
TC-20	Admin Logout	Admin is logged in.	1. Click on the "Logout" button.	Admin should be logged out, and the session should be terminated.

7.2 Results

Home Page: This is the project's landing page.

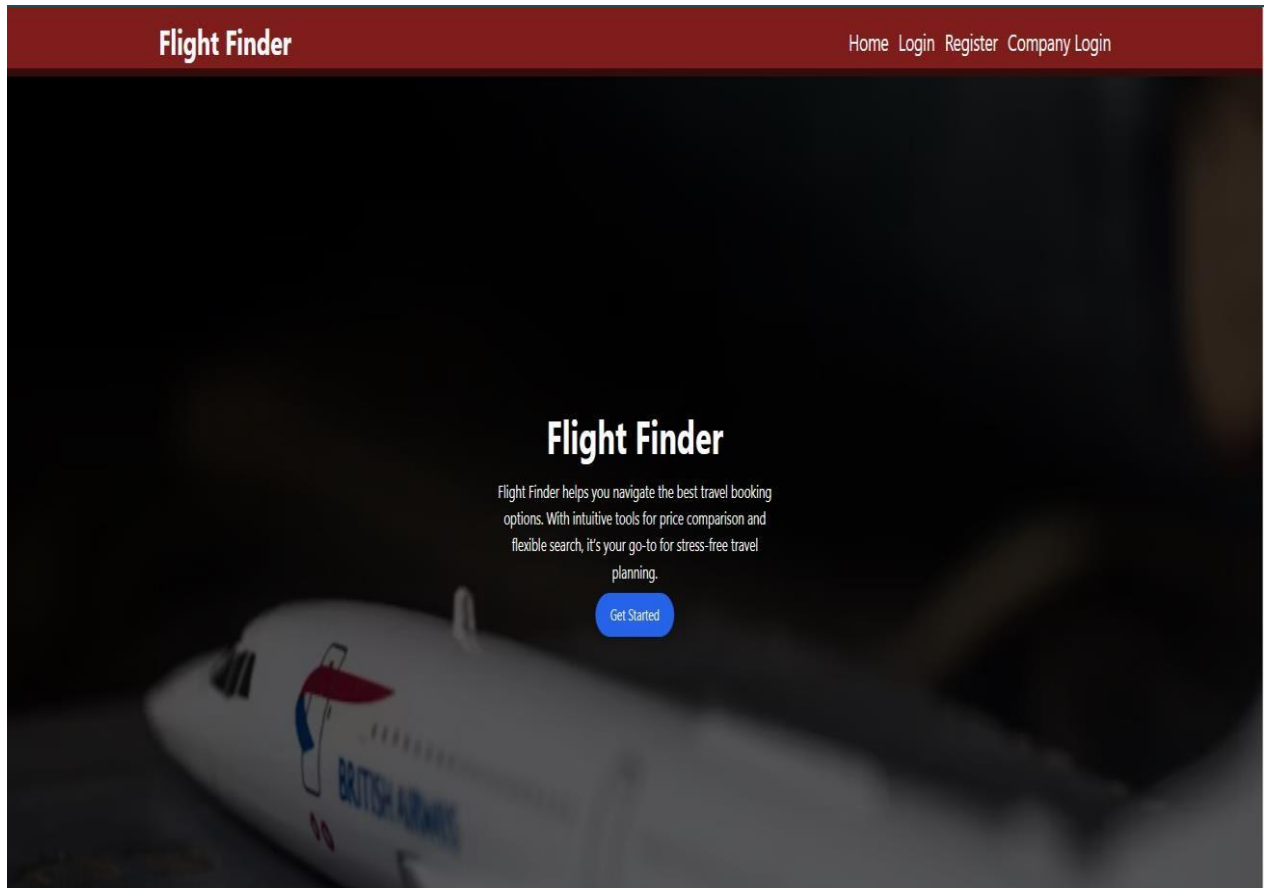
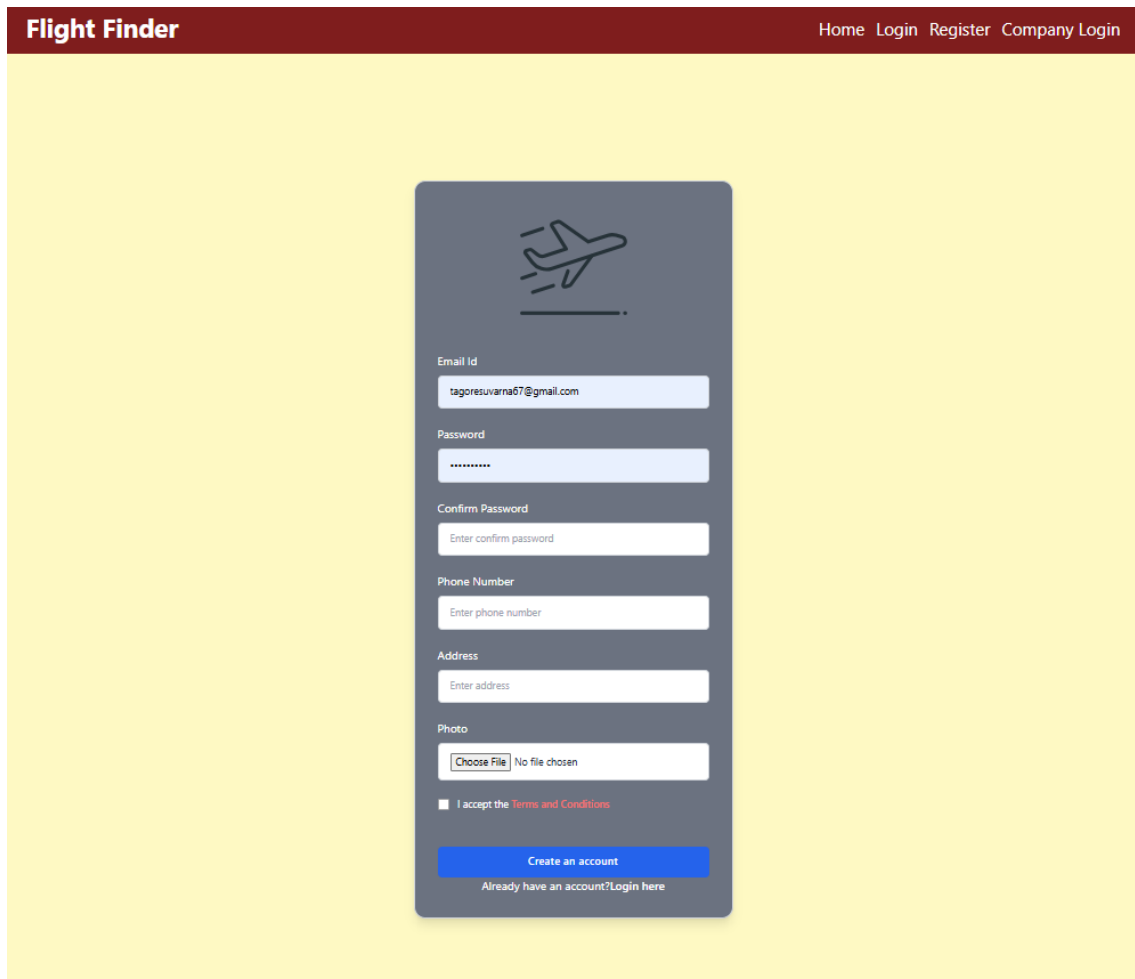


Fig :7.2.1 Home Page

Customer Registration Page: New customers must create an account on this portal before booking. To do so, they need to fill out the required information for account creation.



The screenshot displays the 'Flight Finder' customer registration page. The page has a dark red header with the site name 'Flight Finder' on the left and navigation links 'Home', 'Login', 'Register', and 'Company Login' on the right. The main content area has a light yellow background. Centered on the page is a dark gray registration form. At the top of the form is an airplane icon. Below the icon are input fields for 'Email Id' (containing 'tagoresuvarna57@gmail.com'), 'Password' (masked with dots), 'Confirm Password' (with placeholder text 'Enter confirm password'), 'Phone Number' (with placeholder text 'Enter phone number'), 'Address' (with placeholder text 'Enter address'), and 'Photo' (with a 'Choose File' button and 'No file chosen' text). Below these fields is a checkbox labeled 'I accept the Terms and Conditions'. At the bottom of the form is a blue 'Create an account' button and a link that says 'Already have an account? Login here'.

Fig : 7.2.2 Customer registration Page

Login Page: Both customers and admins can use this page to access their accounts and perform additional operations after logging in.

Flight Finder

Home Login Register Company Login

Sign In

Email
lagoresuvarna67@gmail.com

Password

Sign in

Don't have an account? Register here

Fig: 7.2.3 Login Page

Admin Home Page: Company Details - The admin can view the details of registered companies. Company personnel can log in only after admin approval. If the admin rejects the request, the company will not be able to log in.

Admin Panel

Company
Users
Flights
Logout

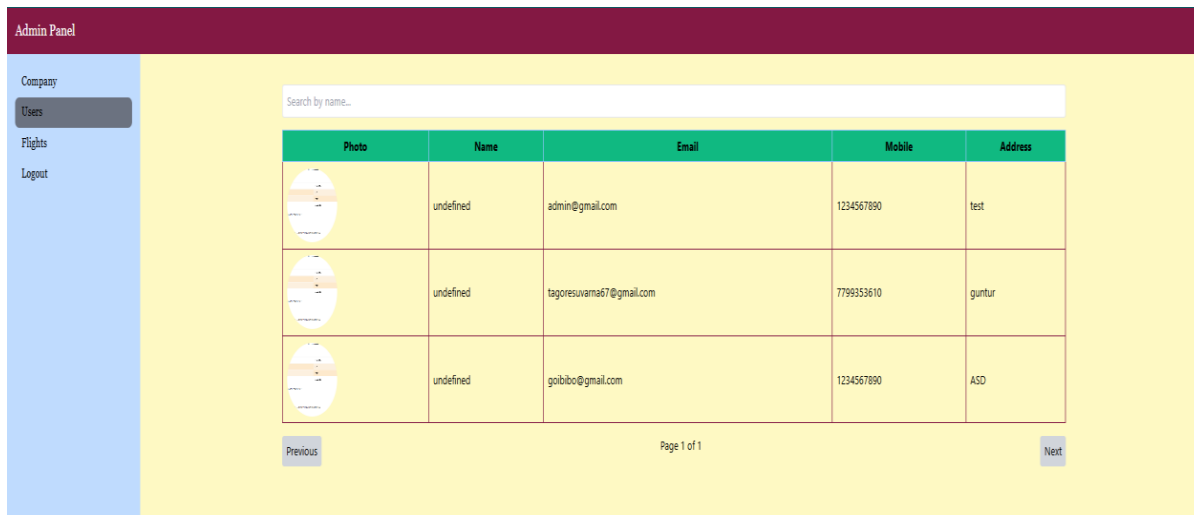
Company Table

Search by name...

Photo	Name	Email	Mobile	Address	Website	Action
	Indigo	indigo@gmail.com	1234567890	test	indigo.com	Approved
	Goibibo	goibibo@gmail.com	1234567890	Hyd	goibibo.com	Approved
	KHIT	Airtravel@gmail.com	1234567890	Guntur	khit.com	Approved

Fig : 7.2.4 Admin Page




View Registered Customer Details: The admin can view the details of registered customers or users.



Admin Panel

Company
Users
Flights
Logout

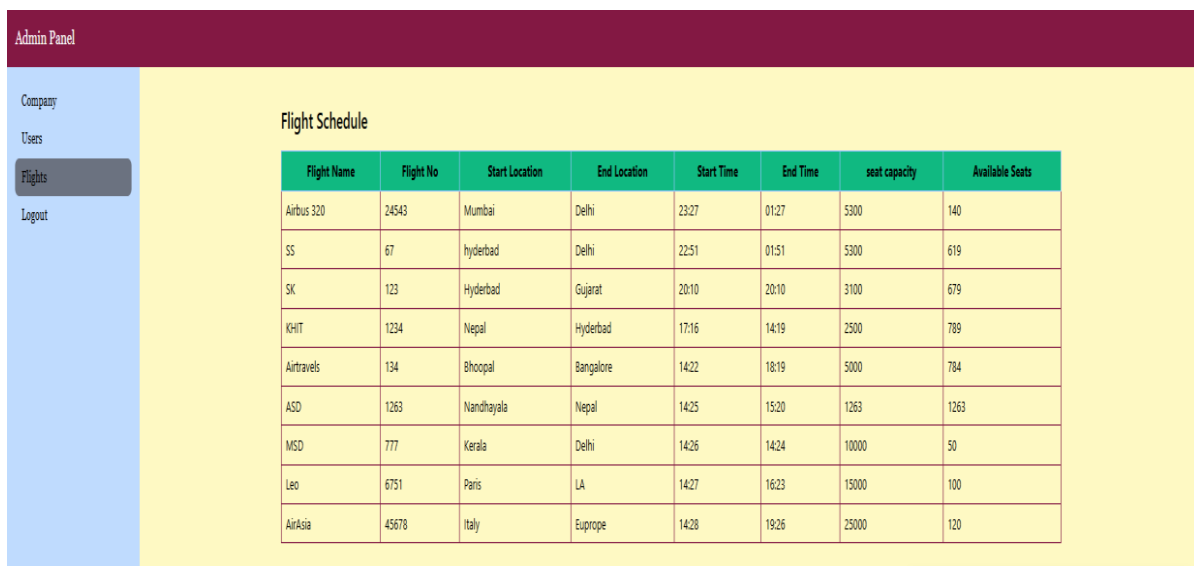
Search by name..

Photo	Name	Email	Mobile	Address
	undefined	admin@gmail.com	1234567890	test
	undefined	tagoresuvarna67@gmail.com	7799353610	guntur
	undefined	goibibo@gmail.com	1234567890	ASD

Previous Page 1 of 1 Next

Fig: 7.2.5 View Registered Customer details

View Flight Details: The admin can view flight details on this page.



Admin Panel

Company
Users
Flights
Logout

Flight Schedule

Flight Name	Flight No	Start Location	End Location	Start Time	End Time	seat capacity	Available Seats
Airbus 320	24543	Mumbai	Delhi	23:27	01:27	5300	140
SS	67	hyderabad	Delhi	22:51	01:51	5300	619
SK	123	Hyderabad	Gujarat	20:10	20:10	3100	679
KHIT	1234	Nepal	Hyderabad	17:16	14:19	2500	789
Airtravels	134	Bhoopal	Bangalore	14:22	18:19	5000	784
ASD	1263	Nandhayala	Nepal	14:25	15:20	1263	1263
MSD	777	Kerala	Delhi	14:26	14:24	10000	50
Leo	6751	Paris	LA	14:27	16:23	15000	100
AirAsia	45678	Italy	Europe	14:28	19:26	25000	120

Fig : 7.2.6 View Flight Details

Company Registration Page: Flight companies can register by filling in their name, email, password, mobile number, address, website, and uploading a photo.

Fig : 7.2.7 Company Registration Page

Company Login Page: A company can log in using its email and password only after receiving approval from the admin.

Fig: 7.2.8 Company Login Page

Flight Adding Page: This page allows users to add flight details by entering the name, flight number, departure and arrival locations, start and end times, seat capacity, and price.

Company Panel
AddFlights
ViewFlights
ViewBooking
Logout

Add Flight

Flight Name

Flight No Please fill out this field.

From Location

To Location

Start Time

End Time

Date

Seat Type

Seat Capacity

Price

Add Flight

Fig : 7.2.9 Flight Adding Page

View Added Flight Details: Users can view flight details here, and if needed, they can update or delete the flight information.

Company Panel
AddFlights
ViewFlights
ViewBooking
Logout

View Flights

Flight Name	Flight No	From	To	Start Time	End Time	Date	Seat Type	Capacity	Price	Status	Actions
Airbus 320	24543	Mumbai	Delhi	23:27	01:27	11/6/2024	Economy	140	5300	cancelled	
SS	67	hyderabad	Delhi	22:51	01:51	11/6/2024	Common	619	5300	active	Update Cancel Flight

Fig : 7.2.10 View Added Flight Details

View Booked Flight Details: Users can view their booked flight details here.

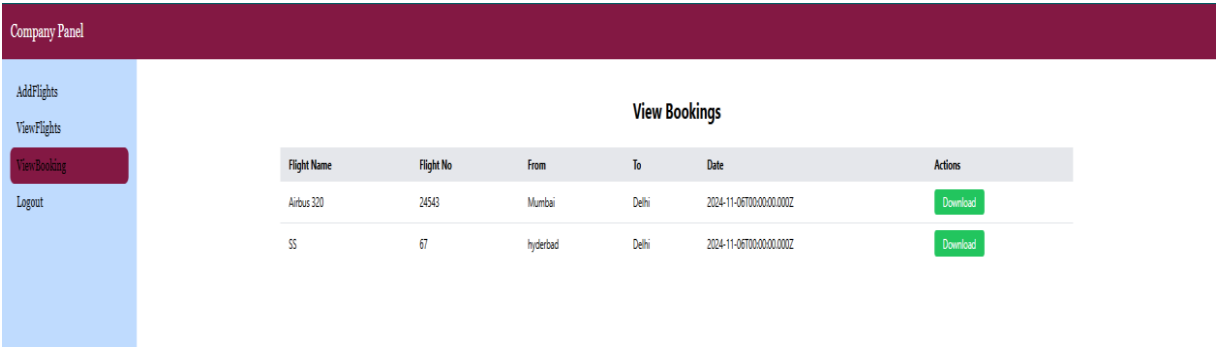


Fig : 7.2.11 View Booked Flight Details

Booked Customer Details in PDF: Company can download the booked details in PDF format, which includes the customer information.

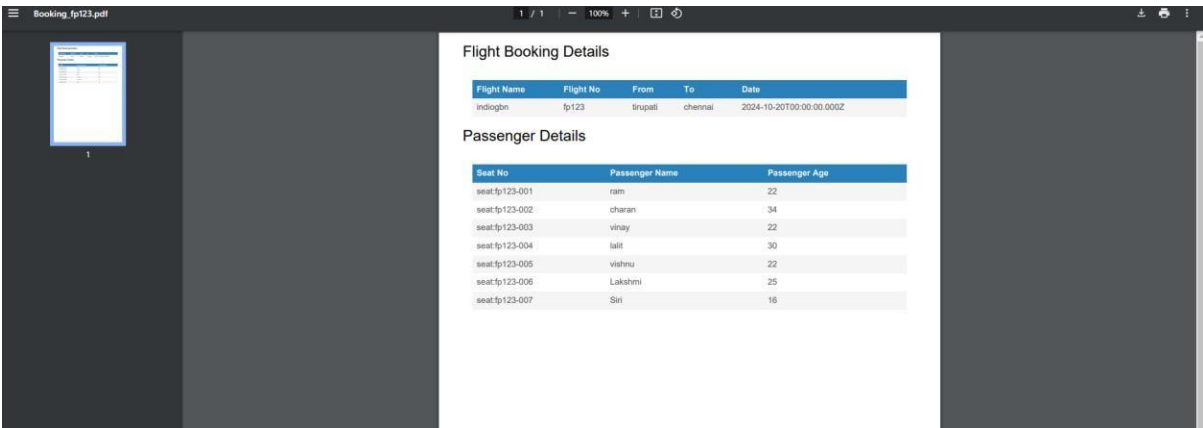


Fig : 7.2.12Booked Customer Details in PDF

After Customer Login: Once logged in, customers can view the flight details as shown below. Flight Booking Form: This form contains the details required for flight booking, allowing customers to reserve their flight seats.

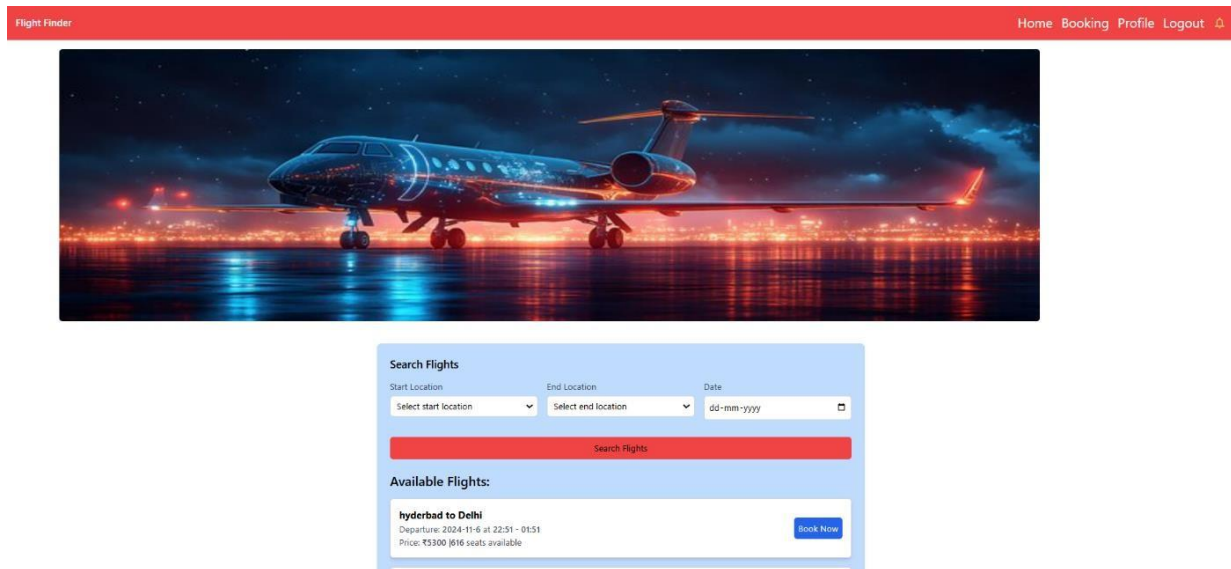


Fig : 7.2.13 After Customer Login

After Booking: Once the ticket is successfully booked, customers can view their booked flight details as shown below.

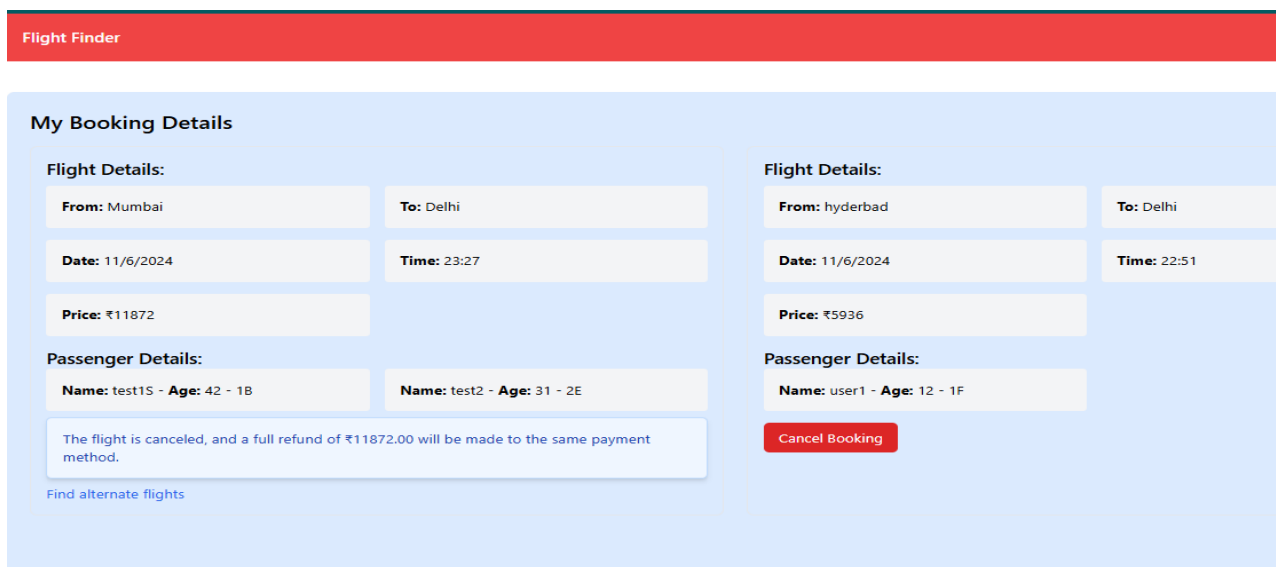


Fig : 7.2.14 After Booking

Profile Page and Change Password Page: Customers can view their profile, and if they wish to change their password, they can do so here.

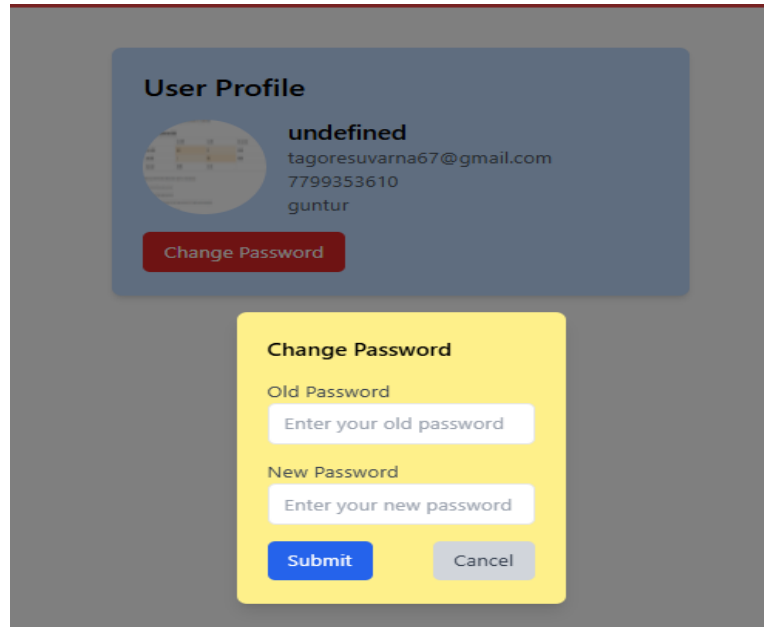


Fig: 7.2.15 Profile Page and Change Password Page

Price Fluctuation Notification : When the companies tend to Increase or Decrease their Flight Booking Prices , a Notification will Displayed to users.

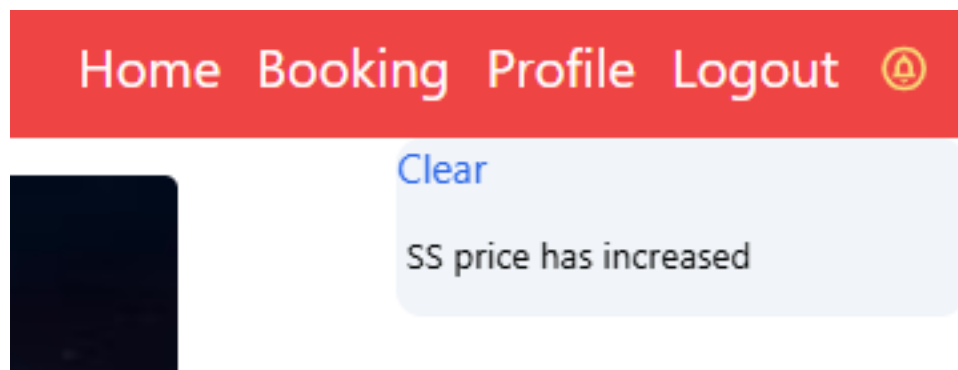


Fig : 7.2.16 Price Fluctuation Notification

Alternate Flight Suggestions: Provide options for alternate flights when the original flight is canceled or delayed.

Flight Finder

My Booking Details

Flight Details:

From: Mumbai	To: Delhi
Date: 11/6/2024	Time: 23:27
Price: ₹11872	

Passenger Details:

Name: test1S - Age: 42 - 1B	Name: test2 - Age: 31 - 2E
---	--

The flight is canceled, and a full refund of ₹11872.00 will be made to the same payment method.

[Find alternate flights](#)

Fig : 7.2.17 Alternate Flight Suggestions

Ticket Refund Charges: Calculate charges when a flight is canceled or delayed.

Flight Details:

From: hyderabad **To:** Delhi

Date: 11/6/2024 **Time:** 22:51

Price: ₹5936

Passenger Details:

Name: user1 -

Cancel Booking

Cancellation Details

Original Price:	₹5300.00
(-) Cancellation Deduction (30%):	₹1590.00
Ticket Refund Amount:	₹3710.00
(+) IGST Refund (12%):	₹636.00
Total Refund Amount:	₹4346.00

Confirm Cancel **Close**

Fig : 7.2.18 Ticket Refund Charges

Seat Allocation During Booking: Show available seats for passengers to choose from while booking tickets.

Email **Mobile Number**

Number of Passengers

1

Passenger 1 Name **Passenger 1 Age** **Passenger 1 Seat**

Select Seat

Card Number

Expiry Date

CVV

Price:

IGST (12%):

Total Amount:

Cancel **Confirm**

Select Seat

- 1A
- 1B
- 1C
- 1D
- 1E
- 1F
- 2A
- 2B
- 2C
- 2D
- 2E
- 2F
- 3A
- 3B
- 3C
- 3D
- 3E
- 3F

Fig : 7.2.19 Seat Allocation during Booking

CHAPTER – 8

CONCLUSION

8 .FUTURE ENHANCEMENT

Enhance user experience with mobile app development and personalized recommendations. Implement two-factor authentication and data encryption for security. Improve the rental module with advanced booking management and pricing analytics. Develop an analytics dashboard for admins, integrate online payment systems, and ensure legal compliance. Finally, create a feedback system for user support and suggestions.

CONCLUSION

User Module: Register: User registration with personal details. Login: Secure login for users to access their account. Search Flights: Users can search for available flights. Book Flights: Users can book available flights. Cancel Flights: Users can cancel their booked flights. View Profile: Users can view and edit their personal details. View Bookings: Users can view their flight booking history. Logout: Secure logout to end the session. 2. Company Module: Register: Company registration with necessary business details. Login: Secure login for companies to access their dashboard. Add Flights (CRUD): Companies can create, read, update, and delete flights. View Bookings: Companies can view all bookings made for their flights. Logout: Secure logout for the company. 3. Admin Module: Login: Admin access to manage the system. View Company: Admin can view company details and accept/reject company registration. View Users: Admin can view registered users and their details. View Flights: Admin can view all flights registered by companies. Logout: Secure logout for the admin

CHAPTER – 9

REFERENCES

9. REFERENCES

- 1. Agarwal, A. & Gupta, P. (2020)**”Design and Implementation of an Online Flight Booking System”. International Journal of Computer Applications, 975, 8887.
[Link to Paper](#)
- 2. Zhang, Y., & Xu, J. (2021)**”Development of a Mobile Application for Flight Booking with Real-Time Updates”.Journal of Mobile Technology in Medicine, 10(1), 12-18.
- 3. Lee, J., & Kim, S. (2019)**”User Experience Design for Flight Booking Systems: A Case Study”.Journal of Usability Studies, 14(4), 183-198.
- 4. Patel, D., & Shah, N. (2018)**”: An Online Flight Reservation System: Features and Functionality”.International Journal of Advanced Research in Computer Science, 9(4), 43-49.
- 5. Santos, A., & Oliveira, L. (2020)**” An Analysis of Payment Integration in Flight Booking Systems”. Journal of Information Systems and Technology Management, 17(1), 1-18.
- 6. Cheng, M., & Wong, W. (2021)**”Cloud-Based Solutions for Flight Booking Systems”.International Journal of Cloud Computing and Services Science, 10(2), 65-72.
- 7. Ravi, P., & Kumar, S. (2019)**”Trends in Online Flight Booking: A Survey”.International Journal of Travel and Tourism Research, 12(3), 55-68.