UDAAN

UDAAN

A FLIGHT BOOKING SYSTEM

Developed by
Swanit Rivankar
Samruddhi Kadam

TABLE OF CONTENT

EXISTING APP

IXIGO is an Indian travel and booking platform that provides a range of services for travelers, including flight, train, bus, and hotel bookings. It was founded in 2007 by **Aloke Bajpai** and **Rajnish Kumar** and has grown to become one of India's most popular travel metasearch engines and booking apps.
The **ixigo app** is a popular Indian travel platform that provides convenient services for booking flights, trains, buses, and hotels.

After using the app we found out this issues:

- **Additional Service Fees**: Extra charges and hidden fees at checkout can increase the final cost unexpectedly.
- **Customer Support Delays**: Slow response times, especially for refund and cancellation requests during peak travel seasons.
- **Technical Glitches**: Occasional crashes or issues with transactions, which can disrupt the booking experience.
- **Inconsistent Prices**: Quick price fluctuations may lead to differences in initially displayed vs. final booking costs.
- **Strict Refund Policies**: Refund processing can be slow and sometimes inflexible, particularly for budget fares.

GAP

- **Data Security and Authentication**:
    - **ixigo**: May use standard security measures but lacks explicit details on advanced hashing for secure login.
    - **Our App**: Employs advanced encryption techniques for password storage, ensuring users' credentials are kept secure, which enhances trust in data protection.

- **Real-Time Flight Information**:
    - **ixigo**: Provides flight information but may not fully leverage real-time data, which can lead to inconsistencies in availability or pricing due to delays in data syncing.
    - **Our App**: Integrates real-time flight data through the Amadeus API, ensuring that users see the most current availability and pricing, reducing surprises in final booking costs

- **In-App Notifications and Alerts**:
    - **ixigo**: While it may notify users of booking confirmations, it may not consistently alert users about specific issues such as API limits, technical downtimes, or data sync delays.
    - **Our App**: Includes real-time notifications that keep users informed about potential issues with API data limits or delays, providing transparency and a more reliable experience.

- **Enhanced Search Flexibility**:
    - **ixigo**: While it provides basic search options, the experience may be rigid, especially with limited filters or advanced search criteria.
    - **Your App**: Designed to offer flexibility in flight searches with options to filter by criteria such as departure and arrival locations, times, and costs, allowing users to find flights that best match their preferences.

UDAAN

A FLIGHT BOOKING SYSTEM

The Flight Booking System is an interactive JavaFX-based application designed to streamline the process of booking flights, managing user profiles, and saving booking histories. Built with user convenience and functionality in mind, it provides a seamless user experience from registration to booking confirmation. The system uses MongoDB as its backend to securely store user data, including profile images, past booking information, and other essential details, ensuring that users can access their data whenever needed.

At the core of the app, users can register using their email, username, and a secure password. The system encrypts passwords with advanced hashing techniques, making sure that user credentials are kept safe and secure. Once registered, users can log in using either their username or email, offering flexibility and ease of access. The profile section allows users to personalize their accounts by uploading a profile picture, which is stored in MongoDB and displayed whenever the user views or edits their profile.

The app's main functionality, booking flights, is designed with flexibility and a rich user experience in mind. Users can search for available flights, view details such as flight numbers, departure and arrival locations, times, and costs, and save their bookings for future reference. The app allows users to review their past bookings, which are displayed in a clear and organized format, making it easy for users to track their travel history.

In addition, the booking section allows users to customize their bookings by selecting options tailored to different passenger types (such as adults and infants), each with individual details. For example, if a child under a certain age is traveling, the app can adjust pricing accordingly, reflecting real-world booking scenarios. Before finalizing a booking, the user is presented with a summary screen that displays all essential details, allowing them to confirm or modify the booking as needed.

One of the standout features of the app is its ability to handle profile images and bookings in a visually interactive way. Profile images are saved as binary data in MongoDB, allowing users to upload images securely and display them seamlessly within the JavaFX interface. This gives a personal touch to the user experience, making it feel less transactional and more engaging.

The app also includes a real-time connection with the Amadeus API, which allows users to search and view real-world flight data. This API integration adds to the system's authenticity by providing accurate and up-to-date flight information. For those who frequently search for flights, the app is built to handle requests efficiently, though occasional API limits may affect performance. In such cases, users receive alerts, ensuring a transparent experience. Additionally, the app includes a payment gateway, allowing users to initiate payments for confirmed bookings. The payment process is integrated in such a way that users have a seamless checkout experience right from the booking confirmation screen.

In summary, this Flight Booking System is more than just a simple booking tool; it's a fully-featured application that combines robust data management, secure user authentication, and a dynamic user interface to create a reliable and enjoyable experience for booking flights. Through features like profile management, detailed booking summaries, and real-time API integration, the app positions itself as an ideal solution for anyone looking to manage their travel needs efficiently and securely.

Java

Java is a high-level, object-oriented programming language developed by **Sun Microsystems** in 1995, now owned by **Oracle Corporation**. It is widely used for building cross-platform applications, from web and mobile applications to enterprise systems. Java's "write once, run anywhere" capability means compiled Java code can run on any platform with a compatible Java Virtual Machine (JVM), making it one of the most versatile languages in software development.

Key Features of Java

- **Platform Independence**: Java is platform-independent at both the source and binary levels, thanks to the JVM, which allows Java applications to run on any operating system without modification.
- **Object-Oriented**: Java follows the principles of OOP (Object-Oriented Programming), such as encapsulation, inheritance, and polymorphism, which makes it modular, flexible, and easy to maintain.
- **Rich API and Library Support**: Java provides an extensive standard library (Java API) that includes classes and methods for tasks like data structures, networking, file I/O, and more, enhancing development speed and productivity.
- **High Performance**: Although traditionally slower than languages like C++, Java's Just-In-Time (JIT) compiler optimizes bytecode into native machine code at runtime, improving performance.

Javafx

JavaFX is a software platform for developing and delivering rich Internet applications (RIAs) that can run across a wide range of devices. It was introduced by Sun Microsystems (now Oracle) as a replacement for Swing in Java SE, aiming to provide modern UI components and tools for creating visually appealing applications. JavaFX is especially popular for building desktop applications and is part of the Java SE platform.

Key Features of JavaFX

- **Rich Set of UI Controls**: JavaFX offers a wide range of pre-built UI controls, such as buttons, tables, trees, sliders, and more, making it easier to build complex user interfaces.
- **CSS Styling**: JavaFX supports CSS-based styling, allowing developers to customize the look and feel of applications similar to web development. CSS can be used to modify colors, fonts, and other style properties.
- **FXML (Declarative UI)**: JavaFX allows for the use of FXML, an XML-based language for defining the user interface structure in a declarative way. This makes UI design more organized and enables a separation between the UI layout and application logic.
- **WebView Component**: JavaFX includes a WebView component based on WebKit, allowing developers to render HTML and interact with web content directly within JavaFX applications.
- **Event Handling**: JavaFX has a robust event-handling mechanism, allowing for effective handling of user input and actions across UI components.
- **Cross-Platform Compatibility**: JavaFX applications can run across multiple platforms (Windows, macOS, Linux) and adapt to each operating system’s look and feel.

Css

CSS (Cascading Style Sheets) is used in JavaFX to style and customize the appearance of user interface components. Just like in web development, CSS in JavaFX allows developers to separate the presentation layer from the application logic, enabling easier maintenance and design updates. Here’s an overview of how CSS works in JavaFX, including its features and usage:

Key Features of CSS in JavaFX

- **Separation of Concerns**: CSS allows you to separate the styling from the JavaFX application logic, making it easier to manage and modify the UI without changing the underlying code.
- **Rich Styling Options**: JavaFX CSS supports a wide range of styling properties similar to standard CSS, including colors, fonts, borders, backgrounds, padding, and margins.
- **Inline Styles**: In addition to external stylesheets, JavaFX allows you to apply CSS styles directly to components using the setStyle() method, making it easy to apply specific styles programmatically.
- **Theming**: CSS can be used to create themes for applications, allowing for easy switching between different styles and color schemes without altering the JavaFX code.

TECHNOLOGIES

ECLIPSE IDE

Eclipse IDE (Integrated Development Environment) is a widely-used open-source platform for software development, primarily known for Java programming but also supports other languages through various plugins. Developed by the Eclipse Foundation, it provides a robust set of tools and features for building, debugging, and managing applications

Key Features of Eclipse IDE

1. **Multi-language Support**: While Eclipse is primarily associated with Java development, it supports multiple programming languages such as C, C++, Python, PHP, and more through the installation of plugins.
2. **Modular Architecture**: Eclipse is built on a modular architecture, allowing developers to customize their IDE by adding or removing components as needed. This modularity is facilitated by its plugin system.
3. **Rich Java Development Tools**: Eclipse offers comprehensive tools for Java development, including:

    Code Editor: Features syntax highlighting, code completion, and refactoring tools.
    Debugger: Integrated debugging capabilities for tracking down and fixing issues in your code.
    JUnit Integration: Built-in support for writing and running unit tests using Junit
    .

Advantages of Using Eclipse IDE

- **Open Source**: Eclipse is free to use and open-source, making it accessible for individual developers and organizations.
- **Active Community**: A large and active community provides support, resources, and plugins, helping developers solve problems and extend the IDE's capabilities.
- **Frequent Updates**: Regular updates from the Eclipse Foundation ensure that the IDE remains modern, secure, and equipped with the latest features.
- **Customizable Environment**: Developers can tailor their workspace and tooling according to their specific needs and preferences.

JAVAFX

In JavaFX, the user interface (UI) is primarily built using a scene graph, which is a hierarchical structure representing all the visual elements in the application. This allows developers to create rich, interactive UIs. Below are the key components of JavaFX's interface architecture, including how to design and manage UI elements effectively.

Key Components of JavaFX Interface

- **Stage**:
  - The top-level container for JavaFX applications, equivalent to a window in desktop applications.
  - A Stage can be created and displayed using the Stage class, and it contains scenes, which hold all UI elements.
- **Scene**:
  - A scene is a container for all the visual elements in a JavaFX application. It represents the contents of a stage.
  - A scene can contain one or more nodes organized in a scene graph.
- **Nodes**:
  - Nodes are the basic building blocks of a JavaFX UI, including UI controls (like buttons, text fields, etc.), shapes (like rectangles and circles), and other visual elements.
  - Each node can be added to a parent node, creating a hierarchy.

- **Layouts**:
  - Layouts manage the positioning and sizing of nodes in a scene. JavaFX provides several layout panes to organize UI components:
    - **VBox**: Arranges children in a vertical column.
    - **HBox**: Arranges children in a horizontal row.
    - **GridPane**: Arranges children in a grid of rows and columns.
    - **BorderPane**: Arranges children in five regions: top, bottom, left, right, and center.
    - **StackPane**: Stacks children on top of each other.
- **UI Controls**:
  JavaFX provides a rich set of UI controls that can be used to create interactive applications:
  - **Button**: A clickable button.
  - **Label**: Displays text.
  - **TextField**: A single-line input field.
  - **TextArea**: A multi-line input field.
  - **ComboBox**: A dropdown list for selecting from multiple options.
  - **TableView**: Displays tabular data.

- **CSS for Styling**:
  - JavaFX allows for styling of UI elements using CSS. You can define styles in external CSS files or inline.

- **Event Handling**:
  - JavaFX provides an event-handling mechanism to respond to user actions (like clicks, key presses, etc.).
  - You can add event listeners to UI components to define the behavior when an event occurs.

MONGODB

MongoDB is a popular open-source NoSQL database designed for storing and managing large volumes of unstructured or semi-structured data. It utilizes a flexible, document-oriented data model, which allows for dynamic schemas and easier data representation compared to traditional relational databases.

Key Features of MongoDB

1. **Document-Oriented Storage**:
   MongoDB stores data in JSON-like documents (BSON format), allowing for the representation of complex data structures, including nested objects and arrays.
2. **Flexible Schema**:
   Unlike traditional relational databases, MongoDB does not require a predefined schema. You can easily change the structure of documents without downtime.
3. **Scalability**:
   MongoDB supports horizontal scaling through sharding, allowing data to be distributed across multiple servers or clusters, making it suitable for handling large datasets and high-traffic applications.

Advantages of Using MongoDB

- **Schema Flexibility**: The ability to store data without a fixed schema simplifies development and allows for rapid iteration on data models.
- **Scalability**: MongoDB’s architecture supports horizontal scaling, making it suitable for applications that experience high growth.
- **Performance**: Its document-oriented nature and indexing capabilities result in high performance for read and write operations.
- **Ease of Use**: The JSON-like data format is intuitive for developers, allowing for easy data manipulation and interaction.
- **Rich Query Capabilities**: MongoDB’s query language is powerful and allows for complex data retrieval and manipulation.

APIs

AMADEUS API

Amadeus API refers to a set of application programming interfaces (APIs) provided by Amadeus, a leading technology provider for the global travel and tourism industry. Amadeus APIs enable developers to access various travel-related services and data, including flight bookings, hotel reservations, car rentals, and more. This allows businesses to create applications and platforms that offer travel-related services to customers efficiently.

Key Features of Amadeus API

- **Comprehensive Travel Data**:
  - Access to a vast database of travel information, including flights, hotels, car rentals, and travel regulations.
- **Real-Time Availability**:
  - Retrieve real-time information on flight availability, pricing, and booking options.
- **Booking Capabilities**:
  - APIs for making reservations for flights, hotels, and cars, allowing developers to integrate booking functionality directly into their applications.
- **Search Functionality**:
  - Search for flights, hotels, and other travel services using various parameters, such as destination, date, and price range.

EXCHANGERATE AP

ExchangeRate API refers to a service that provides exchange rate data for different currencies. Such APIs allow developers to access current and historical foreign exchange rates, enabling them to build applications that require currency conversion functionalities. This is particularly useful for financial applications, e-commerce platforms, and any service dealing with multiple currencies.

Key Features of ExchangeRate APIs

- **Real-Time Exchange Rates**:
  - Access to current exchange rates between various currencies, often updated every few seconds or minutes.
- **Currency Conversion**:
  - Simple endpoints for converting one currency to another using the most recent exchange rates.
- **Secure and Reliable**:
  - Most APIs offer secure access via HTTPS and are designed to provide reliable data for critical applications.

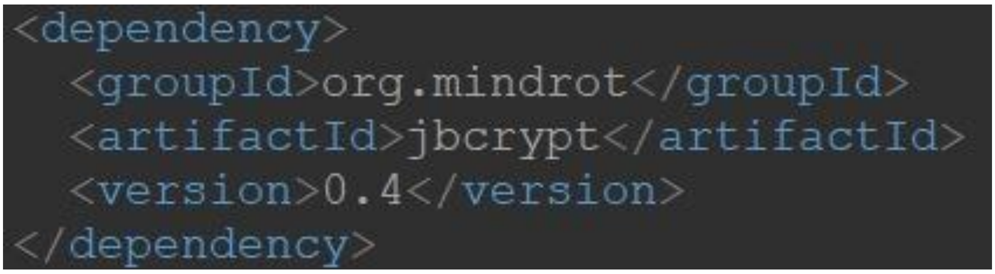
DEPENDENCIES

JBCrypt

JBCrypt is a Java library that provides functionality for cryptographic operations, specifically focusing on password hashing and encryption. This library is particularly useful for developers looking to securely store passwords and perform other cryptographic tasks within Java applications.

JBCrypt Dependency for Maven

To use JBCrypt in a Java project that utilizes Maven for dependency management, you will need to include the JBCrypt library in your pom.xml file. However, it's important to note that JBCrypt is not available in the Maven Central Repository. Instead, you can either download it from a source repository or find a similar library (like Bouncy Castle or JCrypt) that is available on Maven Central.



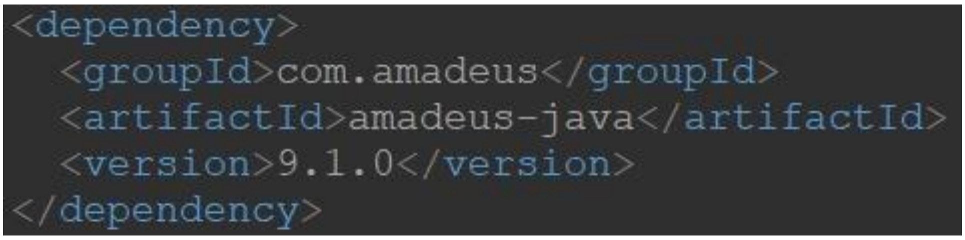
```
<dependency>
  <groupId>org.mindrot</groupId>
  <artifactId>jbcrypt</artifactId>
  <version>0.4</version>
</dependency>
```

Amadeus API

To use the Amadeus API in a Java project, you need to add the Amadeus Java SDK as a dependency. This SDK provides a set of methods to interact with Amadeus's various services like flight booking, hotel information, and more. Here’s how you can add it to your project:

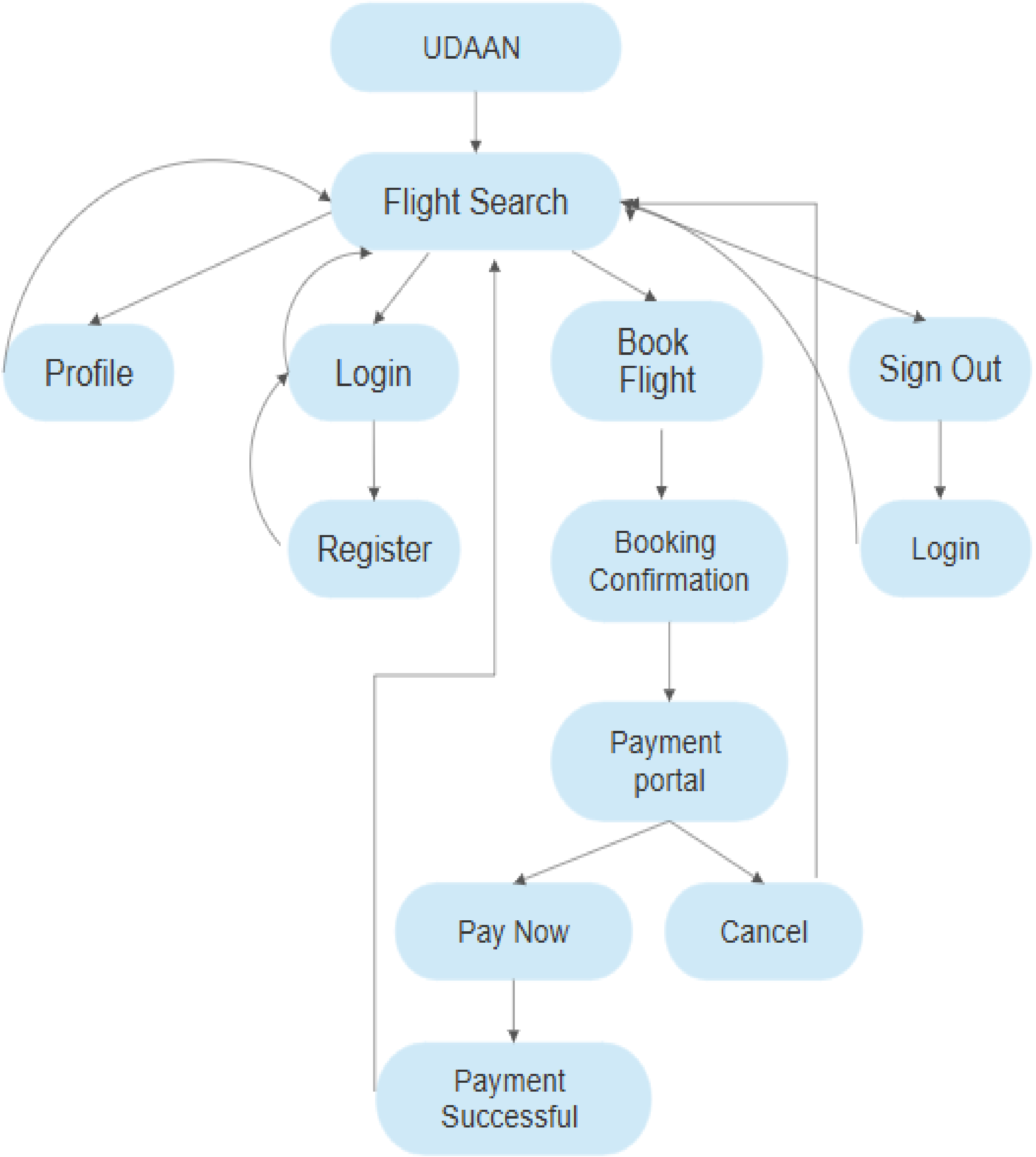
Maven Dependency (pom.xml)

If you are using Maven, add the following dependency to your pom.xml:



```
<dependency>
  <groupId>com.amadeus</groupId>
  <artifactId>amadeus-java</artifactId>
  <version>9.1.0</version>
</dependency>
```

FLOW CHART


UDAAN
Flight Search
Profile
Login
Book Flight
Sign Out
Register
Booking Confirmation
Login
Payment portal
Pay Now
Cancel
Payment Successful

WORKING OF THE APP

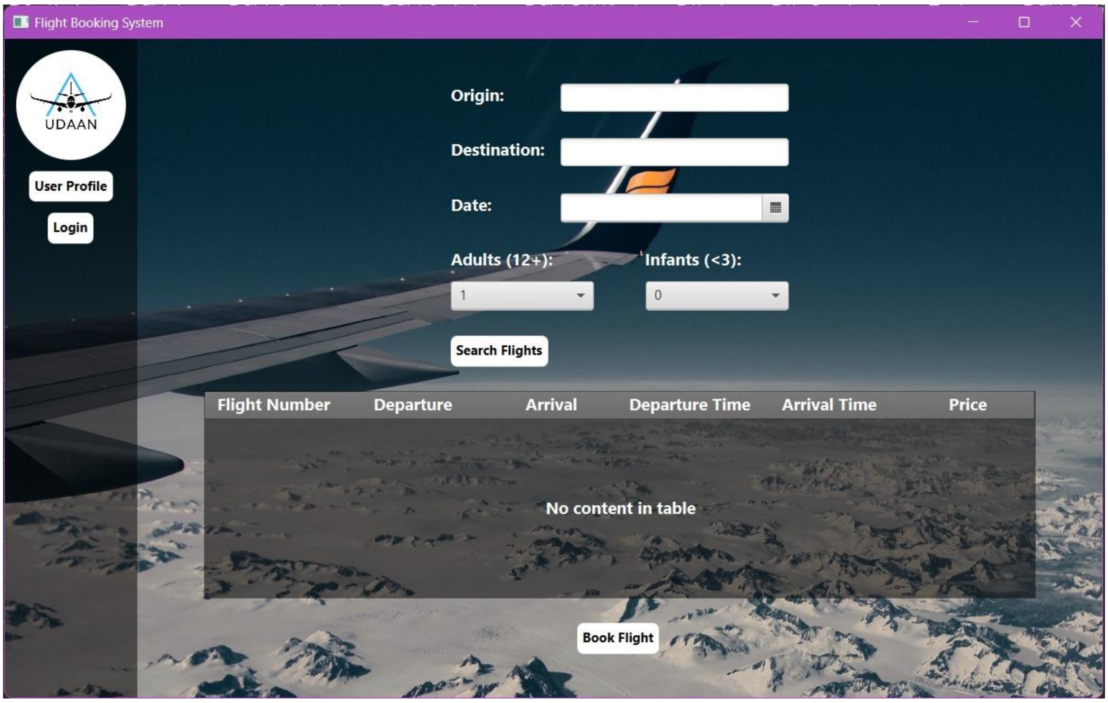


fig1:This is the first page that the user will see when they open the app.

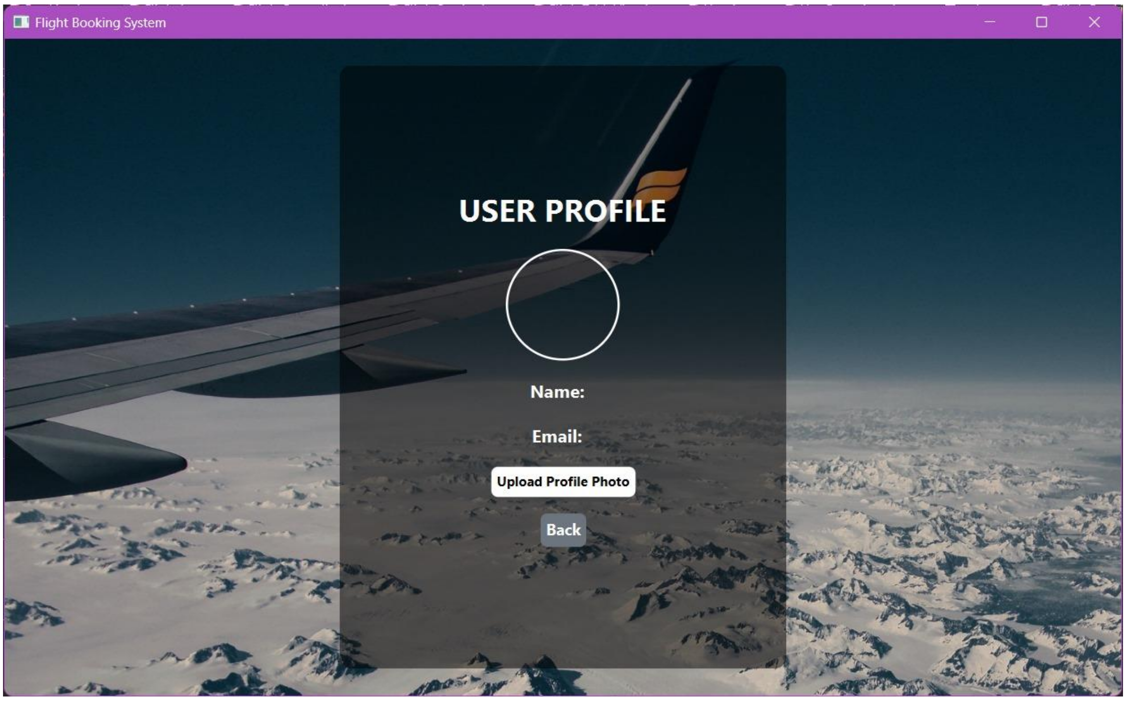
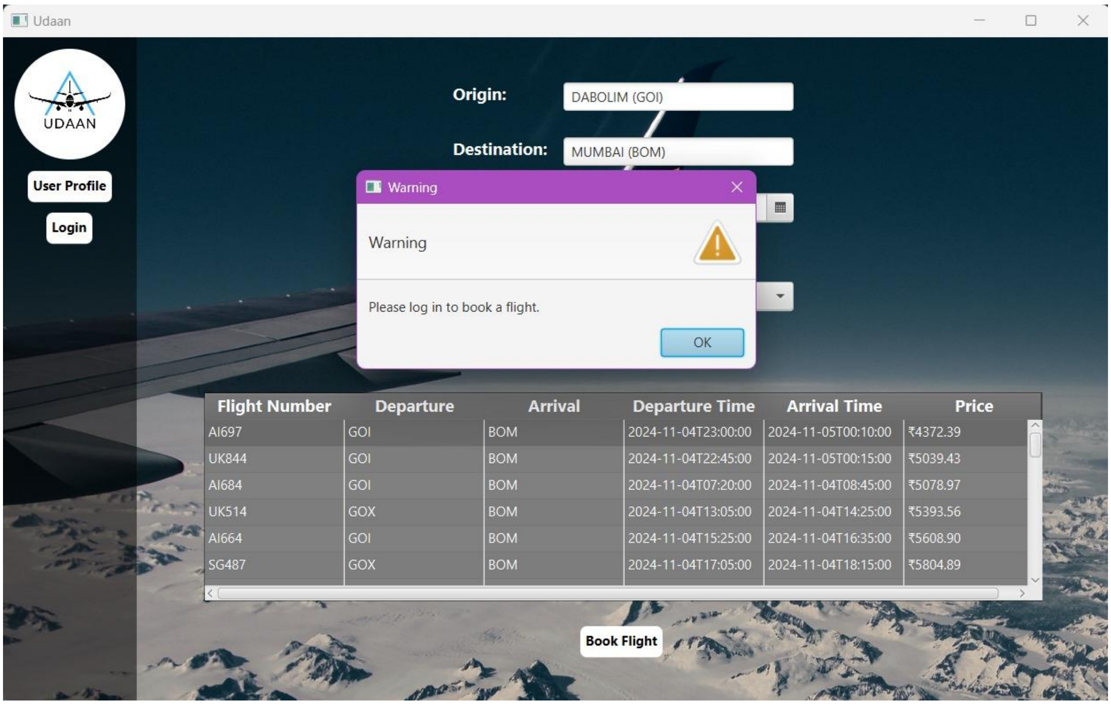


Fig2:This the profile page that user will see before logging in.



Origin: DABOLIM (GOI)

Destination: MUMBAI (BOM)

Warning

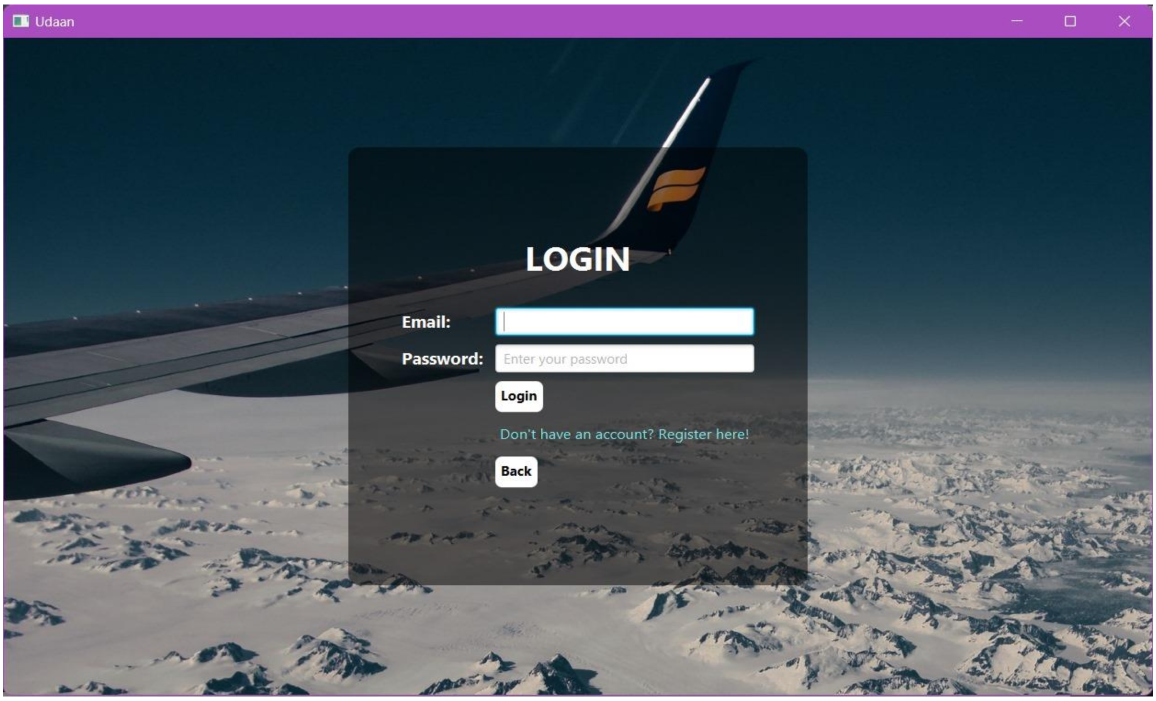
Warning

Please log in to book a flight.

OK

Flight Number	Departure	Arrival	Departure Time	Arrival Time	Price
AI697	GOI	BOM	2024-11-04T23:00:00	2024-11-05T00:10:00	₹4372.39
UK844	GOI	BOM	2024-11-04T22:45:00	2024-11-05T00:15:00	₹5039.43
AI684	GOI	BOM	2024-11-04T07:20:00	2024-11-04T08:45:00	₹5078.97
UK514	GOX	BOM	2024-11-04T13:05:00	2024-11-04T14:25:00	₹5393.56
AI664	GOI	BOM	2024-11-04T15:25:00	2024-11-04T16:35:00	₹5608.90
SG487	GOX	BOM	2024-11-04T17:05:00	2024-11-04T18:15:00	₹5804.89

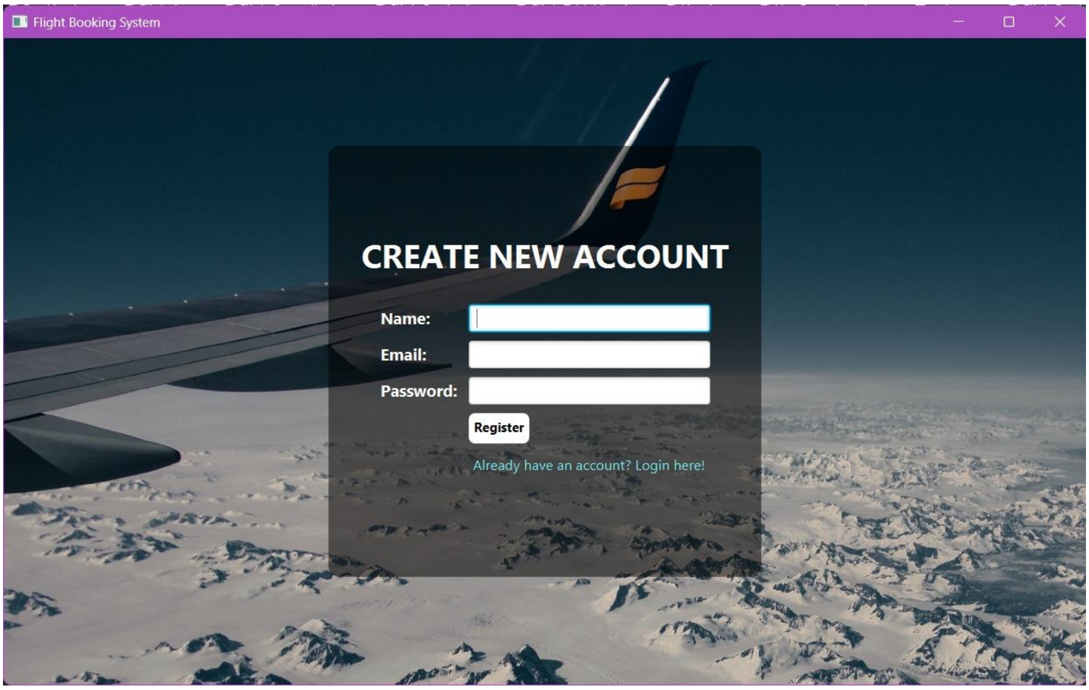
fig3:This is an alert prompting user to login in before proceeding to book the flight tickets



Email:

Password: Enter your password

fig4:This is a login page where the user can login using their email id and password



Flight Booking System

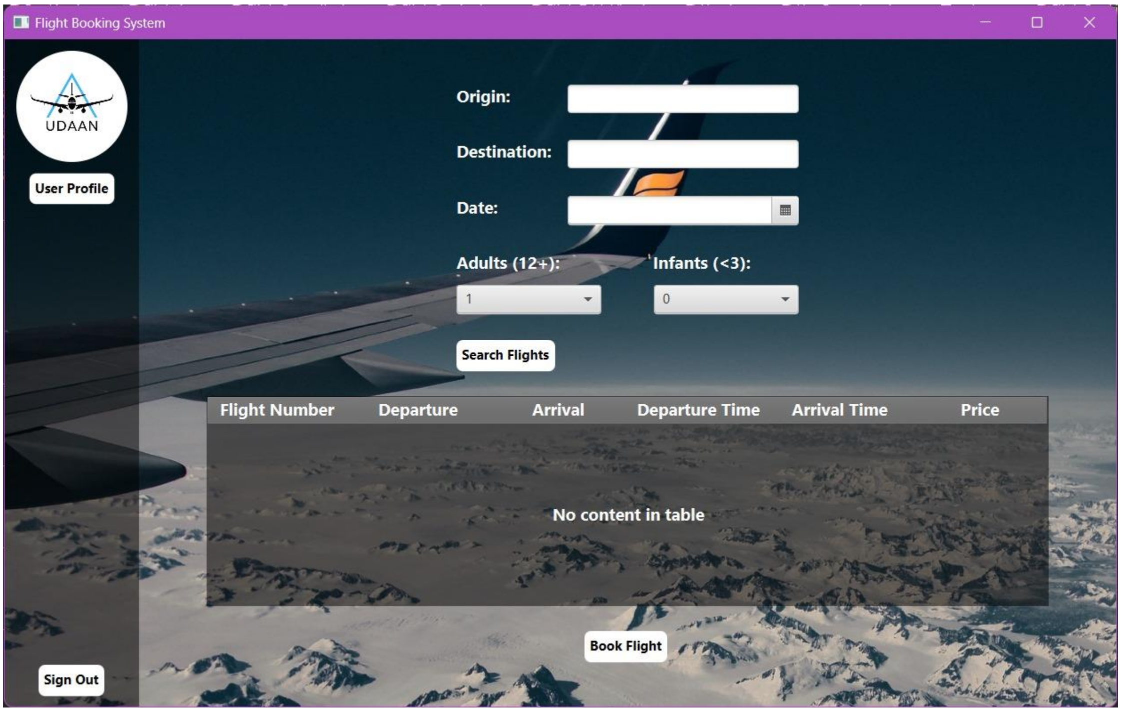
CREATE NEW ACCOUNT

Name:

Email:

Password:

Fig5:This is a Registering page where the user can register using their username email id and password



Flight Booking System

Origin:

Destination:

Date:

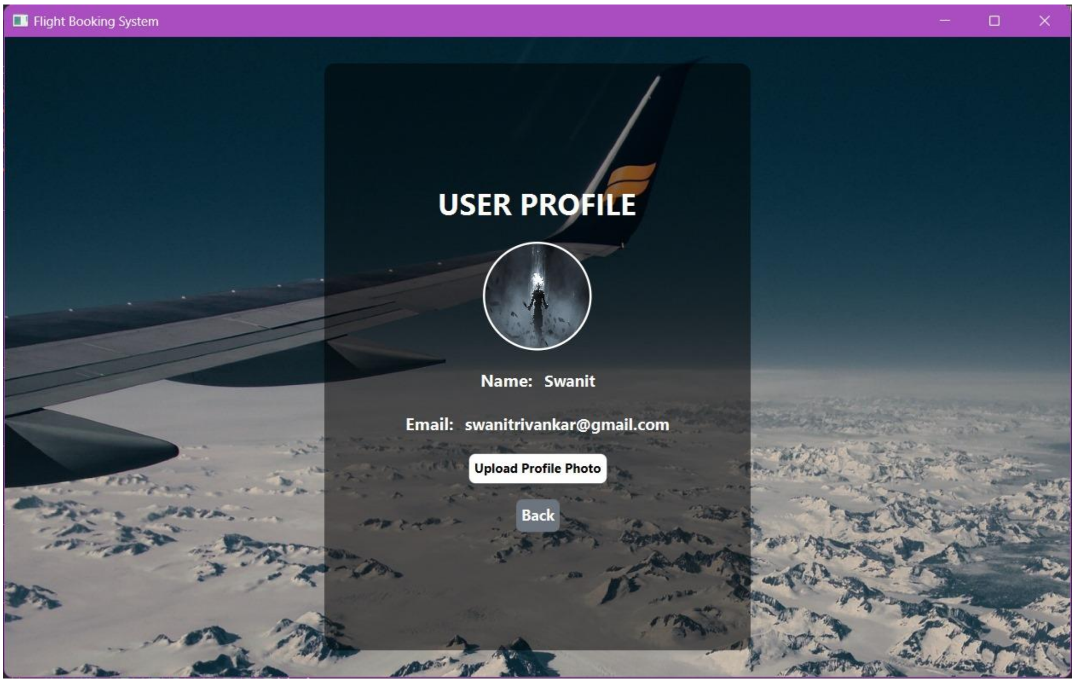
Adults (12+): 1

Infants (<3): 0

Flight Number	Departure	Arrival	Departure Time	Arrival Time	Price

No content in table

Fig6:

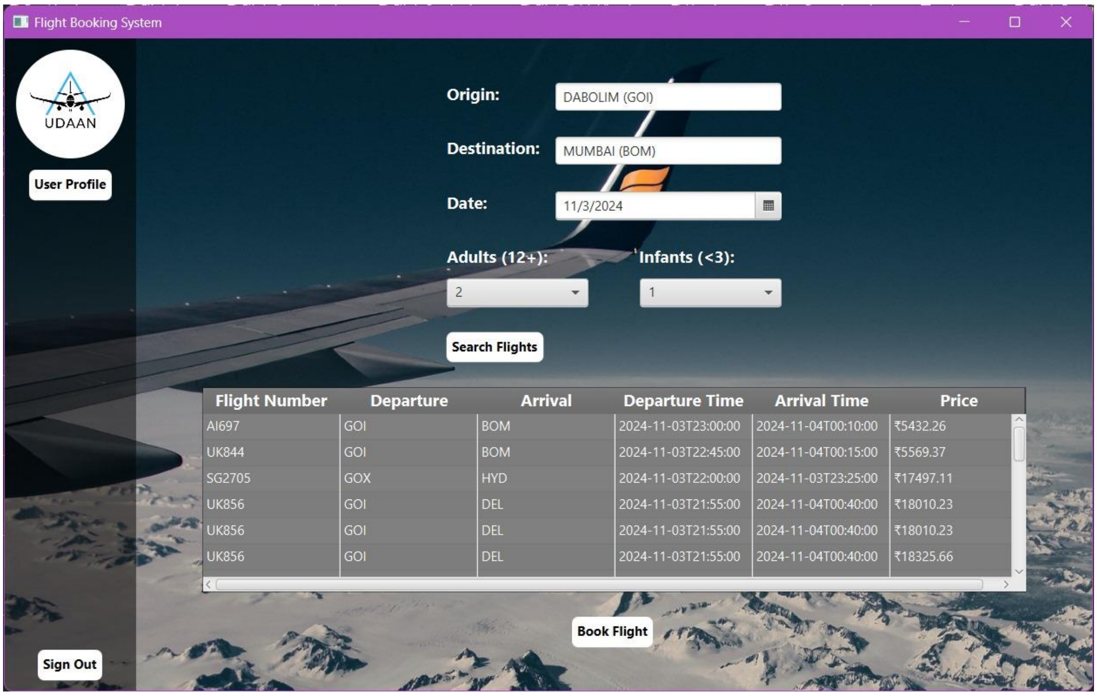


USER PROFILE

Name: Swanit

Email: swanitrivankar@gmail.com

Fig7: This the user profile which the user will see with their username and email id after logging in



Origin: DABOLIM (GOI)

Destination: MUMBAI (BOM)

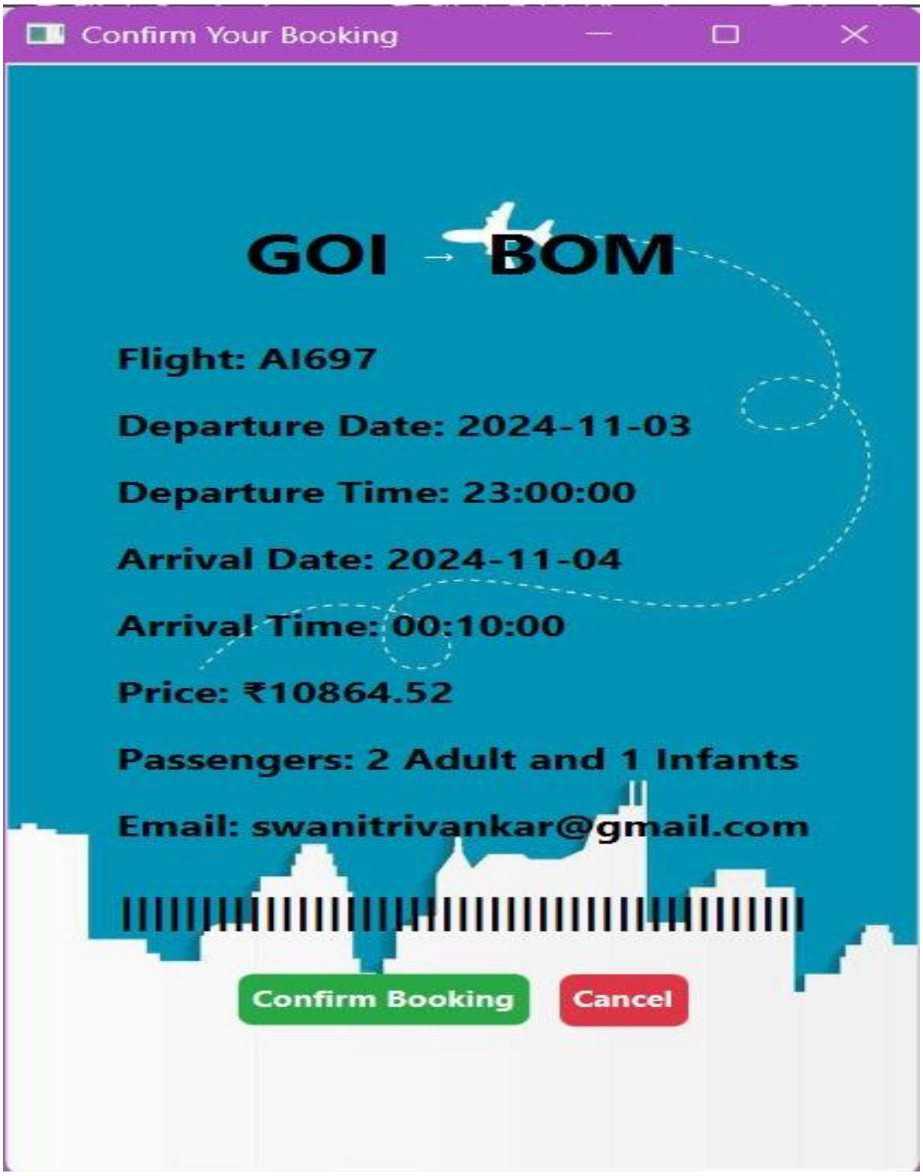
Date: 11/3/2024

Adults (12+): 2

Infants (<3): 1

Flight Number	Departure	Arrival	Departure Time	Arrival Time	Price
AI697	GOI	BOM	2024-11-03T23:00:00	2024-11-04T00:10:00	₹5432.26
UK844	GOI	BOM	2024-11-03T22:45:00	2024-11-04T00:15:00	₹5569.37
SG2705	GOX	HYD	2024-11-03T22:00:00	2024-11-03T23:25:00	₹17497.11
UK856	GOI	DEL	2024-11-03T21:55:00	2024-11-04T00:40:00	₹18010.23
UK856	GOI	DEL	2024-11-03T21:55:00	2024-11-04T00:40:00	₹18010.23
UK856	GOI	DEL	2024-11-03T21:55:00	2024-11-04T00:40:00	₹18325.66

Fig8: The user can search the flight by entering the origin, destination and date.The user can also specify the passenger count



Confirm Your Booking

GOI → BOM

Flight: AI697

Departure Date: 2024-11-03

Departure Time: 23:00:00

Arrival Date: 2024-11-04

Arrival Time: 00:10:00

Price: ₹10864.52

Passengers: 2 Adult and 1 Infants

Email: swanitrivankar@gmail.com

Fig9: This interface shows the details of the flight along with the price where the user can decide whether to proceed for payment or go back

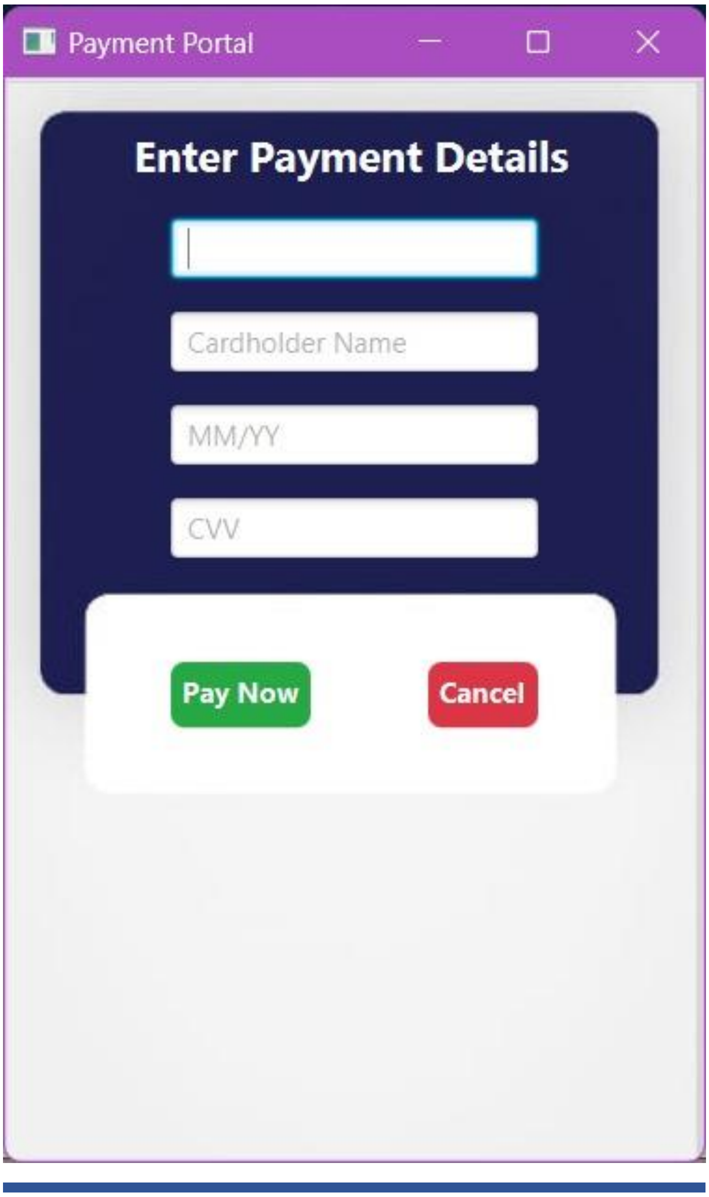


Fig10: This is a payment form where the user can pay using their card details

REFERENCES

https://developers.amadeus.com/

https://www.exchangerate-api.com/

https://openjfx.io/

https://fxdocs.github.io/docs/html5/

https://www.javatpoint.com/javafx-tutorial

https://www.mongodb.com/resources/languages/java

https://www.mongodb.com/docs/drivers/java/sync/v4.3/quick-start/

THANK YOU