LockedMe Application Product and Technical Specification document.

# Flyaway.com

Kartikaya Srivastav

v (0.1)

Author: Kartikaya Srivastav

LockedMe Application Product
and Technical Specification
document.

# 1. Introduction

1.1. Purpose
1.2. Requirement Overview

FlyAway is a ticket-booking portal that lets people book flights on their website.

**Code can be found at : https://github.com/ksrivastav/FlyawayWeb.git**

# 2. Product Specifications.

## 2.1. Product Specifications

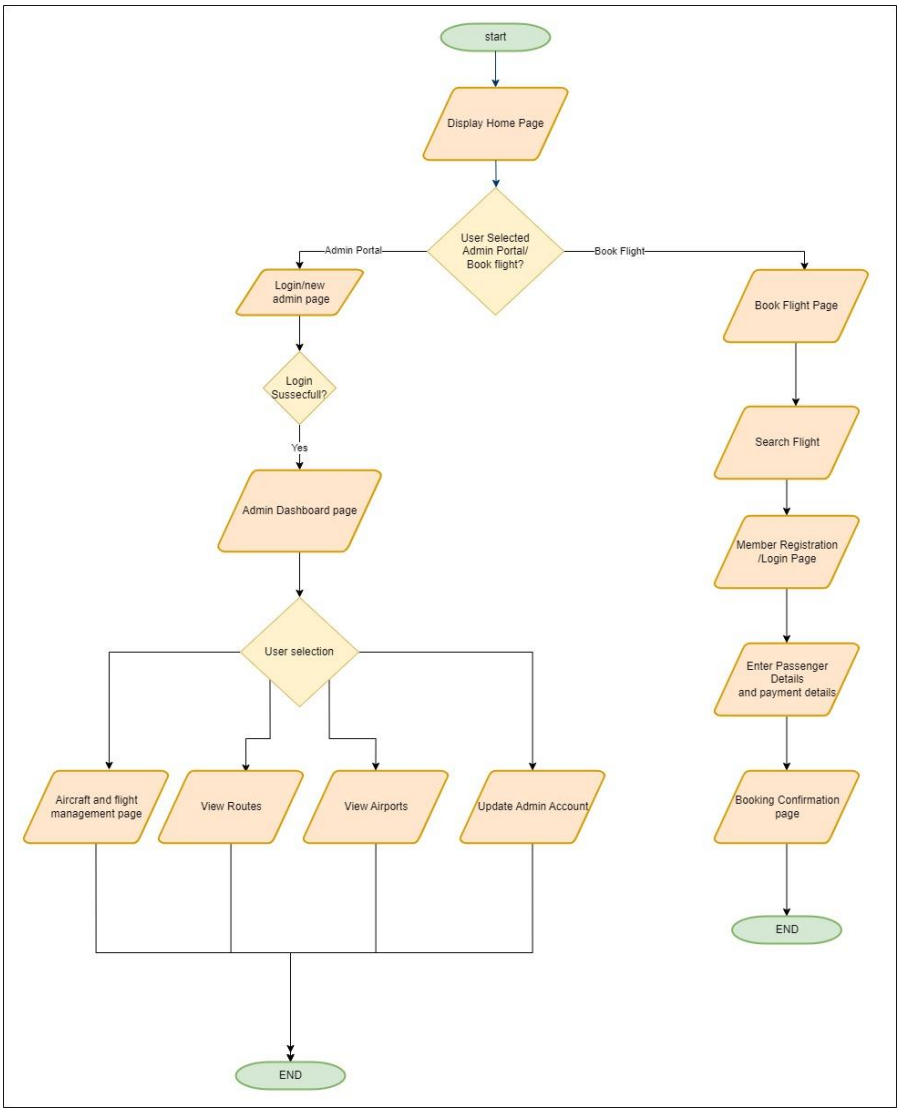
Flyaway Website will have following features.

I. ● A search form in the homepage to allow entry of travel details, like the date of travel, source, destination, and the number of persons.
● Based on the travel details entered, it will show the available flights with their ticket prices.
● Once a person selects a flight to book, they will be taken to a register page where they must fill in their personal details. In the next page, they are shown the flight details of the flight that they are booking, and the payment is done via a dummy payment gateway. On completion of the payment, they are shown a confirmation page with the details of the booking.

II. For the above features to work, there will be an admin backend with the following features:
III. ● An admin login page where the admin can change the password after login, if he wishes
● A master list of places for source and destination
● A master list of airlines
● A list of flights where each flight has a source, destination, airline, and ticket price

## 2.2. Applicaiton Flow Diagram

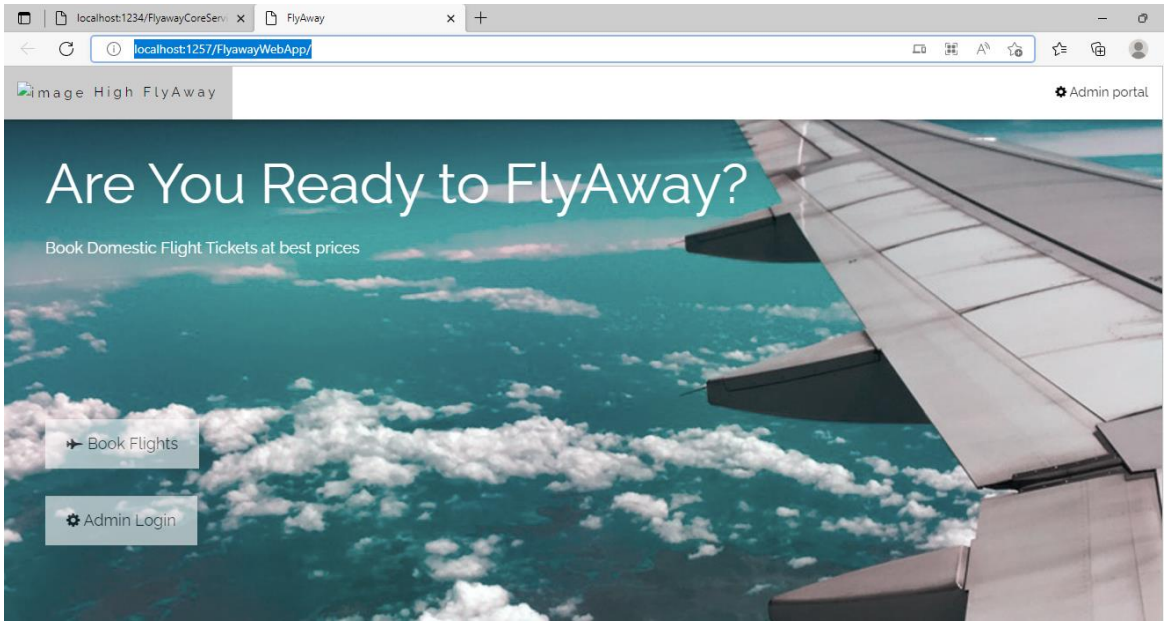
Author: Kartikaya Srivastav




## 2.3. User experience and user manual.

2.3.1. Flyaway is Spring MVC based web application.

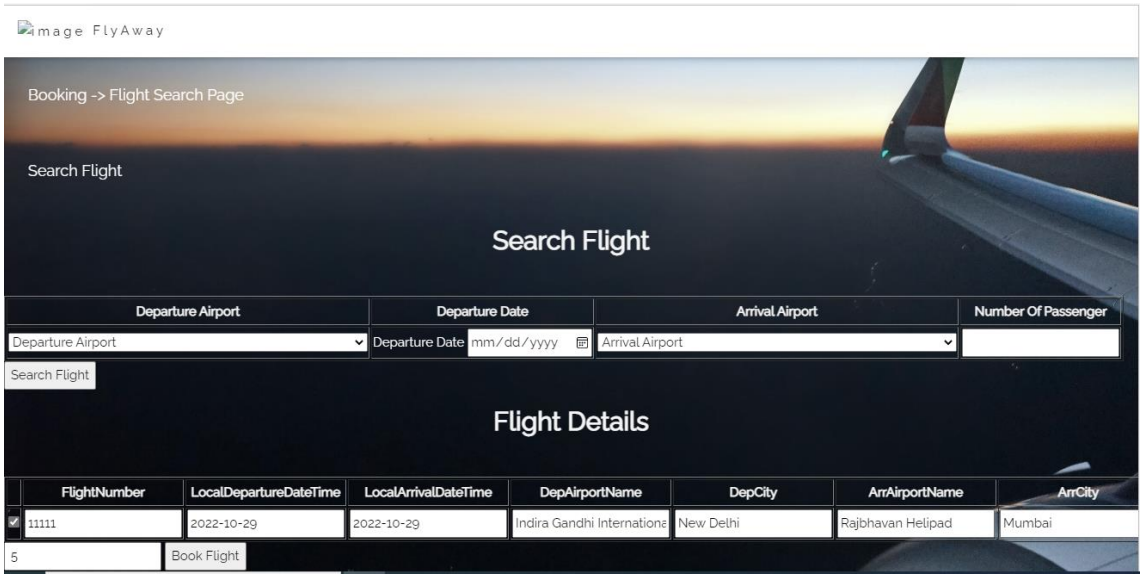
2.3.2. When the Application starts, you are landed on the home page. Below is the screenshot of the home page.




There are two menus in Flyaway web application. Users can choose from the menu simply by clicking on them.

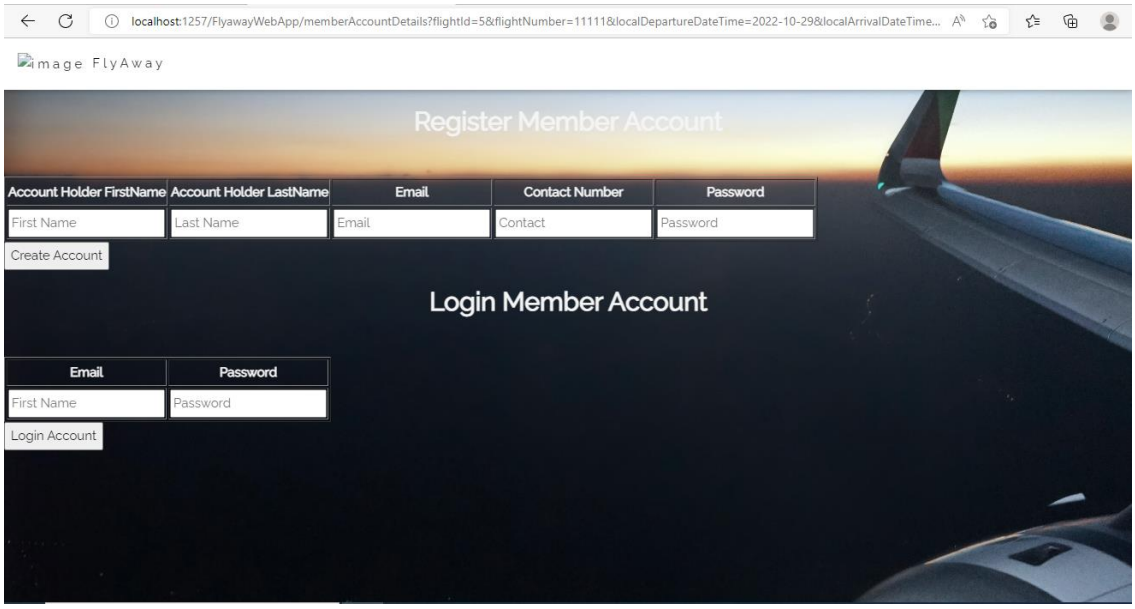
### 2.3.3. Working with Book Flight

2.3.3.1. After navigating to menu 1 from the main screen you will be landed on flight search page. Here you can see list of all available flights and a search panel. Using the search panel , you can search for a particular flight based on the search criteria.

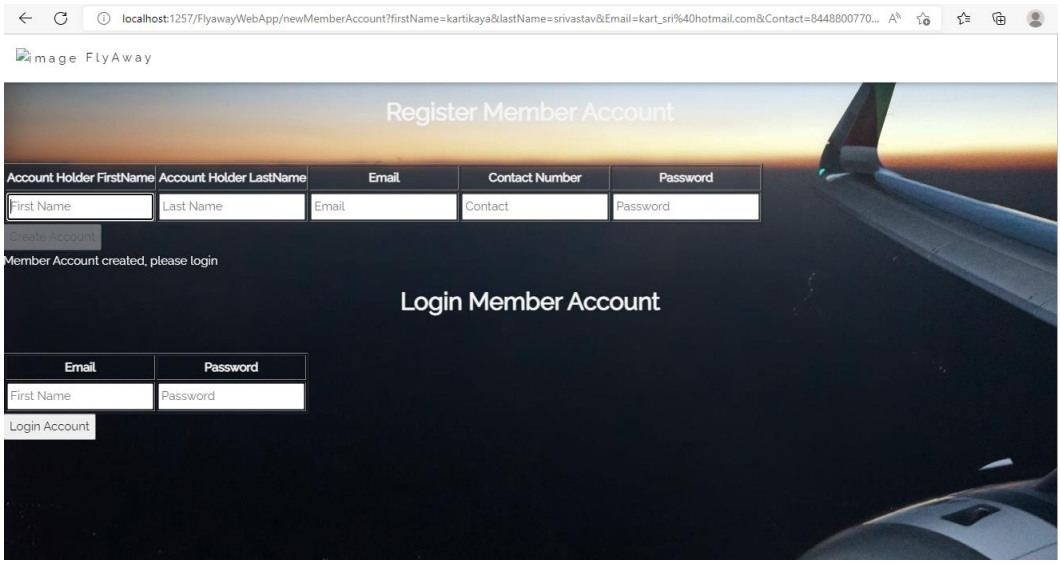



2.3.3.2. Select the flight you want to book and click on book flight.

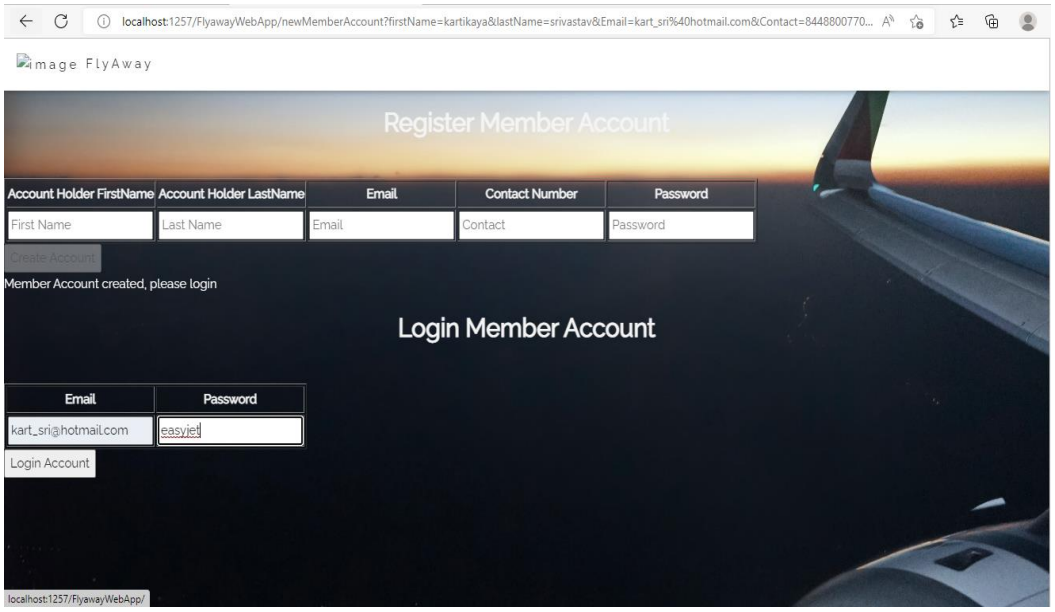
2.3.3.3. On the clicking book flight, you will be redirected to Member account login/registration page.




2.3.3.3.1. Here you can either create new Member account and then login.

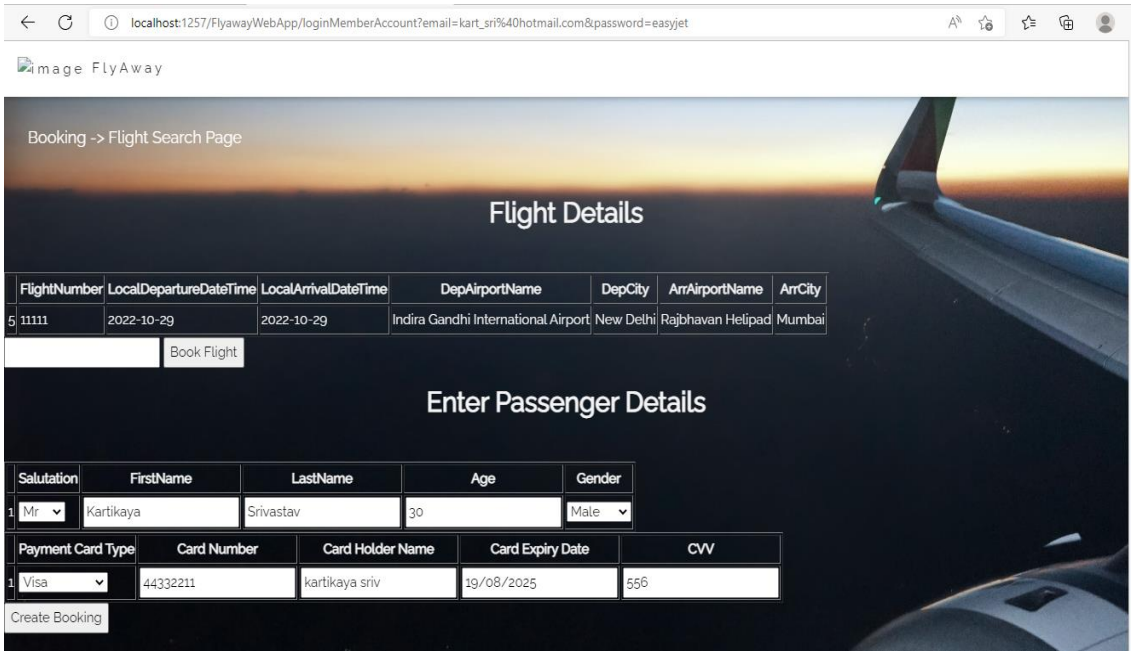



2.3.3.3.2. Or you can login straight away if you are an existing member account holder.

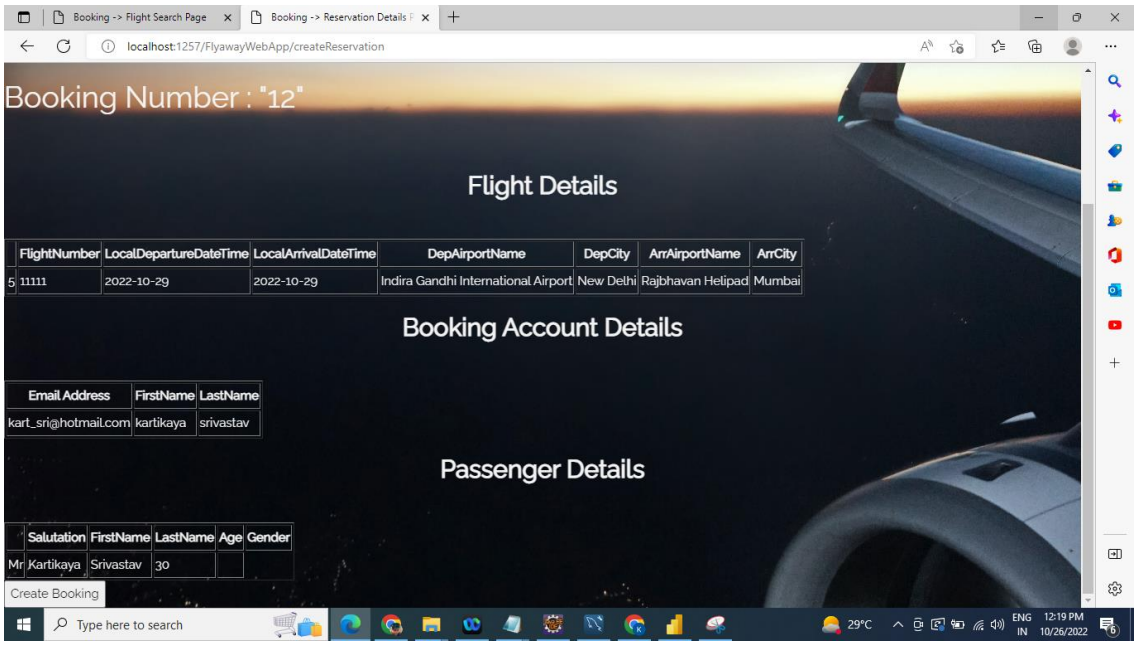



2.3.3.3.3.

2.3.3.4. Once you log in, you are redirected to passenger details and payment page. Here you can simply enter you passenger details and submit booking.

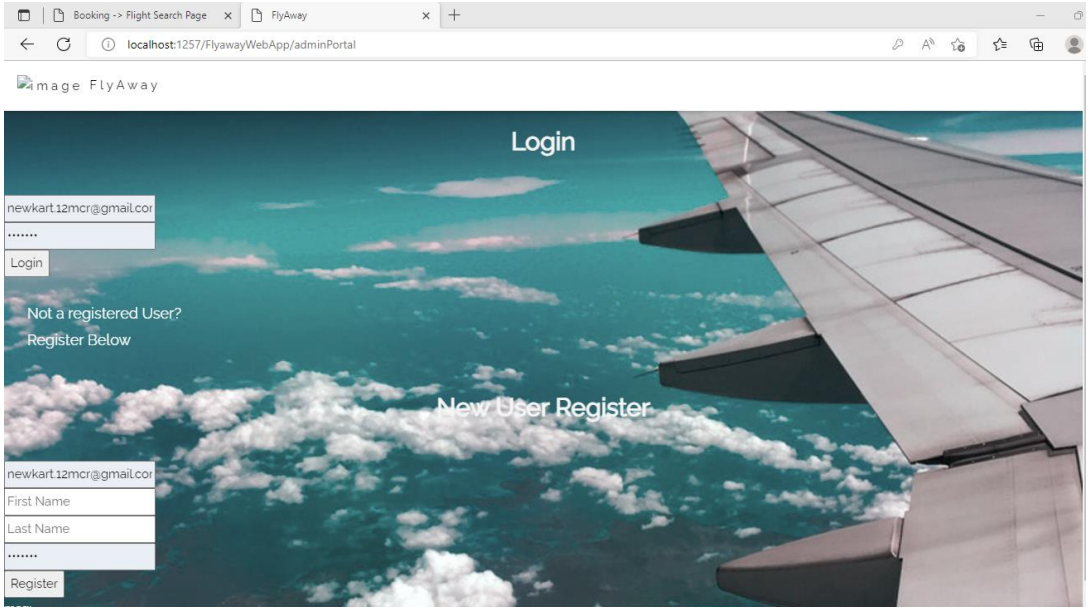


2.3.3.5. Once click on create booking, you will be displayed with booking confirmation page summarizing all your booking details.

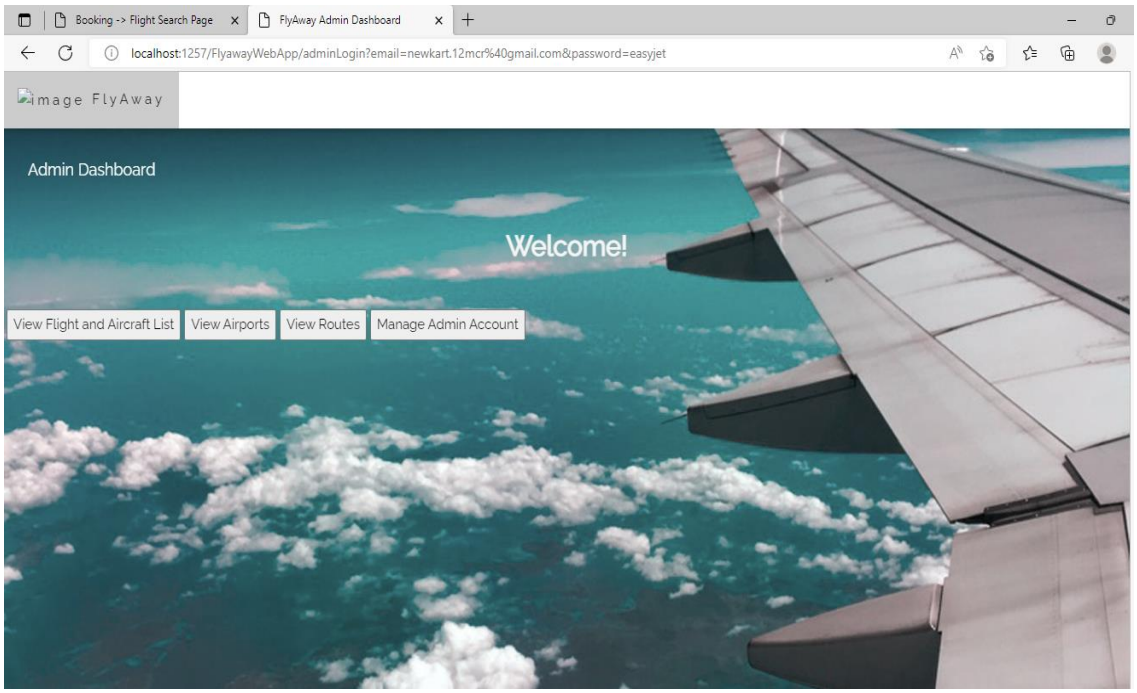



2.3.4. Working with Admin Portal.

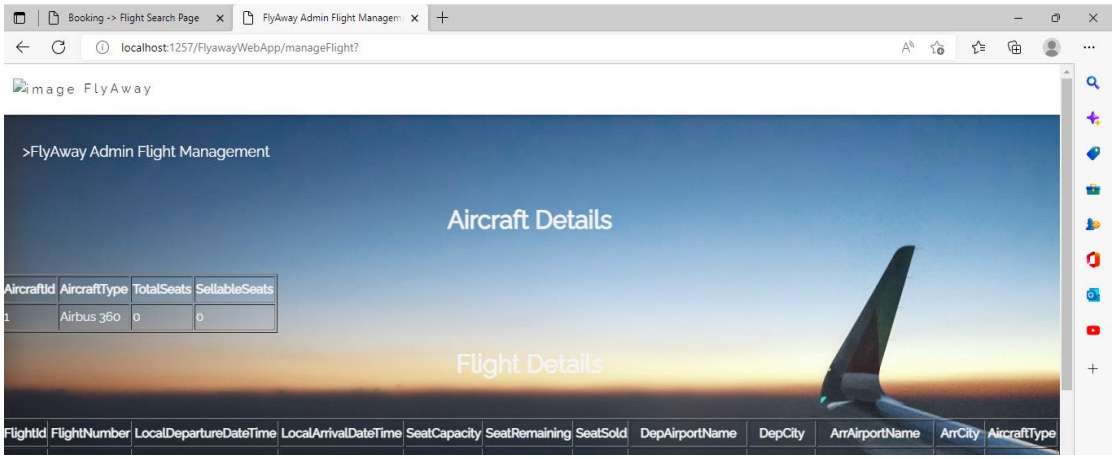
2.3.4.1. On home page, when you click on admin portal, you will be redirected to admin login/registration page.

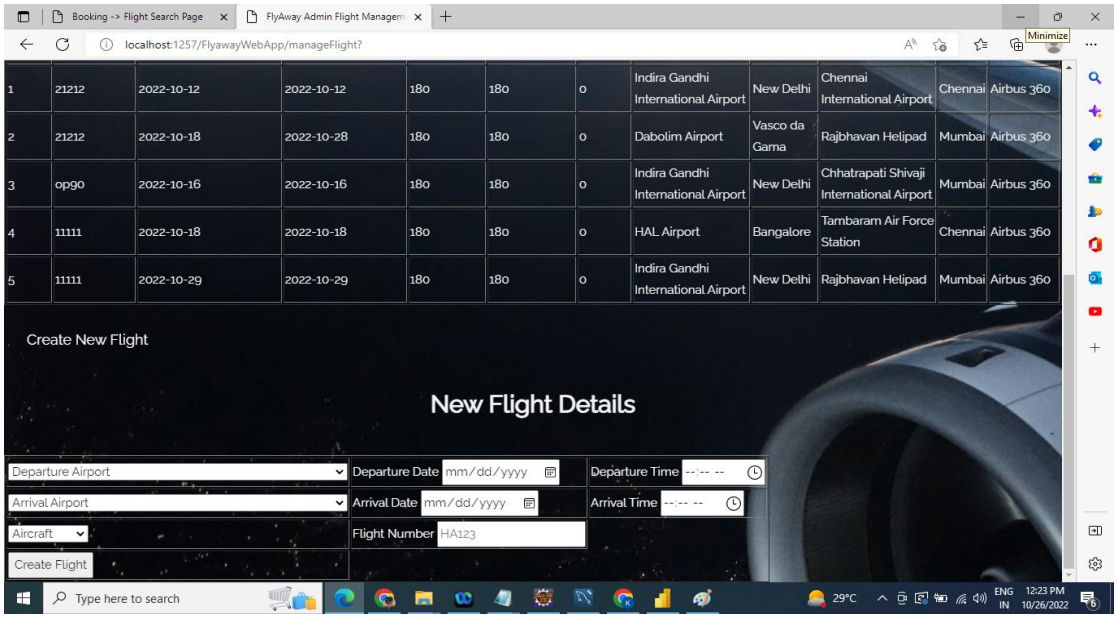



2.3.4.2. If you are existing Admin, you can straightaway login or you can first create admin account and then log in. After you log in, you will be redirected to admin dashboard. Here you are provided with four functionalities, you can choose any of them to proceed further.

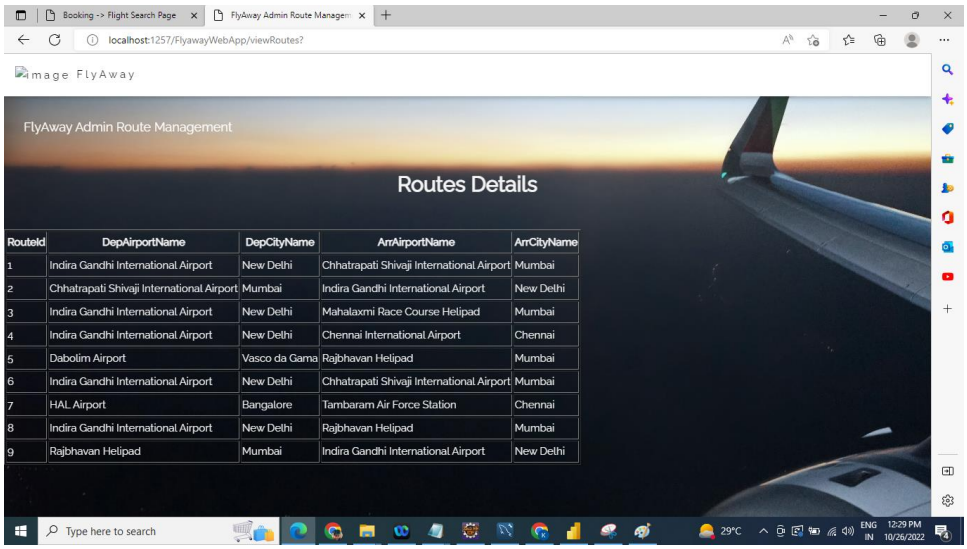



2.3.4.3. View Aircraft and Flights. On this page you can see a master list of all aircraft flyaway owns and all the flights. On this page you can also create new flights.

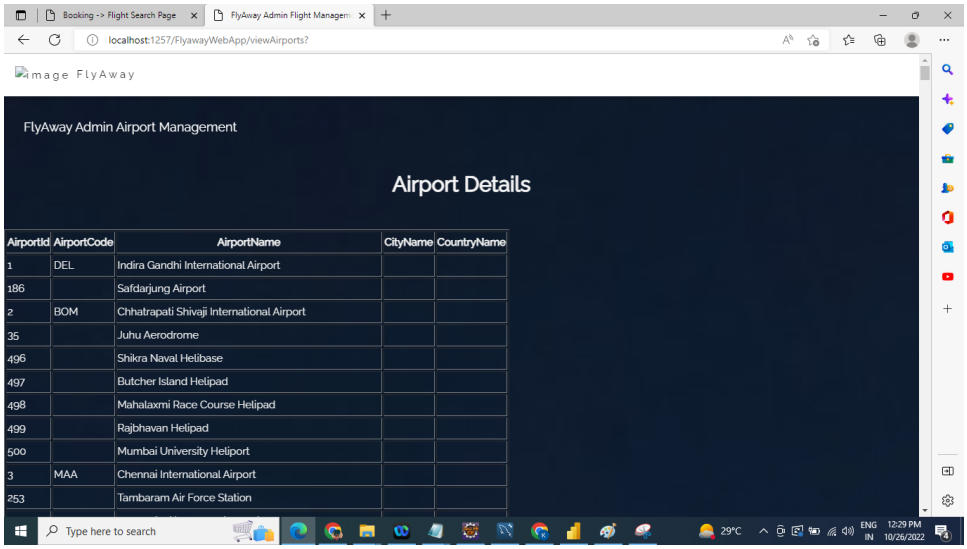




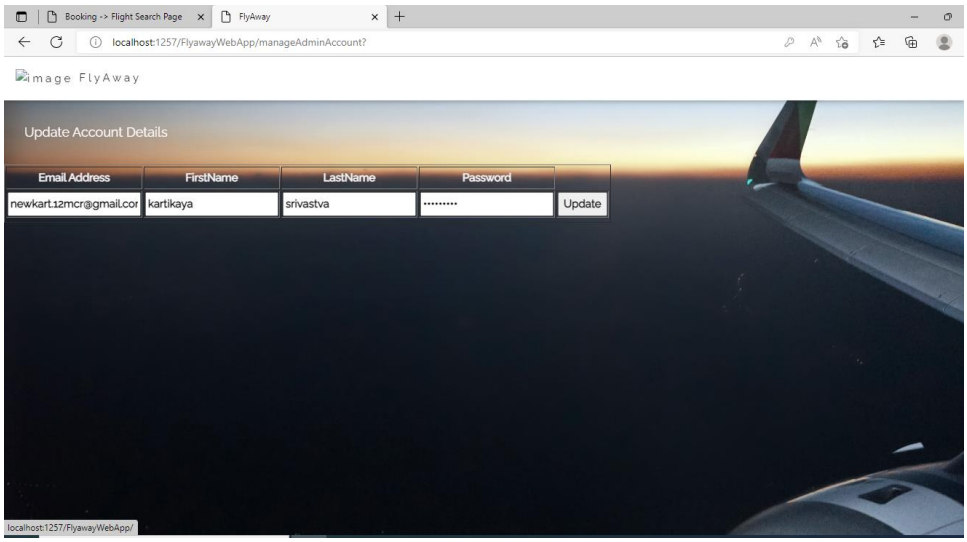

2.3.5. Similarly, you can select to View Routes. You will be redirected to view routes page where you can see master list of all routes Flyaway is operating on.




2.3.6. Similarly, you can select to View Airports. You will be redirected to view routes page where you can see master list of all airports Flyaway is operating on.

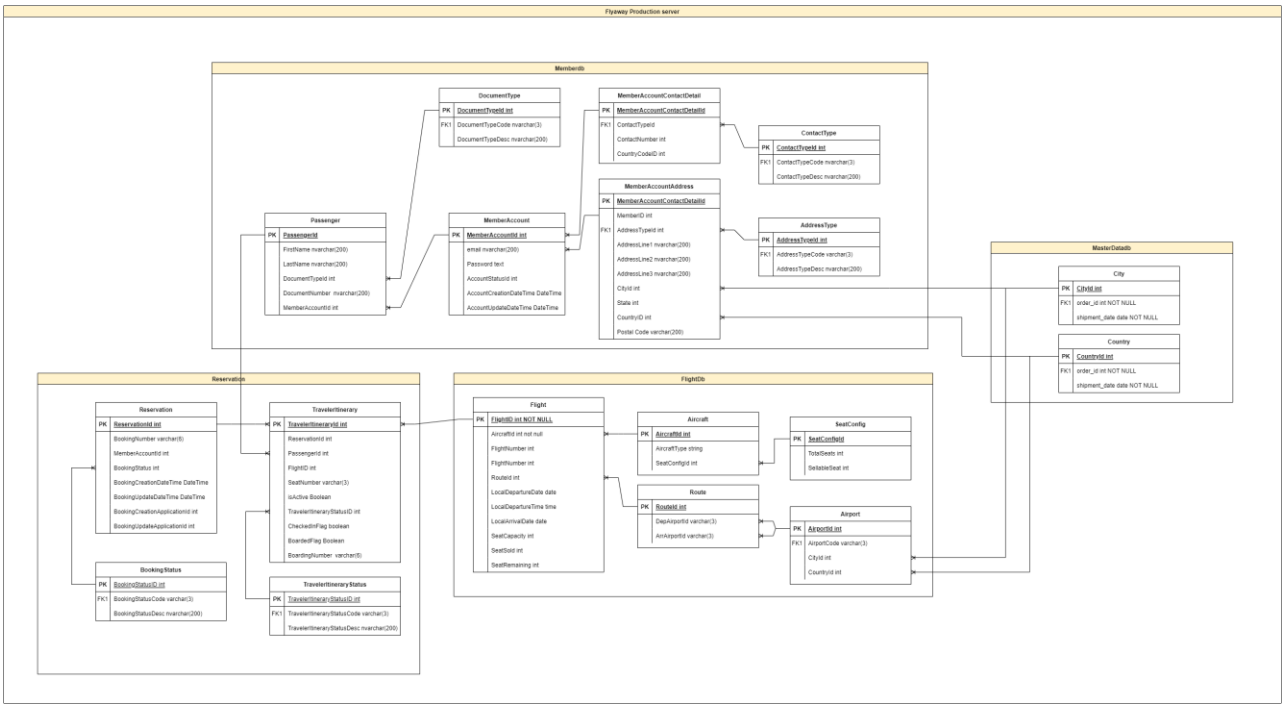



2.3.7. Similarly, you can select Admin account update where admin account holder can update account details including password.




# 3. System Design

## 3.1. ER Diagram




Database Summary:

- **Reservation:** this db contains table which stores data related to booking.
- **Flight:** this db contains data related to flight, airports and routes on which Flyaway operates.
- **Member:** this db contains data related to member account.
- **Masterdata:** this db is intended in storing common configuration and master data tables.

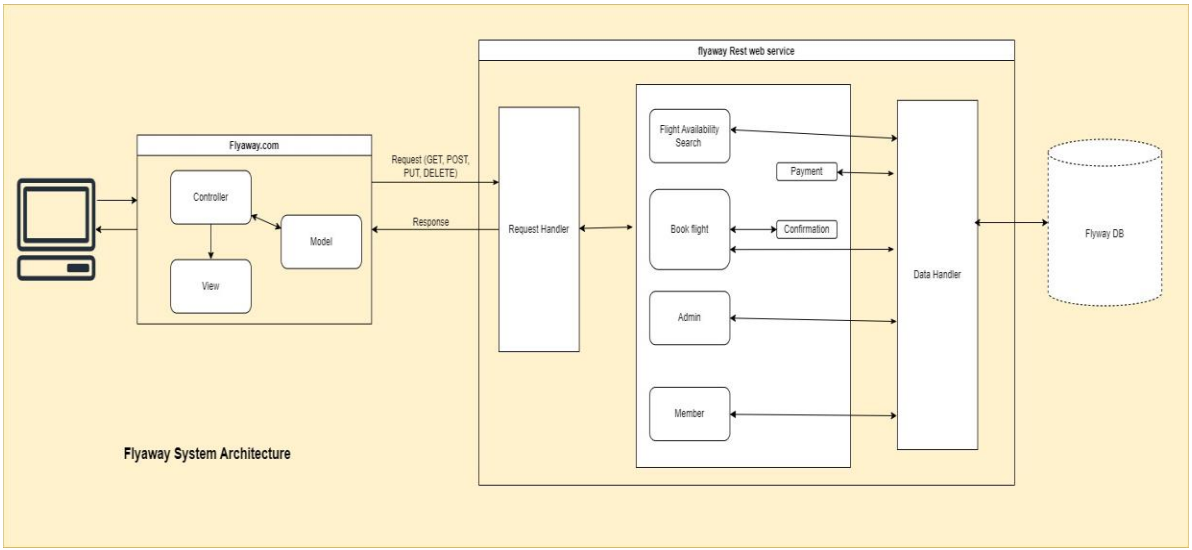
## 3.2. Flyaway System Architecture

On high level Flyaway is composed of three components

3.2.1. **Flyaway web app application:** This is a Spring MVC based web application which interacts with user and takes input and displays output to user.

**3.2.2. Flyaway core Rest Service:** is Spring rest service, it is the heart of flyaway system where all business logic is written. Flyaway web application interacts with flyaway core rest service for each operation, it performs. This service sits between db and flyaway web app and does all the heavy duty of processing business login and interacting with database.

**3.2.3. Flyaway db:** it is the main database server which stores all business data. Flyaway core interacts with db via spring hibernate.




### 3.3. Application Technical Specifications:

Application has been designed to stick to SOLID principles. Application is designed to be easily scalable where new functionalities can be added without impacting existing ones. The following points have been taken into consideration while designing the application:

- Application should be easily scalable.
- Application should have the capability to be added with more menu options easily without impacting existing menu options.
- Individual menu options can be added with new operations easily without impacting the existing ones.
- High code reusability but with maximum loose coupling.
- Thus, the application should be maximum closed for modification and open for extension.

#### 3.3.1. Design Pattern Used:

3.3.1.1. MVC
3.3.1.2. Data Access pattern
3.3.1.3. Dependency Injection

#### 3.3.2. Data structures used

**3.3.2.1. ArrayList: To perform handle collection of files.**
**3.3.2.2. HashMap: maps command with operations.**

# 4. Sprint Planning:

The Application has been implemented in two sprints

| Sprint Number | Days | Functionalities Delivered |
|---|---|---|
| 1 | 15 | • Operations package<br>• Helper package |
| 2 | 15 | • Menu package<br>• Application integration and system testing |

# 5. Unique Selling Points:

5.1. Application is extremely easy to use.

5.2. Loose coupling between business and presentation logic.

5.3. Highly scalable application due to implementation of Rest service.

5.4. Web application implemented using Spring MVC provides extremely flexible and agile capabilities.

# 6. Further improvements:

6.1. Allow users to book multiple flights in one booking.

6.2. Integration of resource file to configure repetitive config and data.