

**PG FSD - Phase-2: Become a Backend expert**



**FlyAway – Airline Booking Portal**

Submitted By:

Parijat Singh

# Product Specifications

**Background of the problem statement:**

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

**The website needs to have the following features:**

● 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.

For the above features to work, there will be an admin backend with the following features:

● 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

# Execution Details

## Sprint 1:

Key features delivered in Sprint 1:

* Home screen with Flight Search
* Flight Search Results
* Error handling on each screen for invalid or incomplete fields

**User Story 1:** As a customer, I would like to enter my search criteria and search for available flights so that I can see the flight options available.

Story Points: 8

## Sprint 2:

Key features delivered in Sprint 2:

* Register customer details for a selected flight
* Book selected flight with registration details
* Display flight confirmation

**User Story 1:** As a customer, I would like enter details of passengers and payment so I can book my reservation.

Story Points: 5

**User Story 2:** As a customer, I would like to get a confirmation of the flight booking so that I can confirm the booking.

Story Points: 5

## Sprint 3:

Key features delivered in Sprint 3:

* Admin login
* Change password
* Display List of Places, Airlines and Flights

**User Story 1:** As an administrator, I would like to login to the application so that I can perform administrative functions.

Story Points: 3

**User Story 2:** As an administrator, I would like to be able to change my password, so that I can keep my login secure.

Story Points: 5

**User Story 3:** As an administrator, I would like to list all the places, airlines and flights at my choice so that I can review them

Story Points: 5

# Application components and flow



# GitHub Repository

The code can be accessed in the Master branch of the GitHub Repository :[**Parijat\_Singh\_Flyaway\_Flight\_booking**](https://github.com/parijat-singh/Parijat_Singh_Flyaway_Flight_booking)

<https://github.com/parijat-singh/Parijat_Singh_Flyaway_Flight_booking>

# mySQL Tables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:bookings** |  |  | **Table:airlines** |  |
| **Columns:** |  |  | **Columns:** |  |
| **id** | int AI PK |  | **id** | int AI PK |
| flight\_id | int |  | airline\_code | varchar(3) |
| num\_of\_seats | int |  | airline\_name | varchar(45) |
| payer\_name | varchar(45) |  | last\_updated | datetime |
| payer\_zip | varchar(45) |  |  |  |
| creditcard\_num | varchar(45) |  | **Table:flights** |  |
| expiration | varchar(45) |  | **Columns:** |  |
| total\_payment\_amount | float |  | **id** | int AI PK |
| passenger1\_name | varchar(45) |  | flight\_num | int |
| passenger1\_gender | varchar(45) |  | airline\_code | varchar(3) |
| passenger1\_dob | varchar(45) |  | from\_airport | varchar(3) |
| passenger2\_name | varchar(45) |  | to\_airport | varchar(3) |
| passenger2\_gender | varchar(45) |  | start\_datetime | datetime |
| passenger2\_dob | varchar(45) |  | end\_datetime | datetime |
| passenger3\_name | varchar(45) |  | num\_of\_seats\_available | int |
| passenger3\_gender | varchar(45) |  | price | float |
| passenger3\_dob | varchar(45) |  | last\_updated | datetime |
| passenger4\_name | varchar(45) |  |  |  |
| passenger4\_gender | varchar(45) |  | **Table:places** |  |
| passenger4\_dob | varchar(45) |  | **Columns:** |  |
| last\_updated | datetime |  | **id** | int AI PK |
|  |  |  | airport\_code | varchar(3) |
| **Table:login** |  |  | airport\_name | varchar(45) |
| **Columns:** |  |  | city\_name | varchar(45) |
| **loginid** | varchar(45) PK |  | last\_updated | datetime |
| password | varchar(45) |  |  |  |
| last\_update | datetime |  |  |  |

# Key Concepts used

The following concepts from the Backend Web Development course were used:

1. Servlet
2. jsp
3. jdbc
4. Maven
5. Session handling
6. mySQL

# Instructions for Execution of Web Application:

* Setup Database: Please use the Attached SQLs to create Schema and tables used in the application.



* Insert some rows into all tables except bookings. Alternatively, use the attached SQL’s below to insert rows into your table.



* Import the Maven Project and all folders from GitHub into Eclipse
* Under src/main/java, navigate to package com.simplilearn.workshop.utils and open the MySQLDatabaseUtils.java.
* Modify user and password to your DB user and password and save it.
* Update Maven Project
* Run the application on a tomcat server.