NAME – MUKHEE JYOTHSNA

COLLEGE – MVGR COLLEGE OF ENGINEERING, VIZIANAGARAM

SIMPLILEARN PHASE-2 PROJECT

PROJECT DESCRIPTION

Tech Stack Used :- 1. JAVA

2. HIBERNATE

3. SERVLETS

Server Used – Tomcat Version 8.5

//brief description of project

The **features** of project are:-

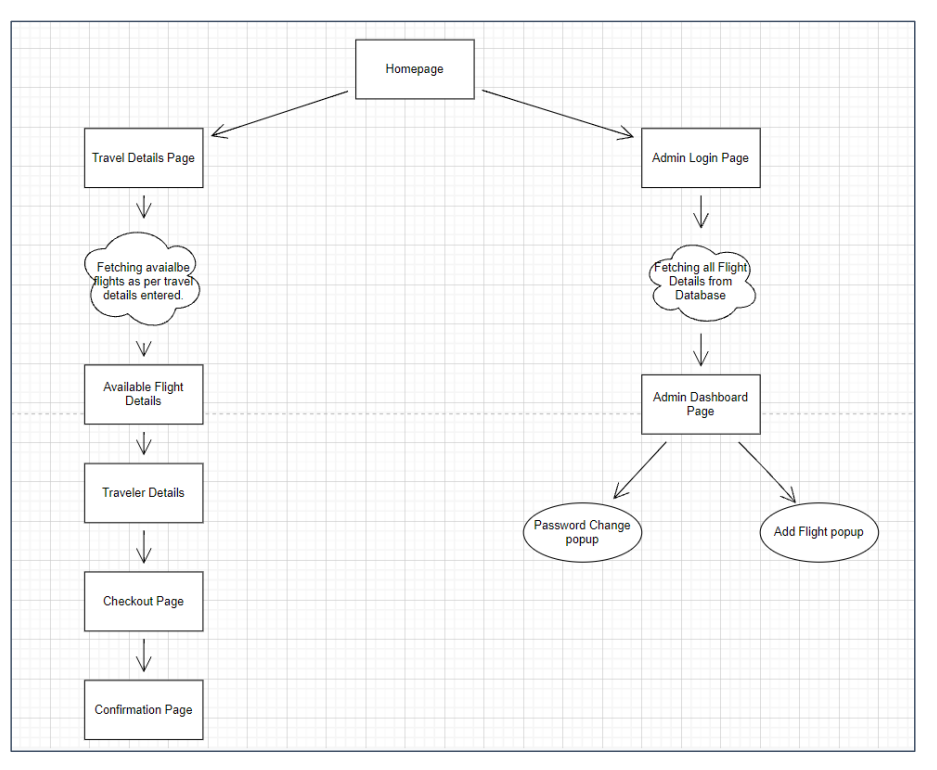
1. Option for user to login/register.
2. A search form in the homepage to allow entry of travel details, like the date of travel, source, destination, and the number of persons.
3. Based on the travel details entered, the available flights with their ticket prices are shown.
4. Once a person selects a flight to book, they are taken to a register page where they must fill in their personal details.
5. 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.
6. An admin login page where the admin can change the password after login, if he/she wishes, after logging in.

A master list of places for source and destination, which can be viewed.

A master list of registered airlines, which can be viewed as well.

A list of flights where each flight has a source, destination, airline, and ticket price.

FLOW CHART



Algorithm Explanation/ Flow of the application

1. A Book Flight button in the homepage, which opens a search form to enter

travel details that is the date of travel, source, destination, and the number of

travellers.

2. Based on the travel details entered, it will show the available flights with their

ticket prices fetching data from the MySQL database, connected to JSP and

Servlets via Hibernate, and is hosted on AWS RDS for MySQL.

3. Once a user selects a book this flight button, they will be taken to a register

page to fill in their personal details.

4. In the next page, they are shown the flight details of the flight that they are

booking and form to enter payment details along with checkout button.

5. After clicking the checkout button, they are shown a confirmation page with

the details of the booking.

6. For the above features to work, there is an admin backend with the following

features:

7. An admin login page, where anyone can register as an admin and the

admin can change password after login, if they wishes.

8. A list of flights where each flight has a source, destination, airline, and

ticket price is shown on the admin dashboard.

PROJECT USER STORIES

1. As a developer, I want to build register/login functionality.
2. As a developer, I want to allow user to view flights as per preference.
3. As a developer, I want to allow admin to add/modify flight related information
4. As a developer, I want the user to be able to successfully view flight prices and names as per preference
5. As a developer, I want the user to be able to book a flight via a payment gateway
6. As a developer, I want the payment gateway to be authenticated properly to prevent any flaws in the transaction process
7. As a user, I want to book a flight securely, with proper amount deductions from my account, post the booking process

Sprint 1

1. As a developer, I want to build register/login functionality.
2. As a developer, I want to allow user to view flights as per preference.
3. As a developer, I want to allow admin to add/modify flight related information

Sprint 2

1. As a developer, I want the user to be able to successfully view flight prices and names as per preference
2. As a developer, I want the user to be able to book a flight via a payment gateway
3. As a developer, I want the payment gateway to be authenticated properly to prevent any flaws in the transaction process

Sprint 3

1. As a user, I want to book a flight securely, with proper amount deductions from my account, post the booking process

PROJECT GITHUB REPOSITORY

Link –https://github.com/jyothsnamukhee03/FlyAway

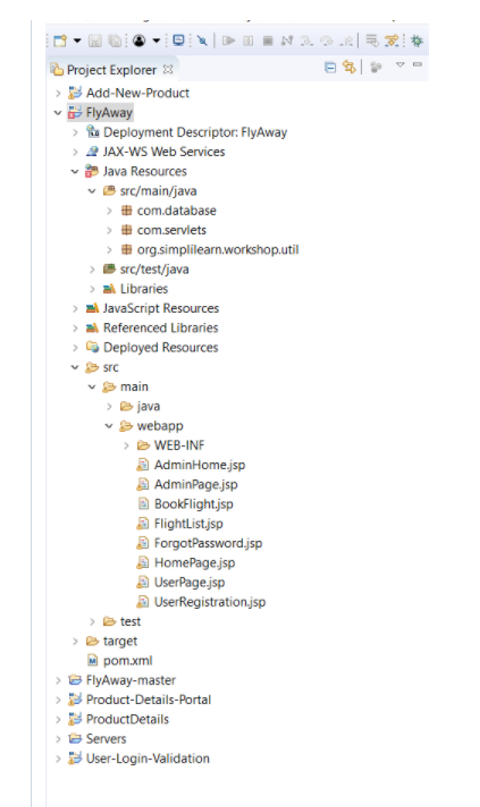
How To Run Project

1. Clone Project
   1. Clone git:

git clone https://github.com/jyothsnamukhee03/FlyAway.**git**

1. Right click on the project 🡪 Run on Server 🡪 and Execute the code.

Directory Structure/Package



**Unique Selling Points of the Application**

1. The application is designed to keep on running and taking user inputs even after exceptions occur. To terminate the application, appropriate option needs to be selected.

2. The application is designed to take input from the Signup and login forms

3. The application also allows user to CRUD in DBS.

4. The application is designed with modularity in mind. Even if one wants to update the path, they can change it through the source code.

**Conclusion**

Further improvements can be made on the application which include:

Adding Bootstrap, Javascript, CSS designs and attributes to make the forms responsive for the user.

Adding Flights database with multiple locations for Airports and adding more variables to make the dbms easier to manipulate the data.