**Phase 2 Assessment**

**Flyaway Documentation**

This document contains sections for:

* Sprint Planning and Task Completion.
* Core concepts used in the project.
* Flow of the Application
* Documentation of the flow of the application
* Demonstrate the product capabilities, appearance and user interactions.

The code for this project is hosted at <https://github.com/pree1609/FlyAway> .

The project is developed by Preeti Das.

**Sprints Planning and Task Completion**

The project is planned to be completed in 2 sprints each of 2 weeks duration. Tasks assumed to be completed in 1st sprint are:

* Creating the flow of the application
* Initializing git repository to track changes as development progresses.
* Writing necessary programs to fulfill the requirements of the project.
* Testing the application with different kinds of User input to check if all edge cases are satisfied.

Tasks assumed to be completed in 2nd sprint are:

* Pushing code to GitHub.
* Creating this specification document highlighting application capabilities, appearance, and user interactions.

The task was completed within the above timeframe of both the sprints successfully.

**Core concepts used in the project**

1. Servlets
2. Java Server Pages
3. Exception handling
4. Maeven dependencies
5. MySQL workbench
6. Session management: hidden fields
7. Filters

**Flow of the application**

Diagram

Description automatically generated

## 

## **Documentation of the flow of the application**

User:

1. The user enters the source, destination, date of travel and number of persons in the search flights form.
2. The list of available flights gets displayed to the user according to the search filters provided.
3. The user selects one of the flights and goes to the next page for registration.
4. If appropriate flight is not found for user, they gets directed to “Flight Not Found”.
5. Once the user completes the registration they are navigated to the payment gateway.
6. And on successful payment the user is directed to the confirmation page, where booking details of the user is displayed.

Admin:

1. The admin logs in to the admin portal.
2. Admin dashboard is displayed.
3. Various buttons are displayed to admin on which the admin gets redirected to various lists to monitor, such as: list of airlines, list of all flights operating, list of all source and destinations.
4. The admin can navigate to any of the lists and monitor them.
5. The admin can also change password of the admin portal he is logging into.

## **Demonstrating the product capabilities, appearance, and user interactions**

1. The User can search flights, select from the available list and book a flight using the payment gateway.
2. Admin can login into admin dashboard where he can monitor the flights, like get a list of all flights available, a list of source and destinations and also a list of airlines operating.
3. Admin can also change their password using the Change Password link. Admin can change their password only after logging in to ensure maximum security.
4. The list of flights, source-destinations and airlines will be available to the admin only if he is logged in to enhance maximum security of the portal.
5. Minimalistic user interface has been made so that it looks appealing and yet easy to use for users and even admins.

The code has been pushing to Github repository by the following steps:

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize repository using the following command:

**git init**

* Add all the files to your git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m <commit message>**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**