

FlyAway.com

Developer Details:

Name : B Sangappa

Email : sangamesh.b.007@gmail.com

Application Overview

FlyAway.com is a simple Web Application which has two portals, one for Users/Travelers, where they can search for Flights and Book the flight tickets and another portal is for Admin, where the company Admin can manage the Flight details.

Sprint Planning

The FlyAway.com project is planned to be completed in 2 sprint and the tasks assumed to be completed in the sprint are:

1. Set the Development environment i.e., Eclipse, Git and GitHub
 - a. Setting up the Eclipse
 - b. Creating a Repository in Git
 - c. Setting up the GitHub Repo
 - d. MySQL Server and Workbench
 - e. Tomcat Server
2. Creating the High level Flow diagram of the application
3. Develop a Dynamic Web Java project to fulfill the requirements of the project
4. Testing the Dynamic Web Java project with different kinds of User input through CMD line
5. Pushing code to GitHub
6. Creating this specification document highlighting application functionalities, appearance, and user interaction

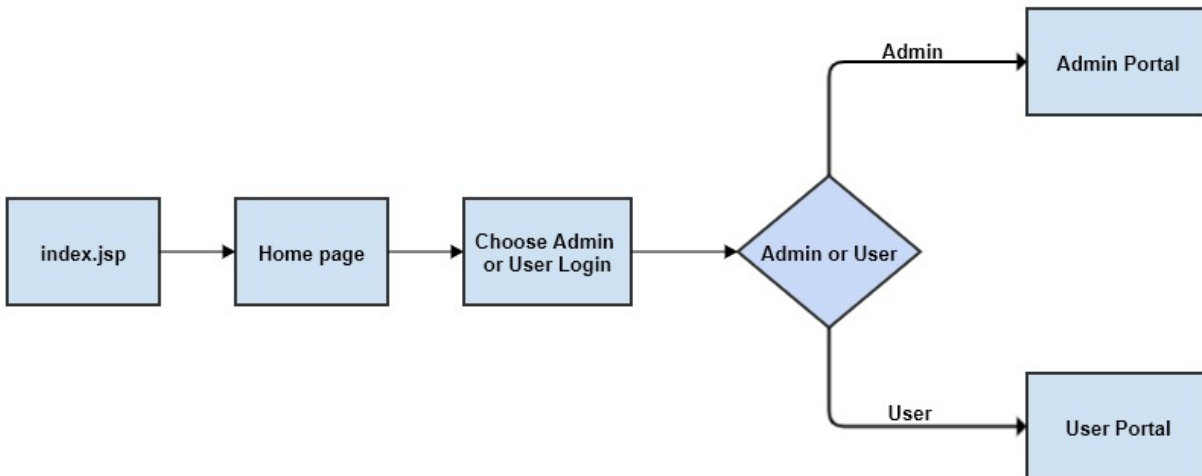
Core and Technical Concepts used in the application

1. **Frontend** : HTML, JSP, CSS, Bootstrap
2. **Backend** : Java, Hibernate, MySQL
3. **Server** : Tomcat
4. **Misc** : XML, Git and GitHub

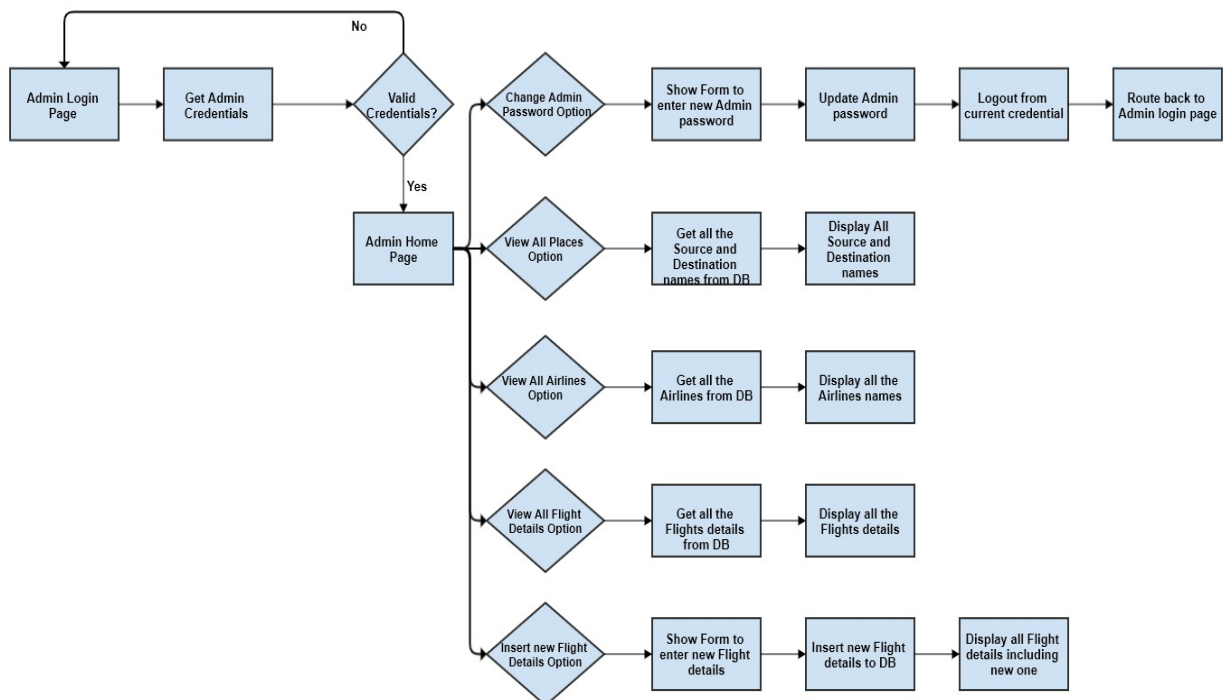
Flow of the Application

Below is the High level flow diagrams which shows the flow of the FlyAway.com application

Index :



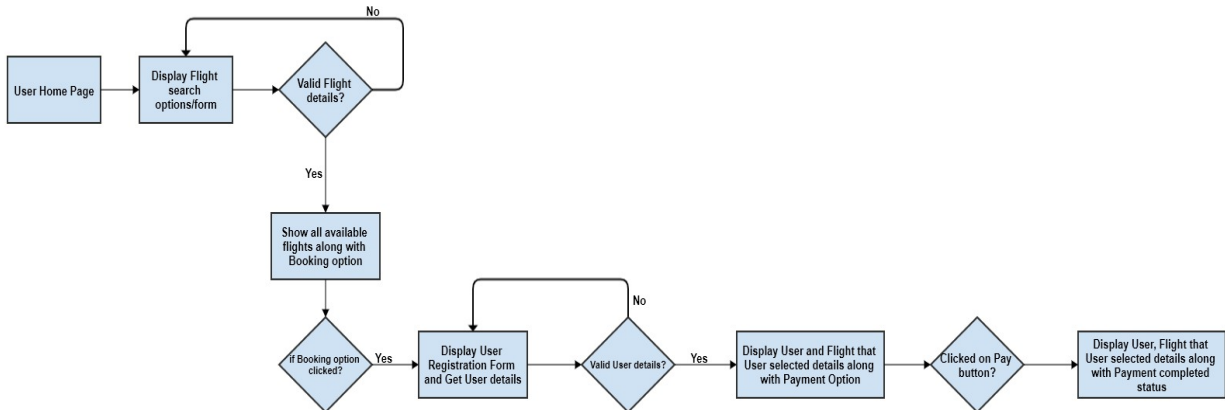
Admin Portal :



In the Admin portal, admin can

1. Change password for Admin account
2. View all Source and Destinations
3. View all Airlines and associated Flights
4. View all Flight details like source, destination, price, airline name and flight name

User Portal :



In the User portal, user can

1. Search for flights with date, source, destination and number of passengers details
2. Book flights by registering
3. Single click payment

Note :

1. All the Flight details will be stored in backend Database(MySQL)
2. All the Airlines details will be stored in backend Database(MySQL)
3. All the User details will be stored in backend Database(MySQL)

Demonstrating the Product capabilities, Appearance, and User interactions

1. Index.jsp :

Below is the code from index file of FlyAway.com App which displays the Login options for Admin and search flight options to User

Referring : adminlogin.jsp, userHomePage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>FlyAway</title>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

<link href="style.css" rel="stylesheet" type="text/css">

</head>

<body>

    <nav class="navbar navbar-default">

        <div class="container-fluid">

            <div class="navbar-header">

                <a class="navbar-brand indexPageNavBar"
href="index.jsp">FlyWay</a>
```

```

        </div>

    </div>

</nav>

<div align="center">

    <h1 class="indexPageNavBar marginTop">FlyAway Login Page</h1>

    <div style=" margin-top: 50px">

        <a href="adminlogin.jsp" style="color: black; font-family: cursive;
padding-right: 20px; font-size: 30px">Admin</a>

        <a href="userHomePage.jsp" style="color: black; font-family: cursive;
font-size: 30px">User</a>

    </div>

</div>

</body>

</html>

```

2. adminlogin.jsp :

Below is the code from adminlogin file of FlyAway.com App which displays the Login form for Admin account

URL Mapping : /login

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>FlyWay - Admin</title>

<link rel="stylesheet"

```

```

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand indexPageNavBar"
href="index.jsp">FlyWay</a>
            </div>
        </div>
    </nav>
    <div align="center" class="marginTop">
        <div>
            <h3 class="indexPageNavBar">Admin Login</h3>
            <form action="login" method="post">
                <div class="form-group">
                    <input name="adminUsername" type="text"
class="form-control"
                    style="width: 400px" placeholder="Admin
Username *" />
                </div>
                <div class="form-group">
                    <input name="adminPassword" type="password"
class="form-control"
                    style="width: 400px" placeholder="Admin
Password">

```

```

        </div>

        <div class="form-group" style="margin-top: 10px;">
            <input type="submit" value="Login" class="btn
btn-info">

        </div>

    </form>

</div>

<p align="center" style="color: red">If your Admin and logging in
    for the first time, please use "admin" as username and "admin" as
    password(both without quotes)</p>

<br> <a href="index.jsp" style="color: black;">Back to User
    Home Page</a>

</div>

</body>

</html>

```

3. showAdminPage.jsp :

Below is the code from showAdminPage file of FlyAway.com App which is displayed after successful login from Admin account

Referring : adminPasswordChange.jsp, /viewAllPlaces, /viewAllAirlines, /viewAllFlightDetails, insertFlightDetails.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

```



```
<title>Admin Console</title>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<link href="style.css" rel="stylesheet" type="text/css">
</head>

<body>

    <%@include file="navigationBar.jsp"%>

    <div align="center" class="marginTop">

        <h2 class="indexPageNavBar">Welcome, Admin</h2>

    </div>

    <div align="center" class="indexPageNavBar" style="margin-top: 5%">

        <a href="adminPasswordChange.jsp" class="btn btn-info"><span
class="black">Change
        Password</span></a> <a href="viewAllPlaces" class="btn
btn-info"><span class="black">View All
        Places</span></a> <a href="viewAllAirlines" class="btn btn-info"><span
class="black">View All
        Airlines</span></a> <a href="viewAllFlightDetails" class="btn
btn-info"><span class="black">View
        Flight Details</span></a> <a href="insertFlightDetails.jsp" class="btn
btn-info"><span class="black">Insert
        New Flight Details</span></a>

    </div>

</body>

</html>
```

4. **adminPasswordChange.jsp** : Below is the code from adminPasswordChange file of FlyAway.com App which displays the form to change the password for Admin account

URL Mapping : /passwordChange

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Admin Password Change Console</title>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

</head>

<body>

    <%@include file="navigationBar.jsp" %>

    <div align="center" style="margin: 50px">

        <h2 class="indexPageNavBar">Please enter the new password for Admin
Account</h2>

        <form action="passwordChange" method="post">

            <div style="margin: 10px">

                <input type="hidden" name="admin" value="admin">

                <input type="text" placeholder="Enter New Password"
name="newPassword" style="margin: 10px">

                <br>

                <input type="submit" value="Change Password" class="btn
btn-warning btn-sm">

            </div>

        </form>

    </div>

</body>

</html>
```

```

        </div>

    </form>

</div>

<div align="center">

    <a href="showAdminPage.jsp" class="white">Back to Admin Page</a>

</div>

</body>

</html>

```

5. showAllPlaces.jsp :

Below is the code from showAllPlaces file of FlyAway.com App which displays all the Source and Destination

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Flight Details</title>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

</head>

<body>

    <%@include file="navigationBar.jsp" %>

    <div align="center" style="margin: 50px">

        <h2 class="indexPageNavBar">Source and Destination</h2>

        <table border="2" cellpadding="8" class="black">

```

```

        <tr>
            <td><strong>Source</strong></td>
            <td><strong>Destination</strong></td>
        </tr>
        <c:forEach var="flight" items="${flights}">
            <tr>
                <td><c:out value="${flight.getSource()}" /></td>
                <td><c:out value="${flight.getDestination()}" /></td>
            </tr>
        </c:forEach>
    </table>
</div>
<div align="center">
    <a href="showAdminPage.jsp" class="black">Back to Admin Page</a>
</div>
</body>
</html>

```

6. showAllFlightDetails.jsp :

Below is the code from showAllFlightDetails file of FlyAway.com App which displays all the Flight details like, Source, Destination, Flight name, Airline name and Price

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">

```

```

<title>Flight Details</title>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>

<body>

    <%@include file="navigationBar.jsp"%>

    <div align="center" style="margin: 50px">

        <table border="2" cellpadding="5">

            <caption style="caption-side: top;">

                <h2 class="indexPageNavBar">Flight Details</h2>

            </caption>

            <tr>

                <td><strong>Airline</strong></td>

                <td><strong>Flight Name</strong></td>

                <td><strong>Source</strong></td>

                <td><strong>Destination</strong></td>

                <td><strong>Price</strong></td>

                <c:if test="${isThisFromSearchServlet}">

                    <td><strong>Book</strong></td>

                </c:if>

            </tr>

            <c:forEach var="flight" items="${flights}">

                <tr>

                    <td><c:out value="${flight.getAirline()}" /></td>

                    <td><c:out value="${flight.getFlightName()}" /></td>

                    <td><c:out value="${flight.getSource()}" /></td>

```

```

<td><c:out value="\${flight.getDestination()}" /></td>
<td><c:out value="\${flight.getPrice()}" /></td>
<c:if test="\${isThisFromSearchServlet}">
    <td>
        <c:set var="selectedFlightName"
scope="session" value="\${flight.getFlightName()}" ></c:set>
        <c:set var="selectedFlightAirline"
scope="session" value="\${flight.getAirline()}" ></c:set>
        <c:set var="selectedFlightSource"
scope="session" value="\${flight.getSource()}" ></c:set>
        <c:set var="selectedFlightDestination"
scope="session" value="\${flight.getDestination()}" ></c:set>
        <c:set var="selectedFlightPrice"
scope="session" value="\${flight.getPrice()}" ></c:set>
        <a href="userRegistration.jsp"
class="white">Book Flight</a>
    </td>
</c:if>
</tr>
</c:forEach>
</table>
</div>
<c:choose>
    <c:when test="\${isThisFromSearchServlet}">
        <div align="center">
            <a href="userHomePage.jsp" class="black">Back to Search
Page</a>
        </div>

```

```

        </c:when>
        <c:otherwise>
            <div align="center">
                <a href="showAdminPage.jsp" class="black">Back to Admin
Page</a>
            </div>
        </c:otherwise>
    </c:choose>
</body>
</html>

```

7. showAllAirlines.jsp :

Below is the code from showAllAirlines file of FlyAway.com App which displays all the Airlines and associated Flight details

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Flight Details</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>
<body>
    <%@include file="navigationBar.jsp" %>
    <div align="center" style="margin-top: 50px">

```

```

<table border="2" cellpadding="5">
    <caption style="caption-side:top">
        <h3 style="white-space: nowrap;"
class="indexPageNavBar">Airlines and Flights</h3>
    </caption>
    <tr>
        <td><strong>Airline</strong></td>
        <td><strong>Flight Name</strong></td>
    </tr>
    <c:forEach var="flight" items="${flights}">
        <tr>
            <td><c:out value="${flight.getAirline()}" /></td>
            <td><c:out value="${flight.getFlightName()}" /></td>
        </tr>
    </c:forEach>
</table>
</div>
<div align="center" style="margin-top: 10px">
    <a href="showAdminPage.jsp" class="black">Back to Admin Page</a>
</div>
</body>
</html>

```

8. insertFlightDetails.jsp :

Below is the code from insertFlightDetails file of FlyAway.com App which takes new Flight details entered by Admin and stores it to DB

Referring : /flightDetails

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```



```
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Flight Management</title>
<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
</head>
<body>
    <%@include file="navigationBar.jsp" %>
    <div align="center">
        <h2 style="margin: 30px" class="indexPageNavBar">Enter New Flight Details
below</h2>

        <form action="flightDetails" method="post">
            <table border="1" cellpadding="3">
                <tr>
                    <th>Airline Name :</th>
                    <td><input type="text" name="airlines"></td>
                </tr>
                <tr>
                    <th>Flight Name :</th>
                    <td><input type="text" name="flightName"></td>
                </tr>
            </table>
        </form>
    </div>
</body>
</html>
```

```

        <tr>
            <th>Source :</th>
            <td><input type="text" name="source"></td>
        </tr>
        <tr>
            <th>Destination :</th>
            <td><input type="text" name="destination"></td>
        </tr>
        <tr>
            <th>Price :</th>
            <td><input type="number" name="price"></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
                <input type="submit" value="Save" class="btn
btn-warning btn-sm">
            </td>
        </tr>
    </table>
</form>
</div>
<div align="center" style="margin-top: 20px">
    <a href="showAdminPage.jsp" class="black">Back to Admin Page</a>
</div>
</body>
</html>

```

9. userHomePage.jsp :

Below is the code from userHomePage file of FlyAway.com App which displays Flights search option for User

Referring : /searchFlight

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>FlyAway - User</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
    <link href="userstyle.css" rel="stylesheet" type="text/css">
</head>
<body>
    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand indexPageNavBar"
href="index.jsp">FlyWay</a>
            </div>
        </div>
    </nav>
    <div align="center" class="marginTop">
        <div class="col-md-6 login-form-2">
```

```
<h3>Search Flight here</h3>
<form action="searchFlight" method="post">
    <div class="form-group">
        <input name="date" type="date" class="form-control"
            style="width: 300px" placeholder="Date of
Travel in YYYY-MM-DD *" />
    </div>
    <div class="form-group">
        <input name="source" type="text" class="form-control"
            style="width: 300px" placeholder="Source *" />
    </div>
    <div class="form-group">
        <input name="destination" type="text"
class="form-control"
            style="width: 300px" placeholder="Destination
*" />
    </div>
    <div class="form-group">
        <input name="numberOfPass" type="number"
class="form-control"
            style="width: 300px" placeholder="Number of
Travelers *" />
    </div>
    <div class="form-group">
        <input type="submit" class="btn btn-info"
value="Search" />
    </div>
</form>
```

```
</div>

<div align="center">

    <a href="index.jsp" class="black">Back to Home Page</a>

</div>

</div>

</body>

</html>
```

10. userRegistration.jsp :

Below is the code from userRegistration file of FlyAway.com App which displays User registration form where the user can enter their personal details

Referring : /bookingDetails

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>FlyWay - User</title>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

<link href="userstyle.css" rel="stylesheet" type="text/css">

</head>

<body>

    <nav class="navbar navbar-default">

        <div class="container-fluid">

            <div class="navbar-header">
```

```
                <a class="navbar-brand indexPageNavBar"
href="index.jsp">FlyWay</a>
            </div>
        </div>
    </nav>

    <div align="center" class="marginTop">
        <div class="col-md-6 login-form-2">
            <h3>User Registration</h3>
            <form action="bookingDetails" method="post">
                <div class="form-group">
                    <input name="firstname" type="text" class="form-control"
style="width: 400px" placeholder="First Name *"/>
                </div>
                <div class="form-group">
                    <input name="lastname" type="text" class="form-control"
style="width: 400px" placeholder="Last Name *"/>
                </div>
                <div class="form-group">
                    <input name="age" type="number" class="form-control"
style="width: 400px" placeholder="Age *"/>
                </div>
                <div class="form-group">
                    <input name="email" type="email" class="form-control"
style="width: 400px" placeholder="Your Email *"/>
                </div>
                <div class="form-group">
                    <input type="submit" class="btn btn-info" value="Register" />
                </div>
            </form>
        </div>
    </div>
</div>
```

```

        </form>

    </div>

    <a href="userRegistration.jsp" class="black">Back to User Home Page</a>

</div>

</body>

</html>

```

11. selectedFlightDetailsAndPayment.jsp :

Below is the code from selectedFlightDetailsAndPaymentfile of FlyAway.com App which displays User, Booked Flight and payment details

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>FlyAway - Booking</title>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

<link href="style.css" rel="stylesheet" type="text/css">

</head>

<body>

    <%@include file="navigationBar.jsp"%>

    <c:if test="${paymenDone == 'false'}">

        <c:set var="firstname" value="${passenger.getFirstname()}"

            scope="session"></c:set>

```

```

        <c:set var="lastname" value="${passenger.getLastname()}"
            scope="session"></c:set>

        <c:set var="age" value="${passenger.getAge()}" scope="session"></c:set>

        <c:set var="email" value="${passenger.getEmailId()}"
scope="session"></c:set>

    </c:if>

    <div align="center" style="margin: 50px">

        <c:if test="${paymenDone}">

            <h3 class="indexPageNavBar">Payment Successful, Bon voyage!!</h3>

        </c:if>

        <table border="2" cellpadding="5">

            <caption style="caption-side: top;">

                <h2 class="indexPageNavBar">Passenger Details</h2>

            </caption>

            <tr>

                <td><strong>Passenger Name</strong></td>

                <td><strong>Passenger Age</strong></td>

                <td><strong>Passenger Email</strong></td>

            </tr>

            <tr>

                <td><c:out
                    value="${passenger.getFirstname()}
${passenger.getLastname()}" /></td>

                <td><c:out value="${passenger.getAge()}" /></td>

                <td><c:out value="${passenger.getEmailId()}" /></td>

            </tr>

        </table>

```



```

<hr>
<table border="2" cellpadding="5">
    <caption style="caption-side: top;">
        <h2 class="indexPageNavBar">Selected Flight Details</h2>
    </caption>
    <tr>
        <td><strong>Airline</strong></td>
        <td><strong>Flight Name</strong></td>
        <td><strong>Source</strong></td>
        <td><strong>Destination</strong></td>
        <td><strong>Passengers</strong></td>
        <td><strong>Price</strong></td>
    </tr>
    <tr>
        <td><c:out value="\${selectedFlight.getAirline()}" /></td>
        <td><c:out value="\${selectedFlight.getFlightName()}" /></td>
        <td><c:out value="\${selectedFlight.getSource()}" /></td>
        <td><c:out value="\${selectedFlight.getDestination()}" /></td>
        <td><c:out value="\${totalPassengers}" /></td>
        <td><c:out value="\${totalPrice}" /></td>
    </tr>
</table>
<br>
<c:choose>
    <c:when test="\${paymenDone}">
        <h3>Total Amount Paid : \${totalPrice}</h3>

```

```

        </c:when>
        <c:otherwise>
            <a href="bookingDetails" class="btn btn-info black">Proceed
to Pay <c:out
                value="\${totalPrice}" /></a>
        </c:otherwise>
    </c:choose>
</div>
<div align="center">
    <a href="userHomePage.jsp" class="black">Back to Search Page</a>
</div>
</body>
</html>

```

12. navigationBar.jsp :

Below is the code from navigationBar of FlyAway.com App which is used to maintain common Navigation bar code

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
    <nav class="navbar navbar-default">
        <div class="container-fluid">

```

```

        <div class="navbar-header">
            <a class="navbar-brand indexPageNavBar"
href="index.jsp">FlyWay</a>
        </div>

        <ul class="nav navbar-nav">
            <li><a href="logout.jsp"
class="indexPageNavBar">Logout</a></li>
        </ul>
    </div>
</nav>
</body>
</html>

```

13. logout.jsp :

Below is the code from logout.jsp file of FlyAway.com App which is used logging out either Admin or User from current session

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Logging Out</title>
</head>
<body>
    <%
        session.invalidate();
        response.sendRedirect("index.jsp");
    %>

```

```
%>
</body>
</html>
```

14. style.css :

Below is the code from style.css file of FlyAway.com App which is used maintain styling properties

```
body {
    background-image:
url("https://akam.cdn.jdmagicbox.com/images/icontent/travel/flight.jpg");
}
```

```
.indexPageNavBar{
    color: white;
    font-family: cursive;
    font-weight: bold;
}
```

```
.marginTop{
    margin: 10% 10% 0 10%;
}
```

```
.black{
    color: black;
}
```

```
.white{
```

```
        color: white;
    }
}
```

15. userstyle.css :

Below is the code from user.css file of FlyAway.com App which is used maintain styling properties

```
body {
    background-image:
    url("https://s3-us-west-1.amazonaws.com/gaytravel-blog/plane-family-beach-teera-wanit-malikun.jpg");
}

.indexPageNavBar{
    color: black;
    font-family: cursive;
    font-weight: bold;
}

.marginTop{
    margin: 10% 10% 0 10%;
}

.black{
    color: black;
}

.white{ color: white;}
```

16. AdminLoginServlet.java :

Below is the code from AdminLoginServlet file of FlyAway.com App which handles all the requests pointing to **/login**

```
package com.assessment.controller;

import java.io.IOException;
```

```
import java.io.PrintWriter;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.assessment.model.Admin;
import com.assessment.util.HibernateUtil;

/**
 * Servlet implementation class DemoServlet
 */
@WebServlet("/login")
public class AdminLoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()

```

```

    */

    public AdminLoginServlet() {
        super();
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        System.out.println("Inside service method");
        SessionFactory factory = HibernateUtil.getSessionFactory();
        Session session = factory.openSession();
        Transaction transaction = session.beginTransaction();
        System.out.println("after Inside service method");
        String enteredUserName = request.getParameter("adminUsername");
        String enteredPassword = request.getParameter("adminPassword");
        @SuppressWarnings("unchecked")
        List<Admin> adminDb = session.createQuery("from Admin").list();
        System.out.println("admin DB : " + adminDb);
        RequestDispatcher adminLoginDispatcher = null;
        if (adminDb.isEmpty()) {
            Admin admin = new Admin(enteredUserName, enteredPassword);
            request.setAttribute("admin", adminDb);
            session.save(admin);
            transaction.commit();
            session.close();
        }
    }
}

```

```

        adminLoginDispatcher =
request.getRequestDispatcher("showAdminPage");

        PrintWriter printWriter = response.getWriter();

        printWriter.print("<html><body><h1>You have successfully logged
in</h1></body></html>");

    } else {

        Admin admin = adminDb.get(0);

        String currentAdminPassword = admin.getAdminPassword();

        if (currentAdminPassword.equals(enteredPassword)) {

            request.setAttribute("admin", adminDb);

            session.close();

            adminLoginDispatcher =
request.getRequestDispatcher("showAdminPage.jsp");

            adminLoginDispatcher.forward(request, response);

        } else {

            session.close();

            PrintWriter writer = response.getWriter();

            adminLoginDispatcher =
request.getRequestDispatcher("adminlogin.jsp");

            adminLoginDispatcher.include(request, response);

            writer.println("<center> <span style='color:red'> Invalid
Credentials!!, current Admin password is : " + currentAdminPassword +
"</span></center>");

        }

    }

}

}

}

```


17. AdminPasswordChangeServlet.java :

Below is the code from AdminPasswordChangeServlet file of FlyAway.com App which handles all the requests pointing to **/passwordChange**

```
package com.assessment.controller;

import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.assessment.model.Admin;
import com.assessment.util.HibernateUtil;

/**
 * Servlet implementation class DemoServlet
 */
@WebServlet("/passwordChange")
public class AdminPasswordChangeServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
}
```

```

public AdminPasswordChangeServlet() {
    super();
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    System.out.println("Inside doPost method of
AdminPasswordChangeServlet");
    SessionFactory factory = HibernateUtil.getSessionFactory();
    Session session = factory.openSession();
    Transaction transaction = session.beginTransaction();
    String adminName = request.getParameter("admin");
    String newPassword = request.getParameter("newPassword");
    Admin adminDataFromDb = session.get(Admin.class, 1);
    adminDataFromDb.setAdminPassword(newPassword);
    @SuppressWarnings("unchecked")
    List<Admin> adminDb = session.createQuery("from Admin").list();
    System.out.println("admin data from db : " + adminDb.get(0).toString());
    int adminId = adminDb.get(0).getId();
    Admin admin = new Admin(adminId, adminName, newPassword);
    System.out.println("New details of admin : " + admin.toString());
    session.update(adminDataFromDb);
    transaction.commit();
    request.getRequestDispatcher("logout.jsp");
}

```

```
        request.getRequestDispatcher("adminlogin.jsp").forward(request, response);
    }}
```

18. BookingDetailsServlet.java :

Below is the code from BookingDetailsServlet file of FlyAway.com App which handles all the requests pointing to **/bookingDetails**

```
package com.assessment.controller;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.assessment.model.Flight;
import com.assessment.model.User;
import com.assessment.util.HibernateUtil;

@WebServlet("/bookingDetails")
public class BookingDetailsServlet extends HttpServlet {
```

```
private static final long serialVersionUID = 1L;
```

```
@Override
```

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {

    PrintWriter writer = resp.getWriter();

    RequestDispatcher requestDispatcher = null;

    SessionFactory sessionFactory = HibernateUtil.getSessionFactory();

    Session session = sessionFactory.openSession();

    Transaction transaction = session.beginTransaction();

    String flightName =
req.getSession().getAttribute("selectedFlightName").toString();

    String airline =
req.getSession().getAttribute("selectedFlightAirline").toString();

    int price =
Integer.parseInt(req.getSession().getAttribute("selectedFlightPrice").toString());

    String source =
req.getSession().getAttribute("selectedFlightSource").toString();

    String destination =
req.getSession().getAttribute("selectedFlightDestination").toString();

    int totalPassengers =
Integer.parseInt(req.getSession().getAttribute("numberOfPass").toString());

    int totalPrice = totalPassengers * price;

    Flight selectedFlight = new Flight(flightName, source, destination, airline,
price);

    System.out.println("selectedFlight : " + selectedFlight.toString());

    String firstname = req.getParameter("firstname");
```

```
String lastname = req.getParameter("lastname");

int age = Integer.parseInt(req.getParameter("age").isEmpty() ? "0" :
req.getParameter("age"));

String email = req.getParameter("email");

if (firstname.isEmpty() || lastname.isEmpty() || email.isEmpty() || age == 0)
{

    writer = resp.getWriter();

    requestdispatcher =
req.getRequestDispatcher("userRegistration.jsp");

    requestdispatcher.include(req, resp);

    writer.println(

        "<center> <span style='color:green; font-family: cursive;
font-weight: bold;'>Enter Valid Details. All fields are Mandatory</span></center>");

    } else {

        User user = new User(firstname, lastname, age, email);

        session.save(user);

        transaction.commit();

        req.setAttribute("paymenDone", "false");

        req.setAttribute("totalPrice", totalPrice);

        req.setAttribute("totalPassengers", totalPassengers);

        req.setAttribute("passenger", user);

        req.setAttribute("selectedFlight", selectedFlight);

req.getRequestDispatcher("selectedFlightDetailsAndPayment.jsp").forward(req,
resp);

    }

}
```

@Override

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {

    req.setAttribute("paymenDone", true);

    String flightName =
req.getSession().getAttribute("selectedFlightName").toString();

    String airline =
req.getSession().getAttribute("selectedFlightAirline").toString();

    int price =
Integer.parseInt(req.getSession().getAttribute("selectedFlightPrice").toString());

    String source =
req.getSession().getAttribute("selectedFlightSource").toString();

    String destination =
req.getSession().getAttribute("selectedFlightDestination").toString();

    int totalPassengers =
Integer.parseInt(req.getSession().getAttribute("numberOfPass").toString());

    int totalPrice = totalPassengers * price;

    Flight selectedFlight = new Flight(flightName, source, destination, airline,
price);

    System.out.println("selectedFlight : " + selectedFlight.toString());

    String firstname = (String) req.getSession().getAttribute("firstname");

    System.out.println("firstname : " + firstname);

    String lastname = (String) req.getSession().getAttribute("lastname");

    int age = Integer.parseInt(req.getSession().getAttribute("age").toString());

    String email = (String) req.getSession().getAttribute("email");

    User user = new User(firstname, lastname, age, email);

    System.out.println("user inside do get of Booking ; " + user);

    req.setAttribute("totalPrice", totalPrice);
```

```
        req.setAttribute("totalPassengers", totalPassengers);
        req.setAttribute("passenger", user);
        req.setAttribute("selectedFlight", selectedFlight);

    req.getRequestDispatcher("selectedFlightDetailsAndPayment.jsp").forward(req,
    resp);
    }
}
```

19. FlightServlet.java :

Below is the code from FlightServlet file of FlyAway.com App which handles all the requests pointing to **/flightDetails**

```
package com.assessment.controller;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
```

```

import org.hibernate.Transaction;

import com.assessment.model.Airline;
import com.assessment.model.Flight;
import com.assessment.util.HibernateUtil;

@WebServlet("/flightDetails")
public class FlightServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        PrintWriter writer = resp.getWriter();

        RequestDispatcher requestdispatcher = null;

        SessionFactory factory = HibernateUtil.getSessionFactory();
        Session session = factory.openSession();
        Transaction transaction = session.beginTransaction();

        String flightName = req.getParameter("flightName");
        String source = req.getParameter("source");
        String destination = req.getParameter("destination");
        String airline = req.getParameter("airlines");

        int price = Integer.parseInt(req.getParameter("price").isEmpty() ? "0" :
req.getParameter("price"));

        if (flightName.isEmpty() || source.isEmpty() || destination.isEmpty() ||
airline.isEmpty() || price == 0) {

```



```

        writer = resp.getWriter();

        requestdispatcher =
req.getRequestDispatcher("insertFlightDetails.jsp");

        requestdispatcher.include(req, resp);

        writer.println(

                "<center> <span style='color:green; font-family: cursive;
font-weight: bold;'>Enter Valid Flight Details</span></center>");

    }else {

        Airline airlineCreated = new Airline(airline);

        System.out.println("airlineCreated : " + airlineCreated.toString());

        session.save(airlineCreated);

        Flight flightCreated = new Flight(flightName, source, destination,
airlineCreated.getAirlineName(), price);

        System.out.println("flightCreated : " + flightCreated.toString());

        session.save(flightCreated);

        transaction.commit();

        List<Flight> flights = session.createQuery("from Flight").list();

        req.setAttribute("flights", flights);

        req.getRequestDispatcher("showAllFlightDetails.jsp").forward(req,
resp);

    }

}

}

```

20. SearchFlightServlet.java :

Below is the code from SearchFlightServlet file of FlyAway.com App which handles all the requests pointing to **/searchFlight**

```

package com.assessment.controller;

import java.io.IOException;

```

```
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import com.assessment.model.Flight;
import com.assessment.util.HibernateUtil;
@WebServlet("/searchFlight")
public class SearchFlightServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        PrintWriter writer = resp.getWriter();

        RequestDispatcher requestispatcher = null;

        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();

        Session session = sessionFactory.openSession();

        Transaction transaction = session.beginTransaction();
```

```

String date = req.getParameter("date");
System.out.println("date : " + date);
String source = req.getParameter("source");
String destination = req.getParameter("destination");
int numberOfPass =
Integer.parseInt(req.getParameter("numberOfPass").isEmpty() ? "0" :
req.getParameter("numberOfPass"));

if (source == null || destination == null || numberOfPass == 0 ||
date.isEmpty() || source.isEmpty() || destination.isEmpty()) {

    writer = resp.getWriter();
    requestdispatcher = req.getRequestDispatcher("userHomePage.jsp");
    requestdispatcher.include(req, resp);
    writer.println(
        "<center> <span style='color:green; font-family: cursive;
font-weight: bold;'>Enter Valid Details</span></center>");
} else {
    @SuppressWarnings("unchecked")
    List<Flight> flights = session.createQuery("from Flight").list();
    List<Flight> availableFlights = new ArrayList<Flight>();
    transaction.commit();
    for (Flight flight : flights) {
        if (flight.getSource().equalsIgnoreCase(source)
            &&
flight.getDestination().equalsIgnoreCase(destination)) {
            availableFlights.add(flight);
        }
    }
}

```

```

        if (availableFlights.size() > 0) {
            req.getSession().setAttribute("numberOfPass",
numberOfPass);

            req.setAttribute("isThisFromSearchServlet", true);
            req.setAttribute("flights", availableFlights);

req.getRequestDispatcher("showAllFlightDetails.jsp").forward(req, resp);
        } else {
            writer = resp.getWriter();
            requestispatcher =
req.getRequestDispatcher("userHomePage.jsp");
            requestispatcher.include(req, resp);
            writer.println(

                "<center> <span style='color:red'>No Flight
available for the Entered Soure and Destination</span></center>");
        }
    }
}
}
}

```

21. ViewAllAirlines.java :

Below is the code from ViewAllAirlines file of FlyAway.com App which handles all the requests pointing to **/viewAllAirlines**

```
package com.assessment.controller;
```

```
import java.io.IOException;
```

```
import java.util.List;
```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
```

```
import com.assessment.model.Flight;
import com.assessment.util.HibernateUtil;
```

```
@WebServlet("/viewAllAirlines")
```

```
public class ViewAllAirlines extends HttpServlet{
```

```
    private static final long serialVersionUID = 1L;
```

```
    @Override
```

```
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
```

```
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
```

```
        Session session = sessionFactory.openSession();
```

```
        Transaction transaction = session.beginTransaction();
```

```
        @SuppressWarnings("unchecked")
```

```
        List<Flight> flights = session.createQuery("from Flight").list();
```

```
        transaction.commit();

        req.setAttribute("flights", flights);

        req.getRequestDispatcher("showAllAirlines.jsp").forward(req, resp);

    }

}
```

22. ViewAllFlightDetails.java :

Below is the code from ViewAllFlightDetails file of FlyAway.com App which handles all the requests pointing to **/viewAllFlightDetails**

```
package com.assessment.controller;

import java.io.IOException;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.assessment.model.Flight;
import com.assessment.util.HibernateUtil;
```

```

@WebServlet("/viewAllFlightDetails")

public class ViewAllFlightDetails extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {

        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();

        Session session = sessionFactory.openSession();

        Transaction transaction = session.beginTransaction();

        @SuppressWarnings("unchecked")

        List<Flight> flights = session.createQuery("from Flight").list();

        transaction.commit();

        req.setAttribute("flights", flights);

        req.getRequestDispatcher("showAllFlightDetails.jsp").forward(req, resp);

    }

}

```

23. ViewAllPlaces.java :

Below is the code from ViewAllPlaces file of FlyAway.com App which handles all the requests pointing to **/viewAllPlaces**

```

package com.assessment.controller;

import java.io.IOException;

import java.util.List;

```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
```

```
import com.assessment.model.Flight;
import com.assessment.util.HibernateUtil;
```

```
@WebServlet("/viewAllPlaces")
```

```
public class ViewAllPlaces extends HttpServlet{
```

```
    private static final long serialVersionUID = 1L;
```

```
    @Override
```

```
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
```

```
        SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
```

```
        Session session = sessionFactory.openSession();
```

```
        Transaction transaction = session.beginTransaction();
```

```
        @SuppressWarnings("unchecked")
```



```

        List<Flight> flights = session.createQuery("from Flight").list();
        transaction.commit();
        req.setAttribute("flights", flights);
        req.getRequestDispatcher("showAllPlaces.jsp").forward(req, resp);
    }
}

```

24. Admin.java :

Below is the code from Admin.java file of FlyAway.com App which is an Entity Class for Admin

Refers to : admin table

```

package com.assessment.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "admin")
public class Admin {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")

```

```
private int id;
```

```
@Column(name = "username")
```

```
private String adminUserName;
```

```
@Column(name = "password")
```

```
private String adminPassword;
```

```
public Admin() {
```

```
    super();
```

```
}
```

```
public Admin(String adminUserName, String adminPassword) {
```

```
    this.adminUserName = adminUserName;
```

```
    this.adminPassword = adminPassword;
```

```
}
```

```
public Admin(int id, String adminUserName, String adminPassword) {
```

```
    super();
```

```
    this.id = id;
```

```
    this.adminUserName = adminUserName;
```

```
    this.adminPassword = adminPassword;
```

```
}
```

```
public String getAdminUserName() {
```

```
    return adminUserName;
```

```

    }

    public void setAdminUserName(String adminUserName) {
        this.adminUserName = adminUserName;
    }

    public String getAdminPassword() {
        return adminPassword;
    }

    public void setAdminPassword(String adminPassword) {
        this.adminPassword = adminPassword;
    }

    public int getId() {
        return id;
    }

    @Override
    public String toString() {
        return "Admin [id=" + id + ", adminUserName=" + adminUserName + ",
adminPassword=" + adminPassword + "]";
    }

}

```

25. Airline.java :

Below is the code from Airline.java file of FlyAway.com App which is an Entity Class for Airline

Refers to : airlines table

```
package com.assessment.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "airlines")
public class Airline {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "airline_name")
    private String airlineName;

    public Airline() {
    }
}
```

```
public Airline(String airlineName) {  
    super();  
    this.airlineName = airlineName;  
}
```

```
public Airline(int id, String airlineName) {  
    super();  
    this.id = id;  
    this.airlineName = airlineName;  
}
```

```
public String getAirlineName() {  
    return airlineName;  
}
```

```
public void setAirlineName(String airlineName) {  
    this.airlineName = airlineName;  
}
```

```
public int getId() {  
    return id;  
}
```

```
@Override
```

```
public String toString() {  
    return "Airline [id=" + id + ", airlineName=" + airlineName + "];"
```

```
}  
}
```

26. Flight.java :

Below is the code from Flight.java file of FlyAway.com App which is an Entity Class for Flights

Refers to : flightdetails table

```
package com.assessment.model;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name = "flightdetails")
```

```
public class Flight {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    @Column(name = "id")
```

```
    private int id;
```

```
    @Column(name = "flight_name")
```

```
    private String flightName;
```

```
@Column(name = "source")
```

```
private String source;
```

```
@Column(name = "destination")
```

```
private String destination;
```

```
@Column(name = "airline_name")
```

```
private String airline;
```

```
private int price;
```

```
public Flight() {
```

```
}
```

```
public Flight(String flightName, String source, String destination, String airline,  
int price) {
```

```
    super();
```

```
    this.flightName = flightName;
```

```
    this.source = source;
```

```
    this.destination = destination;
```

```
    this.airline = airline;
```

```
    this.price = price;
```

```
}
```

```
public Flight(int id, String flightName, String source, String destination, String  
airline, int price) {
```

```
    super();
```

```
        this.id = id;
        this.flightName = flightName;
        this.source = source;
        this.destination = destination;
        this.airline = airline;
        this.price = price;
    }
```

```
    public String getFlightName() {
        return flightName;
    }
```

```
    public void setFlightName(String flightName) {
        this.flightName = flightName;
    }
```

```
    public String getSource() {
        return source;
    }
```

```
    public void setSource(String source) {
        this.source = source;
    }
```

```
    public String getDestination() {
        return destination;
    }
```



```
}
```

```
public void setDestination(String destination) {
```

```
    this.destination = destination;
```

```
}
```

```
public String getAirline() {
```

```
    return airline;
```

```
}
```

```
public void setAirline(String airline) {
```

```
    this.airline = airline;
```

```
}
```

```
public int getPrice() {
```

```
    return price;
```

```
}
```

```
public void setPrice(int price) {
```

```
    this.price = price;
```

```
}
```

```
public int getId() {
```

```
    return id;
```

```
}
```

```

@Override

public String toString() {

    return "Flight [id=" + id + ", flightName=" + flightName + ", source=" +
source + ", destination=" + destination

        + ", airline=" + airline + ", price=" + price + "];"

    }

}

```

27. User.java :

Below is the code from User.java file of FlyAway.com App which is an Entity Class for Users

Refers to : users table

```

package com.assessment.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "users")
public class User {

    @Id

```

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
@Column(name = "id")
```

```
private int id;
```

```
@Column(name = "firstname")
```

```
private String firstname;
```

```
@Column(name = "lastname")
```

```
private String lastname;
```

```
@Column(name = "age")
```

```
private int age;
```

```
@Column(name = "email")
```

```
private String emailId;
```

```
public User() {
```

```
}
```

```
public User(String firstname, String lastname, int age, String emailId) {
```

```
    super();
```

```
    this.firstname = firstname;
```

```
    this.lastname = lastname;
```

```
    this.age = age;
```

```
    this.emailId = emailId;
```

```
}
```

```
public User(int id, String firstname, String lastname, int age, String emailId) {  
    super();  
    this.id = id;  
    this.firstname = firstname;  
    this.lastname = lastname;  
    this.age = age;  
    this.emailId = emailId;  
}
```

```
public String getFirstname() {  
    return firstname;  
}
```

```
public void setFirstname(String firstname) {  
    this.firstname = firstname;  
}
```

```
public String getLastname() {  
    return lastname;  
}
```

```
public void setLastname(String lastname) {  
    this.lastname = lastname;  
}
```

```

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getEmailId() {
        return emailId;
    }

    public void setEmailId(String emailId) {
        this.emailId = emailId;
    }

    public int getId() {
        return id;
    }
}

```

28. HibernateUtil.java :

Below is the code from HibernateUtil.java file of FlyAway.com App which is used to create Hibernate SessionFactory

Refers to : hibernate.cfg.xml

```
package com.assessment.util;
```

```
import org.hibernate.SessionFactory;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

public class HibernateUtil {

    private static final SessionFactory sessionFactory;

    static {
        try {
            StandardServiceRegistry standard = new
StandardServiceRegistryBuilder()
                .configure("hibernate.cfg.xml").build();

            Metadata metaData = new
MetadataSources(standard).getMetadataBuilder().build();
            sessionFactory = metaData.getSessionFactoryBuilder().build();
        } catch (Throwable e) {
            // TODO: handle exception
            throw new ExceptionInInitializerError(e);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

```
}
```

































29. hibernate.cfg.xml :

Below is the code from hibernate.cfg.xml file of FlyAway.com App which maintains all DB Connection properties and Mapping classes

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-5.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/course2project?auto
Reconnect=true&useSSL=false</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">admin</property>
        <property name="hibernate.connection.pool_size">10</property>
        <property name="show_sql">>true</property>
        <property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>
        <property
name="hibernate.current_session_context_class">thread</property>
        <!-- <property name="hibernate.hbm2ddl.auto">create</property> -->
        <mapping class="com.assessment.model.Admin"></mapping>
        <mapping class="com.assessment.model.Airline"></mapping>
        <mapping class="com.assessment.model.Flight"></mapping>
        <mapping class="com.assessment.model.User"></mapping>
    </session-factory>
```

</hibernate-configuration>

Project Structure

- ▼  AdminPortal
 - >  Deployment Descriptor: AdminPortal
 - ▼  Java Resources
 - ▼  src
 - >  com.assessment.controller
 - >  com.assessment.model
 - >  com.assessment.util
 -  hibernate.cfg.xml
 - >  Libraries
 - >  JavaScript Resources
 - >  build
 - >  target
 - ▼  WebContent
 - >  META-INF
 - >  WEB-INF
 -  adminlogin.jsp
 -  adminPasswordChange.jsp
 -  index.jsp
 -  insertFlightDetails.jsp
 -  logout.jsp
 -  navigationBar.jsp
 -  selectedFlightDetailsAndPayment.jsp
 -  showAdminPage.jsp
 -  showAllAirlines.jsp
 -  showAllFlightDetails.jsp
 -  showAllPlaces.jsp
 -  style.css
 -  TableCreationScripts.sql
 -  user.jsp
 -  userHomePage.jsp
 -  userRegistration.jsp
 -  userstyle.css

Unique Selling Points

1. Same home page for either Admin or User
2. Admin can add as many as Flight details
3. Admin can change password whenever they wishes
4. Admin can view all Source and Destinations
5. Admin can view all Airlines and associated flights with it
6. Admin can view all the Flight details along with price
7. User can search for any flights, if available it will show, if not, it will again ask for search instead of throwing exception
8. User can book flights ticket by registering themselves
9. Simple payment option

Source Code

Source code of the FlyAway.com app is hosted or maintained in GitHub. Below is the link to the GitHub repository

FlyAway.com app GitHub Repo link : https://github.com/bsangamesh/AdminPortal_FlyAway