

## Phase 2 - Assessment

**Project :FlyAway (An Airline Booking Portal).**

**Developer :Adarsh M L**

**<https://github.com/Adarshml/Practice-Projects/tree/main/FlyAway2>**

### Source Code

#### 1. Index.jsp

```
<html>
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.button {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    margin: 4px 2px;
    cursor: pointer;
    transition-duration: 0.4s;
    border-radius: 8px;
}

.myname {
    text-align: right;
    padding: 15px 32px;
    color: white;
}
</style>
<title>FlyAway.com</title>
<body>
    <br>
```

```

        <br>
        <h1 style="text-align: center; font-size: 40px; color:
1D2F5A;">FlyAway</h1>
        <h2 style="text-align: center;color: 1C2F5E;">Fly anywhere. Fly
everywhere.</h2>
        <p align="center">
        <a class="button" href="adminlogin.html">Admin Login</a>
        <a class="button" href="userlogin.html">User Login</a>
        <a class="button" href="userregister.html">New User ?</a>
        </p>
        <h3 style="text-align: center;">Airline booking system.</h3>
        <br><br><br><br><br><br><br><br>
        <br><br><br><br><br><br><br>

        <h3 class="myname">Developed by Adarsh M L</h3>

</body>
</body>

```

## 2. userlogin.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}
.field {
    background-color: #1D2F5A;
    display: block;
    margin-left: 600px;
    margin-right: 600px;
    padding-top: 0.35em;
    padding-bottom: 0.625em;
    padding-left: 0.75em;
    padding-right: 0.75em;
    border: 2px groove(internal value);
}
</style>
<title>Login as User</title>
</head>

```

```

<body>
<br><br>
    <h2 align="center">Welcome to FlyAway, Please Login</h2>
    <br>
    <fieldset class="field">
        <h3 style='color: white; text-align: center;'>USER LOGIN</h3>
        <br>
        <form action="loginuser" method="post">
            <table>
                <tr>
                    <td style='color: white;' >User Name</td>
                    <td><input name="name" id="name"
type="text"></td>
                </tr>
                <tr>
                    <td style='color: white;'>Password</td>
                    <td><input name="password" id="name"
type="password"></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input style='color: 1D2F5A;
text-align:center;' type="submit" value="Login"></td>
                </tr>
            </table>
        </form>
    </fieldset>
</body>
</html>

```

### 3. userregister.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.field {
    background-color: #1D2F5A;

```

```

        display: block;
        margin-left: 600px;
        margin-right: 600px;
        padding-top: 0.35em;
        padding-bottom: 0.625em;
        padding-left: 0.75em;
        padding-right: 0.75em;
        border: 2px groove(internal value);
    }
</style>
<title>Register as new User</title>
</head>
<body>
    <br>
    <br>
    <fieldset class="field">
        <h3 style='color: white; text-align: center;'>USER
REGISTRATION</h3>
        <br>
        <form action="registeruser" method="post">
            <table>
                <tr>
                    <td style='color: white;' >User Name</td>
                    <td><input name="name" id="name"
type="text"></td>
                </tr>
                <tr>
                    <td style='color: white;'>Password</td>
                    <td><input name="password" id="name"
type="password"></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input style='color: 1D2F5A;
text-align:center;' type="submit" value="Register Now"></td>
                </tr>
            </table>
        </form>
    </fieldset>
</body>
</html>

```

#### 4. Adminlogin.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.field {
    background-color: #1D2F5A;
    display: block;
    margin-left: 600px;
    margin-right: 600px;
    padding-top: 0.35em;
    padding-bottom: 0.625em;
    padding-left: 0.75em;
    padding-right: 0.75em;
    border: 2px groove(internal value);
}
</style>
<title>Admin Login</title>
</head>
<body>
    <br>
    <br>
    <br>
    <br>
    <fieldset class="field">
        <h3 style='color: white; text-align:center;'>ADMIN PANEL</h3>
        <br>
        <form action="loginadmin" method="post">
            <table>
                <tr>
                    <td style='color: white;'>Admin
Name</td>
                    <td><input name="name" id="name"
type="text"></td>
                </tr>
                <tr>
```

```

white;'}>Password</td>
                                <td style='color:
                                <td><input name="password"
                                id="name" type="password"></td>
                                </tr>
                                <tr>
                                <td></td>
                                <td><input style='color: 1D2F5A;
                                text-align:center;' type="submit" value="Login"></td>
                                </tr>
                                </table>
                                </form>
                                </fieldset>
</body>
</html>

```

## 5. addflights.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.field {
    background-color: #1D2F5A;
    display: block;
    margin-left: 500px;
    margin-right: 500px;
    padding-top: 0.35em;
    padding-bottom: 0.625em;
    padding-left: 0.75em;
    padding-right: 0.75em;
    border: 2px groove(internal value);
}
</style>
<title>Add Flights</title>
</head>
<body>

```

```

<br>
<fieldset class="field">
    <h3 style='color: white; text-align: center;'>ENTER FLIGHT
    DETAILS</h3>
    <br>
    <form method="post" action="addflightdata">
        <table>
            <tr>
                <td style='color: white;'>Name of
                Airline</td>
                <td><input type="text" name="airline"
                id="airline"></td>
            </tr>
            <tr>
                <td style='color: white;'>Source</td>
                <td><input type="text" name="source"
                id="source"></td>
            </tr>
            <tr>
                <td style='color: white;'>Destination</td>
                <td><input type="text" name="destination"
                id="destination"></td>
            </tr>
            <tr>
                <td style='color: white;'>Ticket Price</td>
                <td><input type="text" name="price"
                id="price"></td>
            </tr>
            <tr>
                <td style='color: white;'>Enter Available
                Days (Eg: Sunday)</td>
                <td><input type="text"
                name="availabledays" id="availabledays"></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" value="Add
                Flight"></td>
            </tr>
        </table>
    </form>
</fieldset>

</body>

```

</html>

## 6. changepassword.html

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.field {
    background-color: #1D2F5A;
    display: block;
    margin-left: 600px;
    margin-right: 600px;
    padding-top: 0.35em;
    padding-bottom: 0.625em;
    padding-left: 0.75em;
    padding-right: 0.75em;
    border: 2px groove(internal value);
}
</style>
<meta charset="UTF-8">
<title>Changed Password</title>
</head>
<body>
    <br>
    <fieldset class="field">
        <h3 style='color: white; text-align: center;'>Change
Password</h3>
        <br>
        <form action="changePassword" method="post">
            <table>
                <tr>
                    <td style='color: white;'>New Password</td>
                    <td><input type="password"
name="password"></td>
                </tr>
            </table>
        </form>
    </fieldset>
</body>
</html>
```



```
 </td>  <input type="submit"></td> </tr> </table> </form> </fieldset> </body> </html> | |
```

## 7. searchform.html

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.field {
    background-color: #1D2F5A;
    display: block;
    margin-left: 500px;
    margin-right: 500px;
    padding-top: 0.35em;
    padding-bottom: 0.625em;
    padding-left: 0.75em;
    padding-right: 0.75em;
    border: 2px groove(internal value);
}
</style>
<meta charset="ISO-8859-1">
<title>Customer Details</title>
</head>
<body>
    <br>
    <fieldset class="field">
        <h3 style='color: white; text-align: center;'>ENTER YOUR
TRAVEL
        DETAILS</h3>
        <form action="availableflights"
            style="font-family: Arial; font-size: 21px;">
            <table>

```

```

        <tr>
            <td style='color: white;'>Your Name</td>
            <td><input type="text" name="username"
required /></td>
        </tr>
        <tr>
            <td style='color: white;'>Date of Travel</td>
            <td><input type="text" name="dateoftravel"
required /></td>
        </tr>
        <tr>
            <td style='color: white;'>Current City</td>
            <td><input type="text" name="source"
required /></td>
        </tr>
        <tr>
            <td style='color: white;'>Destination
City</td>
            <td><input type="text" name="destination"
required /></td>
        </tr>
        <tr>
            <td style='color: white;'>Number of
Travellers<sub><sub>(1
- 10)</sub></sub> :
            <td><input type="number"
name="numberofpersons" required /></td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Proceed
Ahead"></td>
        </tr>
    </table>

</form>
</fieldset>
</body>
</html>

```

## 8. bookflight.html

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.field {
    background-color: #1D2F5A;
    display: block;
    margin-left: 550px;
    margin-right: 550px;
    padding-top: 0.35em;
    padding-bottom: 0.625em;
    padding-left: 0.75em;
    padding-right: 0.75em;
    border: 2px groove(internal value);
}

.tab {
    border: 2px solid black;
    text-align: center;
    padding: 10px;
}
</style>
<meta charset="UTF-8">
<title>Book a Flight</title>
</head>
<body>
    <br>
    <fieldset class="field">
        <h3 style='color: white; text-align: center;'>FLIGHT ID</h3>
        <br>
        <form action="userbooking" method="post">
            <table>
                <tr>
                    <td style='color: white;'>Enter Flight ID to
book</td>
                    <td><input type="number"
name="flightnumber"></td>
                </tr>
                <tr>
                    <td></td>

```

```

Flight"></td>
                                <td><input type="submit" value="Book
                                Flight"></td>
                                </tr>
                                </table>
                                </form>
                                </fieldset>
</body>
</html>

```

## 9. payment.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.field {
    background-color: #1D2F5A;
    display: block;
    margin-left: 600px;
    margin-right: 600px;
    padding-top: 0.35em;
    padding-bottom: 0.625em;
    padding-left: 0.75em;
    padding-right: 0.75em;
    border: 2px groove(internal value);
}
</style>
<title>Make Payment</title>
</head>
<body>
    <br>
    <br>
    <h2 align="center">ENTER YOUR PAYMENT DETAILS</h2>

    <fieldset class="field">
        <h3 style='color: white; text-align: center;'>BANK DETAILS</h3>
        <br>
        <form action="payment">

```

```

        <table>
            <tr>
                <td style='color: white;'>Bank Name</td>
                <td><input type="text" name="bankname"
required></td>
            </tr>
            <tr>
                <td style='color: white;'>Card Number</td>
                <td><input type="text" name="cardnumber"
required></td>
            </tr>
            <tr>
                <td style='color: white;'>Enter
password</td>
                <td><input type="password"
name="paymentpassword" required></td>
            </tr>
            <tr>
                <td></td>
                <td><input style='color: 1D2F5A; text-align:
center;'
                    type="submit"
value="Continue"></td>
            </tr>
        </table>
    </form>
</fieldset>

</body>
</html>

```

## 10. optionsadmin.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
body {
    background-image: url("https://wallpaperaccess.com/full/254361.jpg");
    background-size: cover;
}

.button {

```

```

        background-color:#22335C;
        border: none;
        color: white;
        padding: 15px 32px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
        margin: 4px 2px;
        cursor: pointer;
        transition-duration: 0.4s;
        border-radius: 8px;
    }
</style>
<title>Admin</title>
</head>
<body>
    <br>
    <h1 align = "center">ADMIN DASHBOARD</h1>
    <br>
    <p align="center">
        <a class="button" href="changePassword.html">Change
Password</a>
        <a class="button" href="addFlights.html">Add Flights</a>
        <a class="button" href="viewplaces">View Sources and
Destinations</a>
        <a class="button" href="viewairlines">View Airlines</a>
        <a class="button" href="logoutadmin">Logout</a>
    </p>
</body>
</html>

```

## **AdminLogin**

### **1. Addflights.java**

```

package adminlogin;

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;

import java.io.IOException;

```

```

import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

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

import entityclasses.*;
import util.HibernateSessionUtil;

@WebServlet("/addflightdata")
public class addflights extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        HttpSession s=request.getSession(false);
        if(s!=null) {
            // fetch data from form
            String source = request.getParameter("source");
            String destination =
request.getParameter("destination");
            String airline=request.getParameter("airline");
            double
price=Double.parseDouble(request.getParameter("price"));
            String
availabledays=request.getParameter("availabledays");
            try {
                // 1. build hibernate session factory
                SessionFactory factory =
HibernateSessionUtil.buildSessionFactory();

                // 2. create session object

```

```

        Session session = factory.openSession();
        // 3. create a object with 5000 as default
        balance of wallet
        flight obj = new
        flight(source,destination,airline,price,availabledays);

        // 4. begin transaction
        Transaction tx =
        session.beginTransaction();

        // 5. save product
        session.save(obj);

        // 6. commit transaction
        tx.commit();

        if(session != null ) {

        request.getRequestDispatcher("optionsadmin.html").include(request,
        response);
                                out.print("<h3 style='color:green'>
        Flight added sucessfully </h3>");
                                }

        // 3. close session
        session.close();

        }
        catch (Exception e) {
            out.print("<h3 style='color:red'>Addition
        failed ! </h3>");
        }
        }
        else
            out.print("<h2>Problem with session, please
        re-login</h2>");
        }

    }

```

## 2. viewairlines.java

```

package adminlogin;
import java.io.IOException;

```



```

import java.io.PrintWriter;
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 entityclasses.*;
import util.HibernateSessionUtil;

@WebServlet("/viewairlines")
public class viewairlines extends HttpServlet{

    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        request.getRequestDispatcher("optionsadmin.html").include(request,
            response);

        try {
            // 1. build hibernate session factory
            SessionFactory factory =
                HibernateSessionUtil.buildSessionFactory();

            // 2. create session object
            Session session = factory.openSession();

            // 3. read products
            List<flight> obj = session.createQuery("from
                flight").list();

            //show data as table.
            out.print("<h3>Registered Airlines </h3>");

```

```

        out.print("<style> table,td,th {"
                    + "border:2px solid
black;text-align:center;"
                    + "padding: 10px; "
                    + "></style>");
        out.print("<table >");
        out.print("<tr>");
            out.print("<th> ID</th>");
            out.print("<th> Airline Name</th>");
        out.print("</tr>");

        for(flight p : obj) {
            out.print("<tr>");
            out.print("<td>" + p.getId() + "</td>");
            out.print("<td>" + p.getAirline() + "</td>");
            out.print("</tr>");
        }
        out.print("</table>");
        // 3. close session
        session.close();
    } catch (Exception e) {
        out.print("<h3 style='color:red'> Hibernate session
is failed ! " + e + "</h3>");
    }
}
}

```

### 3. viewplaces.java

```

package adminlogin;
import java.io.IOException;
import java.io.PrintWriter;
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 entityclasses.*;
import util.HibernateSessionUtil;

@WebServlet("/viewplaces")
public class viewplaces extends HttpServlet{

    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        request.getRequestDispatcher("optionsadmin.html").include(request,
            response);

        try {
            // 1. build hibernate session factory
            SessionFactory factory =
                HibernateSessionUtil.buildSessionFactory();

            // 2. create session object
            Session session = factory.openSession();

            // 3. read products
            List<flight> obj = session.createQuery("from
flight").list();

            //show data as table.
            out.print("<h3> Registered Sources and
Destinations</h3>");

            out.print("<style> table,td,th {"
                + "border:2px solid black;"
                + "padding: 10px; "
                + "}</style>");
            out.print("<table>");
            out.print("<tr>");
                out.print("<th> ID</th>");
                out.print("<th> Source Place</th>");
                out.print("<th> Destination Place</th>");
            out.print("</tr>");

```

```

        for(flight p : obj) {
            out.print("<tr>");
            out.print("<td>" + p.getId() + "</td>");
            out.print("<td>" + p.getSource() + "</td>");
            out.print("<td>" + p.getDestination() + "</td>");
            out.print("</tr>");
        }
        out.print("</table>");
        // 3. close session
        session.close();
    } catch (Exception e) {
        out.print("<h3 style='color:red'> Hibernate session
is failed ! "+e+"</h3>");
    }
}
}

```

## **UserLogin**

### **1. userbooking.java**

```

package userlogin;

import java.io.IOException;
import java.io.PrintWriter;
import java.text.DateFormat;
import java.text.Format;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
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 javax.servlet.http.HttpSession;

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

import entityclasses.*;
import util.HibernateSessionUtil;

@WebServlet("/userbooking")
public class userbooking extends HttpServlet{

    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        request.getRequestDispatcher("options.html").include(request,
response);

        int
flightnumber=Integer.parseInt(request.getParameter("flightnumber"));
        HttpSession ss=request.getSession(false);
        if(ss!=null) {

            try {
                // 1. build hibernate session factory
                SessionFactory factory =
HibernateSessionUtil.buildSessionFactory();

                // 2. create session object
                Session session = factory.openSession();

                // 3. read products
                List<flight> obj = session.createQuery("from flight").list();

                for(flight p : obj) {
                    if(p.getId()==flightnumber) {
                        ss.setAttribute("idvalue", p.getId());
                        ss.setAttribute("flightname",p.getAirline());
                        ss.setAttribute("sourceplace",
p.getSource());

```

```

                ss.setAttribute("destinationplace",
p.getDestination());
                response.sendRedirect("payment.html");
            }
        }

        // 3. close session
        session.close();

    } catch (Exception e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
}
}

```

## 2. availableflights.java

```

package userlogin;

import java.io.IOException;
import java.io.PrintWriter;
import java.text.DateFormat;
import java.text.Format;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
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 javax.servlet.http.HttpSession;

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

```

```

import entityclasses.*;
import util.HibernateSessionUtil;

@WebServlet("/availableflights")
public class availableflights extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        request.getRequestDispatcher("options.html").include(request,
response);

        String source = request.getParameter("source");
        String destination = request.getParameter("destination");
        String userdate = request.getParameter("dateoftravel");
        String no_of_people =
request.getParameter("numberofpersons");
        // my logic to check date is which day

        SimpleDateFormat format1 = new
SimpleDateFormat("dd/MM/yyyy");
        Date dt1;
        HttpSession ss = request.getSession(false);
        if (ss != null) {
            try {

                dt1 = format1.parse(userdate);
                DateFormat format2 = new
SimpleDateFormat("EEEE");
                String str = format2.format(dt1);

                try {

                    // 1. build hibernate session factory
                    SessionFactory factory =
HibernateSessionUtil.buildSessionFactory();

                    // 2. create session object
                    Session session = factory.openSession();

```

```

// 3. read products
List<flight> obj = session.createQuery("from
flight").list();

int flag = 0;
for (flight p : obj) {
    // res has days stored on which
    flights are available

    // str has the user required day
    String[] res =
    p.getAvailabledays().split("[,]", 0);

    for (String myStr : res) {
        // check user source
        destination with flight path along with date availability
        if (myStr.equals(str) &&
        p.getSource().equalsIgnoreCase(source)
        &&
        p.getDestination().equalsIgnoreCase(destination))
            flag = 1;
    }
}

// show data as table.
out.print("<h4>Available Flights for " +
userdate + " :- </h4>");

if (flag == 1) {
    flag = 0;
    out.print("<style> table,td,th {" +
"border:2px solid black;text-align:center;"
+ "padding: 10px; " +
"}</style>");

    out.print("<table >");
    out.print("<tr>");
    out.print("<th> ID</th>");
    out.print("<th> Airline Name</th>");
    out.print("<th> Ticket Price</th>");
    out.print("</tr>");

    int flag2 = 0;
    for (flight p : obj) {
        String[] res =
        p.getAvailabledays().split("[,]", 0);

        for (String myStr : res) {

```



```

        if (myStr.equals(str) &&
p.getSource().equalsIgnoreCase(source)
        &&
p.getDestination().equalsIgnoreCase(destination)) {
            flag2 = 1;
        }
    }
    if (flag2 == 1) {
        flag2 = 0;
        out.print("<tr>");
        out.print("<td>" +
p.getId() + "</td>");
        out.print("<td>" +
p.getAirline() + "</td>");
        out.print("<td>" +
p.getPrice() + "</td>");
        out.print("</tr>");
    }
}

out.print("</table>");
out.print("<br>");
out.print("<input style='color: green;'
type=\"submit\" value=\"Continue\">");

request.getRequestDispatcher("bookflight.html").include(request, response);
;

response.sendRedirect("bookflight.html");
}

// if flight is unavailable on that day
else {
    out.print("<h4 style='color:red'>No
Flights for selected date</h4>");
}
// 3. close session
ss.setAttribute("numberofpeople",
no_of_people);
session.close();
} catch (Exception e) {
    out.print("<h3 style='color:red'> Hibernate
session is failed ! " + e + "</h3>");
}

```

```

    }
    } catch (ParseException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    }
    }
}

```

### 3. payment.java

```

package userlogin;

import java.io.IOException;
import java.io.PrintWriter;
import java.text.DateFormat;
import java.text.Format;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
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 javax.servlet.http.HttpSession;

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

import entityclasses.*;
import util.HibernateSessionUtil;

@WebServlet("/payment")
public class payment extends HttpServlet {

```

```

private static final long serialVersionUID = 1L;

protected void service(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    request.getRequestDispatcher("options.html").include(request,
response);
    String password = request.getParameter("paymentpassword");
    HttpSession ss = request.getSession(false);
    if (ss != null) {
        try {
            // 1. build hibernate session factory
            SessionFactory factory =
HibernateSessionUtil.buildSessionFactory();

            // 2. create session object
            Session session = factory.openSession();

            // 3. read products
            List<user> obj = session.createQuery("from
user").list();

            List<flight> obj2 = session.createQuery("from
flight").list();

            int flag = 0;
            double temp = 0.0;
            double tempf = 0.0;
            for (user p : obj) {
                if (p.getPassword().equals(password)) {

                    // update user balance
                    int val = (Integer)
ss.getAttribute("idvalue");

                    int price = Integer.parseInt((String)
ss.getAttribute("numberofpeople"));

                    for (flight x : obj2) {
                        if (x.getId() == val) {
                            temp = (double)
x.getPrice() * price; // if user has booked tickets for 2 people then

                                // 2*ticketprice

```

```

        }
    }
    int userid = p.getId();

    tempf = p.getBalance() - temp;

    // session.createQuery("UPDATE
    user_data" +
    // "SET balance="+tempf+
    // "WHERE
    id="+userid).executeUpdate();

    // insufficient balance in account
    if (tempf < 0) {
        out.write("<h3
style='color:red'>Insufficient Account Balance !</h3>");
    } else {
        Transaction tx =
session.beginTransaction();

        Query q =
session.createQuery("update user set balance=:n where id=:i");
        q.setParameter("n", tempf);
        q.setParameter("i", userid);

        int status =

q.executeUpdate();

        tx.commit();
        ss.setAttribute("usersid",
p.getId());

response.sendRedirect("ticket");
    }
}

    session.close();
} catch (Exception e) {
    out.print("<h3 style='color:red'> Hibernate session
is failed !! " + e + "</h3>");
}
} else {
    out.write("<h2 style='color:red;'>Please Login</h2>");
}
}

```

```
    }  
}
```

#### 4. ticket.java

```
package userlogin;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.text.DateFormat;  
import java.text.Format;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
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 javax.servlet.http.HttpSession;  
  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
  
import entityclasses.*;  
import util.HibernateSessionUtil;  
  
@WebServlet("/ticket")  
public class ticket extends HttpServlet{  
  
    private static final long serialVersionUID = 1L;  
  
    protected void service(HttpServletRequest request,  
        HttpServletResponse response)  
        throws ServletException, IOException {  
  
        response.setContentType("text/html");  

```

```

        PrintWriter out = response.getWriter();
        HttpSession ss=request.getSession(false);
        if(ss!=null) {

            try {
                // 1. build hibernate session factory
                SessionFactory factory =
                HibernateSessionUtil.buildSessionFactory();

                // 2. create session object
                Session session = factory.openSession();

                Date date = new Date();
                SimpleDateFormat formatter = new
                SimpleDateFormat("dd/MM/yy");
                String str = formatter.format(date);

                // 3. read products
                List<user> obj = session.createQuery("from user").list();
                String n="";
                int flag=0;
                for(user p : obj) {
                    if(ss.getAttribute("userid").equals(p.getId())) {
                        n=p.getName();
                    }
                }
                out.print("<html><body>");
                out.print("<center><h1
                style=\"color:green\">CONGRATULATIONS!</h1>\"
                        + "</center>\n\" + "<center><h2
                style=\"color:green\">Your booking is confirmed.</h2>\"
                        + "</center>\n\" +
                        \" <img style=\"border-radius: 5px 5px 0
                0;width:35%;margin-left:50px;\"
                src=\"https://img.freepik.com/premium-vector/flying-airplane-icon-air-force-sy
                mbol-flight-sign-isolated-white-background_445292-153.jpg?w=2000\"
                alt=\"Avatar\">\n\" +
                        \" <div style=\"padding: 2px 16px;\">\n\");

                out.print("<center><h4><b> BOOKED BY -
                "+n+"</b></h4> \n\" +
                        "<h4><b> BOOKING ID -
                "+ss.getAttribute("userid")+"</b></h4>"+

```

```

        "<h4><b> AIRLINES -
"+ss.getAttribute("flightname")+"</b></h4>" +
        "<h4 style='color:green;'><b> From -
"+ss.getAttribute("sourceplace")+" || To -
"+ss.getAttribute("destinationplace")+"</b></h4>" +
        "<h4 style='color:green;'><b> Date of
booking - "+str+"</b></h4>" +
        "    <p>Thank you for choosing us!</p>
</center>\n" +
        " </div>\n" +
        "</div>\n" +
        "</center>");
        out.print("</body></html>");
        session.close();
    } catch (Exception e) {
        out.print("<h3 style='color:red'> Hibernate session is
failed ! "+e+"</h3>");
    }
}
}
}
}

```

## **Entry Classes**

### **1. admin.java**

```

package entityclasses;

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

public class admin {
    private String name;
    private String password;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private int id;

    public int getId() {

```

```

        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public admin(String name, String password, int id) {
        super();
        this.name = name;
        this.password = password;
        this.id = id;
    }

    public admin() {
        super();
        // TODO Auto-generated constructor stub
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    //public void changePassword(String name, String password,int id) {
    //    this.name = name;
    //    this.password = password;
    //    this.id=id;
    //}

}

```



## 2. user.java

```
package entityclasses;

import java.util.Date;
import util.HibernateSessionUtil;
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 = "user_data")
public class user {
    public static boolean usersesion = false;
    // properties
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private int id;

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

    @Column(name = "password")
    private String password;

    private double balance;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }
}
```

```

    public void setName(String name) {
        this.name = name;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public user() {

    }

    public user(int id, String name, String password, double balance) {
        super();
        this.id = id;
        this.name = name;
        this.balance = balance;
        this.password = password;
    }

    public user(String name, String password, double b) {

        this.name = name;
        this.password = password;
        balance = b;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

}

```

### 3. flights.java

```
package entityclasses;

import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;

public class flight {

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

    public flight() {
        super();
        // TODO Auto-generated constructor stub
    }

    private String source;
    private String destination;
    private String airline;
    private String availabledays;

    public String getAvailabledays() {
        return availabledays;
    }

    public void setAvailabledays(String availabledays) {
        this.availabledays = availabledays;
    }
}
```

```
    public flight(double price, int id, String source, String destination, String
airline) {
        super();
        this.price = price;
        this.id = id;
        this.source = source;
        this.destination = destination;
        this.airline = airline;
    }
```

```
    public flight(String source, String destination, String airline, double
price, String availabledays) {
        super();
        this.price = price;
        this.source = source;
        this.destination = destination;
        this.airline = airline;
        this.availabledays = availabledays;
    }
```

```
    public String getAirline() {
        return airline;
    }
```

```
    public void setAirline(String airline) {
        this.airline = airline;
    }
```

```
    public double getPrice() {
        return price;
    }
```

```
    public void setPrice(double price) {
        this.price = price;
    }
```

```
    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 void setId(int id) {
            this.id = id;
        }

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

```

#### 4. places.java

```

package entityclasses;

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

public class places {
    private String source;
    private String destination;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private int id;

    public int getId() {
        return id;
    }

    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 void setId(int id) {
        this.id = id;
    }

    public places() {
        super();
        // TODO Auto-generated constructor stub
    }

    public places(String source, String destination, int id) {
        super();
        this.source = source;
        this.destination = destination;
        this.id = id;
    }

    public places(String source, String destination) {
        super();
        this.source = source;
        this.destination = destination;
    }
}

```

## **Util**

### **HibernateSessionUtil.java**

```

package 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 HibernateSessionUtil {

    private static SessionFactory factory;

    public static SessionFactory buildSessionFactory() {

        // load configuration
        StandardServiceRegistry ssRegistry = new
StandardServiceRegistryBuilder()
            .configure("hibernate.cfg.xml").build();

        // prepare metadata / mapping
        Metadata metadata = new
MetadataSources(ssRegistry).getMetadataBuilder().build();

        // build session factory
        factory = metadata.getSessionFactoryBuilder().build();

        return factory;
    }
}

```

## **Configuration files**

### **hibernate.cfg.xml**

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</propert
y>

```

```

    <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/flyaway</proper
ty>
    <property name="hibernate.connection.username">root</property>
    <property
name="hibernate.connection.password">Achu@952628</property>
    <property name="hibernate.show_sql">true</property>
    <property name="hbm2ddl.auto">update</property>
    <!-- Hibernate Mapping -->
    <mapping resource="fly.hbm.xml"/>
    </session-factory>
</hibernate-configuration>

```

### **Fly.hbm.xml**

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="entityclasses.user" table="user_data">
        <id name="id" type="int" column="id">
            <generator class="increment"></generator>
        </id>
        <property name="name" type="string" column="name"
    ></property>
        <property name="password" type="string" column="password"
    ></property>
        <property name="balance" type="double" column="balance"
    ></property>
    </class>

    <class name="entityclasses.admin" table="admin_data">
        <id name="id" type="int" column="id">
            <generator class="increment"></generator>
        </id>
        <property name="name" type="string" column="name"
    ></property>
        <property name="password" type="string" column="password"
    ></property>
    </class>

```



```

    <class name="entityclasses.flight" table="flight_data">
        <id name="id" type="int" column="id">
            <generator class="increment"></generator>
        </id>

        <property name="source" type="string" column="source"
></property>
        <property name="destination" type="string"
column="destination" ></property>
        <property name="airline" type="string" column="airline"
></property>
        <property name="price" type="double" column="fare"
></property>
        <property name="availabledays" type="string"
column="availabledays" ></property>
    </class>

</hibernate-mapping>

```

## Project git Repositories

Link: <https://github.com/Adarshml/Practice-Projects/tree/main/FlyAway2>