flyaway.com

Airline Booking Portal

Prototype of the Application

Name : ISHAN LAHOTI

GitHub : https://github.com/inu1103/JavaPhase2.git

The prototype of the application starts from the frontend, and it can also directly start from the project folder. This portal allows us to do flight management across administrator and provide flight booking facilities across client side which will at the end page goes to the payment portal(dummy). This prototype is built through various webpages (mainly .jsp file) which are interconnected with backend (servlets, database, models).

The implementation is done with the help of Hibernate, maven, Servlet, Java EE, Apache tomcat v8.5 and Eclipse.

Sprint Planning

The Implementation is done in two sprints which are mentioned below:

Sprint 1:

- Clarify the specification and requirements.
- Implement a framework of certain pages such as admin login, homepage.
- Implement a blueprint of Controller, Models part at backend and its core functionality.
- Identifying the various association for mapping of Passenger with Flight database along with its attributes containing primary key in both the tables.

Sprint 2:

- Building a platform for the prototype with hibernate, maven (webapp archetype) integration along with MySQL as a database which will run on local server (Tomcat v8.5) and required dependencies.
- Creating JSP Page as a starting point containing a hyperlink that will take us to the Admin Login page and contains a html table containing booking details such as source, destination, date of boarding and number of persons.
- Afterward, as this part is broadly categorized into two parts: admin and passenger section.
- Introducing a single controller (Servlet) as the data will be share not just within admin or passenger section but also with each other, packages consisting of models, hibernate configuration, data transfer objects for database connectivity.

Sprint 3:

- Implement functionality in controller for validation in admin login page consisting of email and password which is stated as email (admin@test.com) and password (admin).

- Implement functionality for changing password in admin section which consist of new password and confirmation password (same as new password for recheck).
- Implementing a web page for admin login page and after successful login will jump to the admin main page.
- Implementing another JSP page for changing password which is not connected currently.

Sprint 4:

- Developing the main page displaying flight details along with add, change password and logout button.
- Adding functionality for adding flights acting as hyperlink that will take us to next page asking for Flight Details such as flight number, airline, origin, destination, flight date.
- Once the admin submits the required details, it will store all the valid data into the database through servlet along with auto increment primary key not null values. And it will display in the main page along with a “edit” and “delete” button on status column.

Sprint 5:

- From the passenger side, once a user registered the details for their travel, a filter operation is performed in backend by retrieving the flight details from database and filter it according to source, destination, and boarding date.
- Along each flight details, there will be a hyperlink button name “ Book Now”.
- From the “Book Now” button, it will jump to register page where user have to put his/her details according to the number of persons.

Sprint 6:

- In the registration page, there will be a option to select add passenger and a view table where user can view all the passenger details and modifications can also be done through status column option.
- Once the registration is done, it will show all the flight summary such as source, destination, date of boarding and total flight price.
- If a user registered passenger details more than number of persons, it will go back to the homepage again.

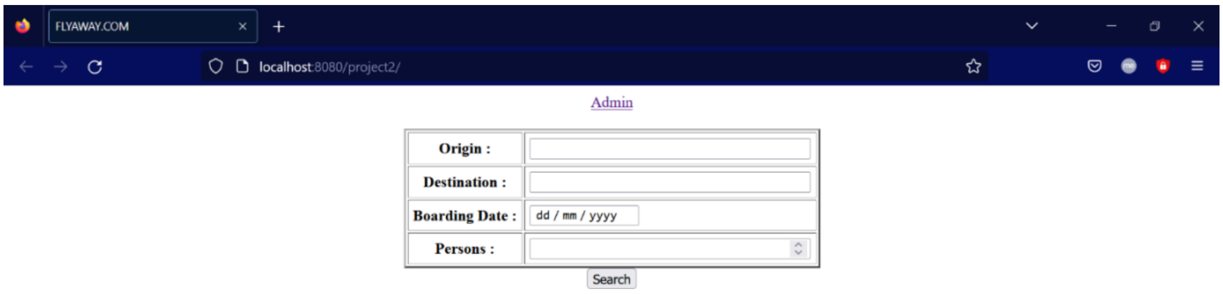
Sprint 7:

- From the passenger side, once a user registered the details for their travel, a filter operation is performed in backend by retrieving the flight details from database and filter it according to source, destination, and boarding date.
- Afterward, a dummy payment page will be displayed having an option to go back to home page as a hyperlink.
- Ensuring all the operations are tight and working well.
- Documentation.

Documentation of the functionality:

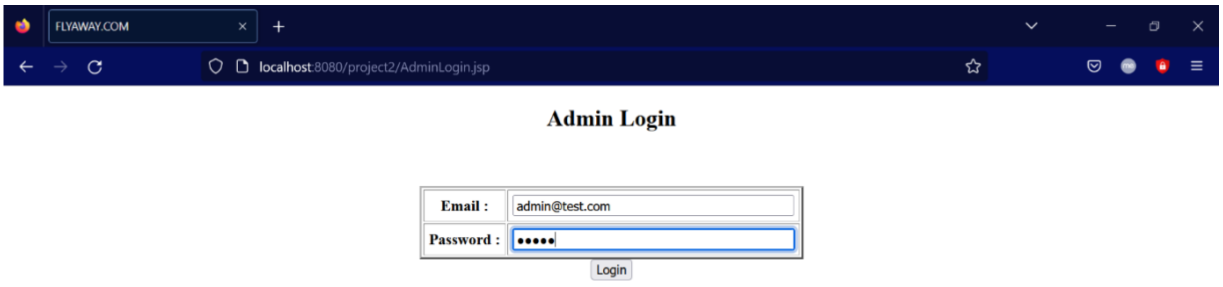
Here is the various static Java Servlet Pages that user will come across along the way.

1: Home Page



Origin :	
Destination :	
Boarding Date :	dd / mm / yyyy
Persons :	

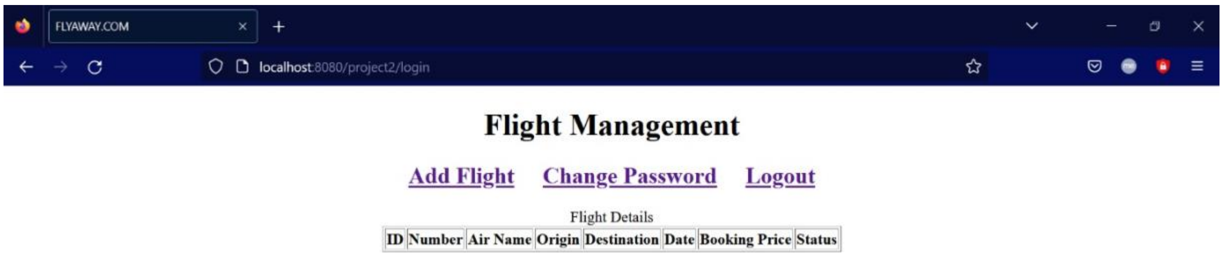
2: Admin Login Page (From admin side)



Admin Login

Email :	admin@test.com
Password :	•••••

3: Admin main page (From admin side)



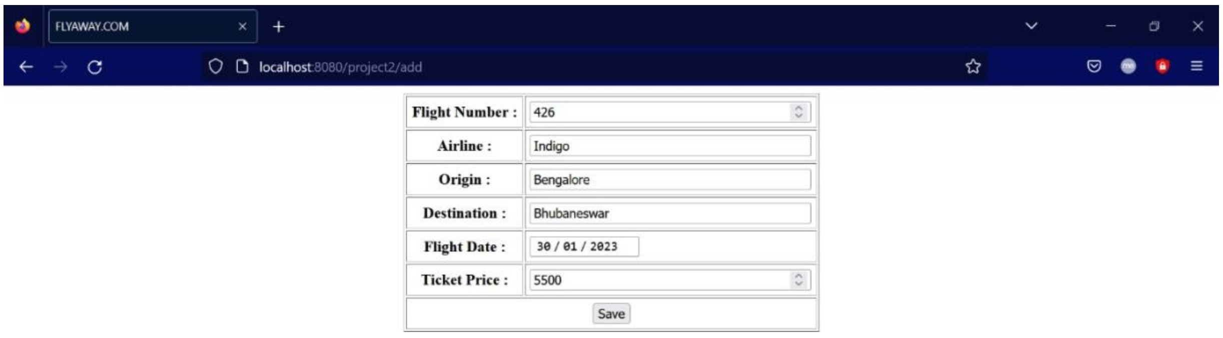
Flight Management

Add Flight Change Password Logout

Flight Details

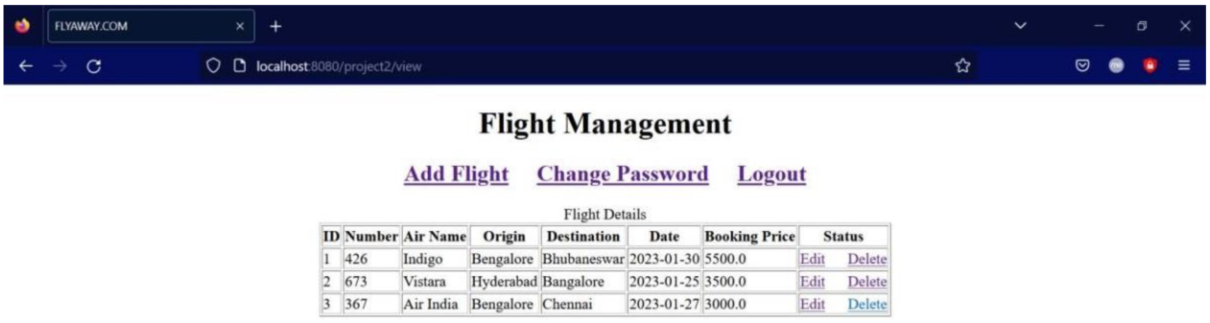
ID	Number	Air Name	Origin	Destination	Date	Booking Price	Status

4: Add Flight Page (From admin side)



Flight Number :	426
Airline :	Indigo
Origin :	Bengalore
Destination :	Bhubaneswar
Flight Date :	30 / 01 / 2023
Ticket Price :	5500

*Note: Once the user save it after inserting details, it will be inserted into the database.



Flight Management

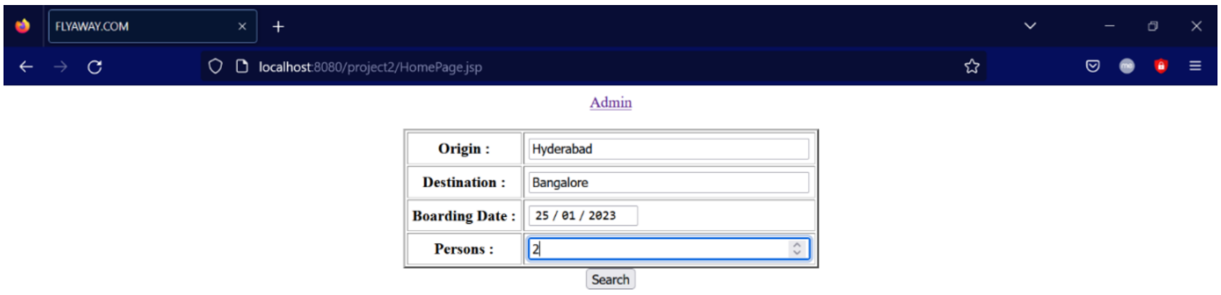
Add Flight   Change Password   Logout

Flight Details

ID	Number	Air Name	Origin	Destination	Date	Booking Price	Status	
1	426	Indigo	Bengalore	Bhubaneswar	2023-01-30	5500.0	Edit	Delete
2	673	Vistara	Hyderabad	Bangalore	2023-01-25	3500.0	Edit	Delete
3	367	Air India	Bengalore	Chennai	2023-01-27	3000.0	Edit	Delete

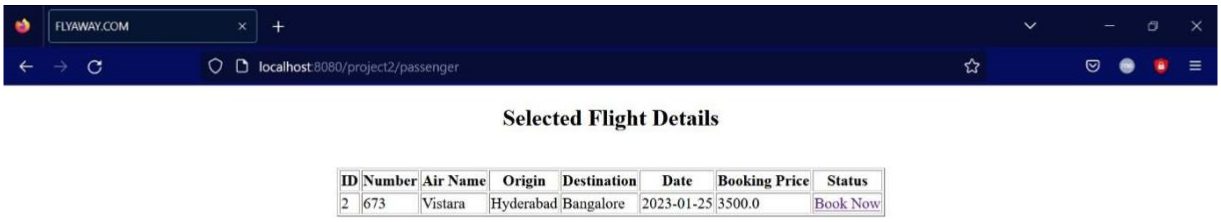
After, Admin can logout back to the homepage.

5: Insert the booking details from client side.



Origin :	Hyderabad
Destination :	Bangalore
Boarding Date :	25 / 01 / 2023
Persons :	2

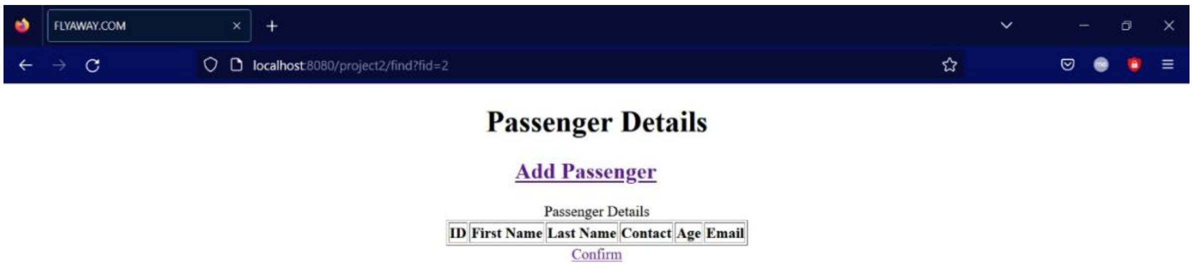
6: After inserting booking details, when user click the “search” button.



Selected Flight Details

ID	Number	Air Name	Origin	Destination	Date	Booking Price	Status
2	673	Vistara	Hyderabad	Bangalore	2023-01-25	3500.0	Book Now

7: After filtering flight details and user pressed the “Book Now” button. It will surf to Passenger Details.



Passenger Details

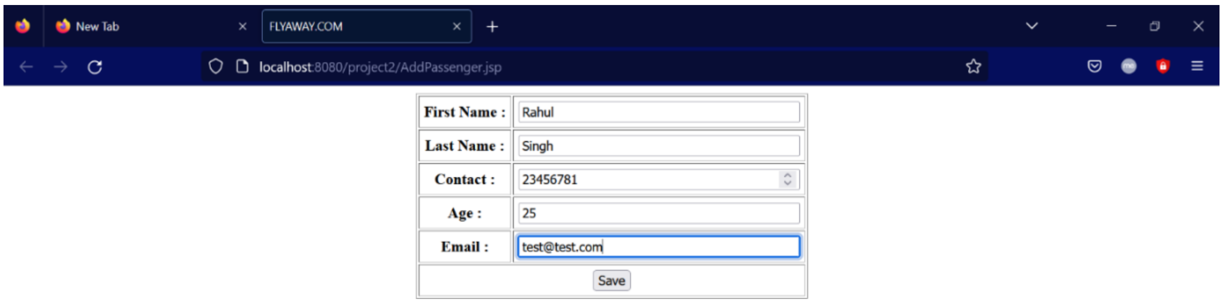
Add Passenger

Passenger Details

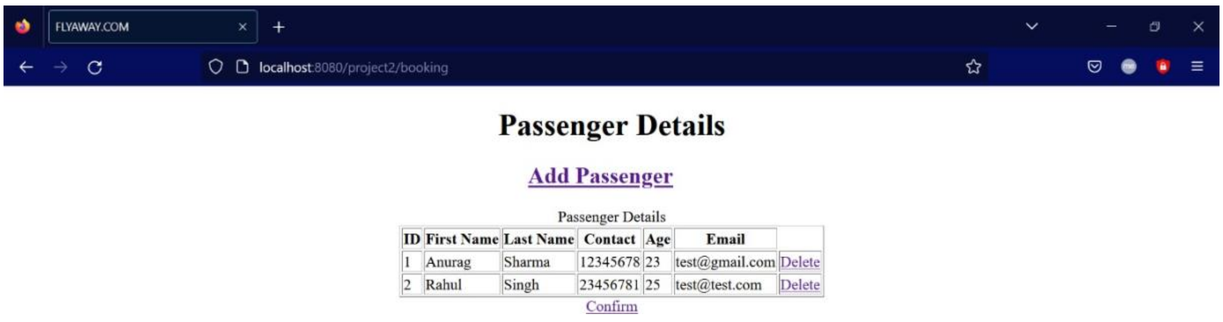
ID	First Name	Last Name	Contact	Age	Email

Confirm

8: Add Passenger Page, Entering Passenger Details



First Name :	Rahul
Last Name :	Singh
Contact :	23456781
Age :	25
Email :	test@test.com



Passenger Details

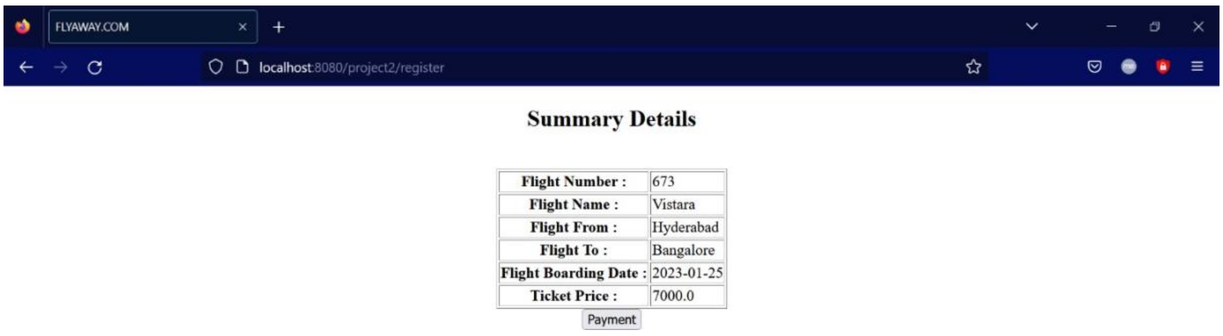
Add Passenger

Passenger Details

ID	First Name	Last Name	Contact	Age	Email	
1	Anurag	Sharma	12345678	23	test@gmail.com	Delete
2	Rahul	Singh	23456781	25	test@test.com	Delete

Confirm

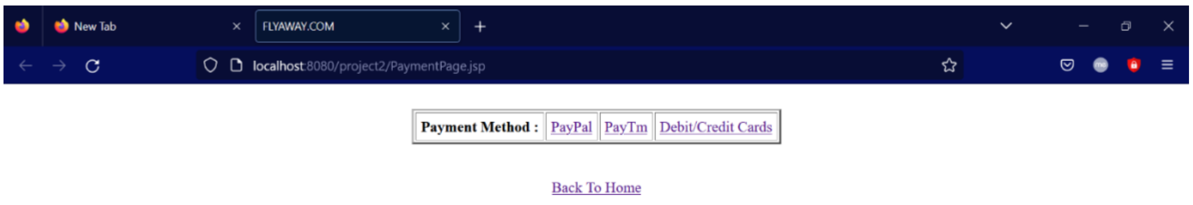
9: Summary Page



Summary Details

Flight Number :	673
Flight Name :	Vistara
Flight From :	Hyderabad
Flight To :	Bangalore
Flight Boarding Date :	2023-01-25
Ticket Price :	7000.0

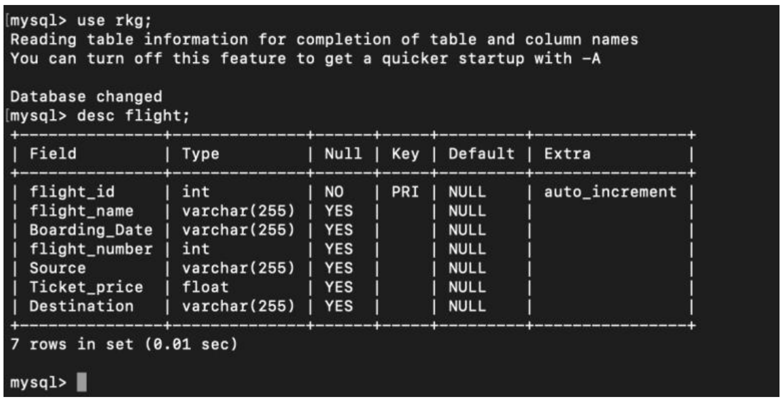
10: Payment Page



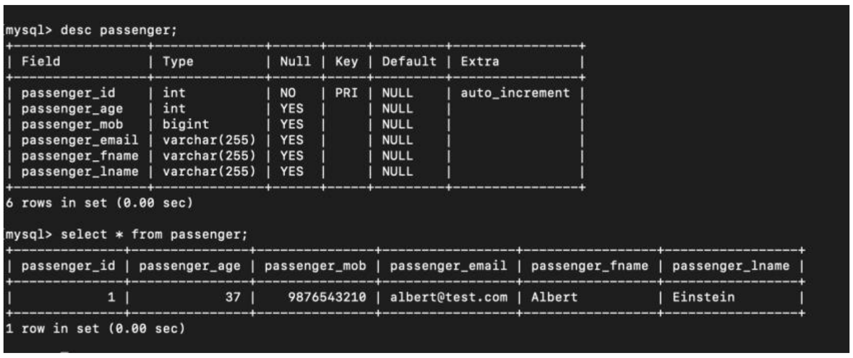
Payment Method :	PayPal	PayTm	Debit/Credit Cards

Back To Home

11: In MySqL Database, we have implemented @ManyToMany associations, here is the two tables flight and passengers table with its attributes and primary key.

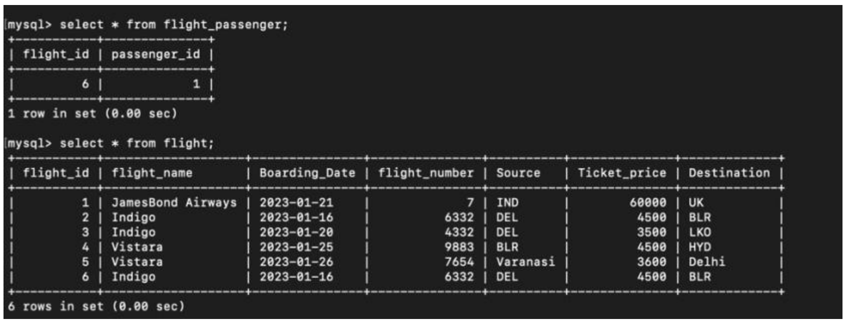


```
mysql> use rkg;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> desc flight;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| flight_id     | int          | NO   | PRI | NULL    | auto_increment |
| flight_name   | varchar(255) | YES  |     | NULL    |                |
| Boarding_Date | varchar(255) | YES  |     | NULL    |                |
| flight_number | int          | YES  |     | NULL    |                |
| Source        | varchar(255) | YES  |     | NULL    |                |
| Ticket_price  | float        | YES  |     | NULL    |                |
| Destination   | varchar(255) | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)

mysql> 
```



```
mysql> desc passenger;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| passenger_id    | int          | NO   | PRI | NULL    | auto_increment |
| passenger_age   | int          | YES  |     | NULL    |                |
| passenger_mob   | bigint       | YES  |     | NULL    |                |
| passenger_email | varchar(255) | YES  |     | NULL    |                |
| passenger_fname | varchar(255) | YES  |     | NULL    |                |
| passenger_lname | varchar(255) | YES  |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> select * from passenger;
+--------------+---------------+---------------+-----------------+-----------------+-----------------+
| passenger_id | passenger_age | passenger_mob | passenger_email | passenger_fname | passenger_lname |
+--------------+---------------+---------------+-----------------+-----------------+-----------------+
|            1 |            37 |    9876543210 | albert@test.com | Albert          | Einstein        |
+--------------+---------------+---------------+-----------------+-----------------+-----------------+
1 row in set (0.00 sec)
```

And a third table named (flight_passenger) containing the mapping of primary key of both the table is done.

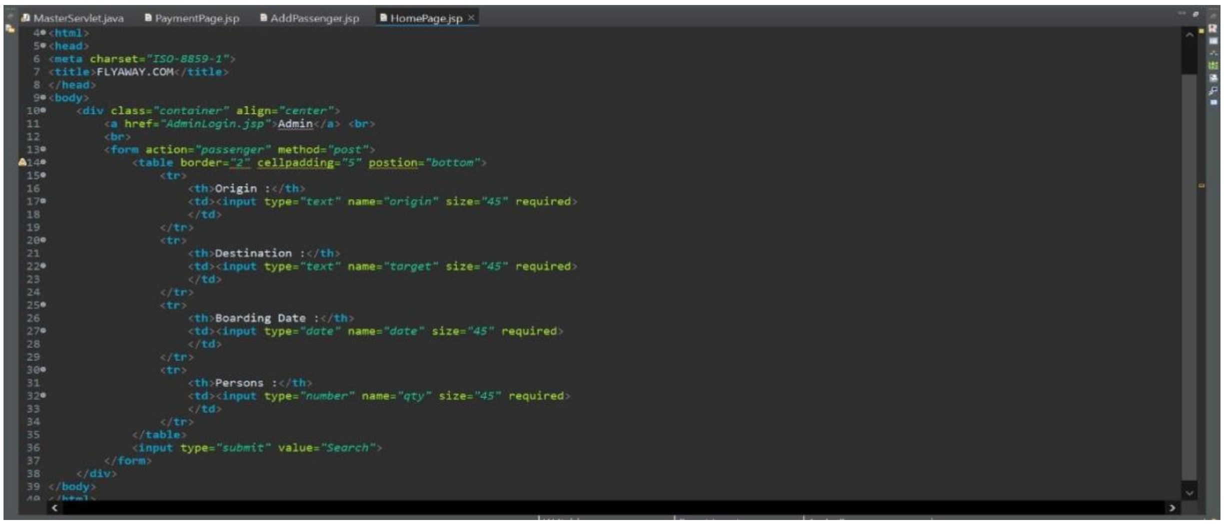


```
mysql> select * from flight_passenger;
+-----------+--------------+
| flight_id | passenger_id |
+-----------+--------------+
|         6 |            1 |
+-----------+--------------+
1 row in set (0.00 sec)

mysql> select * from flight;
+-----------+-------------------+---------------+---------------+----------+--------------+-------------+
| flight_id | flight_name       | Boarding_Date | flight_number | Source   | Ticket_price | Destination |
+-----------+-------------------+---------------+---------------+----------+--------------+-------------+
|         1 | JamesBond Airways | 2023-01-21    |             7 | IND      |        60000 | UK          |
|         2 | Indigo            | 2023-01-16    |          6332 | DEL      |         4500 | BLR         |
|         3 | Indigo            | 2023-01-20    |          4332 | DEL      |         3500 | LKO         |
|         4 | Vistara           | 2023-01-25    |          9883 | BLR      |         4500 | HYD         |
|         5 | Vistara           | 2023-01-26    |          7654 | Varanasi |         3600 | Delhi       |
|         6 | Indigo            | 2023-01-16    |          6332 | DEL      |         4500 | BLR         |
+-----------+-------------------+---------------+---------------+----------+--------------+-------------+
6 rows in set (0.00 sec)
```

Source Code:

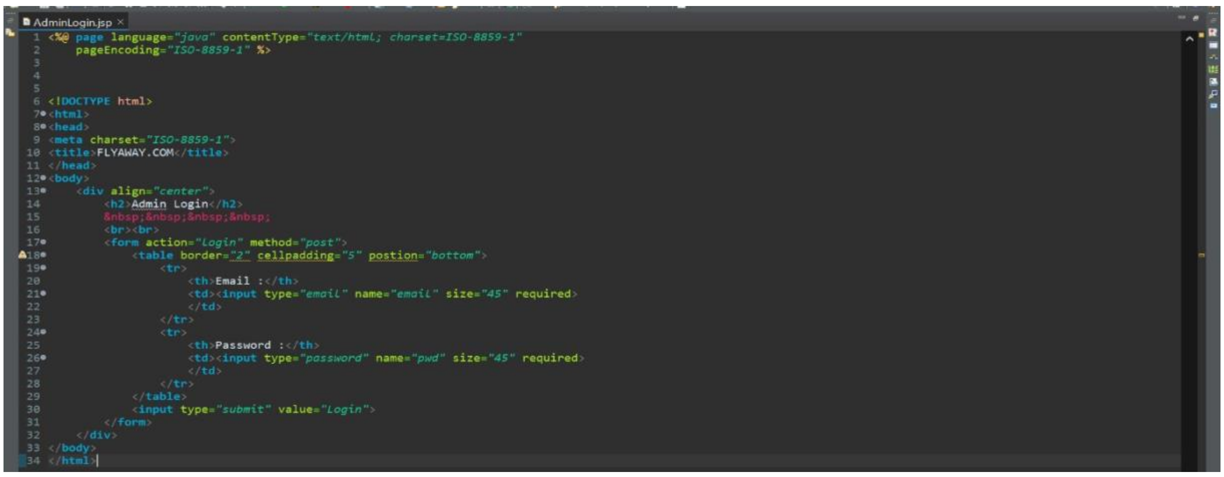
1: JSP Pages

a> Home Page (HomePage.jsp)



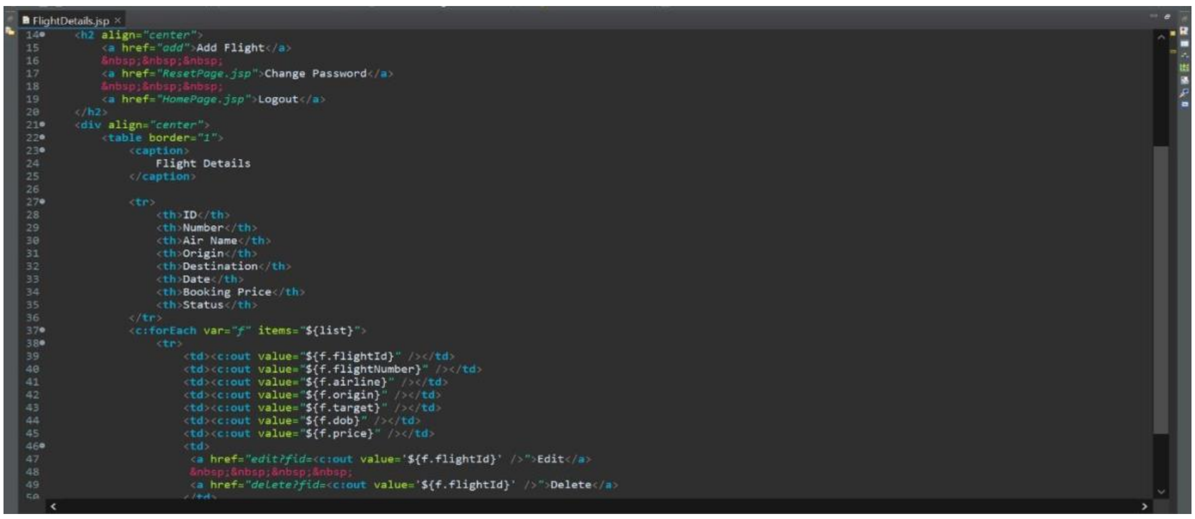
```
<html>
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <div class="container" align="center">
        <a href="AdminLogin.jsp">Admin</a> <br>
        <br>
        <form action="passenger" method="post">
            <table border="2" cellpadding="5" postion="bottom">
                <tr>
                    <th>Origin :</th>
                    <td><input type="text" name="origin" size="45" required>
                    </td>
                </tr>
                <tr>
                    <th>Destination :</th>
                    <td><input type="text" name="target" size="45" required>
                    </td>
                </tr>
                <tr>
                    <th>Boarding Date :</th>
                    <td><input type="date" name="date" size="45" required>
                    </td>
                </tr>
                <tr>
                    <th>Persons :</th>
                    <td><input type="number" name="qty" size="45" required>
                    </td>
                </tr>
            </table>
            <input type="submit" value="Search">
        </form>
    </div>
</body>
```

b> Admin Login Page (AdminLogin.jsp)



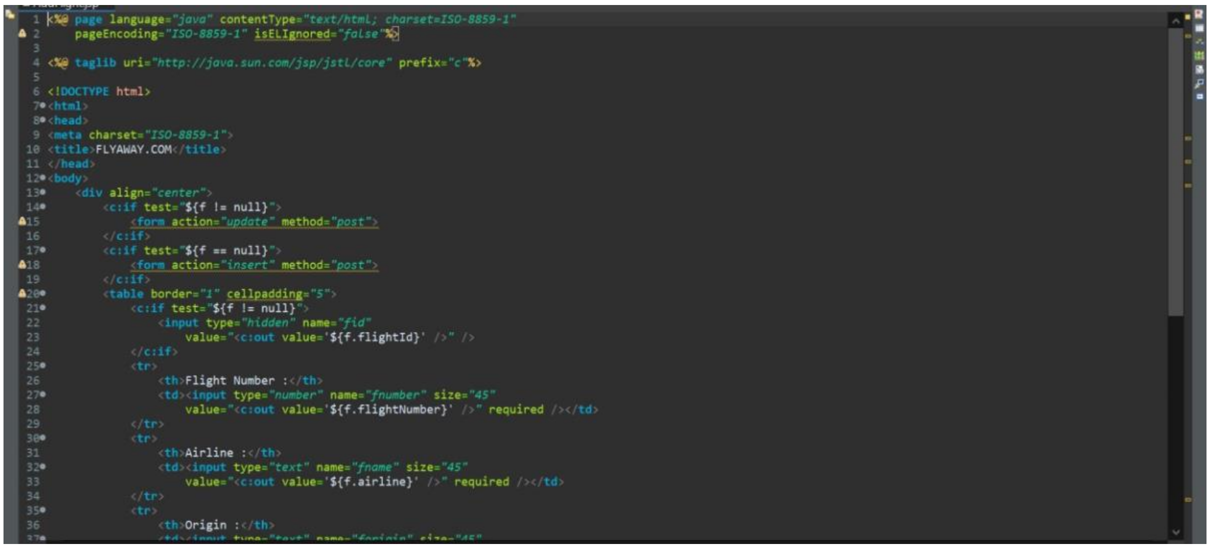
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" %>



<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <div align="center">
        <h2>Admin Login</h2>
        &nbsp;&nbsp;&nbsp;&nbsp;
        <br><br>
        <form action="Login" method="post">
            <table border="2" cellpadding="5" postion="bottom">
                <tr>
                    <th>Email :</th>
                    <td><input type="email" name="email" size="45" required>
                    </td>
                </tr>
                <tr>
                    <th>Password :</th>
                    <td><input type="password" name="pwd" size="45" required>
                    </td>
                </tr>
            </table>
            <input type="submit" value="Login">
        </form>
    </div>
</body>
</html>
```

c> Admin Main Page (FlightDetails.jsp)

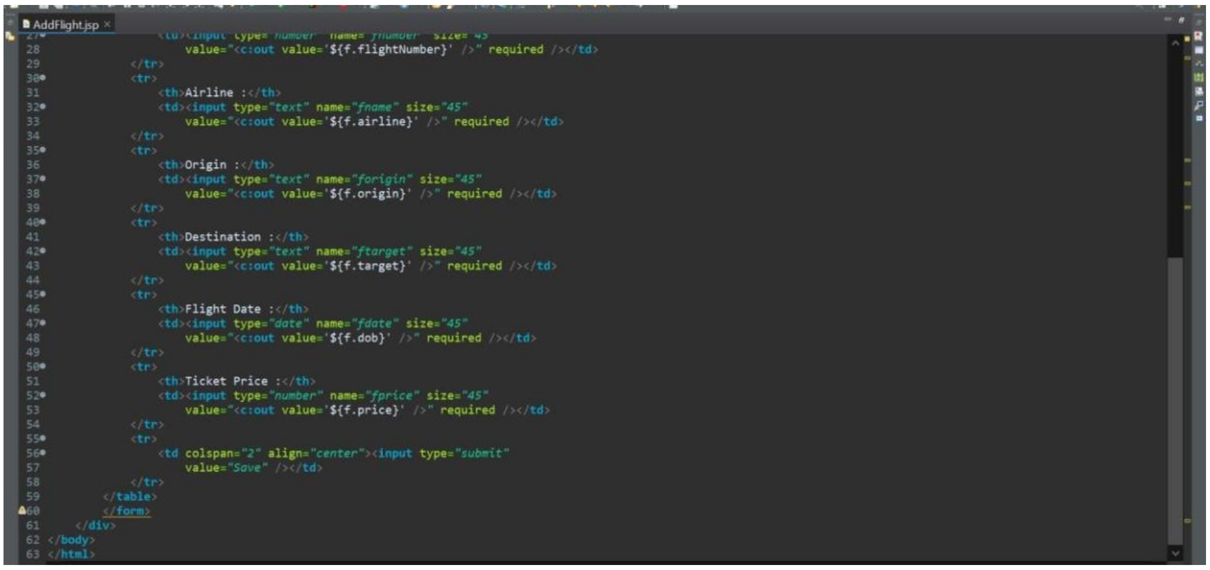


```
    <h2 align="center">
        <a href="add">Add Flight</a>
        &nbsp;&nbsp;&nbsp;
        <a href="ResetPage.jsp">Change Password</a>
        &nbsp;&nbsp;&nbsp;
        <a href="HomePage.jsp">Logout</a>
    </h2>
    <div align="center">
        <table border="1">
            <caption>
                Flight Details
            </caption>

            <tr>
                <th>ID</th>
                <th>Number</th>
                <th>Air Name</th>
                <th>Origin</th>
                <th>Destination</th>
                <th>Date</th>
                <th>Booking Price</th>
                <th>Status</th>
            </tr>
            <c:forEach var="f" items="${list}">
                <tr>
                    <td><c:out value="${f.flightId}" /></td>
                    <td><c:out value="${f.flightNumber}" /></td>
                    <td><c:out value="${f.airline}" /></td>
                    <td><c:out value="${f.origin}" /></td>
                    <td><c:out value="${f.target}" /></td>
                    <td><c:out value="${f.dob}" /></td>
                    <td><c:out value="${f.price}" /></td>
                    <td>
                     <a href="edit?fid=<c:out value='${f.flightId}' />">Edit</a>
                     &nbsp;&nbsp;&nbsp;&nbsp;
                     <a href="delete?fid=<c:out value='${f.flightId}' />">Delete</a>
```

d> Add Flight Details (AddFlight.jsp)

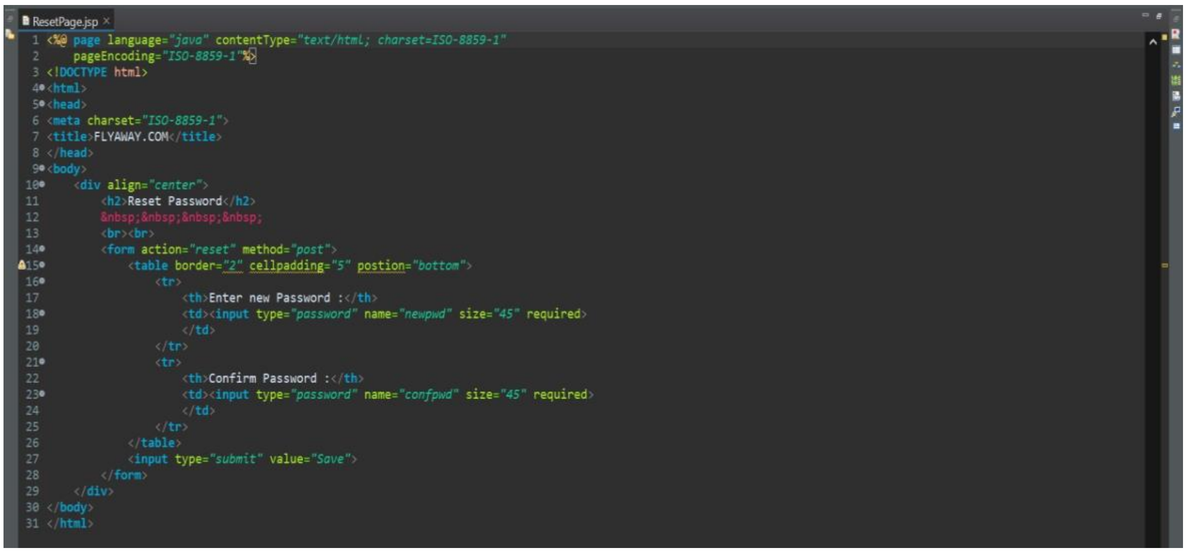


```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>

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

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <div align="center">
        <c:if test="${f != null}">
            <form action="update" method="post">
        </c:if>
        <c:if test="${f == null}">
            <form action="insert" method="post">
        </c:if>
        <table border="1" cellpadding="5">
            <c:if test="${f != null}">
                <input type="hidden" name="fid"
                    value="<c:out value='${f.flightId}' />" />
            </c:if>
            <tr>
                <th>Flight Number :</th>
                <td><input type="number" name="fnumber" size="45"
                    value="<c:out value='${f.flightNumber}' />" required /></td>
            </tr>
            <tr>
                <th>Airline :</th>
                <td><input type="text" name="fname" size="45"
                    value="<c:out value='${f.airline}' />" required /></td>
            </tr>
            <tr>
                <th>Origin :</th>
                <td><input type="text" name="forigin" size="45"
```



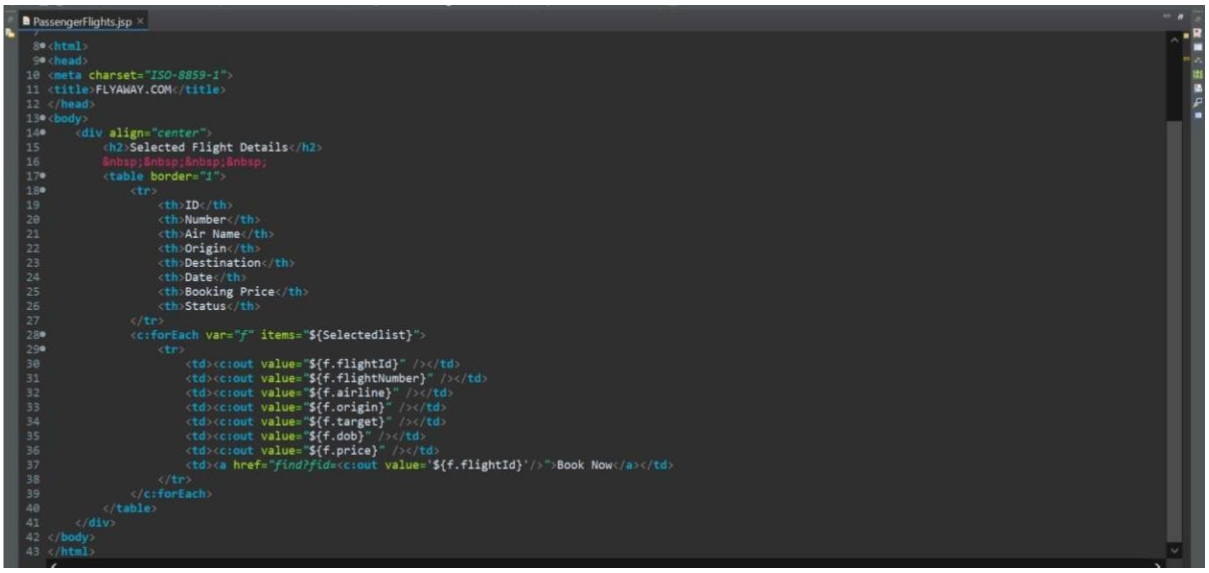
```
                <td><input type="number" name="fnumber" size="45"
                    value="<c:out value='${f.flightNumber}' />" required /></td>
            </tr>
            <tr>
                <th>Airline :</th>
                <td><input type="text" name="fname" size="45"
                    value="<c:out value='${f.airline}' />" required /></td>
            </tr>
            <tr>
                <th>Origin :</th>
                <td><input type="text" name="forigin" size="45"
                    value="<c:out value='${f.origin}' />" required /></td>
            </tr>
            <tr>
                <th>Destination :</th>
                <td><input type="text" name="ftarget" size="45"
                    value="<c:out value='${f.target}' />" required /></td>
            </tr>
            <tr>
                <th>Flight Date :</th>
                <td><input type="date" name="fdate" size="45"
                    value="<c:out value='${f.dob}' />" required /></td>
            </tr>
            <tr>
                <th>Ticket Price :</th>
                <td><input type="number" name="fprice" size="45"
                    value="<c:out value='${f.price}' />" required /></td>
            </tr>
            <tr>
                <td colspan="2" align="center"><input type="submit"
                    value="Save" /></td>
            </tr>
        </table>
        </form>
    </div>
</body>
</html>
```

e> Change Admin Password (ResetPage.jsp)



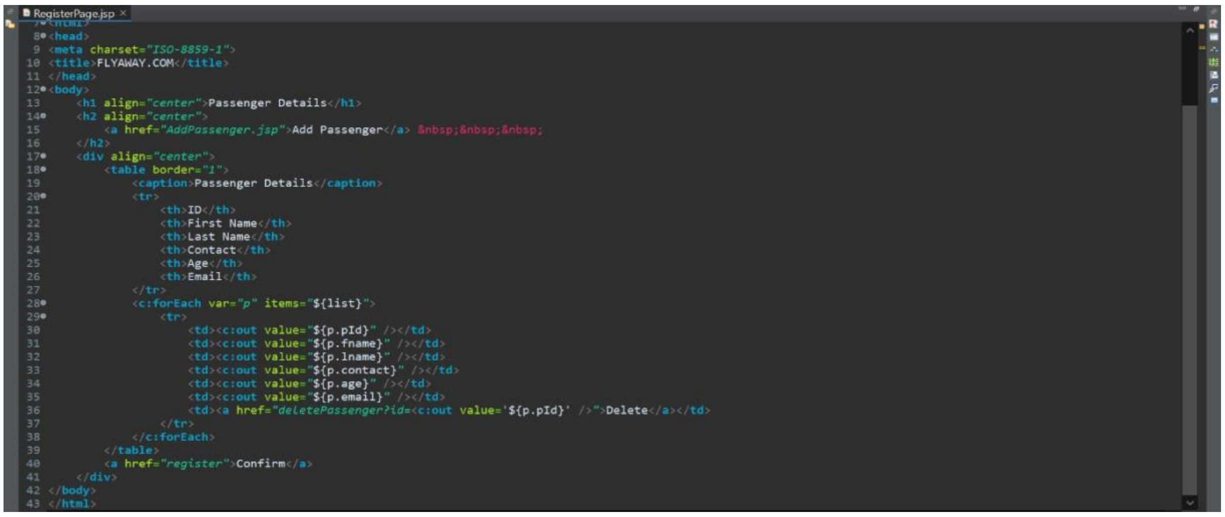
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <div align="center">
        <h2>Reset Password</h2>
        &nbsp;&nbsp;&nbsp;&nbsp;
        <br><br>
        <form action="reset" method="post">
            <table border="2" cellpadding="5" postion="bottom">
                <tr>
                    <th>Enter new Password :</th>
                    <td><input type="password" name="newpwd" size="45" required>
                    </td>
                </tr>
                <tr>
                    <th>Confirm Password :</th>
                    <td><input type="password" name="confpwd" size="45" required>
                    </td>
                </tr>
            </table>
            <input type="submit" value="Save">
        </form>
    </div>
</body>
</html>
```

f> Passenger Flights Page (PassengerFlights.jsp)



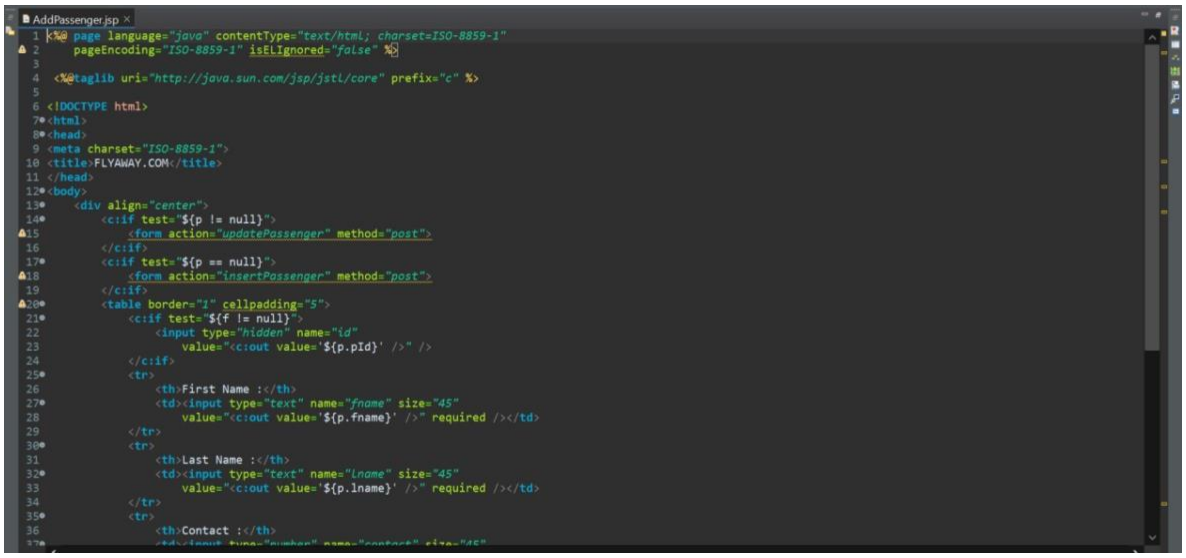
```
<html>
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <div align="center">
        <h2>Selected Flight Details</h2>
        &nbsp;&nbsp;&nbsp;&nbsp;
        <table border="1">
            <tr>
                <th>ID</th>
                <th>Number</th>
                <th>Air Name</th>
                <th>Origin</th>
                <th>Destination</th>
                <th>Date</th>
                <th>Booking Price</th>
                <th>Status</th>
            </tr>
            <c:forEach var="f" items="${Selectedlist}">
                <tr>
                    <td><c:out value="${f.flightId}" /></td>
                    <td><c:out value="${f.flightNumber}" /></td>
                    <td><c:out value="${f.airline}" /></td>
                    <td><c:out value="${f.origin}" /></td>
                    <td><c:out value="${f.target}" /></td>
                    <td><c:out value="${f.dob}" /></td>
                    <td><c:out value="${f.price}" /></td>
                    <td><a href="find?fid=<c:out value='${f.flightId}'/>">Book Now</a></td>
                </tr>
            </c:forEach>
        </table>
    </div>
</body>
</html>
```

g> Passenger Register Page (RegisterPage.jsp)

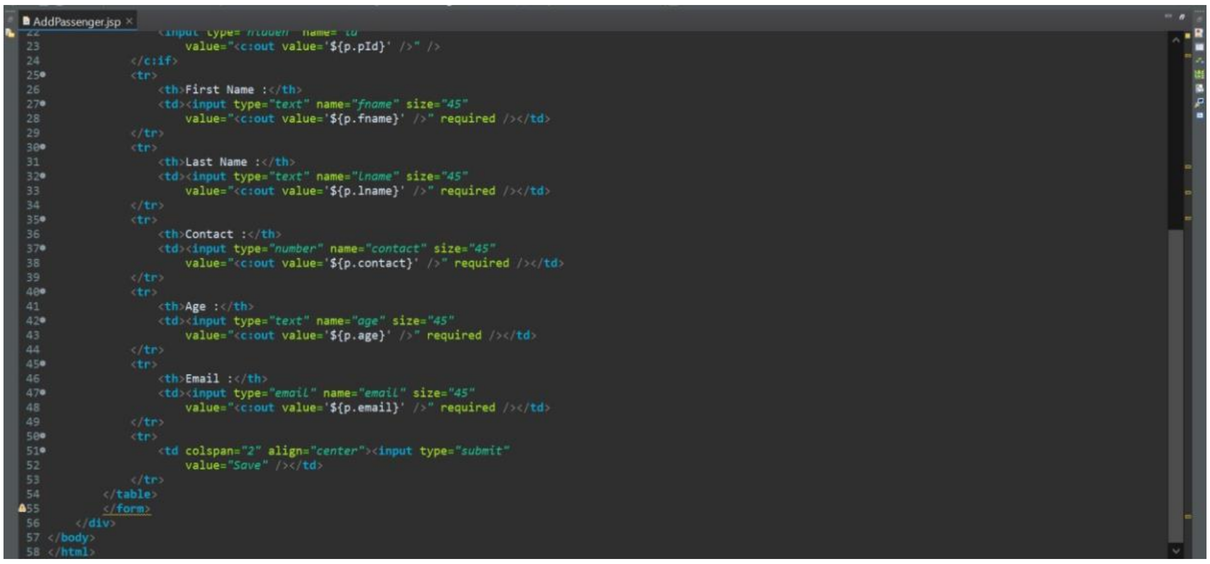


```
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <h1 align="center">Passenger Details</h1>
    <h2 align="center">
        <a href="AddPassenger.jsp">Add Passenger</a> &nbsp;&nbsp;&nbsp;
    </h2>
    <div align="center">
        <table border="1">
            <caption>Passenger Details</caption>
            <tr>
                <th>ID</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Contact</th>
                <th>Age</th>
                <th>Email</th>
            </tr>
            <c:forEach var="p" items="${list}">
                <tr>
                    <td><c:out value="${p.pId}" /></td>
                    <td><c:out value="${p.fname}" /></td>
                    <td><c:out value="${p.lname}" /></td>
                    <td><c:out value="${p.contact}" /></td>
                    <td><c:out value="${p.age}" /></td>
                    <td><c:out value="${p.email}" /></td>
                    <td><a href="deletePassenger?id=<c:out value='${p.pId}' />">Delete</a></td>
                </tr>
            </c:forEach>
        </table>
        <a href="register">Confirm</a>
    </div>
</body>
</html>
```

h> Add Passenger (AddPassenger.jsp)

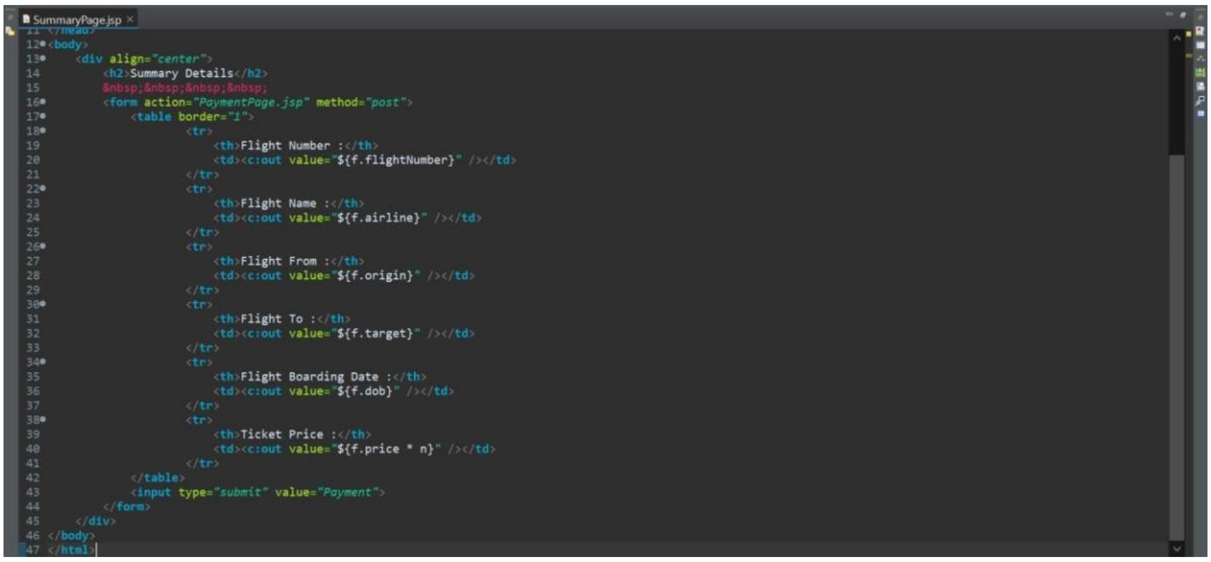


```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false" %>

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

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <div align="center">
        <c:if test="${p != null}">
            <form action="updatePassenger" method="post">
        </c:if>
        <c:if test="${p == null}">
            <form action="insertPassenger" method="post">
        </c:if>
        <table border="1" cellpadding="5">
            <c:if test="${f != null}">
                <input type="hidden" name="id"
                    value="<c:out value='${p.pId}' />" />
            </c:if>
            <tr>
                <th>First Name :</th>
                <td><input type="text" name="fname" size="45"
                    value="<c:out value='${p.fname}' />" required /></td>
            </tr>
            <tr>
                <th>Last Name :</th>
                <td><input type="text" name="lname" size="45"
                    value="<c:out value='${p.lname}' />" required /></td>
            </tr>
            <tr>
                <th>Contact :</th>
                <td><input type="number" name="contact" size="45"
```



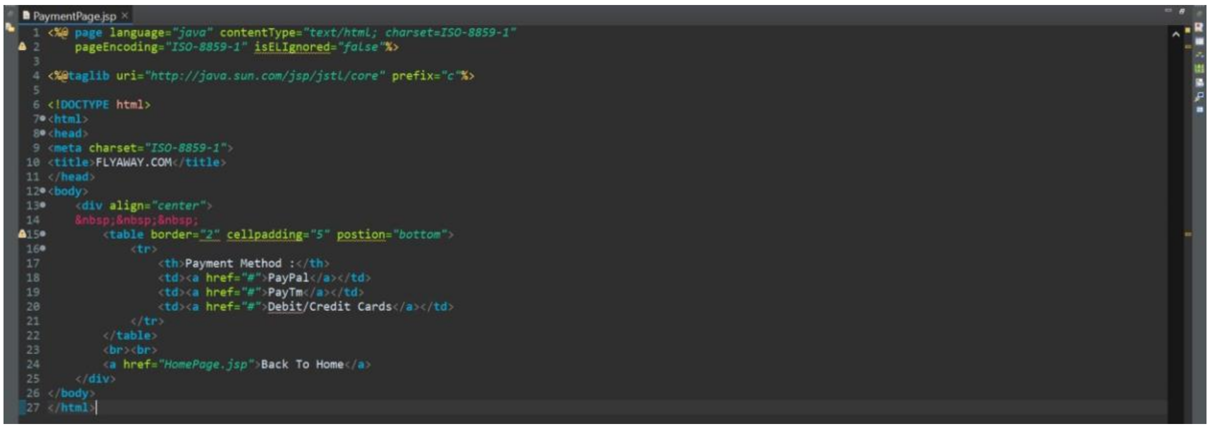
```
                <input type="hidden" name="id"
                    value="<c:out value='${p.pId}' />" />
            </c:if>
            <tr>
                <th>First Name :</th>
                <td><input type="text" name="fname" size="45"
                    value="<c:out value='${p.fname}' />" required /></td>
            </tr>
            <tr>
                <th>Last Name :</th>
                <td><input type="text" name="lname" size="45"
                    value="<c:out value='${p.lname}' />" required /></td>
            </tr>
            <tr>
                <th>Contact :</th>
                <td><input type="number" name="contact" size="45"
                    value="<c:out value='${p.contact}' />" required /></td>
            </tr>
            <tr>
                <th>Age :</th>
                <td><input type="text" name="age" size="45"
                    value="<c:out value='${p.age}' />" required /></td>
            </tr>
            <tr>
                <th>Email :</th>
                <td><input type="email" name="email" size="45"
                    value="<c:out value='${p.email}' />" required /></td>
            </tr>
            <tr>
                <td colspan="2" align="center"><input type="submit"
                    value="Save" /></td>
            </tr>
        </table>
        </form>
    </div>
</body>
</html>
```

i> Flight Summary Page (SummaryPage.jsp)



```
</head>
<body>
    <div align="center">
        <h2>Summary Details</h2>
        &nbsp;&nbsp;&nbsp;&nbsp;
        <form action="PaymentPage.jsp" method="post">
            <table border="1">
                <tr>
                    <th>Flight Number :</th>
                    <td><c:out value="${f.flightNumber}" /></td>
                </tr>
                <tr>
                    <th>Flight Name :</th>
                    <td><c:out value="${f.airline}" /></td>
                </tr>
                <tr>
                    <th>Flight From :</th>
                    <td><c:out value="${f.origin}" /></td>
                </tr>
                <tr>
                    <th>Flight To :</th>
                    <td><c:out value="${f.target}" /></td>
                </tr>
                <tr>
                    <th>Flight Boarding Date :</th>
                    <td><c:out value="${f.dob}" /></td>
                </tr>
                <tr>
                    <th>Ticket Price :</th>
                    <td><c:out value="${f.price * n}" /></td>
                </tr>
            </table>
            <input type="submit" value="Payment">
        </form>
    </div>
</body>
</html>
```

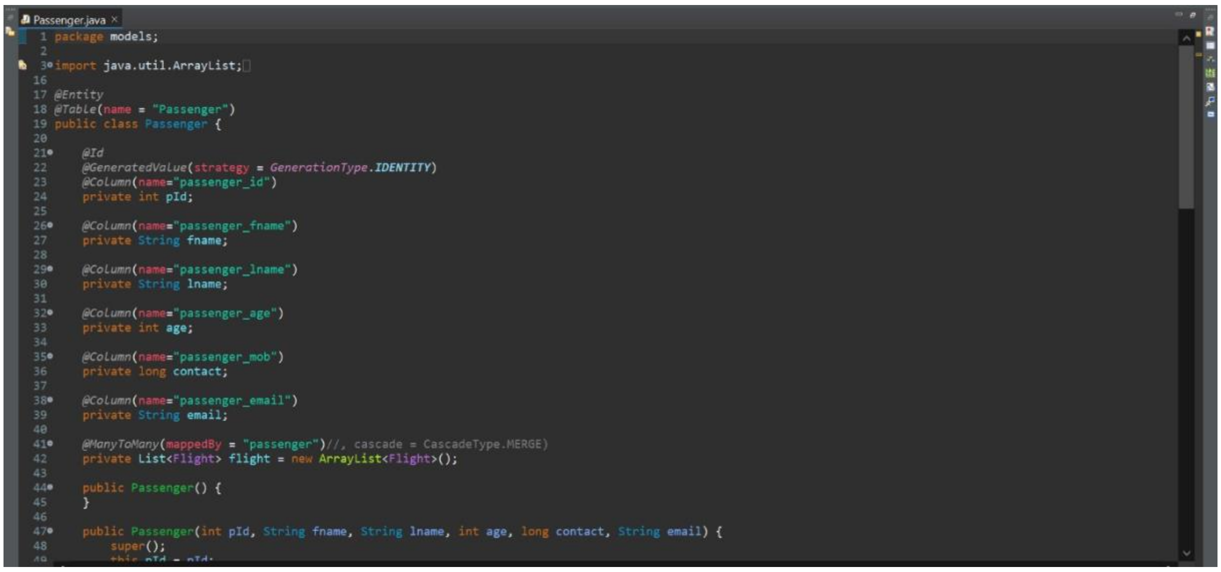
j> Dummy Payment Gateway (PaymentPage.jsp)



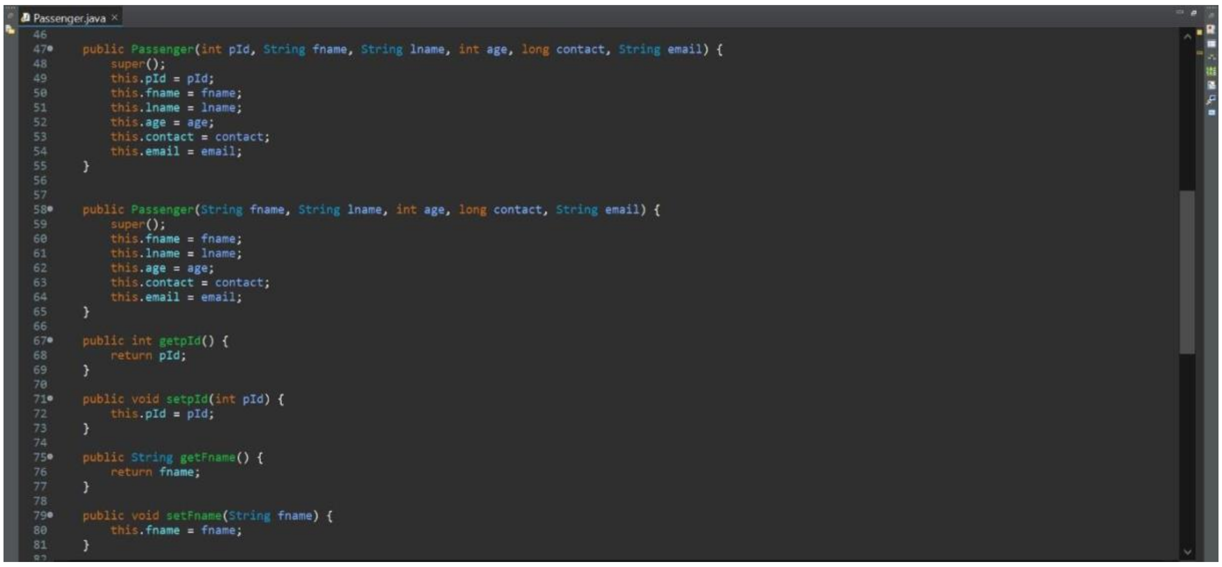
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>

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

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>FLYAWAY.COM</title>
</head>
<body>
    <div align="center">
    &nbsp;&nbsp;&nbsp;
        <table border="2" cellpadding="5" postion="bottom">
            <tr>
                <th>Payment Method :</th>
                <td><a href="#">PayPal</a></td>
                <td><a href="#">PayTm</a></td>
                <td><a href="#">Debit/Credit Cards</a></td>
            </tr>
        </table>
        <br><br>
        <a href="HomePage.jsp">Back To Home</a>
    </div>
</body>
</html>
```

2: Models (Passenger.java, Flight.java, Password.java)

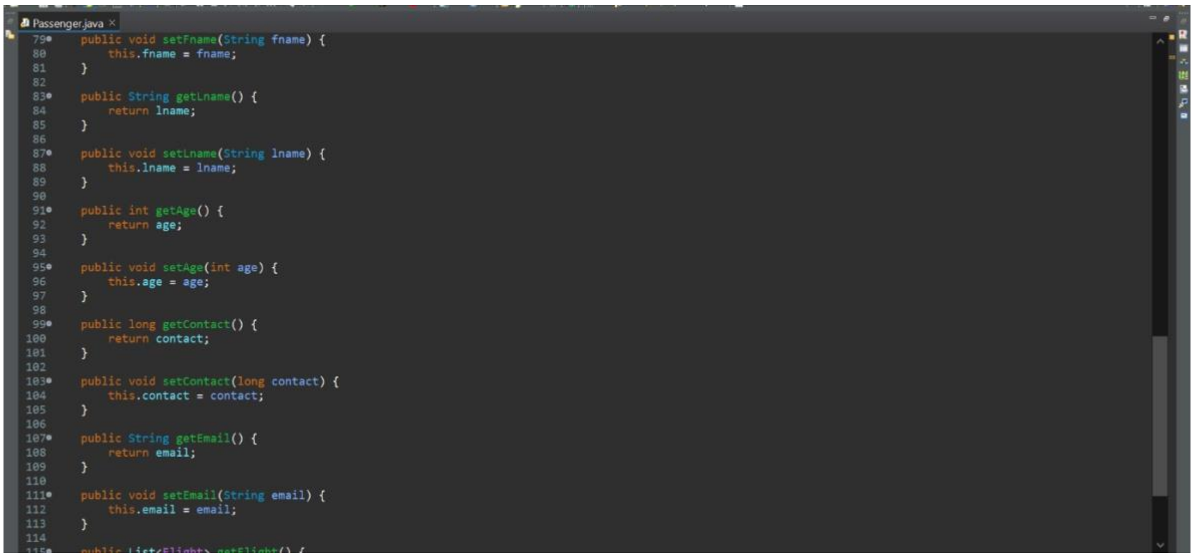
a> Passenger.java



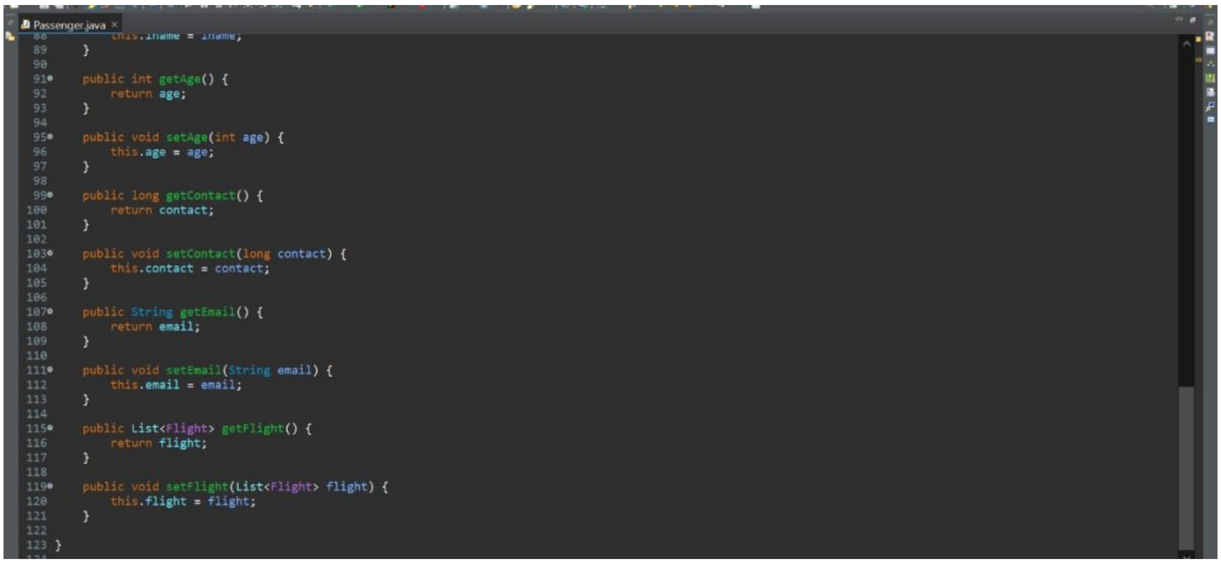
```java
package models;

import java.util.ArrayList;

@Entity
@Table(name = "Passenger")
public class Passenger {

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

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

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

    @Column(name="passenger_age")
    private int age;

    @Column(name="passenger_mob")
    private long contact;

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

    @ManyToMany(mappedBy = "passenger")//, cascade = CascadeType.MERGE)
    private List<Flight> flight = new ArrayList<Flight>();

    public Passenger() {
    }

    public Passenger(int pId, String fname, String lname, int age, long contact, String email) {
        super();
```



```java

    public Passenger(int pId, String fname, String lname, int age, long contact, String email) {
        super();
        this.pId = pId;
        this.fname = fname;
        this.lname = lname;
        this.age = age;
        this.contact = contact;
        this.email = email;
    }


    public Passenger(String fname, String lname, int age, long contact, String email) {
        super();
        this.fname = fname;
        this.lname = lname;
        this.age = age;
        this.contact = contact;
        this.email = email;
    }

    public int getpId() {
        return pId;
    }

    public void setpId(int pId) {
        this.pId = pId;
    }

    public String getFname() {
        return fname;
    }

    public void setFname(String fname) {
        this.fname = fname;
    }
```

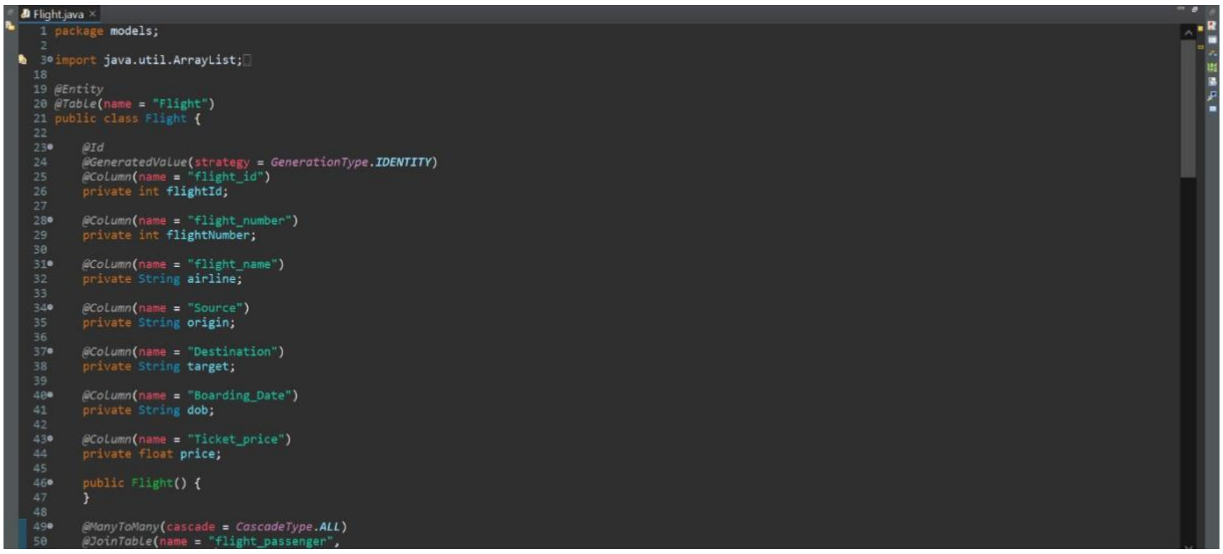


```java
    public void setFname(String fname) {
        this.fname = fname;
    }

    public String getLname() {
        return lname;
    }

    public void setLname(String lname) {
        this.lname = lname;
    }

    public int getAge() {
        return age;
    }

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

    public long getContact() {
        return contact;
    }

    public void setContact(long contact) {
        this.contact = contact;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

```

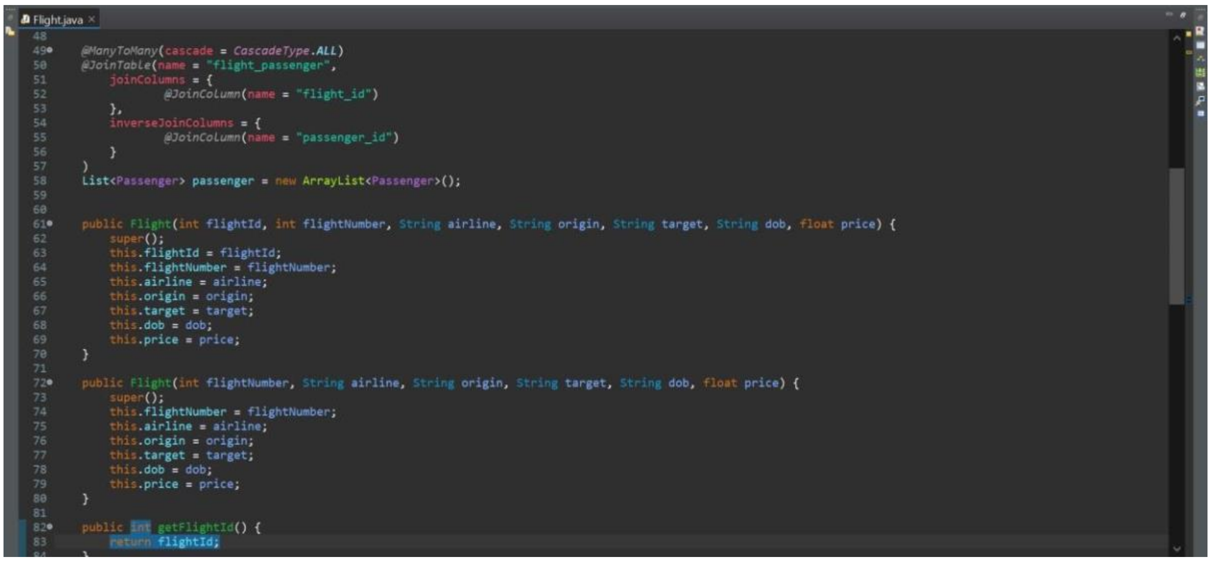


```java
        this.lname = lname;
    }

    public int getAge() {
        return age;
    }

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

    public long getContact() {
        return contact;
    }

    public void setContact(long contact) {
        this.contact = contact;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public List<Flight> getFlight() {
        return flight;
    }

    public void setFlight(List<Flight> flight) {
        this.flight = flight;
    }

}
```

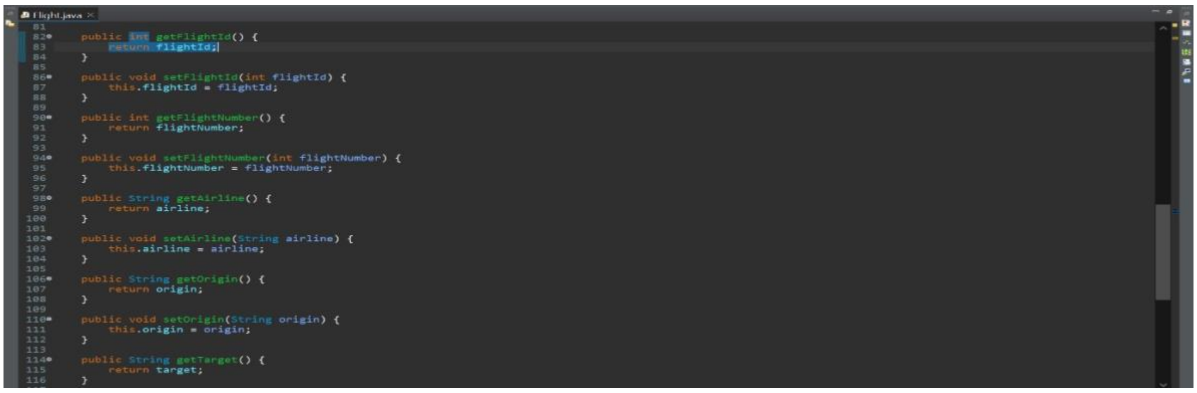
b> Flight.java



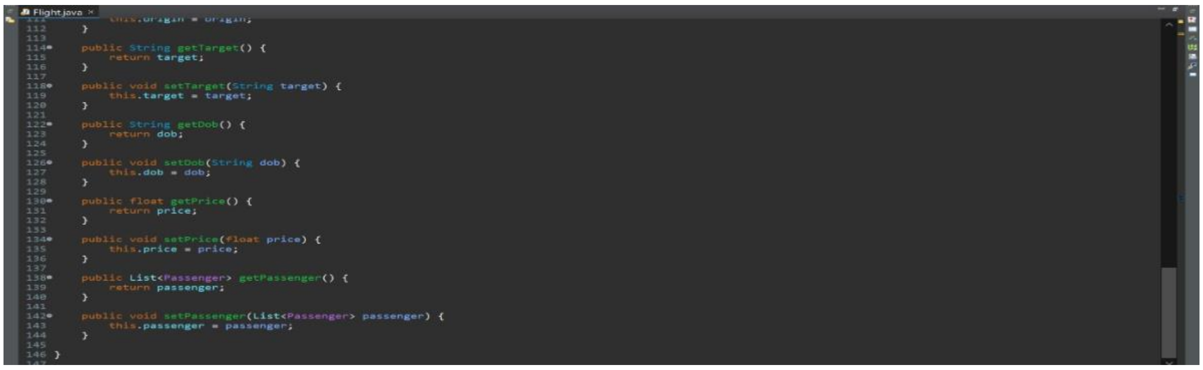
```java
package models;

import java.util.ArrayList;

@Entity
@Table(name = "Flight")
public class Flight {

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

    @Column(name = "flight_number")
    private int flightNumber;

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

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

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

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

    @Column(name = "Ticket_price")
    private float price;

    public Flight() {
    }

    @ManyToMany(cascade = CascadeType.ALL)
    @JoinTable(name = "flight_passenger",
```



```java

    @ManyToMany(cascade = CascadeType.ALL)
    @JoinTable(name = "flight_passenger",
        joinColumns = {
                @JoinColumn(name = "flight_id")
        },
        inverseJoinColumns = {
                @JoinColumn(name = "passenger_id")
        }
    )
    List<Passenger> passenger = new ArrayList<Passenger>();


    public Flight(int flightId, int flightNumber, String airline, String origin, String target, String dob, float price) {
        super();
        this.flightId = flightId;
        this.flightNumber = flightNumber;
        this.airline = airline;
        this.origin = origin;
        this.target = target;
        this.dob = dob;
        this.price = price;
    }

    public Flight(int flightNumber, String airline, String origin, String target, String dob, float price) {
        super();
        this.flightNumber = flightNumber;
        this.airline = airline;
        this.origin = origin;
        this.target = target;
        this.dob = dob;
        this.price = price;
    }

    public int getFlightId() {
        return flightId;
```

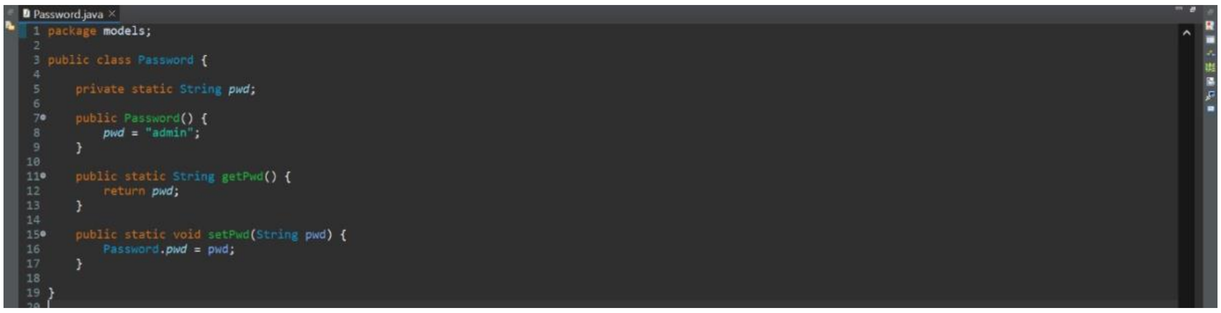


```java
    public int getFlightId() {
        return flightId;
    }

    public void setFlightId(int flightId) {
        this.flightId = flightId;
    }

    public int getFlightNumber() {
        return flightNumber;
    }

    public void setFlightNumber(int flightNumber) {
        this.flightNumber = flightNumber;
    }

    public String getAirline() {
        return airline;
    }

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

    public String getOrigin() {
        return origin;
    }

    public void setOrigin(String origin) {
        this.origin = origin;
    }

    public String getTarget() {
        return target;
    }
```



```java
    }

    public String getTarget() {
        return target;
    }

    public void setTarget(String target) {
        this.target = target;
    }

    public String getDob() {
        return dob;
    }

    public void setDob(String dob) {
        this.dob = dob;
    }

    public float getPrice() {
        return price;
    }

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

    public List<Passenger> getPassenger() {
        return passenger;
    }

    public void setPassenger(List<Passenger> passenger) {
        this.passenger = passenger;
    }

}
```

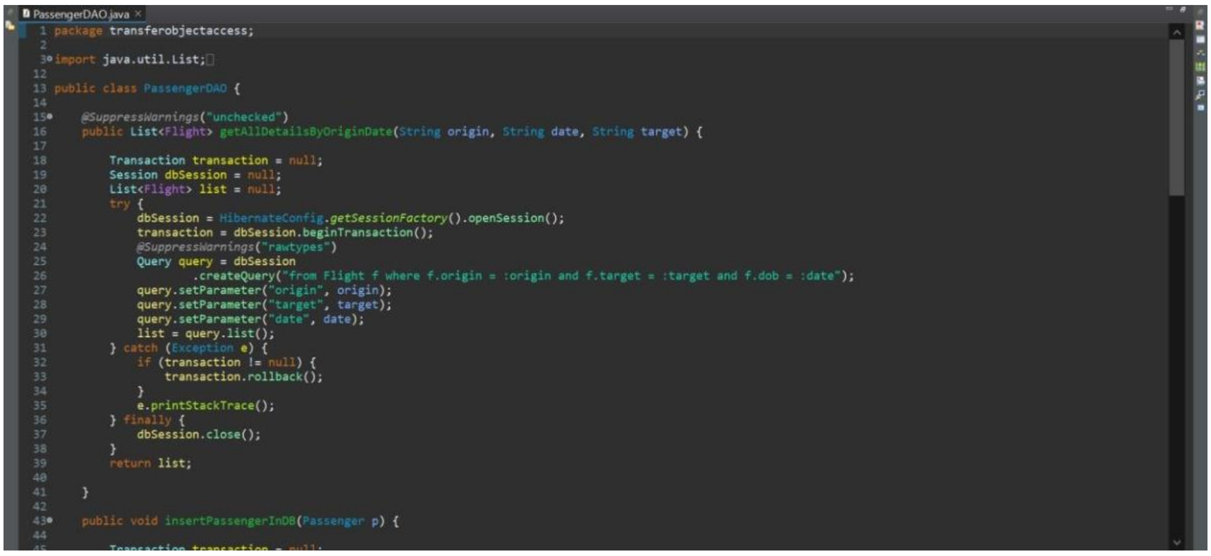
c> Password.java



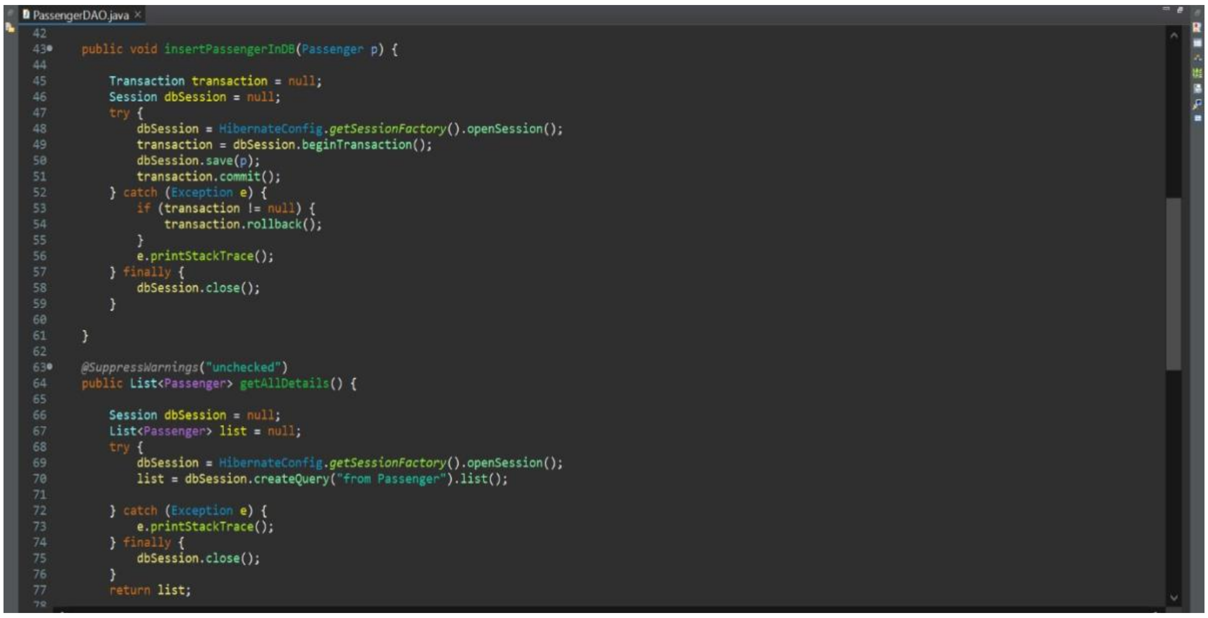
```java
package models;

public class Password {

    private static String pwd;

    public Password() {
        pwd = "admin";
    }

    public static String getPwd() {
        return pwd;
    }

    public static void setPwd(String pwd) {
        Password.pwd = pwd;
    }

}
```

3: Data Access Objects

a> PassengerDAO.java

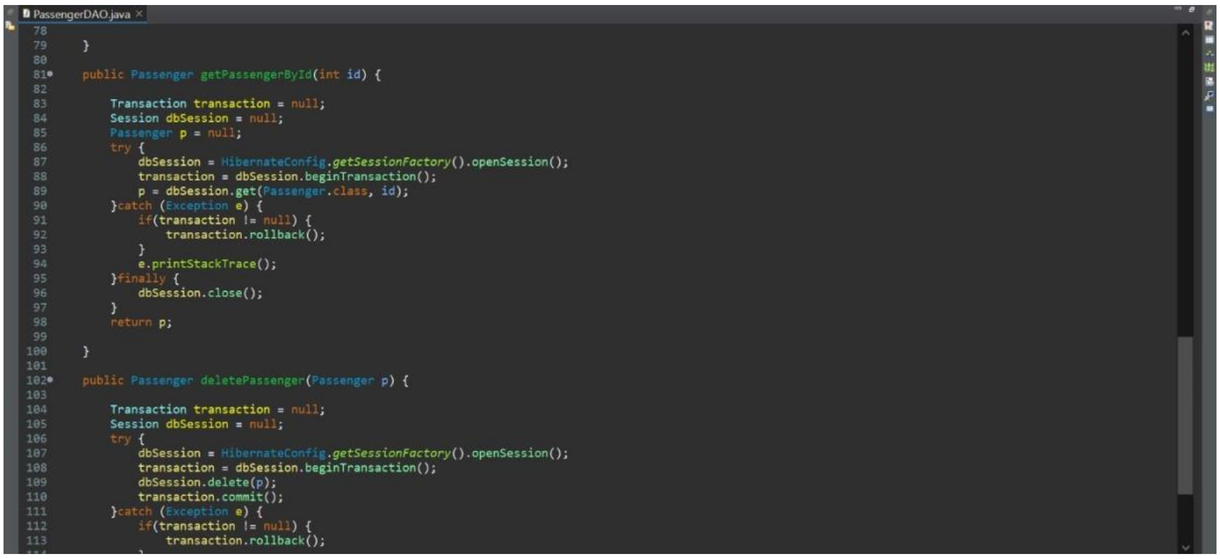


```java
package transferobjectaccess;

import java.util.List;

public class PassengerDAO {

    @SuppressWarnings("unchecked")
    public List<Flight> getAllDetailsByOriginDate(String origin, String date, String target) {

        Transaction transaction = null;
        Session dbSession = null;
        List<Flight> list = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            @SuppressWarnings("rawtypes")
            Query query = dbSession
                    .createQuery("from Flight f where f.origin = :origin and f.target = :target and f.dob = :date");
            query.setParameter("origin", origin);
            query.setParameter("target", target);
            query.setParameter("date", date);
            list = query.list();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();

            }
            e.printStackTrace();
        } finally {
            dbSession.close();
        }
        return list;

    }

    public void insertPassengerInDB(Passenger p) {

        Transaction transaction = null;
```



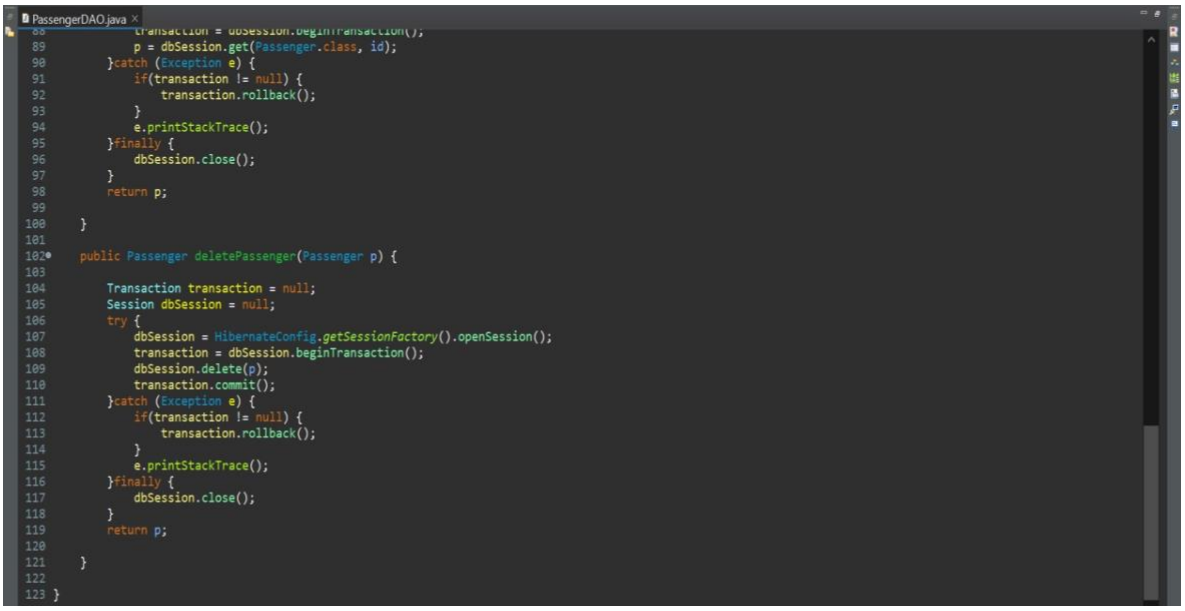
PassengerDAO.java

```java

    public void insertPassengerInDB(Passenger p) {

        Transaction transaction = null;
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            dbSession.save(p);
            transaction.commit();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        } finally {
            dbSession.close();
        }

    }

    @SuppressWarnings("unchecked")
    public List<Passenger> getAllDetails() {

        Session dbSession = null;
        List<Passenger> list = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            list = dbSession.createQuery("from Passenger").list();

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            dbSession.close();
        }
        return list;
```



PassengerDAO.java

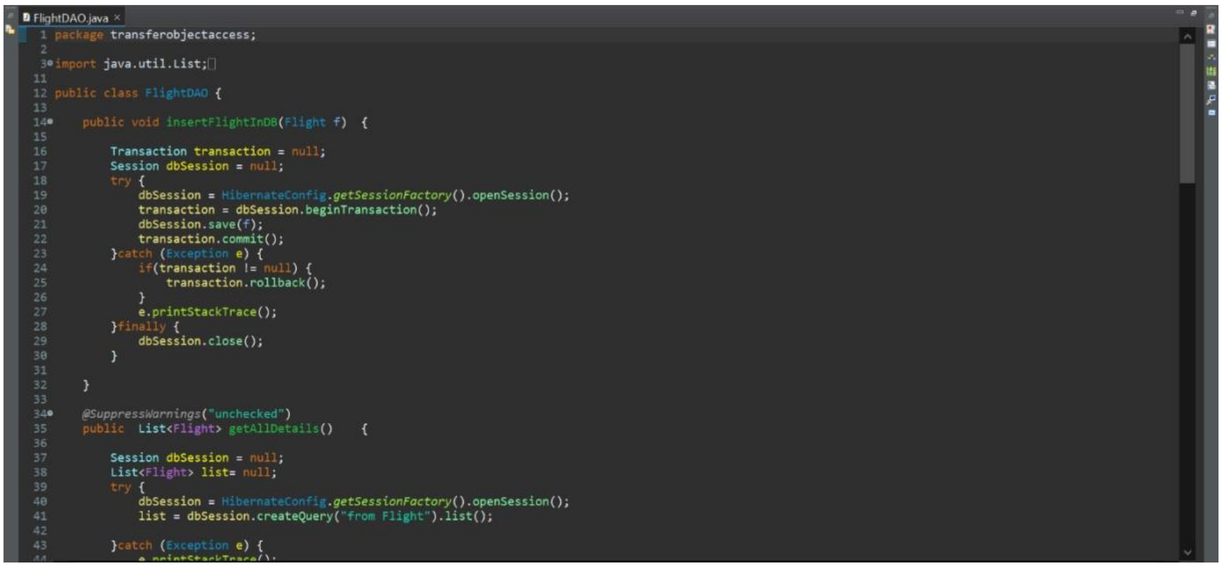
```java

    }

    public Passenger getPassengerById(int id) {

        Transaction transaction = null;
        Session dbSession = null;
        Passenger p = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            p = dbSession.get(Passenger.class, id);
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return p;

    }

    public Passenger deletePassenger(Passenger p) {

        Transaction transaction = null;
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            dbSession.delete(p);
            transaction.commit();
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
```



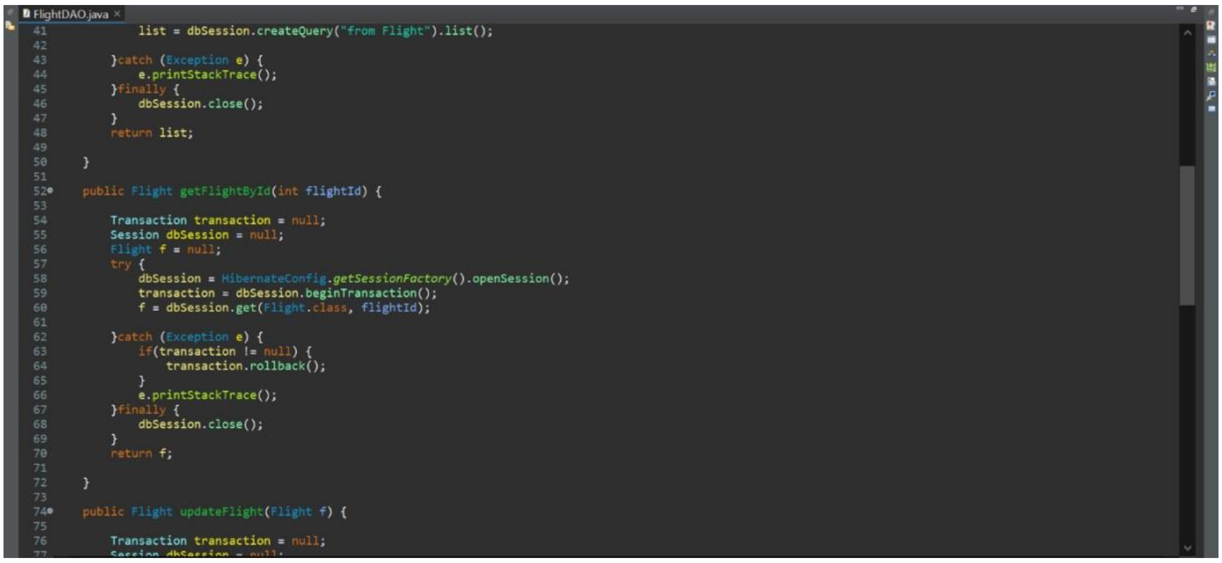
PassengerDAO.java

```java
            transaction = dbSession.beginTransaction();
            p = dbSession.get(Passenger.class, id);
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return p;

    }

    public Passenger deletePassenger(Passenger p) {

        Transaction transaction = null;
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            dbSession.delete(p);
            transaction.commit();
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return p;

    }

}
```

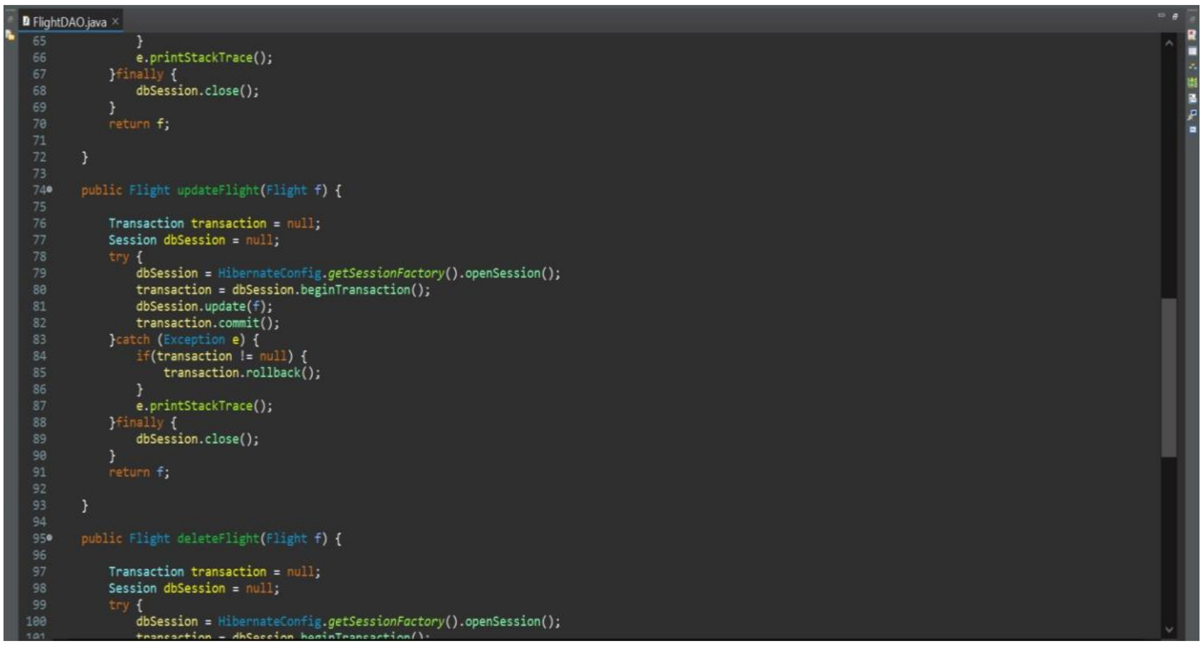
b> FlightDAO.java



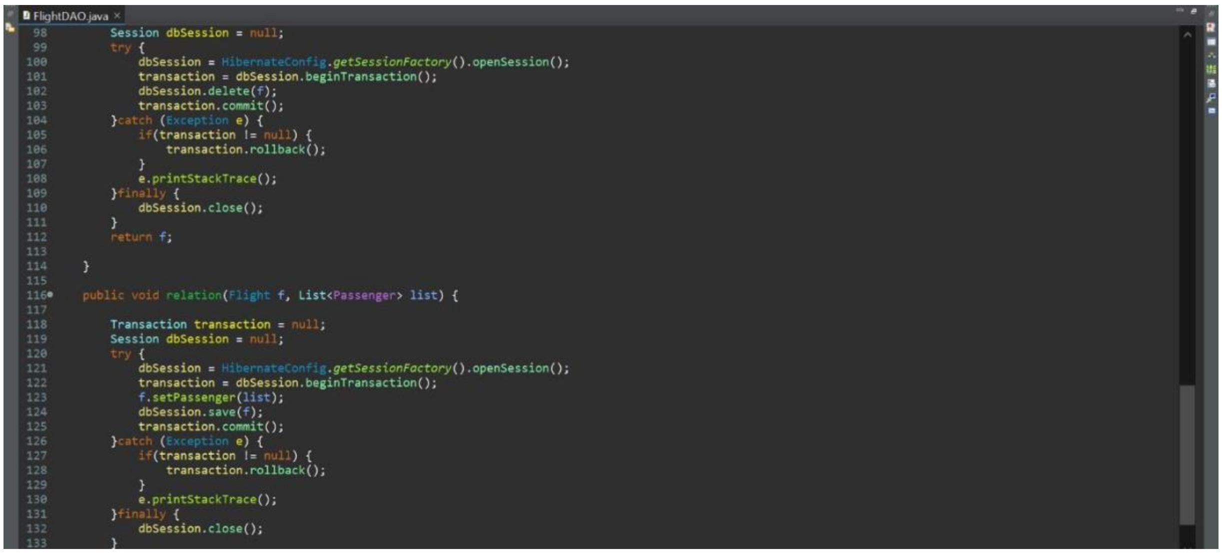
```java
package transferobjectaccess;

import java.util.List;

public class FlightDAO {

    public void insertFlightInDB(Flight f)  {

        Transaction transaction = null;
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            dbSession.save(f);
            transaction.commit();
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }

    }

    @SuppressWarnings("unchecked")
    public  List<Flight> getAllDetails()    {

        Session dbSession = null;
        List<Flight> list= null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            list = dbSession.createQuery("from Flight").list();

        }catch (Exception e) {
```



```java
            list = dbSession.createQuery("from Flight").list();

        }catch (Exception e) {
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return list;

    }

    public Flight getFlightById(int flightId) {

        Transaction transaction = null;
        Session dbSession = null;
        Flight f = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            f = dbSession.get(Flight.class, flightId);

        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return f;

    }

    public Flight updateFlight(Flight f) {

        Transaction transaction = null;
```



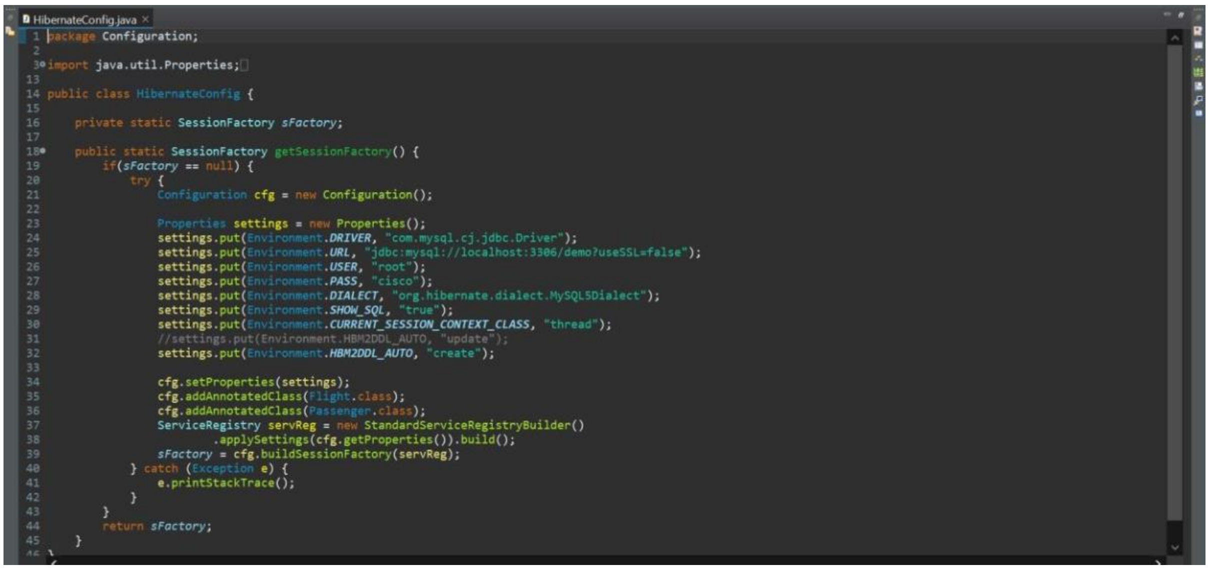
```java
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return f;

    }

    public Flight updateFlight(Flight f) {

        Transaction transaction = null;
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            dbSession.update(f);
            transaction.commit();
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return f;

    }

    public Flight deleteFlight(Flight f) {

        Transaction transaction = null;
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
```



FlightDAO.java ×

```java
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            dbSession.delete(f);
            transaction.commit();
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
        return f;

    }

    public void relation(Flight f, List<Passenger> list) {

        Transaction transaction = null;
        Session dbSession = null;
        try {
            dbSession = HibernateConfig.getSessionFactory().openSession();
            transaction = dbSession.beginTransaction();
            f.setPassenger(list);
            dbSession.save(f);
            transaction.commit();
        }catch (Exception e) {
            if(transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        }finally {
            dbSession.close();
        }
```

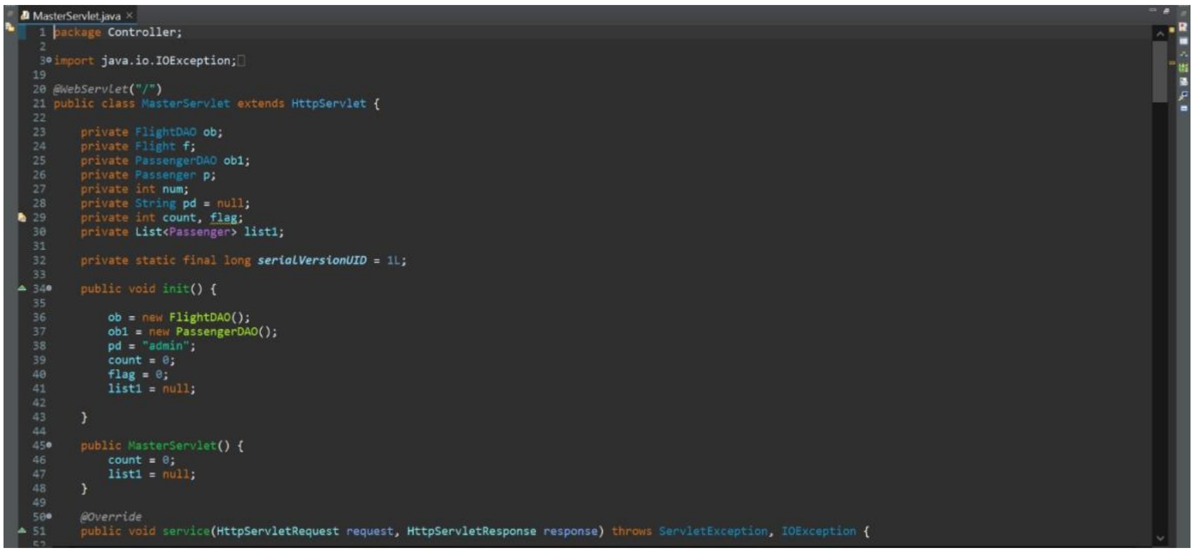
4: Hibernate Integration (HibernateConfig.java)



HibernateConfig.java ×

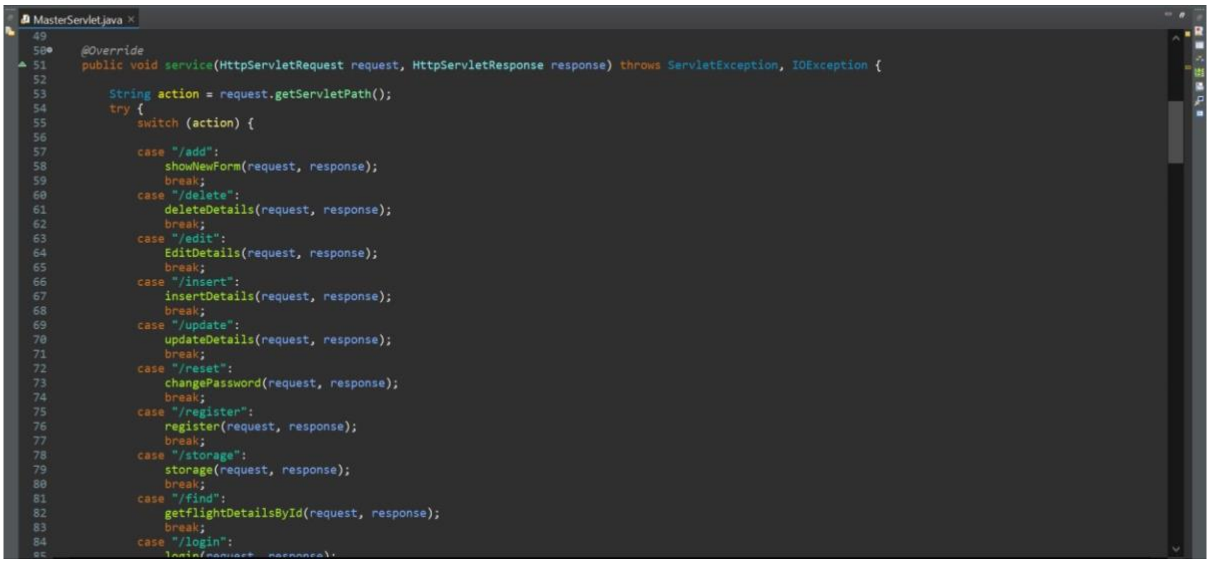
```java
package Configuration;

import java.util.Properties;

public class HibernateConfig {

    private static SessionFactory sFactory;

    public static SessionFactory getSessionFactory() {
        if(sFactory == null) {
            try {
                Configuration cfg = new Configuration();

                Properties settings = new Properties();
                settings.put(Environment.DRIVER, "com.mysql.cj.jdbc.Driver");
                settings.put(Environment.URL, "jdbc:mysql://localhost:3306/demo?useSSL=false");
                settings.put(Environment.USER, "root");
                settings.put(Environment.PASS, "cisco");
                settings.put(Environment.DIALECT, "org.hibernate.dialect.MySQL5Dialect");
                settings.put(Environment.SHOW_SQL, "true");
                settings.put(Environment.CURRENT_SESSION_CONTEXT_CLASS, "thread");
                //settings.put(Environment.HBM2DDL_AUTO, "update");
                settings.put(Environment.HBM2DDL_AUTO, "create");

                cfg.setProperties(settings);
                cfg.addAnnotatedClass(Flight.class);
                cfg.addAnnotatedClass(Passenger.class);
                ServiceRegistry servReg = new StandardServiceRegistryBuilder()
                        .applySettings(cfg.getProperties()).build();
                sFactory = cfg.buildSessionFactory(servReg);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return sFactory;
    }
```

5: Controller (MasterServlet.java)



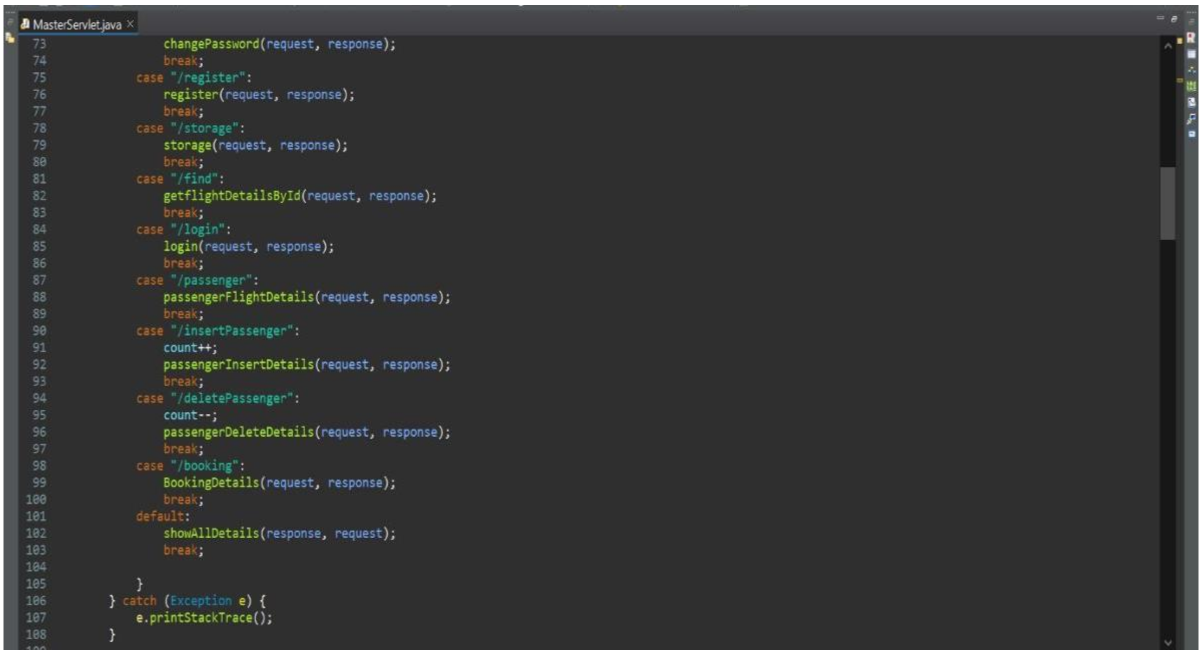
MasterServlet.java ×

```java
package Controller;

import java.io.IOException;

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

    private FlightDAO ob;
    private Flight f;
    private PassengerDAO ob1;
    private Passenger p;
    private int num;
    private String pd = null;
    private int count, flag;
    private List<Passenger> list1;

    private static final long serialVersionUID = 1L;

    public void init() {

        ob = new FlightDAO();
        ob1 = new PassengerDAO();
        pd = "admin";
        count = 0;
        flag = 0;
        list1 = null;

    }

    public MasterServlet() {
        count = 0;
        list1 = null;
    }

    @Override
    public void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```



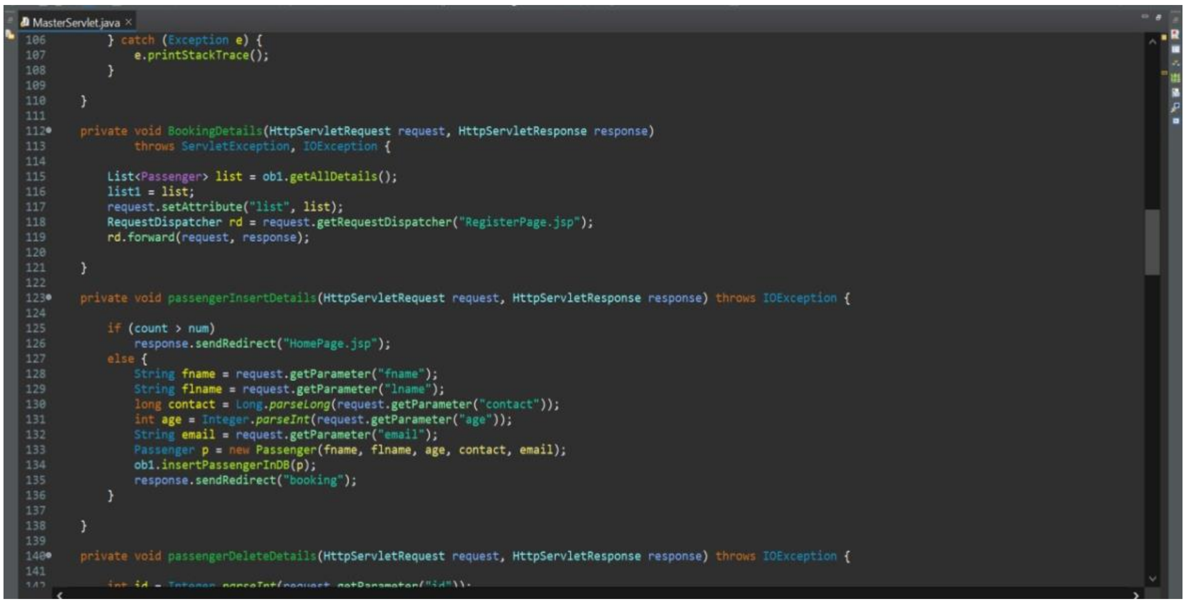
MasterServlet.java

```java

    @Override
    public void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String action = request.getServletPath();
        try {
            switch (action) {

            case "/add":
                showNewForm(request, response);
                break;
            case "/delete":
                deleteDetails(request, response);
                break;
            case "/edit":
                EditDetails(request, response);
                break;
            case "/insert":
                insertDetails(request, response);
                break;
            case "/update":
                updateDetails(request, response);
                break;
            case "/reset":
                changePassword(request, response);
                break;
            case "/register":
                register(request, response);
                break;
            case "/storage":
                storage(request, response);
                break;
            case "/find":
                getflightDetailsById(request, response);
                break;
            case "/login":
```



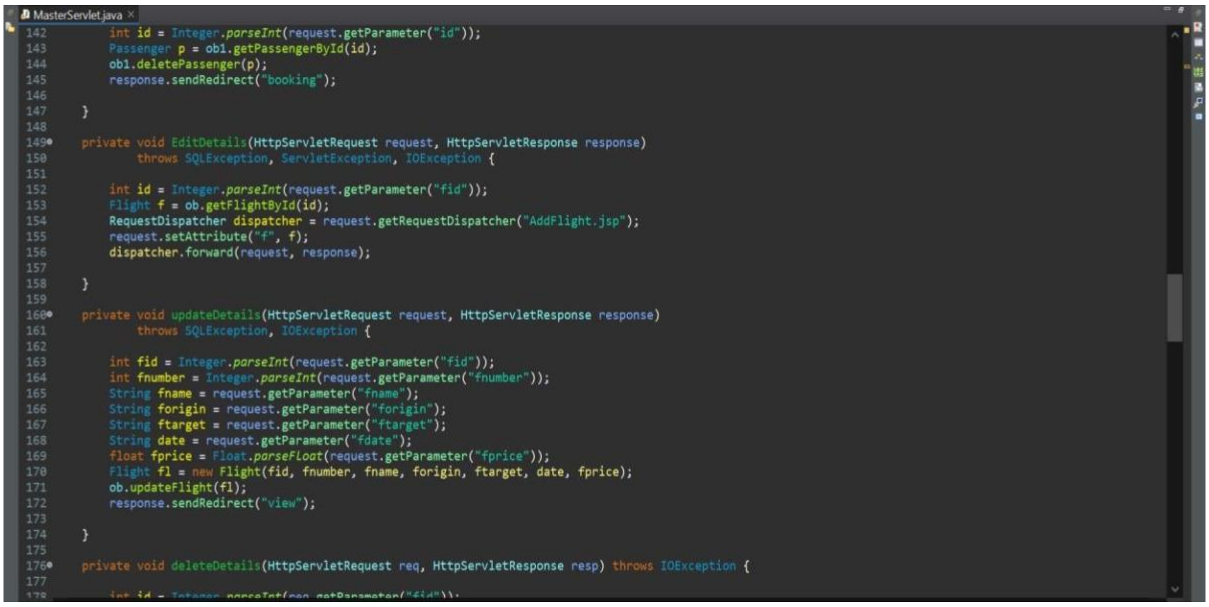
MasterServlet.java

```java
                changePassword(request, response);
                break;
            case "/register":
                register(request, response);
                break;
            case "/storage":
                storage(request, response);
                break;
            case "/find":
                getflightDetailsById(request, response);
                break;
            case "/login":
                login(request, response);
                break;
            case "/passenger":
                passengerFlightDetails(request, response);
                break;
            case "/insertPassenger":
                count++;
                passengerInsertDetails(request, response);
                break;
            case "/deletePassenger":
                count--;
                passengerDeleteDetails(request, response);
                break;
            case "/booking":
                BookingDetails(request, response);
                break;
            default:
                showAllDetails(response, request);
                break;

            }
        } catch (Exception e) {
            e.printStackTrace();
        }
```



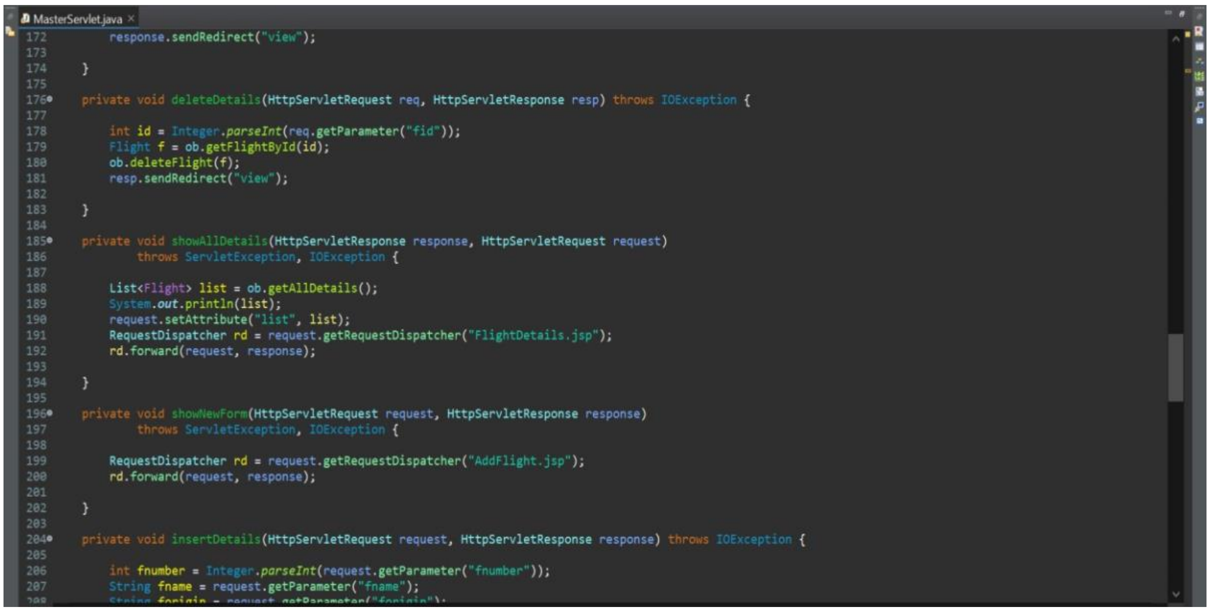
MasterServlet.java

```java
        } catch (Exception e) {
            e.printStackTrace();
        }

    }

    private void BookingDetails(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        List<Passenger> list = ob1.getAllDetails();
        list1 = list;
        request.setAttribute("list", list);
        RequestDispatcher rd = request.getRequestDispatcher("RegisterPage.jsp");
        rd.forward(request, response);

    }

    private void passengerInsertDetails(HttpServletRequest request, HttpServletResponse response) throws IOException {

        if (count > num)
            response.sendRedirect("HomePage.jsp");
        else {
            String fname = request.getParameter("fname");
            String flname = request.getParameter("lname");
            long contact = Long.parseLong(request.getParameter("contact"));
            int age = Integer.parseInt(request.getParameter("age"));
            String email = request.getParameter("email");
            Passenger p = new Passenger(fname, flname, age, contact, email);
            ob1.insertPassengerInDB(p);
            response.sendRedirect("booking");
        }

    }

    private void passengerDeleteDetails(HttpServletRequest request, HttpServletResponse response) throws IOException {

```



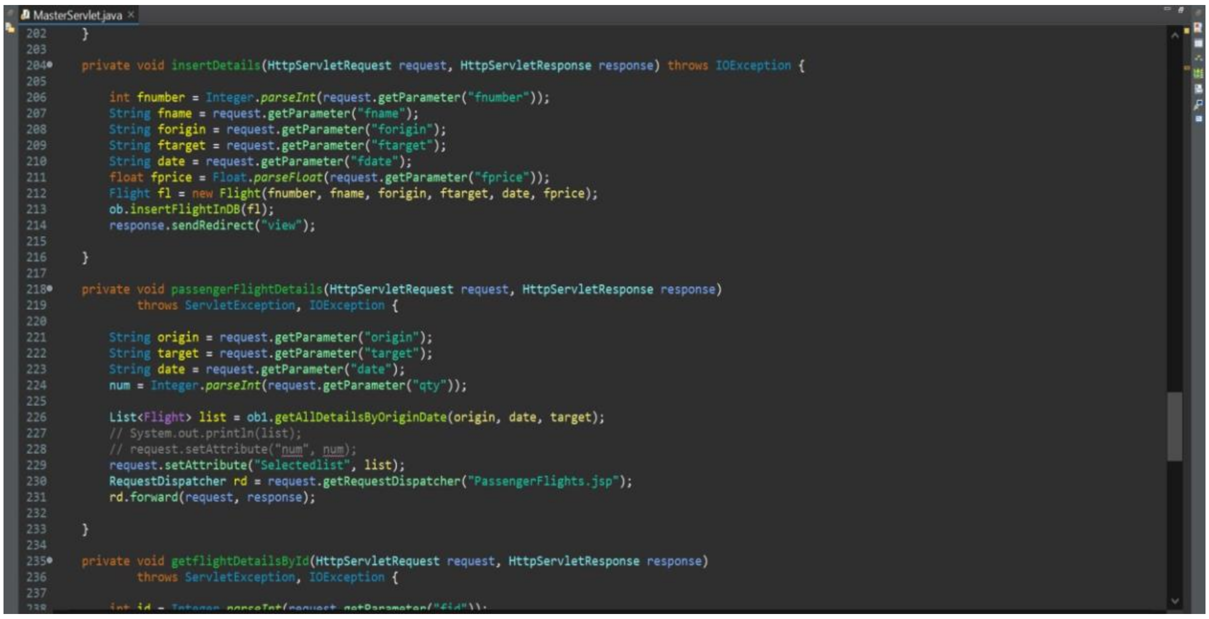
MasterServlet.java

```java
        int id = Integer.parseInt(request.getParameter("id"));
        Passenger p = ob1.getPassengerById(id);
        ob1.deletePassenger(p);
        response.sendRedirect("booking");

    }

    private void EditDetails(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, ServletException, IOException {

        int id = Integer.parseInt(request.getParameter("fid"));
        Flight f = ob.getFlightById(id);
        RequestDispatcher dispatcher = request.getRequestDispatcher("AddFlight.jsp");
        request.setAttribute("f", f);
        dispatcher.forward(request, response);

    }

    private void updateDetails(HttpServletRequest request, HttpServletResponse response)
            throws SQLException, IOException {

        int fid = Integer.parseInt(request.getParameter("fid"));
        int fnumber = Integer.parseInt(request.getParameter("fnumber"));
        String fname = request.getParameter("fname");
        String forigin = request.getParameter("forigin");
        String ftarget = request.getParameter("ftarget");
        String date = request.getParameter("fdate");
        float fprice = Float.parseFloat(request.getParameter("fprice"));
        Flight fl = new Flight(fid, fnumber, fname, forigin, ftarget, date, fprice);
        ob.updateFlight(fl);
        response.sendRedirect("view");

    }

    private void deleteDetails(HttpServletRequest req, HttpServletResponse resp) throws IOException {

```



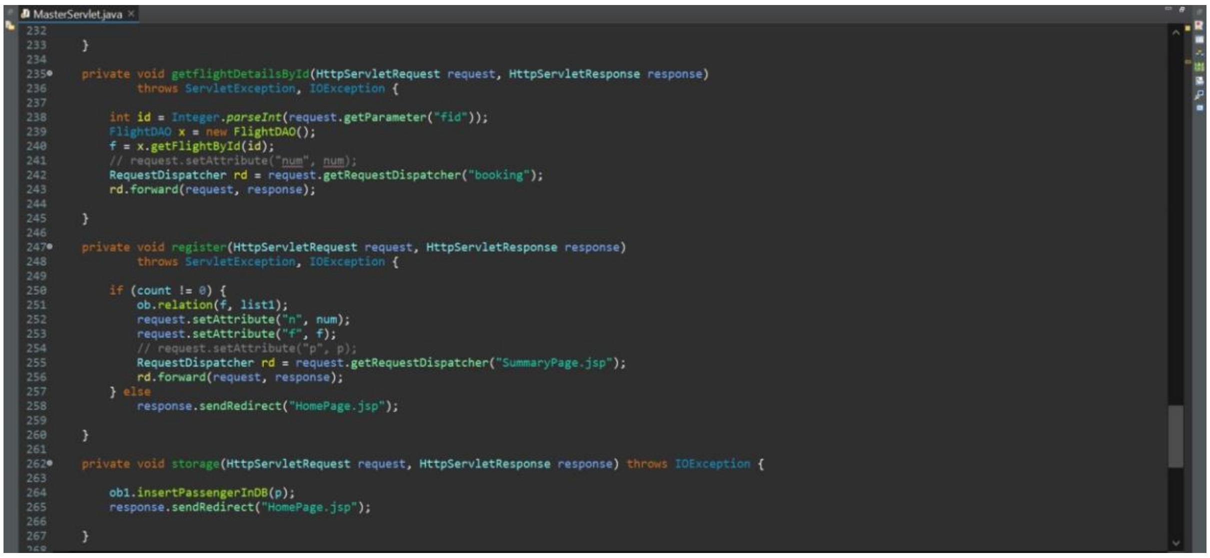
MasterServlet.java

```java
        response.sendRedirect("view");

    }

    private void deleteDetails(HttpServletRequest req, HttpServletResponse resp) throws IOException {

        int id = Integer.parseInt(req.getParameter("fid"));
        Flight f = ob.getFlightById(id);
        ob.deleteFlight(f);
        resp.sendRedirect("view");

    }

    private void showAllDetails(HttpServletResponse response, HttpServletRequest request)
            throws ServletException, IOException {

        List<Flight> list = ob.getAllDetails();
        System.out.println(list);
        request.setAttribute("list", list);
        RequestDispatcher rd = request.getRequestDispatcher("FlightDetails.jsp");
        rd.forward(request, response);

    }

    private void showNewForm(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        RequestDispatcher rd = request.getRequestDispatcher("AddFlight.jsp");
        rd.forward(request, response);

    }

    private void insertDetails(HttpServletRequest request, HttpServletResponse response) throws IOException {

        int fnumber = Integer.parseInt(request.getParameter("fnumber"));
        String fname = request.getParameter("fname");
```

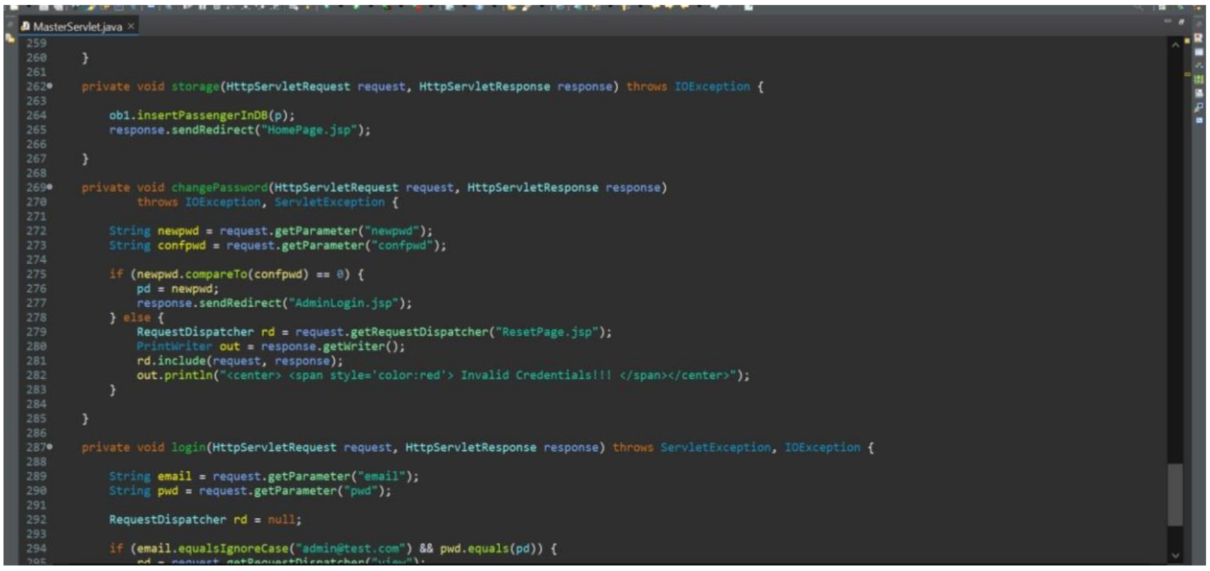


MasterServlet.java

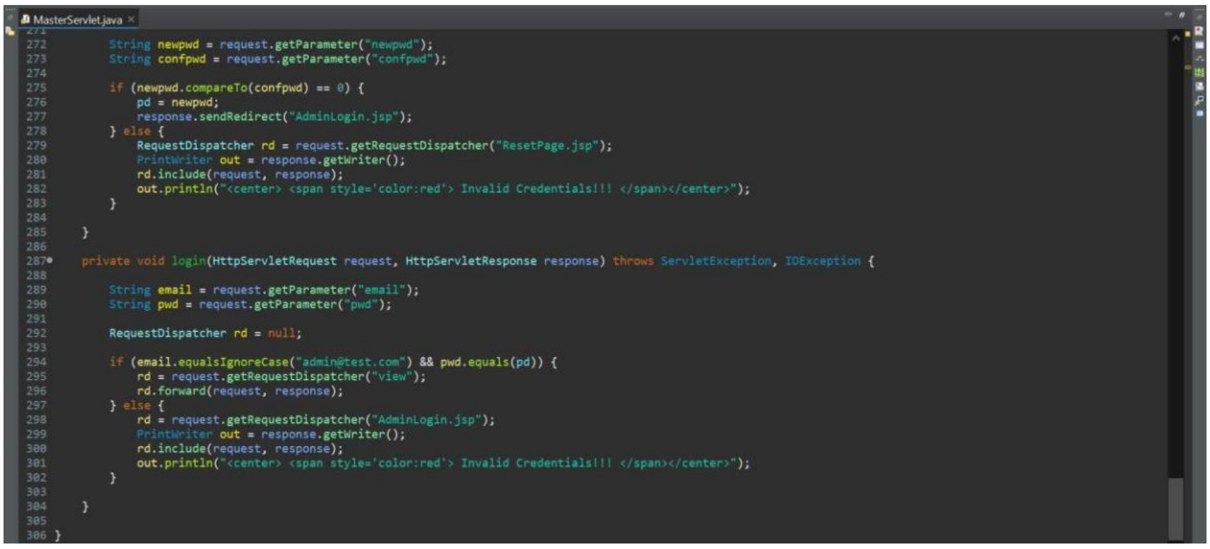
```java
    }

    private void insertDetails(HttpServletRequest request, HttpServletResponse response) throws IOException {

        int fnumber = Integer.parseInt(request.getParameter("fnumber"));
        String fname = request.getParameter("fname");
        String forigin = request.getParameter("forigin");
        String ftarget = request.getParameter("ftarget");
        String date = request.getParameter("fdate");
        float fprice = Float.parseFloat(request.getParameter("fprice"));
        Flight fl = new Flight(fnumber, fname, forigin, ftarget, date, fprice);
        ob.insertFlightInDB(fl);
        response.sendRedirect("view");

    }

    private void passengerFlightDetails(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        String origin = request.getParameter("origin");
        String target = request.getParameter("target");
        String date = request.getParameter("date");
        num = Integer.parseInt(request.getParameter("qty"));

        List<Flight> list = ob1.getAllDetailsByOriginDate(origin, date, target);
        // System.out.println(list);
        // request.setAttribute("num", num);
        request.setAttribute("Selectedlist", list);
        RequestDispatcher rd = request.getRequestDispatcher("PassengerFlights.jsp");
        rd.forward(request, response);

    }

    private void getflightDetailsById(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

```



```java
    }

    private void getflightDetailsById(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        int id = Integer.parseInt(request.getParameter("fid"));
        FlightDAO x = new FlightDAO();
        f = x.getFlightById(id);
        // request.setAttribute("num", num);
        RequestDispatcher rd = request.getRequestDispatcher("booking");
        rd.forward(request, response);

    }

    private void register(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        if (count != 0) {
            ob.relation(f, list1);
            request.setAttribute("n", num);
            request.setAttribute("f", f);
            // request.setAttribute("p", p);
            RequestDispatcher rd = request.getRequestDispatcher("SummaryPage.jsp");
            rd.forward(request, response);
        } else
            response.sendRedirect("HomePage.jsp");

    }

    private void storage(HttpServletRequest request, HttpServletResponse response) throws IOException {

        ob1.insertPassengerInDB(p);
        response.sendRedirect("HomePage.jsp");

    }
```

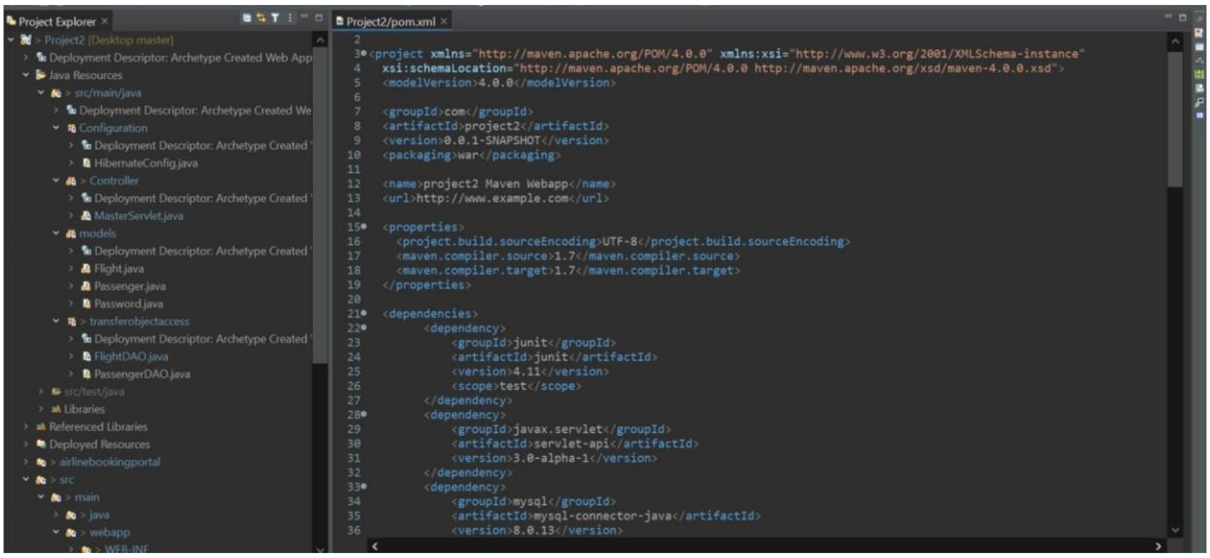


```java
    }

    private void storage(HttpServletRequest request, HttpServletResponse response) throws IOException {

        ob1.insertPassengerInDB(p);
        response.sendRedirect("HomePage.jsp");

    }

    private void changePassword(HttpServletRequest request, HttpServletResponse response)
            throws IOException, ServletException {

        String newpwd = request.getParameter("newpwd");
        String confpwd = request.getParameter("confpwd");

        if (newpwd.compareTo(confpwd) == 0) {
            pd = newpwd;
            response.sendRedirect("AdminLogin.jsp");
        } else {
            RequestDispatcher rd = request.getRequestDispatcher("ResetPage.jsp");
            PrintWriter out = response.getWriter();
            rd.include(request, response);
            out.println("<center> <span style='color:red'> Invalid Credentials!!! </span></center>");
        }

    }

    private void login(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String email = request.getParameter("email");
        String pwd = request.getParameter("pwd");

        RequestDispatcher rd = null;

        if (email.equalsIgnoreCase("admin@test.com") && pwd.equals(pd)) {
```



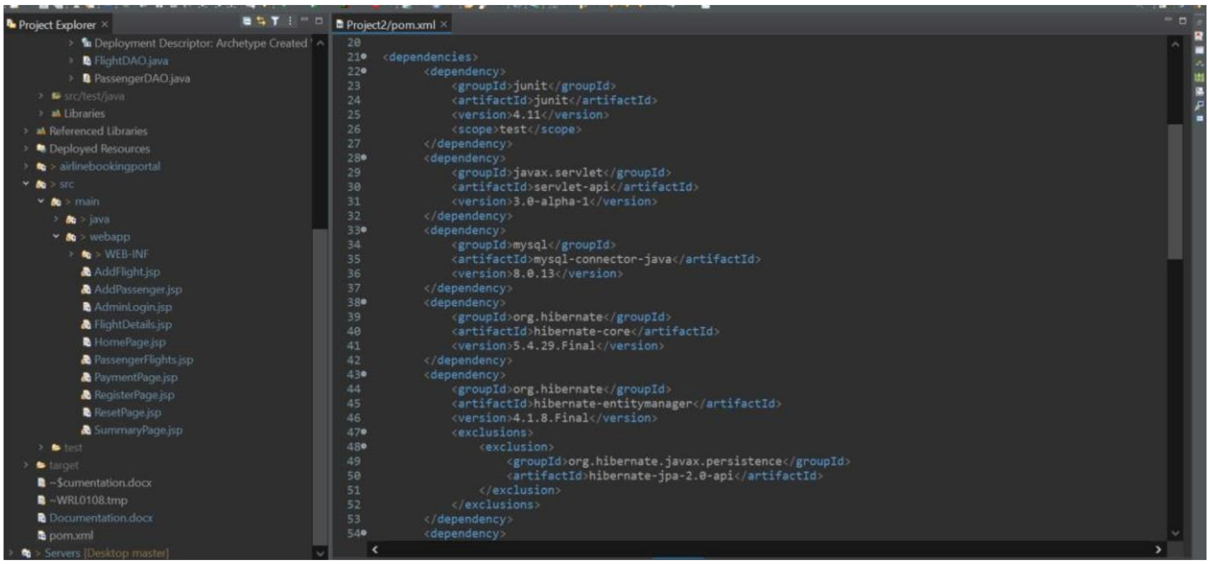
```java
        String newpwd = request.getParameter("newpwd");
        String confpwd = request.getParameter("confpwd");

        if (newpwd.compareTo(confpwd) == 0) {
            pd = newpwd;
            response.sendRedirect("AdminLogin.jsp");
        } else {
            RequestDispatcher rd = request.getRequestDispatcher("ResetPage.jsp");
            PrintWriter out = response.getWriter();
            rd.include(request, response);
            out.println("<center> <span style='color:red'> Invalid Credentials!!! </span></center>");
        }

    }

    private void login(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String email = request.getParameter("email");
        String pwd = request.getParameter("pwd");

        RequestDispatcher rd = null;

        if (email.equalsIgnoreCase("admin@test.com") && pwd.equals(pd)) {
            rd = request.getRequestDispatcher("view");
            rd.forward(request, response);
        } else {
            rd = request.getRequestDispatcher("AdminLogin.jsp");
            PrintWriter out = response.getWriter();
            rd.include(request, response);
            out.println("<center> <span style='color:red'> Invalid Credentials!!! </span></center>");
        }

    }
}
```

6: Program Structure and pom.xml



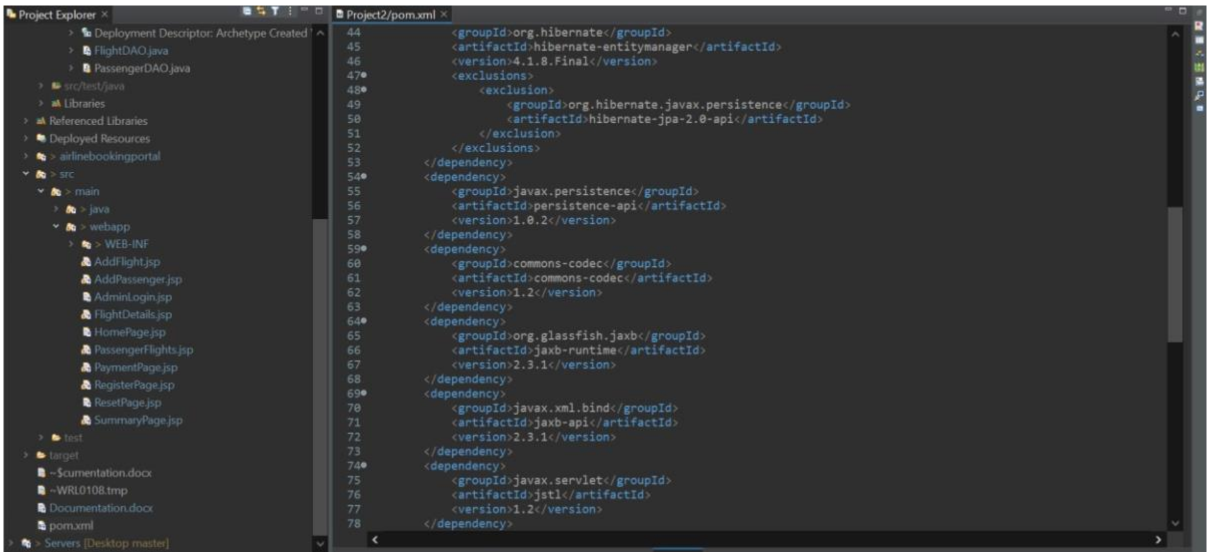
```
Project Explorer
Project2 [Desktop master]
  Deployment Descriptor: Archetype Created Web App
  Java Resources
    src/main/java
      Deployment Descriptor: Archetype Created We
      Configuration
        Deployment Descriptor: Archetype Created '
        HibernateConfig.java
      Controller
        Deployment Descriptor: Archetype Created '
        MasterServlet.java
      models
        Deployment Descriptor: Archetype Created '
        Flight.java
        Passenger.java
        Password.java
      transferobjectaccess
        Deployment Descriptor: Archetype Created '
        FlightDAO.java
        PassengerDAO.java
    src/test/java
    Libraries
  Referenced Libraries
  Deployed Resources
  airlinebookingportal
  src
    main
      java
      webapp
        WEB-INF
```

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com</groupId>
  <artifactId>project2</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>project2 Maven Webapp</name>
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>servlet-api</artifactId>
            <version>3.0-alpha-1</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.13</version>
```



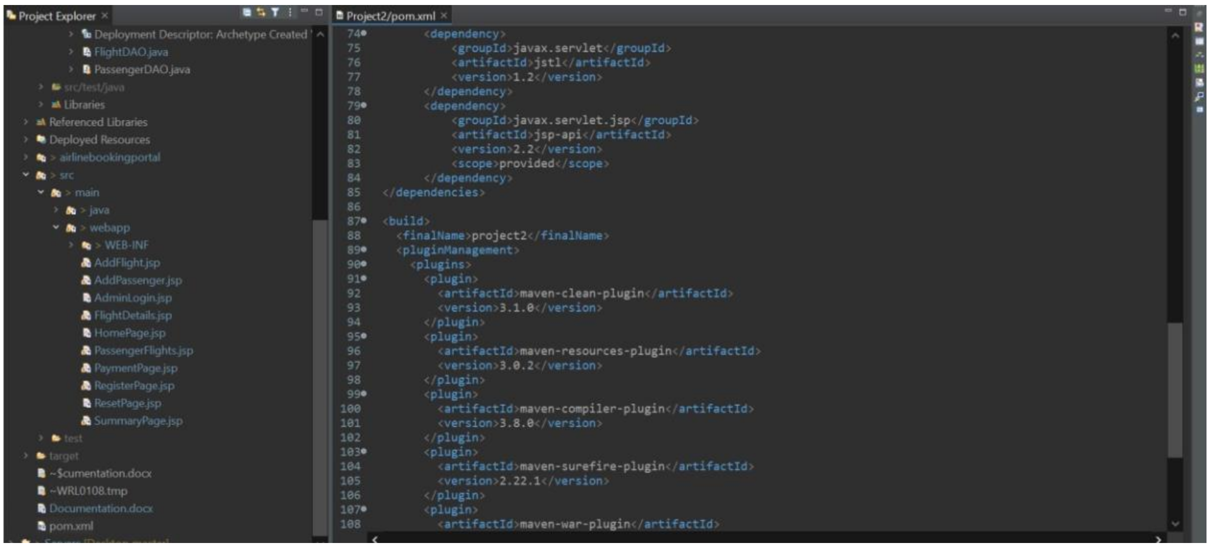
```
Project Explorer
        Deployment Descriptor: Archetype Created '
        FlightDAO.java
        PassengerDAO.java
    src/test/java
    Libraries
  Referenced Libraries
  Deployed Resources
  airlinebookingportal
  src
    main
      java
      webapp
        WEB-INF
        AddFlight.jsp
        AddPassenger.jsp
        AdminLogin.jsp
        FlightDetails.jsp
        HomePage.jsp
        PassengerFlights.jsp
        PaymentPage.jsp
        RegisterPage.jsp
        ResetPage.jsp
        SummaryPage.jsp
    test
  target
  ~$cumentation.docx
  ~WRL0108.tmp
  Documentation.docx
  pom.xml
Servers [Desktop master]
```

```xml

  <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.11</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>servlet-api</artifactId>
            <version>3.0-alpha-1</version>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.13</version>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>5.4.29.Final</version>
        </dependency>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-entitymanager</artifactId>
            <version>4.1.8.Final</version>
            <exclusions>
                <exclusion>
                    <groupId>org.hibernate.javax.persistence</groupId>
                    <artifactId>hibernate-jpa-2.0-api</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
```



```
Project Explorer
        Deployment Descriptor: Archetype Created '
        FlightDAO.java
        PassengerDAO.java
    src/test/java
    Libraries
  Referenced Libraries
  Deployed Resources
  airlinebookingportal
  src
    main
      java
      webapp
        WEB-INF
        AddFlight.jsp
        AddPassenger.jsp
        AdminLogin.jsp
        FlightDetails.jsp
        HomePage.jsp
        PassengerFlights.jsp
        PaymentPage.jsp
        RegisterPage.jsp
        ResetPage.jsp
        SummaryPage.jsp
    test
  target
  ~$cumentation.docx
  ~WRL0108.tmp
  Documentation.docx
  pom.xml
Servers [Desktop master]
```

```xml
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-entitymanager</artifactId>
            <version>4.1.8.Final</version>
            <exclusions>
                <exclusion>
                    <groupId>org.hibernate.javax.persistence</groupId>
                    <artifactId>hibernate-jpa-2.0-api</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>javax.persistence</groupId>
            <artifactId>persistence-api</artifactId>
            <version>1.0.2</version>
        </dependency>
        <dependency>
            <groupId>commons-codec</groupId>
            <artifactId>commons-codec</artifactId>
            <version>1.2</version>
        </dependency>
        <dependency>
            <groupId>org.glassfish.jaxb</groupId>
            <artifactId>jaxb-runtime</artifactId>
            <version>2.3.1</version>
        </dependency>
        <dependency>
            <groupId>javax.xml.bind</groupId>
            <artifactId>jaxb-api</artifactId>
            <version>2.3.1</version>
        </dependency>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
        </dependency>
```

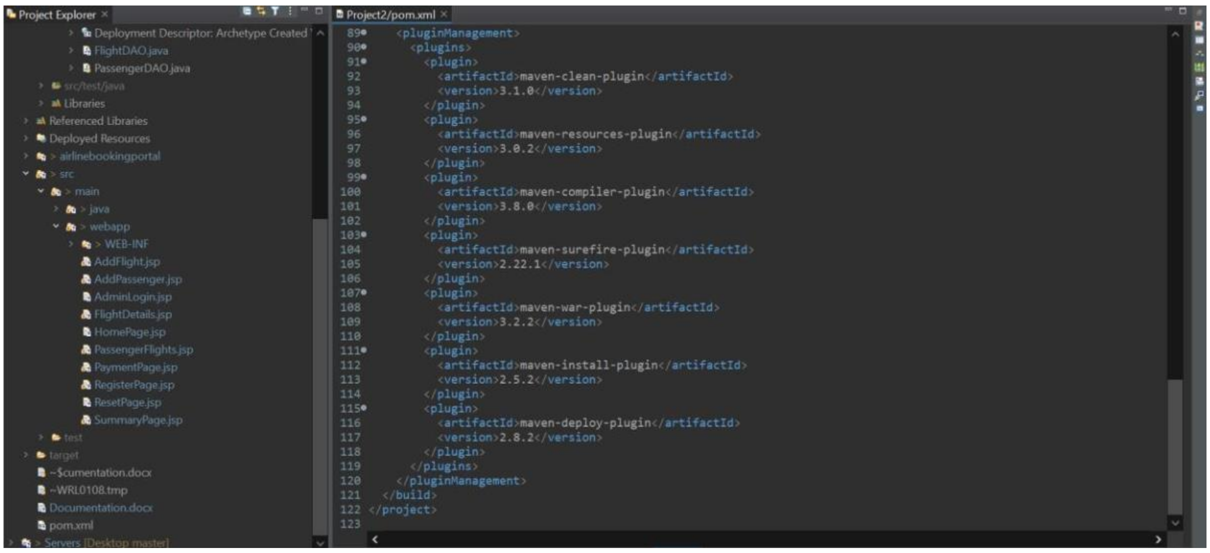


Project Explorer

- Deployment Descriptor: Archetype Created
- FlightDAO.java
- PassengerDAO.java
- src/test/java
- Libraries
- Referenced Libraries
- Deployed Resources
- > airlinebookingportal
- > src
- > main
- > java
- > webapp
- > WEB-INF
- AddFlight.jsp
- AddPassenger.jsp
- AdminLogin.jsp
- FlightDetails.jsp
- HomePage.jsp
- PassengerFlights.jsp
- PaymentPage.jsp
- RegisterPage.jsp
- ResetPage.jsp
- SummaryPage.jsp
- test
- target
- ~$cumentation.docx
- ~WRL0108.tmp
- Documentation.docx
- pom.xml

Project2/pom.xml

```xml
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
        </dependency>
        <dependency>
            <groupId>javax.servlet.jsp</groupId>
            <artifactId>jsp-api</artifactId>
            <version>2.2</version>
            <scope>provided</scope>
        </dependency>
  </dependencies>

  <build>
    <finalName>project2</finalName>
    <pluginManagement>
      <plugins>
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-war-plugin</artifactId>
```



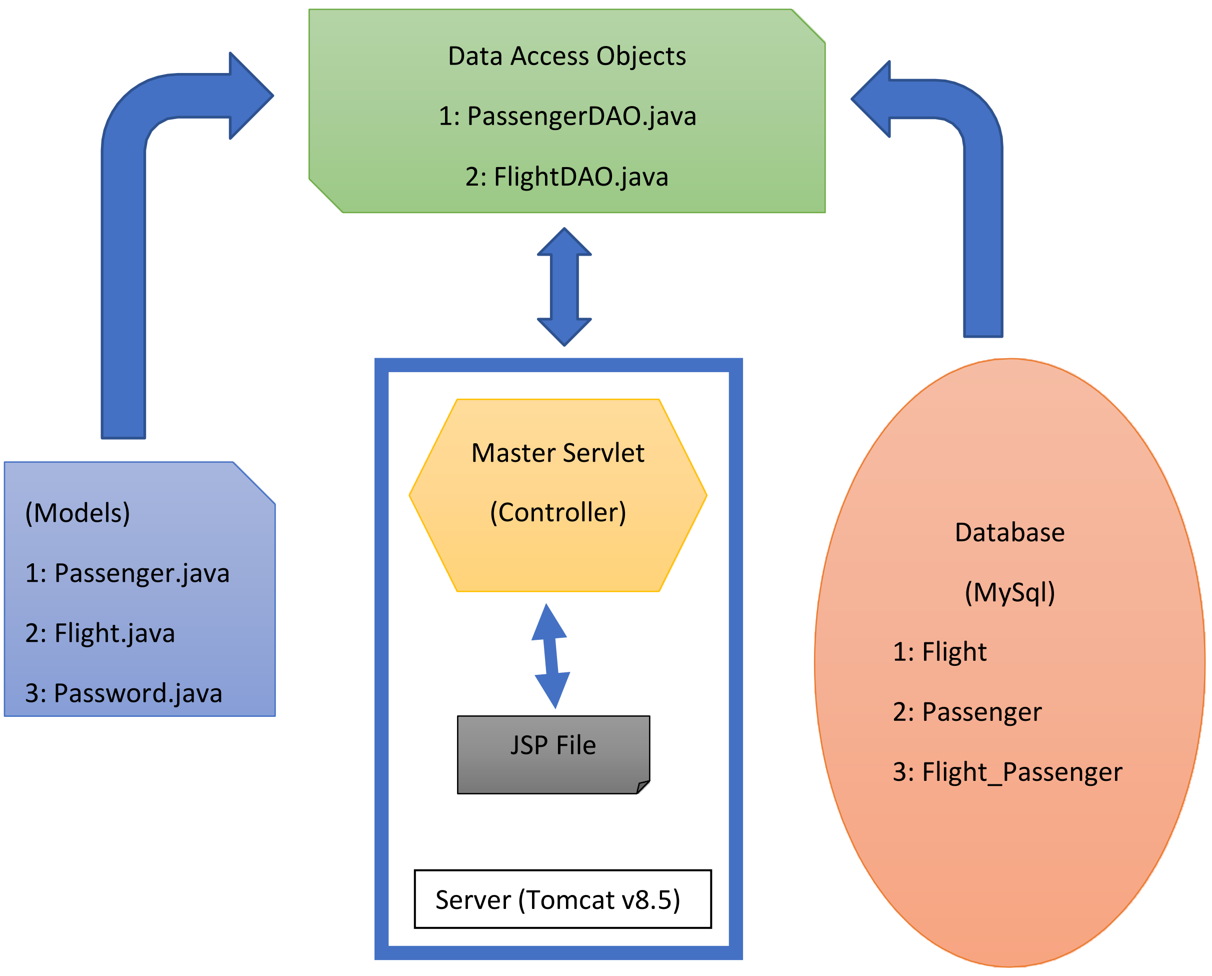
Project Explorer

- Deployment Descriptor: Archetype Created
- FlightDAO.java
- PassengerDAO.java
- src/test/java
- Libraries
- Referenced Libraries
- Deployed Resources
- > airlinebookingportal
- > src
- > main
- > java
- > webapp
- > WEB-INF
- AddFlight.jsp
- AddPassenger.jsp
- AdminLogin.jsp
- FlightDetails.jsp
- HomePage.jsp
- PassengerFlights.jsp
- PaymentPage.jsp
- RegisterPage.jsp
- ResetPage.jsp
- SummaryPage.jsp
- test
- target
- ~$cumentation.docx
- ~WRL0108.tmp
- Documentation.docx
- pom.xml
- > Servers [Desktop master]

Project2/pom.xml

```xml
    <pluginManagement>
      <plugins>
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-war-plugin</artifactId>
          <version>3.2.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-install-plugin</artifactId>
          <version>2.5.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-deploy-plugin</artifactId>
          <version>2.8.2</version>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>
</project>
```

Flow Diagram



- Core Concepts used in this project are mostly maven, hibernate, MySQL, jdbc connector, associations, java, CRUD operations in a database, web development.

Algorithm

Step 1> Start.

Step 2> Two options in the home page:

Case 1: If user select “admin” section then go to step 3.

Case 2: If user goes to passenger side through registering all booking details, then go step 7.

Step 3> Once a user select admin, it will prompt for admin email and admin password.

Step 4> An admin main page will be displayed containing three options and showing list of flights that admin has entered:

Case 1: Add Flights -> step 5.

Case 2: Change Password -> step 6.

Case 3: Logout and go back to step 2.

Step 5> A new window will be shown where admin can enter flight details and go back step 4.

Step 6> For changing password, it will ask for new and confirmation password from admin. Once it is validated correctly it will go back to home page.

Step 7> This will show a window having flights details that are filtered out through booking details. And the user has to select the flight for further action.

Step 8> Once, the user has selected the flight, user has to register his/her details and should be less than or equal to number of persons that user has specified.

Step 9> After continuation, it will show the summary of user's flight and will prompt the user for payment (Dummy Payment Gateway).

Step 10> Once payment is successful, it will take the user back to step 2.

Step 11> Stop

Conclusion

1: The prototype is robust and platform independent.

2: User can easily use the prototype and safely exit out of it.

3: As a developer, we can enhance it by introducing several new features such as guards along each web pages as currently its statically connected with each along with backend as will not allow to go back once admin has been logout, routing, custom validators and can have more user-friendly by adding styling (CSS, Bootstrap), custom loaders.

4: Though this prototype is tightly connected, the data will only persist in database until server is running and gets reset with each restarting of sever because of manual configuration of hibernate.

5: This prototype can also be implemented with multithreading to enable better performance.

6: And lastly, this prototype can be upgraded by implementing with securities patches to make it more versatile and secure in both local environment and global and later can be configured dynamically with connection of database through hibernate.