Views(JSP pages)

user_index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Search flight</title>
<script type="text/javascript">
      function validateForm() {
            let source = document.getElementById("source").value;
            let destination = document.getElementById("destination").value;
            let isValid = true;
            let inputDate = document.getElementById("date").value;
            let currentDate = (new Date()).toLocaleDateString();
            let enteredDate = (new Date(inputDate)).toLocaleDateString();
            let seats = parseInt(document.getElementById("seats").value);
            if (source === destination) {
                  alert("Source and Destination cannot be same. Please enter
different source and destination details");
                  isValid = false;
            }
            if (enteredDate < currentDate) {
                  alert("Entered date is in past. Please enter a valid
date");
                  isValid = false;
            }
            if(seats <= 0){
                  alert("Seats cannot be less than are equal to 0. Please
enter valid number of seats");
                  isValid = false;
            }

            return isValid;
      }
</script>
</head>
<body>
      <h1>Welcome to FlyAway</h1>
      <form action="search" method="post" onsubmit="return validateForm()">

            <table>
                  <tr>
                        <td>Source</td>
                        <td><input type="text" name="source"
placeholder="Source city"
                              id="source" required /></td>
                  </tr>
                  <tr>
                        <td>Destination</td>
                        <td><input type="text" name="destination"
```

```
                    placeholder="Destination city" id="destination"
required /></td>
                </tr>
                <tr>
                    <td>Number of seats</td>
                    <td><input type="number" name="seats"
                        placeholder="Number of seats" id="seats"
required /></td>
                </tr>
                <tr>
                    <td>Date</td>
                    <td><input type="date" name="date" placeholder="Date"
id="date"
                        required /></td>
                </tr>
                <tr>
                    <td colspan="2"><input type="submit" value="Search"
/></td>
                </tr>
            </table>


        </form>
</body>
</html>
```

user_searchresult.jsp

```
<%@page import="com.scode.user.model.Flight"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    if(session.getAttribute("source") == null)
        response.sendRedirect("user_index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Flights</title>
</head>
<body>
    <%
    ArrayList<Flight> flights = (ArrayList<Flight>)
request.getAttribute("flights");
    String message = (String) request.getAttribute("message");
    if(flights != null){
    %>
    <h1><%=message%></h1>
    <form action="book" method="post">
        <table border="1">
            <thead>
                <tr>
                    <th>FlightId</th>
```

```
					<th>Airline</th>
					<th>Available seats</th>
					<th>Date</th>
					<th>Price per seat</th>
					<th></th>
				</tr>
			</thead>
			<%
			if (!flights.isEmpty()) {
				for (Flight flight : flights) {
			%>
			<tr>
				<td><%=flight.getId()%></td>
				<td><%=flight.getAirline()%></td>
				<td><%=flight.getSeats()%></td>
				<td><%=flight.getDate()%></td>
				<td><%=flight.getPrice()%></td>
				<td><input type="submit" value="Book flight" /><input
					type="hidden" value=<%=flight.getId()%>
name="bookingflightid" /></td>
			</tr>
			<%
			}
			}
			%>
		</table>
	</form>
	<a href="user_index.jsp">Go to search</a>
	<%} %>

</body>
</html>
```

user_confirmbooking.jsp

```
<%@page import="com.scode.user.model.Flight"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
	pageEncoding="UTF-8"%>
<%
Flight flight = null;
if (session.getAttribute("source") == null ||
request.getAttribute("flightdetails") == null)
	response.sendRedirect("user_index.jsp");
else {
	flight = (Flight) request.getAttribute("flightdetails");
	session.setAttribute("flight", flight);
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Confirm booking</title>

</head>
```

```
<body>
      <% if(flight != null) {%>
      <h1>Confirm booking</h1>
      <form action="confirmbooking" method="post"
            onsubmit="return validateForm()">
            <input type="hidden" name="flight" value="<%=flight%>" />
            <table>
                  <tr>
                        <td>Flight ID</td>
                        <td><input type="text" value=<%=flight.getId()%>
readonly /></td>
                  </tr>
                  <tr>
                        <td>Source</td>
                        <td><input type="text" value=<%=flight.getSource()%>
readonly /></td>
                  </tr>
                  <tr>
                        <td>Destination</td>
                        <td><input type="text"
value=<%=flight.getDestination()%>
                              readonly /></td>
                  </tr>
                  <tr>
                        <td>Airline</td>
                        <td><input type="text" value=<%=flight.getAirline()%>
readonly /></td>
                  </tr>
                  <tr>
                        <td>Available seats</td>
                        <td><input type="text" value=<%=flight.getSeats()%>
readonly /></td>
                  </tr>
                  <tr>
                        <td>Date</td>
                        <td><input type="text" value=<%=flight.getDate()%>
readonly /></td>
                  </tr>
                  <tr>
                        <td>First Name</td>
                        <td><input type="text" name="firstname"
                              placeholder="First Name" required /></td>
                  </tr>
                  <tr>
                        <td>Last Name</td>
                        <td><input type="text" name="lastname"
placeholder="Last Name"
                              required /></td>
                  </tr>
                  <tr>
                        <td>Phone number</td>
                        <td><input type="text" name="phno" placeholder="Phone
number"
                              required /></td>
                  </tr>
                  <tr>
                        <td>Email</td>
                        <td><input type="email" name="email"
placeholder="Email"
                              required /></td>
                  </tr>
                  <tr>
                        <td>Confirm number of seats</td>
                        <td><input type="number" name="seats"
placeholder="Seats"
                              id="seats" required /></td>
                  </tr>
                  <tr>
                        <td>Total Price</td>
                        <td><input type="number" name="totalprice" id="price"
readonly /></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" name="book"
/></td>
                  </tr>
            </table>

      </form>


</body>
<script type="text/javascript">
      let price = document.getElementById("price");
      let seat  = document.getElementById("seats");
      function validateForm() {
            let isValid = true;
            let seats = parseInt(seat.value);
            if (seats <= 0) {
                  alert("Seats cannot be less than are equal to 0. Please
enter valid number of seats");
                  isValid = false;
            }
            if (seats >
<%=flight.getSeats()%>
      ) {
                  alert("Entered number of seats is greater than available
seats");
                  isValid = false;
            }

            return isValid;
      }

      document.getElementById("seats").addEventListener('focusout',() => {
            let seats = parseInt(seat.value);
            price.value = seats * <%=flight.getPrice()%>;
      });

</script>
<% } %>
</html>
```

user_booking.jsp

```
<%@page import="com.scode.user.model.Booking"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("source") == null)
      response.sendRedirect("user_index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Confirm page</title>
</head>
<body>
      <h1 id="payment">Processing payment</h1>
      <div id="container" style="visibility: hidden">
            <%
            String id = (String) request.getAttribute("messageid");
            String message = (String) request.getAttribute("message");
            Booking booking = (Booking) request.getAttribute("booking");
            %>
            <h1><%=message%></h1>
            <%
            if (id == "1") {
            %>
            <table>
                  <tr>
                        <td>Booking ID</td>
                        <td><%=booking.getBookingId()%></td>
                  </tr>
                  <tr>
                        <td>Flight ID</td>
                        <td><%=booking.getFlightId()%></td>
                  </tr>
                  <tr>
                        <td>First Name</td>
                        <td><%=booking.getFirstName()%></td>
                  </tr>
                  <tr>
                        <td>Last Name</td>
                        <td><%=booking.getLastName()%></td>
                  </tr>
                  <tr>
                        <td>Phone Number</td>
                        <td><%=booking.getNumber()%></td>
                  </tr>
                  <tr>
                        <td>Email</td>
                        <td><%=booking.getEmail()%></td>
                  </tr>
                  <tr>
                        <td>Date</td>
                        <td><%=booking.getDate()%></td>
                  </tr>
                  <tr>
                        <td>Seats</td>
                        <td><%=booking.getSeats()%></td>
                  </tr>
                  <tr>
                        <td>Price</td>
                        <td><%=booking.getPrice()%></td>
                  </tr>
                  <tr>
                        <td colspan="2"><a href="user_index.jsp">Go to
search</a></td>
                  </tr>
            </table>
            <%
            }
            %>
      </div>
</body>
<script>
let heading = document.getElementById("payment");
let container = document.getElementById("container");
const myTimeout = setTimeout(() => {
      heading.style.visibility = "hidden";

      container.style.visibility = "visible";
}, 5000);

</script>
</html>
```

admin_index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>FlyAway</title>
</head>
<body>
      <h1>Welcome to FlyAway Admin portal</h1>
      <form action="login" method="post">

            <table>
                  <tr>
                        <td>Username</td>
                        <td><input type="text" name="username"
placeholder="Username" required/></td>
                  </tr>
                  <tr>
                        <td>Password</td>
                        <td><input type="password" name="password"
placeholder="password" required/></td>
                  </tr>
                  <tr>
```

```
                        <td colspan="2"><input type="submit"
value="Login"/></td>
                  </tr>
            </table>


      </form>
</body>
</html>
```

admin_dashboard.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
//System.out.println(session.getAttribute("isValid"));
if (session.getAttribute("isValid") == null)
      response.sendRedirect("admin_index.jsp");
else {
      boolean isValid = (Boolean) session.getAttribute("isValid");

      if (!isValid)
            response.sendRedirect("admin_index.jsp");
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Dashboard</title>
</head>
<body>
      <h1>Welcome to dashboard</h1>
      <table>
            <tr>
                  <td><a href="admin_changepassword.jsp">Change
password</a></td>
            </tr>
            <tr>
                  <td><a href="admin_addplace.jsp">Add new place</a></td>
            </tr>
            <tr>
                  <td><a href="admin_addairline.jsp">Add new airline</a></td>
            </tr>
            <tr>
                  <td><a href="admin_addflight.jsp">Add new flight</a></td>
            </tr>
            <tr>
                  <td><a href="showflights">Show flights</a></td>
            </tr>
            <tr>
                  <td><a href="showbookings">Show bookings</a></td>
            </tr>
            <tr>
                  <td><a href="showairlines">Show airlines</a></td>
            </tr>
            <tr>
            <td><a href="showplaces">Show places</a></td>
        </tr>
        <tr>
            <td><a href="logout">Logout</a></td>
        </tr>
    </table>
</body>
</html>
```

admin_changepassword.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
    response.sendRedirect("admin_index.jsp");
else {
    boolean isValid = (Boolean) session.getAttribute("isValid");

    if (!isValid)
        response.sendRedirect("admin_index.jsp");
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Change password</title>
</head>
<body>
    <h1>Enter a new password</h1>
    <form action="change" method="post">
        <table>
            <tr>
                <td>New Password:</td>
                <td><input type="password" name="password"
                    placeholder="New password" required /></td>
            </tr>
            <tr>
                <td colspan="2"><input type="submit" value="Change
password" /></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

admin_addplace.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
    response.sendRedirect("admin_index.jsp");
else {
    boolean isValid = (Boolean) session.getAttribute("isValid");

    if (!isValid)
```

```
            response.sendRedirect("admin_index.jsp");
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add place</title>
</head>
<body>
      <h1>Enter place name to add</h1>
      <form action="addplace" method="post">
            <table>
                  <tr>
                        <td>New Place:</td>
                        <td><input type="text" name="place"
placeholder="Place name"
                              required /></td>
                  </tr>
                  <tr>
                        <td colspan="2"><input type="submit" value="Add
place" /></td>
                  </tr>
            </table>
      </form>
</body>
</html>
```

admin_addairline.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
      response.sendRedirect("admin_index.jsp");
else {
      boolean isValid = (Boolean) session.getAttribute("isValid");

      if (!isValid)
            response.sendRedirect("admin_index.jsp");
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add airline</title>
</head>
<body>
      <h1>Enter airline name to add</h1>
      <form action="addairline" method="post">
            <table>
                  <tr>
                        <td>New Airline:</td>
                        <td><input type="text" name="airline"
                              placeholder="Airline name" required /></td>
                  </tr>
            <tr>
                <td colspan="2"><input type="submit" value="Add
airline" /></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

admin_addflight.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
    response.sendRedirect("admin_index.jsp");
else {
    boolean isValid = (Boolean) session.getAttribute("isValid");

    if (!isValid)
        response.sendRedirect("admin_index.jsp");
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add flight</title>
<script>
    //  alert(new Date());
    function validateForm() {
        let source = document.getElementById("source").value;
        let destination = document.getElementById("destination").value;
        let isValid = true;
        let inputDate = document.getElementById("date").value;
        let currentDate = (new Date()).toLocaleDateString();

        let enteredDate = (new Date(inputDate)).toLocaleDateString();

        if (source === destination) {
            alert("Source and Destination cannot be same. Please enter
different source and destination details");
            isValid = false;
        }
        if (enteredDate < currentDate) {
            alert("Entered date is in past. Please enter a valid
date");
            isValid = false;
        }
        return isValid;
    }
</script>

</head>
<body>
```

```
    <h1>Enter flight details to add</h1>
    <form action="addflight" method="post" onsubmit="return
validateForm()">
        <table>
            <tr>
                <td>New Flight ID:</td>
                <td><input type="text" name="flightid"
placeholder="Flight ID"
                    required /></td>
            </tr>
            <tr>
                <td>Source:</td>
                <td><input type="text" name="source"
placeholder="Source"
                    id="source" required /></td>
            </tr>
            <tr>
                <td>Destination:</td>
                <td><input type="text" name="destination"
                    placeholder="Destination" id="destination"
required /></td>
            </tr>
            <tr>
                <td>Airline:</td>
                <td><input type="text" name="airline"
placeholder="Airline"
                    required /></td>
            </tr>
            <tr>
                <td>Total seats:</td>
                <td><input type="text" name="seats"
placeholder="Seats"
                    required /></td>
            </tr>
            <tr>
                <td>Date:</td>
                <td><input type="date" name="date" placeholder="Date"
id="date"
                    required /></td>
            </tr>
            <tr>
                <td>Price:</td>
                <td><input type="number" name="price"
placeholder="Price"
                    required /></td>
            </tr>
            <tr>
                <td colspan="2"><input type="submit" value="Add
flight" /></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

admin_loginerror.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
      response.sendRedirect("admin_index.jsp");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Invalid login page</title>
</head>
<body>
      <h1>Invalid login deatils</h1>
      <%
            session.removeAttribute("isValid");
      %>
      <a href="admin_index.jsp">Go to login</a>
</body>
</html>
```

admin_result.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
String message = (String) request.getAttribute("message");
%>
<%
if (session.getAttribute("isValid") == null)
      response.sendRedirect("admin_index.jsp");
else {
      boolean isValid = (Boolean) session.getAttribute("isValid");

      if (!isValid)
            response.sendRedirect("admin_index.jsp");
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title><%=message%></title>
</head>
<body>

      <h1><%=message%></h1>
      <a href="admin_dashboard.jsp">Go to dashboard</a>
</body>
</html>
```

admin_showairlines.jsp

```
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
      response.sendRedirect("admin_index.jsp");
else {
      boolean isValid = (Boolean) session.getAttribute("isValid");

      if (!isValid)
            response.sendRedirect("admin_index.jsp");
}
if (request.getAttribute("airlines") == null) {
      response.sendRedirect("admin_dashboard.jsp");
} else {
      String message = (String) request.getAttribute("message");
      ArrayList<String> airlines = (ArrayList<String>)
request.getAttribute("airlines");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Places</title>
</head>
<body>
      <h1><%=message%></h1>
      <%
      if (!airlines.isEmpty()) {
      %>
      <table>
            <tr>
                  <th>Airlines</th>
            </tr>
            <%
            for (String airline : airlines) {
            %>
                  <tr><td><%= airline %></td></tr>
            <%
            }
            %>
      </table>
      <a href ="admin_dashboard.jsp">Go to dashboard</a>
</body>
</html>
<%
}
}
%>
```

admin_showbookings.jsp

```
<%@page import="com.scode.user.model.Booking"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
```

```
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
    response.sendRedirect("admin_index.jsp");
else {
    boolean isValid = (Boolean) session.getAttribute("isValid");

    if (!isValid)
        response.sendRedirect("admin_index.jsp");
}
if (request.getAttribute("bookings") == null) {
    response.sendRedirect("admin_dashboard.jsp");
} else {
    String message = (String) request.getAttribute("message");
    ArrayList<Booking> bookings = (ArrayList<Booking>)
request.getAttribute("bookings");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Bookings</title>
</head>
<body>
    <h1><%=message%></h1>
    <%
    if (!bookings.isEmpty()) {
    %>
    <table>
        <tr>
            <th>Booking ID</th>
            <th>Flight ID</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Phone Number</th>
            <th>Email</th>
            <th>Date</th>
            <th>Seats</th>
            <th>Price</th>
        </tr>
        <%
        for (Booking booking : bookings) {
        %>
        <tr>
            <td><%=booking.getBookingId()%></td>
            <td><%=booking.getFlightId()%></td>
            <td><%=booking.getFirstName()%></td>
            <td><%=booking.getLastName()%></td>
            <td><%=booking.getNumber()%></td>
            <td><%=booking.getEmail()%></td>
            <td><%=booking.getDate()%></td>
            <td><%=booking.getSeats()%></td>
            <td><%=booking.getPrice()%></td>
        </tr>
        <%
        }
        %>
```

```
	</table>
	<a href="admin_dashboard.jsp">Go to dashboard</a>
</body>
</html>
<%
}
}
%>
```

admin_showflights.jsp

```
<%@page import="com.scode.admin.model.Flight"%>
<%@page import="com.scode.user.model.Booking"%>
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
	pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
	response.sendRedirect("admin_index.jsp");
else {
	boolean isValid = (Boolean) session.getAttribute("isValid");

	if (!isValid)
		response.sendRedirect("admin_index.jsp");
}
if (request.getAttribute("flights") == null) {
	response.sendRedirect("admin_dashboard.jsp");
} else {
	String message = (String) request.getAttribute("message");
	ArrayList<Flight> flights = (ArrayList<Flight>)
request.getAttribute("flights");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Flights</title>
</head>
<body>
	<h1><%=message%></h1>
	<%
	if (!flights.isEmpty()) {
	%>
	<table>
		<tr>
			<th>Flight ID</th>
			<th>Source</th>
			<th>Destination</th>
			<th>Airline</th>
			<th>seats</th>
			<th>Date</th>
			<th>Price</th>
		</tr>
		<%
		for (Flight flight : flights) {
        %>
        <tr>
            <td><%=flight.getId()%></td>
            <td><%=flight.getSource()%></td>
            <td><%=flight.getDestination()%></td>
            <td><%=flight.getAirline()%></td>
            <td><%=flight.getSeats()%></td>
            <td><%=flight.getDate()%></td>
            <td><%=flight.getPrice()%></td>
        </tr>
        <%
        }
        %>
    </table>
    <a href="admin_dashboard.jsp">Go to dashboard</a>
</body>
</html>
<%
}
}
%>
```

admin_showplaces.jsp

```
<%@page import="java.util.ArrayList"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
if (session.getAttribute("isValid") == null)
    response.sendRedirect("admin_index.jsp");
else {
    boolean isValid = (Boolean) session.getAttribute("isValid");

    if (!isValid)
        response.sendRedirect("admin_index.jsp");
}
if (request.getAttribute("places") == null) {
    response.sendRedirect("admin_dashboard.jsp");
} else {
    String message = (String) request.getAttribute("message");
    ArrayList<String> places = (ArrayList<String>)
request.getAttribute("places");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Places</title>
</head>
<body>
    <h1><%=message%></h1>
    <%
    if (!places.isEmpty()) {
    %>
    <table>
        <tr>
            <th>Places</th>
            </tr>
            <%
            for (String place : places) {
            %>
                  <tr><td><%= place %></td></tr>
            <%
            }
            %>
      </table>
      <a href ="admin_dashboard.jsp">Go to dashboard</a>
</body>
</html>
<%
}
}
%>
```

ConnectionUtil.java

```
package com.scode.util;


import java.sql.*;


import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.sql.DataSource;


public class ConnectionUtil {

      public static Connection getConnection() {

            Connection connection = null;

            try {

                  InitialContext context = new InitialContext();

                  DataSource dataSource = (DataSource)
context.lookup("java:comp/env/myds");


                  connection = dataSource.getConnection();
//                      System.out.println(connection);
                } catch (NamingException e) {
                        e.printStackTrace();
                } catch (SQLException e) {
                        e.printStackTrace();
                }
                return connection;
        }
}
```

Models(Java classes)

Flight.java

```
package com.scode.user.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.Date;

import com.scode.util.ConnectionUtil;

public class Flight {
    private String id;

    private String source;

    private String destination;

    private String airline;

    private int seats;

    private String date;

    private double price;

    private int count;


    private Connection con;

    private PreparedStatement stmt, stmt1, stmt2, stmt3;


    public Flight() {

        try {

            con = ConnectionUtil.getConnection();

            stmt = con.prepareStatement(

                    "select * from flights where source = ? and
destination = ? and date = ? and seats >= ?");

            stmt1 = con.prepareStatement("select * from flights where
source = ? and destination = ?");

            stmt2 = con.prepareStatement(

                    "select count(*) as count from flights where
source = ? and destination = ? and date = ? and seats >= ?");

            stmt3 = con.prepareStatement("select * from flights where
flightid = ?");

        } catch (SQLException e) {
```

```
        e.printStackTrace();
    }

}

public String getId() {
    return id;
}

public void setId(String id) {
    this.id = 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 String getAirline() {
    return airline;
}

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

public int getSeats() {
    return seats;
}

public void setSeats(int seats) {
    this.seats = seats;
}

public String getDate() {
    return date;
}
```

```
    public void setDate(String date) {
        this.date = date;
    }

    public double getPrice() {
        return price;
    }

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

    @Override
    public String toString() {
        return "Flight [id=" + id + ", source=" + source + ", destination=" +
destination + ", airline=" + airline
                + ", seats=" + seats + ", date=" + date + ", price=" + price
+ "]";
    }

    public int getCount() {
        return count;
    }
```

```
public ArrayList<Flight> searchFlight() {
    ArrayList<Flight> flightList = new ArrayList<Flight>();
    try {
        stmt2.setString(1, source);
        stmt2.setString(2, destination);
        stmt2.setString(3, date);
        stmt2.setInt(4, seats);
        ResultSet rs = stmt2.executeQuery();
        rs.next();
        count = rs.getInt(1);
        if (count > 0) {
            stmt.setString(1, source);
            stmt.setString(2, destination);
            stmt.setString(3, date);
            stmt.setInt(4, seats);
            rs = stmt.executeQuery();
        } else {
            stmt1.setString(1, source);
            stmt1.setString(2, destination);
            rs = stmt1.executeQuery();
        }
        while (rs.next()) {
            Flight flight = new Flight();
            flight.setSource(source);
```

```
                flight.setDestination(destination);
                flight.setDate(rs.getString(6));
                flight.setAirline(rs.getString(3));
                flight.setId(rs.getString(1));
                flight.setSeats(rs.getInt(5));
                flight.setPrice(rs.getDouble(7));
                flightList.add(flight);
                flight.destroy();
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return flightList;
    }

    public Flight searchFlight(String id) {
        Flight flight = new Flight();
        try {
            stmt3.setString(1, id);
            ResultSet rs = stmt3.executeQuery();
            rs.next();
            flight.setId(id);
            flight.setSource(rs.getString(2));
        flight.setDestination(rs.getString(3));
        flight.setAirline(rs.getString(4));
        flight.setSeats(rs.getInt(5));
        flight.setDate(rs.getString(6));
        flight.setPrice(rs.getDouble(7));
        flight.destroy();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return flight;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
        stmt3.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

```
}
```

Booking.java

```
package com.scode.user.model;


import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;


import com.scode.util.ConnectionUtil;


public class Booking {


	private String bookingId;
	private String flightId;
	private String firstName;
	private String lastName;
	private String number;
	private String email;
	private String date;
	private int seats;
	private double price;
```

```
    private Connection con;
    private PreparedStatement stmt, stmt1,stmt2;

    public Booking() {
        con = ConnectionUtil.getConnection();
        try {
            stmt = con.prepareStatement("insert into bookings values
(?,?,?,?,?,?,?,?,?)");
            stmt1 = con.prepareStatement("update flights set seats = ?
where flightid= ?");
            stmt2 = con.prepareStatement("select * from bookings");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public String getBookingId() {
        return bookingId;
    }

    public void setBookingId(String bookingId) {
        this.bookingId = bookingId;
    }

    public String getFlightId() {
```

```
		return flightId;
	}

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

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public String getNumber() {
        return number;
}

public void setNumber(String number) {
        this.number = number;
}

public String getEmail() {
        return email;
}

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

public String getDate() {
        return date;
}

public void setDate(String date) {
        this.date = date;
}

public int getSeats() {
        return seats;
    }

    public void setSeats(int seats) {
        this.seats = seats;
    }

    public double getPrice() {
        return price;
    }

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

    @Override
    public String toString() {
        return "Booking [bookingId=" + bookingId + ", flightId=" + flightId + ", firstName=" + firstName + ", lastName="
                + lastName + ", number=" + number + ", email=" + email + ", date=" + date + ", seats=" + seats
                + ", price=" + price + "]";
    }

    public int saveBooking() {
```

```
    int result = 0;
    try {
        stmt.setString(1, bookingId);
        stmt.setString(2, flightId);
        stmt.setString(3, firstName);
        stmt.setString(4, lastName);
        stmt.setString(5, number);
        stmt.setString(6, email);
        stmt.setString(7, date);
        stmt.setInt(8, seats);
        stmt.setDouble(9, price);
        result = stmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public int updateSeats(int s) {
    int remainingSeats = s - seats;
    int result = 0;
    try {
        stmt1.setInt(1, remainingSeats);
        stmt1.setString(2, flightId);
        result = stmt1.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public ArrayList<Booking> getBookings() {
    ArrayList<Booking> bookings = new ArrayList<Booking>();
    try {
        ResultSet rs = stmt2.executeQuery();
        while(rs.next()) {
            Booking booking = new Booking();
```

```
                booking.setBookingId(rs.getString(1));
                booking.setFlightId(rs.getString(2));
                booking.setFirstName(rs.getString(3));
                booking.setLastName(rs.getString(4));
                booking.setNumber(rs.getString(5));
                booking.setEmail(rs.getString(6));
                booking.setDate(rs.getString(7));
                booking.setSeats(rs.getInt(8));
                booking.setPrice(rs.getDouble(9));
                bookings.add(booking);
                booking.destroy();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return bookings;
    }

}
```

Admin.java

```
package com.scode.admin.model;

import java.sql.*;
import com.scode.util.ConnectionUtil;

public class Admin {

    private String username;
    private String password;

    private Connection con;
    private PreparedStatement stmt,stmt1;

    public Admin() {
        try {
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from admin where
username = ? and password = ?");
            stmt1 = con.prepareStatement("update admin set password =
?");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

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

    public boolean validate() {
        boolean result = false;
        try {
            stmt.setString(1, username);
            stmt.setString(2, password);
            ResultSet rs = stmt.executeQuery();
            result = rs.next();
//          System.out.println(result);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    public int updatePassword(String password) {
        int result = 0;
        try {
            stmt1.setString(1, password);
            result = stmt1.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    public void destroy() {
        try {
            con.close();
            stmt.close();
            stmt1.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }

    }
}
```

Airline.java

```
package com.scode.admin.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import com.scode.util.ConnectionUtil;

public class Airline {

    private String name;

    private Connection con;
    private PreparedStatement stmt,stmt1,stmt2;


    public Airline() {
        try {
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from airline where
airlinename = ?");
            stmt1 = con.prepareStatement("insert into airline values(?)");
            stmt2 = con.prepareStatement("select * from airline");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void setName(String name) {
```

```
    this.name = name;
}

public int addPlace() {
    int result = 0;
    try {
        stmt.setString(1, name);
        ResultSet rs = stmt.executeQuery();
        if(rs.next()) {
            result = -1;
        }else {
            stmt1.setString(1, name);
            result = stmt1.executeUpdate();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
```

```
            stmt1.close();
            stmt2.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ArrayList<String> getAirlines() {
        ArrayList<String> airlines = new ArrayList<String>();
        try {
            ResultSet rs = stmt2.executeQuery();
            while(rs.next())
                airlines.add(rs.getString(1));
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return airlines;
    }

}
```

Flight.java

```
package com.scode.admin.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import com.scode.util.ConnectionUtil;



public class Flight {
    private String id;
    private String source;
    private String destination;
    private String airline;
    private int seats;
    private String date;
    private double price;

    private Connection con;
    private PreparedStatement stmt,stmt1,stmt2,stmt3,stmt4;

    public Flight() {
        try {
            con = ConnectionUtil.getConnection();
            stmt = con.prepareStatement("select * from airline where airlinename = ?");
            stmt1 = con.prepareStatement("select * from places where place = ?");
            stmt2 = con.prepareStatement("select * from flights where flightid = ?");
            stmt3 = con.prepareStatement("insert into flights values(?,?,?,?,?,?,?)");
            stmt4 = con.prepareStatement("select * from flights");
        } catch (SQLException e) {
            e.printStackTrace();
        }

    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = 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 String getAirline() {
        return airline;
}

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

public int getSeats() {
        return seats;
}
```

```
    }

    public void setSeats(int seats) {
        this.seats = seats;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public double getPrice() {
        return price;
    }



    @Override
    public String toString() {
        return "Flight [id=" + id + ", source=" + source + ", destination=" +
destination + ", airline=" + airline
                + ", seats=" + seats + ", date=" + date + ", price=" + price
+ "]";
    }
```

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

public int addFlight() {
    int result = 0;
    try {
        stmt.setString(1, airline);
        ResultSet rs = stmt.executeQuery();
        if(!rs.next())
            return -1;
        stmt1.setString(1, source);
        rs = stmt1.executeQuery();
        if(!rs.next())
            return -2;
        stmt1.setString(1, destination);
        rs = stmt1.executeQuery();
        if(!rs.next())
            return -3;
        stmt2.setString(1, id);
        rs = stmt2.executeQuery();
        if(rs.next()) {
            System.out.println(rs.getString(1));
            return -4;
        }

        stmt3.setString(1, id);
        stmt3.setString(2, source);
        stmt3.setString(3, destination);
        stmt3.setString(4, airline);
        stmt3.setInt(5, seats);
        stmt3.setString(6, date);
        stmt3.setDouble(7, price);
        result = stmt3.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}
public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
        stmt3.close();
        stmt4.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ArrayList<Flight> getBookings() {
        ArrayList<Flight> flights = new ArrayList<Flight>();
        try {
            ResultSet rs = stmt4.executeQuery();
            while(rs.next()) {
                Flight flight = new Flight();
                flight.setId(rs.getString(1));
                flight.setSource(rs.getString(2));
                flight.setDestination(rs.getString(3));
                flight.setAirline(rs.getString(4));
                flight.setSeats(rs.getInt(5));
                flight.setDate(rs.getString(6));
                flight.setPrice(rs.getDouble(7));
                flights.add(flight);
                flight.destroy();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
```

```
		return flights;
	}
}
```

Place.java

```
package com.scode.admin.model;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import com.scode.util.ConnectionUtil;

public class Place {

	private String name;

	private Connection con;
	private PreparedStatement stmt, stmt1,stmt2;

	public Place() {
		try {
			con = ConnectionUtil.getConnection();
			stmt = con.prepareStatement("select * from places where place = ?");
			stmt1 = con.prepareStatement("insert into places values(?)");
			stmt2 = con.prepareStatement("select * from places");
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}

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

	public int addPlace() {
		int result = 0;
		try {
			stmt.setString(1, name);
			ResultSet rs = stmt.executeQuery();
			if (rs.next()) {
				result = -1;
			} else {
				stmt1.setString(1, name);
				result = stmt1.executeUpdate();
			}
		} catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}

public ArrayList<String> getPlaces() {
    ArrayList<String> places = new ArrayList<String>();
    try {
        ResultSet rs = stmt2.executeQuery();
        while(rs.next())
            places.add(rs.getString(1));
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return places;
}

public void destroy() {
    try {
        con.close();
        stmt.close();
        stmt1.close();
        stmt2.close();
    } catch (SQLException e) {
            e.printStackTrace();
        }
    }


}
```

Controllers (Servlets)

BookController.java

```
package com.scode.user.controllers;

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

import java.io.IOException;

import com.scode.user.model.Flight;

/**
```

```
 * Servlet implementation class BookController
 */
@WebServlet("/book")
public class BookController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public BookController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        doPost(request, response);
    }
    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("source") == null ||
request.getParameter("bookingflightid") == null) {
            RequestDispatcher rd =
request.getRequestDispatcher("user_index.jsp");
            rd.forward(request, response);
        }

        String flightid = request.getParameter("bookingflightid");
        Flight flight = new Flight();
        Flight result = flight.searchFlight(flightid);
        RequestDispatcher rd =
request.getRequestDispatcher("user_confirmbooking.jsp");
        request.setAttribute("flightdetails", result);
        flight.destroy();
        rd.forward(request, response);
    }
```

```
}
```

ConfirmBookingController.java

```
package com.scode.user.controllers;

import java.io.IOException;
import java.util.Random;

import com.scode.user.model.Booking;
import com.scode.user.model.Flight;

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

/**
 * Servlet implementation class ConfirmBookingController
 */
@WebServlet("/confirmbooking")
public class ConfirmBookingController extends HttpServlet {
	private static final long serialVersionUID = 1L;
```

```
    /**
     * @see HttpServlet#HttpServlet()
     */
    public ConfirmBookingController() {
        super();
        // TODO Auto-generated constructor stub
    }

            /**
             * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
             */
            protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                        doPost(request, response);
            }

            /**
             * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
             */
            protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
//                      String flight = request.getParameter("flight");
                Flight flight = (Flight) request.getSession().getAttribute("flight");
```

```
                        HttpSession session = request.getSession();

                        if (session.getAttribute("source") == null ||
request.getParameter("firstname") == null) {

                                    RequestDispatcher rd =
request.getRequestDispatcher("user_index.jsp");

                                    rd.forward(request, response);

                        }

                Random random = new Random();

        String characters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

        int length = 10;

        StringBuilder randomText = new StringBuilder();

        for (int i = 0; i < length; i++) {

            int randomIndex = random.nextInt(characters.length());

            randomText.append(characters.charAt(randomIndex));

        }


        String firstname = request.getParameter("firstname");

        String lastname = request.getParameter("lastname");

        String phno = request.getParameter("phno");

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

        int seats = Integer.parseInt(request.getParameter("seats"));

        double price = seats * flight.getPrice();


        Booking booking = new Booking();
        booking.setBookingId(randomText.toString());
        booking.setFlightId(flight.getId());
        booking.setFirstName(firstname);
        booking.setLastName(lastname);
        booking.setNumber(phno);
        booking.setEmail(email);
        booking.setDate(flight.getDate());
        booking.setSeats(seats);
        booking.setPrice(price);
        int result = booking.saveBooking();
        booking.updateSeats(flight.getSeats());
//        response.setContentType("text/html");
//        PrintWriter out = response.getWriter();
//        out.print("<h1>Processing payment request</h1>");
//        try {
//            Thread.sleep(3000);
//        } catch (InterruptedException e) {
//            e.printStackTrace();
//        }
        RequestDispatcher rd =
request.getRequestDispatcher("user_bookingresult.jsp");
        if(result > 0) {
          request.setAttribute("messageid", "1");
          request.setAttribute("booking", booking);
          request.setAttribute("message", "Flight booked. Find the details below");
        }else {
          request.setAttribute("messageid", "-1");
          request.setAttribute("message", "Some error occured. Please try again later");
        }
        booking.destroy();
        rd.forward(request, response);
//                System.out.println(booking.toString());
          }


}
```

SearchController.java

```java
package com.scode.user.controllers;

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

import java.io.IOException;
import java.util.ArrayList;

import com.scode.user.model.Flight;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SearchController() {
        super();
        // TODO Auto-generated constructor stub
    }
    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        if(request.getParameter("source") == null) {
            RequestDispatcher rd =
request.getRequestDispatcher("user_index.jsp");
            rd.forward(request, response);
        }
        String source = request.getParameter("source");
        String destination = request.getParameter("destination");
        int seats = Integer.parseInt(request.getParameter("seats"));
        String date = request.getParameter("date");
        System.out.println(date);
        Flight flight = new Flight();
        flight.setDestination(destination);
        flight.setDate(date);
        flight.setSource(source);
        flight.setSeats(seats);
        RequestDispatcher rd =
request.getRequestDispatcher("user_searchresult.jsp");
        ArrayList<Flight> result = flight.searchFlight();

        if(flight.getCount() == 0 && !result.isEmpty()) {
            request.setAttribute("message", "Not able to find flights
for your search. Here are some other flights between "+ source+" and
"+destination);
            request.setAttribute("flights", result);
        }else if(result.isEmpty()) {
            request.setAttribute("message", "No flights between "+
source+" and "+destination);
            request.setAttribute("flights", result);
        }else {
            request.setAttribute("message", "Here are some flights
between "+ source+" and "+destination);
            request.setAttribute("flights", result);
        }
        if(session.getAttribute("source") == null)
            session.setAttribute("source", source);
        flight.destroy();
        rd.forward(request, response);

    }
```

```
}
```

AddAirlineController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;

import com.scode.admin.model.Airline;
import com.scode.admin.model.Place;

/**
 * Servlet implementation class AddPlaceController
 */
@WebServlet("/addairline")
public class AddAirlineController extends HttpServlet {
	private static final long serialVersionUID = 1L;
```

```
    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddAirlineController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("airline") == null) {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_dashboard.jsp");
            rd.forward(request, response);
        } else {
            String name = request.getParameter("airline");
            RequestDispatcher rd =
request.getRequestDispatcher("admin_result.jsp");
            Airline airline = new Airline();
            airline.setName(name);
            int result = airline.addPlace();
            if (result == -1) {
                request.setAttribute("message", "Airline already exists in
database. Please enter a new airline.");
            } else {
                request.setAttribute("message", name + " added
succesfully into database");
            }
            airline.destroy();
```

```
			rd.forward(request, response);
		}
	}


}
```

AddFlightController.java

```
package com.scode.admin.controllers;


import java.io.IOException;


import com.scode.admin.model.Flight;


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


/**
 * Servlet implementation class AddFlightController
 */
@WebServlet("/addflight")
```

```
public class AddFlightController extends HttpServlet {
        private static final long serialVersionUID = 1L;

        /**
         * @see HttpServlet#HttpServlet()
         */
        public AddFlightController() {
                super();
                // TODO Auto-generated constructor stub
        }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse
         *      response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse
response)
                        throws ServletException, IOException {
                doPost(request, response);
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
```

```
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("flightid") == null) {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_dashboard.jsp");
            rd.forward(request, response);
        } else {
            String id = request.getParameter("flightid");
            String source = request.getParameter("source");
            String destination = request.getParameter("destination");
            String airline = request.getParameter("airline");
            int seats = Integer.parseInt(request.getParameter("seats"));
            String date = request.getParameter("date");
            double price =
Double.parseDouble(request.getParameter("price"));

            Flight flight = new Flight();
            flight.setId(id);
            flight.setSource(source);
            flight.setDestination(destination);
            flight.setAirline(airline);
            flight.setSeats(seats);
            flight.setDate(date);
            flight.setPrice(price);

            RequestDispatcher rd =
request.getRequestDispatcher("admin_result.jsp");

            if (source == destination) {
                request.setAttribute("message",
                        "Source and destination are same. Please
enter different source and destination");
            }
            int result = flight.addFlight();
            if (result == -1) {
                request.setAttribute("message", "Airline " + airline
                        + " does not exist in database. Please use
existing airline or create a new airline");
            }
            if (result == -2) {
                request.setAttribute("message", "Source " + source
                        + " does not exist in database. Please use
existing places or create a new place");
```

```
            }
            if (result == -3) {
                request.setAttribute("message", "Destination " +
destination
                        + " does not exist in database. Please use
existing places or create a new place");
            }
            if (result == -4) {
                request.setAttribute("message",
                        "Flight id " + id + " already exists in
database. Please use new flight id");
            }
            if (result > 0) {
                request.setAttribute("message", "Flight " +
flight.toString() + " created successfully");

            }
            flight.destroy();
            rd.forward(request, response);
//      System.out.println(request.getParameter("date"));
//      System.out.println(flight.toString());
        }
    }

}
```

AddplaceController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;

import com.scode.admin.model.Place;

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

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

```
     */
    public AddPlaceController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse
     *      response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        }

        if (session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("place") == null) {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_dashboard.jsp");
            rd.forward(request, response);
        } else {
            String name = request.getParameter("place");
            RequestDispatcher rd =
request.getRequestDispatcher("admin_result.jsp");
            Place place = new Place();
            place.setName(name);
            int result = place.addPlace();
            if (result == -1) {
                request.setAttribute("message", "Place already exists in
database. Please enter a new place.");
            } else {
                request.setAttribute("message", name + " added
succesfully into database");
            }
            place.destroy();
            rd.forward(request, response);
        }
```

```
	}

}
```

ChangePasswordController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;

import com.scode.admin.model.Admin;

/**
 * Servlet implementation class ChangePasswordController
 */
@WebServlet("/change")
public class ChangePasswordController extends HttpServlet {
	private static final long serialVersionUID = 1L;
```

```
    /**
     * @see HttpServlet#HttpServlet()
     */
    public ChangePasswordController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse
     *     response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        doPost(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
     *     response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        if (session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
            rd.forward(request, response);
        } else if (request.getParameter("password") == null) {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_dashboard.jsp");
            rd.forward(request, response);
        } else {
            String password = request.getParameter("password");
            Admin admin = new Admin();
            int result = admin.updatePassword(password);
            RequestDispatcher rd =
request.getRequestDispatcher("admin_result.jsp");
            System.out.println(result);
            if (result > 0) {
                request.setAttribute("message", "Password updated
succesfully");
            } else {
                request.setAttribute("message", "Password not updated.
Please try again later.");
```

```
            }
            admin.destroy();
//          System.out.println(request.getAttribute("message"));
            rd.forward(request, response);
        }
    }

}
```

LoginController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;

import com.scode.admin.model.Admin;

/**
```

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginController() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse
     *     response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        doPost(request, response);
    }
    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();


        String username = request.getParameter("username");
        String password = request.getParameter("password");
        RequestDispatcher rd = null;
        Admin admin = new Admin();
        admin.setPassword(password);
        admin.setUsername(username);
        boolean isValid = admin.validate();
        if (isValid) {
            rd = request.getRequestDispatcher("admin_dashboard.jsp");


        } else {
            rd = request.getRequestDispatcher("admin_loginerror.jsp");
        }
        if(session.getAttribute("isValid") == null)
            session.setAttribute("isValid", isValid);
        admin.destroy();
        rd.forward(request, response);

    }
}
```

LogoutController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Airline;

/**
 * Servlet implementation class LogoutController
 */
```

```
@WebServlet("/logout")
public class LogoutController extends HttpServlet {
        private static final long serialVersionUID = 1L;

        /**
         * @see HttpServlet#HttpServlet()
         */
        public LogoutController() {
                super();
                // TODO Auto-generated constructor stub
        }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse
         *      response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse
response)
                        throws ServletException, IOException {
                HttpSession session = request.getSession();
                RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
                if (session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
```

```
            rd.forward(request, response);
        } else {
            session.invalidate();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
     *      response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

ShowAirlinesController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Airline;
import com.scode.admin.model.Place;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showairlines")
public class ShowAirlinesController extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ShowAirlinesController() {
        super();
        // TODO Auto-generated constructor stub
    }


            /**
             * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
             */
            protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                        HttpSession session = request.getSession();
                        if(session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
                                    RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
                                    rd.forward(request, response);
                        }else {
                                    Airline airline = new Airline();
                                    ArrayList<String> airlines = airline.getAirlines();
                                    RequestDispatcher rd =
request.getRequestDispatcher("admin_showairlines.jsp");
                                    if(airlines.isEmpty()) {
                                                request.setAttribute("message", "No airlines found in
database");
                                    }else {
                                                request.setAttribute("message", "Found below airlines in
database");
```

```
            }
            request.setAttribute("airlines", airlines);
            airline.destroy();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

ShowBookingsController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Airline;
import com.scode.admin.model.Place;
import com.scode.user.model.Booking;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showbookings")
public class ShowBookingsController extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ShowBookingsController() {
        super();
        // TODO Auto-generated constructor stub
```

```
   }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        if(session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
            rd.forward(request, response);
        }else {
            Booking booking = new Booking();
            ArrayList<Booking> bookings = booking.getBookings();
            RequestDispatcher rd =
request.getRequestDispatcher("admin_showbookings.jsp");
            if(bookings.isEmpty()) {
                request.setAttribute("message", "No bookings found in
database");
            }else {
                request.setAttribute("message", "Found below bookings
in database");
            }
            request.setAttribute("bookings", bookings);

            booking.destroy();

            rd.forward(request, response);

        }

    }


    /**

     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

     */

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

        // TODO Auto-generated method stub

        doGet(request, response);

    }


}
```

ShowFlightsController.java

```
package com.scode.admin.controllers;


import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Airline;
import com.scode.admin.model.Flight;
import com.scode.admin.model.Place;
import com.scode.user.model.Booking;

/**
 * Servlet implementation class ShowPlacesController
 */
@WebServlet("/showflights")
public class ShowFlightsController extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ShowFlightsController() {
        super();
        // TODO Auto-generated constructor stub
```

```
   }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        if(session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
            RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
            rd.forward(request, response);
        }else {
            Flight flight = new Flight();
            ArrayList<Flight> flights = flight.getBookings();
            RequestDispatcher rd =
request.getRequestDispatcher("admin_showflights.jsp");
            if(flights.isEmpty()) {
                request.setAttribute("message", "No flights found in
database");
            }else {
                request.setAttribute("message", "Found below flights in
database");
            }
            request.setAttribute("flights", flights);
            flight.destroy();
            rd.forward(request, response);
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

ShowPlacesController.java

```
package com.scode.admin.controllers;

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

import java.io.IOException;
import java.util.ArrayList;

import com.scode.admin.model.Place;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ShowPlacesController() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
```

```
	 * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		HttpSession session = request.getSession();
		if(session.getAttribute("isValid") == null ||
session.getAttribute("isValid") == "false") {
			RequestDispatcher rd =
request.getRequestDispatcher("admin_index.jsp");
			rd.forward(request, response);
		}else {
			Place place = new Place();
			ArrayList<String> places = place.getPlaces();
			RequestDispatcher rd =
request.getRequestDispatcher("admin_showplaces.jsp");
			if(places.isEmpty()) {
				request.setAttribute("message", "No places found in
database");
			}else {
				request.setAttribute("message", "Found below places in
database");
			}
			request.setAttribute("places", places);
			place.destroy();
			rd.forward(request, response);
```

```
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```