

SOURCE CODE – FlyAway (An Airline Booking Portal)

FOR ADMIN

ChangePassword.java:

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

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

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

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

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out = resp.getWriter();
        String pass = req.getParameter("password");
        if (!Login.isLoggedIn){
            out.println("You must login first");
        }
        else if (pass.equals("")){
```

```

        out.println("Password can't be empty");
    }
    else if (Login.isLoggedIn && !pass.equals("")){
        Login.password = pass;
        out.println("Password changed. New Password is "+Login.password);
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    else {
        out.println("something went wrong");

    }

    try {
        req.getRequestDispatcher("passwordverified.html").forward(req, resp);
    } catch (ServletException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    out.close();
}
}

```

Login.java:

```

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;

```

```

import javax.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/login")
public class Login extends HttpServlet {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    public static boolean isLoggedIn = false;
    public static String password = "Airline";
    public static String email = "airline@flyaway.com";

    @Override
    public void doPost (HttpServletRequest req, HttpServletResponse resp) throws IOException {
        PrintWriter out = resp.getWriter();

        String email = req.getParameter("username");
        String pass = req.getParameter("password");

        if (email.equals(Login.email) && pass.equals(Login.password)){
            isLoggedIn = true;
            out.println("You have LoggedIn");
            // resp.sendRedirect("/dashboard.html");
            // out.print("<a href=\"dashboard.html\">Home</a>");
            try {
                req.getRequestDispatcher("dashboard.html").forward(req, resp);
            } catch (ServletException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        else {
            isLoggedIn = false;
            out.println("Login Failed : Incorrect email or Password");
        }
        out.close();
    }
}

```

FOR USER:

Confirmation.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

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

import com.simplilearn.tables.Flight;
import com.simplilearn.util.HibernateUtil;

/**
 * Servlet implementation class ConfirmationPage
 */
@WebServlet("/confirmation-page")
public class ConfirmationPage extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public ConfirmationPage() {
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // print writer
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        HttpSession httpsession = request.getSession();
        String flightid = (String) httpsession.getAttribute("flightid");
        String firstName = (String) httpsession.getAttribute("firstname");
        int flightIdConverted = Integer.parseInt(flightid);
        //open a connection
        try {
            SessionFactory sFactory = HibernateUtil.buildSessionFactory();
            Session session = sFactory.openSession();
            session.beginTransaction();
```

```

String query = "select f from Flight f" +
    " inner join Airline a" +
    " on f.airline=a.id" +
    " and f.numberOfSeats > 0" +
    " and f.id = " + flightIdConverted;
@SuppressWarnings("unchecked")
List<Flight> flights = session.createQuery(query).list();

if (flights.size() > 0) {
    out.print("<a href=\"index.html\">Home</a><br/>");
    out.println("<h1>Congratulations! You're all set, " + firstName +
"!</h1><br/><br/>");
    out.println("<h1>List of Confirmed Flights for " + firstName + "</h1><hr>");
    out.println("<style> table,th,td { border : 1px solid black ; padding :15px;}
</style>");
    out.println("<table>");
    out.println("<tr>");
        out.println("<th>"); out.println("Flight ID"); out.println("</th>");
        out.println("<th>"); out.println("Airline"); out.println("</th>");
        out.println("<th>"); out.println("Number Of Seats");
    out.println("</th>");
        out.println("<th>"); out.println("Source"); out.println("</th>");
        out.println("<th>"); out.println("Desintation"); out.println("</th>");
        out.println("<th>"); out.println("Departure Date");
    out.println("</th>");
        out.println("<th>"); out.println("Arrival Date"); out.println("</th>");
        out.println("<th>"); out.println("Price"); out.println("</th>");
    out.println("</tr>");
    for(Flight i:flights) {
        if (i.getNumberOfSeats() > 0) {
            out.println("<tr>");
                out.println("<td>"); out.println(i.getId());
            out.println("</td>");
                out.println("<td>");
            out.println(i.getAirline().getAirline().toString()); out.println("</td>");
                out.println("<td>");
            out.println(i.getNumberOfSeats()); out.println("</td>");
                out.println("<td>"); out.println(i.getSource());
            out.println("</td>");
                out.println("<td>"); out.println(i.getDestination());
            out.println("</td>");
                out.println("<td>");
            out.println(i.getDateOfDeparture()); out.println("</td>");
                out.println("<td>"); out.println(i.getDateOfArrival());
            out.println("</td>");
                out.println("<td>"); out.println(i.getPrice());
            out.println("</td>");
        out.println("</tr>");
    }
}

```

```

        }
    }
    out.println("</table>");
    } else {
        out.print("<a href=\"index.html\">Home</a>");
        out.print("<h1>Sorry You don't have any are no flights
currently!<h1>");
    }

    session.close();

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        out.close();
    }
}

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

}

}

```

PayOnline.java:

```
import java.io.IOException;

import java.io.PrintWriter;
import java.sql.Date;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.regex.Pattern;

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

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

import com.simplilearn.tables.Flight;
import com.simplilearn.tables.Payment;
import com.simplilearn.tables.Person;
import com.simplilearn.util.HibernateUtil;

/**
 * Servlet implementation class PayOnline
 */
@WebServlet("/pay-online")
public class PayOnline extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public PayOnline() {

    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        response.sendRedirect("payment.html");
    }

    /**
```

```

    * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
    */
    @SuppressWarnings({ "unchecked", "rawtypes" })
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        HttpSession httpsession = request.getSession(false);

        String personflightid = (String) httpsession.getAttribute("flightid");
        String personfirstname = (String) httpsession.getAttribute("firstname");
        String personlastname = (String) httpsession.getAttribute("lastname");
        String personemail = (String) httpsession.getAttribute("email");
        String personbirthday = (String) httpsession.getAttribute("birthday");
        long millis = System.currentTimeMillis();
        Date timeBooked = new Date(millis);
        Date dateSqlPerson = null;
        try {
            if ((dateSqlPerson=Date.valueOf(personbirthday)) == null) {
                out.println("<h1>You must enter the date correctly into yyyy-MM-
dd format<h1>");
                out.print("<a href = \"payment.html\"> Re-enter Information
</a><br/>");
            };
        } catch (Exception e1) {
            out.println("<h1>You must enter the date correctly into yyyy-MM-dd
format<h1>");
            out.print("<a href = \"payment.html\"> Re-enter Information </a><br/>");
        }

        String fName = request.getParameter("firstname");
        String lName = request.getParameter("lastname");
        String cardNumber = request.getParameter("cardnumber").replaceAll("\\s", "");
        System.out.println(cardNumber);
        String expiration = request.getParameter("expiration");
        int securitycode = Integer.parseInt(request.getParameter("securitycode"));
        Date dateSqlPayment = null;
        try {
            dateSqlPayment=Date.valueOf(expiration);

        } catch (Exception e1) {
            out.println("<h1>You must enter the date correctly into yyyy-MM-dd
format<h1>");
            out.print("<a href = \"payment.html\"> Re-enter Information </a><br/>");
        }
    }

```



```

if (!Pattern.matches("\\d{15,16}", cardNumber)) {
    out.print("<a href = \"payment.html\"> Re-enter Information </a><br/>");
    out.print("Make sure you are entering your card number in correctly");

} else if (!Pattern.matches("\\d{3,4}", Integer.toString(securitycode))) {
    out.print("<a href = \"payment.html\"> Re-enter Information </a><br/>");
    out.print("Make sure you are entering your security code in correctly");
} else if (dateSqlPayment.before(timeBooked)) {
    out.println("<h1>You must ensure your card is not expired<h1>");
    out.print("<a href = \"payment.html\"> Re-enter Information </a><br/>");
}
else {

```

```

//step 1. Confirm Fields Match Criteria

```

```

    //Else redirect to payment

```

```

//step 2. insert into person table

```

```

//open a connection

```

```

try {

```

```

    SessionFactory sFactory = HibernateUtil.buildSessionFactory();

```

```

    Session session = sFactory.openSession();

```

```

    session.beginTransaction();

```

```

    int personflightidConverted = Integer.parseInt(personflightid);

```

```

    String query1 = " from Flight where id = " + personflightidConverted;

```

```

    List<Flight> list = (List<Flight>) session.createQuery(query1).list();

```

```

    Set<Flight> flights = new HashSet<Flight>(list);

```

```

    Person person = new Person();

```

```

    person.setFirstName(personfirstname);

```

```

    person.setLastName(personlastname);

```

```

    person.setEmail(personemail);

```

```

    person.setBirthday(dateSqlPerson);

```

```

    person.setTimeBooked(timeBooked);

```

```

    person.setFlights(flights);

```

```

    Payment payment = new Payment();

```

```

    payment.setFirstNameOnCard(fName);

```

```

    payment.setLastNameOnCard(lName);

```

```

    payment.setCardNumber(cardNumber);

```

```

    payment.setExpiration(dateSqlPayment);

```

```

    payment.setSecurityCode(securitycode);

```

```

    payment.setPerson(person);

```

```

        session.save(person);
        session.save(payment);

        String query = "select numberOfSeats from Flight f" +
            " where id = " + personflightidConverted;
        int resultFromQ1 = (int) session.createQuery(query).getSingleResult();
        Query q3=session.createQuery("update Flight set numberOfSeats=:n where
id=:i");

        q3.setParameter("n",(resultFromQ1 - 1));
        q3.setParameter("i",personflightidConverted);

        int status=q3.executeUpdate();
        System.out.println(status);

        session.getTransaction().commit();
        session.close();
        response.sendRedirect("confirmation-page");
        out.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
//step 3. insert into payment table
//step 4. reduce person count
//step 5. Redirect to confirmation page

}

}

```

Registration.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Date;
import java.util.List;
import java.util.regex.Pattern;

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

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

import com.simplilearn.tables.Flight;
import com.simplilearn.util.HibernateUtil;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Registration() {

    }

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

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    @SuppressWarnings("unchecked")
```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    // print writer
    PrintWriter out = response.getWriter();
    response.setContentType("text/html");

    HttpSession httpsession = request.getSession();
    String flightid = request.getParameter("flightid");
    String firstname = request.getParameter("firstname");
    String lastname = request.getParameter("lastname");
    String email = request.getParameter("email");
    String birthday = request.getParameter("birthday");

    request.setAttribute("firstname", firstname);
    Date dateSqlBirthday = null;
    try {
        dateSqlBirthday = Date.valueOf(birthday);
    } catch (Exception e1) {
        out.print("<a href = \"registration.html\"> Re-Enter Information
</a><br/>");
        out.print("Make sure you are entering your date code in correctly (yyyy-
MM-dd)");
    }

    //email
    if (!(Pattern.matches("[A-Za-z0-9]+(\\.)@(\\.)", email))) {
        out.print("<a href = \"registration.html\"> Re-enter Information
</a><br/>");
        out.print("Make sure you are entering in a valid email address");
    } else if (dateSqlBirthday==null){
        out.print("<a href = \"registration.html\"> Re-enter Information
</a><br/>");
        out.print("Make sure you are entering in a valid date format (yyyy-MM-
dd)");
    } else {
        httpsession.setAttribute("flightid", flightid);
        httpsession.setAttribute("firstname", firstname);
        httpsession.setAttribute("lastname", lastname);
        httpsession.setAttribute("email", email);
        httpsession.setAttribute("birthday", birthday);

        //open a connection
        try {
            SessionFactory sFactory = HibernateUtil.buildSessionFactory();
            Session session = sFactory.openSession();
            session.beginTransaction();

            String query = "select f from Flight f" +

```

```

        " inner join Airline a" +
        " on f.airline=a.id" +
        " and f.numberofSeats > 0";
        if (flightid !=null && flightid.trim().length() > 0) {
            try {
                int flightIdParsed =

Integer.parseInt(flightid);

                String flightIdQuery = " and f.id =" +

                    flightIdParsed;

                query = query.concat(flightIdQuery);
            } catch (Exception e){
                out.println("<h1>You must enter a valid

flight id<h1>");
            }
        }

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

if (flights.size() > 0) {
    out.print("<a href=\"index.html\">Home</a>");
    out.println("<h1>Confirm Flight Details: </h1>");
    for(Flight i:flights) {
        if (i.getNumberOfSeats() > 0) {
            out.println("Flight ID: " + i.getId() + "<br/>");
            out.println("Airline: " + i.getAirline().getAirline().toString() +

"<br/>");

            out.println("Number Of Seats: " + i.getNumberOfSeats() + "<br/>");
            out.println("Source " + i.getSource() + "<br/>");
            out.println("Desintation:" + i.getDestination() + "<br/>");
            out.println("Departure Date: "+i.getDateOfDeparture() + "<br/>");
            out.println("Arrival Date: " + i.getDateOfArrival() + "<br/>");
            out.println("<h1>Your Total is: Rs. " + i.getPrice() + "<h1><br/>");
            out.println("<hr>");
        }
    }
    request.getRequestDispatcher("payment.html").include(request, response);
} else {
    out.print("<a href=\"index.html\">Home</a>");
    out.print("<h1>You must enter a valid flight id!<h1>");
}
session.close();
out.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}

```

SearchAllFlights.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Date;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

import com.simplilearn.tables.Flight;
import com.simplilearn.util.HibernateUtil;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SearchFlights() {
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    @SuppressWarnings({ "null", "unchecked" })
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // print writer
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");

        String source = request.getParameter("source");
        String destination = request.getParameter("destination");
        String price = request.getParameter("price");
        String date = request.getParameter("date");

        //open a connection
        try {
```

```

SessionFactory sFactory = HibernateUtil.buildSessionFactory();
Session session = sFactory.openSession();
session.beginTransaction();

String query = "select f from Flight f" +
    " inner join Airline a" +
    " on f.airline=a.id" +
    " and f.numberOfSeats > 0";
if (source !=null && source.trim().length() > 0) {
    String sourceQuery = " and f.source = '" + source +
        "';

    query = query.concat(sourceQuery);
}
if (destination != null && destination.trim().length() > 0) {
    String destinationQuery = " and f.destination = '" +
        destination + "'";

    query = query.concat(destinationQuery);
}
if (price !=null && price.trim().length() > 0) {
    try {
        Double priceParsed =
            Double.parseDouble(price);

        String priceQuery = " and f.price <= " +
            priceParsed;

        query = query.concat(priceQuery);
    } catch (Exception e){
        out.println("<h1>You must enter a valid
        number for price<h1>");
    }
}
if (date !=null && date.trim().length()> 0) {
    try {
        Date dateSql=Date.valueOf(date);
        String dateQuery = " and f.dateOfDeparture
        >= '" + dateSql + "'";

        query = query.concat(dateQuery);
    } catch (Exception e1) {
        out.println("<h1>You must enter the date
        correctly into yyyy-MM-dd format<h1>");
    }
}
System.out.println(query);
List<Flight> flights = session.createQuery(query).list();

if (flights.size() > 0) {
    out.print("<a href='\"index.html\"'>Home</a>");
    out.println("<h1>List of all Flights: </h1>");
    out.println("<style> table,th,td { border : 1px solid black ; padding :15px;}
    </style>");
}

```

```

        out.println("<table>");
        out.println("<tr>");
            out.println("<th>"); out.println("Flight ID"); out.println("</th>");
            out.println("<th>"); out.println("Airline"); out.println("</th>");
            out.println("<th>"); out.println("Number Of Seats");

out.println("</th>");

            out.println("<th>"); out.println("Source"); out.println("</th>");
            out.println("<th>"); out.println("Desintation"); out.println("</th>");
            out.println("<th>"); out.println("Departure Date");

out.println("</th>");

            out.println("<th>"); out.println("Arrival Date"); out.println("</th>");
            out.println("<th>"); out.println("Price"); out.println("</th>");
        out.println("</tr>");
        for(Flight i:flights) {
            if (i.getNumberOfSeats() > 0) {
                out.println("<tr>");
                    out.println("<td>"); out.println(i.getId());

out.println("</td>");

                    out.println("<td>");
out.println(i.getAirline().getAirline().toString()); out.println("</td>");
                    out.println("<td>");
out.println(i.getNumberOfSeats()); out.println("</td>");
                    out.println("<td>"); out.println(i.getSource());

out.println("</td>");

                    out.println("<td>"); out.println(i.getDestination());

out.println("</td>");

                    out.println("<td>");
out.println(i.getDateOfDeparture()); out.println("</td>");
                    out.println("<td>"); out.println(i.getDateOfArrival());

out.println("</td>");

                    out.println("<td>"); out.println(i.getPrice());

out.println("</td>");

                out.println("</tr>");
            }
        }
        out.println("</table>");
        request.getRequestDispatcher("registration.html").include(request,
response);

        } else {
            out.print("<a href=\"index.html\">Home</a>");
            out.print("<h1>Sorry there are no flights currently!</h1>");
        }

        session.close();
        out.close();

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

```



```

    }

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

```

SeeAllFlights.java:

```

package com.simplilearn.servlets.users;

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

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

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

import com.simplilearn.tables.Flight;
import com.simplilearn.util.HibernateUtil;

/**
 * Servlet implementation class SeeAllFlights
 */
@WebServlet("/see-all-flights")
public class SeeAllFlights extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SeeAllFlights() {

    }

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

```

```

        @SuppressWarnings({ "unused", "unchecked" })
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            // print writer
            PrintWriter out = response.getWriter();
            response.setContentType("text/html");

            String source = request.getParameter("source");
            String destination = request.getParameter("destination");
            String price = request.getParameter("price");
            String date = request.getParameter("date");

            //open a connection
            try {
                SessionFactory sFactory =
HibernateUtil.buildSessionFactory();

                Session session = sFactory.openSession();
                session.beginTransaction();

                String query = "select f from Flight f" +
                    " inner join Airline a" +
                    " on f.airline=a.id" +
                    " and f.numberofSeats > 0";

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

                if (flights.size() > 0) {
                    out.print("<a href='\"index.html\"'>Home</a>");
                    out.println("<h1>List of all Flights: </h1>");
                    out.println("<style> table,th,td { border : 1px solid black ;
padding :15px;} </style>");

                    out.println("<table>");
                    out.println("<tr>");
                        out.println("<th>"); out.println("Flight ID");

                        out.println("<th>"); out.println("Airline");

                        out.println("<th>"); out.println("Number Of Seats");

                        out.println("<th>"); out.println("Source");

                        out.println("<th>"); out.println("Desintation");

                        out.println("<th>"); out.println("Departure Date");

                        out.println("<th>"); out.println("Arrival Date");

                        out.println("<th>"); out.println("Price");
                }
            }
        }
    }
}

```

```

        out.println("</tr>");
        for(Flight i:flights) {
            if (i.getNumberOfSeats() > 0) {
                out.println("<tr>");
                out.println("<td>");
out.println(i.getId()); out.println("</td>");
                out.println("<td>");
out.println(i.getAirline().getAirline().toString()); out.println("</td>");
                out.println("<td>");
out.println(i.getNumberOfSeats()); out.println("</td>");
                out.println("<td>");
out.println(i.getSource()); out.println("</td>");
                out.println("<td>");
out.println(i.getDestination()); out.println("</td>");
                out.println("<td>");
out.println(i.getDateOfDeparture()); out.println("</td>");
                out.println("<td>");
out.println(i.getDateOfArrival()); out.println("</td>");
                out.println("<td>");
out.println(i.getPrice()); out.println("</td>");
                out.println("</tr>");
            }
        }
        out.println("</table>");

        request.getRequestDispatcher("registration.html").include(request, response);
    } else {
        out.print("<a href=\"index.html\">Home</a>");
        out.print("<h1>Sorry there are no flights
currently!<h1>");
    }

    session.close();
    out.close();

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

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

```

setupDB.sql:

```
use flyawaydb;
create table airline(
    a_id int NOT NULL AUTO_INCREMENT,
    airline varchar(255) NOT NULL,
    PRIMARY KEY (a_id)
);
create table flight(
    f_id int(11) NOT NULL AUTO_INCREMENT,
    number_of_seats int(11) NOT NULL,
    price decimal(10,2) NOT NULL,
    source varchar(255) NOT NULL,
    destination varchar(255) NOT NULL,
    date_of_departure date NOT NULL,
    date_of_arrival date NOT NULL,
    a_id int(11) NOT NULL,
    PRIMARY KEY (f_id),
    CONSTRAINT airline_fk FOREIGN KEY (a_id) REFERENCES airline (a_id)
);

create table person(
    per_id int(11) NOT NULL AUTO_INCREMENT,
    first_name varchar(255) NOT NULL,
    last_name varchar(255) NOT NULL,
    email varchar(255) NOT NULL,
    birthday date NOT NULL,
    time_booked DATETIME default CURRENT_TIMESTAMP NOT NULL,
    PRIMARY KEY (per_id)
);
create table personflight(
    per_id int(11) NOT NULL,
    f_id int(11) NOT NULL,
    PRIMARY KEY (per_id, f_id),
    CONSTRAINT person_fk FOREIGN Key (per_id) REFERENCES person (per_id),
    CONSTRAINT flights_fk FOREIGN Key (f_id) REFERENCES flight (f_id)
);
create table payment(
    pay_id int(11) NOT NULL AUTO_INCREMENT,
    card_number varchar(255) NOT NULL,
    expiration DATE NOT NULL,
    security_code int(11) NOT NULL,
    card_fname varchar(255) NOT NULL,
    card_lname varchar(255) NOT NULL,
    per_id int(11) NOT NULL,
    PRIMARY KEY (pay_id),
    CONSTRAINT person_payment_fk FOREIGN KEY (per_id) REFERENCES person (per_id)
);
create table admin(
    admin_id int (11) NOT NULL AUTO_INCREMENT,
```

```

first_name varchar(255) NOT NULL,
last_name varchar(255) NOT NULL,
email varchar(255) NOT NULL,
password varchar(255) NOT NULL,
PRIMARY KEY (admin_id)
);

INSERT INTO airline ( airline)
values ("AirIndia"), ("SpiceJet"), ("Indigo"), ("Vistara"), ("AirAsia"),("Emirates"),("Qatar Airways");

```

```

INSERT INTO flight (number_of_seats, price, source, destination, date_of_departure,
date_of_arrival, a_id)
values (180, 500, "Hyderabad", "Chennai", "2022-10-20", "2022-10-20",
(select a_id from airline where airline = "AirIndia" LIMIT 1)),
(10, 1200, "Hyderabad", "Delhi", "2022-10-23", "2022-10-23",
(select a_id from airline where airline = "SpiceJet"LIMIT 1)),
(150, 200, "Hyderabad", "Kolkata", "2022-10-21", "2022-10-21",
(select a_id from airline where airline = "Indigo"LIMIT 1)),
(20, 150, "Tirupathi", "Hyderabad", "2022-10-20", "2022-10-20",
(select a_id from airline where airline = "Indigo"LIMIT 1)),
(120, 700, "India", "Dubai", "2022-10-24", "2022-10-25",
(select a_id from airline where airline = "Emirates"LIMIT 1)),
(123, 250, "India", "Qatar", "2022-10-22", "2022-10-23",
(select a_id from airline where airline = "Qatar Airways"LIMIT 1));

```

```

INSERT INTO person (first_name, last_name, email, birthday)
values ("P" , "Suresh" , "pablo@gmail.com", "1990-02-01");

```

```

SET FOREIGN_KEY_CHECKS=0;
INSERT INTO personflight (per_id, f_id)
values (1, 3),
(2, 2),
(3, 1),
(4, 5),
(5, 2),
(6, 3),
(7, 6),
(8, 7);

```

```

INSERT INTO payment (card_number, expiration, security_code, card_fname, card_lname, per_id)
values ( "1234 5678 1021 9182" , "2022-10-31", 123, "Steve", "Smith", 1);
INSERT INTO admin (first_name, last_name, email, password)
values ("Tarun", "P", "tarun@gmail.com", "Msd@7781");

```

HibernateUtil.java:

```
package com.simplilearn.util;

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

public class HibernateUtil {

    private static SessionFactory factory;

    public static SessionFactory buildSessionFactory() {

        // 1. Load Configuration
        StandardServiceRegistry sService = new StandardServiceRegistryBuilder()
            .configure("hibernate.annotation.cfg.xml").build();
        Metadata meta = new MetadataSources(sService).getMetadataBuilder().build();

        // 2. Create Session factory
        factory = meta.getSessionFactoryBuilder().build();

        return factory;
    }
}
```

Hibernate.annotation.cfg.xml:

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

<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/flyawaydb</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">Msdhoni@7781</property>
        <property name="hibernate.connection.pool_size">1</property>
        <property name="hibernate.current_session_context_class">thread</property>
        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.id.new_generator_mappings">>false</property>

        <mapping class="com.simplilearn.tables.Admin" />
        <mapping class="com.simplilearn.tables.Airline" />
        <mapping class="com.simplilearn.tables.Flight" />
    </session-factory>
</hibernate-configuration>
```

```

        <mapping class="com.simplilearn.tables.Payment" />
        <mapping class="com.simplilearn.tables.Person" />
    </session-factory>
</hibernate-configuration>

```

HTML FILES:

Index.html:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome To Fly Away Ticket Booking Portal</title>
</head>
<body>
    <a href="login.html">Admin Login</a>
    <hr>
    <h1>Search For Flights</h1>
    <form name="searchForFlights" method="GET" action="search-flights">
        Flying From : <input type="text" name="source"> <br/>
        Flying To : <input type="text" name="destination"> <br/>
        Your Max Price : <input type="number" name="price"> <br/>
        Flights On or After Date : <input type="date" name="date">
    <br/>
    <br/>
    <input type="submit" value="Search for flights!">
</form>
    <form name="seeAllFlights" method="GET" action="see-all-flights">
        <input type="submit" value="See All Flights">
    </form>

</body>
</html>

```

Login.html:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Admin Login</title>
</head>
<body>
    <a href="index.html">Home</a>

    <h1>Admin Login Page</h1>
    <form name="loginForm" method="POST" action="login">
        <td> Username: <input type="text" name="username"> <br/></td>
        <td> Password : <input type="password" name="password"> <br/>
    </td>
    <br/>
    <td> <input type="submit" value="Login"></td>
    </form>

</body>
</html>

```

Dashboard.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Admin Dashboard</title>
</head>
<body>
  <a href="index.html">Home</a> |
  <a href="index.html">Logout</a> |
  <a href="changepassword.html">Change Password</a> |
  <a href="index.html">Source/Destination Master View</a> |
  <a href="see-all-flights?">Flight Master View</a><br/>
  <hr>
</body>
</html>
```

ChangePassword.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Admin Login Page</title>
</head>
<body>
  <a href="dashboard.html">Dashboard</a>

  <h1>Change Password</h1>
  <form name="change password" method="POST" action="change-password">
    New Password : <input type="password" name="password"> <br/>
    Confirm New Password: <input type="password"
name="confirmpassword"> <br/>
    <input type="submit" value="Change password">
  </form>

</body>
</html>
```

Passwordverified.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>password changed</title>
</head>
<body>
  <h1>your password has been changed successfully </h1>
  <a href="dashboard.html">Dashboard</a>

</body>
</html>
```

Payment.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
```



```

<title>Payment</title>
</head>
<body>
    <h1>Enter Payment Details</h1>
    <form name="paymentDetails" method="POST" action="pay-online">
        First Name On Card : <input type="text" name="firstname"
placeholder="John" required> <br/>
        Last Name On Card : <input type="text" name="lastname"
placeholder="Smith" required> <br/>
        Card Number: <input type="text" name="cardnumber"
placeholder="1234 1234 4321 9123" required> <br/>
        Expiration (yyyy-MM-dd) : <input type="date" name="expiration"
placeholder="2027-10-31" required> <br/>
        Security Code : <input type="number" name="securitycode"
placeholder="987" required> <br/>
        <br/>
        <input type="submit" value="Pay Ticket">
    </form>

</body>
</html>

```

Registration.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Flight Registration Page</title>
</head>
<body>
    <h1>Flight Registration Page</h1>
    <form name="flightRegistraton" method="POST" action="registration">
        Flight ID : <input type="number" name="flightid" placeholder="
" required> <br/>
        First Name : <input type="text" name="firstname" placeholder="
" required> <br/>
        Last Name : <input type="text" name="lastname" placeholder="
" required> <br/>
        Email : <input type="email" name="email"
placeholder="xxxxxxx@gmail.com" required> <br/>
        Birthday (yyyy-MM-dd) : <input type="date" name="birthday"
placeholder="1990-04-02" required> <br/>
        <br/>
        <input type="submit" value="Submit Flight Registration">
    </form>

</body>
</html>

```