**FlyAway(An Airline Booking Portal)**

**Developer: Jyothi Kothapally**

|  |  |
| --- | --- |
| Author | Jyothi kothapally |
| Date | 03-oct-2021 |
| Version | 1.0 |
| Purpose | Specific documentation |

Version History

Flyaway

Flyaway is a ticket-booking portal that lets people book flights on their website.

# **GitHub Link**

This code for this project is hosted at [kothapallyjyothi/FlyAway (github.com)](https://github.com/kothapallyjyothi/FlyAway).

# **Specifications of the project**

1. MVC model

2. Eclipse

3. maven

4. apache tomcat 9.0

5. Hibernate

6. Mysql - Database

7. searching for flights

8. see all flights

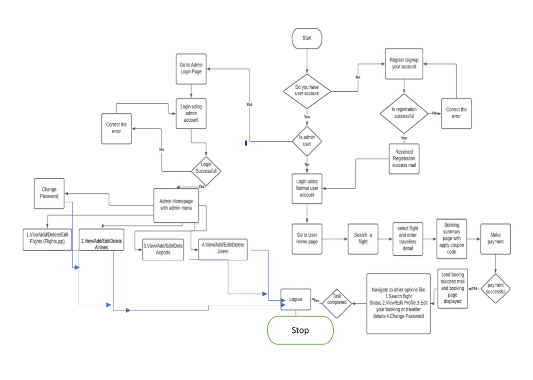
9. admin login

10. payment options

11. conformation option

12. html files: login page, logout, dashboard, search for flights.

# **3.Flow chart of the application**



# **4.Sprint Planning and Task Completion**

The Project is planned to be completed in 3 sprints. Tasks assumed to be completed in the sprint are:

**Sprint 1:**

* Initialized the maven project, using Hibernate annotation and added dependencies i.e.., MySql connector, Hibernate libraries and Servlet API.
* Setting MySQL database environment and tomcat server
* Created E.R. diagram for database and entities relationship.
* Created class for entities and established relationship between them.
* Configured hibernate and mapped classes as entities.
* Populated the data by hibernate

**Sprint 2:**

* Planned the structure of the website and navigation across it.
* Made a Hibernate util. class to use it across the servlets.
* Made a common View page for each class’s view rendering.
* Made servlets for each class list and set up navigation link for them.
* Made an error page errorpage.jsp and set it as the error page by whole website by including it in the header tag  file.

**Sprint 3:**

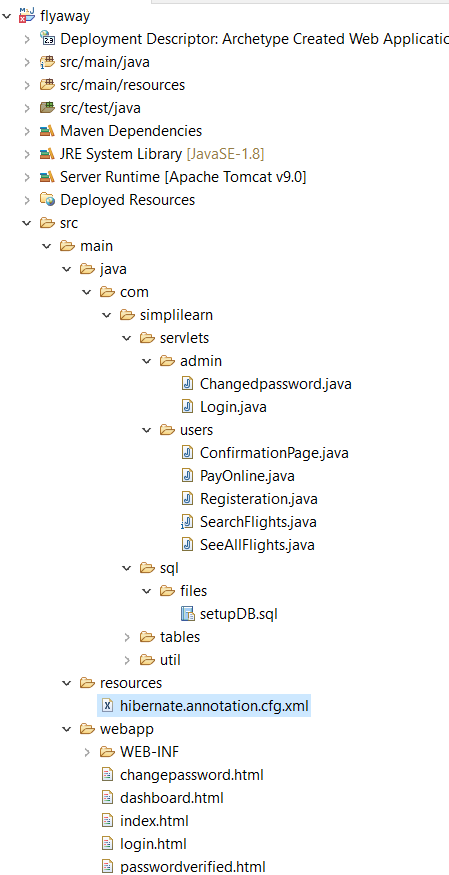
* Made the master list for master list of places for source and destination, master list of airlines
* Also Made A list of flights where each flight has a source, destination, airline, and ticket price
* Set up Servlets for each of the master list to send data as a list.
* Made main.css for applying styles throughout the website.
* Added bootstrap CSS library and jQuery JavaScript library to speedup designing process using ready to use libraries.
* Added & configure SendMail api to send mail on successful registration/ booking.
* Modified the navigation with the help of button’s and added the navigation to home page in the Header Tag.
* Revamped and modified the structure of the data rendering with the additional styles and best practices.

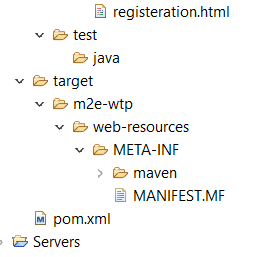
# **5.Core concepts used in the project**

* Server request response object
* HTTP Session, Cookies URL Redirect Hidden Field.
* Collection Framework.
* DateTime object with simpleDateFormat.
* Hibernate JPA, Different types of mapping (one to one, one to many, many to one, many to many).
* Sorting technique using database query as well as using Collection framework comparator methods.
* MVC Framework where View (jsp/Html pages) Controller (Servlet pages), Data Model (Entity classes and interface which interact to backend database using hibernate framework)
* Table Joins, Filters, Subqueries Transaction Control
* Java’s SendMail API to send email.
* Maven and Servlet
* HTML/CSS to design view
* •Web container such as Tomcat server usage deployment of web application and running/configuration
* MySQL Database and SQL Query language.

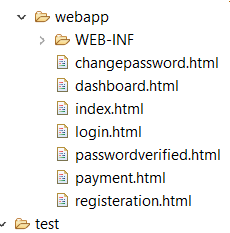
# **6.Project Structure and Code**

Folder structure:

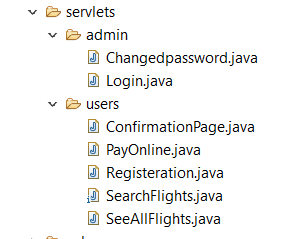




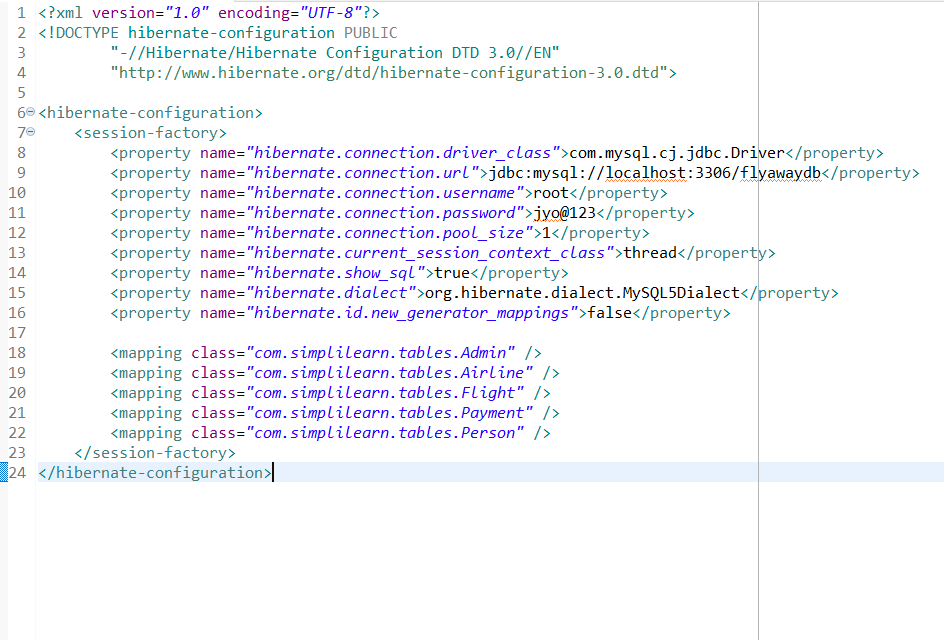
FrontHtml(View):



Servlets (Controller):



Hibernate:



Login.java: code for login flyaway portal , where user login with credentials

package com.simplilearn.servlets.admin;

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 = "12345";

public static String email = "jyo@gmail.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();

}

}

Changedpassword.java: code for changed password if user wants to change the password

package com.simplilearn.servlets.admin;

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 changedpassword

\*/

@WebServlet("/change-password")

public class Changedpassword extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Changedpassword() {

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 ");

out.println("please try again");

}

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();

}

}

Confirmation code: Code for confirmation booking tickets if user wants to book the flights tickets

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 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: code for Payment where user wants to book flights paying online mode

package com.simplilearn.servlets.users;

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: code for Registration ,where user register basic information required for flights booking

package com.simplilearn.servlets.users;

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("/registeration")

public class Registeration extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Registeration() {

}

/\*\*

\* @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 = \"registeration.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 = \"registeration.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 = \"registeration.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: $" + 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();

}

}

}

}

Searchflights.java:

package com.simplilearn.servlets.users;

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("registeration.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("<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("registeration.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 {

}

}

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/>

<br/>

<input type=*"submit"* value=*"Change password"*>

</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> <br/>

<hr>

</body>

</html>

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 Page</title>

</head>

<body>

<a href=*"index.html"*>Home</a>

<h1>Admin Login Page</h1>

<form name=*"loginForm"* method=*"POST"* action=*"login"*>

Username: <input type=*"text"* name=*"username"*> <br/>

Password : <input type=*"password"* name=*"password"*> <br/>

<br/>

<input type=*"submit"* value=*"Login"*>

</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=*"registeration"*>

Flight ID : <input type=*"number"* name=*"flightid"* placeholder=*"12"* required> <br/>

First Name : <input type=*"text"* name=*"firstname"* placeholder=*"John"* required> <br/>

Last Name : <input type=*"text"* name=*"lastname"* placeholder=*"Doe"* required> <br/>

Email : <input type=*"email"* name=*"email"* placeholder=*"johndoe@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>

Pom.xml:

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.simplilearn</groupId>

<artifactId>flyaway</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>flyaway Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.4.1.Final</version>

</dependency>

<!-- mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.21</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

</dependency>

</dependencies>

<build>

<finalName>flyaway</finalName>

<sourceDirectory>src/main/java</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

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** ("Emirates"), ("Delta"), ("United Airlines"), ("Turkish Airlines"), ("Uganda Airlines");

**INSERT** **INTO** flight (number\_of\_seats, price, source, destination, date\_of\_departure, date\_of\_arrival, a\_id)

**values** (180, 500, "ORD", "DBX", "2020-10-20", "2020-10-22",

(**select** a\_id **from** airline **where** airline = "Emirates")),

(10, 1200, "EBB", "LAX", "2020-10-23", "2020-10-24",

(**select** a\_id **from** airline **where** airline = "Uganda Airlines")),

(150, 200, "ORD", "LAX", "2020-10-21", "2020-10-21",

(**select** a\_id **from** airline **where** airline = "Delta")),

(20, 150, "ORD", "LAX", "2020-10-20", "2020-10-21",

(**select** a\_id **from** airline **where** airline = "United Airlines")),

(120, 700, "DBX", "IATA", "2020-10-24", "2020-10-25",

(**select** a\_id **from** airline **where** airline = "Turkish Airlines")),

(123, 250, "DBX", "EBB", "2020-10-22", "2020-10-22",

(**select** a\_id **from** airline **where** airline = "Emirates"));

**INSERT** **INTO** person (first\_name, last\_name, email, birthday)

**values** ("Pablo" , "Smith" , "pablo@gmail.com", "1990-02-01"),

("Pablo" , "Tisasirana" , "pablo@gmail.com", "1990-02-01"),

("Veronica" , "Villagomez" , "veronica@gmail.com", "1990-02-01"),

("Shay" , "Finn" , "shay@gmail.com", "1990-02-01"),

("Nick" , "Lorance" , "nick@gmail.com", "1990-02-01"),

("John" , "Brook" , "john@gmail.com", "1990-02-01"),

("Albert" , "Gaca" , "albert@gmail.com", "1990-02-01"),

("Chris" , "Smith" , "chris@gmail.com", "1990-02-01");

**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, "Pablo", "Smith", 1),

( "1234 5678 1021 9182" , "2022-10-31", 123, "Pablo", "Smith", 2),

( "1234 5678 1021 9182" , "2022-10-31", 123, "Pablo", "Smith", 3),

( "1234 5678 1021 9182" , "2022-10-31", 123, "Pablo", "Smith", 4),

( "1234 5678 1021 9182" , "2022-10-31", 123, "Pablo", "Smith", 5),

( "1234 5678 1021 9182" , "2022-10-31", 123, "Pablo", "Smith", 6),

( "1234 5678 1021 9182" , "2022-10-31", 123, "Pablo", "Smith", 7),

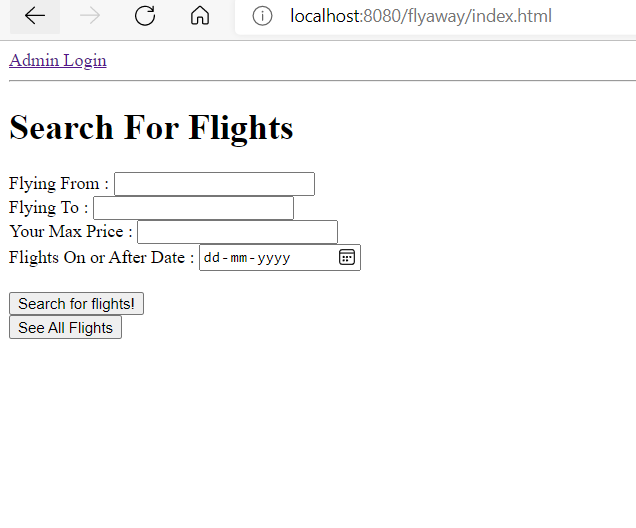
( "1234 5678 1021 9182" , "2022-10-31", 123, "Pablo", "Smith", 8);

**INSERT** **INTO** admin ( first\_name, last\_name, email, password)

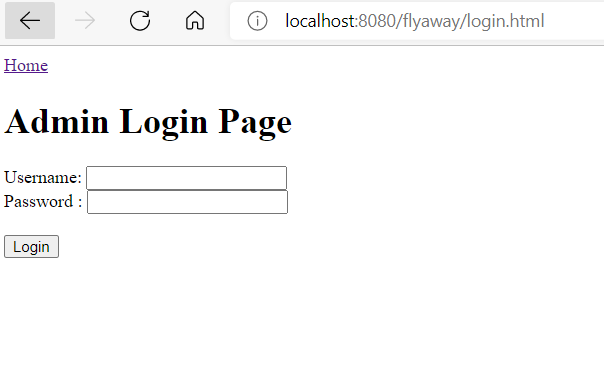
**values** ("Bryan", "Finn", "admin@gmail.com", "admin");

output:

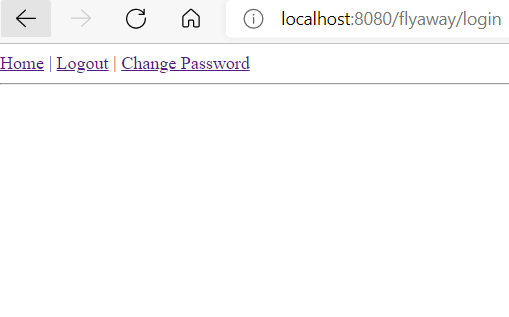
1.Home:



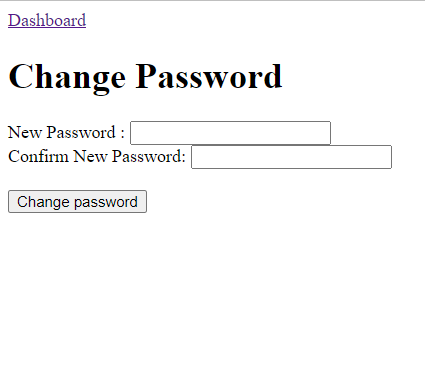
2.Admin login Page



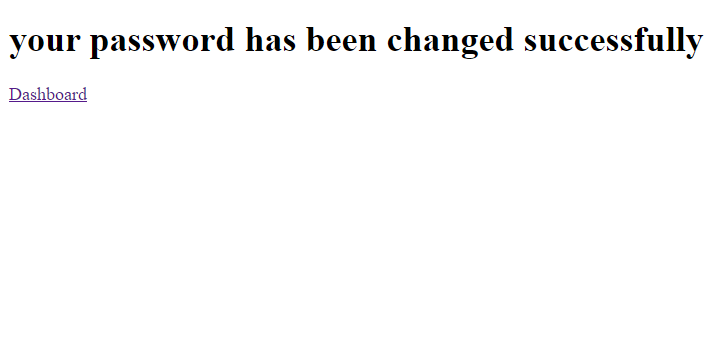
3.Admin Dashboard



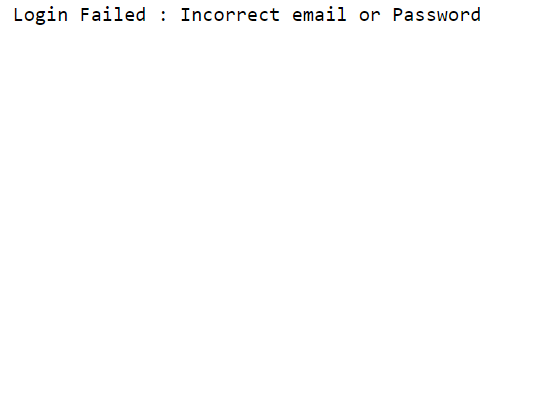
ChangedPassword:



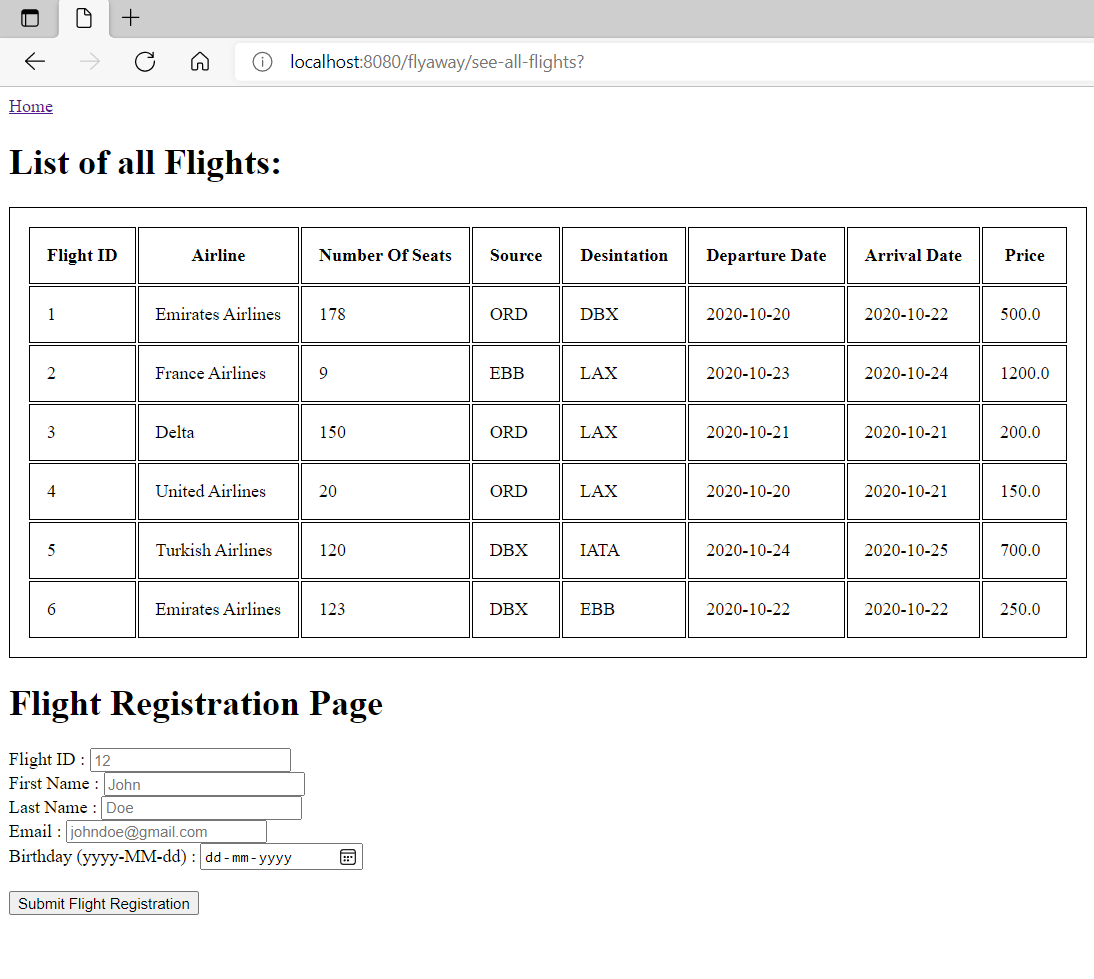
Changed Password success:



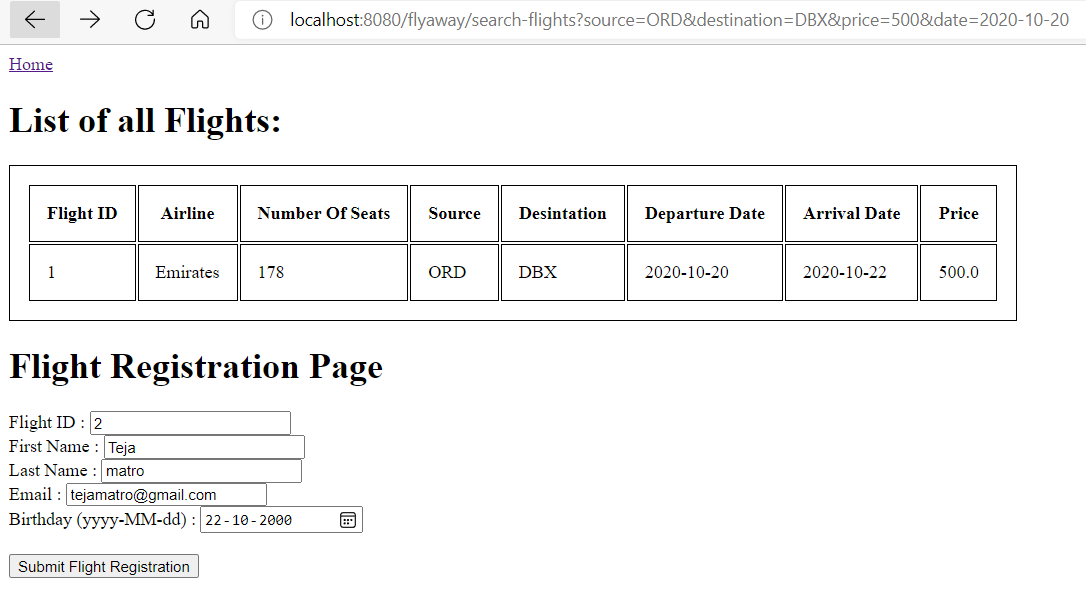
Login Failed:



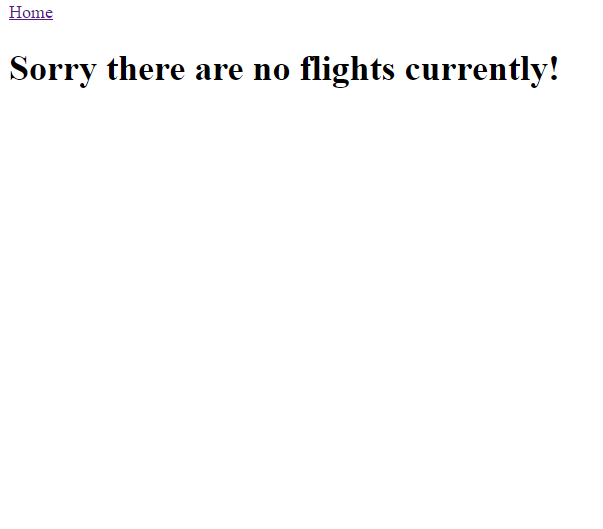
See All flights and Registration:



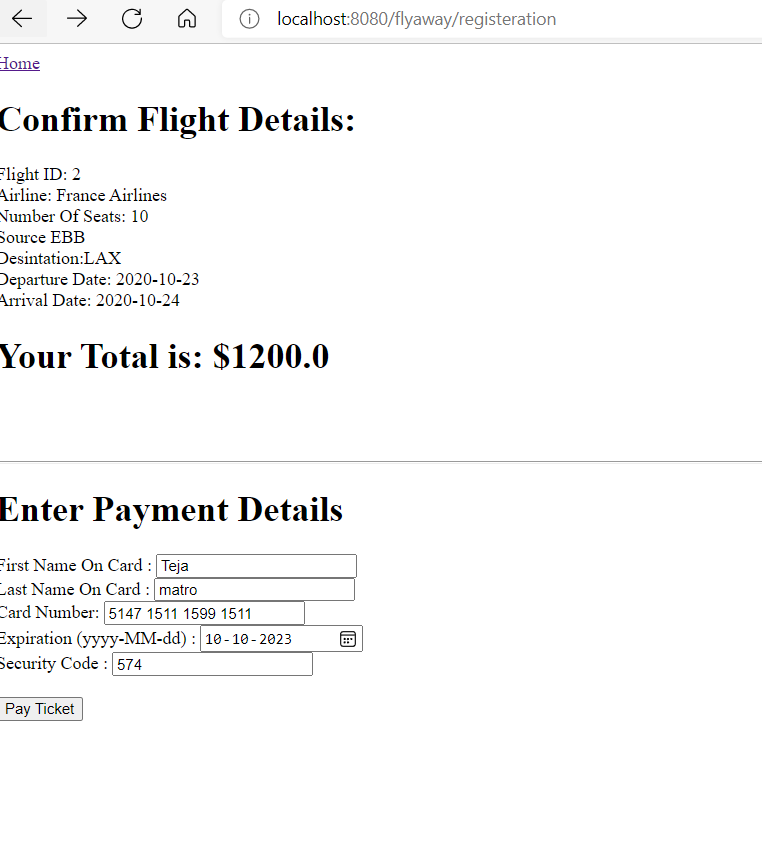
Search for flights :



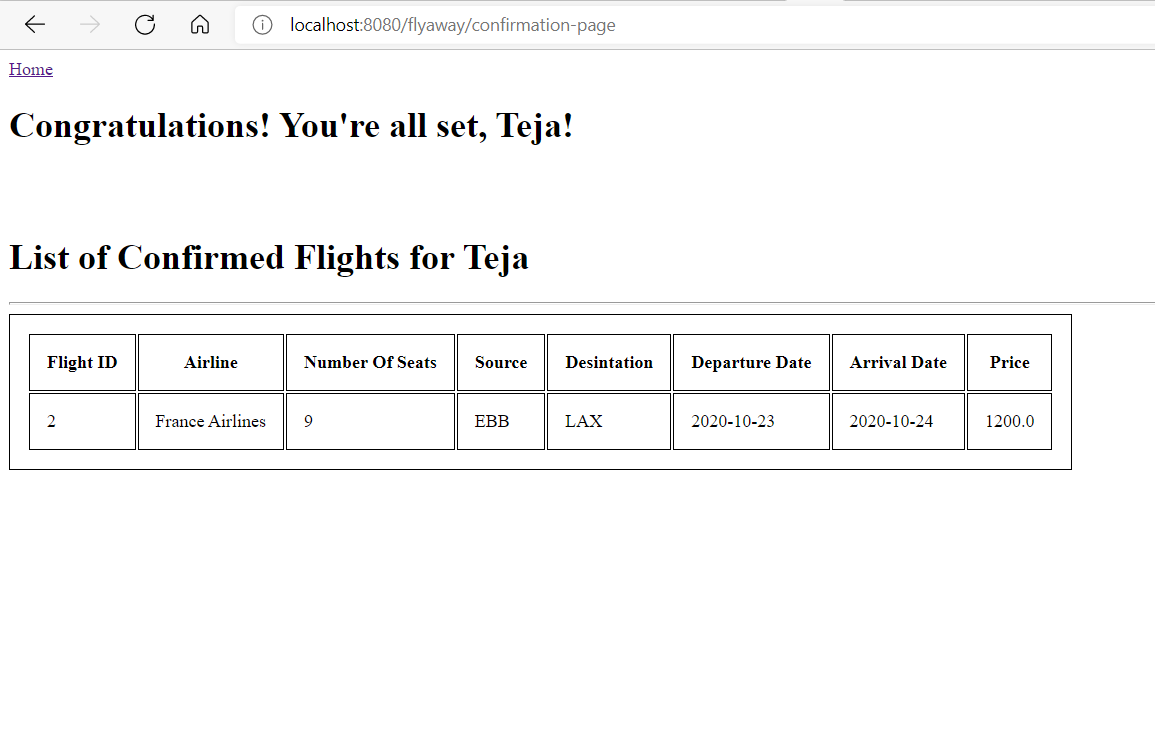
No Flights if we don’t have flights asked date



Registration for flights:



Payment successful:



# **7.Pushing the code to GitHub repository**

Instructions are to be followed:

* First of all, we need account in Github to push the code in GitHub.
* Create account in GitHub.
* Create new repository.
* Open your command prompt and navigate to the folder where you have created your files.

cd

* Initialize repository using the following command:

git init

* Add all the files to your git repository using the following the command:

git add

* Commit the changes using the following command:

git commit . -m <commit message>

* Push the files to the folder you initially created using the following command:

git push -u origin main

# **8.Technologies used**

Apache tomcat server 9.0.52

MySql 8.0.26

Java 1.8

Maven dependency

Hiber

Eclipse/IntelliJ