<https://github.com/malinimurthy123/FlyAwayFlightsProjectwithHibernate.git>

pom.xml

*<?*xml version="1.0" encoding="UTF-8"*?>*<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com</groupId>  
 <artifactId>flyaway</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <name>flyaway</name>  
 <packaging>war</packaging>  
  
 <properties>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <junit.version>5.7.0</junit.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.1</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.23</version>  
 </dependency>  
  
 <dependency>  
 <groupId>log4j</groupId>  
 <artifactId>log4j</artifactId>  
 <version>1.2.17</version>  
 </dependency>  
  
 <dependency>  
 <groupId>com.stripe</groupId>  
 <artifactId>stripe-java</artifactId>  
 <version>10.12.1</version>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
 </dependency>  
  
 <dependency>  
 <groupId>com.paypal.sdk</groupId>  
 <artifactId>rest-api-sdk</artifactId>  
 <version>1.14.0</version>  
 </dependency>  
 <dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>5.4.10.Final</version>  
 </dependency>  
 <dependency>  
 <groupId>org.hibernate.validator</groupId>  
 <artifactId>hibernate-validator</artifactId>  
 <version>6.1.6.Final</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.hibernate.javax.persistence</groupId>  
 <artifactId>hibernate-jpa-2.1-api</artifactId>  
 <version>1.0.0.Final</version>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <version>RELEASE</version>  
 <scope>compile</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-war-plugin</artifactId>  
 <version>3.3.0</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

Userlogin;

package com.flyaway;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.List;  
import java.util.UUID;  
  
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 entity.userdetails;  
import org.hibernate.Session;  
  
  
@WebServlet("/login")  
public class UserLogin extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
 private final String admin = "malini";  
 private final String pass = "murthy";  
  
 public UserLogin() {  
 super();  
  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 response.setContentType("text/html");  
 response.sendRedirect("login.html");  
 }  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
  
 response.setContentType("text/html");  
 PrintWriter out = response.getWriter();  
 request.getRequestDispatcher("navigate.html").include(request, response);  
 String username = request.getParameter("name");  
 String password = request.getParameter("password");  
 Session session = (Session) HibernateUtil.*getSessionFactory*();  
 List<userdetails> list = session.createQuery("from userdetails ").list();  
  
 if (username.equals(admin) && password.equals(pass)) {  
 HttpSession s = request.getSession(true);  
 s.setAttribute("uname", username);  
 s.setAttribute("userId", UUID.*randomUUID*().toString());  
 response.sendRedirect("adminsuccess.jsp");  
 } else {  
 boolean flag1 = false, flag2 = false;  
 String name = "";  
 for (userdetails user : list) {  
 if (user.getName().equalsIgnoreCase(username)) {  
 if (user.getPassword().equals(password)) {  
 flag1 = true;  
 name = user.getName();  
 break;  
 }  
 flag2 = true;  
 break;  
 }  
 }  
 if (flag1 == true) {  
 HttpSession s = request.getSession(true);  
 s.setAttribute("uname", name);  
 s.setAttribute("userId", UUID.*randomUUID*().toString());  
 s.setMaxInactiveInterval(2 \* 60 \* 60);  
 response.sendRedirect("getflightdetails");  
 } else if (flag2 == true) {  
 out.println("<font color=blue><br><br><h1>User Password is wrong!</h1></font>");  
 request.getRequestDispatcher("login.html").include(request, response);  
 } else {  
  
 out.println("<font color=red><br><br><h1>User does not exists!.. Register First..!<h1></font>");  
 request.getRequestDispatcher("registration.html").include(request, response);  
 }  
 }  
 }  
}

registration:

package com.flyaway;  
  
import java.io.IOException;  
  
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 entity.userdetails;  
  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
  
@WebServlet("/Registration")  
public class Registration extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 public Registration() {  
 super();  
  
 }  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 response.setContentType("text/html");  
 response.sendRedirect("registration.html");  
 }  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 response.setContentType("text/html");  
 String name=request.getParameter("name");  
 String password=request.getParameter("password");  
 String email=request.getParameter("email");  
 userdetails user=new userdetails(name,password,email);  
 Session session= (Session) HibernateUtil.*getSessionFactory*();  
 Transaction tx=session.beginTransaction();  
 session.save(user);  
 tx.commit();  
 *//theSession.close();* response.sendRedirect("registrationSuccess.jsp");  
 }  
  
}

logout:

package com.flyaway;  
  
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;  
import javax.servlet.http.HttpSession;  
  
import org.hibernate.Session;  
  
  
@WebServlet("/logout")  
public class Logout extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
 Session theSession;  
  
 public Logout() {  
 super();  
  
 }  
  
 protected void dSoGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 response.setContentType("text/html");  
 PrintWriter out=response.getWriter();  
 request.getRequestDispatcher("navigate.html").include(request, response);  
 HttpSession session=request.getSession();  
 session.removeAttribute("id");  
 session.invalidate();  
 Session ses= (Session) HibernateUtil.*getSessionFactory*();  
 ses.flush();  
 ses.clear();  
 *//ses.close();* out.println("<br><h3>You are successfully logged out</h3>");  
 }  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 doGet(request, response);  
 }  
  
}

HibernateUtil

package com.flyaway;  
  
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 final SessionFactory *sessionFactory*;  
  
 static {  
 try {  
 StandardServiceRegistry standardRegistry = new StandardServiceRegistryBuilder()  
 .configure("hibernate.cfg.xml").build();  
 Metadata metaData = new MetadataSources(standardRegistry).getMetadataBuilder().build();  
 *sessionFactory* = metaData.getSessionFactoryBuilder().build();  
 } catch (Throwable th) {  
 throw new ExceptionInInitializerError(th);  
 }  
 }  
  
 public static SessionFactory getSessionFactory() {  
  
 return *sessionFactory*;  
 }  
  
  
}

flight-details:

package com.flyaway;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.sql.SQLException;  
import java.util.List;  
import javax.servlet.RequestDispatcher;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
import entity.flightdetail;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
  
@WebServlet("/flight-details")  
public class FlightDetails extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
 private FlightDAO FlightDAO;  
  
 public void init(){  
 FlightDAO= new FlightDAO();  
  
 }  
  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
  
  
 String source=request.getParameter("source");  
 String destination=request.getParameter("destination");  
 int travellers=Integer.*parseInt*(request.getParameter("travellers"));  
 String date=request.getParameter("date");  
 flightdetail flight=new flightdetail(source,destination,travellers,date);  
  
*/\*  
 Session session= HibernateUtil.getSessionFactory().openSession();  
 Transaction tx=session.beginTransaction();  
  
 // using HQL  
 List<flightdetail> listFlight = session.createQuery("from flightdetail", flightdetail.class).getResultList();  
 //session.save(flight);  
 //tx.commit();  
  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<b>Flight Listing</b><br>");  
 for (flightdetail flightdetail : listFlight) {  
 System.out.println(flightdetail.getId());  
 out.println("ID: " + String.valueOf(flightdetail.getId()) + ", source: " + flightdetail.getSource() +  
 ",Destination: " + String.valueOf(flightdetail.getDestination()) + ", flightdate: " + flightdetail.getDate() + "<br>");  
 }  
 out.println("<a href='index.jsp'>Return to homepage</a><br>");  
 out.println("</body></html>");  
 request.setAttribute("listFlight", listFlight);  
 request.getRequestDispatcher("/flight-details.jsp").forward(request, response);\*/* }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 String action = request.getServletPath();  
 String source=request.getParameter("source");  
 String destination=request.getParameter("destination");  
 int travellers=Integer.*parseInt*(request.getParameter("travellers"));  
 String date=request.getParameter("date");  
 flightdetail flight=new flightdetail(source,destination,travellers,date);  
  
  
 Session session= HibernateUtil.*getSessionFactory*().openSession();  
 Transaction tx=session.beginTransaction();  
  
  
 *// using HQL* List<flightdetail> listFlight = session.createQuery("from flightdetail", flightdetail.class).getResultList();  
 *//session.save(flight);  
 //tx.commit();* PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 out.println("<b>Flight Listing</b><br>");  
 for (flightdetail flightdetail : listFlight) {  
 System.*out*.println(flightdetail.getId());  
 out.println("ID: " + String.*valueOf*(flightdetail.getId()) + ", source: " + flightdetail.getSource() +  
 ",Destination: " + String.*valueOf*(flightdetail.getDestination()) + ", flightdate: " + flightdetail.getDate() + "<br>");  
 }  
 out.println("<a href='index.jsp'>Return to homepage</a><br>");  
 out.println("</body></html>");  
 request.setAttribute("listFlight", listFlight);  
 request.getRequestDispatcher("/flight-details.jsp").forward(request, response);  
  
 try {  
 listFlight(request, response);  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
  
 }  
  
 public void listFlight(HttpServletRequest request, HttpServletResponse response)  
 throws SQLException, IOException, ServletException {  
 List < flightdetail > listFlight = FlightDAO.getAllFlight();  
 request.setAttribute("listFlight", listFlight);  
 RequestDispatcher dispatcher = request.getRequestDispatcher("user-list.jsp");  
 dispatcher.forward(request, response);  
 }  
}

FlightDAO:

package com.flyaway;  
  
import entity.flightdetail;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import java.util.List;  
  
public class FlightDAO {  
  
 */\*\*  
 \* Get all Flights  
 \* @return  
 \*/* @SuppressWarnings("unchecked")  
  
 Session session= HibernateUtil.*getSessionFactory*().openSession();  
 Transaction tx=session.beginTransaction();  
  
 public List<flightdetail> getAllFlight() {  
  
 Transaction transaction = null;  
 List < flightdetail > listOfFlight = null;  
 try (Session session = HibernateUtil.*getSessionFactory*().openSession()) {  
 *// start a transaction* transaction = session.beginTransaction();  
 *// get an user object  
  
 // using HQL* listOfFlight = session.createQuery("from flightdetail", flightdetail.class).getResultList();  
  
 transaction.commit();  
 } catch (Exception e) {  
 if (transaction != null) {  
 transaction.rollback();  
 }  
 e.printStackTrace();  
 }  
 return listOfFlight;  
 }  
}

Available Flights:

package com.flyaway;  
  
import java.io.IOException;  
  
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;  
  
@WebServlet("/AvailableFlights")  
public class AvailableFlights extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 public AvailableFlights() {  
 super();  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 response.setContentType("text/html");  
 response.sendRedirect("login.html");  
 }  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 String st=request.getParameter("airlines");  
 int id=Integer.*parseInt*(st);  
 HttpSession idValue=request.getSession(true);  
 idValue.setAttribute("id",id);  
 doGet(request,response);  
  
 }  
  
}

AuthorizePaymentServlet

package com.flyaway;  
  
import java.io.IOException;  
 import javax.servlet.ServletException;  
 import javax.servlet.annotation.WebServlet;  
 import javax.servlet.http.\*;  
  
 import com.paypal.base.rest.PayPalRESTException;  
  
@WebServlet("/authorize\_payment")  
public class AuthorizePaymentServlet extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 public AuthorizePaymentServlet() {  
 }  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 String product = request.getParameter("product");  
 String subtotal = request.getParameter("subtotal");  
 String shipping = request.getParameter("shipping");  
 String tax = request.getParameter("tax");  
 String total = request.getParameter("total");  
  
 OrderDetail orderDetail = new OrderDetail(product, subtotal, shipping, tax, total);  
  
 try {  
 PaymentServices paymentServices = new PaymentServices();  
 String approvalLink = paymentServices.authorizePayment(orderDetail);  
  
 response.sendRedirect(approvalLink);  
  
 } catch (PayPalRESTException ex) {  
 request.setAttribute("errorMessage", ex.getMessage());  
 ex.printStackTrace();  
 request.getRequestDispatcher("error.jsp").forward(request, response);  
 }  
 }  
}

bookingconfirmation

package com.flyaway;  
  
import entity.booking;  
import org.hibernate.HibernateException;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.cfg.Configuration;  
  
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 java.io.IOException;  
import java.io.PrintWriter;  
import java.util.logging.Logger;  
  
@WebServlet("/bookingconfirmation")  
public class bookingconfirmation extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
 private final static Logger *logger* = Logger.*getLogger*("bookingconfirmation.class");  
 public bookingconfirmation() {  
 super();  
  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
 response.setContentType("text/html");  
 response.sendRedirect("booking.jsp");  
 }  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 Logger.*getLogger*("inside post method");  
  
 PrintWriter out = response.getWriter();  
 out.println("<html><body>");  
 *//boolean status = false;* try {  
  
 *//response.setContentType("text/html");* Session session = HibernateUtil.*getSessionFactory*().openSession();  
 Logger.*getLogger*("inside post method");  
 int min = 1;  
 int max = 200;  
 double price = 1200.00;  
 int randomValue = (int) Math.*floor*(Math.*random*() \* (max - min + 1) + min);  
 System.*out*.println(randomValue);  
 String bookingconfirmation = String.*valueOf*(randomValue);  
  
 *//String booking=request.getParameter("booking");* String fname = request.getParameter("fname");  
 String lname = request.getParameter("lname");  
 String totalpassengers = request.getParameter("totalpassengers");  
 *//double price = Double.parseDouble(request.getParameter("price"));* System.*out*.println(randomValue);  
  
  
 booking booking = new booking();  
 *//booking,fname,lname,totalpassengers,price  
 //booking.setBookingconfirmation(booking);* booking.setFname(fname);  
 booking.setLname(lname);  
 booking.setTotalpassengers(totalpassengers);  
 booking.setPrice(price);  
 Transaction tx = session.beginTransaction();  
 session.save(booking);  
 tx.commit();  
 *//theSession.close();* Logger.*getLogger*("save");  
 System.*out*.println("Object Saved");  
 out.println("<p>booking confirmed..</p>");  
 out.println("<html><body>");  
 out.println("<b>Flight Booking Confirmed</b><br>");  
  
 response.sendRedirect("bookingconfirmation.jsp");  
  
 } catch (HibernateException e) {  
 e.printStackTrace();  
 } catch (NumberFormatException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
}

ExecutePaymentServlet

package com.flyaway;  
  
import java.io.IOException;  
 import javax.servlet.ServletException;  
 import javax.servlet.annotation.WebServlet;  
 import javax.servlet.http.\*;  
  
 import com.paypal.api.payments.\*;  
 import com.paypal.base.rest.PayPalRESTException;  
  
@WebServlet("/execute\_payment")  
public class ExecutePaymentServlet extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 public ExecutePaymentServlet() {  
 }  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 String paymentId = request.getParameter("paymentId");  
 String payerId = request.getParameter("PayerID");  
  
 try {  
 PaymentServices paymentServices = new PaymentServices();  
 Payment payment = paymentServices.executePayment(paymentId, payerId);  
  
 PayerInfo payerInfo = payment.getPayer().getPayerInfo();  
 Transaction transaction = payment.getTransactions().get(0);  
  
 request.setAttribute("payer", payerInfo);  
 request.setAttribute("transaction", transaction);  
  
 request.getRequestDispatcher("receipt.jsp").forward(request, response);  
  
 } catch (PayPalRESTException ex) {  
 request.setAttribute("errorMessage", ex.getMessage());  
 ex.printStackTrace();  
 request.getRequestDispatcher("error.jsp").forward(request, response);  
 }  
 }  
}

PaymentServices

package com.flyaway;  
  
import java.util.\*;  
  
import com.paypal.api.payments.\*;  
import com.paypal.base.rest.\*;  
  
public class PaymentServices {  
 private static final String *CLIENT\_ID* = "ASUCNg-b\_aNhuCS7fyhoiBKk1kOlkMoAjgcLxnmKO2T9ukFhl2ovu-1sTyMhF2R-Zhfmkrmfg8hHCvKk";  
 private static final String *CLIENT\_SECRET* = "EOlHvw5nj7Y4w5ByUTcF7gHzWzuS4zMIGbPkhBpVKRLOEkWIfm0sGME0CatXSDoSOTF9MKNUaPvdQmpW";  
 private static final String *MODE* = "sandbox";  
  
 public String authorizePayment(OrderDetail orderDetail)  
 throws PayPalRESTException {  
  
 Payer payer = getPayerInformation();  
 RedirectUrls redirectUrls = getRedirectURLs();  
 List<Transaction> listTransaction = getTransactionInformation(orderDetail);  
  
 Payment requestPayment = new Payment();  
 requestPayment.setTransactions(listTransaction);  
 requestPayment.setRedirectUrls(redirectUrls);  
 requestPayment.setPayer(payer);  
 requestPayment.setIntent("authorize");  
  
 APIContext apiContext = new APIContext(*CLIENT\_ID*, *CLIENT\_SECRET*, *MODE*);  
  
 Payment approvedPayment = requestPayment.create(apiContext);  
  
 return getApprovalLink(approvedPayment);  
  
 }  
  
 private Payer getPayerInformation() {  
 Payer payer = new Payer();  
 payer.setPaymentMethod("paypal");  
  
 PayerInfo payerInfo = new PayerInfo();  
 payerInfo.setFirstName("Malini")  
 .setLastName("Murthy")  
 .setEmail("krmalini25@gmail.com");  
  
 payer.setPayerInfo(payerInfo);  
  
 return payer;  
 }  
  
 private RedirectUrls getRedirectURLs() {  
 RedirectUrls redirectUrls = new RedirectUrls();  
 redirectUrls.setCancelUrl("https://www.sandbox.paypal.com/PaypalTest/cancel.html");  
 redirectUrls.setReturnUrl("https://www.sandbox.paypal.com/PaypalTest/review\_payment");  
  
 return redirectUrls;  
 }  
  
 private List<Transaction> getTransactionInformation(OrderDetail orderDetail) {  
 Details details = new Details();  
 details.setShipping(orderDetail.getShipping());  
 details.setSubtotal(orderDetail.getSubtotal());  
 details.setTax(orderDetail.getTax());  
  
 Amount amount = new Amount();  
 amount.setCurrency("USD");  
 amount.setTotal(orderDetail.getTotal());  
 amount.setDetails(details);  
  
 Transaction transaction = new Transaction();  
 transaction.setAmount(amount);  
 transaction.setDescription(orderDetail.getProductName());  
  
 ItemList itemList = new ItemList();  
 List<Item> items = new ArrayList<>();  
  
 Item item = new Item();  
 item.setCurrency("USD");  
 item.setName(orderDetail.getProductName());  
 item.setPrice(orderDetail.getSubtotal());  
 item.setTax(orderDetail.getTax());  
 item.setQuantity("1");  
  
 items.add(item);  
 itemList.setItems(items);  
 transaction.setItemList(itemList);  
  
 List<Transaction> listTransaction = new ArrayList<>();  
 listTransaction.add(transaction);  
  
 return listTransaction;  
 }  
  
 private String getApprovalLink(Payment approvedPayment) {  
 List<Links> links = approvedPayment.getLinks();  
 String approvalLink = null;  
  
 for (Links link : links) {  
 if (link.getRel().equalsIgnoreCase("approval\_url")) {  
 approvalLink = link.getHref();  
 break;  
 }  
 }  
  
 return approvalLink;  
 }  
  
  
 public Payment executePayment(String paymentId, String payerId)  
 throws PayPalRESTException {  
 PaymentExecution paymentExecution = new PaymentExecution();  
 paymentExecution.setPayerId(payerId);  
  
 Payment payment = new Payment().setId(paymentId);  
  
 APIContext apiContext = new APIContext(*CLIENT\_ID*, *CLIENT\_SECRET*, *MODE*);  
  
 return payment.execute(apiContext, paymentExecution);  
 }  
  
  
 public Payment getPaymentDetails(String paymentId) throws PayPalRESTException {  
 APIContext apiContext = new APIContext(*CLIENT\_ID*, *CLIENT\_SECRET*, *MODE*);  
 return Payment.*get*(apiContext, paymentId);  
 }  
}

ReviewPaymentServlet

package com.flyaway;  
  
import java.io.IOException;  
 import javax.servlet.\*;  
 import javax.servlet.annotation.WebServlet;  
 import javax.servlet.http.\*;  
  
 import com.paypal.api.payments.\*;  
 import com.paypal.base.rest.PayPalRESTException;  
  
@WebServlet("/review\_payment")  
public class ReviewPaymentServlet extends HttpServlet {  
 private static final long *serialVersionUID* = 1L;  
  
 public ReviewPaymentServlet() {  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 String paymentId = request.getParameter("paymentId");  
 String payerId = request.getParameter("PayerID");  
  
 try {  
 PaymentServices paymentServices = new PaymentServices();  
 Payment payment = paymentServices.getPaymentDetails(paymentId);  
  
 PayerInfo payerInfo = payment.getPayer().getPayerInfo();  
 Transaction transaction = payment.getTransactions().get(0);  
 ShippingAddress shippingAddress = transaction.getItemList().getShippingAddress();  
  
 request.setAttribute("payer", payerInfo);  
 request.setAttribute("transaction", transaction);  
 request.setAttribute("shippingAddress", shippingAddress);  
  
 String url = "review.jsp?paymentId=" + paymentId + "&PayerID=" + payerId;  
  
 request.getRequestDispatcher(url).forward(request, response);  
  
 } catch (PayPalRESTException ex) {  
 request.setAttribute("errorMessage", ex.getMessage());  
 ex.printStackTrace();  
 request.getRequestDispatcher("error.jsp").forward(request, response);  
 }  
 }  
}

entities: POJO classes

booking

package entity;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import javax.persistence.Table;  
import javax.persistence.Embeddable;  
  
  
@Embeddable  
@Entity  
@Table(name = "booking")  
  
public class booking {  
  
 public booking() {  
  
 }  
  
 @Column(name = "bookingconfirmation")  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Id  
 int bookingconfirmation;  
  
 @Column(name = "fname")  
 String fname;  
  
  
 @Column(name = "lname")  
 String lname;  
  
  
 @Column(name = "totalpassengers")  
 String totalpassengers;  
  
 @Column(name = "price")  
 double price;  
  
 public int getBookingconfirmation() {  
 return bookingconfirmation;  
 }  
  
 public void setBookingconfirmation(int bookingconfirmation) {  
 this.bookingconfirmation = bookingconfirmation;  
 }  
  
 public String getFname() {  
 return fname;  
 }  
  
 public void setFname(String fname) {  
 this.fname = fname;  
 }  
  
 public String getLname() {  
 return lname;  
 }  
  
 public void setLname(String lname) {  
 this.lname = lname;  
 }  
  
 public String getTotalpassengers() {  
 return totalpassengers;  
 }  
  
 public void setTotalpassengers(String totalpassengers) {  
 this.totalpassengers = totalpassengers;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
  
 public void bookingconfirmation(String fname, String lname, String totalpassengers, double price) {  
 *//this.ProductID = ProductID;* this.fname = fname;  
 this.lname = lname;  
 this.totalpassengers = totalpassengers;  
 this.price = price;  
 }  
  
}

flightdetail

package entity;  
  
import com.flyaway.HibernateUtil;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
  
import javax.persistence.\*;  
import java.util.List;  
  
@Embeddable  
@Entity  
@Table(name="flightdetail")  
  
 public class flightdetail {  
 @Id  
 @Column(name="flight\_id")  
 private int id;  
  
 @Column(name="source")  
 private String source;  
  
 @Column(name="destination")  
 private String destination;  
  
 @Column(name="no\_of\_travllers")  
 private int travellers;  
  
 @Column(name="date")  
 private String date;  
  
 public flightdetail() {  
 super();  
 }  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int 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 int getTravellers() {  
 return travellers;  
 }  
  
 public void setTravellers(int travellers) {  
 this.travellers = travellers;  
 }  
  
 public String getDate() {  
 return date;  
 }  
  
 public void setDate(String date) {  
 this.date = date;  
 }  
  
  
 public flightdetail(String source, String destination, int travellers, String date) {  
 super();  
 this.source = source;  
 this.destination = destination;  
 this.travellers = travellers;  
 this.date = date;  
 }  
  
  
 }

userdetails

package entity;  
  
import javax.persistence.\*;  
  
@Embeddable  
@Entity  
@Table(name="userdetails")  
public class userdetails {  
  
 @Id  
 @GeneratedValue(strategy=GenerationType.*AUTO*)  
 @Column(name="userId")  
 private int userId;  
 @Column(name="name")  
 private String name;  
 @Column(name="password")  
 private String password;  
 @Column(name="email")  
 private String email;  
 public userdetails() {  
 super();  
 }  
  
 public int getUserId() {  
 return userId;  
 }  
  
 public void setUserId(int userId) {  
 this.userId = userId;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
  
 public userdetails(String name, String password, String email) {  
 super();  
 this.name = name;  
 this.password = password;  
 this.email = email;  
 }  
  
}

airlines

package entity;  
import java.util.List;  
  
import javax.persistence.CascadeType;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.JoinTable;  
import javax.persistence.OneToMany;  
import javax.persistence.Table;  
import javax.persistence.\*;  
import javax.persistence.Embeddable;  
import javax.persistence.\*;  
  
  
 @Embeddable  
 @Entity  
 @Table(name = "airlines")  
  
 public class airlines {  
 @Id  
 private int id;  
  
 @Column(name = "flights")  
 private String airline;  
 @Column(name = "price")  
 private double price;  
  
 public airlines() {  
 super();  
  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getAirline() {  
 return airline;  
 }  
  
 public void setAirline(String airline) {  
 this.airline = airline;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
  
  
 public airlines(int id, String airline, double price) {  
 super();  
 this.id = id;  
 this.airline = airline;  
 this.price = price;  
 }  
 }

*<?*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>  
 *<!-- Database connection settings -->* <property name="connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <property name="connection.url">jdbc:mysql://localhost:3306/flyawayflight</property>  
 <property name="connection.username">root</property>  
 <property name="connection.password"></property>  
  
 *<!-- JDBC connection pool settings -->* <property name="connection.pool\_size">2</property>  
  
 *<!-- Our SQL dialect -->* <property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>  
  
 *<!-- Print the sql generated on the console -->* <property name="show\_sql">true</property>  
  
 *<!-- Set the current session context -->* <property name="current\_session\_context\_class">thread</property>  
  
 *<!-- Update the database schema -->* <property name="hbm2ddl.auto">update</property>  
  
 <mapping class="entity.flightdetail"/>  
 <mapping class="entity.airlines"/>  
 <mapping class="entity.userdetails"/>  
 <mapping class="entity.booking"/>  
  
  
 </session-factory>  
</hibernate-configuration>

*# Root logger option*log4j.rootLogger=DEBUG, stdout  
  
*# Redirect log messages to console*log4j.appender.stdout=org.apache.log4j.ConsoleAppender  
log4j.appender.stdout.Target=System.out  
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout  
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n  
  
  
*# Display the low-level log messages on console*log4j.logger.org.hibernate=DEBUG

*/\*\*  
 \* Addproduct.js Javascript  
 \*/*function *addpet\_onclick*() {  
 *alert*("Inside function JS");  
 var x = document.getElementsByClassName("bookingconfirmation");  
 x[0].submit(); *//form submission*}

Adminsuccess.jsp:

<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Admin Home Page</title>  
<style type="text/css">  
button {  
 poition:relative;  
}  
input{  
font-size: 30px;  
}  
</style>  
</head>  
<body>  
<%@ include file="navigate.html" %>  
<h1 style="color: red; text-align: center;">Welcome Admin !!</h1>  
 <h2 style="text-align: center;"> <%= session.getAttribute("uname") %> </h2><br><br><br>  
 <input type="button" onclick="document.location.href='source\_destination\_list.jsp'" value="List Source & Destination" style="color: blue; width: 415px; height: 300px;">  
 <input type="button" onclick="document.location.href='airline\_list.jsp'" value="List Airlines" style="color: blue; width: 495px; height: 300px;">  
 <input type="button" onclick="document.location.href='flight-details.jsp'" value="List Flight Details" style="color: blue; width: 415px; height: 300px;">  
  
<img src="image/Map\_of\_USAState.png" width="40%"> </img>  
</body>  
</html>

Bookingconfirmation.jsp;

<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Registration Success Page</title>  
<style>  
a{  
font-size: 20px;  
color: Blue;  
}  
</style>  
</head>  
<body>  
<h1 style="color: forestgreen;">Booking Confirmed!</h1>  
<%  
  
 out.print("<td>" +request.getParameter("uname"));  
 out.print("<td>" +request.getParameter("fname"));  
 out.print("<td>" +request.getParameter("lname"));  
%>  
  
</body>  
</html>

Checkout.html

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>Check Out</title>  
 <style type="text/css">  
 table { border: 0; }  
 table td { padding: 10px; }  
 </style>  
</head>  
<body>  
<div align="center">  
 <h1>Check Out</h1>  
 <br/>  
 <form action="authorize\_payment" method="post">  
 <table>  
 <tr>  
 <td>Product/Service:</td>  
 <td><input type="text" name="product" value="Next iPhone" /></td>  
 </tr>  
 <tr>  
 <td>Sub Total:</td>  
 <td><input type="text" name="subtotal" value="100" /></td>  
 </tr>  
 <tr>  
 <td>Shipping:</td>  
 <td><input type="text" name="shipping" value="10" /></td>  
 </tr>  
 <tr>  
 <td>Tax:</td>  
 <td><input type="text" name="tax" value="10" /></td>  
 </tr>  
 <tr>  
 <td>Total Amount:</td>  
 <td><input type="text" name="total" value="120" /></td>  
 </tr>  
 <tr>  
 <td colspan="2" align="center">  
 <input type="submit" value="Checkout" />  
 </td>  
 </tr>  
 </table>  
 </form>  
</div>  
</body>  
</html>

Checkout.js

<form action='/charge' method='POST' id='checkout-form' xmlns:th="http://www.w3.org/1999/xhtml">  
 <input type='hidden' th:value='${amount/100}' name='amount'/>  
 <h1>Price:<span th:text='${amount/100}'/></h1>  
 <script  
 src='https://checkout.stripe.com/checkout.js'  
 className='stripe-button'  
 th:attr='data-key=${pk\_test\_51IedCRDKgFH4TxsgPn0TasZDwT27QBq43ci2Jm5Wk0qnvMt374eLAwXjGgdZUHG02oeV4B3WAz27ANeZI7bujtRv007mDXZARH},  
 data-amount=${1}'  
 data-name='StackAbuse Services'  
 data-description='Product Checkout'  
 data-image  
 ='http://www.stackabuse.com/assets/images/sa-java-dark.svg?v=b5a08453df'  
 data-locale='auto'  
 data-zip-code='false'>  
 </script>  
</form>

Error.jsp:

*<%--  
 Created by IntelliJ IDEA.  
 User: krmal  
 Date: 4/10/2021  
 Time: 3:28 AM  
 To change this template use File | Settings | File Templates.  
--%>*<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>Payment Error</title>  
</head>  
<body>  
<div align="center">  
 <h1>Payment Error</h1>  
 <br/>  
 <h3>${errorMessage}</h3>  
 <br/>  
</div>  
</body>  
</html>

Flight-details.jsp:

<%@ page import="java.util.List" %>  
<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
<%@ page import="entity.flightdetail" %>  
<%@ page import="org.hibernate.Session" %>  
<%@ page import="com.flyaway.HibernateUtil" %>  
<%@ page import="org.hibernate.Transaction" %>  
<%@ page import="java.io.PrintWriter" %>  
<%@page import="java.sql.DriverManager"%>  
<%@page import="java.sql.ResultSet"%>  
<%@page import="java.sql.Statement"%>  
<%@page import="java.sql.Connection"%>  
  
<!DOCTYPE html">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">  
  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>  
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js" integrity="sha384-SR1sx49pcuLnqZUnnPwx6FCym0wLsk5JZuNx2bPPENzswTNFaQU1RDvt3wT4gWFG" crossorigin="anonymous"></script>  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.min.js" integrity="sha384-j0CNLUeiqtyaRmlzUHCPZ+Gy5fQu0dQ6eZ/xAww941Ai1SxSY+0EQqNXNE6DZiVc" crossorigin="anonymous"></script>  
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>  
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>  
<!-- Datepicker -->  
<link href='bootstrap-datepicker/dist/css/bootstrap-datepicker.min.css' rel='stylesheet' type='text/css'>  
<script src='bootstrap-datepicker/dist/js/bootstrap-datepicker.min.js' type='text/javascript'></script>  
<meta name="viewport" content="width=device-width, initial-scale=2">  
<title>Flight search Details</title>  
<style>  
 table, td, th, h1 {  
 align-content: center;  
 border: 2px solid gray;  
 padding: 5px;  
 margin-left: auto;  
 margin-right: auto;  
 }  
  
 th {  
 background-color: #4287f5;  
 color: white;  
 }  
 <style>  
 .img-container {  
 text-align: center;  
 display: block;  
 }  
 </style>  
</style>  
</head>  
<body>  
<h1>Flight Search Results</h1>  
<form method="post" action="passengerpayment.jsp"">  
<table>  
 <tr>  
 <th>Source</th>  
 <th>Destination</th>  
 <th>Airline</th>  
 <th>Ticket Price</th>  
 </tr>  
  
  
 <%  
  
 String driver = "com.mysql.cj.jdbc.Driver";  
 String connectionUrl = "jdbc:mysql://localhost:3306/";  
 String database = "flyawayflight";  
 String userid = "root";  
 String password = "";  
 try {  
 Class.*forName*(driver);  
 } catch (ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 Connection connection = null;  
 Statement statement = null;  
 ResultSet resultSet = null;  
  
 %>  
  
 <%  
 try{  
 connection = DriverManager.*getConnection*(connectionUrl+database, userid, password);  
 statement=connection.createStatement();  
 String sql ="select *\** from flightdetail";  
 resultSet = statement.executeQuery(sql);  
 while(resultSet.next()){  
 %>  
 <tr>  
 <td><%=resultSet.getString("flight\_id") %></td>  
 <td><%=resultSet.getString("source") %></td>  
 <td><%=resultSet.getString("destination") %></td>  
 <td><%=resultSet.getString("price") %></td>  
 <td><%=resultSet.getString("date") %></td>  
 <td><button value="submit" href="passengerPayment.html">continue with booking</button></td>  
  
 <br>  
  
 </tr>  
 <%  
 }  
 connection.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 %>  
<br>  
<%  
 out.print("<td>" +"Flight Search Information searched from Home Page"+"</td>");  
  
%>  
<%  
  
 out.print("<td>" +request.getParameter("source"));  
 out.print("<td>" +request.getParameter("destination"));  
 out.print("<td>" +request.getParameter("travellers"));  
 out.print("<td>" +request.getParameter("date"));  
  
%>  
  
  
<c:forEach var="flight1" items="${listFlight}">  
 <tr>  
 <td><c:out value="${flight1.source}"/></td>  
 <td><c:out value="${flight1.destination}"/></td>  
 <td><c:out value="${flight1.travellers}"/></td>  
 <td><c:out value="${flight1.date}"/></td>  
  
 </tr>  
  
  
 <a href="edit?id=<c:out value='${user.id}' />">Edit</a>  
</c:forEach>  
  
</table>  
<Div>  
  
 <img src="Map\_of\_USA.png" height="303" width="780"/>  
 <img src="How-to-find.jpg" height="303" width="780"/>  
  
</Div>  
  
</body>  
</html>

Index.html:

<!DOCTYPE html>  
<html>  
<head>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>  
 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js" integrity="sha384-SR1sx49pcuLnqZUnnPwx6FCym0wLsk5JZuNx2bPPENzswTNFaQU1RDvt3wT4gWFG" crossorigin="anonymous"></script>  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.min.js" integrity="sha384-j0CNLUeiqtyaRmlzUHCPZ+Gy5fQu0dQ6eZ/xAww941Ai1SxSY+0EQqNXNE6DZiVc" crossorigin="anonymous"></script>  
 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>  
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>  
 *<!-- Datepicker -->* <link href='bootstrap-datepicker/dist/css/bootstrap-datepicker.min.css' rel='stylesheet' type='text/css'>  
 <script src='bootstrap-datepicker/dist/js/bootstrap-datepicker.min.js' type='text/javascript'></script>  
 <meta name="viewport" content="width=device-width, initial-scale=2">  
<title>FlyAway Flights Booking Engine</title>  
<style>  
h1 {  
 text-align: center;  
 border-style: solid;  
 color: #069a43;  
}  
</style>  
</head>  
<body style="background: white;">  
 <fieldset>  
   
 <legend style="text-align: center;">FlyAway Flights Booking Engine</legend>  
  
 <img src="FlyAway.jpg" height="115" width="168"/><Br>  
  
 <input type="button" style="position:relative; color: blue; width: 300px; font-size: 20px;" onclick="document.location.href='login'" value="Admin Login">  
 <br>  
  
 <fieldset>  
 <legend>Search Flights</legend>  
 <form method="post" action="flight-details.jsp">  
 <label>Source</label> <select id="source" name="source">  
 <option></option>  
 <option value="Seattle">Seattle</option>  
 <option value="Orlando">Orlando</option>  
 <option value="Hawaii">Hawaii</option>  
 <option value="Alaska">Alaska</option>  
 <option value="Chicago">Chicago</option>  
 <option value="California">California</option>  
 <option value="Texas">Texas</option>  
 <option value="Wisconsin">Wisconsin</option>  
 </select> <label>Destination</label> <select id="destination"  
 name="destination">  
 <option></option>  
 <option value="Seattle">Seattle</option>  
 <option value="Orlando">Orlando</option>  
 <option value="Hawaii">Hawaii</option>  
 <option value="Alaska">Alaska</option>  
 <option value="Chicago">Chicago</option>  
 <option value="California">California</option>  
 <option value="Texas">Texas</option>  
 <option value="Wisconsin">Wisconsin</option>  
 </select> <label>Travellers</label>   
 <input type="number"  
 onkeypress="return event.charCode >= 48" min="1" name="travellers"> <label>Flight  
 Date</label> <input type="date" id="date" name="date">  
 <button value="submit" >Submit</button>  
 </form>  
 </fieldset>  
 <br>  
 </fieldset>  
  
 <br>  
  
 <br>  
 <fieldset style="vertical-align: bottom; background-color: floralwhite;">  
  
 </fieldset>  
 <br>  
  
<Div>  
  
 <img src="flight1.jpg" height="303" width="480"/>  
 <img src="Map\_of\_USA.png" height="303" width="480"/>  
</Div>  
</body>  
</html>

Login.html:

<!DOCTYPE html>  
<html>  
<head>  
<meta charset="ISO-8859-1">  
<title>Login Page</title>  
<style type="text/css">  
label, input {  
 display: inline-block;  
 vertical-align: baseline;  
 width: 125px;  
}  
  
label {  
 color: #2D2D2D;  
 font-size: 25px;  
}  
  
form {  
 width: 300px;  
}  
a{  
font-size: 20px;  
color: Blue;  
}  
</style>  
  
</head>  
<body style="background-color: whitesmoke">  
  
<a href="index.html">Home</a> ||  
<a href="available-flight.jsp">Back</a> <br><br>  
  
<fieldset>  
<legend style="font-size: 50px">Login Here</legend>  
<div id="main">  
<form method="post" action="login">  
 <label for="name">Username:</label><input type="text" id="name" name="name"><br/><br>  
 <label for="password">Password:</label><input type="password" id="password" name="password"><br/><br>  
<button type="submit" style="font-size: 20px;">login</button><br><br>  
<br><br><br>  
<input type="button" onclick="document.location.href='Registration'" value="New Member Register" style="color: blue; width: 200px; ">  
  
</form>  
</div>  
</fieldset>  
<div>  
<img src="flight1.jpg" height="303" width="480"/>  
</div>  
</body>  
</html>

Paymentconfirmation

<!DOCTYPE html>  
<html>  
<head>  
<meta charset="ISO-8859-1">  
<title>Insert title here</title>  
</head>  
<body style="text-align: center">  
<br>  
<button onclick="*paymentmsg*()">Confirm Payment</button>  
<h1 id="msg" style="color:#4287f5"></h1>  
<script>  
function *paymentmsg*() {  
 document.getElementById("msg").innerHTML = "Booking Confirmation is Successful!";  
}  
</script>  
</body>  
</html>

Receipts.jsp:

*<%--  
 Created by IntelliJ IDEA.  
 User: krmal  
 Date: 4/10/2021  
 Time: 4:00 AM  
 To change this template use File | Settings | File Templates.  
--%>*<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>Payment Receipt</title>  
 <style type="text/css">  
 table { border: 0; }  
 table td { padding: 5px; }  
 </style>  
</head>  
<body>  
<div align="center">  
 <h1>Payment Done. Thank you for purchasing our products</h1>  
 <br/>  
 <h2>Receipt Details:</h2>  
 <table>  
 <tr>  
 <td><b>Merchant:</b></td>  
 <td>Company ABC Ltd.</td>  
 </tr>  
 <tr>  
 <td><b>Payer:</b></td>  
 <td>${payer.firstName} ${payer.lastName}</td>  
 </tr>  
 <tr>  
 <td><b>Description:</b></td>  
 <td>${transaction.description}</td>  
 </tr>  
 <tr>  
 <td><b>Subtotal:</b></td>  
 <td>${transaction.amount.details.subtotal} USD</td>  
 </tr>  
 <tr>  
 <td><b>Shipping:</b></td>  
 <td>${transaction.amount.details.shipping} USD</td>  
 </tr>  
 <tr>  
 <td><b>Tax:</b></td>  
 <td>${transaction.amount.details.tax} USD</td>  
 </tr>  
 <tr>  
 <td><b>Total:</b></td>  
 <td>${transaction.amount.total} USD</td>  
 </tr>  
 </table>  
</div>  
</body>  
</html>