<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);  
 }  
 }  
}