|  |
| --- |
| Flyaway  **(Sprint work and Project Specifications)** |

**Version History:**

|  |  |
| --- | --- |
| Author | KUMAR ABHISHEK |
| Purpose | Sprint work and Specification of project |
| Date | 06 October 2021 |
| Version | 1.0 |

**Contents**

[1. MODULE OF THE PROJECT: 3](#_Toc84252034)

[2. SPRINT WISE WORK: 3](#_Toc84252035)

[3. GIT HUB LINK: 4](#_Toc84252036)

[4. PROJECT CODE: 5](#_Toc84252037)

[5. MAPPING: 28](#_Toc84252038)

[6. SPRINT: 29](#_Toc84252039)

[7. DATABASE TABLE DESIGN AND RELATIONSHIP: 30](#_Toc84252040)

[8. CORE CONCEPTS USED IN THE PROJECT: 30](#_Toc84252041)

[9. CLASSES: 31](#_Toc84252042)

# **Module 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.

# **Sprint wise work:**

|  |  |
| --- | --- |
| **Sprint Number** | **Modules** |
| 1 | Index file:It have an Admin login, Search for Flight , See All Flights |
| 2 | Admin login: Email, password, submit |
| 3 | Dashboard:  Logout, change password, |
| 4 | Change password |
| 5 | Payment options: To connect to MySQL to give a payment |
| 6 | See all flights by connecting to MySQL and throw hibernate |

# **Git hub link:**

|  |  |
| --- | --- |
| **REPOSITORY NAME** | [kumarabhishek7885](https://github.com/kumarabhishek7885)/[flyaway1](https://github.com/kumarabhishek7885/flyaway1) |
| **REPOSITORY LINK** | https://github.com/kumarabhishek7885/flyaway1.git |

# Project Code:

|  |
| --- |
| **Folder Structure** |
|  |
|  |

|  |
| --- |
| **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> |

|  |
| --- |
| **Web.xml** |
| <!DOCTYPE web-app PUBLIC  "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  "http://java.sun.com/dtd/web-app\_2\_3.dtd" >  <element>  <web-app>  <display-name>Archetype Created Web Application</display-name>  <servlet>  <servlet-name>login</servlet-name>  <display-name>login</display-name>  <description></description>  <servlet-class>com.simplilearn.servlets.admin.login</servlet-class>  </servlet>  <servlet>  <servlet-name>changedpassword</servlet-name>  <display-name>changedpassword</display-name>  <description></description>  <servlet-class>com.simplilearn.servlets.admin.changedpassword</servletclass> </servlet>  <servlet-mapping>  <servlet-name>login</servlet-name>  <url-pattern>/login</url-pattern>  </servlet-mapping>  <servlet-mapping>  <servlet-name>changedpassword</servlet-name>  <url-pattern>/changedpassword</url-pattern>  </servlet-mapping>  </web-app>  </element> |

|  |
| --- |
| **Hibernate.annotation.cfg.xml** |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE hibernate-configuration PUBLIC  "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  <hibernate-configuration>  <session-factory> <property  name=*"hibernate.connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>  <property  name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/flyawaydb</property>  <property name=*"hibernate.connection.username"*>root</property>  <property name=*"hibernate.connection.password"*>Abhi18@ayaan19</property>  <property name=*"hibernate.connection.pool\_size"*>1</property>  <property name=*"hibernate.current\_session\_context\_class"*>thread</property>  <property name=*"hibernate.show\_sql"*>true</property>  <property  name=*"hibernate.dialect"*>org.hibernate.dialect.MySQL5Dialect</property> <property name=*"hibernate.id.new\_generator\_mappings"*>false</property>  <mapping class=*"com.simplilearn.tables.Admin"* />  <mapping class=*"com.simplilearn.tables.Airline"* /> <mapping class=*"com.simplilearn.tables.Flight"* />  <mapping class=*"com.simplilearn.tables.Payment"* />  <mapping class=*"com.simplilearn.tables.Person"* />  </session-factory>  </hibernate-configuration> |

|  |
| --- |
| **Index.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>Welcome To Fly Away Ticket Booking Portal</title> </head>    <body bgcolor=*"cyan"* >  <center>  <a href=*"login.html"*>  <input type=*"submit"* value=*"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>  </center>  </body>  </html> |

|  |
| --- |
| **login.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>Admin Login Page</title>  </head>  <body align=*"center"*>  <a href=*"index.html"*>  <input type=*"submit"* value=*"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> |

|  |
| --- |
| **Login.java** |
| 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 login   \*/    @WebServlet("/login")  public class login extends HttpServlet {  /\*\*   * \*/   private static final long serialVersionUID = 1L; public static boolean isLoggedIn = false; public static String password = "Abhi18@ayaan19"; public static String email = "kumar.abhishek7885@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");    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();  }  } |

|  |
| --- |
| **Databash.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"ISO-8859-1"*>  <title>Admin Dashboard</title>  </head>  <body bgcolor=*"skyblue"* align=*"center"*>  <a href=*"index.html"*><input type=*"submit"* value=*"Home"*></a> |  <a href=*"index.html"*><input type=*"submit"* value=*"Logout"*></a> | <a href=*"changepassword.html"*><input type=*"submit"* value=*"Change password"*></a> |<br>  <img src=*"https://images.financialexpress.com/2020/08/Air-IndiaReuters.jpg"*>    <hr>  </body>  </html> |

|  |
| --- |
| **Changepassword.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>Admin Login Page</title>  </head>  <body align=*"center"*>  <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> |

|  |
| --- |
| **Changepassword.java** |
| 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");    }  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();  }    } |

|  |
| --- |
| **Verifiedpassword.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"ISO-8859-1"*>  <title>password changed</title>  </head>  <body align=*"center"*>  <h1>your password has been changed successfully </h1>    <form action=*"dashboard.html"*>  <input type=*"submit"* value=*"Dashboard"*>  </form>  </body>  </html> |
| **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 {  }  } |

|  |
| --- |
| **registration.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>Flight Registration Page</title>  </head>  <body align=*"center"*>  <h1>Flight Registration Page</h1>  <form name=*"flightRegistraton"* method=*"POST"* action=*"registeration"*>  Flight ID : <input type=*"number"* name=*"flightid"* placeholder=*"flight id"* required> <br/>  First Name : <input type=*"text"* name=*"firstname"* placeholder=*"First name"* required> <br/>  Last Name : <input type=*"text"* name=*"lastname"* placeholder=*"Last name"* required> <br/>  Email : <input type=*"email"* name=*"email"* placeholder=*"enter email id"* required> <br/>  Birthday (yyyy-MM-dd) : <input type=*"date"* name=*"birthday"* placeholder=*"Dob"* required> <br/>  <br/>  <input type=*"submit"* value=*"Submit Flight Registration"*>  </form>    </body>  </html> |

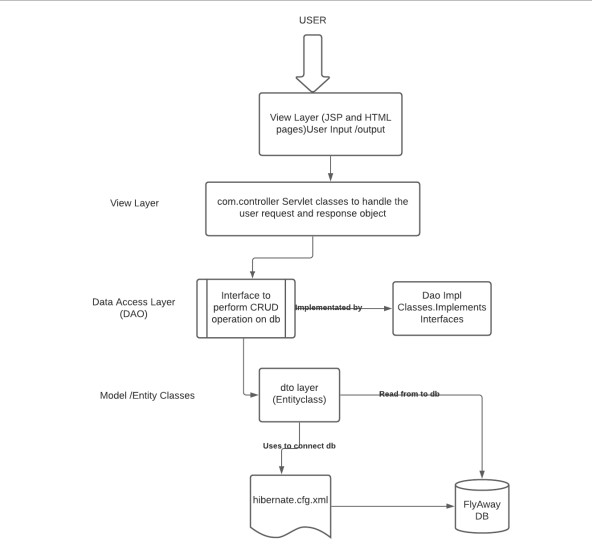
|  |
| --- |
| **Registration.java** |
| package com.simplilearn.servlets.users;  import java.io.IOException;  import java.io.PrintWriter;  import java.sql.Date;  import java.util.List;  import java.util.regex.Pattern;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  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();  }  }  }  } |

|  |
| --- |
| **Payment.html** |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>Payment</title> </head>  <body align =*"center"*>  <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> |

|  |
| --- |
| **PayOnline.java** |
| 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    }  } |

|  |
| --- |
| **ConfirmationPage.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 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 {    }  } |

# **Mapping:**



# **Sprint:**

Planned and the tasks achieved in them. Flyaway was divided in 3 sprints of 1 week each.

Sprint 1: -

Initialized the project, made it Hibernate project and added dependencies: ‐ MySQL connector, Hibernate libraries, JSTL, Servlet API.

* Setting MySQL database environment and tomcat server
* Made E.R. diagram for database and entities relationship.
* Made classes for entities and established relationship between them.
* Configured hibernate and mapped classes as entities.
* Populated the data by hibernate

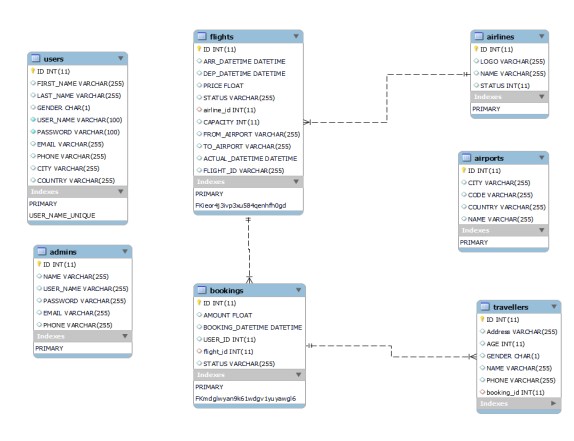
#### Sprint 2: ‐

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

#### Sprint 3: ‐

* Made the master list JSP 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 Send Mail API to send mail on successful registration/ booking.
* Modified the navigation with the help of buttons 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.

# **Database table design and relationship:**

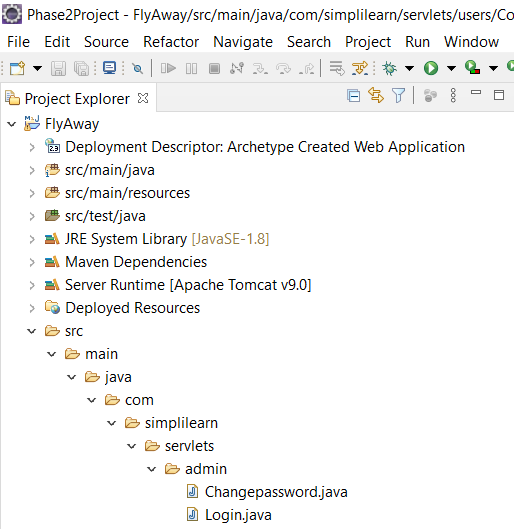


# **Core concepts used in the project:**

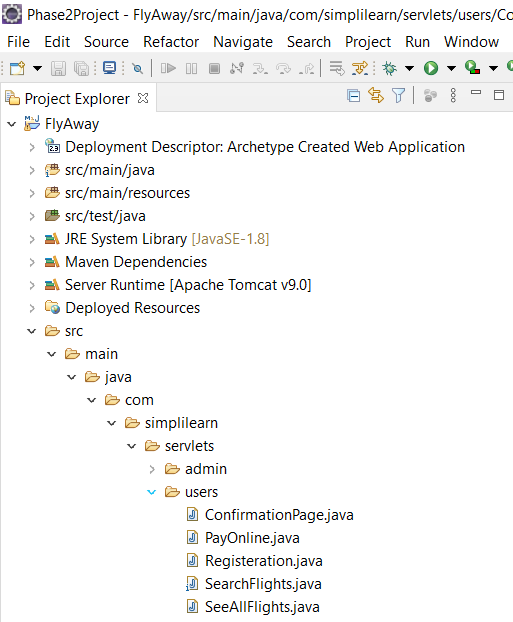
* Server request response object
* HTTP Session, Cookies URL Redirect Hidden Field.
* Collection Framework
* Date, Time object with simpleDateFormat
* Hibernate JPA, Different types of mapping (one to one, one to many, many to one, many to many).
* Sorting technique using database query as well as using 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 Send Mail API to send email.
* Java server page 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.

# **Classes:**

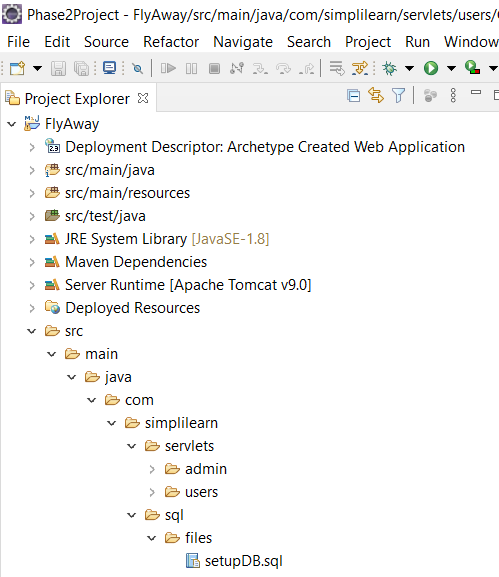
**Com.simplilearn.servlets.admin:**

****

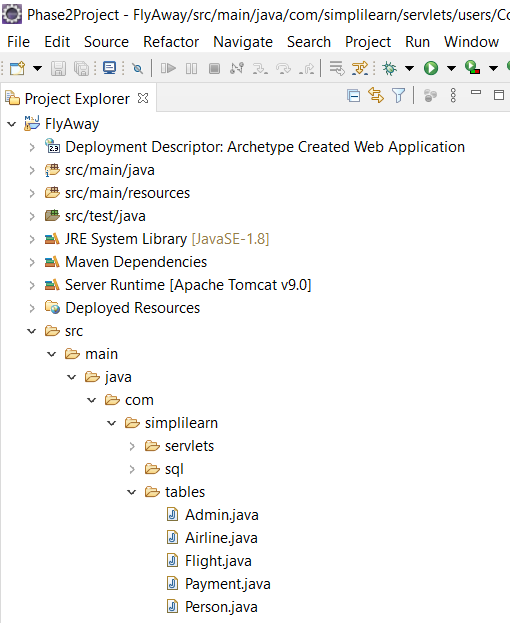
**Com.simplilearn.servlets.user:**

****

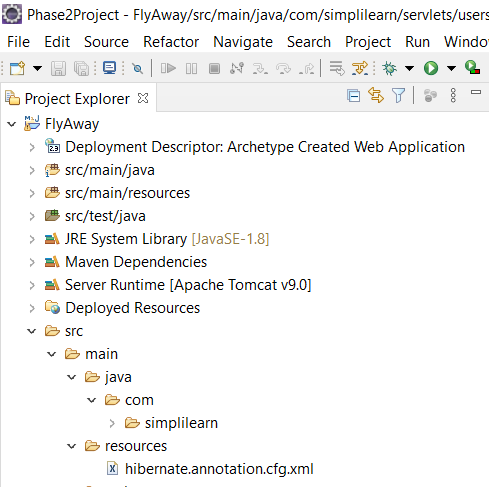
**Com.simplilearn.sql.file:**

****

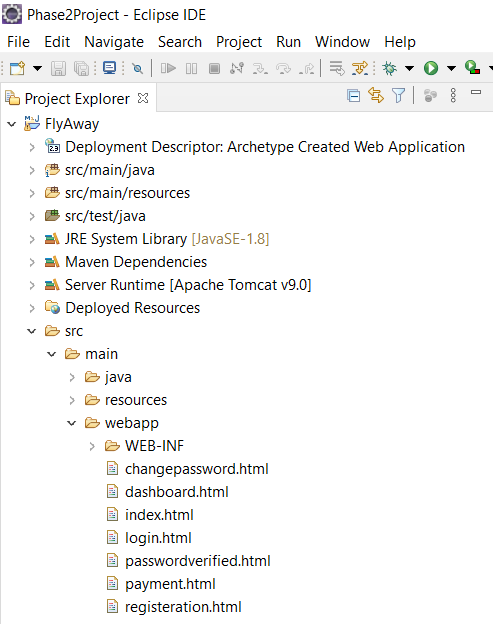
**Com.simplilearn.tables:**

****

**Com.simplilearn.util:**

****

**Webapp:**

****

**ThankYou,**

**KUMAR ABHISHEK,**

**Simplilearn.**