|  |
| --- |
| **FlyAway**  **(Project Source Code)** |

**Version History:**

|  |  |
| --- | --- |
| Author | Nikhil Jain |
| Purpose | Source Code of the application |
| Date | 27th September 2021 |
| Version | 1.0 |

Table of Contents

[ **Project Link** 3](#_Toc83730387)

[ **Folder Structure** 3](#_Toc83730388)

[ **Controllers** 4](#_Toc83730389)

[1. AddFlight.java 4](#_Toc83730390)

[2. CardDetails.java 5](#_Toc83730391)

[3. ChangePassword.java 6](#_Toc83730392)

[4. ValidateUser.java 7](#_Toc83730393)

[5. RegisterUser.java 8](#_Toc83730394)

[6. SerachResults.java 9](#_Toc83730395)

[7. TravellerDetails.java 10](#_Toc83730396)

[8. LogoutServlet.java 11](#_Toc83730397)

[ **Data Access Object** 12](#_Toc83730398)

[1. AdminDao.java 12](#_Toc83730399)

[2. AdminDaoImpl.java 13](#_Toc83730400)

[3. FlightDetailsDao.java 15](#_Toc83730401)

[4. FlightDetailsDaoImpl.java 16](#_Toc83730402)

[ **Data Transfer Object** 18](#_Toc83730403)

[1. FlightDetails.java 18](#_Toc83730404)

[2. Admin.java 20](#_Toc83730405)

[ **Utility** 21](#_Toc83730406)

[1. HibernateUtil.java 21](#_Toc83730407)

# **Project Link**

|  |  |
| --- | --- |
| Repository Name | FlyAway |
| GitHub Link | <https://github.com/Niks4u2/FlyAway> |
| Hosted On | <http://flyaway-env-1.eba-4f85ddgn.ap-south-1.elasticbeanstalk.com/> |

# **Folder Structure**

|  |
| --- |
|  |

# **Controllers**

# AddFlight.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** com.dao.FlightDetailsDao;  **import** com.dao.FlightDetailsDaoImpl;  **import** com.dto.FlightDetails;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **public** **class** AddFlight **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** AddFlight() {};  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  response.setContentType("text/html");  PrintWriter out = response.getWriter();    String flightNumber = request.getParameter("flightNumber");  String airline = request.getParameter("airline");  String source = request.getParameter("source");  String destination = request.getParameter("destination");  String time = request.getParameter("time");  String price = request.getParameter("price");    FlightDetails flight = **new** FlightDetails(flightNumber, airline, source, destination, time, price);  FlightDetailsDao dao = **new** FlightDetailsDaoImpl();  Integer flightId = dao.addFlight(flight);    **if**(flightId != **null**)  {  request.getRequestDispatcher("dashboard.jsp").include(request, response);  out.print("<script> window.alert('Flight Details added successfully!')</script>");  }  }  } |

# CardDetails.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **public** **class** CardDetails **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** CardDetails() {  // **TODO** Auto-generated constructor stub  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String cardname = request.getParameter("cardname");  String cardnum = request.getParameter("cardnum");    HttpSession session = request.getSession();  session.setAttribute("cardname", cardname);  session.setAttribute("cardnum", cardnum);    response.sendRedirect("confirmation.jsp");  }  } |

# ChangePassword.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** com.dao.AdminDao;  **import** com.dao.AdminDaoImpl;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **public** **class** ChangePassword **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** ChangePassword() {  // **TODO** Auto-generated constructor stub  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  response.setContentType("text/html");  PrintWriter out = response.getWriter();  String email = request.getParameter("email");  String password = request.getParameter("password");    AdminDao dao = **new** AdminDaoImpl();  dao.updateAdminPassword(email, password);    request.getRequestDispatcher("login.jsp").include(request, response);  out.print("<header style='color:green; margin-left:7%'> Password updated successfully! </header>");  }  } |

# ValidateUser.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** com.dao.AdminDao;  **import** com.dao.AdminDaoImpl;  **import** com.dto.Admin;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **public** **class** ValidateUser **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** ValidateUser() {  // **TODO** Auto-generated constructor stub  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  response.setContentType("text/html");  PrintWriter out = response.getWriter();  String email = request.getParameter("email");  String password = request.getParameter("password");    Admin admin = **new** Admin(email, password);  AdminDao dao = **new** AdminDaoImpl();    **boolean** isAdminValid = dao.validateAdmin(admin);    **if**(isAdminValid)  {  HttpSession session = request.getSession();  session.setAttribute("email", admin.getEmail());  session.setAttribute("password", admin.getPassword());  response.sendRedirect("dashboard.jsp");  } **else** {  request.getRequestDispatcher("login.jsp").include(request, response);  out.print("<header style='color:red; margin-left:7%'> Invalid email or password! </header>");  }  }  } |

# RegisterUser.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** java.io.PrintWriter;  **import** com.dao.AdminDao;  **import** com.dao.AdminDaoImpl;  **import** com.dto.Admin;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **public** **class** RegisterUser **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** RegisterUser() {  // **TODO** Auto-generated constructor stub  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  response.setContentType("text/html");  PrintWriter out = response.getWriter();  String email = request.getParameter("email");  String password = request.getParameter("password");    Admin admin = **new** Admin(email, password);  AdminDao dao = **new** AdminDaoImpl();    Integer adminId = dao.addAdmin(admin);    **if**(adminId != **null**)  response.sendRedirect("login.jsp");  }  } |

# SerachResults.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** java.util.List;  **import** com.dao.FlightDetailsDao;  **import** com.dao.FlightDetailsDaoImpl;  **import** com.dto.FlightDetails;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **public** **class** SearchResults **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** SearchResults() {  // **TODO** Auto-generated constructor stub  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String source = request.getParameter("source");  String destination = request.getParameter("destination");  String date = request.getParameter("date");  String travellers = request.getParameter("travellers");    HttpSession session = request.getSession();  session.setAttribute("source", source);  session.setAttribute("destination", destination);  session.setAttribute("date", date);  session.setAttribute("travellers", travellers);    FlightDetailsDao dao = **new** FlightDetailsDaoImpl();  List<FlightDetails> flights = dao.searchFlights(source, destination);  session.setAttribute("flights", flights);  response.sendRedirect("search-results.jsp");  }  } |

# TravellerDetails.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **public** **class** TravellerDetails **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** TravellerDetails() {  // **TODO** Auto-generated constructor stub  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String pname = request.getParameter("pname");  String pemail = request.getParameter("pemail");  String phone = request.getParameter("phone");    HttpSession session = request.getSession();  session.setAttribute("pname", pname);  session.setAttribute("pemail", pemail);  session.setAttribute("phone", phone);    response.sendRedirect("checkout.jsp");  }  } |

# LogoutServlet.java

|  |
| --- |
| **package** com.controllers;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** javax.servlet.http.HttpSession;  **public** **class** LogoutServlet **extends** HttpServlet {  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  **public** LogoutServlet() {  // **TODO** Auto-generated constructor stub  }  **protected** **void** service(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {    HttpSession session = request.getSession();  session.removeAttribute("email");  session.invalidate();  response.sendRedirect("login.jsp");  }  } |

# **Data Access Object**

# AdminDao.java

|  |
| --- |
| **package** com.dao;  **import** com.dto.Admin;  **public** **interface** AdminDao {  **public** **boolean** validateAdmin(Admin admin);  **public** Integer addAdmin(Admin admin);  **public** **void** updateAdminPassword(String email, String password);  **public** **void** deleteAdmin(Integer adminId);  **public** Admin searchAdminByEmail(String email);  } |

# AdminDaoImpl.java

|  |
| --- |
| **package** com.dao;  **import** javax.persistence.TypedQuery;  **import** org.hibernate.Session;  **import** org.hibernate.SessionFactory;  **import** org.hibernate.Transaction;  **import** org.hibernate.boot.Metadata;  **import** org.hibernate.boot.MetadataSources;  **import** org.hibernate.boot.registry.StandardServiceRegistry;  **import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;  **import** com.dto.Admin;  **public** **class** AdminDaoImpl **implements** AdminDao{    **private** SessionFactory factory;    **public** AdminDaoImpl()  {  StandardServiceRegistry ssr = **new** StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  Metadata meta = **new** MetadataSources(ssr).getMetadataBuilder().build();  factory = meta.getSessionFactoryBuilder().build();  }    **public** Integer addAdmin(Admin admin) {  Integer adminId = **null**;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  adminId = (Integer) session.save(admin);  txn.commit();  session.close();  **return** adminId;  }  **public** **void** updateAdminPassword(String email, String password) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();    Admin admin = searchAdminByEmail(email);  admin.setPassword(password);    session.update(admin);  txn.commit();  session.close();  }  **public** **void** deleteAdmin(Integer adminId) {  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  Admin employee = session.get(Admin.**class**, adminId);  session.delete(employee);  txn.commit();  session.close();  }    **public** Admin searchAdminByEmail(String email) {  Session session = factory.openSession();    String hql = "FROM Admin where email = :mail";  TypedQuery<Admin> query = session.createQuery(hql);  query.setParameter("mail", email);  Admin admin = query.getSingleResult();  session.close();  **return** admin;  }  **public** **boolean** validateAdmin(Admin admin) {  **boolean** isAdminValid = **false**;  String email = admin.getEmail();  String password = admin.getPassword();  Admin admindb = searchAdminByEmail(email);    **if**(email.equals(admindb.getEmail()) && password.equals(admindb.getPassword()))  isAdminValid = **true**;    **return** isAdminValid;  }  } |

# FlightDetailsDao.java

|  |
| --- |
| **package** com.dao;  **import** java.util.List;  **import** com.dto.FlightDetails;  **public** **interface** FlightDetailsDao {  **public** Integer addFlight(FlightDetails flight);  **public** List<FlightDetails> searchFlights(String source, String destination);  } |

# FlightDetailsDaoImpl.java

|  |
| --- |
| **package** com.dao;  **import** java.util.List;  **import** javax.persistence.TypedQuery;  **import** org.hibernate.Session;  **import** org.hibernate.SessionFactory;  **import** org.hibernate.Transaction;  **import** org.hibernate.boot.Metadata;  **import** org.hibernate.boot.MetadataSources;  **import** org.hibernate.boot.registry.StandardServiceRegistry;  **import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;  **import** com.dto.FlightDetails;  **public** **class** FlightDetailsDaoImpl **implements** FlightDetailsDao {    **private** SessionFactory factory;  FlightDetails flight = **null**;    **public** FlightDetailsDaoImpl()  {  StandardServiceRegistry ssr = **new** StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  Metadata meta = **new** MetadataSources(ssr).getMetadataBuilder().build();  factory = meta.getSessionFactoryBuilder().build();  }  **public** Integer addFlight(FlightDetails flight) {  Integer flightId = **null**;  Session session = factory.openSession();  Transaction txn = session.beginTransaction();  flightId = (Integer) session.save(flight);  txn.commit();  session.close();  **return** flightId;  }    **public** List<FlightDetails> searchFlights(String source, String destination) {  List<FlightDetails> flights = **null**;  Session session = factory.openSession();    String hql = "FROM FlightDetails where source = :src AND destination = : dest";  TypedQuery<FlightDetails> query = session.createQuery(hql);  query.setParameter("src", source);  query.setParameter("dest", destination);  flights = query.getResultList();  session.close();  **return** flights;  }    **public** List<FlightDetails> getAllFlights() {  List<FlightDetails> flights = **null**;  Session session = factory.openSession();    String hql = "FROM FlightDetails";  TypedQuery<FlightDetails> query = session.createQuery(hql);  flights = query.getResultList();  session.close();  **return** flights;  }  } |

# **Data Transfer Object**

# FlightDetails.java

|  |
| --- |
| **package** com.dto;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.Table;  @Entity  @Table  **public** **class** FlightDetails {    @Id  @GeneratedValue(strategy = GenerationType.***IDENTITY***)  **private** **int** flightId;    **private** String flightNumber;    **private** String airline;    **private** String source;    **private** String destination;    **private** String time;    **private** String price;    **public** FlightDetails() {};    **public** FlightDetails(String flightNumber, String airline, String source, String destination, String time,  String price) {  **super**();  **this**.flightNumber = flightNumber;  **this**.airline = airline;  **this**.source = source;  **this**.destination = destination;  **this**.time = time;  **this**.price = price;  }    **public** **int** getFlightId() {  **return** flightId;  }  **public** **void** setFlightId(**int** flightId) {  **this**.flightId = flightId;  }  **public** String getFlightNumber() {  **return** flightNumber;  }  **public** **void** setFlightNumber(String flightNumber) {  **this**.flightNumber = flightNumber;  }  **public** String getAirline() {  **return** airline;  }  **public** **void** setAirline(String airline) {  **this**.airline = airline;  }  **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** String getTime() {  **return** time;  }  **public** **void** setTime(String time) {  **this**.time = time;  }  **public** String getPrice() {  **return** price;  }  **public** **void** setPrice(String price) {  **this**.price = price;  }  } |

# Admin.java

|  |
| --- |
| **package** com.dto;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.Table;  @Entity  @Table  **public** **class** Admin  {  @Id  @GeneratedValue(strategy = GenerationType.***IDENTITY***)  **private** **int** adminId;    **private** String email;  **private** String password;    **public** Admin() {};    **public** Admin(String email, String password) {  **super**();  **this**.email = email;  **this**.password = password;  }    **public** **int** getAdminId() {  **return** adminId;  }  **public** **void** setAdminId(**int** adminId) {  **this**.adminId = adminId;  }    **public** String getEmail() {  **return** email;  }  **public** **void** setEmail(String email) {  **this**.email = email;  }  **public** String getPassword() {  **return** password;  }    **public** **void** setPassword(String password) {  **this**.password = password;  }  @Override  **public** String toString() {  **return** "Admin [email=" + email + ", password=" + password + "]";  }  } |

# **Utility**

# HibernateUtil.java

|  |
| --- |
| **package** com.util;  **import** org.hibernate.SessionFactory;  **import** org.hibernate.boot.Metadata;  **import** org.hibernate.boot.MetadataSources;  **import** org.hibernate.boot.registry.StandardServiceRegistry;  **import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;  **public** **class** HibernateUtil {  **private** **static** StandardServiceRegistry *registry*;  **private** **static** SessionFactory *sessionFactory*;  **public** **static** SessionFactory getSessionFactory() {  **if** (*sessionFactory* == **null**) {  **try** {  // Create registry  *registry* = **new** StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  // Create MetadataSources  MetadataSources sources = **new** MetadataSources(*registry*);  // Create Metadata  Metadata metadata = sources.getMetadataBuilder().build();  // Create SessionFactory  *sessionFactory* = metadata.getSessionFactoryBuilder().build();  } **catch** (Exception e) {  e.printStackTrace();  **if** (*registry* != **null**) {  StandardServiceRegistryBuilder.*destroy*(*registry*);  }  }  }  **return** *sessionFactory*;  }  **public** **static** **void** shutdown() {  **if** (*registry* != **null**) {  StandardServiceRegistryBuilder.*destroy*(*registry*);  }  }  } |