

HibConfig

```
package org.SimplilearnProject.aabp.config;

import java.util.Properties;

import org.SimplilearnProject.aabp.entities.Admin;
import org.SimplilearnProject.aabp.entities.Airline;
import org.SimplilearnProject.aabp.entities.Booking;
import org.SimplilearnProject.aabp.entities.Flight;
import org.SimplilearnProject.aabp.entities.Payment;
import org.SimplilearnProject.aabp.entities.Place;
import org.SimplilearnProject.aabp.entities.User;
import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;
import org.hibernate.cfg.Environment;

public class HibConfig {

    public static SessionFactory getSessionFactory() {

        Configuration configuration = new Configuration();
        Properties properties = new Properties();
        // Data source properties
        properties.put(Environment.DRIVER, "com.mysql.cj.jdbc.Driver");
        properties.put(Environment.URL, "jdbc:mysql://localhost:3306/flyawaydb");
        properties.put(Environment.USER, "root");
        properties.put(Environment.PASS, "123456789");
```

```

        // Hibernate properties
        properties.put(Environment.SHOW_SQL, "true");
        properties.put(Environment.HBM2DDL_AUTO, "update");
        properties.put(Environment.FORMAT_SQL, true);
        configuration.setProperties(properties);
        configuration.addAnnotatedClass(Admin.class);
        configuration.addAnnotatedClass(Airline.class);
        configuration.addAnnotatedClass(Booking.class);
        configuration.addAnnotatedClass(Flight.class);
        configuration.addAnnotatedClass(Payment.class);
        configuration.addAnnotatedClass(Place.class);
        configuration.addAnnotatedClass(User.class);
        SessionFactory factory = configuration.buildSessionFactory();
        return factory;
    }
}

```

AdminLoginController

```

package org.SimplilearnProject.aabp.controller;

import java.io.IOException;
import java.io.PrintWriter;

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 javax.servlet.http.HttpSession;

import org.SimplilearnProject.aabp.dao.AdminDao;
import org.SimplilearnProject.aabp.daoImpl.AdminDaoImpl;
import org.SimplilearnProject.aabp.entities.Admin;
import org.SimplilearnProject.aabp.models.AdminLoginModel;
import org.SimplilearnProject.aabp.models.LoginModel;
import org.SimplilearnProject.aabp.services.AdminServices;
import org.SimplilearnProject.aabp.servicesImpl.AdminServicesImpl;
```

```
@WebServlet("/admin-login")
```

```
public class AdminLoginController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private AdminServices adminService = new AdminServicesImpl();
    private AdminDao adminDao = new AdminDaoImpl();

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        // Retrieve adminName and password from the request
        String adminName = request.getParameter("adminName");
        String password = request.getParameter("password");

        // Create an AdminLoginModel object
        AdminLoginModel loginModel = new AdminLoginModel();
```

```

loginModel.setAdminName(adminName);

loginModel.setPassword(password);


// Call the getAdmin method of the AdminService with the AdminLoginModel
Admin admin = adminService.getAdmin(loginModel);


if (admin != null) {
    HttpSession session = request.getSession();
    session.setAttribute("admin", admin);


    // Forward the request to the admin home page
    RequestDispatcher rd = request.getRequestDispatcher("admin-home.jsp");
    rd.forward(request, response);
} else {
    RequestDispatcher rd = request.getRequestDispatcher("admin-login.jsp");
    request.setAttribute("error", "Admin Name/Password is invalid");
    rd.forward(request, response);
}
}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Redirect GET requests to the login page
    response.sendRedirect("admin-login.jsp");
}
}

```

```

AdminRegisterController
package org.SimplilearnProject.aabp.controller;

```

```
import java.io.IOException;
```

```
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 org.hibernate.SessionFactory;
```

```
import org.SimplilearnProject.aabp.dao.AdminDao;
```

```
import org.SimplilearnProject.aabp.daoImpl.AdminDaoImpl;
```

```
import org.SimplilearnProject.aabp.entities.Admin;
```

```
import org.SimplilearnProject.aabp.config.HibConfig;
```

```
@WebServlet("/AdminRegister")
```

```
public class AdminRegisterController extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private AdminDao adminDao;
```

```
    @Override
```

```
    public void init() throws ServletException {
```

```
        // Initialize AdminDaoImpl with the session factory
```

```
        SessionFactory sessionFactory = HibConfig.getSessionFactory();
```

```
        adminDao = new AdminDaoImpl();
```

```
        adminDao.setSessionFactory(sessionFactory);
```

```
    }
```

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Retrieve admin registration form data
    String adminName = request.getParameter("adminName");
    String email = request.getParameter("email");
    String password = request.getParameter("password");

    // Validate form data (e.g., check for empty fields, password strength, etc.)
    boolean isValid = validateForm(adminName, email, password);

    if (isValid) {
        // Create a new Admin object and set the form data
        Admin admin = new Admin();
        admin.setAdminName(adminName);
        admin.setEmail(email);
        admin.setPassword(password);

        // Save the admin to the database using the AdminDao
        adminDao.saveAdmin(admin);

        // Forward to a success page after successful registration
        RequestDispatcher dispatcher = request.getRequestDispatcher("AdminRegisterSucc.jsp");
        dispatcher.forward(request, response);
    } else {
        // If form data is invalid, forward back to the registration page with an error message
        RequestDispatcher dispatcher = request.getRequestDispatcher("admin-register.jsp?error=1");
        dispatcher.forward(request, response);
    }
}
```

```

    }
}

private boolean validateForm(String adminName, String email, String password) {
    // Perform form validation logic here
    // You can add your own validation rules based on your requirements
    // For example, check for empty fields, validate email format, password strength, etc.

    // In this example, we are checking if all form fields are non-empty
    return !adminName.isEmpty()
        && !email.isEmpty()
        && !password.isEmpty();
}
}

```

AirlineController

```

package org.SimplilearnProject.aabp.controller;

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 org.SimplilearnProject.aabp.dao.AirlineDao;
import org.SimplilearnProject.aabp.daoImpl.AirlineDaoImpl;

```

```

import org.SimplilearnProject.aabp.entities.Airline;
import org.SimplilearnProject.aabp.models.AirlineModel;

import org.SimplilearnProject.aabp.services.AirlineServices;
import org.SimplilearnProject.aabp.servicesImpl.AirlineServicesImpl;

@WebServlet("/addAirline")
public class AirlineController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    private AirlineServices airlineService;

    @Override
    public void init() throws ServletException {
        // Initialize AirlineService with the AirlineDao
        AirlineDao airlineDao = new AirlineDaoImpl();
        airlineService = new AirlineServicesImpl(airlineDao);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        // Retrieve data from the request
        String airlineName = request.getParameter("airlineName");
        String airlineHeadquarters = request.getParameter("airlineHeadquarters");
        String websites = request.getParameter("websites");
        String code = request.getParameter("code");
        String country = request.getParameter("country");

```



```

// Create a new AirlineModel object
AirlineModel airlineModel = new AirlineModel();
airlineModel.setAirlineName(airlineName);
airlineModel.setAirlineHeadquarters(airlineHeadquarters);
airlineModel.setWebsites(websites);
airlineModel.setCode(code);
airlineModel.setCountry(country);

// Call the createAirline method of the airlineService
airlineService.createAirline(airlineModel);

// Set the airline model as an attribute in the request
request.setAttribute("airline", airlineModel);

// Forward the request to the success page
request.getRequestDispatcher("addAirlineSuccess.jsp").forward(request, response);
}
}

```

BookingController

```
package org.SimplilearnProject.aabp.controller;
```

```

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import java.util.HashSet;
import java.util.Set;

```

```
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.SimplilearnProject.aabp.dao.BookingDao;
import org.SimplilearnProject.aabp.dao.FlightDao;
import org.SimplilearnProject.aabp.dao.UserDao;
import org.SimplilearnProject.aabp.daoImpl.BookingDaoImpl;
import org.SimplilearnProject.aabp.daoImpl.FlightDaoImpl;
import org.SimplilearnProject.aabp.daoImpl.UserDaoImpl;
import org.SimplilearnProject.aabp.entities.Booking;
import org.SimplilearnProject.aabp.entities.Flight;
import org.SimplilearnProject.aabp.entities.User;
import org.SimplilearnProject.aabp.models.BookingModel;
import org.SimplilearnProject.aabp.models.FlightModel;
import org.SimplilearnProject.aabp.models.UserModel;
import org.SimplilearnProject.aabp.services.BookingServices;
import org.SimplilearnProject.aabp.services.FlightServices;
import org.SimplilearnProject.aabp.servicesImpl.BookingServicesImpl;
import org.SimplilearnProject.aabp.servicesImpl.FlightServicesImpl;
```

```
@WebServlet("/addBooking")
```

```
public class BookingController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private BookingServices bookingService;
    private FlightServices flightService;
```

@Override

```
public void init() throws ServletException {  
    // Initialize services with respective DAO implementations  
    BookingDao bookingDao = new BookingDaoImpl();  
    FlightDao flightDao = new FlightDaoImpl();  
    UserDao userDao = new UserDaoImpl();  
  
    bookingService = new BookingServicesImpl(bookingDao);  
    flightService = new FlightServicesImpl(flightDao);  
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
  
    // Retrieve data from the request  
    int userId = Integer.parseInt(request.getParameter("userId"));  
    Date bookingDate = new Date(request.getParameter("bookingDate"));  
    int passengerCount = Integer.parseInt(request.getParameter("passengerCount"));  
    int totalPrice = Integer.parseInt(request.getParameter("totalPrice"));  
    String[] flightNumbers = request.getParameterValues("flightNumber");  
  
    // Create a new UserModel object  
    UserModel userModel = new UserModel();  
    userModel.setUserId(userId);  
  
    // Create a new BookingModel object  
    BookingModel bookingModel = new BookingModel();
```

```
bookingModel.setBookingDate(bookingDate);  
bookingModel.setPassengerCount(passengerCount);  
bookingModel.setTotalPrice(totalPrice);  
bookingModel.setUser(userModel);
```

```
// Set the flights  
Set<FlightModel> flightModels = new HashSet<>();  
for (String flightNumber : flightNumbers) {  
    FlightModel flightModel = new FlightModel();  
    flightModel.setFlightNumber(flightNumber);  
    flightModels.add(flightModel);  
}  
bookingModel.setFlights(flightModels);
```

```
// Call the createBooking method of the bookingService  
bookingService.createBooking(bookingModel);
```

```
// Set the booking model as an attribute in the request  
request.setAttribute("booking", bookingModel);
```

```
// Forward the request to the success page  
request.getRequestDispatcher("addBookingSuccess.jsp").forward(request, response);  
}
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    // Forward the request to the addBooking.jsp page
```

```
    request.getRequestDispatcher("addBooking.jsp").forward(request, response);  
}  
}
```

FlightController

```
package org.SimplilearnProject.aabp.controller;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
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 org.SimplilearnProject.aabp.dao.FlightDao;
```

```
import org.SimplilearnProject.aabp.daoImpl.FlightDaoImpl;
```

```
import org.SimplilearnProject.aabp.models.FlightModel;
```

```
import org.SimplilearnProject.aabp.services.FlightServices;
```

```
import org.SimplilearnProject.aabp.servicesImpl.FlightServicesImpl;
```

```
@WebServlet("/addflight")
```

```
public class FlightController extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private FlightServices flightServices;
```

@Override

```
public void init() throws ServletException {  
    // Initialize FlightServices with the FlightDao  
    FlightDao flightDao = new FlightDaoImpl();  
    flightServices = new FlightServicesImpl(flightDao);  
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();
```

```
    // Retrieve data from the request  
    String flightNumber = request.getParameter("flightNumber");  
    String source = request.getParameter("source");  
    String destination = request.getParameter("destination");
```

```
    // Create a new FlightModel object  
    FlightModel flightModel = new FlightModel();  
    flightModel.setFlightNumber(flightNumber);  
    flightModel.setSource(source);  
    flightModel.setDestination(destination);
```

```
    // Call the addFlight method of the flightService  
    flightServices.addFlight(flightModel);
```

```
    // Set the flight model as an attribute in the request  
    request.setAttribute("flight", flightModel);
```

```

        // Forward the request to the success page
        request.getRequestDispatcher("addFlightSuccess.jsp").forward(request, response);
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        // Retrieve all flights from the flightService
        List<FlightModel> flights = flightServices.getAllFlights();

        // Set the flights as an attribute in the request
        request.setAttribute("flights", flights);

        // Forward the request to the flights.jsp page
        RequestDispatcher dispatcher = request.getRequestDispatcher("flights.jsp");
        dispatcher.forward(request, response);
    }
}

```

```

LoginController
package org.SimplilearnProject.aabp.controller;

import java.io.IOException;
import java.io.PrintWriter;

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 javax.servlet.http.HttpSession;

import org.SimplilearnProject.aabp.dao.UserDao;
import org.SimplilearnProject.aabp.daoImpl.UserDaoImpl;
import org.SimplilearnProject.aabp.entities.User;
import org.SimplilearnProject.aabp.models.LoginModel;
import org.SimplilearnProject.aabp.models.UserModel;
import
org.SimplilearnProject.aabp.services.UserService;
import
org.SimplilearnProject.aabp.servicesImpl.UserServiceImp
l;

@WebServlet("/login")
public class LoginController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private UserService userService = new
UserServiceImpl();
    private UserDao userDao = new UserDaoImpl();

    protected void doPost(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        // Retrieve username and password from the
request
        String username =
request.getParameter("username");
        String password =
request.getParameter("password");

        // Create a LoginModel object
        LoginModel loginModel = new LoginModel();
        loginModel.setUsername(username);
        loginModel.setPassword(password);

```



```

        // Call the getUser method of the UserService
with the LoginModel
        User user = userService.getUser(loginModel);

        if (user != null) {
            HttpSession session = request.getSession();
            session.setAttribute("user", user);

            UserDao userDao = new UserDaoImpl(); //
Create an instance of UserDao

            // Pass the retrieved user object to the
view
            request.setAttribute("user", user);

            // Forward the request to the home.jsp page

            // Get the user by username using the
UserDao
            User userByUsername =
userDao.getUserByUsername(username);
            request.setAttribute("user",
userByUsername);

            RequestDispatcher rd =
request.getRequestDispatcher("home.jsp");
            rd.forward(request, response);
        } else {
            RequestDispatcher rd =
request.getRequestDispatcher("login.jsp");
            request.setAttribute("error",
"Username/Password is invalid");
            rd.forward(request, response);
        }
    }
}

```

PaymentController

```
package org.SimplilearnProject.aabp.controller;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import java.util.Date;
```

```
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.SimplilearnProject.aabp.dao.PaymentDao;
```

```
import org.SimplilearnProject.aabp.daoImpl.PaymentDaoImpl;
```

```
import org.SimplilearnProject.aabp.models.PaymentModel;
```

```
import org.SimplilearnProject.aabp.services.PaymentServices;
```

```
import org.SimplilearnProject.aabp.servicesImpl.PaymentServicesImpl;
```

```
@WebServlet("/addPayment")
```

```
public class PaymentController extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private PaymentServices paymentService;
```

```
    @Override
```

```
    public void init() throws ServletException {
```

```
        // Initialize PaymentService with the PaymentDao
```

```
        PaymentDao paymentDao = new PaymentDaoImpl();
```

```
paymentService = new PaymentServicesImpl(paymentDao);  
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();  
  
    // Retrieve data from the request  
    int bookingId = Integer.parseInt(request.getParameter("bookingId"));  
    int paymentAmount = Integer.parseInt(request.getParameter("paymentAmount"));  
    Date paymentDate = new Date(request.getParameter("paymentDate"));  
    String paymentMethod = request.getParameter("paymentMethod");  
  
    // Create a new PaymentModel object  
    PaymentModel paymentModel = new PaymentModel();  
    paymentModel.setPaymentId(bookingId); // Assuming paymentId is the same as bookingId  
    paymentModel.setPaymentAmount(paymentAmount);  
    paymentModel.setPaymentDate(paymentDate);  
    paymentModel.setPaymentMethod(paymentMethod);  
  
    // Call the processPayment method of the paymentService  
    paymentService.processPayment(paymentModel);  
  
    // Set the payment model as an attribute in the request  
    request.setAttribute("payment", paymentModel);  
  
    // Forward the request to the success page  
    request.getRequestDispatcher("addPaymentSuccess.jsp").forward(request, response);
```

```
}  
}
```

PlaceController

```
package org.SimplilearnProject.aabp.controller;  
  
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 org.SimplilearnProject.aabp.dao.PlaceDao;  
import org.SimplilearnProject.aabp.daoImpl.PlaceDaoImpl;  
import org.SimplilearnProject.aabp.entities.Place;  
import org.SimplilearnProject.aabp.models.PlaceModel;  
import org.SimplilearnProject.aabp.services.PlaceServices;  
import org.SimplilearnProject.aabp.servicesImpl.PlaceServicesImpl;  
  
@WebServlet("/addPlace")  
public class PlaceController extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
    private PlaceServices placeService;  
  
    @Override
```

```
public void init() throws ServletException {  
    // Initialize PlaceService with the PlaceDao  
    PlaceDao placeDao = new PlaceDaoImpl();  
    placeService = new PlaceServicesImpl(placeDao);  
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();
```

```
    // Retrieve data from the request  
    String pinCodeStr = request.getParameter("pinCode");  
    String placeName = request.getParameter("placeName");  
    String country = request.getParameter("country");
```

```
    // Create a new PlaceModel object  
    PlaceModel placeModel = new PlaceModel();  
    placeModel.setPinCode(Integer.parseInt(pinCodeStr)); // Updated to use pinCode  
    placeModel.setPlaceName(placeName);  
    placeModel.setCountry(country);
```

```
    // Call the createPlace method of the placeService  
    placeService.createPlace(placeModel);
```

```
    // Set the place model as an attribute in the request  
    request.setAttribute("place", placeModel);
```

```
    // Forward the request to the success page
```

```
        request.getRequestDispatcher("addPlaceSuccess.jsp").forward(request, response);
    }
}
```

RegisterController

```
package org.SimplilearnProject.aabp.controller;
```

```
import java.io.IOException;
```

```
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 org.hibernate.SessionFactory;
```

```
import org.SimplilearnProject.aabp.dao.UserDao;
```

```
import org.SimplilearnProject.aabp.daoImpl.UserDaoImpl;
```

```
import org.SimplilearnProject.aabp.entities.User;
```

```
import org.SimplilearnProject.aabp.models.UserModel;
```

```
import org.SimplilearnProject.aabp.config.HibConfig;
```

```
@WebServlet("/register")
```

```
public class RegisterController extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private UserDao userDao;
```

@Override

```
public void init() throws ServletException {  
    // Initialize UserDaoImpl with the session factory  
    SessionFactory sessionFactory = HibConfig.getSessionFactory();  
    userDao = new UserDaoImpl();  
    userDao.setSessionFactory(sessionFactory);  
}
```

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    // Retrieve user registration form data  
    String username = request.getParameter("username");  
    String email = request.getParameter("email");  
    String password = request.getParameter("password");  
    String confirmPassword = request.getParameter("confirmPassword");  
  
    // Validate form data (e.g., check for empty fields, password match, etc.)  
    boolean isValid = validateForm(username, email, password, confirmPassword);  
  
    if (isValid) {  
        // Create a new User object and set the form data  
        User user = new User();  
        user.setUserName(username);  
        user.setEmail(email);  
        user.setPassword(password);  
  
        // Save the user to the database using the UserDao  
        userDao.saveUser(user);  
    }  
}
```

```

        // Forward to a success page after successful registration
        RequestDispatcher dispatcher = request.getRequestDispatcher("registration-success.jsp");
        dispatcher.forward(request, response);
    } else {
        // If form data is invalid, forward back to the registration page with an error message
        RequestDispatcher dispatcher = request.getRequestDispatcher("register.jsp?error=1");
        dispatcher.forward(request, response);
    }
}

private boolean validateForm(String username, String email, String password, String
confirmPassword) {
    // Perform form validation logic here
    // You can add your own validation rules based on your requirements
    // For example, check for empty fields, validate email format, compare passwords, etc.

    // In this example, we are checking if all form fields are non-empty
    return !username.isEmpty()
        && !email.isEmpty()
        && !password.isEmpty()
        && !confirmPassword.isEmpty();
}
}

```

```

AddAirlineControllerSuccess
package org.SimplilearnProject.aabp.controllerSucc;

```

```

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;

@WebServlet("/addAirlineSuccess")
public class AddAirlineControllerSuccess extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");

        // Display a success message
        response.getWriter().println("<h2>Airline added successfully!</h2>");
        response.getWriter().println("<p>Thank you for adding the airline.</p>");
    }
}

```

```

AddFlightSuccessController
package org.SimplilearnProject.aabp.controllerSucc;

```

```

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;

@WebServlet("/addFlightSuccess")

public class AddFlightSuccessController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");

        // Display a success message
        response.getWriter().println("<h2>Flight added successfully!</h2>");
        response.getWriter().println("<p>Thank you for adding the flight.</p>");
    }
}

```

```

AddPlaceControllerSuccess
package org.SimplilearnProject.aabp.controllerSucc;

```

```

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;

```

```

@WebServlet("/addPlaceSuccess")

public class AddPlaceControllerSuccess extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");

        // Display a success message
        response.getWriter().println("<h2>Place added successfully!</h2>");
        response.getWriter().println("<p>Thank you for adding the place.</p>");
    }
}

```

```

AdminRegistrationSuccessController
package org.SimplilearnProject.aabp.controllerSucc;

```

```

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;

```

```

@WebServlet("/AdminRegisterSucc")

public class AdminRegistrationSuccessController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    @Override

```

```

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    // Handle GET request for the admin registration success page

    // You can add any logic or retrieve data related to the successful registration here

    // Forward the request to the admin registration success page
    request.getRequestDispatcher("AdminRegisterSucc.jsp").forward(request, response);
}
}

```

```

RegistrationSuccessController
package org.SimplilearnProject.aabp.controllerSucc;

```

```

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;

@WebServlet("/registration-success")
public class RegistrationSuccessController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // Handle GET request for the registration success page
    }
}

```

```

        // You can add any logic or retrieve data related to the successful registration here

        // Forward the request to the registration success page
        request.getRequestDispatcher("registration-success.jsp").forward(request, response);
    }
}

```

AdminDao

```
package org.SimplilearnProject.aabp.dao;
```

```
import org.SimplilearnProject.aabp.entities.Admin;
```

```
import org.hibernate.SessionFactory;
```

```

public interface AdminDao {
    Admin getAdminById(int adminId);
    Admin getAdminByEmail(String email);
    Admin getAdmin(String adminName, String password); // Modified method name
    Admin getAdminByUsername(String adminName); // Modified method name
    void saveAdmin(Admin admin);
    void updateAdmin(Admin admin);
    void deleteAdmin(Admin admin);
    void setSessionFactory(SessionFactory sessionFactory);
}

```

AirlineDao

```
package org.SimplilearnProject.aabp.dao;
```

```
import org.SimplilearnProject.aabp.entities.Airline;
```

```

public interface AirlineDao {
    Airline getAirlineById(int airlineId);
}

```

```
    void saveAirline(Airline airline);  
    void updateAirline(Airline airline);  
    void deleteAirline(Airline airline);  
}
```

BookingDao

```
package org.SimplilearnProject.aabp.dao;  
  
import org.SimplilearnProject.aabp.entities.Booking;  
  
public interface BookingDao {  
    Booking getBookingById(int bookingId);  
    void saveBooking(Booking booking);  
    void updateBooking(Booking booking);  
    void deleteBooking(Booking booking);  
}
```

FlightDao

```
package org.SimplilearnProject.aabp.dao;  
  
import java.util.List;  
  
import org.SimplilearnProject.aabp.entities.Flight;  
import org.hibernate.SessionFactory;  
  
public interface FlightDao {  
    void setSessionFactory(SessionFactory sessionFactory);  
    void saveFlight(Flight flight);  
    Flight getFlightById(int flightId);  
    List<Flight> getAllFlights();  
    void updateFlight(Flight flight);  
    void deleteFlight(int flightId);  
    List<Flight> getFlightsBySource(String source);  
    List<Flight> getFlightsByDestination(String destination);  
}
```

```
List<Flight> getFlightsBySourceAndDestination(String source, String destination);  
  
}
```

PaymentDao

```
package org.SimplilearnProject.aabp.dao;  
  
import org.SimplilearnProject.aabp.entities.Payment;  
  
public interface PaymentDao {  
    Payment getPaymentById(int paymentId);  
    void savePayment(Payment payment);  
    void updatePayment(Payment payment);  
    void deletePayment(Payment payment);  
}
```

PlaceDao

```
package org.SimplilearnProject.aabp.dao;  
  
import org.SimplilearnProject.aabp.entities.Place;  
  
public interface PlaceDao {  
    Place getPlaceByPinCode(int pinCode);  
    void savePlace(Place place);  
    void updatePlace(Place place);  
    void deletePlace(Place place);  
}
```

UserDao

```
package org.SimplilearnProject.aabp.dao;  
  
import org.SimplilearnProject.aabp.entities.User;  
  
import org.hibernate.SessionFactory;  
  
public interface UserDao {  
    User getUserById(int userId);  
}
```

```
User getUserByEmail(String email);
User getUserByUsername(String username);
User getUser(String username, String password);

void saveUser(User user);
void updateUser(User user);
void deleteUser(User user);
void insert(User user);
void setSessionFactory(SessionFactory sessionFactory);
}
```

`AdminDaoImpl`

```
package org.SimplilearnProject.aabp.daoImpl;
```

```
import org.SimplilearnProject.aabp.dao.AdminDao;
import org.SimplilearnProject.aabp.entities.Admin;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.SimplilearnProject.aabp.config.HibConfig;
```

```
public class AdminDaoImpl implements AdminDao {
    private SessionFactory sessionFactory;

    public AdminDaoImpl() {
        sessionFactory = HibConfig.getSessionFactory();
    }
}
```

```
@Override
```



```
public Admin getAdminById(int adminId) {  
    Session session = sessionFactory.openSession();  
    Admin admin = session.get(Admin.class, adminId);  
    session.close();  
    return admin;  
}
```

@Override

```
public Admin getAdminByEmail(String email) {  
    Session session = sessionFactory.openSession();  
    Admin admin = null;  
    try {  
        admin = session.createQuery("SELECT a FROM Admin a WHERE a.email = :email", Admin.class)  
            .setParameter("email", email)  
            .uniqueResult();  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        session.close();  
    }  
    return admin;  
}
```

@Override

```
public Admin getAdmin(String adminName, String password) {  
    Session session = sessionFactory.openSession();  
    Admin admin = null;  
    try {
```

```

        admin = session.createQuery("SELECT a FROM Admin a WHERE a.adminName = :adminName
AND a.password = :password", Admin.class)

        .setParameter("adminName", adminName)

        .setParameter("password", password)

        .uniqueResult();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        session.close();
    }
    return admin;
}

```

```

@Override
public Admin getAdminByUsername(String adminName) {
    Session session = sessionFactory.openSession();

    Admin admin = null;

    try {
        admin = session.createQuery("SELECT a FROM Admin a WHERE a.adminName = :adminName",
Admin.class)

        .setParameter("adminName", adminName)

        .uniqueResult();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        session.close();
    }
    return admin;
}

```

@Override

```
public void saveAdmin(Admin admin) {  
    Session session = null;  
    Transaction transaction = null;  
  
    try {  
        session = sessionFactory.openSession();  
        transaction = session.beginTransaction();  
        session.persist(admin);  
        transaction.commit();  
    } catch (Exception e) {  
        if (transaction != null) {  
            transaction.rollback();  
        }  
        e.printStackTrace();  
    } finally {  
        if (session != null) {  
            session.close();  
        }  
    }  
}
```

@Override

```
public void updateAdmin(Admin admin) {  
    Session session = null;  
    Transaction transaction = null;  
  
    try {
```

```

        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        session.merge(admin);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

@Override

```

public void deleteAdmin(Admin admin) {
    Session session = null;
    Transaction transaction = null;

    try {
        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        session.delete(admin);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
    }
}

```

```

    }

    e.printStackTrace();
} finally {
    if (session != null) {
        session.close();
    }
}
}

@Override

public void setSessionFactory(SessionFactory sessionFactory) {
    this.sessionFactory = sessionFactory;
}
}

```

AirlineDaoImpl

```

package org.SimplilearnProject.aabp.daoImpl;

import org.SimplilearnProject.aabp.dao.AirlineDao;
import org.SimplilearnProject.aabp.entities.Airline;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.SimplilearnProject.aabp.config.HibConfig;

public class AirlineDaoImpl implements AirlineDao {
    private SessionFactory sessionFactory;

    public AirlineDaoImpl() {

```

```
    sessionFactory = HibConfig.getSessionFactory();  
}
```

@Override

```
public Airline getAirlineById(int airlineId) {  
    Session session = sessionFactory.openSession();  
    Airline airline = session.get(Airline.class, airlineId);  
    session.close();  
    return airline;  
}
```

@Override

```
public void saveAirline(Airline airline) {  
    Session session = null;  
    Transaction transaction = null;  
  
    try {  
        session = sessionFactory.openSession();  
        transaction = session.beginTransaction();  
        session.persist(airline);  
        transaction.commit();  
    } catch (Exception e) {  
        if (transaction != null) {  
            transaction.rollback();  
        }  
        e.printStackTrace();  
    } finally {  
        if (session != null) {  
            session.close();  
        }  
    }  
}
```

```
    }  
    }  
}
```

@Override

```
public void updateAirline(Airline airline) {  
    Session session = null;  
    Transaction transaction = null;  
  
    try {  
        session = sessionFactory.openSession();  
        transaction = session.beginTransaction();  
        session.merge(airline);  
        transaction.commit();  
    } catch (Exception e) {  
        if (transaction != null) {  
            transaction.rollback();  
        }  
        e.printStackTrace();  
    } finally {  
        if (session != null) {  
            session.close();  
        }  
    }  
}
```

@Override

```
public void deleteAirline(Airline airline) {  
    Session session = null;
```

```

Transaction transaction = null;

try {
    session = sessionFactory.openSession();
    transaction = session.beginTransaction();
    session.delete(airline);
    transaction.commit();
} catch (Exception e) {
    if (transaction != null) {
        transaction.rollback();
    }
    e.printStackTrace();
} finally {
    if (session != null) {
        session.close();
    }
}
}
}

```

BookingDaoImpl

```

package org.SimplilearnProject.aabp.daoImpl;

import org.SimplilearnProject.aabp.dao.BookingDao;
import org.SimplilearnProject.aabp.entities.Booking;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.SimplilearnProject.aabp.config.HibConfig;

```



```
public class BookingDaoImpl implements BookingDao {  
    private SessionFactory sessionFactory;  
  
    public BookingDaoImpl() {  
        sessionFactory = HibConfig.getSessionFactory();  
    }  
}
```

@Override

```
public Booking getBookingById(int bookingId) {  
    Session session = sessionFactory.openSession();  
    Booking booking = session.get(Booking.class, bookingId);  
    session.close();  
    return booking;  
}
```

@Override

```
public void saveBooking(Booking booking) {  
    Session session = null;  
    Transaction transaction = null;  
  
    try {  
        session = sessionFactory.openSession();  
        transaction = session.beginTransaction();  
        session.persist(booking);  
        transaction.commit();  
    } catch (Exception e) {  
        if (transaction != null) {  
            transaction.rollback();  
        }  
    }  
}
```

```

    }
    e.printStackTrace();
} finally {
    if (session != null) {
        session.close();
    }
}
}

```

@Override

```

public void updateBooking(Booking booking) {
    Session session = null;
    Transaction transaction = null;

    try {
        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        session.merge(booking);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

```

    }

    @Override
    public void deleteBooking(Booking booking) {
        Session session = null;
        Transaction transaction = null;

        try {
            session = sessionFactory.openSession();
            transaction = session.beginTransaction();
            session.delete(booking);
            transaction.commit();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        } finally {
            if (session != null) {
                session.close();
            }
        }
    }
}

```

```
FlightDaoImpl
```

```
package org.SimplilearnProject.aabp.daoImpl;
```

```
import org.SimplilearnProject.aabp.dao.FlightDao;
```

```
import org.SimplilearnProject.aabp.entities.Flight;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.SimplilearnProject.aabp.config.HibConfig;
```

```
import java.util.List;
```

```
public class FlightDaoImpl implements FlightDao {
    private SessionFactory sessionFactory;

    @Override
    public void setSessionFactory(SessionFactory sessionFactory) {
        this.sessionFactory = sessionFactory;
    }
}
```

```
public FlightDaoImpl() {
    sessionFactory = HibConfig.getSessionFactory();
}
```

```
@Override
public Flight getFlightById(int flightId) {
    Session session = sessionFactory.openSession();
    Flight flight = session.get(Flight.class, flightId);
    session.close();
    return flight;
}
```

```
@Override
```

```

public void saveFlight(Flight flight) {

    Session session = null;

    Transaction transaction = null;

    try {

        session = sessionFactory.openSession();

        transaction = session.beginTransaction();

        session.persist(flight);

        transaction.commit();

    } catch (Exception e) {

        if (transaction != null) {

            transaction.rollback();

        }

        e.printStackTrace();

    } finally {

        if (session != null) {

            session.close();

        }

    }

}

```

@Override

```

public void updateFlight(Flight flight) {

    Session session = null;

    Transaction transaction = null;

    try {

        session = sessionFactory.openSession();

        transaction = session.beginTransaction();

    }

```

```

        session.merge(flight);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

@Override

```

public void deleteFlight(int flightId) {
    Session session = null;
    Transaction transaction = null;

    try {
        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        Flight flight = session.get(Flight.class, flightId);
        if (flight != null) {
            session.delete(flight);
        }
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {

```

```

        transaction.rollback();
    }
    e.printStackTrace();
} finally {
    if (session != null) {
        session.close();
    }
}
}

```

@Override

```

public List<Flight> getAllFlights() {
    Session session = null;

    try {
        session = sessionFactory.openSession();

        return session.createQuery("FROM Flight", Flight.class)
            .list();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
    return null;
}

```

@Override

```
public List<Flight> getFlightsBySource(String source) {
```

```
    Session session = null;
```

```
    try {
```

```
        session = sessionFactory.openSession();
```

```
        return session.createQuery("FROM Flight WHERE flightSource = :source", Flight.class)
```

```
            .setParameter("source", source)
```

```
            .list();
```

```
    } catch (Exception e) {
```

```
        e.printStackTrace();
```

```
    } finally {
```

```
        if (session != null) {
```

```
            session.close();
```

```
        }
```

```
    }
```

```
    return null;
```

```
}
```

@Override

```
public List<Flight> getFlightsByDestination(String destination) {
```

```
    Session session = null;
```

```
    try {
```

```
        session = sessionFactory.openSession();
```

```
        return session.createQuery("FROM Flight WHERE flightDestination = :destination", Flight.class)
```

```
            .setParameter("destination", destination)
```



```

        .list();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
    return null;
}

```

@Override

```

public List<Flight> getFlightsBySourceAndDestination(String source, String destination) {
    Session session = null;

    try {
        session = sessionFactory.openSession();

        return session.createQuery("FROM Flight WHERE flightSource = :source AND flightDestination = :destination", Flight.class)
            .setParameter("source", source)
            .setParameter("destination", destination)
            .list();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

```
    }  
    return null;  
}  
}
```

```
PaymentDaoImpl
```

```
package org.SimplilearnProject.aabp.daoImpl;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
import org.hibernate.Transaction;
```

```
import org.hibernate.cfg.Configuration;
```

```
import org.SimplilearnProject.aabp.dao.PaymentDao;
```

```
import org.SimplilearnProject.aabp.entities.Payment;
```

```
public class PaymentDaoImpl implements PaymentDao {
```

```
    private SessionFactory sessionFactory;
```

```
    public PaymentDaoImpl() {
```

```
        sessionFactory = new Configuration().configure().buildSessionFactory();
```

```
    }
```

```
@Override
```

```
public Payment getPaymentById(int paymentId) {
```

```
    Session session = sessionFactory.openSession();
```

```
    Payment payment = session.get(Payment.class, paymentId);
```

```
    session.close();
```

```
    return payment;
```

```
}
```

```
@Override
```

```
public void savePayment(Payment payment) {  
    Session session = sessionFactory.openSession();  
    Transaction transaction = null;  
    try {  
        transaction = session.beginTransaction();  
        session.save(payment);  
        transaction.commit();  
    } catch (Exception e) {  
        if (transaction != null) {  
            transaction.rollback();  
        }  
        e.printStackTrace();  
    } finally {  
        session.close();  
    }  
}
```

```
@Override
```

```
public void updatePayment(Payment payment) {  
    Session session = sessionFactory.openSession();  
    Transaction transaction = null;  
    try {  
        transaction = session.beginTransaction();  
        session.update(payment);  
        transaction.commit();  
    } catch (Exception e) {
```

```

        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        session.close();
    }
}

```

@Override

```

public void deletePayment(Payment payment) {
    Session session = sessionFactory.openSession();
    Transaction transaction = null;
    try {
        transaction = session.beginTransaction();
        session.delete(payment);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        session.close();
    }
}
}

```

PlaceDaoImpl

```
package org.SimplilearnProject.aabp.daoImpl;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import org.SimplilearnProject.aabp.dao.PlaceDao;
import org.SimplilearnProject.aabp.entities.Place;
import org.SimplilearnProject.aabp.config.HibConfig;

public class PlaceDaoImpl implements PlaceDao {
    private SessionFactory sessionFactory;

    public PlaceDaoImpl() {
        sessionFactory = HibConfig.getSessionFactory();
    }

    @Override
    public Place getPlaceByPinCode(int pinCode) {
        Session session = sessionFactory.openSession();
        Place place = session.get(Place.class, pinCode);
        session.close();
        return place;
    }

    @Override
    public void savePlace(Place place) {
        Session session = null;
        Transaction transaction = null;
```

```

try {
    session = sessionFactory.openSession();
    transaction = session.beginTransaction();
    session.persist(place);
    transaction.commit();
} catch (Exception e) {
    if (transaction != null) {
        transaction.rollback();
    }
    e.printStackTrace();
} finally {
    if (session != null) {
        session.close();
    }
}
}

```

@Override

```

public void updatePlace(Place place) {
    Session session = null;
    Transaction transaction = null;

    try {
        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        session.merge(place);
        transaction.commit();
    } catch (Exception e) {

```

```

        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

@Override

```

public void deletePlace(Place place) {
    Session session = null;
    Transaction transaction = null;

    try {
        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        session.delete(place);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

```
    }  
    }  
    }  
}
```

UserDaoImpl

```
package org.SimplilearnProject.aabp.daoImpl;
```

```
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.SimplilearnProject.aabp.config.HibConfig;  
import org.SimplilearnProject.aabp.dao.UserDao;  
import org.SimplilearnProject.aabp.entities.User;
```

```
public class UserDaoImpl implements UserDao {
```

```
    private SessionFactory sessionFactory;
```

```
    @Override
```

```
    public void setSessionFactory(SessionFactory sessionFactory) {  
        this.sessionFactory = sessionFactory;  
    }
```

```
    @Override
```

```
    public void saveUser(User user) {  
        Session session = null;  
        Transaction transaction = null;  
  
        try {
```



```

        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        session.save(user);
        transaction.commit();
    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

@Override

```

public User getUser(String username, String password) {
    SessionFactory factory = HibConfig.getSessionFactory();
    Session session = factory.openSession();
    try {

        return session.createQuery("FROM User WHERE username = :username AND password = :password", User.class)
            .setParameter("username", username)
            .setParameter("password", password)
            .uniqueResult();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```
    } finally {  
        if (session != null) {  
            session.close();  
        }  
    }  
    return null;  
}
```

@Override

```
public User getUserByUsername(String username) {  
    Session session = null;  
  
    try {  
        session = sessionFactory.openSession();  
  
        return session.createQuery("FROM User WHERE username = :username", User.class)  
            .setParameter("username", username)  
            .uniqueResult();  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        if (session != null) {  
            session.close();  
        }  
    }  
    return null;  
}
```

@Override

```
public User getUserById(int id) {  
    Session session = null;  
  
    try {  
        session = sessionFactory.openSession();  
  
        return session.get(User.class, id);  
    } catch (Exception e) {  
        e.printStackTrace();  
    } finally {  
        if (session != null) {  
            session.close();  
        }  
    }  
    return null;  
}
```

@Override

```
public void updateUser(User user) {  
    Session session = null;  
    Transaction transaction = null;  
  
    try {  
        session = sessionFactory.openSession();  
        transaction = session.beginTransaction();  
        session.update(user);  
        transaction.commit();  
    }
```

```

    } catch (Exception e) {
        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

@Override

```

public void insert(User user) {
    // Implementation to insert a new user into the database
    // ...
}

```

@Override

```

public void deleteUser(User user) {
    Session session = null;
    Transaction transaction = null;

    try {
        session = sessionFactory.openSession();
        transaction = session.beginTransaction();
        session.delete(user);
        transaction.commit();
    } catch (Exception e) {

```

```

        if (transaction != null) {
            transaction.rollback();
        }
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

@Override

```

public User getUserByEmail(String email) {
    Session session = null;

    try {
        session = sessionFactory.openSession();

        return session.createQuery("FROM User WHERE email = :email", User.class)
            .setParameter("email", email)
            .uniqueResult();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

```

```
        return null;
    }
}
```

Admin

```
package org.SimplilearnProject.aabp.entities;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

@Entity
public class Admin {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int adminId;
    private String adminName;
    private String password;
    private String email;

    @ManyToOne
    @JoinColumn(name = "flight_id")
    private Flight flight;

    public Admin() {
```

```
}
```

```
public Admin(String adminName, String password, String email) {  
    this.adminName = adminName;  
    this.password = password;  
    this.email = email;  
}
```

```
public int getAdminId() {  
    return adminId;  
}
```

```
public void setAdminId(int adminId) {  
    this.adminId = adminId;  
}
```

```
public String getAdminName() {  
    return adminName;  
}
```

```
public void setAdminName(String adminName) {  
    this.adminName = adminName;  
}
```

```
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 Flight getFlight() {
        return flight;
    }

    public void setFlight(Flight flight) {
        this.flight = flight;
    }
}

```

Airline

```
package org.SimplilearnProject.aabp.entities;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```



```
import javax.persistence.Id;
```

```
import javax.persistence.OneToMany;
```

```
@Entity
```

```
public class Airline {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private int airlineId;
```

```
    private String airlineName;
```

```
    private String airlineHeadquarters;
```

```
    private String websites;
```

```
    private String code;
```

```
    private String country;
```

```
    @OneToMany(mappedBy = "airline")
```

```
    private Set<Flight> flights = new HashSet<>();
```

```
    public Airline() {
```

```
    }
```

```
    public Airline(String airlineName, String airlineHeadquarters, String websites, String code, String country) {
```

```
        this.airlineName = airlineName;
```

```
        this.airlineHeadquarters = airlineHeadquarters;
```

```
        this.websites = websites;
```

```
        this.code = code;
```

```
        this.country = country;
```

```
    }
```

```
public int getAirlineId() {  
    return airlineId;  
}
```

```
public void setAirlineId(int airlineId) {  
    this.airlineId = airlineId;  
}
```

```
public String getAirlineName() {  
    return airlineName;  
}
```

```
public void setAirlineName(String airlineName) {  
    this.airlineName = airlineName;  
}
```

```
public String getAirlineHeadquarters() {  
    return airlineHeadquarters;  
}
```

```
public void setAirlineHeadquarters(String airlineHeadquarters) {  
    this.airlineHeadquarters = airlineHeadquarters;  
}
```

```
public String getWebsites() {  
    return websites;  
}
```

```
public void setWebsites(String websites) {
```

```
        this.websites = websites;
    }
```

```
    public String getCode() {
        return code;
    }
```

```
    public void setCode(String code) {
        this.code = code;
    }
```

```
    public String getCountry() {
        return country;
    }
```

```
    public void setCountry(String country) {
        this.country = country;
    }
```

```
    public Set<Flight> getFlights() {
        return flights;
    }
```

```
    public void setFlights(Set<Flight> flights) {
        this.flights = flights;
    }
```

```
// Helper methods
```

```
    public void addFlight(Flight flight) {
```

```

        flights.add(flight);
    }

    public void removeFlight(Flight flight) {
        flights.remove(flight);
    }

    public int getFlightCount() {
        return flights.size();
    }
}

```

Booking

```
package org.SimplilearnProject.aabp.entities;
```

```

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToMany;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import java.util.Date;
import java.util.HashSet;
import java.util.Set;

```

```
@Entity
```

```
public class Booking {
```

```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)

private int bookingId;
private Date bookingDate;
private int passengerCount;
private int totalPrice;

@ManyToOne(cascade = CascadeType.ALL)
private User user;

@ManyToMany(cascade = CascadeType.ALL)
private Set<Flight> flights = new HashSet<>();

@OneToMany

private Set<Payment> payments = new HashSet<>();

public int getBookingId() {
    return bookingId;
}

public void setBookingId(int bookingId) {
    this.bookingId = bookingId;
}

public Date getBookingDate() {
    return bookingDate;
}

public void setBookingDate(Date bookingDate) {
```

```
        this.bookingDate = bookingDate;
    }

    public int getPassengerCount() {
        return passengerCount;
    }

    public void setPassengerCount(int passengerCount) {
        this.passengerCount = passengerCount;
    }

    public int getTotalPrice() {
        return totalPrice;
    }

    public void setTotalPrice(int totalPrice) {
        this.totalPrice = totalPrice;
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public Set<Flight> getFlights() {
        return flights;
    }
}
```

```
}
```

```
public void setFlights(Set<Flight> flights) {  
    this.flights = flights;  
}
```

```
public Set<Payment> getPayments() {  
    return payments;  
}
```

```
public void setPayments(Set<Payment> payments) {  
    this.payments = payments;  
}
```

```
// Helper methods
```

```
public void addFlight(Flight flight) {  
    flights.add(flight);  
    flight.getBookings().add(this);  
}
```

```
public void removeFlight(Flight flight) {  
    flights.remove(flight);  
    flight.getBookings().remove(this);  
}
```

```
public int getFlightCount() {  
    return flights.size();  
}
```

```

    public void addPayment(Payment payment) {
        payments.add(payment);
        payment.setBooking(this);
    }

    public void removePayment(Payment payment) {
        payments.remove(payment);
        payment.setBooking(null);
    }

    public int getPaymentAmount() {
        int totalAmount = 0;
        for (Payment payment : payments) {
            totalAmount += payment.getPaymentAmount();
        }
        return totalAmount;
    }
}

```

Flight

```
package org.SimplilearnProject.aabp.entities;
```

```
import java.time.LocalDateTime;
```

```
import java.util.HashSet;
```

```
import java.util.List;
```

```
import java.util.Set;
```

```
import javax.persistence.CascadeType;
```

```
import javax.persistence.Entity;
```



```
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToMany;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
```

@Entity

```
public class Flight {
```

```
    @Id
```

```
    private int flightId;
```

```
    private String airlineName;
```

```
    private int availableSeat;
```

```
    private String flightSource;
```

```
    private String flightDestination;
```

```
    private int ticketPrice;
```

```
    private String flightNumber; // New property for flight number
```

```
    private LocalTime arrivalTime;
```

```
    private LocalTime departureTime;
```

```
    @ManyToOne(cascade = CascadeType.ALL)
```

```
    @JoinColumn(name = "airline_id")
```

```
    private Airline airline;
```

```
    @ManyToOne(cascade = CascadeType.ALL)
```

```
    @JoinColumn(name = "place_id")
```

```
    private Place place;
```

```
    @OneToMany(mappedBy = "flights")
```

```
    private Set<Booking> bookings = new HashSet<>();
```

```
public int getFlightId() {  
    return flightId;  
}
```

```
public void setFlightId(int flightId) {  
    this.flightId = flightId;  
}
```

```
public String getAirlineName() {  
    return airlineName;  
}
```

```
public void setAirlineName(String airlineName) {  
    this.airlineName = airlineName;  
}
```

```
public int getAvailableSeat() {  
    return availableSeat;  
}
```

```
public void setAvailableSeat(int availableSeat) {  
    this.availableSeat = availableSeat;  
}
```

```
public String getFlightSource() {  
    return flightSource;  
}
```

```
public void setFlightSource(String flightSource) {  
    this.flightSource = flightSource;  
}
```

```
public String getFlightDestination() {  
    return flightDestination;  
}
```

```
public void setFlightDestination(String flightDestination) {  
    this.flightDestination = flightDestination;  
}
```

```
public int getTicketPrice() {  
    return ticketPrice;  
}
```

```
public void setTicketPrice(int ticketPrice) {  
    this.ticketPrice = ticketPrice;  
}
```

```
public String getFlightNumber() {  
    return flightNumber;  
}
```

```
public void setFlightNumber(String flightNumber) {  
    this.flightNumber = flightNumber;  
}
```

```
public LocalTime getArrivalTime() {
```

```
    return arrivalTime;
}
```

```
public void setArrivalTime(LocalTime arrivalTime) {
    this.arrivalTime = arrivalTime;
}
```

```
public LocalTime getDepartureTime() {
    return departureTime;
}
```

```
public void setDepartureTime(LocalTime departureTime) {
    this.departureTime = departureTime;
}
```

```
public Airline getAirline() {
    return airline;
}
```

```
public void setAirline(Airline airline) {
    this.airline = airline;
}
```

```
public Place getPlace() {
    return place;
}
```

```
public void setPlace(Place place) {
    this.place = place;
}
```

```
}
```

```
public Set<Booking> getBookings() {  
    return bookings;  
}
```

```
public void setBookings(Set<Booking> bookings) {  
    this.bookings = bookings;  
}
```

```
public String getSource() {  
    return flightSource;  
}
```

```
public void setSource(String source) {  
    this.flightSource = source;  
}
```

```
public String getDestination() {  
    return flightDestination;  
}
```

```
public void setDestination(String destination) {  
    this.flightDestination = destination;  
}
```

```
// Helper methods
```

```
public void addBooking(Booking booking) {
```

```

        bookings.add(booking);
        booking.getFlights().add(this);
    }

    public void removeBooking(Booking booking) {
        bookings.remove(booking);
        booking.getFlights().remove(this);
    }

    public int getBookingCount() {
        return bookings.size();
    }
}

```

Payment

```
package org.SimplilearnProject.aabp.entities;
```

```
import java.util.Date;
```

```
import javax.persistence.CascadeType;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.ManyToOne;
```

```
@Entity
```

```
public class Payment {
```

```
    @Id
```

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
private int paymentId;
```

```
private int paymentAmount;
```

```
private Date paymentDate;
```

```
private String paymentMethod;
```

```
@ManyToOne(cascade = CascadeType.ALL)
```

```
private Booking booking;
```

```
public int getPaymentId() {
```

```
    return paymentId;
```

```
}
```

```
public void setPaymentId(int paymentId) {
```

```
    this.paymentId = paymentId;
```

```
}
```

```
public int getPaymentAmount() {
```

```
    return paymentAmount;
```

```
}
```

```
public void setPaymentAmount(int paymentAmount) {
```

```
    this.paymentAmount = paymentAmount;
```

```
}
```

```
public Date getPaymentDate() {
```

```
    return paymentDate;
```

```
}
```

```
public void setPaymentDate(Date paymentDate) {  
    this.paymentDate = paymentDate;  
}
```

```
public String getPaymentMethod() {  
    return paymentMethod;  
}
```

```
public void setPaymentMethod(String paymentMethod) {  
    this.paymentMethod = paymentMethod;  
}
```

```
public Booking getBooking() {  
    return booking;  
}
```

```
public void setBooking(Booking booking) {  
    this.booking = booking;  
}  
}
```

```
Place  
package org.SimplilearnProject.aabp.entities;
```

```
import java.util.HashSet;  
import java.util.Set;
```

```
import javax.persistence.CascadeType;  
import javax.persistence.Entity;
```



```
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.OneToMany;
```

```
@Entity
```

```
public class Place {
```

```
    @Id
```

```
    private int pinCode; // Renamed placeId to pinCode
```

```
    private String placeName;
```

```
    private String country; // Added the country attribute
```

```
    @OneToMany(cascade = CascadeType.ALL)
```

```
    @JoinTable(
```

```
        name = "Place_Flight",
```

```
        joinColumns = @JoinColumn(name = "place_id"),
```

```
        inverseJoinColumns = @JoinColumn(name = "flight_id")
```

```
    )
```

```
    private Set<Flight> flights = new HashSet<>();
```

```
    public int getPinCode() {
```

```
        return pinCode;
```

```
    }
```

```
    public void setPinCode(int pinCode) {
```

```
        this.pinCode = pinCode;
```

```
    }
```

```
    public String getPlaceName() {
```

```
        return placeName;
    }
}
```

```
public void setPlaceName(String placeName) {
    this.placeName = placeName;
}
}
```

```
public String getCountry() {
    return country;
}
}
```

```
public void setCountry(String country) {
    this.country = country;
}
}
```

```
public Set<Flight> getFlights() {
    return flights;
}
}
```

```
public void setFlights(Set<Flight> flights) {
    this.flights = flights;
}
}
```

```
// Helper methods
```

```
public void addFlight(Flight flight) {
    flights.add(flight);
}
}
```

```
public void removeFlight(Flight flight) {
```

```
        flights.remove(flight);
    }

    public int getFlightCount() {
        return flights.size();
    }
}
```

User

```
package org.SimplilearnProject.aabp.entities;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.OneToOne;
```

```
@Entity
```

```
public class User {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private int userId;
```

```
    private String userName;
```

```
    private String email;
```

```
    private String password;
```

```
    private int mobileNumber;
```

```
@OneToMany(mappedBy = "user")
```

```
private Set<Booking> bookings = new HashSet<>();
```

```
public User() {
```

```
}
```

```
public User(String userName, String email, String password, int mobileNumber) {
```

```
    this.userName = userName;
```

```
    this.email = email;
```

```
    this.password = password;
```

```
    this.mobileNumber = mobileNumber;
```

```
}
```

```
public int getUserId() {
```

```
    return userId;
```

```
}
```

```
public void setUserId(int userId) {
```

```
    this.userId = userId;
```

```
}
```

```
public String getUserName() {
```

```
    return userName;
```

```
}
```

```
public void setUserName(String userName) {
```

```
    this.userName = userName;
```

```
}
```

```
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;  
}
```

```
public int getMobileNumber() {  
    return mobileNumber;  
}
```

```
public void setMobileNumber(int mobileNumber) {  
    this.mobileNumber = mobileNumber;  
}
```

```
public Set<Booking> getBookings() {  
    return bookings;  
}
```

```
public void setBookings(Set<Booking> bookings) {  
    this.bookings = bookings;  
}  
}
```

AdminLoginModel

```
package org.SimplilearnProject.aabp.models;
```

```
public class AdminLoginModel {  
    private String adminName;  
    private String password;  
  
    public String getAdminName() {  
        return adminName;  
    }  
  
    public void setAdminName(String adminName) {  
        this.adminName = adminName;  
    }  
  
    public String getPassword() {  
        return password;  
    }  
  
    public void setPassword(String password) {  
        this.password = password;  
    }  
}
```

AdminModel

```
package org.SimplilearnProject.aabp.models;
```

```
public class AdminModel {  
    private int adminId;  
    private String adminName;  
    private String email;  
    private String password;  
    private String confirmPassword;
```

```
public AdminModel() {  
    }  
  
    public AdminModel(int adminId, String adminName,  
String email, String password, String confirmPassword)  
{  
    this.adminId = adminId;  
    this.adminName = adminName;  
    this.email = email;  
    this.password = password;  
    this.confirmPassword = confirmPassword;  
}  
  
    public int getAdminId() {  
        return adminId;  
    }  
  
    public void setAdminId(int adminId) {  
        this.adminId = adminId;  
    }  
  
    public String getAdminName() {  
        return adminName;  
    }  
  
    public void setAdminName(String adminName) {  
        this.adminName = adminName;  
    }  
  
    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;
    }

    public String getConfirmPassword() {
        return confirmPassword;
    }

    public void setConfirmPassword(String
confirmPassword) {
        this.confirmPassword = confirmPassword;
    }
}

```

AirlineModel

```
package org.SimplilearnProject.aabp.models;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```

public class AirlineModel {

    private int airlineId;

    private String airlineName;

    private String airlineHeadquarters;

    private String websites;

    private String code;

    private String country;

    private Set<FlightModel> flights = new HashSet<>();

    public AirlineModel() {

    }
}

```



```
public int getAirlineId() {  
    return airlineId;  
}
```

```
public AirlineModel(int airlineId, String airlineName, String airlineHeadquarters, String websites, String  
code,  
                    String country, Set<FlightModel> flights) {  
    super();  
    this.airlineId = airlineId;  
    this.airlineName = airlineName;  
    this.airlineHeadquarters = airlineHeadquarters;  
    this.websites = websites;  
    this.code = code;  
    this.country = country;  
    this.flights = flights;  
}
```

```
public void setAirlineId(int airlineId) {  
    this.airlineId = airlineId;  
}
```

```
public String getAirlineName() {  
    return airlineName;  
}
```

```
public void setAirlineName(String airlineName) {  
    this.airlineName = airlineName;  
}
```

```
public String getAirlineHeadquarters() {  
    return airlineHeadquarters;  
}
```

```
public void setAirlineHeadquarters(String airlineHeadquarters) {  
    this.airlineHeadquarters = airlineHeadquarters;  
}
```

```
public String getWebsites() {  
    return websites;  
}
```

```
public void setWebsites(String websites) {  
    this.websites = websites;  
}
```

```
public String getCode() {  
    return code;  
}
```

```
public void setCode(String code) {  
    this.code = code;  
}
```

```
public String getCountry() {  
    return country;  
}
```

```
public void setCountry(String country) {  
    this.country = country;  
}  
  
public Set<FlightModel> getFlights() {  
    return flights;  
}  
  
public void setFlights(Set<FlightModel> flights) {  
    this.flights = flights;  
}  
}
```

BookingModel

```
package org.SimplilearnProject.aabp.models;  
  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
  
public class BookingModel {  
    private int bookingId;  
    private Date bookingDate;  
    private int passengerCount;  
    private int totalPrice;  
    private UserModel user;  
    private AirlineModel airline;  
    private Set<FlightModel> flights = new HashSet<>();  
}
```

```
private Set<PaymentModel> payments = new HashSet<>();
```

```
public int getBookingId() {  
    return bookingId;  
}
```

```
public void setBookingId(int bookingId) {  
    this.bookingId = bookingId;  
}
```

```
public Date getBookingDate() {  
    return bookingDate;  
}
```

```
public void setBookingDate(Date bookingDate) {  
    this.bookingDate = bookingDate;  
}
```

```
public int getPassengerCount() {  
    return passengerCount;  
}
```

```
public void setPassengerCount(int passengerCount) {  
    this.passengerCount = passengerCount;  
}
```

```
public int getTotalPrice() {  
    return totalPrice;  
}
```

```
public void setTotalPrice(int totalPrice) {  
    this.totalPrice = totalPrice;  
}
```

```
public UserModel getUser() {  
    return user;  
}
```

```
public void setUser(UserModel user) {  
    this.user = user;  
}
```

```
public AirlineModel getAirline() {  
    return airline;  
}
```

```
public void setAirline(AirlineModel airline) {  
    this.airline = airline;  
}
```

```
public Set<FlightModel> getFlights() {  
    return flights;  
}
```

```
public void setFlights(Set<FlightModel> flights) {  
    this.flights = flights;  
}
```

```

    public Set<PaymentModel> getPayments() {
        return payments;
    }

    public void setPayments(Set<PaymentModel> payments) {
        this.payments = payments;
    }
}

```

```

FlightModel
package org.SimplilearnProject.aabp.models;

public class FlightModel {
    private int flightId;
    private String flightNumber;
    private String source;
    private String destination;
    private AirlineModel airline; // Assuming there is
an AirlineModel associated with the flight

    public FlightModel() {

    }

    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 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 AirlineModel getAirline() {
        return airline;
    }

    public void setAirline(AirlineModel airline) {
        this.airline = airline;
    }
}

```

LoginModel

```
package org.SimplilearnProject.aabp.models;
```

```

public class LoginModel {
    private String username;
    private String password;

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {

```

```
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

PaymentModel

```
package org.SimplilearnProject.aabp.models;

import java.util.Date;

public class PaymentModel {
    private int paymentId;
    private int paymentAmount;
    private Date paymentDate;
    private String paymentMethod;
    private BookingModel booking;

    public int getPaymentId() {
        return paymentId;
    }

    public void setPaymentId(int paymentId) {
        this.paymentId = paymentId;
    }

    public int getPaymentAmount() {
        return paymentAmount;
    }

    public void setPaymentAmount(int paymentAmount) {
        this.paymentAmount = paymentAmount;
    }
}
```



```

    public Date getPaymentDate() {
        return paymentDate;
    }

    public void setPaymentDate(Date paymentDate) {
        this.paymentDate = paymentDate;
    }

    public String getPaymentMethod() {
        return paymentMethod;
    }

    public void setPaymentMethod(String paymentMethod)
{
        this.paymentMethod = paymentMethod;
    }

    public BookingModel getBooking() {
        return booking;
    }

    public void setBooking(BookingModel booking) {
        this.booking = booking;
    }
}

```

PlaceModel

```

package org.SimplilearnProject.aabp.models;

```

```

public class PlaceModel {
    private int pinCode; // Renamed placeId to pinCode
    private String placeName;
    private String country;

    public int getPinCode() {
        return pinCode;
    }

    public void setPinCode(int pinCode) {
        this.pinCode = pinCode;
    }
}

```

```

    }

    public String getPlaceName() {
        return placeName;
    }

    public void setPlaceName(String placeName) {
        this.placeName = placeName;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
        this.country = country;
    }
}

```

UserModel

```
package org.SimplilearnProject.aabp.models;
```

```

public class UserModel {
    private int userId;
    private String userName;
    private String email;
    private String password;
    private String confirmPassword;

    public UserModel() {
    }

    public UserModel(int userId, String userName,
String email, String password, String confirmPassword)
{
    this.userId = userId;
    this.userName = userName;
    this.email = email;
    this.password = password;
    this.confirmPassword = confirmPassword;
}

```

```
}

public int getUserId() {
    return userId;
}

public void setUserId(int userId) {
    this.userId = userId;
}

public String getUsername() {
    return userName;
}

public void setUsername(String userName) {
    this.userName = userName;
}

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;
}

public String getConfirmPassword() {
    return confirmPassword;
}

public void setConfirmPassword(String
confirmPassword) {
```

```
        this.confirmPassword = confirmPassword;
    }
}
```

AdminServices

```
package org.SimplilearnProject.aabp.services;
```

```
import org.SimplilearnProject.aabp.entities.Admin;
import org.SimplilearnProject.aabp.models.AdminLoginModel;
import org.SimplilearnProject.aabp.models.AdminModel;
import org.SimplilearnProject.aabp.models.LoginModel;
```

```
public interface AdminServices {
    void registerAdmin(AdminModel adminModel);
    Admin getAdminByUsername(String username);

    Admin getAdmin(AdminLoginModel loginModel);
}
```

AirlineServices

```
package org.SimplilearnProject.aabp.services;
```

```
import org.SimplilearnProject.aabp.entities.Airline;
import org.SimplilearnProject.aabp.models.AirlineModel;
```

```
public interface AirlineServices {
    Airline getAirlineById(int airlineId);
}
```

```
void createAirline(AirlineModel airlineModel);  
void updateAirline(AirlineModel airlineModel);  
void deleteAirline(int airlineId);  
}
```

BookingServices

```
package org.SimplilearnProject.aabp.services;  
  
import org.SimplilearnProject.aabp.models.BookingModel;  
  
public interface BookingServices {  
    void createBooking(BookingModel bookingModel);  
}
```

FlightServices

```
package org.SimplilearnProject.aabp.services;  
  
import org.SimplilearnProject.aabp.dao.FlightDao;  
import org.SimplilearnProject.aabp.entities.Flight;  
import org.SimplilearnProject.aabp.models.FlightModel;  
import java.util.List;  
  
public interface FlightServices {  
    void addFlight(FlightModel flightModel);  
    void setFlightDao(FlightDao flightDao);  
    FlightModel getFlightById(int flightId);  
    void updateFlight(FlightModel flightModel);  
    void deleteFlight(int flightId);  
    List<FlightModel> getAllFlights(); // Add getAllFlights method  
}
```

PaymentServices

```
package org.SimplilearnProject.aabp.services;

import org.SimplilearnProject.aabp.models.PaymentModel;

public interface PaymentServices {
    void processPayment(PaymentModel paymentModel);
}
```

```
PlaceServices
package org.SimplilearnProject.aabp.services;

import org.SimplilearnProject.aabp.models.PlaceModel;

public interface PlaceServices {
    void createPlace(PlaceModel placeModel);
}
```

```
UserService
package org.SimplilearnProject.aabp.services;

import org.SimplilearnProject.aabp.entities.User;
import org.SimplilearnProject.aabp.models.LoginModel;
import org.SimplilearnProject.aabp.models.UserModel;

public interface UserService {

    void registerUser(UserModel userModel);

    User getUserByUsername(String username);

    User getUser(LoginModel loginModel);

}
```

```
AdminServicesImpl
package org.SimplilearnProject.aabp.servicesImpl;
```

```
import org.SimplilearnProject.aabp.config.HibConfig;
import org.SimplilearnProject.aabp.dao.AdminDao;
import org.SimplilearnProject.aabp.daoImpl.AdminDaoImpl;
import org.SimplilearnProject.aabp.entities.Admin;
import org.SimplilearnProject.aabp.models.AdminLoginModel;
import org.SimplilearnProject.aabp.models.AdminModel;
import org.SimplilearnProject.aabp.models.LoginModel;
import org.SimplilearnProject.aabp.services.AdminServices;
```

```
public class AdminServicesImpl implements AdminServices {
    private AdminDao adminDao = new AdminDaoImpl();
```

```
    @Override
```

```
    public void registerAdmin(AdminModel adminModel) {
        Admin admin = convertToAdminEntity(adminModel);
        adminDao.setSessionFactory(HibConfig.getSessionFactory());
        adminDao.saveAdmin(admin);
    }
```

```
    @Override
```

```
    public Admin getAdmin(AdminLoginModel loginModel) {
        String adminName = loginModel.getAdminName();
        String password = loginModel.getPassword();
        return adminDao.getAdmin(adminName, password);
    }
```

```
    @Override
```

```
public Admin getAdminByadminName(String username) {  
    return adminDao.getAdminByUsername(username);  
}  
  
private Admin convertToAdminEntity(AdminModel adminModel) {  
    Admin admin = new Admin();  
    admin.setAdminName(adminModel.getAdminName());  
    admin.setEmail(adminModel.getEmail());  
    admin.setPassword(adminModel.getPassword());  
    return admin;  
}  
}
```

AirlineServicesImpl

```
package org.SimplilearnProject.aabp.servicesImpl;  
  
import org.SimplilearnProject.aabp.dao.AirlineDao;  
import org.SimplilearnProject.aabp.entities.Airline;  
import org.SimplilearnProject.aabp.models.AirlineModel;  
import org.SimplilearnProject.aabp.services.AirlineServices;  
  
public class AirlineServicesImpl implements AirlineServices {  
    private AirlineDao airlineDao;  
  
    public AirlineServicesImpl(AirlineDao airlineDao) {  
        this.airlineDao = airlineDao;  
    }  
}
```


@Override

```
public Airline getAirlineById(int airlineId) {  
    return airlineDao.getAirlineById(airlineId);  
}
```

@Override

```
public void createAirline(AirlineModel airlineModel) {  
    Airline airline = convertToAirlineEntity(airlineModel);  
    airlineDao.saveAirline(airline);  
}
```

@Override

```
public void updateAirline(AirlineModel airlineModel) {  
    Airline airline = airlineDao.getAirlineById(airlineModel.getAirlineId());  
    if (airline != null) {  
        // Update the properties of the airline entity based on the airlineModel  
        airline.setAirlineName(airlineModel.getAirlineName());  
        airline.setAirlineHeadquarters(airlineModel.getAirlineHeadquarters());  
        airline.setWebsites(airlineModel.getWebsites());  
        airline.setCode(airlineModel.getCode());  
        airline.setCountry(airlineModel.getCountry());  
        // Update any other properties as needed  
        // ...  
  
        airlineDao.updateAirline(airline);  
    }  
}
```

@Override

```

public void deleteAirline(int airlineId) {
    Airline airline = airlineDao.getAirlineById(airlineId);
    if (airline != null) {
        airlineDao.deleteAirline(airline);
    }
}

```

```

private Airline convertToAirlineEntity(AirlineModel airlineModel) {
    Airline airline = new Airline();
    airline.setAirlineName(airlineModel.getAirlineName());
    airline.setAirlineHeadquarters(airlineModel.getAirlineHeadquarters());
    airline.setWebsites(airlineModel.getWebsites());
    airline.setCode(airlineModel.getCode());
    airline.setCountry(airlineModel.getCountry());
    // Set any other properties as needed
    // ...

    return airline;
}
}

```

```

BookingServicesImpl
package org.SimplilearnProject.aabp.servicesImpl;

```

```

import java.util.HashSet;
import java.util.Set;

import org.SimplilearnProject.aabp.dao.BookingDao;
import org.SimplilearnProject.aabp.entities.Booking;

```

```
import org.SimplilearnProject.aabp.entities.Flight;
import org.SimplilearnProject.aabp.entities.Payment;
import org.SimplilearnProject.aabp.entities.User;
import org.SimplilearnProject.aabp.models.BookingModel;
import org.SimplilearnProject.aabp.models.FlightModel;
import org.SimplilearnProject.aabp.models.PaymentModel;
import org.SimplilearnProject.aabp.services.BookingServices;

public class BookingServicesImpl implements BookingServices {

    private BookingDao bookingDao;

    public BookingServicesImpl(BookingDao bookingDao) {
        this.bookingDao = bookingDao;
    }

    @Override
    public void createBooking(BookingModel bookingModel) {

        Booking booking = new Booking();

        // Set the booking properties based on the bookingModel
        booking.setBookingDate(bookingModel.getBookingDate());
        booking.setPassengerCount(bookingModel.getPassengerCount());
        booking.setTotalPrice(bookingModel.getTotalPrice());

        // Set the user
        User user = new User();
        user.setUserName(bookingModel.getUser().getUserName());
        user.setEmail(bookingModel.getUser().getEmail());

        // Set other user properties if needed
```

```
        booking.setUser(user);

        // Set the flights
        Set<Flight> flights = new HashSet<>();
        for (FlightModel flightModel : bookingModel.getFlights()) {
            Flight flight = new Flight();
            flight.setFlightNumber(flightModel.getFlightNumber());
            // Set other flight properties if needed
            flights.add(flight);
        }
        booking.setFlights(flights);

        // Set the payments
        Set<Payment> payments = new HashSet<>();
        for (PaymentModel paymentModel : bookingModel.getPayments()) {
            Payment payment = new Payment();
            payment.setPaymentMethod(paymentModel.getPaymentMethod());
            // Set other payment properties if needed
            payments.add(payment);
        }
        booking.setPayments(payments);

        bookingDao.saveBooking(booking);
    }
}
```

```
FlightServicesImpl
package org.SimplilearnProject.aabp.servicesImpl;
```

```
import org.SimplilearnProject.aabp.dao.AirlineDao;
import org.SimplilearnProject.aabp.dao.FlightDao;
import org.SimplilearnProject.aabp.entities.Airline;
import org.SimplilearnProject.aabp.entities.Flight;
import org.SimplilearnProject.aabp.models.AirlineModel;
import org.SimplilearnProject.aabp.models.FlightModel;
import org.SimplilearnProject.aabp.services.FlightServices;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
public class FlightServicesImpl implements FlightServices {
    private FlightDao flightDao;
```

```
    public FlightServicesImpl(FlightDao flightDao) {
        this.flightDao = flightDao;
    }
```

```
@Override
```

```
public void addFlight(FlightModel flightModel) {
    Flight flight = convertToFlightEntity(flightModel);
    flightDao.saveFlight(flight);
}
```

```
@Override
```

```
public FlightModel getFlightById(int flightId) {
    Flight flight = flightDao.getFlightById(flightId);
    return convertToFlightModel(flight);
}
```

@Override

```
public void updateFlight(FlightModel flightModel) {  
    Flight flight = convertToFlightEntity(flightModel);  
    flightDao.updateFlight(flight);  
}
```

@Override

```
public void deleteFlight(int flightId) {  
    flightDao.deleteFlight(flightId);  
}
```

@Override

```
public List<FlightModel> getAllFlights() {  
    List<Flight> flights = flightDao.getAllFlights();  
    return convertToFlightModelList(flights);  
}
```

```
private Flight convertToFlightEntity(FlightModel flightModel) {  
    Flight flight = new Flight();  
    flight.setFlightId(flightModel.getFlightId());  
    flight.setFlightNumber(flightModel.getFlightNumber());  
    flight.setFlightSource(flightModel.getSource());  
    flight.setFlightDestination(flightModel.getDestination());  
    // Set other properties of the Flight entity as needed  
    return flight;  
}
```

```
private FlightModel convertToFlightModel(Flight flight) {
```

```

        FlightModel flightModel = new FlightModel();
        flightModel.setFlightId(flight.getFlightId());
        flightModel.setFlightNumber(flight.getFlightNumber());
        flightModel.setSource(flight.getFlightSource());
        flightModel.setDestination(flight.getFlightDestination());
        // Set other properties of the FlightModel as needed
        return flightModel;
    }

    private List<FlightModel> convertToFlightModelList(List<Flight> flights) {
        List<FlightModel> flightModels = new ArrayList<>();

        for (Flight flight : flights) {
            FlightModel flightModel = convertToFlightModel(flight);
            flightModels.add(flightModel);
        }

        return flightModels;
    }

```

```

@Override
public void setFlightDao(FlightDao flightDao) {
    this.flightDao = flightDao;
}
}

```

```

PaymentServicesImpl
package org.SimplilearnProject.aabp.servicesImpl;

```

```

import org.SimplilearnProject.aabp.dao.PaymentDao;
import org.SimplilearnProject.aabp.entities.Payment;
import org.SimplilearnProject.aabp.models.PaymentModel;
import org.SimplilearnProject.aabp.services.PaymentServices;

public class PaymentServicesImpl implements PaymentServices {
    private PaymentDao paymentDao;

    public PaymentServicesImpl(PaymentDao paymentDao) {
        this.paymentDao = paymentDao;
    }

    @Override
    public void processPayment(PaymentModel paymentModel) {
        Payment payment = new Payment();
        // Set the payment properties based on the paymentModel
        payment.setPaymentId(paymentModel.getPaymentId());
        payment.setPaymentAmount(paymentModel.getPaymentAmount());
        payment.setPaymentDate(paymentModel.getPaymentDate());
        payment.setPaymentMethod(paymentModel.getPaymentMethod());
        // Set other payment properties if needed

        paymentDao.savePayment(payment);
    }
}

```

PlaceServicesImpl

```
package org.SimplilearnProject.aabp.servicesImpl;
```



```

import org.SimplilearnProject.aabp.dao.PlaceDao;
import org.SimplilearnProject.aabp.entities.Place;
import org.SimplilearnProject.aabp.models.PlaceModel;
import org.SimplilearnProject.aabp.services.PlaceServices;

public class PlaceServicesImpl implements PlaceServices {
    private PlaceDao placeDao;

    public PlaceServicesImpl(PlaceDao placeDao) {
        this.placeDao = placeDao;
    }

    @Override
    public void createPlace(PlaceModel placeModel) {
        Place place = new Place();
        place.setPinCode(placeModel.getPinCode()); // Updated to use pinCode
        place.setPlaceName(placeModel.getPlaceName());
        place.setCountry(placeModel.getCountry());
        // Set other place properties if needed

        placeDao.savePlace(place);
    }

    // Implement other methods of PlaceServices if needed
}

```

UserServiceImpl

```

package org.SimplilearnProject.aabp.servicesImpl;

```

```
import org.SimplilearnProject.aabp.dao.UserDao;
import org.SimplilearnProject.aabp.daoImpl.UserDaoImpl;
import org.SimplilearnProject.aabp.entities.User;
import org.SimplilearnProject.aabp.models.LoginModel;
import org.SimplilearnProject.aabp.models.UserModel;
import org.SimplilearnProject.aabp.services.UserService;
```

```
public class UserServiceImpl implements UserService {
    private UserDao userDao = new UserDaoImpl();
```

```
    @Override
```

```
    public void registerUser(UserModel userModel) {
        User user = convertToUserEntity(userModel);
        userDao.saveUser(user);
    }
```

```
    @Override
```

```
    public User getUser(LoginModel loginModel) {
        String username = loginModel.getUsername();
        String password = loginModel.getPassword();
        return userDao.getUser(username, password);
    }
```

```
    @Override
```

```
    public User getUserByUsername(String username) {
        return userDao.getUserByUsername(username);
    }
```

```

private User convertToUserEntity(UserModel userModel) {

    User user = new User();

    user.setUserName(userModel.getUserName());

    user.setEmail(userModel.getEmail());

    user.setPassword(userModel.getPassword());

    return user;

}
}

```

Respective JSP FILE

```

<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Add Place Success</title>
    <style>
        table {
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid black;
            padding: 8px;
        }
        th {
            background-color: lightgray;
        }
    </style>
</head>
<body>
    <h1>Place Added Successfully!</h1>

```

```

    <p>The place has been successfully added to the
database.</p>
    <p>Place Details:</p>
    <table>
        <tr>
            <th>Pin Code</th>
            <th>Place Name</th>
            <th>Country</th>
        </tr>
        <tr>
            <td>${place.pinCode}</td>
            <td>${place.placeName}</td>
            <td>${place.country}</td>
        </tr>
    </table>
    <br>
    <a href="addPlace.jsp">Add Another Place</a>
    <br>
    <a href="admin-home.jsp">Go to Admin Homepage</a>
</body>
</html>

```

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Admin Home</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }
        h2 {
            color: #333;
        }
        p {

```

```

        margin-bottom: 10px;
    }
    h3 {
        margin-top: 30px;
        margin-bottom: 10px;
    }
    a {
        display: block;
        margin-bottom: 5px;
    }
</style>
</head>
<body>
    <h2>Welcome, ${admin.adminName}!</h2>
    <p>Admin ID: ${admin.adminId}</p>
    <p>Email: ${admin.email}</p>

    <h3>Add Airline</h3>
    <a href="addAirline.jsp">Go to Add Airline Page</a>
    <a href="addflight.jsp">Go to Add Flight Page</a>
    <a href="addPlace.jsp">Go to Add Place Page</a>

    <a href="logout">Logout</a>
</body>
</html>

<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Admin Login</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
            padding: 20px;

```

```

    }
    h2 {
        color: #333;
        margin-bottom: 20px;
    }
    form {
        background-color: #fff;
        padding: 20px;
        border-radius: 5px;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }
    label {
        display: block;
        margin-bottom: 10px;
    }
    input[type="text"],
    input[type="password"] {
        width: 250px;
        padding: 5px;
        border: 1px solid #ccc;
        border-radius: 3px;
    }
    input[type="submit"] {
        padding: 10px 20px;
        background-color: #4CAF50;
        color: #fff;
        border: none;
        border-radius: 3px;
        cursor: pointer;
    }
    .error-message {
        color: red;
        margin-top: 10px;
    }
}
</style>
</head>
<body>
    <h2>Admin Login</h2>

    <c:if test="${not empty error}">

```

```

        <p class="error-message">${error}</p>
    </c:if>

    <form action="admin-login" method="post">
        <label for="adminName">Admin Name:</label>
        <input type="text" id="adminName"
name="adminName" required><br><br>
        <label for="password">Password:</label>
        <input type="password" id="password"
name="password" required><br><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>

<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Admin Registration</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
            padding: 20px;
        }
        h1 {
            color: #333;
            margin-bottom: 20px;
        }
        form {
            background-color: #fff;
            padding: 20px;
            border-radius: 5px;
            box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
        }
        label {
            display: block;

```

```

        margin-bottom: 10px;
    }
    input[type="text"],
    input[type="email"],
    input[type="password"] {
        width: 250px;
        padding: 5px;
        border: 1px solid #ccc;
        border-radius: 3px;
    }
    button[type="submit"] {
        padding: 10px 20px;
        background-color: #4CAF50;
        color: #fff;
        border: none;
        border-radius: 3px;
        cursor: pointer;
    }
    p {
        margin-top: 20px;
    }
    footer {
        margin-top: 40px;
        color: #888;
        text-align: center;
    }
</style>
</head>
<body>
    <h1>Admin Registration</h1>

    <form method="POST" action="AdminRegister">
        <label for="adminName">Admin Name:</label>
        <input type="text" id="adminName"
name="adminName" required><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email"
required><br>

```



```

        <label for="password">Password:</label>
        <input type="password" id="password"
name="password" required><br>

        <button type="submit">Register</button>
    </form>

    <p>Already have an account? <a href="admin-
login.jsp">Login here</a></p>

    <footer>
        &copy; 2023 My Web Application. All rights
reserved.
    </footer>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
    <title>Admin Registration Success</title>
    <style>
        body {
            background-color: #f7f7f7;
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .container {
            max-width: 600px;
            margin: 50px auto;
            background-color: #ffffff;
            border-radius: 5px;
            box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
            padding: 20px;
        }

        h1 {
            color: #333333;

```

```

        text-align: center;
    }

    p {
        color: #666666;
        text-align: center;
        margin-top: 20px;
    }

    .back-link {
        text-align: center;
        margin-top: 20px;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Admin Registration Successful</h1>
        <p>Thank you for registering as an admin.</p>
        <div class="back-link">
            <a href="index.jsp">Go back to the Home
Page</a>
        </div>
    </div>
</body>
</html>

<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="java.util.List" %>
<%@ page
import="org.SimplilearnProject.aabp.models.FlightModel"
%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>View Flights</title>
</head>

```

```

<body>
    <h1>View Flights</h1>

    <table>
        <tr>
            <th>Flight Number</th>
            <th>Source</th>
            <th>Destination</th>
            <th>Airline</th>
        </tr>
        <!-- Retrieve the flights attribute and cast it
to List<FlightModel> -->
        <% List<FlightModel> flights =
(List<FlightModel>) request.getAttribute("flights");
        if (flights != null) {
            for (FlightModel flight : flights) { %>
                <tr>
                    <td><%= flight.getFlightNumber()
%></td>
                    <td><%= flight.getSource() %></td>
                    <td><%= flight.getDestination()
%></td>
                    <td><%=
flight.getAirline().getAirlineName() %></td>
                </tr>
            <% }
        } %>
    </table>

    <a href="index.jsp">Go Back</a>
</body>
</html>

<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">

```

```
<title>Welcome to the Home Page</title>
<style>
  body {
    background-color: #f7f7f7;
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
  }

  .container {
    max-width: 600px;
    margin: 50px auto;
    background-color: #ffffff;
    border-radius: 5px;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
    padding: 20px;
  }

  h2 {
    color: #333333;
    text-align: center;
  }

  p {
    color: #666666;
    text-align: center;
    margin-top: 20px;
  }

  h3 {
    color: #333333;
    margin-top: 40px;
  }

  .button-container {
    display: flex;
    justify-content: center;
    margin-top: 20px;
  }
```

```

.button-container a {
    display: inline-block;
    background-color: #4caf50;
    color: #ffffff;
    padding: 10px 20px;
    border-radius: 5px;
    text-decoration: none;
    transition: background-color 0.3s ease;
}

.button-container a:hover {
    background-color: #45a049;
}

.logout-container {
    display: flex;
    justify-content: center;
    margin-top: 40px;
}

.logout-container input[type="submit"] {
    background-color: #f44336;
    color: #ffffff;
    padding: 10px 20px;
    border-radius: 5px;
    border: none;
    text-decoration: none;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.logout-container input[type="submit"]:hover {
    background-color: #d32f2f;
}
</style>
</head>
<body>
    <div class="container">
        <h2>Welcome to the Home Page</h2>

```

```

        <p>Hello, <strong>${user.userName}</strong>!
You have successfully logged in.</p>
        <h3>Add Booking</h3>
        <div class="button-container">
            <a href="addBooking.jsp">Go to Add Booking
Page</a>
        </div>
        <div class="logout-container">
            <form action="logout" method="post">
                <input type="submit" value="Logout">
            </form>
        </div>
    </div>
</body>
</html>

```

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Flyaway - An Air Booking Portal</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .container {
            max-width: 800px;
            margin: 0 auto;
            padding: 20px;
        }

        h1 {
            color: #333333;
            text-align: center;

```

```
        margin-top: 40px;
    }

    nav {
        background-color: #f7f7f7;
        padding: 10px;
        margin-top: 40px;
    }

    ul {
        list-style: none;
        padding: 0;
        margin: 0;
        display: flex;
        justify-content: center;
    }

    li {
        margin: 0 10px;
    }

    li a {
        color: #333333;
        text-decoration: none;
        padding: 10px;
        transition: color 0.3s ease;
    }

    li a:hover {
        color: #4caf50;
    }

    footer {
        background-color: #f7f7f7;
        padding: 10px;
        text-align: center;
        margin-top: 40px;
        color: #666666;
    }
</style>
```

```
</head>
<body>
    <div class="container">
        <h1>Flyaway - An Air Booking Portal</h1>

        <nav>
            <ul>
                <li><a href="flights.jsp">View
Flights</a></li>
                <li><a href="bookings.jsp">View
Bookings</a></li>
                <li><a href="admin-login">Admin
Login</a></li>
                <li><a href="register.jsp">User
Register</a></li>
                <li><a href="user-login.jsp">User
Login</a></li>
                <li><a href="AdminRegister.jsp">Admin
Register</a></li>
            </ul>
        </nav>

        <footer>
            &copy; 2023 Flyaway - An Air Booking
Portal. All rights reserved. | Developed by Om Shankar
Mishra
        </footer>
    </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <style>
        body {
            background-color: #f7f7f7;
            font-family: Arial, sans-serif;
```



```
    margin: 0;
    padding: 0;
}

.container {
    max-width: 400px;
    margin: 50px auto;
    background-color: #ffffff;
    border-radius: 5px;
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
    padding: 20px;
}

h1 {
    color: #333333;
    text-align: center;
}

p {
    color: #ff5252;
    text-align: center;
    margin-top: 20px;
}

form {
    margin-top: 20px;
}

label {
    display: block;
    margin-bottom: 10px;
    color: #333333;
}

input[type="text"],
input[type="password"] {
    width: 100%;
    padding: 10px;
    border: 1px solid #cccccc;
    border-radius: 3px;
}
```

```

        margin-bottom: 10px;
        font-size: 14px;
    }

    input[type="submit"] {
        display: block;
        width: 100%;
        background-color: #4caf50;
        color: #ffffff;
        padding: 10px;
        border: none;
        border-radius: 3px;
        font-size: 16px;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Login</h1>

        <!-- Display error message if present -->
        <% String error = (String)
request.getAttribute("error"); %>
        <% if (error != null && !error.isEmpty()) { %>
            <p><%= error %></p>
        <% } %>

        <form action="login" method="post">
            <label for="username">Username:</label>
            <input type="text" id="username"
name="username" required><br>

            <label for="password">Password:</label>

```

```
        <input type="password" id="password"
name="password" required><br>
```

```
        <input type="submit" value="Login">
    </form>
</div>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>User Registration</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .container {
            max-width: 400px;
            margin: 50px auto;
            background-color: #ffffff;
            border-radius: 5px;
            box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
            padding: 20px;
        }

        h1 {
            color: #333333;
            text-align: center;
        }

        form {
            margin-top: 20px;
```

```
}
```

```
label {  
    display: block;  
    margin-bottom: 10px;  
    color: #333333;  
}
```

```
input[type="text"],  
input[type="email"],  
input[type="password"] {  
    width: 100%;  
    padding: 10px;  
    border: 1px solid #cccccc;  
    border-radius: 3px;  
    margin-bottom: 10px;  
    font-size: 14px;  
}
```

```
button[type="submit"] {  
    display: block;  
    width: 100%;  
    background-color: #4caf50;  
    color: #ffffff;  
    padding: 10px;  
    border: none;  
    border-radius: 3px;  
    font-size: 16px;  
    cursor: pointer;  
    transition: background-color 0.3s ease;  
}
```

```
button[type="submit"]:hover {  
    background-color: #45a049;  
}
```

```
p {  
    text-align: center;  
    margin-top: 20px;  
}
```

```

        footer {
            background-color: #f7f7f7;
            padding: 10px;
            text-align: center;
            margin-top: 40px;
            color: #666666;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>User Registration</h1>

        <form method="POST" action="register">
            <label for="username">Username:</label>
            <input type="text" id="username"
name="username" required><br>

            <label for="email">Email:</label>
            <input type="email" id="email" name="email"
required><br>

            <label for="password">Password:</label>
            <input type="password" id="password"
name="password" required><br>

            <label for="confirmPassword">Confirm
Password:</label>
            <input type="password" id="confirmPassword"
name="confirmPassword" required><br>

            <button type="submit">Register</button>
        </form>

        <p>Already have an account? <a
href="login.jsp">Login here</a></p>
    </div>

    <footer>

```

© 2023 My Web Application. All rights reserved. | Developer: [Om shankar mishra]

```
</footer>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Registration Success</title>
  <style>
    body {
      background-color: #f7f7f7;
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    .container {
      max-width: 600px;
      margin: 50px auto;
      background-color: #ffffff;
      border-radius: 5px;
      box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
      padding: 20px;
    }

    h1 {
      color: #333333;
      text-align: center;
    }

    p {
      color: #666666;
      text-align: center;
      margin-top: 20px;
    }
  </style>
</head>
<body>
```

```

        /* Add any additional CSS styling here */
    </style>
</head>
<body>
    <div class="container">
        <h1>Registration Successful</h1>
        <p>Congratulations! Your registration was
successful.</p>
        <!-- Add any additional information or success
message here -->

        <!-- Add your HTML content or UI elements here
-->

        <!-- Add your JavaScript or other client-side
scripts here -->
    </div>
</body>
</html>

```

```

<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>User Login</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .container {
            max-width: 400px;
            margin: 50px auto;
            background-color: #ffffff;

```

```
        border-radius: 5px;
        box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
        padding: 20px;
    }

    h2 {
        color: #333333;
        text-align: center;
    }

    form {
        margin-top: 20px;
    }

    label {
        display: block;
        margin-bottom: 10px;
        color: #333333;
    }

    input[type="text"],
    input[type="password"] {
        width: 100%;
        padding: 10px;
        border: 1px solid #cccccc;
        border-radius: 3px;
        margin-bottom: 10px;
        font-size: 14px;
    }

    input[type="submit"] {
        display: block;
        width: 100%;
        background-color: #4caf50;
        color: #ffffff;
        padding: 10px;
        border: none;
        border-radius: 3px;
        font-size: 16px;
        cursor: pointer;
    }
```



```

        transition: background-color 0.3s ease;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <div class="container">
        <h2>User Login</h2>

        <!-- Display error message if login fails -->

        <form action="login" method="post">
            <label for="username">Username:</label>
            <input type="text" id="username"
name="username" required><br><br>

            <label for="password">Password:</label>
            <input type="password" id="password"
name="password" required><br><br>

            <input type="submit" value="Login">
        </form>
    </div>
</body>
</html>

<%@ page language="java" contentType="text/html;
charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Add Airline</title>
</head>
<body>
    <h2>Add Airline</h2>
    <form action="addAirline" method="post">

```

```
<label for="airlineName">Airline Name:</label>
<input type="text" id="airlineName"
name="airlineName" required><br><br>

<label for="airlineHeadquarters">Airline
Headquarters:</label>
<input type="text" id="airlineHeadquarters"
name="airlineHeadquarters" required><br><br>

<label for="websites">Websites:</label>
<input type="text" id="websites"
name="websites" required><br><br>

<label for="code">Code:</label>
<input type="text" id="code" name="code"
required><br><br>

<label for="country">Country:</label>
<input type="text" id="country" name="country"
required><br><br>

<input type="submit" value="Add Airline">
</form>

<div style="color: red;">
  <!-- Display error message if any -->
  <%= request.getAttribute("errorMessage") %>
</div>

<div style="color: green;">
  <!-- Display success message if any -->
  <%= request.getAttribute("successMessage") %>
</div>
</body>
</html>
```