

FLYAWAY – AN AIRLINE BOOKING PORTAL

This document contains the following

- Project and developer details
- Sprint planning and tasks achieved
- Core concepts used in the project
- Flowchart of the application
- Links to the GitHub repository
- Demonstration of product capabilities, appearance, and user interactions
- Unique Selling Points of the application
- Conclusion

Project and Developer details:

Project objective:

As a Full Stack Developer, design and develop an airline booking portal named as FlyAway. Use the GitHub repository to manage the project artifacts.

Background of the problem statement:

FlyAway is a ticket-booking portal that lets people book flights on their website.

The website needs to have the following features:

- A search form in the homepage to allow entry of travel details, like the date of travel, source, destination, and the number of persons.
- Based on the travel details entered, it will show the available flights with their ticket prices.
- Once a person selects a flight to book, they will be taken to a register page where they must fill in their personal details. In the next page, they are shown the flight details of the flight that they are booking, and the payment is done via a dummy payment gateway. On completion of the payment, they are shown a confirmation page with the details of the booking.

For the above features to work, there will be an admin backend with the following features:

- An admin login page where the admin can change the password after login, if he wishes
- A master list of places for source and destination
- A master list of airlines
- A list of flights where each flight has a source, destination, airline, and ticket price

The goal of the company is to deliver a high-end quality product as early as possible.

The flow and features of the application:

- Plan more than two sprints to complete the application
- Document the flow of the application and prepare a flow chart
- List the core concepts and algorithms being used to complete this application
- Implement the appropriate concepts such as exceptions, collections, and sorting techniques for source code optimization and increased performance

You must use the following:

- Eclipse/IntelliJ: An IDE to code for the application
- Java: A programming language to develop web pages, databases, and others
- SQL: To create tables for admin, classes, students, and other specifics
- Maven: To create a web-enabled Maven project
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the application code and track its versions
- Scrum: An efficient agile framework to deliver the product incrementally
- Search and Sort techniques: Data structures used for the project
- Specification document: Any open-source document or Google Docs

The following requirements should be met:

- The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in them.
- The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link to the repository. You can add a section to your document.
- Document the step-by-step process starting from sprint planning to the product release.
- Application should not close, exit, or throw an exception if the user specifies an invalid input.
- You need to submit the final specification document which includes:
 - Project and developer details
 - Sprints planned and the tasks achieved in them
 - Flowchart of the application
 - Core concepts used in the project
 - Links to the GitHub repository to verify the project completion

Developer Details:

Tejeswi Devi Priya Pillarisetty

priyapillarisetty19@gmail.com

Spring planning and Task completion:

This project is intended to be delivered in three sprints.

Sprint 1: Analysed the application's features and prepared a flow chart and Git Repository. Implement the database schema for admin, airline, flight, place, user, booking.

Sprint 2: Create basic CRUD operations for admins, users, airlines, flights, booking, places. Implement the Search Flight feature to display information about flights available based on a specific criteria selected by user.

Sprint 3: Tested the application numerous times to ensure a high-end quality product and pushed it to GitHub. Prepared this document highlighting the application's capabilities, appearance and user interactions.

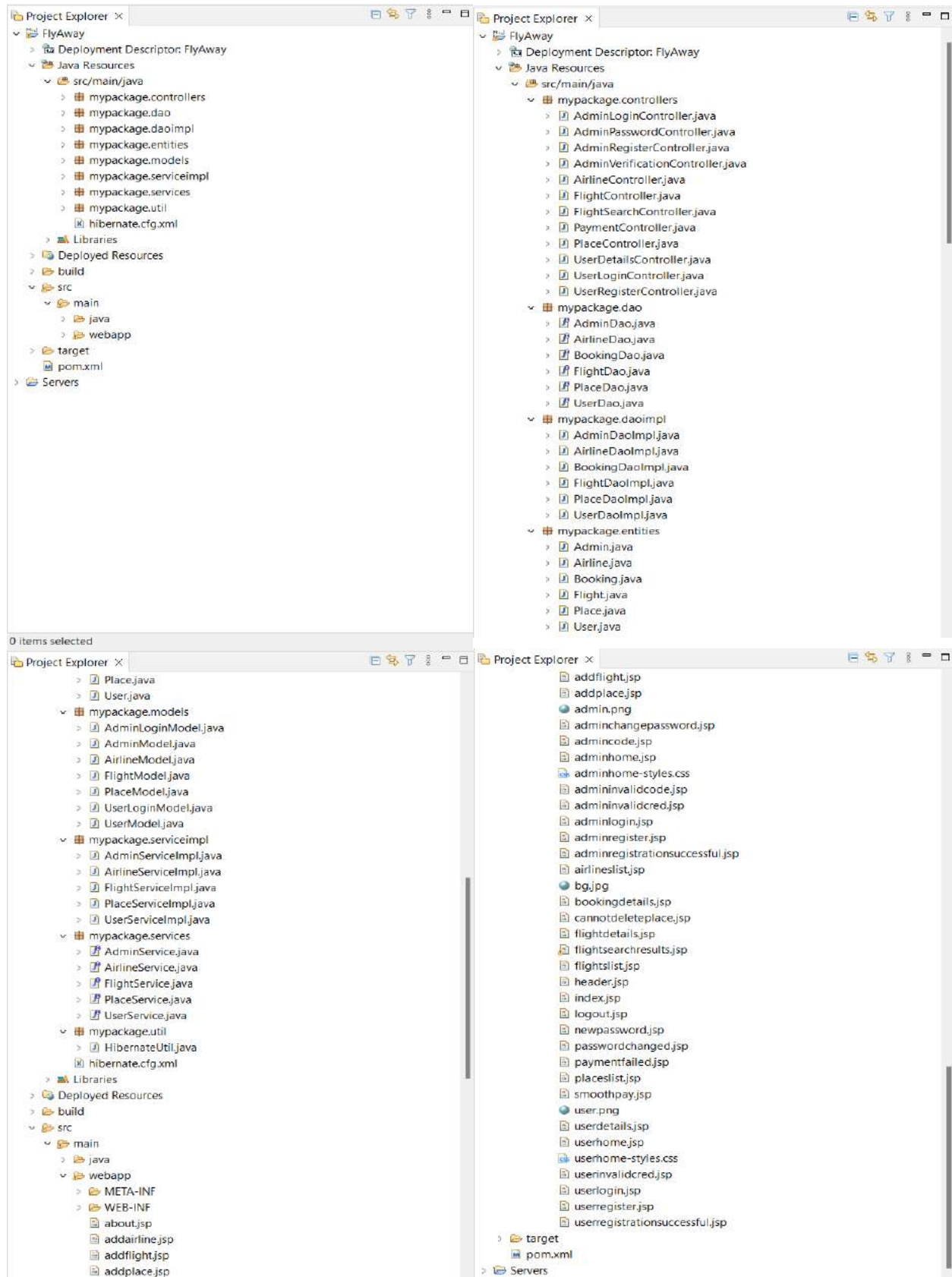
Core concepts used in the project:

- CRUD Operations
- Exception Handling
- Authentication and Authorization
- Session Management
- Frontend technologies like HTML, CSS
- Java Server Pages (JSP)

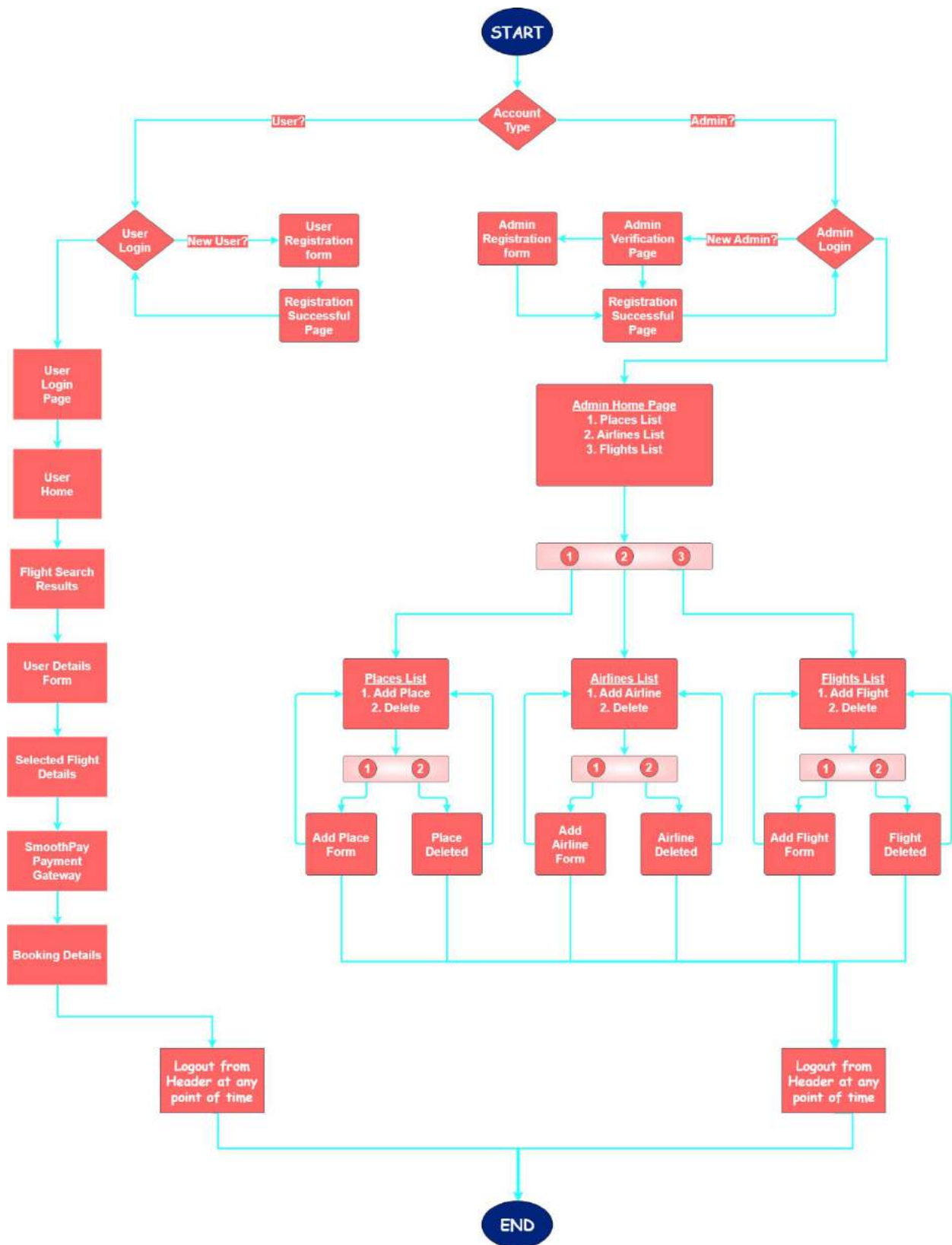
Links to the GitHub repository:

<https://github.com/Priya-2705/Fly-Away.git>

Project Hierarchy



Flowchart of the application:



Demonstration of product capabilities, appearance, and user interactions:

The following sections have been organized to highlight the project's appearance and user interactions in order to demonstrate the capabilities of the product:

1. Creating a Dynamic Web Project in Eclipse
2. Setting up HibernateUtil.java & hibernate.cfg.xml
3. Creating Entity classes
4. Creating DAO interfaces
5. Creating DAOImpl classes
6. Creating .jsp files
7. Creating Controller java servlets
8. Creating Model classes
9. Creating Service interfaces
10. Creating ServiceImpl classes
11. Pushing the code to GitHub Repository.

1. Creating a Dynamic Web Project in Eclipse

- Open Eclipse IDE.
- Go to File → New → Others → Dynamic Web Project.
- Give the project name 'FlyAway', check if the dynamic web module version is above 3.0, and click Finish.
- Right-click on the 'FlyAway' project in project explorer → Configure → Convert to Maven Project → Give Group Id as mypackage and Artifact Id as 'FlyAway' → Click Finish.
- Add Maven Dependencies in the pom.xml file that appears after converting the project to a Maven Project.

Pom.xml file:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>FlyAway</groupId>
  <artifactId>FlyAway</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <dependencies>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
```

```

        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>
    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>5.6.14.Final</version>
    </dependency>
    <dependency>
        <groupId>jstl</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jsp-api</artifactId>
        <version>2.0</version>
        <scope>provided</scope>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
            <configuration>
                <release>17</release>
            </configuration>
        </plugin>
        <plugin>
            <artifactId>maven-war-plugin</artifactId>
            <version>3.2.3</version>
        </plugin>
    </plugins>
</build>
</project>

```

2. Setting up HibernateUtil.java & hibernate.cfg.xml

- Create a package 'mypackage.util' in Java Resources.
- Create a java class HibernateUtil.java in mypackage.util.
- Create a xml file under src/main/java in Java Resources with the name hibernate.cfg.xml.

HibernateUtil.java

```
package mypackage.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import mypackage.entities.Admin;
import mypackage.entities.Airline;
import mypackage.entities.Booking;
import mypackage.entities.Flight;
import mypackage.entities.Place;
import mypackage.entities.User;

public class HibernateUtil {

    static SessionFactory sessionFactory = null;

    public static SessionFactory buildSessionFactory() {

        if(sessionFactory != null) {
            return sessionFactory;
        }

        // STEP 1: Create Configuration Object
        Configuration cfg = new
Configuration().configure("hibernate.cfg.xml").addAnnotatedClass(Flight.class)

.addAnnotatedClass(Airline.class).addAnnotatedClass(Booking.class).addAnnotatedClass(Place.class).addAnnotatedClass(Admin.class).addAnnotatedClass(User.class);
        sessionFactory = cfg.buildSessionFactory();
        return sessionFactory;
    }
}
```

Hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
```



```

        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/flyaway</property>
        <property name="hibernate.connection.username">root</property>
        <property
name="hibernate.connection.password">Priya@0456</property>

        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.format_sql">true</property>

        <property name="hbm2ddl.auto">update</property>
    </session-factory>
</hibernate-configuration>

```

3. Creating Entity classes

- Create a package 'mypackage.entities' in Java Resources.
- Create java classes Admin.java, Airline.java, Flight.java, Booking.java, Place.java, User.java files in mypackage.entities.
- Define data models, add ORM annotations and map relationships.

Admin.java

```

package mypackage.entities;

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

import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name = "Admin")
public class Admin {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int adminId;
    private String aUsername;
    private String aPassword;

    @OneToOne(mappedBy = "admin", fetch = FetchType.EAGER)
    private Set<Place> places = new HashSet<>();
}

```

```
@OneToMany(mappedBy = "admin", fetch = FetchType.EAGER)
private Set<Airline> airlines = new HashSet<>();
```

```
@OneToMany(mappedBy = "admin", fetch = FetchType.EAGER)
private Set<Flight> flights = new HashSet<>();
```

```
// Helper Methods
```

```
public void addPlace(Place place) {
    this.places.add(place);
}
```

```
public void addAirline(Airline airline) {
    this.airlines.add(airline);
}
```

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

```
// Getters and Setters
```

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

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

```
public String getUsername() {
    return aUsername;
}
```

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

```
public String getPassword() {
    return aPassword;
}
```

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

```
public Set<Place> getPlaces() {
```

```

        return places;
    }

    public void setPlaces(Set<Place> places) {
        this.places = places;
    }

    public Set<Airline> getAirlines() {
        return airlines;
    }

    public void setAirlines(Set<Airline> airlines) {
        this.airlines = airlines;
    }

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

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

```

Airline.java

```

package mypackage.entities;

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

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.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "Airline")
public class Airline {

    @Id

```

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int airlineId;
private String airlineName;

@ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
@JoinColumn(name = "adminId")
private Admin admin;

@OneToMany(mappedBy = "airline", cascade = CascadeType.ALL)
private Set<Flight> flights = new HashSet<>();

// Helper Method
public void addFlight(Flight flight) {
    this.flights.add(flight);
}

// Getters and Setters
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 Admin getAdmin() {
    return admin;
}

public void setAdmin(Admin admin) {
    this.admin = admin;
}

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

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

Booking.java

```
package mypackage.entities;
```

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

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

```
@Entity
```

```
@Table(name = "Booking")
```

```
public class Booking {
```

```
    @Id
```

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

```
    private int bookingId;
```

```
    @ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
```

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

```
    private User user;
```

```
    @ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
```

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

```
    private Flight flight;
```

```
    private LocalDateTime bookingDateTime;
```

```
    private int totalTickets;
```

```
    private double totalPrice;
```

```
    private String paymentStatus;
```

```
    // Getters and Setters
```

```
    public int getBookingId() {
```

```
        return bookingId;
```

```
    }
```

```
    public void setBookingId(int bookingId) {
```

```
        this.bookingId = bookingId;
    }

    public User getUser() {
        return user;
    }

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

    public Flight getFlight() {
        return flight;
    }

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

    public LocalDateTime getBookingDateTime() {
        return bookingDateTime;
    }

    public void setBookingDateTime(LocalDateTime bookingDateTime) {
        this.bookingDateTime = bookingDateTime;
    }

    public int getTotalTickets() {
        return totalTickets;
    }

    public void setTotalTickets(int totalTickets) {
        this.totalTickets = totalTickets;
    }

    public double getTotalPrice() {
        return totalPrice;
    }

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

    public String getPaymentStatus() {
        return paymentStatus;
    }
}
```

```

    }

    public void setPaymentStatus(String paymentStatus) {
        this.paymentStatus = paymentStatus;
    }
}

```

Flight.java

```

package mypackage.entities;

import java.time.LocalDateTime;
import java.util.HashSet;
import java.util.Set;

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.ManyToOne;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name = "Flight")
public class Flight {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int flightId;

    @ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
    @JoinColumn(name = "airlineId")
    private Airline airline;

    @ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
    @JoinColumn(name = "sourcePlaceId")
    private Place sourcePlace;

    @ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
    @JoinColumn(name = "destinationPlaceId")
    private Place destinationPlace;

    private LocalDateTime departureDateTime;
    private double ticketPrice;
}

```

```
private int availableSeats;

@ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
@JoinColumn(name = "adminId")
private Admin admin;

@OneToMany(mappedBy = "flight")
private Set<Booking> bookings = new HashSet<>();

public Flight() {
}

public Flight(int flightId) {
    super();
    this.flightId = flightId;
}

// Getters and Setters
public int getFlightId() {
    return flightId;
}

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

public Airline getAirline() {
    return airline;
}

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

public Place getSourcePlace() {
    return sourcePlace;
}

public void setSourcePlace(Place sourcePlace) {
    this.sourcePlace = sourcePlace;
}

public Place getDestinationPlace() {
    return destinationPlace;
}
```



```
    public void setDestinationPlace(Place destinationPlace) {
        this.destinationPlace = destinationPlace;
    }

    public LocalDateTime getDepartureDateTime() {
        return departureDateTime;
    }

    public void setDepartureDateTime(LocalDateTime departureDateTime) {
        this.departureDateTime = departureDateTime;
    }

    public double getTicketPrice() {
        return ticketPrice;
    }

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

    public int getAvailableSeats() {
        return availableSeats;
    }

    public void setAvailableSeats(int availableSeats) {
        this.availableSeats = availableSeats;
    }

    public Admin getAdmin() {
        return admin;
    }

    public void setAdmin(Admin admin) {
        this.admin = admin;
    }

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

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

Place.java

```
package mypackage.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.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name = "Place")
public class Place {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int placeId;
    private String placeName;

    @ManyToOne(cascade = { CascadeType.PERSIST, CascadeType.MERGE })
    @JoinColumn(name = "adminId")
    private Admin admin;

    // Getters and Setters
    public int getPlaceId() {
        return placeId;
    }

    public void setPlaceId(int placeId) {
        this.placeId = placeId;
    }

    public String getPlaceName() {
        return placeName;
    }

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

    public Admin getAdmin() {
        return admin;
    }
}
```

```
    public void setAdmin(Admin admin) {  
        this.admin = admin;  
    }  
}
```

```
}
```

User.java

```
package mypackage.entities;
```

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

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

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

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

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

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

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

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

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name = "User")
```

```
public class User {
```

```
    @Id
```

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

```
    private int userId;
```

```
    private String uUsername;
```

```
    private String uPassword;
```

```
    private String uEmail;
```

```
    private String uFirstName;
```

```
    private String uLastName;
```

```
    private String uPhoneNumber;
```

```
    private String uAddress;
```

```
    @OneToMany(mappedBy = "user", cascade = { CascadeType.PERSIST,  
        CascadeType.MERGE, CascadeType.REMOVE })
```

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

```
    public User() {
```

```
    }
```

```
    public User(int userId) {
```

```
        super();
```

```
        this.userId = userId;
    }

    // Getters and Setters
    public int getUserId() {
        return userId;
    }

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

    public String getUUsername() {
        return uUsername;
    }

    public void setUUsername(String uUsername) {
        this.uUsername = uUsername;
    }

    public String getUPassword() {
        return uPassword;
    }

    public void setUPassword(String uPassword) {
        this.uPassword = uPassword;
    }

    public String getUEmail() {
        return uEmail;
    }

    public void setUEmail(String uEmail) {
        this.uEmail = uEmail;
    }

    public String getUFirstName() {
        return uFirstName;
    }

    public void setUFirstName(String uFirstName) {
        this.uFirstName = uFirstName;
    }

    public String getULastName() {
```

```

        return uLastName;
    }

    public void setuLastName(String uLastName) {
        this.uLastName = uLastName;
    }

    public String getuPhoneNumber() {
        return uPhoneNumber;
    }

    public void setuPhoneNumber(String uPhoneNumber) {
        this.uPhoneNumber = uPhoneNumber;
    }

    public String getuAddress() {
        return uAddress;
    }

    public void setuAddress(String uAddress) {
        this.uAddress = uAddress;
    }

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

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

```

4. Creating DAO interfaces

- Create a package 'mypackage.dao' in Java Resources.
- Create interfaces AdminDao.java, AirlineDao.java, BookingDao.java, FlightDao.java, PlaceDao.java, UserDao.java.
- Define CRUD operations required for each entity.

AdminDao.java

```

package mypackage.dao;

import mypackage.entities.Admin;

public interface AdminDao {

    void insert(Admin admin);

```

```
        void update(Admin admin);
        Admin getAdmin(String aUsername, String aPassword);
    }
```

AirlineDao.java

```
package mypackage.dao;

import java.util.List;

import mypackage.entities.Airline;

public interface AirlineDao {

    void add(Airline airline) throws Exception;
    void delete(int airlineId) throws Exception;
    Airline getAirlineById(int airlineId);
    Airline getAirlineByName(String airlineName);
    List<Airline> getAll();
}
```

BookingDao.java

```
package mypackage.dao;

import java.util.List;

import mypackage.entities.Booking;
import mypackage.entities.User;

public interface BookingDao {

    void add(Booking booking) throws Exception;
    void delete(int bookingId) throws Exception;
    List<Booking> getByUser(User user);
}
```

FlightDao .java

```
package mypackage.dao;

import java.time.LocalDate;
import java.util.List;

import mypackage.entities.Flight;

public interface FlightDao {

    void add(Flight flight) throws Exception;
    void delete(int flightId) throws Exception;
    void update(Flight flight) throws Exception;
```

```

        Flight getFlightById(int flightId);
        List<Flight> getAll();
        List<Flight> getFlightsBySearchCriteria(LocalDate dateOfTravel, int sourceId,
int destinationId, int numberOfPersons);
    }

```

PlaceDao.java

```

package mypackage.dao;

import java.util.List;

import mypackage.entities.Place;

public interface PlaceDao {

    void add(Place place) throws Exception;
    void delete(int placeId) throws Exception;
    Place getPlaceById(int placeId);
    Place getPlaceByName(String placeName);
    List<Place> getAll();
}

```

UserDao.java

```

package mypackage.dao;

import mypackage.entities.User;

public interface UserDao {

    void insert(User user);
    void update(User user);
    User getUser(String uUsername, String uPassword);
}

```

5. Creating DAOImpl classes

- Create a package 'mypackage.daoimpl' in Java Resources.
- Create java classes AdminDaoImpl.java, AirlineDaoImpl.java, BookingDaoImpl.java, FlightDaoImpl.java, UserDaoImpl.java and PlaceDaoImpl.java.
- Provide concrete implementations for the CRUD operations and other data access methods defined in the DAO interface.

AdminDaoImpl.java

```

package mypackage.daoimpl;

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

```

```

import org.hibernate.query.Query;

import mypackage.dao.AdminDao;
import mypackage.entities.Admin;
import mypackage.util.HibernateUtil;

public class AdminDaoImpl implements AdminDao {

    @Override
    public void insert(Admin admin) {

        SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
        Session session = sessionFactory.openSession();
        Transaction tx = null;
        try {
            tx = session.beginTransaction();
            session.save(admin);
            tx.commit();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            session.close();
        }
    }

    @Override
    public void update(Admin admin) {

        SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
        Session session = sessionFactory.openSession();
        Transaction tx = null;
        try {
            tx = session.beginTransaction();
            session.update(admin);
            tx.commit();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            session.close();
        }
    }

    @SuppressWarnings("unchecked")
    @Override
    public Admin getAdmin(String aUsername, String aPassword) {

```



```

        SessionFactory sessionFactory = null;
        Session session = null;
        Query<Admin> query = null;
        Admin admin = null;
        try {
            sessionFactory = HibernateUtil.buildSessionFactory();
            session = sessionFactory.openSession();
            query = session.createQuery("select a from mypackage.entities.Admin a
where a.aUsername = :username and a.aPassword = :password");
            query.setParameter("username", aUsername);
            query.setParameter("password", aPassword);
            admin = query.uniqueResult();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (session != null) {
                session.close();
            }
        }
        return admin;
    }
}

```

AirlineDaoImpl.java

```

package mypackage.daoimpl;

import java.util.List;

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

import mypackage.dao.AirlineDao;
import mypackage.entities.Airline;
import mypackage.util.HibernateUtil;

public class AirlineDaoImpl implements AirlineDao {

    @Override
    public void add(Airline airline) throws Exception {

        SessionFactory sessionFactory = null;
        Session session = null;
        Transaction tx = null;
    }
}

```

```

        try {
            sessionFactory = HibernateUtil.buildSessionFactory();
            session = sessionFactory.openSession();
            tx = session.beginTransaction();
            session.save(airline);
            tx.commit();
        } catch (Exception e) {
            if (tx != null) {
                tx.rollback();
            }
            throw new Exception("Failed to add airline.", e);
        } finally {
            session.close();
        }
    }
}

```

@Override

public void delete(int airlineId) throws Exception {

```

    sessionFactory = null;
    Session session = null;
    Transaction tx = null;
    try {
        sessionFactory = HibernateUtil.buildSessionFactory();
        session = sessionFactory.openSession();
        tx = session.beginTransaction();
        Airline airline = session.get(Airline.class, airlineId);
        session.delete(airline);
        tx.commit();
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        throw new Exception("Failed to delete airline.", e);
    } finally {
        session.close();
    }
}

```

@SuppressWarnings("unchecked")

@Override

public Airline getAirlineById(int airlineId) {

```

    sessionFactory = HibernateUtil.buildSessionFactory();
    Session session = sessionFactory.openSession();

```

```

        Query<Airline> query = session.createQuery("select a from
mypackage.entities.Airline a where a.airlineId = ?1");
        query.setParameter(1, airlineId);
        Airline result = (Airline) query.uniqueResult();
        session.close();
        return result;
    }

    @SuppressWarnings("unchecked")
    @Override
    public Airline getAirlineByName(String airlineName) {

        SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
        Session session = sessionFactory.openSession();
        Query<Airline> query = session.createQuery("select a from
mypackage.entities.Airline a where a.airlineName = ?1");
        query.setParameter(1, airlineName);
        Airline result = (Airline) query.uniqueResult();
        session.close();
        return result;
    }

    @SuppressWarnings("unchecked")
    @Override
    public List<Airline> getAll() {

        SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
        Session session = sessionFactory.openSession();
        Query<Airline> query = session.createQuery("select a from
mypackage.entities.Airline a");
        List<Airline> result = query.list();
        session.close();
        return result;
    }
}

```

BookingDaoImpl.java

```

package mypackage.daoimpl;

import java.util.List;

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

```

```

import mypackage.dao.BookingDao;
import mypackage.entities.Booking;
import mypackage.entities.User;
import mypackage.util.HibernateUtil;

public class BookingDaoImpl implements BookingDao{

    @Override
    public void add(Booking booking) throws Exception {

        SessionFactory sessionFactory = null;
        Session session = null;
        Transaction tx = null;
        try {
            sessionFactory = HibernateUtil.buildSessionFactory();
            session = sessionFactory.openSession();
            tx = session.beginTransaction();
            session.save(booking);
            tx.commit();
        } catch (Exception e) {
            if (tx != null) {
                tx.rollback();
            }
            throw new Exception("Failed to add booking.", e);
        } finally {
            session.close();
        }
    }

    @Override
    public void delete(int bookingId) throws Exception {

        SessionFactory sessionFactory = null;
        Session session = null;
        Transaction tx = null;
        try {
            sessionFactory = HibernateUtil.buildSessionFactory();
            session = sessionFactory.openSession();
            tx = session.beginTransaction();
            Booking booking = session.get(Booking.class, bookingId);
            session.delete(booking);
            tx.commit();
        } catch (Exception e) {
            if (tx != null) {
                tx.rollback();
            }
        }
    }
}

```

```

        }
        throw new Exception("Failed to delete booking.", e);
    } finally {
        session.close();
    }
}

@SuppressWarnings("unchecked")
@Override
public List<Booking> getByUser(User user) {

    SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
    Session session = sessionFactory.openSession();
    Query<Booking> query = session.createQuery("SELECT b FROM
mypackage.entities.Booking b WHERE b.user = :user");
    query.setParameter("user", user);
    List<Booking> bookings = query.list();
    session.close();
    return bookings;
}
}

```

FlightDaoImpl.java

```

package mypackage.daoimpl;

import java.time.LocalDate;
import java.util.List;

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

import mypackage.dao.FlightDao;
import mypackage.entities.Flight;
import mypackage.util.HibernateUtil;

public class FlightDaoImpl implements FlightDao {

    @Override
    public void add(Flight flight) throws Exception {

        SessionFactory sessionFactory = null;
        Session session = null;
        Transaction tx = null;
        try {

```

```

        sessionFactory = HibernateUtil.buildSessionFactory();
        session = sessionFactory.openSession();
        tx = session.beginTransaction();
        session.save(flight);
        tx.commit();
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        throw new Exception("Failed to add flight.", e);
    } finally {
        session.close();
    }
}

```

@Override

public void delete(int flightId) throws Exception {

```

    SessionFactory sessionFactory = null;
    Session session = null;
    Transaction tx = null;
    try {
        sessionFactory = HibernateUtil.buildSessionFactory();
        session = sessionFactory.openSession();
        tx = session.beginTransaction();
        Flight flight = session.get(Flight.class, flightId);
        session.delete(flight);
        tx.commit();
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        throw new Exception("Failed to delete flight.", e);
    } finally {
        session.close();
    }
}

```

@Override

public void update(Flight flight) throws Exception {

```

    SessionFactory sessionFactory = null;
    Session session = null;
    Transaction tx = null;
    try {

```

```

        sessionFactory = HibernateUtil.buildSessionFactory();
        session = sessionFactory.openSession();
        tx = session.beginTransaction();
        session.update(flight);
        tx.commit();
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        throw new Exception("Failed to add flight.", e);
    } finally {
        session.close();
    }
}

@Override
public Flight getFlightById(int flightId) {

    SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
    Session session = sessionFactory.openSession();
    Query<Flight> query = session.createQuery("select f from
mypackage.entities.Flight f where f.flightId = ?1");
    query.setParameter(1, flightId);
    Flight result = (Flight) query.uniqueResult();
    session.close();
    return result;
}

@Override
public List<Flight> getFlightsBySearchCriteria(LocalDate dateOfTravel, int
sourceId, int destinationId,
        int numberOfPersons) {

    SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
    Session session = sessionFactory.openSession();
    Query<Flight> query = session.createQuery("select f from
mypackage.entities.Flight f where "
        + "DATE(f.departureDateTime) = :dateOfTravel " + "and
f.sourcePlace.placeId = :sourceId "
        + "and f.destinationPlace.placeId = :destinationId " + "and
f.availableSeats >= :numberOfPersons",
        Flight.class);
    query.setParameter("dateOfTravel", java.sql.Date.valueOf(dateOfTravel));
    query.setParameter("sourceId", sourceId);

```

```

        query.setParameter("destinationId", destinationId);
        query.setParameter("numberOfPersons", numberOfPersons);
        List<Flight> result = query.getResultList();
        session.close();
        return result;
    }

    @SuppressWarnings("unchecked")
    @Override
    public List<Flight> getAll() {

        SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
        Session session = sessionFactory.openSession();
        Query<Flight> query = session.createQuery("select f from
mypackage.entities.Flight f");
        List<Flight> result = query.list();
        ;
        session.close();
        return result;
    }
}

```

PlaceDaoImpl.java

```

package mypackage.daoimpl;

import java.util.List;

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

import mypackage.dao.PlaceDao;
import mypackage.entities.Place;
import mypackage.util.HibernateUtil;

public class PlaceDaoImpl implements PlaceDao {

    @Override
    public void add(Place place) throws Exception {

        SessionFactory sessionFactory = null;
        Session session = null;
        Transaction tx = null;
        try {
            sessionFactory = HibernateUtil.buildSessionFactory();

```



```

        session = sessionFactory.openSession();
        tx = session.beginTransaction();
        session.save(place);
        tx.commit();
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        throw new Exception("Failed to add flight.", e);
    } finally {
        session.close();
    }
}

```

@Override

```
public void delete(int placeId) throws Exception {
```

```

    SessionFactory sessionFactory = null;
    Session session = null;
    Transaction tx = null;
    try {
        sessionFactory = HibernateUtil.buildSessionFactory();
        session = sessionFactory.openSession();
        tx = session.beginTransaction();

```

// Check if there are any flights associated with the place

```

        Query<Long> countQuery = session.createQuery("select count(*) from
mypackage.entities.Flight f where f.sourcePlace.placeId = :placeId or
f.destinationPlace.placeId = :placeId", Long.class);
        countQuery.setParameter("placeId", placeId);
        long flightCount = countQuery.uniqueResult();

```

```

        if (flightCount > 0) {
            throw new Exception("Cannot delete the place as there are associated
flights.");
        }

```

// Delete the place

```

        Place place = session.get(Place.class, placeId);
        session.delete(place);
        tx.commit();
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
    }
}

```

```

        throw new Exception("Failed to delete flight.", e);
    } finally {
        session.close();
    }
}

@SuppressWarnings("unchecked")
@Override
public Place getPlaceById(int placeId) {

    SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
    Session session = sessionFactory.openSession();
    Query<Place> query = session.createQuery("select p from
mypackage.entities.Place p where p.placeId = ?1");
    query.setParameter(1, placeId);
    Place result = (Place) query.uniqueResult();
    session.close();
    return result;
}

@SuppressWarnings("unchecked")
@Override
public Place getPlaceByName(String placeName) {

    SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
    Session session = sessionFactory.openSession();
    Query<Place> query = session.createQuery("select p from
mypackage.entities.Place p where p.placeName = ?1");
    query.setParameter(1, placeName);
    Place result = (Place) query.uniqueResult();
    session.close();
    return result;
}

@SuppressWarnings("unchecked")
@Override
public List<Place> getAll() {

    SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
    Session session = sessionFactory.openSession();
    Query<Place> query = session.createQuery("select p from
mypackage.entities.Place p");
    List<Place> result = query.list();
    session.close();
    return result;
}

```

```
    }  
}
```

UserDaoImpl.java

```
package mypackage.daoimpl;
```

```
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.Transaction;  
import org.hibernate.query.Query;
```

```
import mypackage.dao.UserDao;  
import mypackage.entities.User;  
import mypackage.util.HibernateUtil;
```

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

```
    @Override
```

```
    public void insert(User user) {
```

```
        SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
```

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

```
        Transaction tx = null;
```

```
        try {
```

```
            tx = session.beginTransaction();
```

```
            session.save(user);
```

```
            tx.commit();
```

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

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

```
                tx.rollback();
```

```
        }
```

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

```
    } finally {
```

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

```
    }
```

```
}
```

```
    @Override
```

```
    public void update(User user) {
```

```
        SessionFactory sessionFactory = HibernateUtil.buildSessionFactory();
```

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

```
        Transaction tx = null;
```

```
        try {
```

```
            tx = session.beginTransaction();
```

```
            session.update(user);
```

```

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

@SuppressWarnings("unchecked")
@Override
public User getUser(String uUsername, String uPassword) {

    SessionFactory sessionFactory = null;
    Session session = null;
    Query<User> query = null;
    User user = null;
    try {
        sessionFactory = HibernateUtil.buildSessionFactory();
        session = sessionFactory.openSession();
        query = session.createQuery("select u from mypackage.entities.User u where
u.uUsername = :username and u.uPassword = :password");
        query.setParameter("username", uUsername);
        query.setParameter("password", uPassword);
        user = query.uniqueResult();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (session != null) {
            session.close();
        }
    }
    return user;
}

}

```

6. Creating .jsp files

- Create .jsp files in webapp folder in src.
- Create index.jsp file which serves as the entry point of the application and other .jsp files for creating views in the web application.
- Create .css files or embed the CSS styles in .jsp file itself.

index.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"% >
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Choose Account Type</title>
<style>
body {
    font-family: Arial, sans-serif;
    text-align: center;
}

.accounttype {
    border: 2px solid #2095bf;
    border-radius: 10px;
    display: inline-block;
    width: 300px;
    height: 300px;
    margin-top: 80px;
}

.accounttype h1 {
    margin: 10px 0;
    padding: 0;
    color: #2095bf;
}

.options {
    display: flex;
    justify-content: center;
    margin-top: 30px;
}

.option {
    margin: 20px;
    cursor: pointer;
}

.option img {
    width: 80px;
    height: 80px;
    border-radius: 50%;
```

```

        box-shadow: 0 4px 8px #2095bf;
        transition: transform 0.3s ease-in-out;
    }

.option img:hover {
    transform: scale(1.1);
}

.option span {
    font-size: 18px;
    margin-top: 10px;
    margin-left: 15px;
}
</style>
</head>
<body>
    <% @ include file="header.jsp"% >
    <div class="accounttype">
        <h1>Choose Account Type</h1>
        <div class="options">
            <div class="option">
                <a href="userlogin.jsp"> 
<span>User</span>
                </a>
            </div>
            <div class="option">
                <a href="adminlogin.jsp"> 
<span>Admin</span>
                </a>
            </div>
        </div>
    </div>
</body>
</html>

```

about.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"% >
<!DOCTYPE html>
<html>
<head>
    <style>
        h1 {
            color: #2095bf;
            text-align: center;

```

```

    }

.details-container {
    width: 600px;
    color: #2095bf;
    text-align: justify;
    padding-left: 618px;
}

.details {
    width: 300px;
    word-wrap: break-word;
    text-align: center;
}
</style>
<meta charset="UTF-8">
<title>About Developer</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <h1> About Developer </h1>

    <div class="details-container">
        <div class="details">
            <p>
                Name : Tejeswi Devi Priya<br><br>
                Email ID : priyapillarisetty19@gmail.com
            </p>
            <p>
                Thank you for exploring Fly Away Airline Booking Portal.
            </p>
            <p>
                Should you have any queries or feedback regarding the portal, please feel free
                to reach out to me at the provided email address.
            </p>
        </div>
    </div>
</body>
</html>

```

[addairline.jsp](#)

```

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

```

```

<html>
<head>
<style>
form {
    padding-top: 50px;
}

fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}

legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}

label {
    display: inline-block;
    width: 80px;
    text-align: right;
    font-weight: 600;
    color: #2095bf;
    margin-right: 5px;
}
input[type="text"] {
    width: 167px;
}
</style>
<meta charset="UTF-8">
<title>Add an Airline</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <form action="airlinecontroller" method="post" style="width: 400px; margin:
0px auto;">
        <fieldset>
            <legend>Add Airline Form</legend><br>
            <label>Name : </label><input type="text" name="airlineName"
required><br><br>

```



```
        <input type="submit" value="Submit">
    </fieldset>
</form>
</body>
</html>
```

addflight.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.util.List"%>
<% @ page import="mypackage.entities.Airline"%>
<% @ page import="mypackage.dao.AirlineDao"%>
<% @ page import="mypackage.daoimpl.AirlineDaoImpl"%>
<% @ page import="mypackage.serviceimpl.AirlineServiceImpl"%>
<% @ page import="mypackage.entities.Place"%>
<% @ page import="mypackage.dao.PlaceDao"%>
<% @ page import="mypackage.daoimpl.PlaceDaoImpl"%>
<% @ page import="mypackage.serviceimpl.PlaceServiceImpl"%>
<head>
<style>
form {
    padding-top: 50px;
}

fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}

legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}

label {
    display: inline-block;
    width: 200px;
    text-align: right;
    font-weight: 600;
    color: #2095bf;
```

```

        margin-right: 5px;
    }

    input[type="text"], input[type="number"], select {
        width: 167px;
    }

    select option[disabled], input[type="datetime-local"] {
        text-align: center;
        padding: 0;
    }
</style>
<meta charset="UTF-8">
<title>Add a Flight</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <%
        AirlineDao airlineDao = new AirlineDaoImpl();
        List<Airline> airlines = airlineDao.getAll();
        request.setAttribute("airlines", airlines);

        PlaceDao placeDao = new PlaceDaoImpl();
        List<Place> places = placeDao.getAll();
        request.setAttribute("places", places);
    %>
    <form action="flightcontroller" method="post" style="width: 500px; margin: 0px
auto;">
        <fieldset>
            <legend>Add Flight Form</legend>
            <br>
            <p>
                <label> Select Airline : </label> <select name="airlineId"
required>
                    <option disabled selected value>-- select airline --
                </option>
                    <% for (Airline airline : airlines) { %>
                        <option value="<%= airline.getAirlineId()
%>"><%= airline.getAirlineName() %></option>
                    <% } %>
                </select>
            </p>
            <p>
            <p>

```

```

        <label> Select Source : </label> <select name="sourceId"
required>
        <option disabled selected value>-- select source --
</option>
        <% for (Place place : places) { %>
        <option value="<%= place.getPlaceId() %>"><%=
place.getPlaceName() %></option>
        <% } %>
        </select>
    </p>
    <p>
        <label> Select Destination : </label> <select
name="destinationId" required>
        <option disabled selected value>-- select destination
--</option>
        <% for (Place place : places) { %>
        <option value="<%= place.getPlaceId() %>"><%=
place.getPlaceName() %></option>
        <% } %>
        </select>
    </p>
    <p>
        <label>Departure Date and Time :</label> <input
type="datetime-local" name="departureDateTime"
min="<%=java.time.LocalDateTime.now().format(java.time.format.DateTimeFormatter.I
SO_LOCAL_DATE_TIME)%>" required>
    </p>
    <p>
        <label>Ticket Price : </label> <input type="number"
name="ticketPrice" step="0.01" required>
    </p>
    <p>
        <label>Available Seats : </label> <input type="number"
name="availableSeats" min="50" max="150" required>
    </p>
    <input type="submit" value="Submit">
</fieldset>
</form>
</body>
</html>

```

addplace.jsp

```

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

```

```

<html>
<head>
<style>
form {
    padding-top: 50px;
}

fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}

legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}

label {
    display: inline-block;
    width: 80px;
    text-align: right;
    font-weight: 600;
    color: #2095bf;
    margin-right: 5px;
}
input[type="text"] {
    width: 167px;
}
</style>
<meta charset="UTF-8">
<title>Add Place</title>
</head>
<body>
    <% @ include file="header.jsp"% >

    <form action="placecontroller" method="post" style="width: 400px; margin: 0px
auto;">
        <fieldset>
            <legend>Add Place Form</legend><br>
            <label>Name : </label><input type="text" name="placeName"
required><br><br>

```

```

        <input type="submit" value="Submit">
    </fieldset>
</form>
</body>
</html>

```

adminchangepassword.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
form {
    padding-top: 50px;
}

fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}

label {
    width: 80px;
    font-weight: 600;
    color: #2095bf;
    margin-right: 5px;
}

input[text="submit"] {
    text-align: center;
}
</style>
<meta charset="UTF-8">
<title>Password Change</title>
</head>
<body>
    <% @ include file="header.jsp"%>
    <div class="box">
        <form action="adminpasswordcontroller" method="post" style="width:
400px; margin: 0px auto;">
            <fieldset>
                <br><label>Do you want to change your password?</label><br><br>
                <input type="submit" name="action" value="Yes">
            </fieldset>
        </form>
    </div>

```

```

        <input type="submit" name="action" value="No">
        <br><br>
    </fieldset>
    </form>
</div>
</body>
</html>

```

admincode.jsp

- Please note that the admin verification code is C3H9S4

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
form {
    padding-top: 50px;
}
fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}
legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}
label {
    color: #2095bf;
}
.option {
    font-size: 12px;
    padding-top: 8px;
    color: #2095bf;
}
</style>
<meta charset="UTF-8">
<title>Admin Verification</title>
</head>
<body>
    <% @ include file="header.jsp"%>

```

```

        <form action="adminverificationcontroller" method="post" style = "width:
400px; margin: 0px auto;">
            <fieldset>
                <legend>Admin Verification</legend><br>
                <label>Please enter a unique 6-digit alphanumeric verification
code to confirm your eligibility for registering as an admin</label><br><br>
                <label>Enter Code : </label><input type="password"
name="code" required><br><br>

                <input type="submit" value="Submit">

                <div class="option">
                    Not an admin?<a href="index.jsp">Click here</a>
                </div>
            </fieldset>
        </form>
</body>
</html>

```

adminhome.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="adminhome-styles.css">
<meta charset="UTF-8">
<title>Admin Home</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <div class="heading">
        <h2>Welcome to Fly Away</h2>
    </div>
    <div class="tiles">
        <div class="column-2 option-1">
            <h2 class="option-text"><a
href="placeslist.jsp">Places<br>List</a></h2>
        </div>
        <div class="column-2 option-2">
            <h2 class="option-text"><a
href="airlineslist.jsp">Airlines<br>List</a></h2>
        </div>
        <div class="column-2 option-3">

```

```
                <h2 class="option-text"><a
href="flightslist.jsp">Flights<br>List</a></h2>
            </div>
    </div>
```

```
</body>
```

```
</html>
```

admininvalidcode.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
```

```
</style>
```

```
<meta charset="UTF-8">
```

```
<title>Invalid Verification Code</title>
```

```
</head>
```

```
<body>
```

```
    <% @ include file="header.jsp"%>
```

```
    <h2>You have entered invalid verification code! <br><br>
```

```
    <a href="admincode.jsp">Click here</a> to try again.</h2>
```

```
</body>
```

```
</html>
```

admininvalidcred.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
```

```
</style>
```

```
<meta charset="UTF-8">
```



```
<title>Invalid Credentials</title>
</head>
<body>

    <% @ include file="header.jsp"%>

    <h2>You have entered invalid credentials! <br><br>
    <a href="adminlogin.jsp">Click here</a> to try again.</h2>
</body>
</html>
```

adminlogin.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
.welcome {
    text-align: center;
    padding-bottom: 20px;
    color: #2095bf;
}
fieldset {
    text-align: center;
    border: 2px #2095bf solid;
}
legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}
.option {
    font-size: 12px;
    padding-top: 8px;
    color: #2095bf;
}
label {
    color: #2095bf;
}
</style>

<meta charset="UTF-8">
<title>Admin Login Page</title>
```

```

</head>
<body>

    <% @ include file="header.jsp"%>

    <div class="welcome">
        <h1>Fly Away</h1>
    </div>
    <form action="adminlogincontroller" method="post" style = "width: 400px;
margin: 0px auto;">
        <fieldset>
            <legend>Admin Login</legend><br>
            <label>Username : </label><input type="text"
name="aUsername" required><br><br>
            <label>Password : </label><input type="password"
name="aPassword" required><br><br>
            <input type="submit" value="Submit">

            <div class="option">
                Not a registered admin?<a href="admincode.jsp">Click
here</a>
            </div>
        </fieldset>
    </form>

</body>
</html>

```

adminregister.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
form {
    padding-top: 50px;
}
fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}
legend {
    text-align: center;

```

```

        color: #2095bf;
        font-size: 20px;
        font-weight: 700;
    }
    label {
        color: #2095bf;
    }
</style>
<meta charset="UTF-8">
<title>Admin Registration</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <form action="adminregistercontroller" method="post" style = "width: 400px;
margin: 0px auto;">
        <fieldset>
            <legend>Admin Registration Form</legend><br>
            <label>Username : </label><input type="text"
name="aUsername" required><br><br>
            <label>Password : </label><input type="password"
name="aPassword" required><br><br>
            <input type="submit" value="Submit">
        </fieldset>
    </form>
</body>
</html>

```

adminregistrationsuccessful.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
</style>
<meta charset="UTF-8">
<title>Registration Successful!</title>
</head>
<body>

```

```
<% @ include file="header.jsp"%>

<h2>Your registration is successful. <br><br>
<a href="adminlogin.jsp">Click here</a> to login.</h2>

</body>
</html>
```

arlineslist.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.util.List" %>
<% @ page import="mypackage.entities.Airline" %>
<% @ page import="mypackage.dao.AirlineDao" %>
<% @ page import="mypackage.daoimpl.AirlineDaoImpl" %>
<head>
<style>
    table {
        border-collapse: collapse;
        width: 40%;
        margin: auto;
    }
    th, td {
        border: 2px #2095bf solid;
        text-align: center;
    }
    th {
        color: #2095bf;
        font-weight: bold;
    }
    .head1 {
        color: #2095bf;
        padding-top: 10px;
        padding-bottom: 10px;
        text-align: center;
    }
</style>

<meta charset="UTF-8">
<title>Airlines List</title>
</head>
<body>
```

```

<% @ include file="header.jsp"%>
<%
    AirlineDao airlineDao = new AirlineDaoImpl();
    List<Airline> airlines = airlineDao.getAll();
    request.setAttribute("airlines", airlines);
%>
<div class="head1">
    <h1> Airlines List </h1>
</div>
<table>
    <tr>
        <th> Airline ID </th>
        <th> Airline Name </th>
        <th> Actions </th>
    </tr>
    <% int sequence = 1; %>
    <%
        for(Airline airline : airlines) {
    %>
        <tr>
            <td><%= sequence %></td>
            <td><%= airline.getAirlineName() %></td>
            <td>
                <form action="airlineslist.jsp" method="POST">
                    <input type="hidden" name="airlineId"
value="<%= airline.getAirlineId() %>">
                    <button type="submit" name="action"
value="delete">Delete</button>
                </form>
            </td>
        </tr>
        <% sequence++; %>
    <%
        }
    %>
    <tr>
        <td colspan="3">
            <form action="addairline.jsp" method="GET">
                <input type="submit" value="Add
Airline"></input>
            </form>
        </td>
    </tr>
</table>
<%

```

```

        String action = request.getParameter("action");
        if (action != null && action.equals("delete")) {
            int airlineId = Integer.parseInt(request.getParameter("airlineId"));
            airlineDao.delete(airlineId);
            response.sendRedirect("airlineslist.jsp");
        }
    %>
</body>
</html>
bookingdetails.jsp
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.util.List"%>
<% @ page import="mypackage.entities.Booking"%>
<% @ page import="java.time.format.DateTimeFormatter"%>
<head>
<style>
.details {
    border: 2px #2095bf solid;
    text-align: center;
    border-collapse: collapse;
    width: 30%;
    margin: auto;
}

label {
    color: #2095bf;
    font-weight: bold;
}

.heading {
    color: #2095bf;
    padding-top: 10px;
    padding-bottom: 10px;
    text-align: center;
}
</style>

<meta charset="UTF-8">
<title>Booking Details</title>
</head>
<body>
    <% @ include file="header.jsp"%>

```

```

<div class="heading">
    <h2>Booking Details</h2>
</div>
<div class="details">
    <%
Booking booking = (Booking) request.getSession().getAttribute("booking");
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-
MM-yyyy HH:mm");
String formattedDateTime =
booking.getBookingDateTime().format(formatter);

    if (booking != null) {
    %>
        <div>
            <br><label>Flight ID : </label> <span><%=
booking.getFlight().getFlightId() %></span><br><br>
        </div>
        <div>
            <label>Booking Date & Time : </label> <span><%=
formattedDateTime %></span><br><br>
        </div>
        <div>
            <label>Total Tickets : </label> <span><%=
booking.getTotalTickets() %></span><br><br>
        </div>
        <div>
            <label>Total Price : </label> <span><%= booking.getTotalPrice()
%></span><br><br>
        </div>
        <div>
            <label>Payment Status : </label> <span><%=
booking.getPaymentStatus() %></span><br><br>
        </div>

        <%
    } else {
    %>
        <div>No booking details available for this user.</div>
        <%
    }
    %>
</div>

</body>

```

```
</html>
```

cannotdeleteplace.jsp

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<style>
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
</style>
<meta charset="ISO-8859-1">
<title>Place cannot be deleted</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <h2>Cannot delete the place as there are associated flights!! <br><br>
    <a href="placeslist.jsp">Click here</a> to return to Places List page.</h2>
</body>
</html>
```

flightdetails.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.util.List"%>
<% @ page import="mypackage.entities.Flight"%>
<% @ page import="java.time.format.DateTimeFormatter"%>
<head>
<style>
.details {
    border: 2px #2095bf solid;
    text-align: center;
    border-collapse: collapse;
    width: 30%;
    margin: auto;
}
</style>
```



```

label {
    color: #2095bf;
    font-weight: bold;
}

.heading {
    color: #2095bf;
    padding-top: 10px;
    padding-bottom: 10px;
    text-align: center;
}
</style>

<meta charset="UTF-8">
<title>Flight Details</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <div class="heading">
        <h2>Flight Details</h2>
    </div>
    <div class="details">
        <%
            Flight flight = (Flight) session.getAttribute("flight");
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy
HH:mm");

            if (flight != null) {
        %>
            <div>
                <br><label>Airline : </label> <span><%=
flight.getAirline().getAirlineName() %></span><br><br>
            </div>
            <div>
                <label>Source : </label> <span><%=
flight.getSourcePlace().getPlaceName() %></span><br><br>
            </div>
            <div>
                <label>Destination : </label> <span><%=
flight.getDestinationPlace().getPlaceName() %></span><br><br>
            </div>
            <div>
                <label>Departure Date & Time : </label> <span><%=
flight.getDepartureDateTime().format(formatter) %></span><br><br>

```

```

        </div>
        <div>
            <label>Ticket Price : </label> <span><%= flight.getTicketPrice()
%></span><br><br>
        </div>
        <div>
            <label>No. of Persons : </label> <span><%=
session.getAttribute("numberOfPersons") %></span><br><br>
        </div>
        <div>
            <label>Total Price : </label> <span><%=
session.getAttribute("totalPrice") %></span><br><br>
        </div>
        <form action="smoothpay.jsp">
            <input type="submit" value="Pay Now"><br><br>
        </form>
        <%
    } else {
%>
        <div>No flight details available.</div>
        <%
    }
%>
    </div>

</body>
</html>

```

flightsearchresults.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.util.List"%>
<% @ page import="mypackage.entities.Flight"%>
<% @ page import="java.time.format.DateTimeFormatter"%>
<head>
<style>
table {
    border-collapse: collapse;
    width: 70%;
    margin: auto;
}

th, td {

```

```

        border: 2px #2095bf solid;
        text-align: center;
    }

    th {
        color: #2095bf;
        font-weight: bold;
    }

    .heading {
        color: #2095bf;
        padding-top: 10px;
        padding-bottom: 10px;
        text-align: center;
    }
</style>

<meta charset="UTF-8">
<title>Flight Search Results</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <div class="heading">
        <h2>Flight Search Results</h2>
    </div>

    <table>
        <tr>
            <th>Airline</th>
            <th>Source</th>
            <th>Destination</th>
            <th>Departure Date & Time</th>
            <th>Ticket Price</th>
            <th>Available Seats</th>
            <th>Actions</th>
        </tr>
        <%
            List<Flight> flights = (List<Flight>) request.getAttribute("flights");
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-
yyyy HH:mm");

            if (flights != null && !flights.isEmpty()) {
                for (Flight flight : flights) {
                    %>

```

```

        <tr>
            <td><%=flight.getAirline().getAirlineName()%></td>
            <td><%=flight.getSourcePlace().getPlaceName()%></td>
            <td><%=flight.getDestinationPlace().getPlaceName()%></td>

            <td><%=flight.getDepartureDateTime().format(formatter)%></td>
            <td><%=flight.getTicketPrice()%></td>
            <td><%=flight.getAvailableSeats()%></td>
            <td>
                <form action="userdetails.jsp" method="POST">
                    <input type="hidden" name="flightId"
value="<%=flight.getFlightId()%>">
                    <input type="hidden" name="ticketPrice"
value="<%=flight.getTicketPrice()%>">
                    <button type="submit" name="action"
value="book">Book</button>
                </form>
            </td>
        </tr>
    <%
    }
    } else {
    %>
    <tr>
        <td colspan="7">No flights found for the selected criteria.</td>
    </tr>
    <%
    }
    %>
</table>
</body>
</html>

```

flightslist.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.util.List" %>
<% @ page import="mypackage.entities.Flight" %>
<% @ page import="mypackage.dao.FlightDao" %>
<% @ page import="mypackage.daoimpl.FlightDaoImpl" %>
<% @ page import="java.time.format.DateTimeFormatter" %>
<head>
<style>

```

```

table {
    border-collapse: collapse;
    width: 70%;
    margin: auto;
}
th, td {
    border: 2px #2095bf solid;
    text-align: center;
}
th {
    color: #2095bf;
    font-weight: bold;
}
.head1 {
    color: #2095bf;
    padding-top: 10px;
    padding-bottom: 10px;
    text-align: center;
}
</style>

<meta charset="UTF-8">
<title>Flights List</title>
</head>
<body>
    <% @ include file="header.jsp"% >
    <%

        FlightDao flightDao = new FlightDaoImpl();
        List<Flight> flights = flightDao.getAll();
        request.setAttribute("flights", flights);

        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-
yyyy HH:mm");
    %>
    <div class="head1">
        <h1> Flights List </h1>
    </div>
    <table>
        <tr>
            <th> Flight ID </th>
            <th> Airline Name </th>
            <th> Source </th>
            <th> Destination </th>
            <th> Departure Date & Time </th>
            <th> Ticket Price </th>

```

```

                <th> Available Seats </th>
                <th> Actions </th>
            </tr>
            <% int sequence = 1; %>
            <%
                for(Flight flight : flights) {
            %>
                <tr>
                    <td><%= sequence %></td>
                    <td><%= flight.getAirline().getAirlineName() %></td>
                    <td><%= flight.getSourcePlace().getPlaceName() %></td>
                    <td><%= flight.getDestinationPlace().getPlaceName()
            %></td>
                    <td><%= flight.getDepartureDateTime().format(formatter)
            %></td>
                    <td><%= flight.getTicketPrice() %></td>
                    <td><%= flight.getAvailableSeats() %></td>
                    <td>
                        <form action="flightslist.jsp" method="POST">
                            <input type="hidden" name="flightId"
            value="<%= flight.getFlightId() %>">
                            <button type="submit" name="action"
            value="delete">Delete</button>
                        </form>
                    </td>
                </tr>
            <% sequence++; %>
            <%
                }
            %>
            <tr>
                <td colspan="8">
                    <form action="addflight.jsp" method="GET">
                        <input type="submit" value="Add
            Flight"></input>
                    </form>
                </td>
            </tr>
        </table>
        <%
            String action = request.getParameter("action");
            if (action != null && action.equals("delete")) {
                int flightId = Integer.parseInt(request.getParameter("flightId"));
                flightDao.delete(flightId);
                response.sendRedirect("flightslist.jsp");
            }
        %>
    
```

```
        }  
    %>  
</body>  
</html>
```

header.jsp

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8" />  
<meta http-equiv="X-UA-Compatible" content="IE=edge" />  
<meta name="viewport" content="width=device-width, initial-scale=1.0" />  
<style>  
@import  
    url("https://fonts.googleapis.com/css2?family=Open+Sans&display=swap");  
  
body {  
    font-family: "Open Sans", sans-serif;  
    margin: 0;  
    background: url(bg.jpg);  
    background-repeat: no-repeat;  
    background-size: 1536px 745px;  
}  
  
a {  
    text-decoration: none;  
    color: #FF5F1F;  
    padding-left: 15px;  
}  
  
a:hover {  
    color: #fff;  
}  
  
.header {  
    border-bottom: 1px solid #ccc;  
    display: flex;  
    background: #2095bf;  
    justify-content: space-between;  
}  
  
.site-name h1 {  
    font-size: 28px;  
    margin: 10px 10px 0px 20px;  
    display: inline-block;
```

```

}

.nav-options ul, .nav-options li {
    margin: 0;
    padding-right: 30px;
}

.nav-options li {
    display: inline-block;
    margin: 23px 15px 16px 16px;
    font-weight: 600;
}
</style>
<title>Header</title>
</head>
<body>
    <div class="bgimg">
        <header class="header">
            <div class="site-name">
                <h1>
                    <a href="index.jsp">Fly Away</a>
                </h1>
            </div>
            <nav class="nav-options">
                <ul class="nav">
                    <li><a href="userlogin.jsp">Home</a></li>
                    <li><a href="about.jsp">About</a></li>
                    <li><a href="logout.jsp">Logout</a></li>
                </ul>
            </nav>
        </header>
    </div>
</body>
</html>

```

logout.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="javax.servlet.http.HttpSession.*" %>
<head>
<style>
h2 {
    padding-left: 10px;
    padding-top: 20px;

```



```
        text-align: center;
    }
</style>
<meta charset="UTF-8">
<title>Logged out</title>
</head>
<body>

    <%@ include file="header.jsp"%>

    <h2>You have logged out successfully. <br><br>
    <a href="index.jsp">Click here</a> to login again.</h2>

    <%
        HttpSession sess = request.getSession(false);
        if (sess != null) {
            session.invalidate();
        }
    %>

</body>
</html>
```

newpassword.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
form {
    padding-top: 50px;
}

fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}

label {
    width: 80px;
    font-weight: 600;
    color: #2095bf;
    margin-right: 5px;
```

```

    }

    input[text="submit"] {
        text-align: center;
    }
</style>
<meta charset="UTF-8">
<title>Password Change</title>
</head>
<body>
    <% @ include file="header.jsp"%>
    <div class="box">
        <form action="adminpasswordcontroller" method="post" style="width:
400px; margin: 0px auto;">
            <fieldset>
                <br><label>Enter New Password : </label><input type="password"
name="newpassword"><br><br>
                <input type="submit" name="action" value="Submit">
                <br><br>
            </fieldset>
        </form>
    </div>
</body>
</html>

```

passwordchanged.jsp

```

    <% @ page language="java" contentType="text/html; charset=UTF-8"
        pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
</style>
<meta charset="UTF-8">
<title>Admin Password Changed</title>
</head>
<body>

    <% @ include file="header.jsp"%>

```

```
<h2>You have changed your password successfully. <br><br>
<a href="adminhome.jsp">Click here</a> to go to home page.</h2>
</body>
</html>
```

paymentfailed.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
</style>
<meta charset="UTF-8">
<title>Payment Failed</title>
</head>
<body>
```

```
    <% @ include file="header.jsp"%>

    <h2>Your payment failed! <br><br>
    <a href="userhome.jsp">Click here</a> to try again.</h2>
</body>
</html>
```

placeslist.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.util.List" %>
<% @ page import="mypackage.entities.Place" %>
<% @ page import="mypackage.dao.PlaceDao" %>
<% @ page import="mypackage.daoimpl.PlaceDaoImpl" %>
<head>
<style>
table {
    border-collapse: collapse;
    width: 40%;
    margin: auto;
}
</style>
```

```

th, td {
    border: 2px #2095bf solid;
    text-align: center;
}
th {
    color: #2095bf;
    font-weight: bold;
}
.head1 {
    color: #2095bf;
    padding-top: 10px;
    padding-bottom: 10px;
    text-align: center;
}
</style>

<meta charset="UTF-8">
<title>Places List</title>
</head>
<body>
    <% @ include file="header.jsp"%>
    <%
        PlaceDao placeDao = new PlaceDaoImpl();
        List<Place> places = placeDao.getAll();
        request.setAttribute("places", places);
    %>
    <div class="head1">
        <h1> Places List </h1>
    </div>
    <table>
        <tr>
            <th> Place ID </th>
            <th> Place Name </th>
            <th> Actions </th>
        </tr>
        <% int sequence = 1; %>
        <%
            for(Place place : places) {
        %>
            <tr>
                <td><%= sequence %></td>
                <td><%= place.getPlaceName() %></td>
                <td>
                    <form action="placeslist.jsp" method="POST">

```

```

        value="<%= place.getId() %>"
        <input type="hidden" name="placeId"
        <button type="submit" name="action"
        value="delete">Delete</button>
    </form>
</td>
</tr>
<% sequence++; %>
<%
    }
%>
    <tr>
        <td colspan="3">
            <form action="addplace.jsp" method="GET">
                <input type="submit" value="Add
Place"></input>
            </form>
        </td>
    </tr>
</table>
<%
    String action = request.getParameter("action");
    if (action != null && action.equals("delete")) {
        int placeId = Integer.parseInt(request.getParameter("placeId"));
        try {
            placeDao.delete(placeId);
            response.sendRedirect("placeslist.jsp");
        } catch (Exception e) {
            response.sendRedirect("cannotdeleteplace.jsp");
        }
    }
%>
</body>
</html>

```

smoothpay.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<% @ page import="java.time.LocalDate" %>
<head>
<style>
form {
    padding-top: 50px;

```

```

    }
    fieldset {
        text-align: center;
        border: 2px #2095bf solid;
        overflow: auto;
    }
    legend {
        text-align: center;
        color: #2095bf;
        font-size: 20px;
        font-weight: 700;
    }
    label, input[type="number"] {
        display: inline-block;
        width: 160px;
        text-align: right;
        font-weight: 600;
        color: #2095bf;
        margin-right: 5px;
    }
    select {
        display: inline-block;
        width: 160px;
        text-align: center;
    }
</style>
<meta charset="UTF-8">
<title>SmoothPay Payment Gateway</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <form action="paymentcontroller" method="post" style="width: 450px; margin:
0px auto;">
    <fieldset>
        <legend>SmoothPay - Payment Details</legend>
        <br><input type="hidden" name="bookingId" value="<%=
request.getParameter("bookingId") %>">
        <label>Card Number : </label><input type="text" name="cardNumber"
required><br><br>
        <label>Card Holder Name : </label><input type="text" name="cardHolderName"
required><br><br>
        <label>Expiration Month : </label>
            <select name="expirationMonth" required>

```

```

                                <option disabled selected value>-- select month --
</option>
                                <% for (int month = 1; month <= 12; month++) {
%>
                                <option value="<%= month %>"><%= month %></option>
                                <% } %>
                                </select><br><br>
                                <label>Expiration Year : </label><input type="number" name="expirationYear"
min="<%= LocalDate.now().getYear() %>" maxlength="4" required><br><br>
                                <label>CVV : </label><input type="number" name="cvv" maxlength="3"
required><br><br>

                                <input type="submit" value="Submit"><br><br>
                                </fieldset>
                                </form>
                                </body>
                                </html>

```

userdetails.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
form {
    padding-top: 50px;
}
fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}
legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}
label {
    display: inline-block;
    width: 130px;
    text-align: right;
    font-weight: 600;
    color: #2095bf;

```

```

        margin-right: 5px;
    }
    span {
        display: inline-block;
        width: 163px;
        text-align: left;
        font-weight: 600;
        margin-right: 5px;
    }
</style>
<meta charset="UTF-8">
<title>User Details for Flight Booking</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <form action="userdetailscontroller" method="post" style = "width: 400px;
margin: 0px auto;">
        <fieldset>
            <legend>User Details Form</legend><br>
            <input type="hidden" name="flightId" value="<%=
request.getParameter("flightId") %>">
            <input type="hidden" name="ticketPrice" value="<%=
request.getParameter("ticketPrice") %>">
            <label>UserName : </label><span><%=
session.getAttribute("uUsername") %></span><br><br>
            <label>Email : </label><input type="text" name="uEmail"
required><br><br>
            <label>First Name : </label><input type="text"
name="uFirstName" required><br><br>
            <label>Last Name : </label><input type="text"
name="uLastName" required><br><br>
            <label>Phone Number : </label><input type="text"
name="uPhoneNumber" required><br><br>
            <label>Address : </label><input type="text" name="uAddress"
required><br><br>
            <input type="submit" value="Submit">
        </fieldset>
    </form>
</body>
</html>

```

userhome.jsp

```

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

```



```

<!DOCTYPE html>
<html>
<% @ page import="java.util.List"%>
<% @ page import="mypackage.entities.Place"%>
<% @ page import="mypackage.dao.PlaceDao"%>
<% @ page import="mypackage.daoimpl.PlaceDaoImpl"%>
<% @ page import="mypackage.serviceimpl.PlaceServiceImpl"%>
<head>
<link rel="stylesheet" href="userhome-styles.css">
<meta charset="UTF-8">
<title>User Home</title>
</head>
<body>
    <% @ include file="header.jsp"%>

    <%
        PlaceDao placeDao = new PlaceDaoImpl();
        List<Place> places = placeDao.getAll();
        request.setAttribute("places", places);
    %>

    <div class="heading">
        <h2>Welcome to Fly Away</h2>
    </div>
    <form action="flightsearchcontroller" method="post" style="width: 450px;
margin: 0px auto;">
        <fieldset>
            <legend>Flight Search</legend>
            <br>
            <p>
                <label>Date of Travel :</label> <input type="date"
name="dateOfTravel" min="<%=java.time.LocalDate.now()%>" required>
            </p>
            <p>
                <label> Select Source : </label> <select name="sourceId"
required>
                    <option disabled selected value>-- select source --
                </option>
                <% for (Place place : places) { %>
                    <option value="<%= place.getPlaceId() %>"><%=
place.getPlaceName() %></option>
                <% } %>
                </select>
            </p>

```

```

        <p>
            <label> Select Destination : </label> <select
name="destinationId" required>
                <option disabled selected value>-- select destination
            --</option>
                <% for (Place place : places) { %>
                <option value="<%= place.getPlaceId() %>"><%=
place.getPlaceName() %></option>
                <% } %>
            </select>
        </p>
        <p>
            <label>No. of persons : </label> <input type="number"
name="numberOfPersons" required>
        </p>
        <input type="submit" value="Submit">
    </fieldset>
</form>
</body>
</html>

```

userinvaledcred.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
</style>
<meta charset="UTF-8">
<title>Invalid Credentials</title>
</head>
<body>

    <% @ include file="header.jsp"%>

    <h2>You have entered invalid credentials! <br><br>
    <a href="userlogin.jsp">Click here</a> to try again.</h2>
</body>
</html>

```

userlogin.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
.welcome {
    text-align: center;
    padding-bottom: 20px;
    color: #2095bf;
}
fieldset {
    text-align: center;
    border: 2px #2095bf solid;
}
legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}
.option {
    font-size: 12px;
    padding-top: 8px;
    color: #2095bf;
}
label {
    color: #2095bf;
}
</style>

<meta charset="UTF-8">
<title>User Login Page</title>
</head>
<body>

    <% @ include file="header.jsp"%>

    <div class="welcome">
        <h1>Fly Away</h1>
    </div>
    <form action="userlogincontroller " method="post" style = "width: 400px;
margin: 0px auto;">
        <fieldset>
```

```

        <legend>User Login</legend><br>
        <label>Username : </label><input type="text"
name="uUsername" required><br><br>
        <label>Password : </label><input type="password"
name="uPassword" required><br><br>
        <input type="submit" value="Submit">

        <div class="option">
            Not a registered user?<a href="userregister.jsp">Click
here</a>
        </div>
    </fieldset>
</form>

</body>
</html>

```

userregister.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"% >
<!DOCTYPE html>
<html>
<head>
<style>
form {
    padding-top: 50px;
}
fieldset {
    text-align: center;
    border: 2px #2095bf solid;
    overflow: auto;
}
legend {
    text-align: center;
    color: #2095bf;
    font-size: 20px;
    font-weight: 700;
}
label {
    color: #2095bf;
}
</style>
<meta charset="UTF-8">
<title>User Registration</title>
</head>

```

```

<body>
    <% @ include file="header.jsp"%>

    <form action="userregistercontroller" method="post" style = "width: 400px;
margin: 0px auto;">
        <fieldset>
            <legend>User Registration Form</legend><br>
            <label>Username : </label><input type="text"
name="uUsername" required><br><br>
            <label>Password : </label><input type="password"
name="uPassword" required><br><br>
            <input type="submit" value="Submit">
        </fieldset>
    </form>
</body>
</html>

```

userregistrationsuccessful.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
h2 {
    padding-left: 10px;
    padding-top: 20px;
    text-align: center;
}
</style>
<meta charset="UTF-8">
<title>Registration Successful!</title>
</head>
<body>

    <% @ include file="header.jsp"%>

    <h2>Your registration is successful. <br><br>
    <a href="userlogin.jsp">Click here</a> to login.</h2>

</body>
</html>

```

adminhome-styles.css

```
@charset "UTF-8";
```

```
.heading {
  text-align:center;
  color:#2095bf;
  margin-top: 30px;
  margin-bottom: 30px;
  font-size: 24px;
}

.column-2 {
  width: calc(32.1% - 40px);
  padding: 10px;
  margin: 20px;
  height: 120px;
}

.tiles{
  display: flex;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
  padding-left: 20px;
}

.option-1 {
  background-color: #2095bf;
}
.option-2 {
  background-color: #2095bf;
}
.option-3 {
  background-color: #2095bf;
}

.option-text {
  text-align:center;
  font-size:30px;
  color: white;
}

.option-1:hover {
  background-color:#98c1d9;
}
.option-2:hover {
  background-color: #98c1d9;
```

```
}  
.option-3:hover {  
    background-color:#98c1d9;  
}
```

userhome-styles.css

```
@charset "ISO-8859-1";
```

```
.heading {  
    text-align:center;  
    color:#2095bf;  
    margin-top: 30px;  
    margin-bottom: 30px;  
    font-size: 24px;  
}
```

```
form {  
    padding-top: 30px;  
}
```

```
fieldset {  
    text-align: center;  
    border: 2px #2095bf solid;  
    overflow: auto;  
}
```

```
legend {  
    text-align: center;  
    color: #2095bf;  
    font-size: 20px;  
    font-weight: 700;  
}
```

```
label {  
    display: inline-block;  
    width: 200px;  
    text-align: right;  
    font-weight: 600;  
    color: #2095bf;  
    margin-right: 5px;  
}
```

```
input[type="date"], input[type="number"], select {  
    width: 167px;  
}
```

```
select option[disabled], input[type="date"] {
```

```
text-align: center;
padding: 0;
}
```

7. Creating Controller java servlets

- Create a package 'mypackage.controllers' in Java Resources.
- Create java servlets AdminLoginController.java, AdminPasswordController.java, AdminRegisterController.java, AdminVerificationController.java, AirlineController.java, FlightController.java, FlightSearchController.java, PaymentController.java, PlaceController.java, UserDetailsController.java, UserLoginController.java and UserRegisterController.java.

AdminLoginController.java

```
package mypackage.controllers;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import mypackage.entities.Admin;
import mypackage.models.AdminLoginModel;
import mypackage.serviceimpl.AdminServiceImpl;
import mypackage.services.AdminService;

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

    private AdminService adminService = new AdminServiceImpl();
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        String aUsername = request.getParameter("aUsername");
        String aPassword = request.getParameter("aPassword");

        AdminLoginModel adminLoginModel = new
AdminLoginModel(aUsername, aPassword);
        adminLoginModel.setaUsername(aUsername);
        adminLoginModel.setaPassword(aPassword);

        Admin admin = adminService.getAdmin(adminLoginModel);
```



```

        if(admin != null)
        {
            HttpSession session = request.getSession();
            session.setAttribute("admin", admin);
            response.sendRedirect("adminchangepassword.jsp");
        }
        else
            response.sendRedirect("admininvalidcred.jsp");
    }
}

```

AdminPasswordController.java

```

package mypackage.controllers;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import mypackage.dao.AdminDao;
import mypackage.daoimpl.AdminDaoImpl;
import mypackage.entities.Admin;

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

    private AdminDao adminDao = new AdminDaoImpl();
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

        String action = request.getParameter("action");

        if (action != null) {
            if (action.equals("Yes")) {
                response.sendRedirect("newpassword.jsp");
            } else if (action.equals("No")) {
                response.sendRedirect("adminhome.jsp");
            } else if (action.equals("Submit")) {

                String newpassword = request.getParameter("newpassword");

                Admin admin = (Admin) request.getSession().getAttribute("admin");
            }
        }
    }
}

```

```

        admin.setaPassword(newpassword);
        adminDao.update(admin);

        response.sendRedirect("passwordchanged.jsp");
    }
}
}
}

```

AdminRegisterContoller.java

```

package mypackage.controllers;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import mypackage.models.AdminModel;
import mypackage.serviceimpl.AdminServiceImpl;
import mypackage.services.AdminService;

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

    private AdminService adminService = new AdminServiceImpl();
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        String aUsername = request.getParameter("aUsername");
        String aPassword = request.getParameter("aPassword");

        AdminModel adminModel = new AdminModel(aUsername, aPassword);
        adminService.register(adminModel);

        response.sendRedirect("adminregistrationsuccessful.jsp");
    }
}

```

AdminVerificationController.java

```

package mypackage.controllers;

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("/adminverificationcontroller")
public class AdminVerificationController extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

        String code = request.getParameter("code");

        if(code.equals("C3H9S4"))
        {
            response.sendRedirect("adminregister.jsp");
        }
        else
            response.sendRedirect("invalidcode.jsp");
    }
}

```

AirlineController.java

```

package mypackage.controllers;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import mypackage.entities.Admin;
import mypackage.models.AirlineModel;
import mypackage.serviceimpl.AirlineServiceImpl;
import mypackage.services.AirlineService;

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

    private AirlineService airlineService = new AirlineServiceImpl();
}

```

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

```
    String airlineName = request.getParameter("airlineName");
```

```
    AirlineModel airlineModel = new AirlineModel(airlineName);
```

```
    HttpSession session = request.getSession();
```

```
        Admin admin = (Admin) session.getAttribute("admin");
```

```
        airlineService.addAirline(admin, airlineModel);
```

```
        response.sendRedirect("airlineslist.jsp");
```

```
    }
```

```
}
```

FlightController.java

```
package mypackage.controllers;
```

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

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

```
import java.time.format.DateTimeFormatter;
```

```
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 mypackage.entities.Admin;
```

```
import mypackage.models.FlightModel;
```

```
import mypackage.serviceimpl.FlightServiceImpl;
```

```
import mypackage.services.FlightService;
```

```
@WebServlet("/flightcontroller")
```

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

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

```
    private FlightService flightService = new FlightServiceImpl();
```

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

```
        int airlineId = Integer.parseInt(request.getParameter("airlineId"));
```

```
        int sourceId = Integer.parseInt(request.getParameter("sourceId"));
```

```

        int destinationId = Integer.parseInt(request.getParameter("destinationId"));
        String departureDateTimeString =
request.getParameter("departureDateTime");
        double ticketPrice =
Double.parseDouble(request.getParameter("ticketPrice"));
        int availableSeats =
Integer.parseInt(request.getParameter("availableSeats"));

        LocalDateTime departureDateTime =
LocalDateTime.parse(departureDateTimeString,
DateTimeFormatter.ISO_LOCAL_DATE_TIME);

        FlightModel flightModel = new FlightModel(airlineId, sourceId,
destinationId, departureDateTime, ticketPrice, availableSeats);
        HttpSession session = request.getSession();
        Admin admin = (Admin) session.getAttribute("admin");
        flightService.addFlight(admin, flightModel);

        response.sendRedirect("flightslist.jsp");
    }
}

```

FlightSearchController.java

```

package mypackage.controllers;

import java.io.IOException;
import java.time.LocalDate;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import mypackage.entities.Flight;
import mypackage.serviceimpl.FlightServiceImpl;
import mypackage.services.FlightService;

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

    private FlightService flightService = new FlightServiceImpl();

```

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

```
    LocalDate dateOfTravel =
LocalDate.parse(request.getParameter("dateOfTravel"));
    int sourceId = Integer.parseInt(request.getParameter("sourceId"));
    int destinationId = Integer.parseInt(request.getParameter("destinationId"));
    int numberOfPersons =
Integer.parseInt(request.getParameter("numberOfPersons"));
```

```
    HttpSession session = request.getSession();
    session.setAttribute("numberOfPersons", numberOfPersons);
```

```
    List<Flight> flights = flightService.getFlightsBySearchCriteria(dateOfTravel,
sourceId, destinationId, numberOfPersons);
```

```
    request.setAttribute("flights", flights);
```

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

PaymentController.java

```
package mypackage.controllers;
```

```
import java.io.IOException;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
```

```
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 mypackage.dao.BookingDao;
import mypackage.dao.FlightDao;
import mypackage.daoimpl.BookingDaoImpl;
import mypackage.daoimpl.FlightDaoImpl;
import mypackage.entities.Booking;
import mypackage.entities.Flight;
import mypackage.entities.User;
```

```
@WebServlet("/paymentcontroller")
public class PaymentController extends HttpServlet {
```

```

private static final long serialVersionUID = 1L;

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

    String bookingId = request.getParameter("bookingId");
    String cardNumber = request.getParameter("cardNumber");
    String cardHolderName = request.getParameter("cardHolderName");
    int expirationMonth =
Integer.parseInt(request.getParameter("expirationMonth"));
    int expirationYear =
Integer.parseInt(request.getParameter("expirationYear"));
    int cvv = Integer.parseInt(request.getParameter("cvv"));

    boolean paymentSuccess = processPayment(cardNumber,
cardHolderName, expirationMonth, expirationYear, cvv);

    if (paymentSuccess) {
        int userId = (int) request.getSession().getAttribute("userId");
        int flightId = (int) request.getSession().getAttribute("flightId");
        int totalTickets = (int)
request.getSession().getAttribute("numberOfPersons");
        double totalPrice = (double)
request.getSession().getAttribute("totalPrice");
        String paymentStatus = "Paid"; // Assuming payment was
successful

        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-
MM-yyyy HH:mm");
        LocalDateTime bookingDateTime = LocalDateTime.now();
        String formattedDateTime = bookingDateTime.format(formatter);
        LocalDateTime parsedDateTime = LocalDateTime.parse(formattedDateTime,
formatter);

        try {
            Booking booking = new Booking();
            booking.setUser(new User(userId));
            booking.setFlight(new Flight(flightId));
            booking.setBookingDateTime(parsedDateTime);
            booking.setTotalTickets(totalTickets);
            booking.setTotalPrice(totalPrice);
            booking.setPaymentStatus(paymentStatus);

            BookingDao bookingDao = new BookingDaoImpl();
            bookingDao.add(booking);

```

```

        HttpSession session = request.getSession();
        session.setAttribute("booking", booking);

        FlightDao flightDao = new FlightDaoImpl();
        Flight flight = flightDao.getFlightById(flightId);
        int availableSeats = flight.getAvailableSeats() -
totalTickets;

        flight.setAvailableSeats(availableSeats);
        flightDao.update(flight);
    } catch (Exception e) {
        e.printStackTrace();
    }

    response.sendRedirect("bookingdetails.jsp?bookingId=" +
bookingId);
    } else {
        response.sendRedirect("paymentfailed.jsp");
    }
}

private boolean processPayment(String cardNumber, String cardHolderName, int
expirationMonth, int expirationYear,
    int cvv) {

    return true;
}
}

```

PlaceController.java

```

package mypackage.controllers;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import mypackage.entities.Admin;
import mypackage.models.PlaceModel;
import mypackage.serviceimpl.PlaceServiceImpl;
import mypackage.services.PlaceService;

```



```

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

    private PlaceService placeService = new PlaceServiceImpl();

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

        String placeName = request.getParameter("placeName");

        PlaceModel placeModel = new PlaceModel(placeName);
        HttpSession session = request.getSession();
        Admin admin = (Admin) session.getAttribute("admin");
        placeService.addPlace(admin, placeModel);

        response.sendRedirect("placeslist.jsp");
    }
}

```

UserDetailsController.java

```

package mypackage.controllers;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import mypackage.dao.FlightDao;
import mypackage.dao.UserDao;
import mypackage.daoimpl.FlightDaoImpl;
import mypackage.daoimpl.UserDaoImpl;
import mypackage.entities.Flight;
import mypackage.entities.User;

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

    private UserDao userDao = new UserDaoImpl();
    private FlightDao flightDao = new FlightDaoImpl();
}

```

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

```
    String uEmail = request.getParameter("uEmail");
    String uFirstName = request.getParameter("uFirstName");
    String uLastName = request.getParameter("uLastName");
    String uPhoneNumber = request.getParameter("uPhoneNumber");
    String uAddress = request.getParameter("uAddress");
```

```
    User user = (User) request.getSession().getAttribute("user");
```

```
    user.setuEmail(uEmail);
    user.setuFirstName(uFirstName);
    user.setuLastName(uLastName);
    user.setuPhoneNumber(uPhoneNumber);
    user.setuAddress(uAddress);
```

```
    try {
        userDao.update(user);
    } catch (Exception e) {
        e.printStackTrace();
    }
```

```
    calculateTicketPrice(request);
```

```
    int flightId = Integer.parseInt(request.getParameter("flightId"));
```

```
    Flight flight = flightDao.getFlightById(flightId);
    HttpSession session = request.getSession();
```

```
    session.setAttribute("flightId", flightId);
    session.setAttribute("flight", flight);
```

```
    response.sendRedirect("flightdetails.jsp");
}
```

```
private void calculateTicketPrice(HttpServletRequest request) {
    int numberOfPersons = (int)
request.getSession().getAttribute("numberOfPersons");
    double ticketPrice =
Double.parseDouble(request.getParameter("ticketPrice"));
```

```
    double totalPrice = numberOfPersons * ticketPrice;
    HttpSession session = request.getSession();
    session.setAttribute("totalPrice", totalPrice);
```

```
}
```

```
}
```

UserLoginController.java

```
package mypackage.controllers;
```

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

```
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
```

```
import mypackage.entities.User;
```

```
import mypackage.models.UserLoginModel;
```

```
import mypackage.serviceimpl.UserServiceImpl;
```

```
import mypackage.services.UserService;
```

```
@WebServlet("/userlogincontroller")
```

```
public class UserLoginController extends HttpServlet {
```

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

```
    private UserService userService = new UserServiceImpl();
```

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

```
        String uUsername = request.getParameter("uUsername");
```

```
        String uPassword = request.getParameter("uPassword");
```

```
        request.getSession().setAttribute("uUsername", uUsername);
```

```
        UserLoginModel userLoginModel = new UserLoginModel();
```

```
        userLoginModel.setuUsername(uUsername);
```

```
        userLoginModel.setuPassword(uPassword);
```

```
        User user = userService.getUser(userLoginModel);
```

```
        if(user != null)
```

```
        {
```

```
            HttpSession session = request.getSession();
```

```
            session.setAttribute("user", user);
```

```
            session.setAttribute("userId", user.getUserId());
```

```
            response.sendRedirect("userhome.jsp");
```

```
        }
```

```

        else
            response.sendRedirect("userinvalidcred.jsp");
    }
}

UserRegisterContoller.java
package mypackage.controllers;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import mypackage.models.UserModel;
import mypackage.serviceimpl.UserServiceImpl;
import mypackage.services.UserService;

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

    private UserService userService = new UserServiceImpl();

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

        String uUsername = request.getParameter("uUsername");
        String uPassword = request.getParameter("uPassword");

        UserModel userModel = new UserModel();
        userModel.setuUsername(uUsername);
        userModel.setuPassword(uPassword);
        userService.register(userModel);

        response.sendRedirect("userregistrationsuccessful.jsp");
    }
}

```

8. Creating Model classes

- Create a package 'mypackage.models' in Java Resources.
- Create java classes AdminLoginModel.java, AdminModel.java, AirlineModel.java, FlightModel.java, UserLoginModel.java, PlaceModel.java and UserModel.java.

AdminLoginModel.java

```
package mypackage.models;
```

```
public class AdminLoginModel {  
  
    private String aUsername;  
    private String aPassword;  
  
    public AdminLoginModel() {  
    }  
  
    public AdminLoginModel(String aUsername, String aPassword) {  
        super();  
        this.aUsername = aUsername;  
        this.aPassword = aPassword;  
    }  
  
    public String getaUsername() {  
        return aUsername;  
    }  
  
    public void setaUsername(String aUsername) {  
        this.aUsername = aUsername;  
    }  
  
    public String getaPassword() {  
        return aPassword;  
    }  
  
    public void setaPassword(String aPassword) {  
        this.aPassword = aPassword;  
    }  
}
```

AdminModel.java

```
package mypackage.models;
```

```
public class AdminModel {
```

```
private String ausername;
private String apassword;

public AdminModel() {
}

public AdminModel(String ausername, String apassword) {
    super();
    this.ausername = ausername;
    this.apassword = apassword;
}

// Getters and Setters
public String getAusername() {
    return ausername;
}

public void setAusername(String ausername) {
    this.ausername = ausername;
}

public String getApassword() {
    return apassword;
}

public void setApassword(String apassword) {
    this.apassword = apassword;
}
}
```

AirlineModel.java

```
package mypackage.models;

public class AirlineModel {

    private String airlineName;

    public AirlineModel() {
    }

    public AirlineModel(String airlineName) {
        super();
        this.airlineName = airlineName;
    }
}
```

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

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

FlightModel.java

```
package mypackage.models;

import java.time.LocalDateTime;

public class FlightModel {

    private int airlineId;
    private int sourceId;
    private int destinationId;
    private LocalDateTime departureDateTime;
    private double ticketPrice;
    private int availableSeats;

    public FlightModel() {
    }

    public FlightModel(int airlineId, int sourceId, int destinationId, LocalDateTime
departureDateTime,
                        double ticketPrice, int availableSeats) {
        super();
        this.airlineId = airlineId;
        this.sourceId = sourceId;
        this.destinationId = destinationId;
        this.departureDateTime = departureDateTime;
        this.ticketPrice = ticketPrice;
        this.availableSeats = availableSeats;
    }

    public int getAirlineId() {
        return airlineId;
    }

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

```
    }

    public int getSourceId() {
        return sourceId;
    }

    public void setSourceId(int sourceId) {
        this.sourceId = sourceId;
    }

    public int getDestinationId() {
        return destinationId;
    }

    public void setDestinationId(int destinationId) {
        this.destinationId = destinationId;
    }

    public LocalDateTime getDepartureDateTime() {
        return departureDateTime;
    }

    public void setDepartureDateTime(LocalDateTime departureDateTime) {
        this.departureDateTime = departureDateTime;
    }

    public double getTicketPrice() {
        return ticketPrice;
    }

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

    public int getAvailableSeats() {
        return availableSeats;
    }

    public void setAvailableSeats(int availableSeats) {
        this.availableSeats = availableSeats;
    }
}
```

UserLoginModel.java

```
package mypackage.models;
```



```
public class UserLoginModel {  
  
    private String uUsername;  
    private String uPassword;  
  
    public UserLoginModel() {  
    }  
  
    public UserLoginModel(String uUsername, String uPassword) {  
        super();  
        this.uUsername = uUsername;  
        this.uPassword = uPassword;  
    }  
  
    public String getuUsername() {  
        return uUsername;  
    }  
  
    public void setuUsername(String uUsername) {  
        this.uUsername = uUsername;  
    }  
  
    public String getuPassword() {  
        return uPassword;  
    }  
  
    public void setuPassword(String uPassword) {  
        this.uPassword = uPassword;  
    }  
}
```

PlaceModel.java

```
package mypackage.models;  
  
public class PlaceModel {  
  
    private String placeName;  
  
    public PlaceModel() {  
    }  
  
    public PlaceModel(String placeName) {  
        super();  
        this.placeName = placeName;  
    }  
}
```

```
    }

    public String getPlaceName() {
        return placeName;
    }

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

UserModel.java

```
package mypackage.models;

public class UserModel {

    private String uUsername;
    private String uPassword;

    public UserModel() {
    }

    public UserModel(String uUsername, String uPassword) {
        super();
        this.uUsername = uUsername;
        this.uPassword = uPassword;
    }

    public String getuUsername() {
        return uUsername;
    }

    public void setuUsername(String uUsername) {
        this.uUsername = uUsername;
    }

    public String getuPassword() {
        return uPassword;
    }

    public void setuPassword(String uPassword) {
        this.uPassword = uPassword;
    }
}
```

9. Creating Service interfaces

- Create a package 'mypackage.services' in Java Resources.
- Create interfaces AdminService.java, AirlineService.java, FlightService.java, PlaceService.java and UserService.java.
- Declare methods that define the operations to be performed on the corresponding models.

AdminService.java

```
package mypackage.services;

import mypackage.entities.Admin;
import mypackage.models.AdminLoginModel;
import mypackage.models.AdminModel;

public interface AdminService {

    void register(AdminModel adminModel);
    Admin getAdmin(AdminLoginModel adminLoginModel);
}
```

AirlineService.java

```
package mypackage.services;

import mypackage.entities.Admin;
import mypackage.models.AirlineModel;

public interface AirlineService {

    void addAirline(Admin admin, AirlineModel airlineModel);
}
```

FlightService.java

```
package mypackage.services;

import java.time.LocalDate;
import java.util.List;

import mypackage.entities.Admin;
import mypackage.entities.Flight;
import mypackage.models.FlightModel;

public interface FlightService {

    void addFlight(Admin admin, FlightModel flightModel);
}
```

```
List<Flight> getFlightsBySearchCriteria(LocalDate dateOfTravel, int sourceId,
int destinationId, int numberOfPersons);
}
```

PlaceService.java

```
package mypackage.services;

import mypackage.entities.Admin;
import mypackage.models.PlaceModel;

public interface PlaceService {

    void addPlace(Admin admin, PlaceModel placeModel);
}
```

UserService.java

```
package mypackage.services;

import mypackage.entities.User;
import mypackage.models.UserLoginModel;
import mypackage.models.UserModel;

public interface UserService {

    void register(UserModel userModel);
    User getUser(UserLoginModel userLoginModel);
}
```

10. Creating ServiceImpl classes

- Create a package 'mypackage.serviceimpl' in Java Resources.
- Create java classes AdminServiceImpl.java, AirlineServiceImpl.java, FlightServiceImpl.java, PlaceServiceImpl.java and UserServiceImpl.java.
- Implement the methods declared in each service interface.

AdminServiceImpl.java

```
package mypackage.serviceimpl;

import mypackage.models.AdminLoginModel;
import mypackage.models.AdminModel;
import mypackage.dao.AdminDao;
import mypackage.daoimpl.AdminDaoImpl;
import mypackage.entities.Admin;
import mypackage.services.AdminService;
```

```

public class AdminServiceImpl implements AdminService{

    private AdminDao adminDao = new AdminDaoImpl();
    @Override
    public void register(AdminModel adminModel) {

        Admin admin = new Admin();
        admin.setAUsername(adminModel.getAusername());
        admin.setAPassword(adminModel.getApassword());
        adminDao.insert(admin);

    }

    @Override
    public Admin getAdmin(AdminLoginModel adminLoginModel) {

        return adminDao.getAdmin(adminLoginModel.getAUsername(),
adminLoginModel.getAPassword());
    }

}

```

AirlineServiceImpl.java

```

package mypackage.serviceimpl;

import mypackage.dao.AdminDao;
import mypackage.dao.AirlineDao;
import mypackage.daoimpl.AdminDaoImpl;
import mypackage.daoimpl.AirlineDaoImpl;
import mypackage.entities.Admin;
import mypackage.entities.Airline;
import mypackage.models.AirlineModel;
import mypackage.services.AirlineService;

public class AirlineServiceImpl implements AirlineService{

    private AirlineDao airlineDao = new AirlineDaoImpl();
    private AdminDao adminDao = new AdminDaoImpl();

    @Override
    public void addAirline(Admin admin, AirlineModel airlineModel) {

        try {

```

```
        Admin adm = adminDao.getAdmin(admin.getaUsername(),
admin.getaPassword());
```

```
        Airline airline = new Airline();
        airline.setAirlineName(airlineModel.getAirlineName());
        airline.setAdmin(admin);
```

```
        adm.addAirline(airline);
        airlineDao.add(airline);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
}
```

FlightServiceImpl.java

```
package mypackage.serviceimpl;
```

```
import java.time.LocalDate;
import java.util.List;
```

```
import mypackage.dao.AdminDao;
import mypackage.dao.AirlineDao;
import mypackage.dao.FlightDao;
import mypackage.dao.PlaceDao;
import mypackage.daoimpl.AdminDaoImpl;
import mypackage.daoimpl.AirlineDaoImpl;
import mypackage.daoimpl.FlightDaoImpl;
import mypackage.daoimpl.PlaceDaoImpl;
import mypackage.entities.Admin;
import mypackage.entities.Airline;
import mypackage.entities.Flight;
import mypackage.entities.Place;
import mypackage.models.FlightModel;
import mypackage.services.FlightService;
```

```
public class FlightServiceImpl implements FlightService {
```

```
    private FlightDao flightDao = new FlightDaoImpl();
    private AdminDao adminDao = new AdminDaoImpl();
    private AirlineDao airlineDao = new AirlineDaoImpl();
    private PlaceDao placeDao = new PlaceDaoImpl();
```

```
    @Override
```

```

    public void addFlight(Admin admin, FlightModel flightModel) {

        try {
            Admin adm = adminDao.getAdmin(admin.getUsername(),
admin.getUsername());
            Airline airline =
airlineDao.getAirlineById(flightModel.getAirlineId());
            Place source = placeDao.getPlaceById(flightModel.getSourceId());
            Place destination =
placeDao.getPlaceById(flightModel.getDestinationId());

            Flight flight = new Flight();
            flight.setAirline(airline);
            flight.setSourcePlace(source);
            flight.setDestinationPlace(destination);

            flight.setDepartureDateTime(flightModel.getDepartureDateTime());
            flight.setTicketPrice(flightModel.getTicketPrice());
            flight.setAvailableSeats(flightModel.getAvailableSeats());
            flight.setAdmin(adm);

            adm.addFlight(flight);
            flightDao.add(flight);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    public List<Flight> getFlightsBySearchCriteria(LocalDate dateOfTravel, int
sourceId, int destinationId,
        int numberOfPersons) {

        return flightDao.getFlightsBySearchCriteria(dateOfTravel,
sourceId, destinationId, numberOfPersons);
    }
}

```

PlaceServiceImpl.java

```

package mypackage.serviceimpl;

import mypackage.dao.AdminDao;
import mypackage.dao.PlaceDao;
import mypackage.daoimpl.AdminDaoImpl;
import mypackage.daoimpl.PlaceDaoImpl;

```

```

import mypackage.entities.Admin;
import mypackage.entities.Place;
import mypackage.models.PlaceModel;
import mypackage.services.PlaceService;

public class PlaceServiceImpl implements PlaceService{

    private PlaceDao placeDao = new PlaceDaoImpl();
    private AdminDao adminDao = new AdminDaoImpl();

    @Override
    public void addPlace(Admin admin, PlaceModel placeModel) {

        try {
            Admin adm = adminDao.getAdmin(admin.getUsername(),
admin.getaPassword());

            Place place = new Place();
            place.setPlaceName(placeModel.getPlaceName());
            place.setAdmin(admin);

            adm.addPlace(place);
            placeDao.add(place);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

UserServiceImpl.java

```

package mypackage.serviceimpl;

import mypackage.dao.UserDao;
import mypackage.daoimpl.UserDaoImpl;
import mypackage.entities.User;
import mypackage.models.UserLoginModel;
import mypackage.models.UserModel;
import mypackage.services.UserService;

public class UserServiceImpl implements UserService {

    private UserDao userDao = new UserDaoImpl();
    @Override
    public void register(UserModel userModel) {

```



```

        User user = new User();
        user.setUUsername(userModel.getUUsername());
        user.setUPassword(userModel.getUPassword());
        userDao.insert(user);
    }

    @Override
    public User getUser(UserLoginModel userLoginModel) {

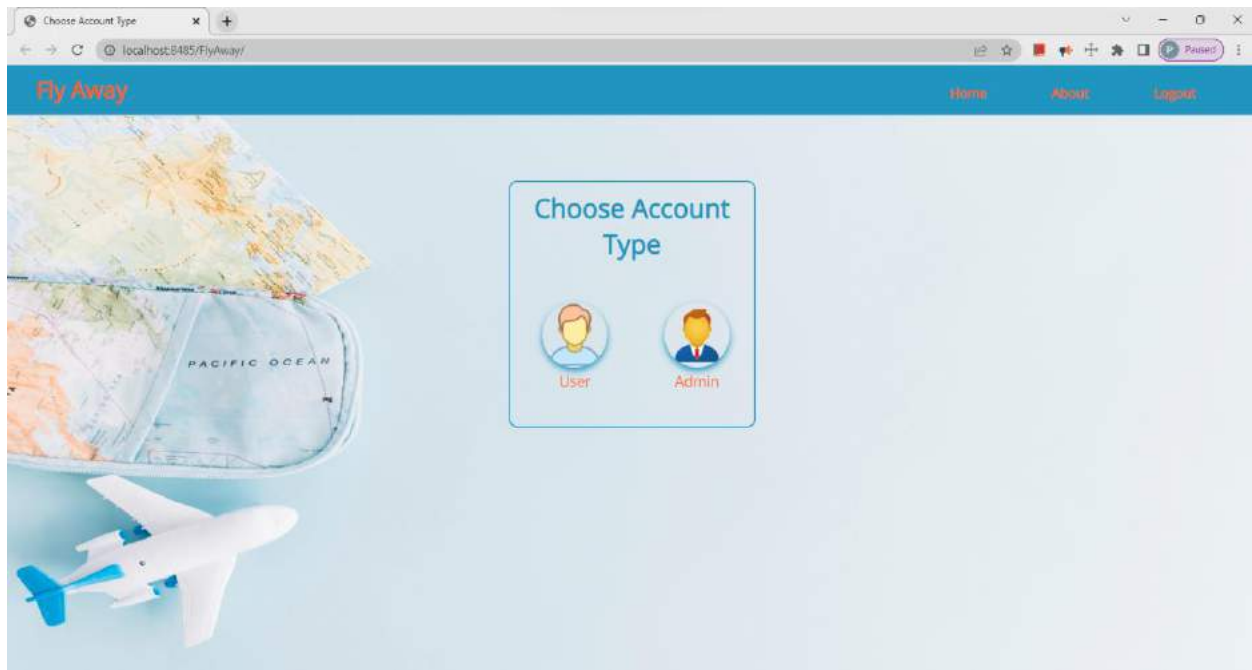
        return userDao.getUser(userLoginModel.getUUsername(),
            userLoginModel.getUPassword());
    }
}

```

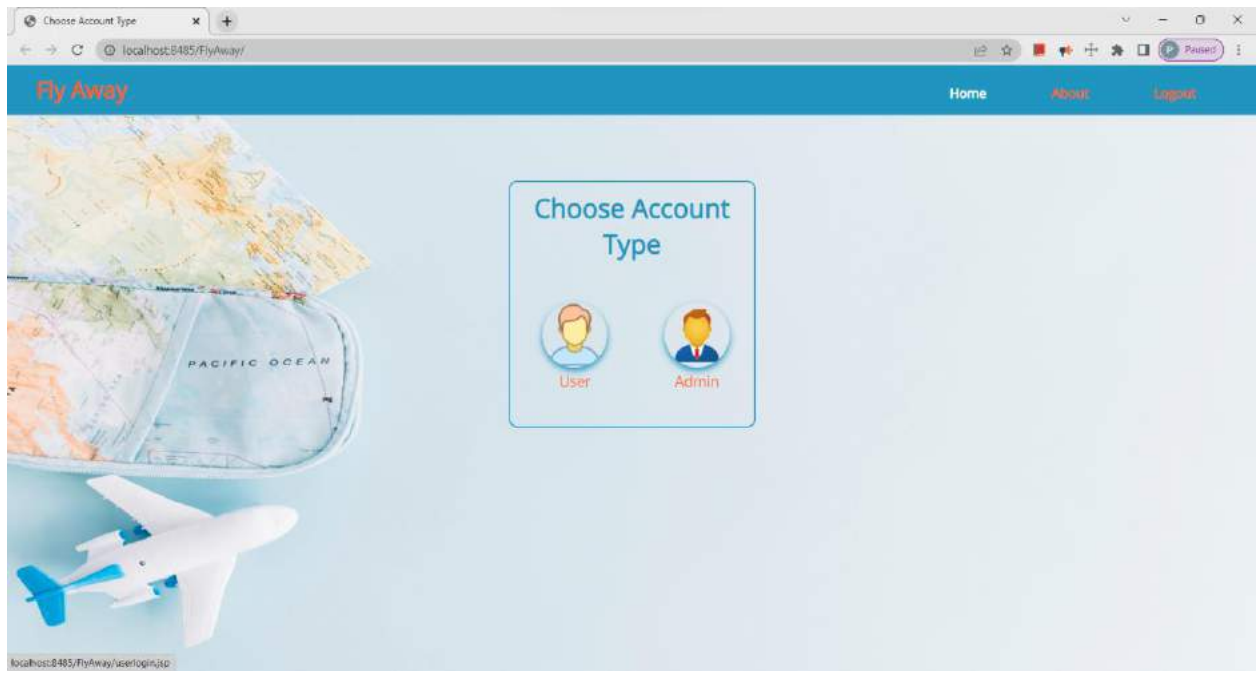
11. Testing and recording the output:

- Right-click on Project → Hover on 'Run as' → Click Run on Server.

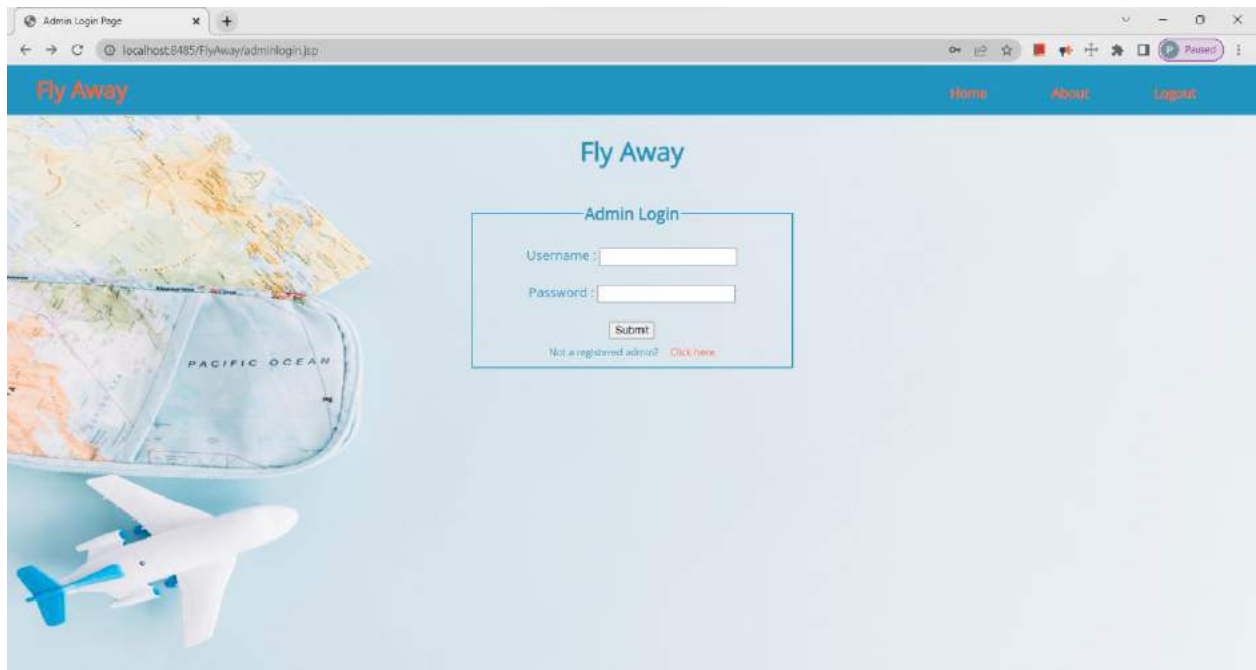
Output1 : Index Page



Output2 : On hover



Output3 : Admin Login Page



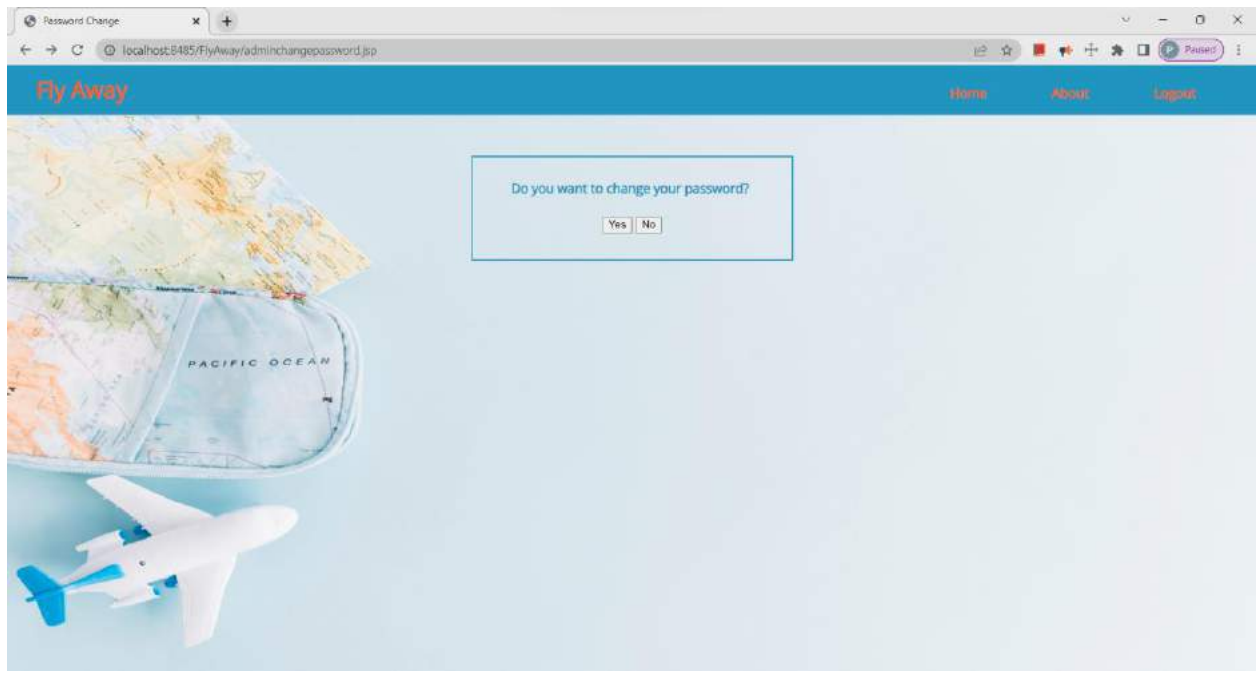
Output4 : Admin Verification Page

The screenshot shows a web browser window with the title 'Admin Verification' and the URL 'localhost:8485/FlyAway/admincode.jsp'. The page features a blue header with the 'Fly Away' logo and navigation links for 'Home', 'About', and 'Logout'. The main content area has a light blue background with a map of the Pacific Ocean and a white airplane. A central form titled 'Admin Verification' contains the following text: 'Please enter a unique 6-digit alphanumeric verification code to confirm your eligibility for registering as an admin'. Below this is a text input field labeled 'Enter Code :', a 'Submit' button, and a link that says 'Not an admin? Click here'.

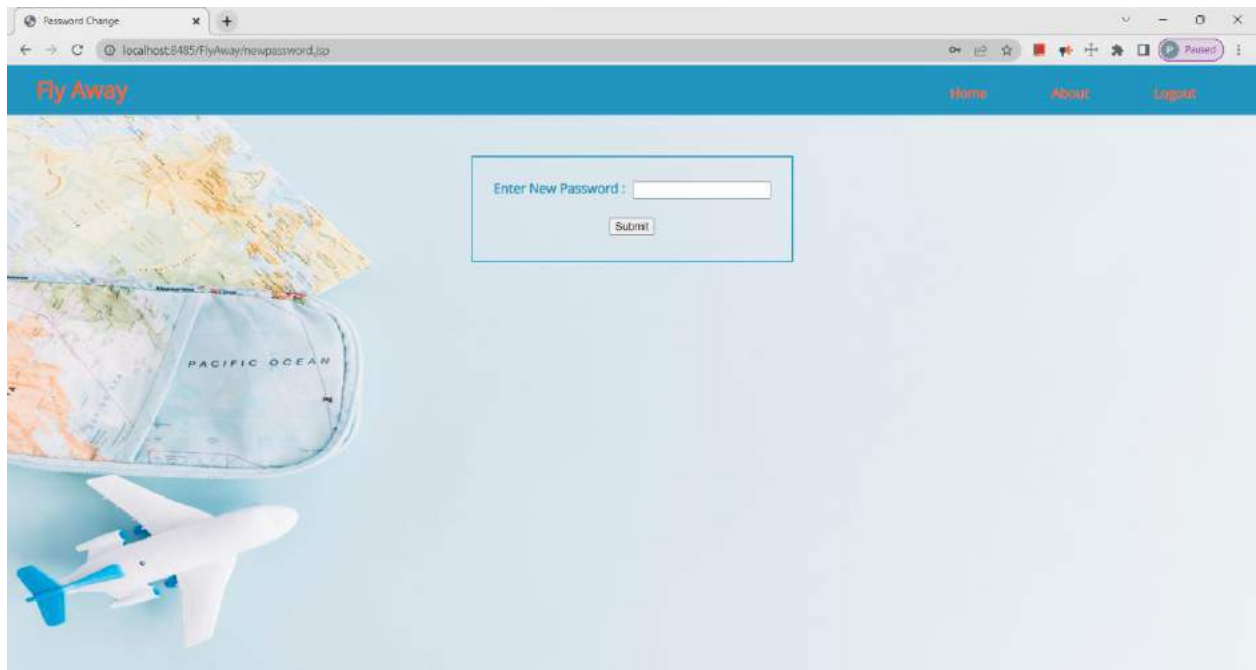
Output5 : Admin Registration Page

The screenshot shows a web browser window with the title 'Admin Registration' and the URL 'localhost:8485/FlyAway/adminregister.jsp'. The page features a blue header with the 'Fly Away' logo and navigation links for 'Home', 'About', and 'Logout'. The main content area has a light blue background with a map of the Pacific Ocean and a white airplane. A central form titled 'Admin Registration Form' contains the following text: 'Username :', 'Password :', and a 'Submit' button.

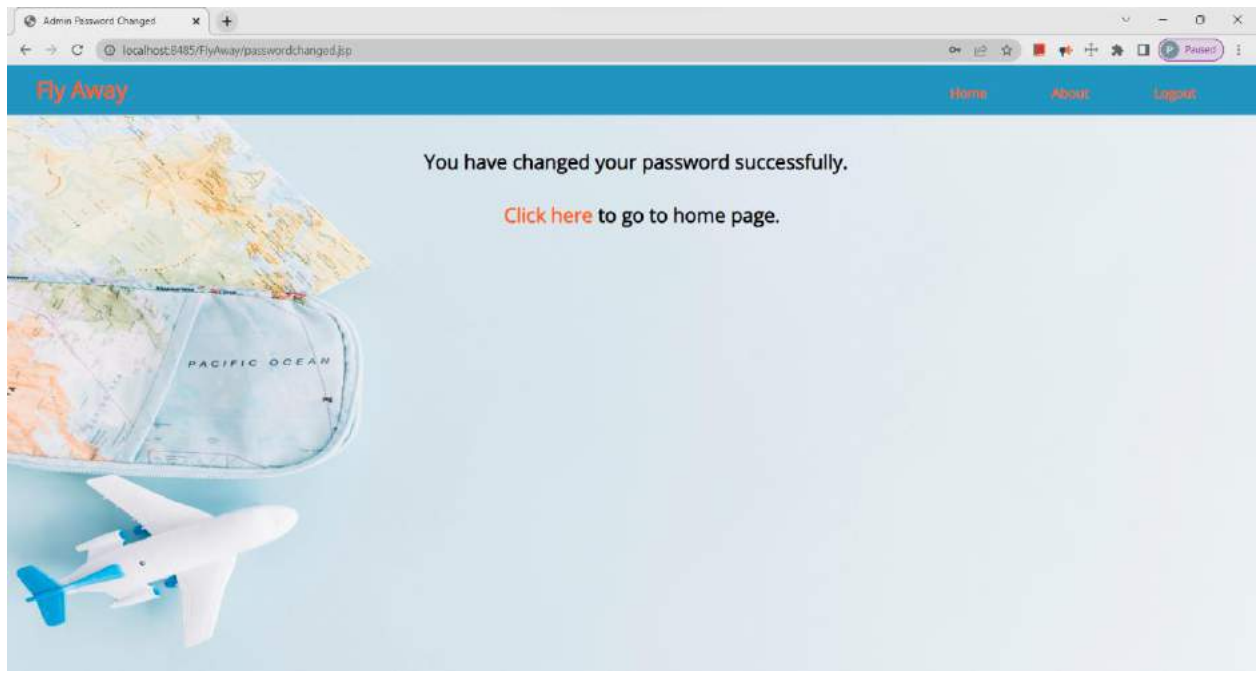
Output6 : Admin Change Password Page



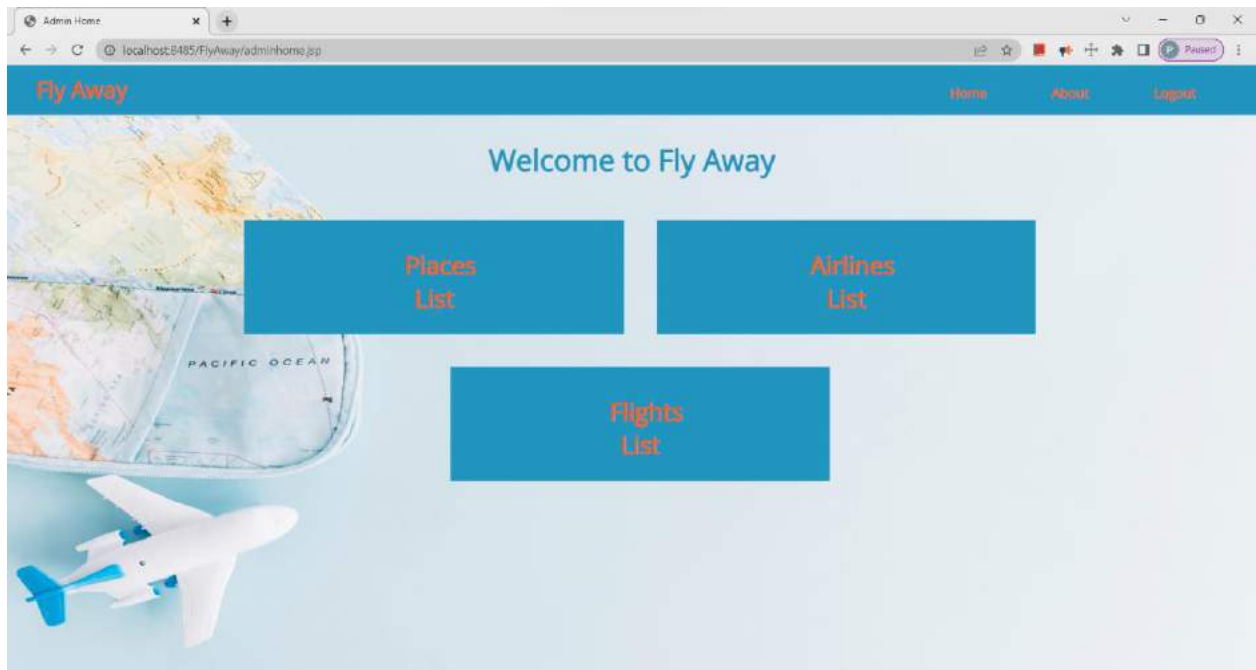
Output7 : New Password Page



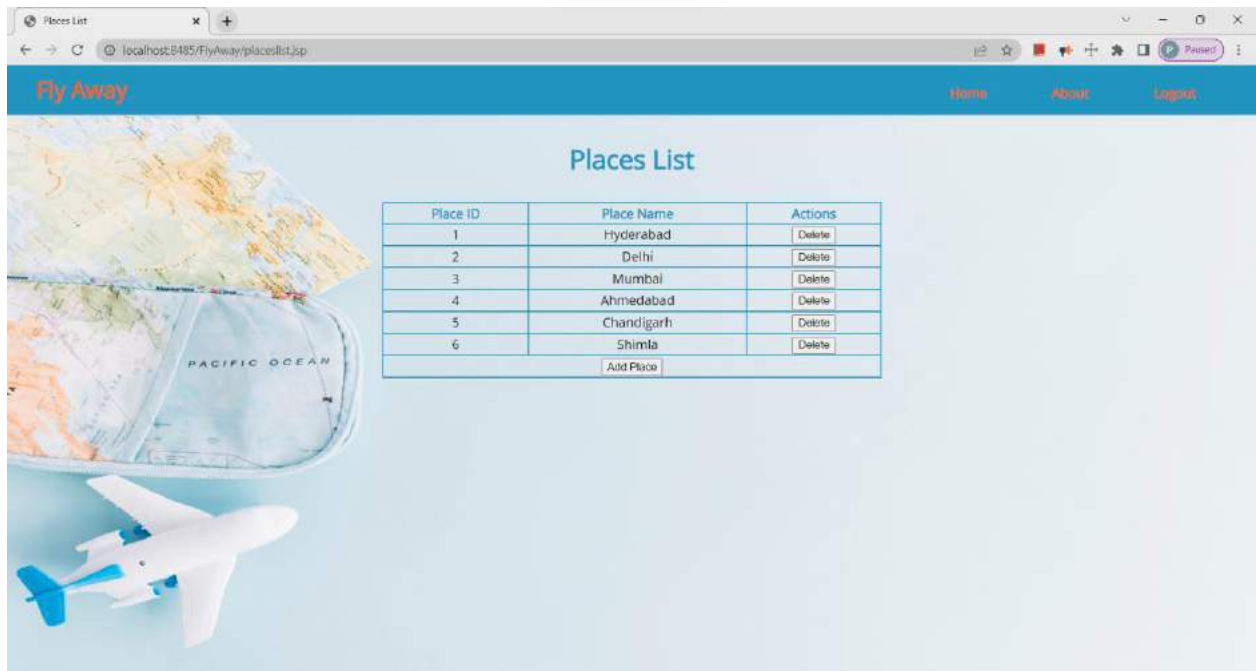
Output8 : Password Changed Page



Output9 : Admin Home Page



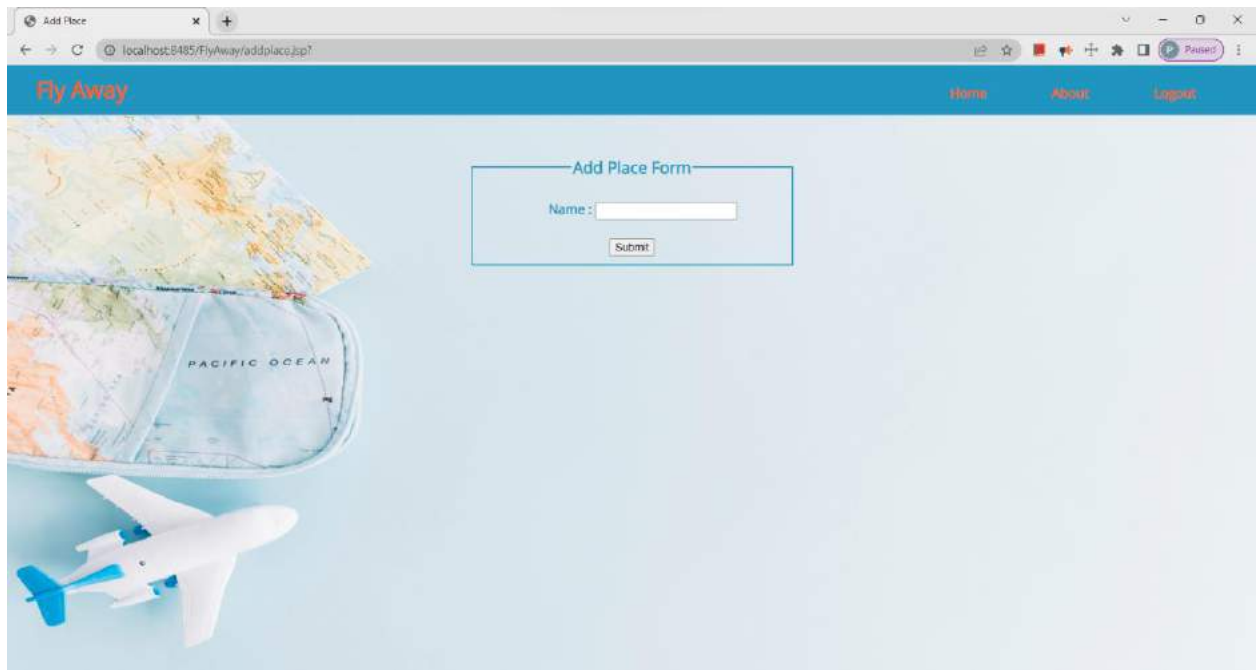
Output10 : Places List Page



Places List

Place ID	Place Name	Actions
1	Hyderabad	Delete
2	Delhi	Delete
3	Mumbai	Delete
4	Ahmedabad	Delete
5	Chandigarh	Delete
6	Shimla	Delete
Add Place		

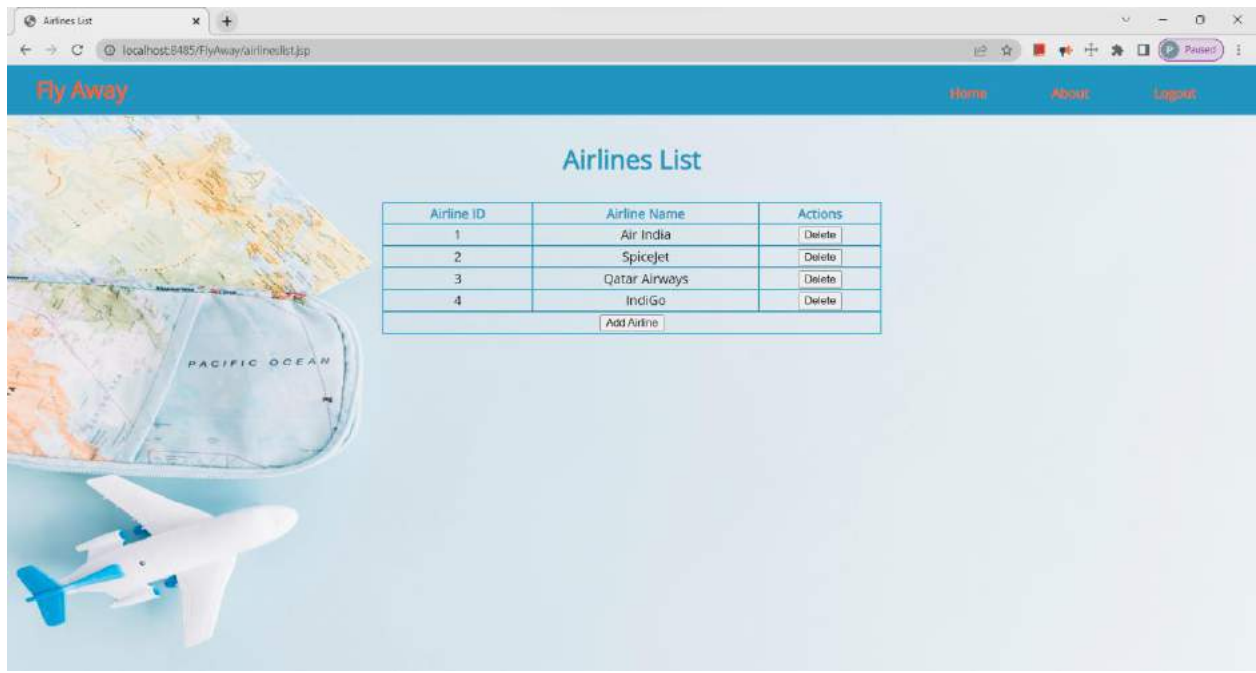
Output11 : Add Place Form



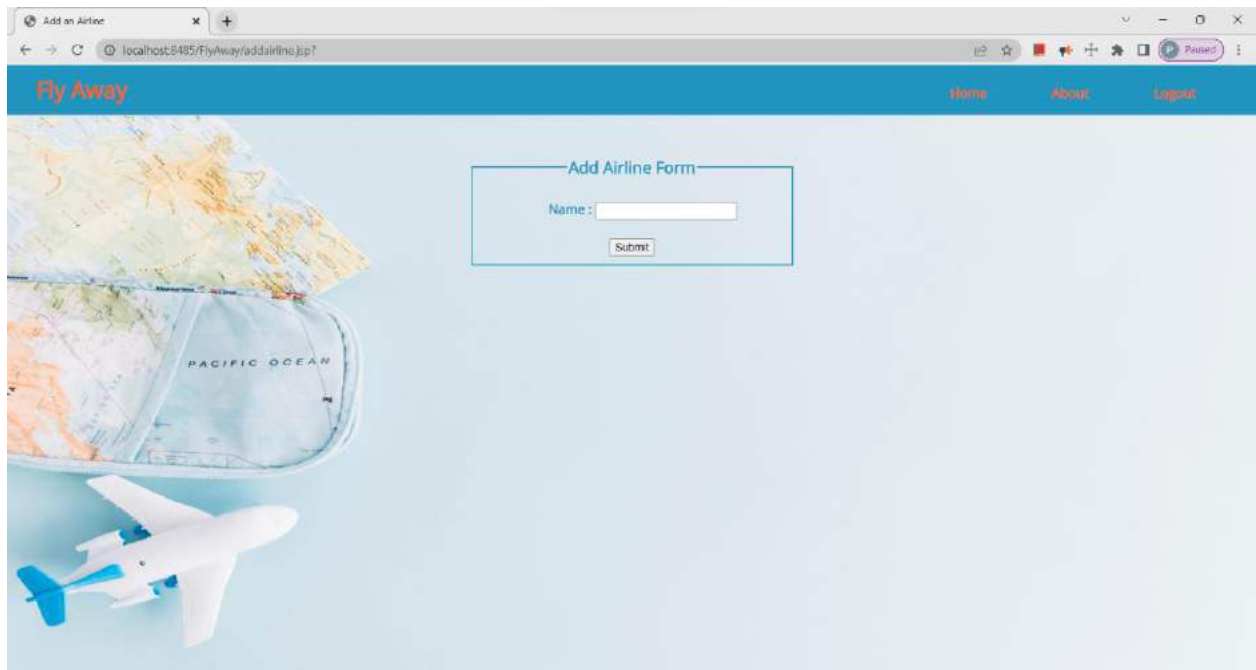
Add Place Form

Name:

Output12 : Airlines List Page



Output13 : Add Airline Form



Output14 : Flights List Page

Flights List

Flight ID	Airline Name	Source	Destination	Departure Date & Time	Ticket Price	Available Seats	Actions
1	Air India	Hyderabad	Delhi	20-07-2023 22:58	7000.0	0	Delete
2	SpiceJet	Mumbai	Chandigarh	20-07-2023 22:59	6000.0	75	Delete
3	IndiGo	Hyderabad	Delhi	19-07-2023 10:40	5500.0	150	Delete
4	IndiGo	Ahmedabad	Shimla	28-07-2023 21:00	4050.0	50	Delete
5	SpiceJet	Hyderabad	Delhi	20-07-2023 23:01	6000.0	33	Delete

[Add Flight](#)

Output15 : Add Flight Form

Add Flight Form

Select Airline :

Select Source :

Select Destination :

Departure Date and Time :

Ticket Price :

Available Seats :

[Submit](#)

Output16 : User Login Page

The screenshot shows a web browser window with the title 'User Login Page' and the URL 'localhost:8485/FlyAway/userlogin.jsp'. The page features a blue header with the 'Fly Away' logo and navigation links for 'Home', 'About', and 'Logout'. The main content area has a light blue background with a map of the Pacific Ocean and a white airplane. A 'User Login' form is centered on the page, containing fields for 'Username' and 'Password', a 'Submit' button, and a link for 'Click here' for new users.

Fly Away

Home About Logout

Fly Away

User Login

Username:

Password:

Submit

Not a registered user? [Click here](#)

Output17 : User Registration Page

The screenshot shows a web browser window with the title 'User Registration' and the URL 'localhost:8485/FlyAway/userregister.jsp'. The page features a blue header with the 'Fly Away' logo and navigation links for 'Home', 'About', and 'Logout'. The main content area has a light blue background with a map of the Pacific Ocean and a white airplane. A 'User Registration Form' is centered on the page, containing fields for 'Username' and 'Password', a 'Submit' button, and a link for 'Click here' for new users.

Fly Away

Home About Logout

Fly Away

User Registration Form

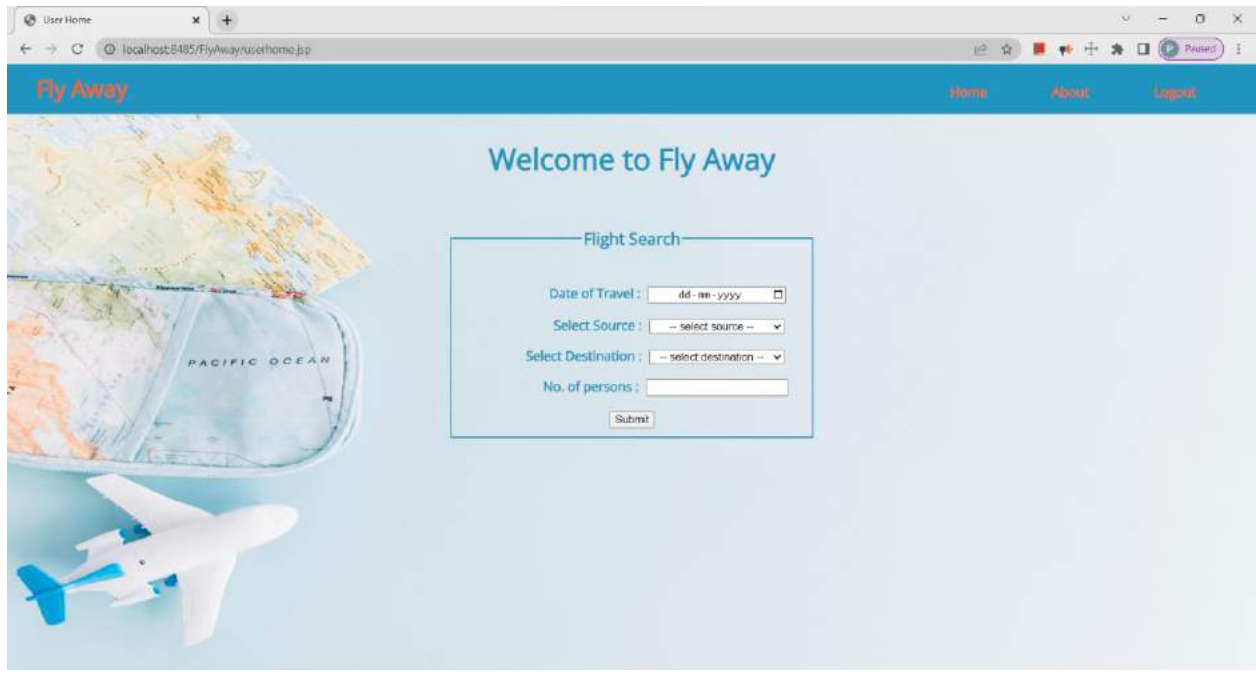
Username:

Password:

Submit

Not a registered user? [Click here](#)

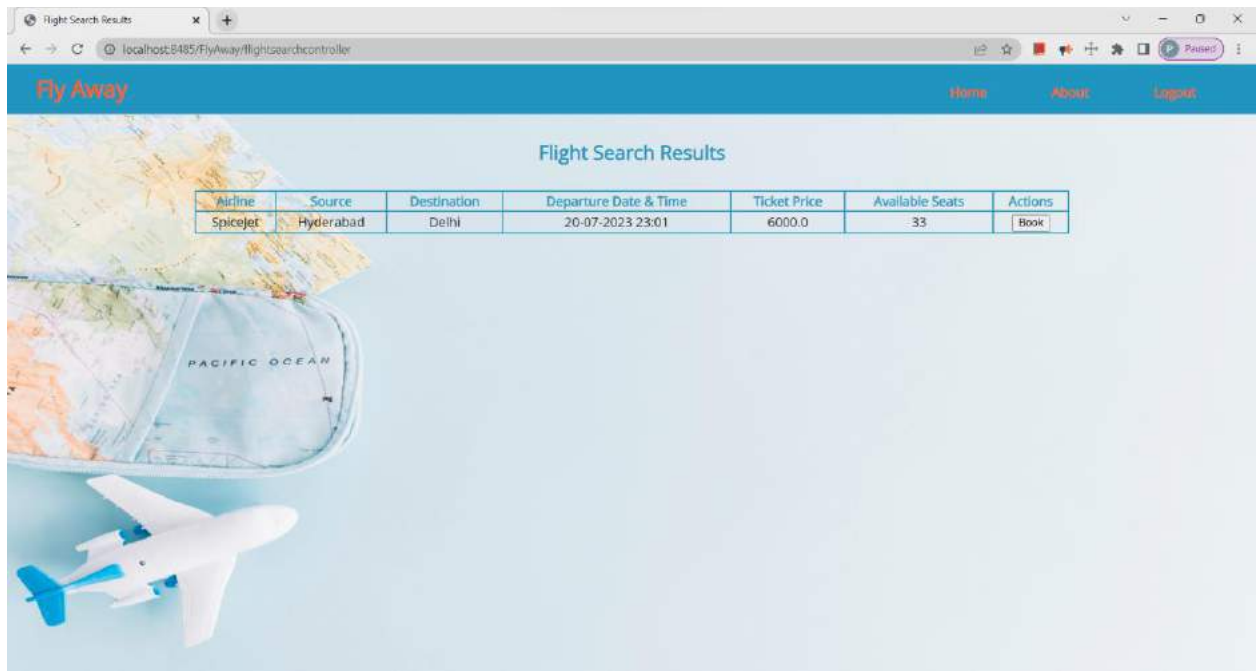
Output18 : User Home – Flight Search Page



The screenshot shows a web browser window with the title "User Home" and the URL "localhost:8485/FlyAway/userhome.jsp". The page has a blue header with the "Fly Away" logo and navigation links for "Home", "About", and "Logout". The main content area features a map of the Pacific Ocean on the left and a "Welcome to Fly Away" message on the right. Below the welcome message is a "Flight Search" form with the following fields:

- Date of Travel:
- Select Source:
- Select Destination:
- No. of persons:
- Submit button

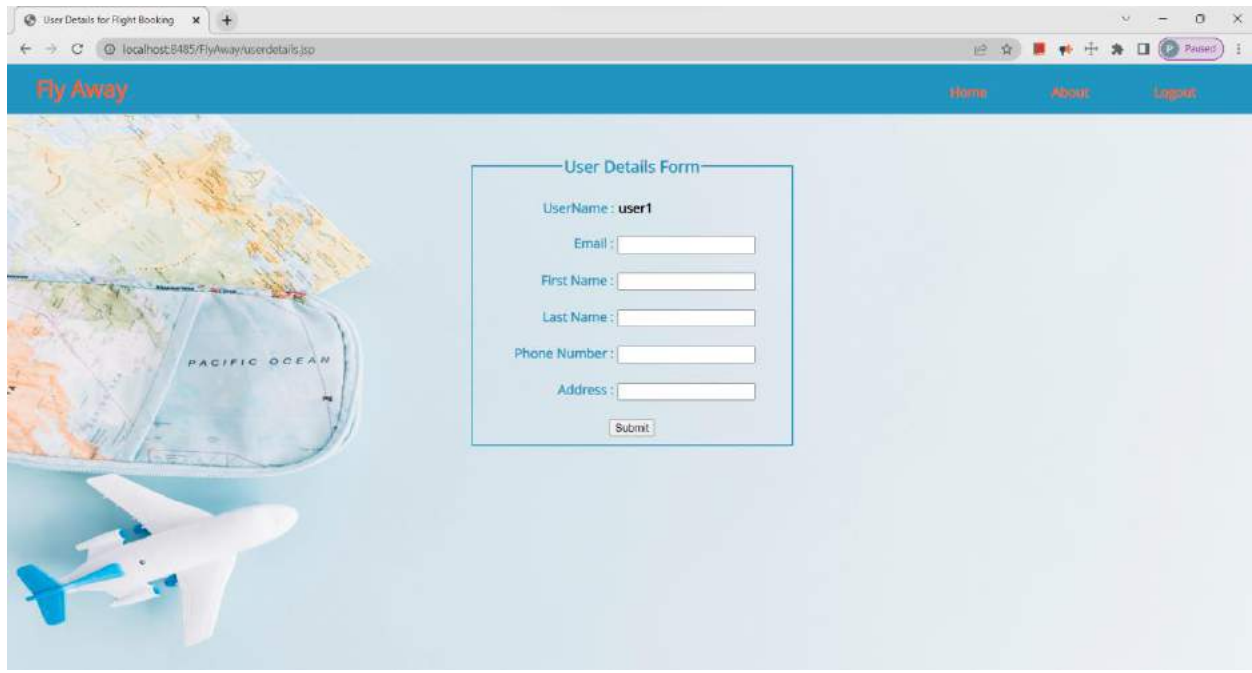
Output19 : Flight Search Results Page



The screenshot shows a web browser window with the title "Flight Search Results" and the URL "localhost:8485/FlyAway/flightsearchcontroller". The page has a blue header with the "Fly Away" logo and navigation links for "Home", "About", and "Logout". The main content area features a map of the Pacific Ocean on the left and a "Flight Search Results" message on the right. Below the message is a table with the following data:

Airline	Source	Destination	Departure Date & Time	Ticket Price	Available Seats	Actions
Spicejet	Hyderabad	Delhi	20-07-2023 23:01	6000.0	33	Book

Output20 : User Details Page



User Details for Flight Booking

localhost:8485/FlyAway/userdetails.jsp

Fly Away Home About Logout

User Details Form

UserName : user1

Email :

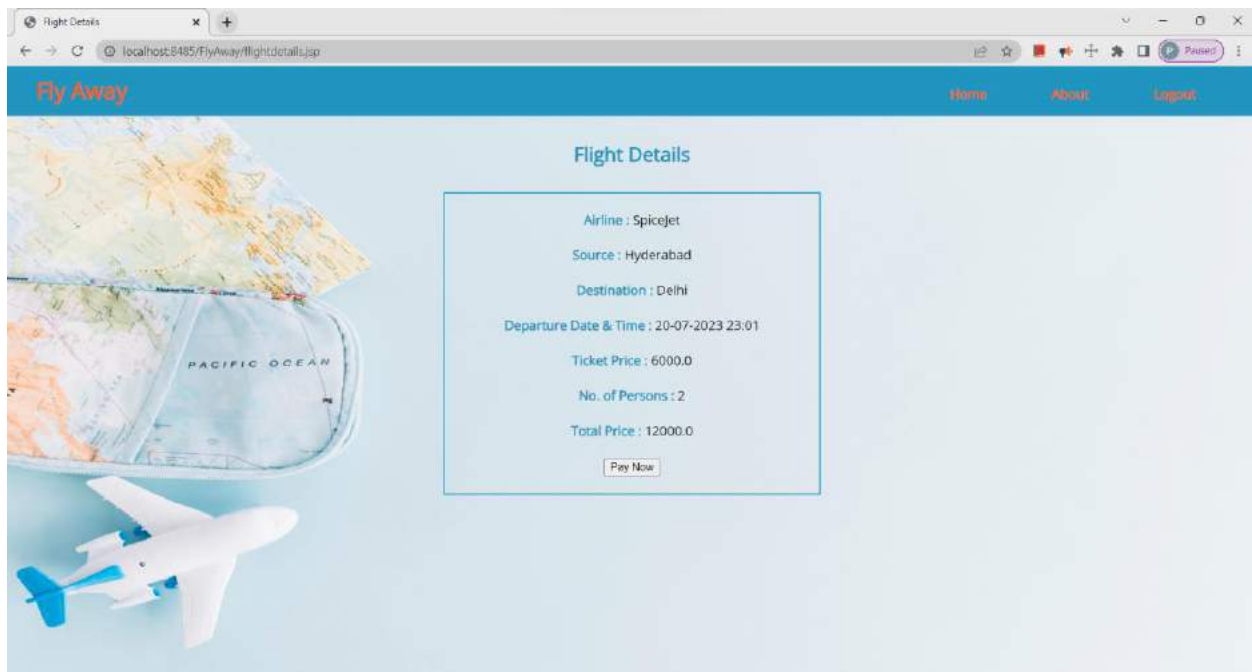
First Name :

Last Name :

Phone Number :

Address :

Output21 : Selected Flight Details Page



Flight Details

localhost:8485/FlyAway/flightdetails.jsp

Fly Away Home About Logout

Flight Details

Airline : SpiceJet

Source : Hyderabad

Destination : Delhi

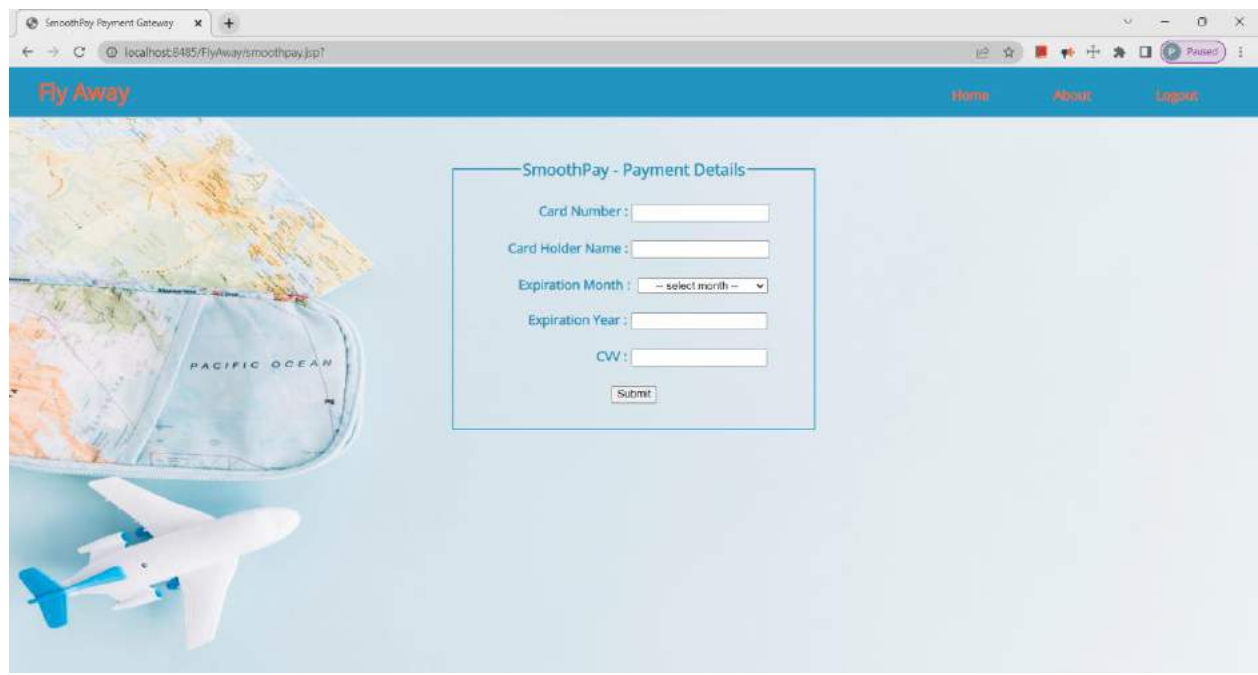
Departure Date & Time : 20-07-2023 23:01

Ticket Price : 6000.0

No. of Persons : 2

Total Price : 12000.0

Output22 : SmoothPay Payment Gateway Page



SmoothPay Payment Gateway

localhost:8485/FlyAway/smoothpay.jsp

Home About Logout

SmoothPay - Payment Details

Card Number :

Card Holder Name :

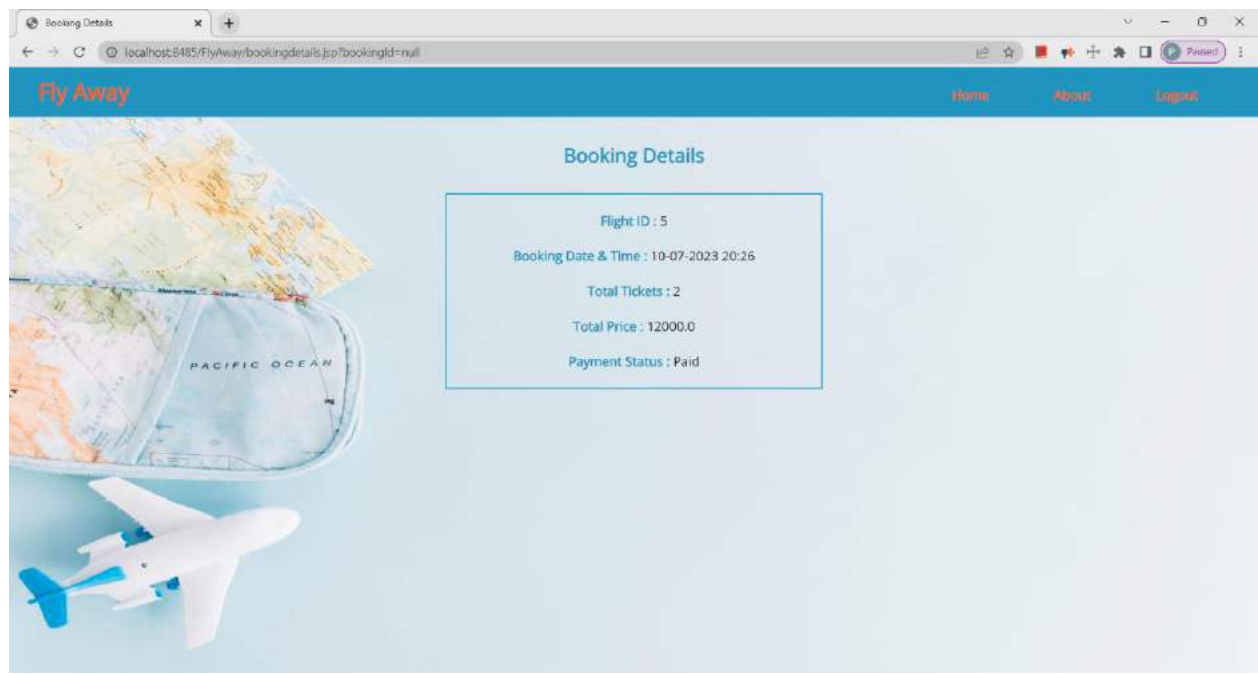
Expiration Month :

Expiration Year :

CW :

Submit

Output23 : Booking Details Page



Booking Details

localhost:8485/FlyAway/bookingdetails.jsp?bookingId=null

Home About Logout

Booking Details

Flight ID : 5

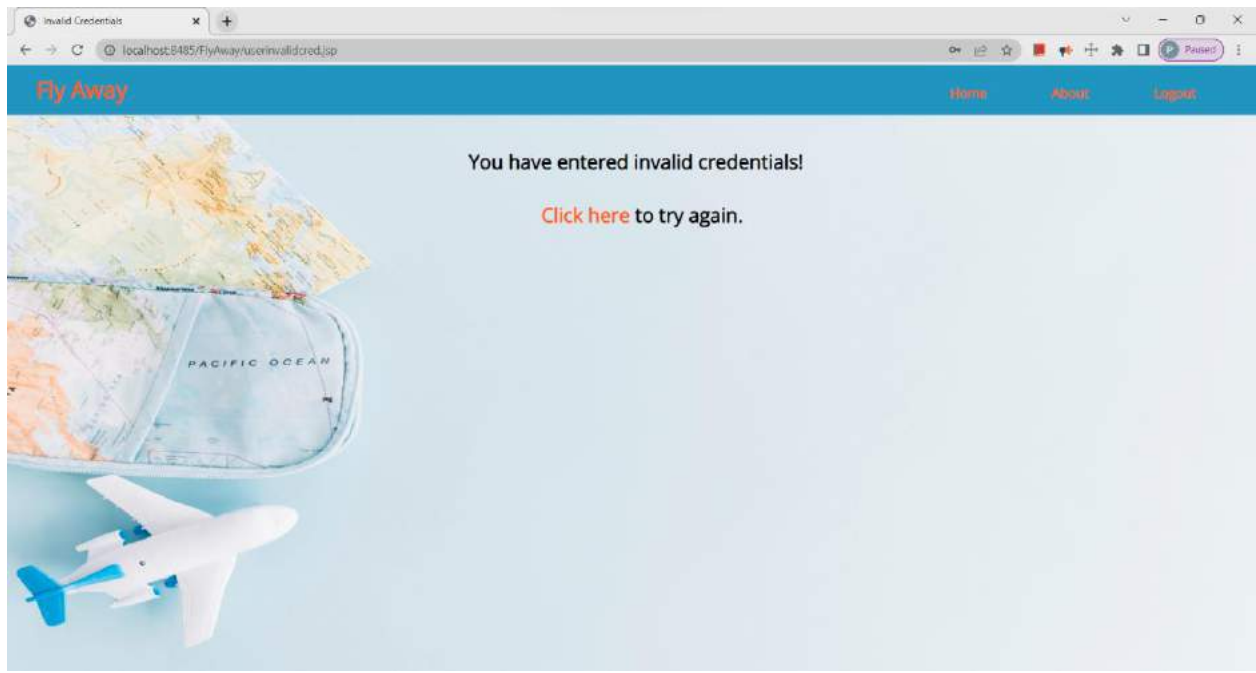
Booking Date & Time : 10-07-2023 20:26

Total Tickets : 2

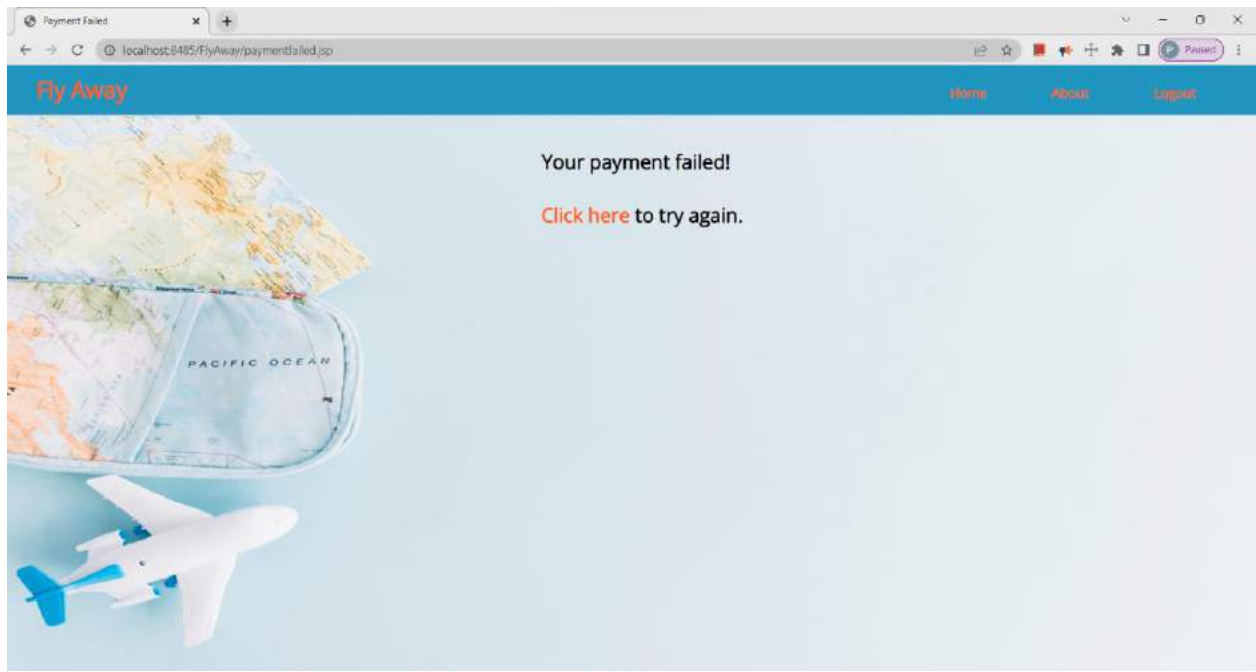
Total Price : 12000.0

Payment Status : Paid

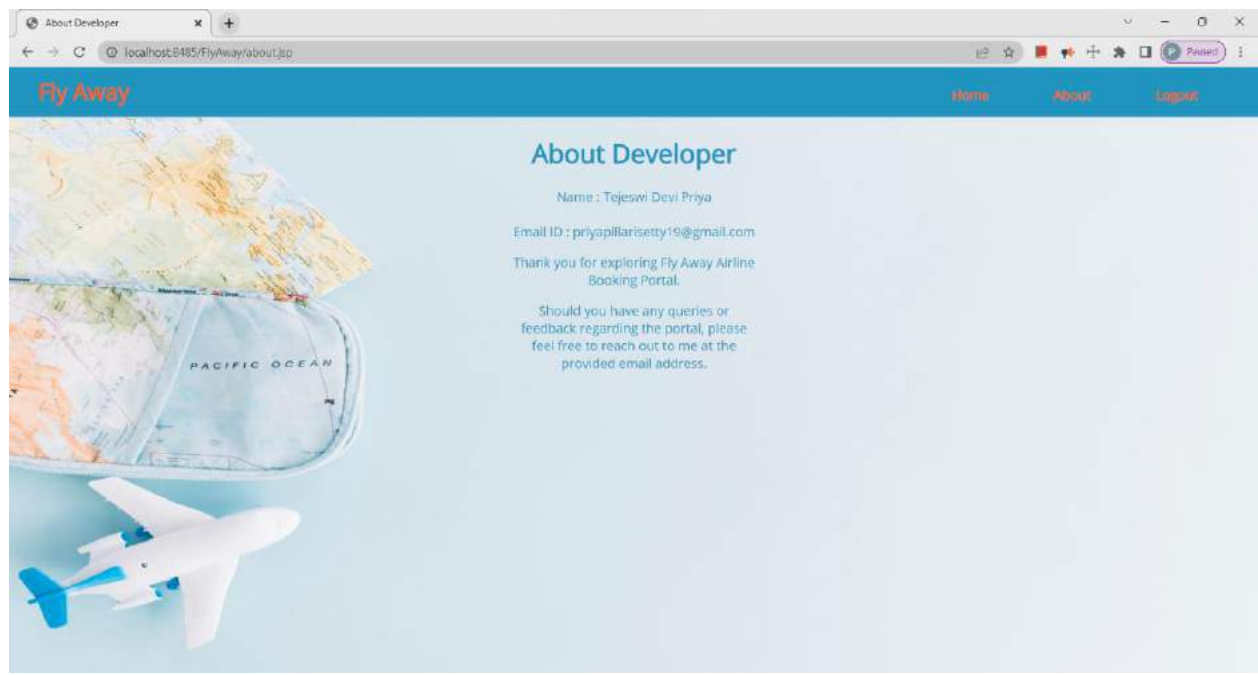
Output24 : Invalid Credentials Page



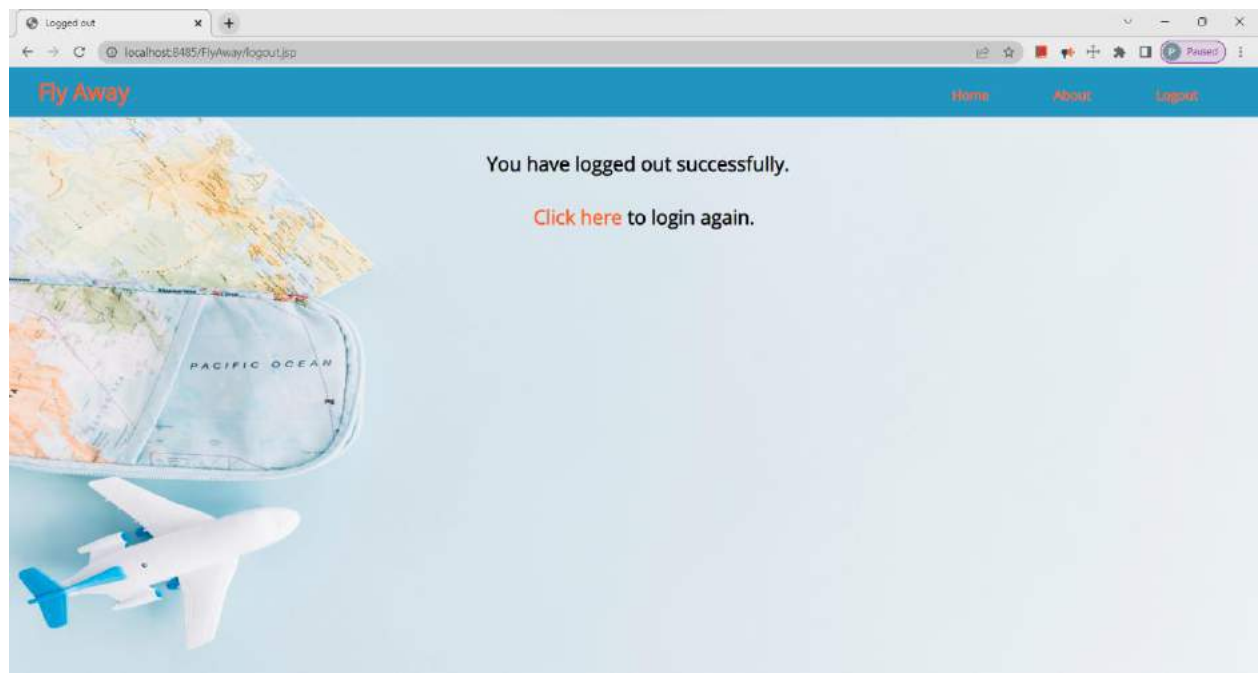
Output25 : Payment Failed Page



Output26 : About Developer Page

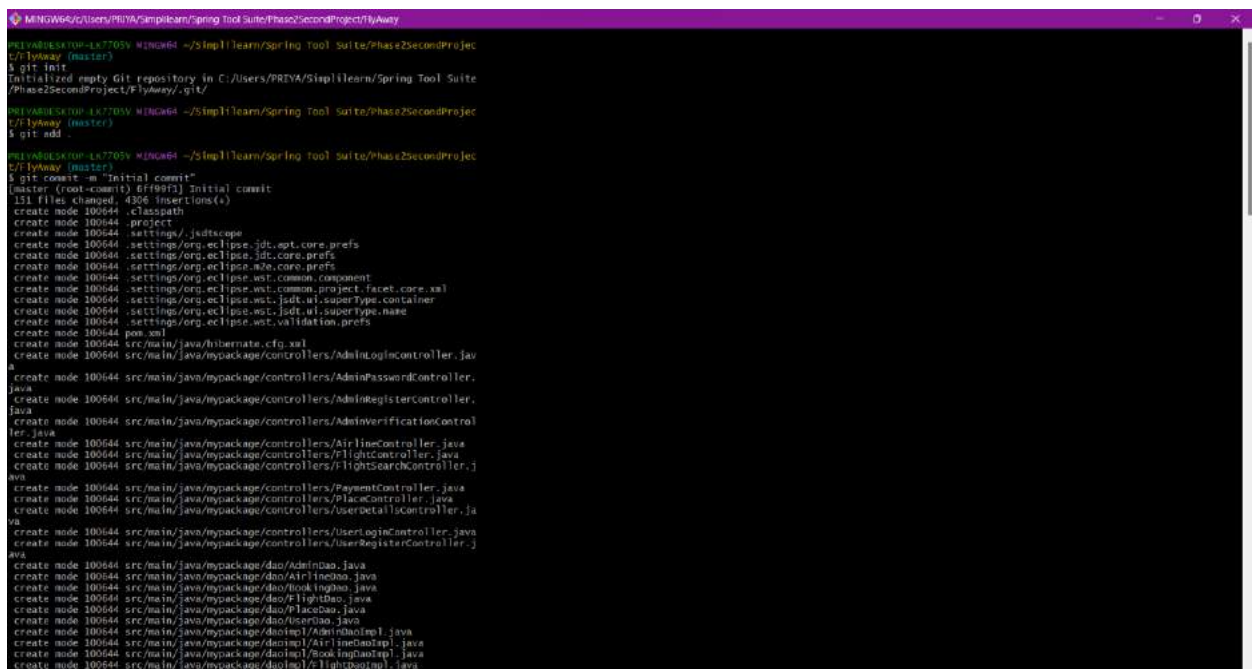


Output27 : Logout Page



12. Pushing the code to GitHub Repository:

- Open Git Bash and navigate to the folder where you have created your files.
cd Simplilearn\Spring Tool Suite\Phase2SecondProject\FlyAway
- Initialize the repository using the below command
git init
- Add all the files to your git repository using the below command
git add .
- Commit the changes using the below command
git commit -m "Initial commit"
- Add the URL for the remote repository where your local repository will be pushed
git remote add origin https://github.com/Priya-2705/Fly-Away.git
- Push the files to the folder you initially created using below command
git push -u origin master



```
MINGW64/G:/Users/PRIYA/Simplilearn/Spring Tool Suite/Phase2SecondProject/FlyAway
PRIYABESK@UP-147705V WING64 ~/Simplilearn/Spring Tool Suite/Phase2SecondProject/FlyAway (master)
$ git init
Initialized empty Git repository in C:/Users/PRIYA/Simplilearn/Spring Tool Suite/Phase2SecondProject/FlyAway/.git/

PRIYABESK@UP-147705V WING64 ~/Simplilearn/Spring Tool Suite/Phase2SecondProject/FlyAway (master)
$ git add .

PRIYABESK@UP-147705V WING64 ~/Simplilearn/Spring Tool Suite/Phase2SecondProject/FlyAway (master)
$ git commit -m "Initial commit"
[master (root-commit) 6f499f2] Initial commit
131 files changed, 4306 insertions(+)
create mode 100644 .classpath
create mode 100644 .project
create mode 100644 .settings/.jdt.scope
create mode 100644 .settings/org.eclipse.jdt.apt.core.prefs
create mode 100644 .settings/org.eclipse.jdt.core.prefs
create mode 100644 .settings/org.eclipse.wst.common.component
create mode 100644 .settings/org.eclipse.wst.common.project.facet.core.xml
create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.container
create mode 100644 .settings/org.eclipse.wst.jsdt.ui.superType.name
create mode 100644 .settings/org.eclipse.wst.validation.prefs
create mode 100644 pom.xml
create mode 100644 src/main/java/hibernate.cfg.xml
create mode 100644 src/main/java/mypackage/controllers/AdminLoginController.java
create mode 100644 src/main/java/mypackage/controllers/AdminPasswordController.java
create mode 100644 src/main/java/mypackage/controllers/AdminRegisterController.java
create mode 100644 src/main/java/mypackage/controllers/AdminVerificationController.java
create mode 100644 src/main/java/mypackage/controllers/AirlineController.java
create mode 100644 src/main/java/mypackage/controllers/FlightController.java
create mode 100644 src/main/java/mypackage/controllers/FlightSearchController.java
create mode 100644 src/main/java/mypackage/controllers/PaymentController.java
create mode 100644 src/main/java/mypackage/controllers/PlaceController.java
create mode 100644 src/main/java/mypackage/controllers/UserDetailsController.java
create mode 100644 src/main/java/mypackage/controllers/UserLoginController.java
create mode 100644 src/main/java/mypackage/controllers/UserRegisterController.java
create mode 100644 src/main/java/mypackage/dao/AdminDao.java
create mode 100644 src/main/java/mypackage/dao/AirlineDao.java
create mode 100644 src/main/java/mypackage/dao/BookingDao.java
create mode 100644 src/main/java/mypackage/dao/FlightDao.java
create mode 100644 src/main/java/mypackage/dao/PlaceDao.java
create mode 100644 src/main/java/mypackage/dao/UserDao.java
create mode 100644 src/main/java/mypackage/daoimpl/AdminDaoImpl.java
create mode 100644 src/main/java/mypackage/daoimpl/AirlineDaoImpl.java
create mode 100644 src/main/java/mypackage/daoimpl/BookingDaoImpl.java
create mode 100644 src/main/java/mypackage/daoimpl/FlightDaoImpl.java
```

```

MINGW64/C:/Users/NTA/Simplicam/gnng-1001 SuitePhaseSecondProject/Java
create mode 100644 src/main/java/mypackage/daoimpl/FlightDaoImpl.java
create mode 100644 src/main/java/mypackage/daoimpl/PlaceDaoImpl.java
create mode 100644 src/main/java/mypackage/daoimpl/UserDaoImpl.java
create mode 100644 src/main/java/mypackage/entities/Admin.java
create mode 100644 src/main/java/mypackage/entities/Airline.java
create mode 100644 src/main/java/mypackage/entities/Booking.java
create mode 100644 src/main/java/mypackage/entities/Flight.java
create mode 100644 src/main/java/mypackage/entities/Place.java
create mode 100644 src/main/java/mypackage/entities/User.java
create mode 100644 src/main/java/mypackage/models/AdminLoginModel.java
create mode 100644 src/main/java/mypackage/models/AdminModel.java
create mode 100644 src/main/java/mypackage/models/AirlineModel.java
create mode 100644 src/main/java/mypackage/models/FlightModel.java
create mode 100644 src/main/java/mypackage/models/PlaceModel.java
create mode 100644 src/main/java/mypackage/models/UserLoginModel.java
create mode 100644 src/main/java/mypackage/models/UserModel.java
create mode 100644 src/main/java/mypackage/serviceimpl/AdminServiceimpl.java
create mode 100644 src/main/java/mypackage/serviceimpl/AirlineServiceimpl.java
create mode 100644 src/main/java/mypackage/serviceimpl/FlightServiceimpl.java
create mode 100644 src/main/java/mypackage/serviceimpl/PlaceServiceimpl.java
create mode 100644 src/main/java/mypackage/serviceimpl/UserServiceimpl.java
create mode 100644 src/main/java/mypackage/services/AdminService.java
create mode 100644 src/main/java/mypackage/services/AirlineService.java
create mode 100644 src/main/java/mypackage/services/PlaceService.java
create mode 100644 src/main/java/mypackage/services/UserService.java
create mode 100644 src/main/java/mypackage/util/ThreadSafeUtil.java
create mode 100644 src/main/webapp/META-INF/MANIFEST.MF
create mode 100644 src/main/webapp/about.jsp
create mode 100644 src/main/webapp/addAirline.jsp
create mode 100644 src/main/webapp/addFlight.jsp
create mode 100644 src/main/webapp/addPlace.jsp
create mode 100644 src/main/webapp/admin.jsp
create mode 100644 src/main/webapp/adminChangePassword.jsp
create mode 100644 src/main/webapp/adminHome.jsp
create mode 100644 src/main/webapp/adminHome.css
create mode 100644 src/main/webapp/adminHome.jsp
create mode 100644 src/main/webapp/adminInvalidCode.jsp
create mode 100644 src/main/webapp/adminInvalidId.jsp
create mode 100644 src/main/webapp/adminLogin.jsp
create mode 100644 src/main/webapp/adminRegister.jsp
create mode 100644 src/main/webapp/adminRegistrationSuccessful.jsp
create mode 100644 src/main/webapp/airlineList.jsp
create mode 100644 src/main/webapp/bg.jpg
create mode 100644 src/main/webapp/bookingDetails.jsp
create mode 100644 src/main/webapp/confirmBooking.jsp
create mode 100644 src/main/webapp/flightDetails.jsp
create mode 100644 src/main/webapp/flightSearchResults.jsp
create mode 100644 src/main/webapp/flightsList.jsp
create mode 100644 src/main/webapp/header.jsp
create mode 100644 src/main/webapp/index.jsp
create mode 100644 src/main/webapp/logout.jsp
create mode 100644 src/main/webapp/newPassword.jsp
create mode 100644 src/main/webapp/passwordChanged.jsp
create mode 100644 src/main/webapp/paymentFailed.jsp
create mode 100644 src/main/webapp/placeList.jsp
create mode 100644 src/main/webapp/smoothway.jsp
create mode 100644 src/main/webapp/user.png

```

```

MINGW64/C:/Users/NTA/Simplicam/gnng-1001 SuitePhaseSecondProject/Java
create mode 100644 src/main/webapp/user.png
create mode 100644 src/main/webapp/userDetails.jsp
create mode 100644 src/main/webapp/userHome.css
create mode 100644 src/main/webapp/userHome.jsp
create mode 100644 src/main/webapp/userInvalidCode.jsp
create mode 100644 src/main/webapp/userLogin.jsp
create mode 100644 src/main/webapp/userRegister.jsp
create mode 100644 src/main/webapp/userRegistrationSuccessful.jsp
create mode 100644 target/classes/mypackage/controllers/AdminLoginController.class
create mode 100644 target/classes/mypackage/controllers/AdminPasswordController.class
create mode 100644 target/classes/mypackage/controllers/AdminRegisterController.class
create mode 100644 target/classes/mypackage/controllers/AdminVerificationController.class
create mode 100644 target/classes/mypackage/controllers/AirlineController.class
create mode 100644 target/classes/mypackage/controllers/FlightController.class
create mode 100644 target/classes/mypackage/controllers/FlightSearchController.class
create mode 100644 target/classes/mypackage/controllers/PaymentController.class
create mode 100644 target/classes/mypackage/controllers/PlaceController.class
create mode 100644 target/classes/mypackage/controllers/UserDetailsController.class
create mode 100644 target/classes/mypackage/controllers/UserLoginController.class
create mode 100644 target/classes/mypackage/controllers/UserRegisterController.class
create mode 100644 target/classes/mypackage/dao/AdminDao.class
create mode 100644 target/classes/mypackage/dao/AirlineDao.class
create mode 100644 target/classes/mypackage/dao/BookingDao.class
create mode 100644 target/classes/mypackage/dao/FlightDao.class
create mode 100644 target/classes/mypackage/dao/PlaceDao.class
create mode 100644 target/classes/mypackage/dao/UserDao.class
create mode 100644 target/classes/mypackage/daoimpl/AdminDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/AirlineDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/BookingDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/FlightDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/PlaceDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/UserDaoImpl.class
create mode 100644 target/classes/mypackage/entities/Admin.class
create mode 100644 target/classes/mypackage/entities/Airline.class
create mode 100644 target/classes/mypackage/entities/Booking.class
create mode 100644 target/classes/mypackage/entities/Flight.class
create mode 100644 target/classes/mypackage/entities/Place.class
create mode 100644 target/classes/mypackage/entities/User.class
create mode 100644 target/classes/mypackage/models/AdminLoginModel.class
create mode 100644 target/classes/mypackage/models/AdminModel.class
create mode 100644 target/classes/mypackage/models/AirlineModel.class
create mode 100644 target/classes/mypackage/models/FlightModel.class
create mode 100644 target/classes/mypackage/models/PlaceModel.class
create mode 100644 target/classes/mypackage/models/UserLoginModel.class
create mode 100644 target/classes/mypackage/models/UserModel.class
create mode 100644 target/classes/mypackage/serviceimpl/AdminServiceimpl.class
create mode 100644 target/classes/mypackage/serviceimpl/AirlineServiceimpl.class
create mode 100644 target/classes/mypackage/serviceimpl/FlightServiceimpl.class

```



```

MINGW64/C:/Users/Pritya/SimpleLearn/Spring Tool Suite/Phase2SecondProject/FlyAway
create mode 100644 target/classes/mypackage/daoimpl/AdminDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/AirlineDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/BookingDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/FlightDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/PlaceDaoImpl.class
create mode 100644 target/classes/mypackage/daoimpl/UserDaoImpl.class
create mode 100644 target/classes/mypackage/entities/Admin.class
create mode 100644 target/classes/mypackage/entities/Airline.class
create mode 100644 target/classes/mypackage/entities/Booking.class
create mode 100644 target/classes/mypackage/entities/Flight.class
create mode 100644 target/classes/mypackage/entities/Place.class
create mode 100644 target/classes/mypackage/entities/User.class
create mode 100644 target/classes/mypackage/models/AdminLoginModel.class
create mode 100644 target/classes/mypackage/models/AirlineModel.class
create mode 100644 target/classes/mypackage/models/FlightModel.class
create mode 100644 target/classes/mypackage/models/PlaceModel.class
create mode 100644 target/classes/mypackage/models/UserLoginModel.class
create mode 100644 target/classes/mypackage/models/UserModel.class
create mode 100644 target/classes/mypackage/serviceimpl/AdminServiceImpl.class
create mode 100644 target/classes/mypackage/serviceimpl/AirlineServiceImpl.class
create mode 100644 target/classes/mypackage/serviceimpl/BookingServiceImpl.class
create mode 100644 target/classes/mypackage/serviceimpl/FlightServiceImpl.class
create mode 100644 target/classes/mypackage/serviceimpl/PlaceServiceImpl.class
create mode 100644 target/classes/mypackage/services/AdminService.class
create mode 100644 target/classes/mypackage/services/FlightService.class
create mode 100644 target/classes/mypackage/services/PlaceService.class
create mode 100644 target/classes/mypackage/services/UserService.class
create mode 100644 target/classes/mypackage/util/HibernateUtil.class
create mode 100644 target/web-resources/META-INF/MANIFEST.MF
create mode 100644 target/web-resources/META-INF/maven/FlyAway/FlyAway/
pom.properties
create mode 100644 target/web-resources/META-INF/maven/FlyAway/FlyAway/
pom.xml

PRIVADESKTOP-147705V MINGW64 ~/SimpleLearn/Spring Tool Suite/Phase2SecondProject/FlyAway (master)
$ git remote add origin https://github.com/Pritya-2705/Fly-Away.git

PRIVADESKTOP-147705V MINGW64 ~/SimpleLearn/Spring Tool Suite/Phase2SecondProject/FlyAway (master)
$ git push -u origin master
Enumerating objects: 183, done.
Counting objects: 100% (183/183), done.
Delta compression using up to 8 threads
Compressing objects: 100% (171/171), done.
Writing objects: 100% (183/183), 985.24 KiB | 1.35 MiB/s, done.
Total 183 (delta 49), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (49/49), done.
To https://github.com/Pritya-2705/Fly-Away.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.

PRIVADESKTOP-147705V MINGW64 ~/SimpleLearn/Spring Tool Suite/Phase2SecondProject/FlyAway (master)

```

Pritya-2705/Fly-Away: Become a maintainer

github.com/Pritya-2705/Fly-Away

Pritya-2705 / Fly-Away

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Fly-Away Public

1 master 1 branch 0 tags

Go to file Add file <> Code About

Priya-2705 Initial commit errors 5 minutes ago 1 commit

settings	Initial commit	5 minutes ago
src/main	Initial commit	5 minutes ago
target	Initial commit	5 minutes ago
.classpath	Initial commit	5 minutes ago
.project	Initial commit	5 minutes ago
pom.xml	Initial commit	5 minutes ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

About

Become a backend expert - FlyAway (An Airline Booking Portal)

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Unique Selling Points of the Application:

- **Seamless Flight Booking:** The application provides a user-friendly interface for users to easily search, compare, and book flights to their desired destinations.
- **Efficient Flight Search:** Flyaway offers a straightforward flight search functionality that allows users to input their desired travel details and quickly retrieve relevant flight options. The search results are presented in a clear and organized manner, making it easy for users to compare and select their preferred flights.
- **Authentication:** Users and Admins are given the option to register themselves and create an account for themselves.
- **Error Handling:** If user enters invalid credentials, then user is redirected to an error page. If the user's payment is failed, then the user is redirected to Payment failed page.

Conclusion:

The application has been developed according to all the required features mentioned in the project description. Further enhancements to the application can be made by the following:

- **User Profile Customization:** Allow users to personalize their profiles by adding a profile picture, updating their contact information, and setting preferences such as preferred airlines or seat preferences.
- **Real-Time Notifications:** Implement real-time notifications to alert administrators of any critical updates or changes made by other administrators.
- **Booking History and Itinerary Management:** Enhance the user experience by providing a comprehensive booking history section where users can view their past and upcoming flights. Enable users to manage their itineraries by allowing them to make changes, add additional services, or request cancellations within a specified timeframe.