

This document contains sections for:

- Sprint planning and Task completion
- Core concepts used in project
- Flow of the website.
- Demonstrating the product capabilities, appearance, and user interactions.
- Unique Selling Points of the website
- Conclusions

The code for this project is hosted at <https://github.com/reemasoud/FlyAway.com.git>

The project is developed by **Reema Sarhan**.

Sprints planning and Task completion

The project is planned to be completed in 3 sprint.

Tasks assumed to be completed in the 1st sprint are:

- Creating the flow of the website(Admin ,passenger).
- Initializing git repository to pushing the project.
- Create Dynamic web project using servlet and jsp page, write some code like: write welcome page ,prime header links and their page.

Tasks assumed to be completed in the 2nd sprint are:

- Create classes of Admin, Flight, Airline and passenger and connect with database using hibernate.
- complete the rest of the code like:
 - Add, list of flight and Airline.
 - login operation for admin.
 - Search for flight and add passenger so on.

Tasks assumed to be completed in the 3rd sprint are:

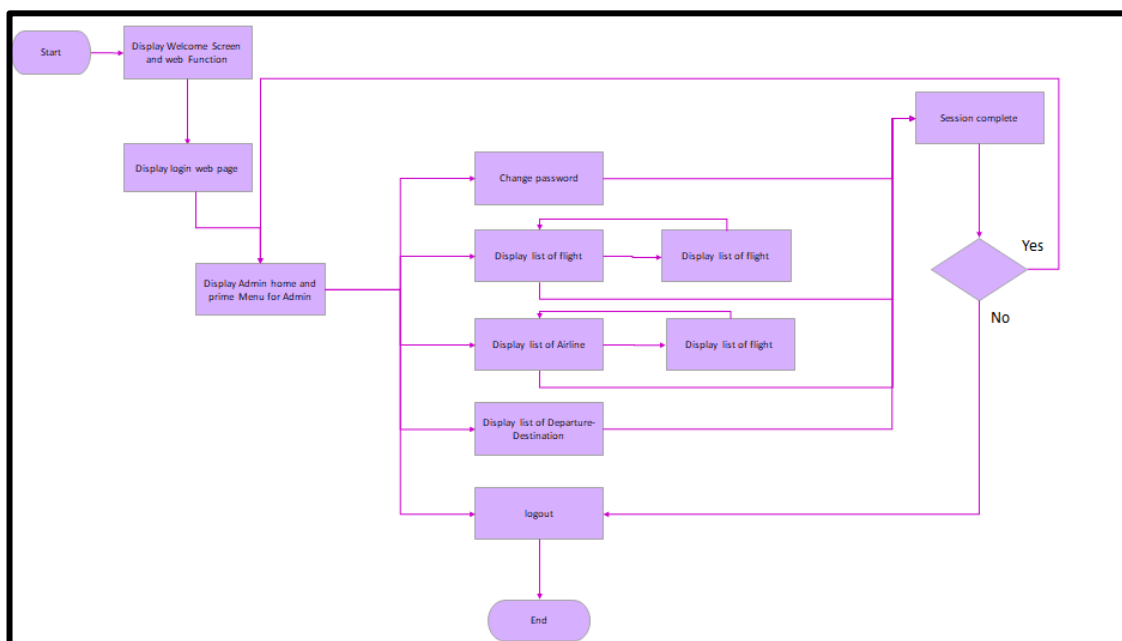
- Testing the websit with different kinds of User input

- Pushing code to GitHub.
- Creating this specification document highlighting website capabilities, appearance, and user interactions.

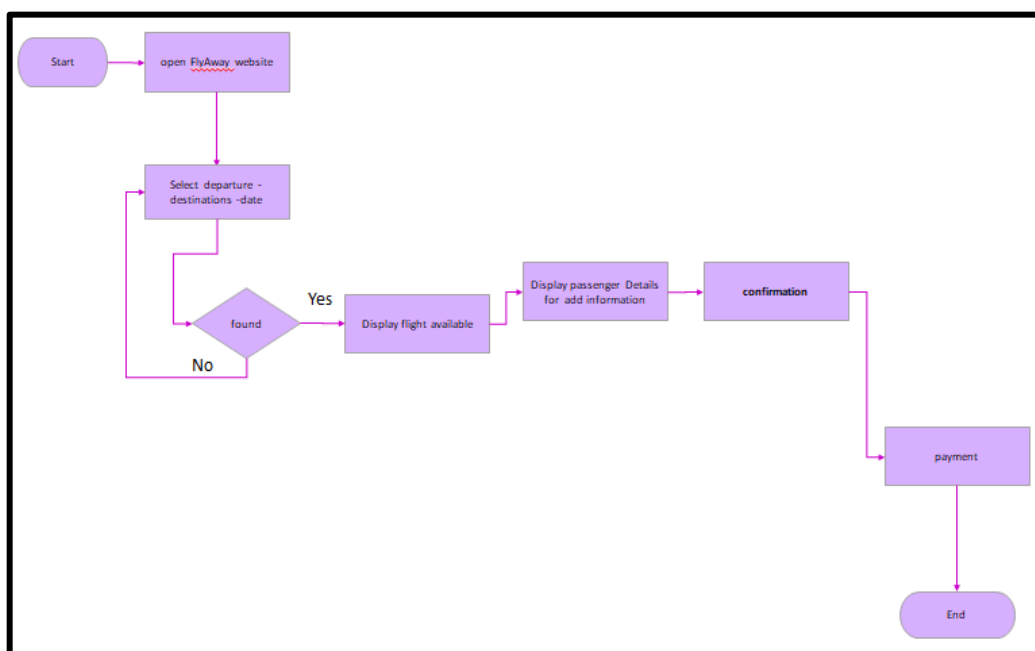
Core concepts used in project

Collections framework, , Sorting, Flow Control, jsp, servlet , Exception Handling, Database using hibernate.

Flow of the website side of Admin



Flow of the website



Demonstrating the product capabilities, appearance, and user interactions

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

- 1- Creating the Dynamic web project in Eclipse
- 2- Writing a web page in Eclipse for the welcome page of the website (index.html).
- 3- Writing a web pages in Eclipse for the Admin operations of the website .
- 4- Writing a web pages in Eclipse for the flight operations of the website.
- 5- Writing a web pages in Eclipse for the passenger operations of the website to allow book flight.
- 6- Pushing the code to GitHub repository.

Step 1: Creating a new project in Eclipse

- Open Eclipse
- Go to File -> New -> the Dynamic web project -> Next.
- Type in any project name and click on “Finish.”
- Select your project and go to File -> New -> html file.
- Enter index.html name of file and click on “Finish.”

Step 2: Writing a web page in Eclipse for the welcome page of the website and classes related with database using hibernate

Step 2.1. It have two button one for Admin login and another for search flight .

```
<body>
<div class="div3">
<table>
<tr><td>
<form action="Login.jsp" method="post">
<input type="submit" value="Admin Login" >
</form>
<td>
<form action="searchflight.jsp" method="post">
<input type="submit" value="book flight" >
</form></td>
<tr>
```

```

</table>
</div>
<div> 

</div>
<div class="div2">
<footer>
<p>&copy; 2021 All rights are reserved by FlyAway.com<p>
</footer>
</div>
</body>

```

Output



Step 2.2.admin class.

```

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

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="AdminId")
    private int Ad_Id;

    @Column(name="AdminEmail")
    private String A_Email;

    @Column(name="AdminPassword")
    private String A_Password;

    // constructor

    public Admin() {
        super();
    }

    // getter and setter methods
    public Admin(String a_Email, String a_Password) {
        super();
        A_Email = a_Email;
        A_Password = a_Password;
    }

    public String getA_Email() {
        return A_Email;
    }
}

```

```

    public void setA_Email(String a_Email) {
        A_Email = a_Email;
    }

    public String getA_Password() {
        return A_Password;
    }
    public void setA_Password(String a_Password) {
        A_Password = a_Password;
    }
}

```

Step 2.3.Flight class.

```

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

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="flightId")
    private      int f_Id;

    @Column(name="flightNumber")
    private      String f_Number;

    @Column(name="flightDeparture")
    private      String f_Departure;

    @Column(name="flightDestination")
    private      String f_Destination;

    @Column(name="flightDepartureTime")
    private      String f_DepartureTime;

    @Column(name="flightArrivalTime")
    private      String f_ArrivalTime;

    @Column(name="flightPrice")
    private      double f_Price;

    @Column(name="flightDate")
    private      String f_Date;

    @Column(name="flightGate")
    private      String f_Gate;

    @ManyToOne
    @JoinColumn(name="airlineId")
    private Airline Airlines;

    // constructor

```

```

    public Flight() {
        super();
    }

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

    public Flight(int f_Id, String f_Number, String f_Departure, String
f_Destination, String f_DepartureTime,
        String f_ArrivalTime, double f_Price, String f_Date,
String f_Gate, Airline airlines) {
        super();
        this.f_Id = f_Id;
        this.f_Number = f_Number;
        this.f_Departure = f_Departure;
        this.f_Destination = f_Destination;
        this.f_DepartureTime = f_DepartureTime;
        this.f_ArrivalTime = f_ArrivalTime;
        this.f_Price = f_Price;
        this.f_Date = f_Date;
        this.f_Gate = f_Gate;
        Airlines = airlines;
    }

    public Flight(String f_Number, String f_Departure, String
f_Destination, String f_DepartureTime, String f_ArrivalTime,
        double f_Price, String f_Date, String f_Gate) {
        super();
        this.f_Number = f_Number;
        this.f_Departure = f_Departure;
        this.f_Destination = f_Destination;
        this.f_DepartureTime = f_DepartureTime;
        this.f_ArrivalTime = f_ArrivalTime;
        this.f_Price = f_Price;
        this.f_Date = f_Date;
        this.f_Gate = f_Gate;
    }

    // getter and setter methods

    public int getF_Id() {
        return f_Id;
    }

    public void setF_Id(int f_Id) {
        this.f_Id = f_Id;
    }

    public String getF_Number() {
        return f_Number;
    }

```

```
public void setF_Number(String f_Number) {
    this.f_Number = f_Number;
}

public String getF_Departure() {
    return f_Departure;
}

public void setF_Departure(String f_Departure) {
    this.f_Departure = f_Departure;
}
public String getF_Destination() {
    return f_Destination;
}

public void setF_Destination(String f_Destination) {
    this.f_Destination = f_Destination;
}

public String getF_DepartureTime() {
    return f_DepartureTime;
}

public void setF_DepartureTime(String f_DepartureTime) {
    this.f_DepartureTime = f_DepartureTime;
}

public String getF_ArrivalTime() {
    return f_ArrivalTime;
}

public void setF_ArrivalTime(String f_ArrivalTime) {
    this.f_ArrivalTime = f_ArrivalTime;
}

public double getF_Price() {
    return f_Price;
}

public void setF_Price(double f_Price) {
    this.f_Price = f_Price;
}

public String getF_Date() {
    return f_Date;
}

public void setF_Date(String f_Date) {
    this.f_Date = f_Date;
}

public String getF_Gate() {
    return f_Gate;
}
```

```

        public void setF_Gate(String f_Gate) {
            this.f_Gate = f_Gate;
        }

        public Airline getAirlines() {
            return Airlines;
        }
        public void setAirlines(Airline airlines) {
            Airlines = airlines;
        }

    }

```

Step 2.4. Airline class.

```

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

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="airlineId")
    private      int A_Id;

    @Column(name="airlineName")
    private      String A_Name;

    @Column(name="CategoryAircraft")
    private      String A_Category;

    @OneToMany(cascade=CascadeType.ALL, fetch=FetchType.LAZY ,
mappedBy="Airlines")
    private Set<Flight> Flights;

    public Airline() {
        super();
    }

    public Airline(String a_Name) {
        super();
        A_Name = a_Name;
    }

    public Airline(String a_Name, String a_Category) {
        super();
        A_Name = a_Name;
        A_Category = a_Category;
    }
}

```



```

// getter and setter methods

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

public int getA_Id() {
    return A_Id;
}

public void setA_Id(int a_Id) {
    A_Id = a_Id;
}

public String getA_Name() {
    return A_Name;
}

public void setA_Name(String a_Name) {
    A_Name = a_Name;
}

public String getA_Category() {
    return A_Category;
}

public void setA_Category(String a_Category) {
    A_Category = a_Category;
}

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

```

Step 2.5. passenger class

```

@Entity
@Table(name="passenger")
public class Passenger {

    //properties
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="passengerId")
    private int p_Id;

    @Column(name="passengerEmail")
    private String p_Email;

    @Column(name="passengerName")
    private String p_Name;
}

```

```

@Column(name="passengerNationality")
private      String p_Nationality;

@Column(name="passengerPassport")
private      String p_passport;

@Column(name="passengerDateOfBirth")
private      String p_DateOfBirth;

@Column(name="passengerPhon")
private      String p_Phon;

@Column(name="passengerAddress")
private      String p_Address;

@ManyToMany(cascade=CascadeType.ALL, fetch=FetchType.LAZY)
@JoinTable(name="passenger_flight",
joinColumns=@JoinColumn(name="passengerId"),
inverseJoinColumns=@JoinColumn(name="flightId"))
private Set<Flight> Flights;

// constructor

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

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

public Passenger() {
    super();
}

    public Passenger(int p_Id, String p_Email, String p_Name, String
p_passport, String p_Phon, String p_Address) {
        super();
        this.p_Id = p_Id;
        this.p_Email = p_Email;
        this.p_Name = p_Name;
        this.p_passport = p_passport;
        this.p_Phon = p_Phon;
        this.p_Address = p_Address;
    }

    public Passenger(String p_Email, String p_Name, String
p_Nationality, String p_passport, String p_DateOfBirth,
        String p_Phon, String p_Address) {
        super();
        this.p_Email = p_Email;
        this.p_Name = p_Name;
        this.p_Nationality = p_Nationality;
        this.p_passport = p_passport;
        this.p_DateOfBirth = p_DateOfBirth;
        this.p_Phon = p_Phon;
    }

```

```
        this.p_Address = p_Address;
    }
```

```
// getter and setter methods
```

```
public int getP_Id() {
    return p_Id;
}
public void setP_Id(int p_Id) {
    this.p_Id = p_Id;
}
public String getP_Email() {
    return p_Email;
}
public void setP_Email(String p_Email) {
    this.p_Email = p_Email;
}
public String getP_Name() {
    return p_Name;
}
public void setP_Name(String p_Name) {
    this.p_Name = p_Name;
}
public String getP_passport() {
    return p_passport;
}
public void setP_passport(String p_passport) {
    this.p_passport = p_passport;
}
public String getP_Phon() {
    return p_Phon;
}
public void setP_Phon(String p_Phon) {
    this.p_Phon = p_Phon;
}
public String getP_Address() {
    return p_Address;
}
public void setP_Address(String p_Address) {
    this.p_Address = p_Address;
}

public String getP_DateOfBirth() {
    return p_DateOfBirth;
}

public void setP_DateOfBirth(String p_DateOfBirth) {
    this.p_DateOfBirth = p_DateOfBirth;
}

public String getP_Nationality() {
    return p_Nationality;
}

public void setP_Nationality(String p_Nationality) {
```

```

        this.p_Nationality = p_Nationality;
    }
}

```

Step 2.6. operations class

```

public class operations {

    // -----Flight-----
    public static List<Flight> listOfFlight() {

        SessionFactory factory =
        HibernateSessionUtil.buildSessionFactory();
        //open a session
        Session session = factory.openSession();

        List<Flight> listOfFlight = session.createQuery("from Flight
as e").list();

        return listOfFlight;
    }

    public static List<Flight> search(String fDeparture,String
fDestination,String fDate) {

        SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
        //open a session
        Session session = factory.openSession();

        List<Flight> listOfFlight = session.createQuery("from Flight as e
where flightDeparture=:fDeparture and flightDestination=:fDestination and
flightDate=:fDate").setParameter("fDeparture", fDeparture)
.setParameter("fDestination",
fDestination).setParameter("fDate", fDate).list();

        return listOfFlight;
    }

    public static List<Flight> search(int fId) {

        SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
        //open a session
        Session session = factory.openSession();

        List<Flight> listOfFlight = session.createQuery("from Flight as e
where flightId=:fId").setParameter("fId", fId).list();

        return listOfFlight;
    }

    public static List<Flight> test1(String fDeparture,String
fDestination,String fDate) {

        SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
        //open a session

```

```

        Session session = factory.openSession();//select * from Flight a
inner join Airline b on a.airlineId = b.airlineId

        List<Flight> listOfFlight = session.createQuery("select * from
Flight a inner join Airline b on a.airlineId = b.airlineId where
flightDeparture=:fDeparture and flightDestination=:fDestination and
flightDate=:fDate").setParameter("fDeparture", fDeparture)
                .setParameter("fDestination",
fDestination).setParameter("fDate", fDate).list();

        return listOfFlight;
    }

    //-----Airline-----

    public static List<Airline> listOfAirline() {

        SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
        //open a session
        Session session = factory.openSession();

        List<Airline> listOfAirline = session.createQuery("from Airline as
e").list();

        return listOfAirline;
    }

    //-----passenger-----
    public static List<Passenger> listOfPassenger(String email) {

        SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
        //open a session
        Session session = factory.openSession();

        List<Passenger> listOfPassenger = session.createQuery("from
Passenger e where passengerEmail=:email").setParameter("email",
email).list();

        return listOfPassenger;
    }

    public static List<Flight> test(String fDeparture,String
fDestination,String fDate) {

        SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
        //open a session
        Session session = factory.openSession();//and
flightDestination=:fDestination and flightDate=:fDate ,String
fDestination,String fDate

        List<Flight> listOfFlight = session.createQuery("from Flight as e
where flightDeparture=:fDeparture and flightDestination=:fDestination and
flightDate=:fDate").setParameter("fDeparture", fDeparture)
                .setParameter("fDestination",
fDestination).setParameter("fDate", fDate).list();

        return listOfFlight;
    }

```

```

}

//-----Admin-----

public static List<Admin> check(String AEmail,String APassword) {

    SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
    //open a session
    Session session = factory.openSession();

    List<Admin> listOfAdmin = session.createQuery("from Admin as e where
AdminEmail=:AEmail and AdminPassword=:APassword ").setParameter("AEmail",
AEmail)
    .setParameter("APassword", APassword).list();

    return listOfAdmin;
}

public static void updateA(Integer AdminId, String APassword ){
    SessionFactory factory = HibernateSessionUtil.buildSessionFactory();

    Session session = factory.openSession();
    Transaction tx = null;

    try {
        tx = session.beginTransaction();
        Admin admin = (Admin)session.get(Admin.class, AdminId);
        admin.setA_Password(APassword);
        session.update(admin);
        tx.commit();
    } catch (HibernateException e) {
        if (tx!=null) tx.rollback();
        e.printStackTrace();
    } finally {
        session.close();
    }
}

public static List<Admin> updateAdmin(String APassword) {

    SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
    //open a session
    Session session = factory.openSession();//"UPDATE Admin set
AdminPassword = :APassword WHERE id = 32"

    List<Admin> listOfAdmin = session.createQuery("UPDATE Admin set
AdminPassword = :APassword WHERE id=32 ").setParameter("APassword",
APassword).list();
    // session.update(APassword, listOfAdmin);
    return listOfAdmin;
}

public static void addAdmin(Admin admin) {
    // build session factory
    SessionFactory factory = HibernateSessionUtil.buildSessionFactory();
    //open a session
    Session session = factory.openSession();
    //begin tx
    Transaction tx = session.beginTransaction();

    //save employee

```

```

        session.save(admin);
        tx.commit();

        //close session
        session.close();
    }
}

```

Step 3: Writing a web pages in Eclipse for the Admin operations of the website

- Create Admin home and have menu bar of links.
- Menu consists pages for :-
 - 3.1. Displaying admin page again.
 - 3.2. Displaying page for change Password.
 - 3.3. Displaying page for a list of flight and add flight inside it.
 - 3.4. Displaying page for a list of airlines.
 - 3.5. Displaying page for a list of Departure Destination.
 - 3.6. logout from admin side.

Step 3.1: Writing login page to admin home and header page for menu bar.

```

<body>
    <div class="div3">
        <form action="Login-handel.jsp" method="post">
            <fieldset>
                <legend>LogIn</legend>
                <table>
                    <tr>
                        <td><label for="Email">Email</label><br>
                            <input type="text" id="Email" name="Email" value=""required><br>
                        </td>
                    </tr>
                    <tr>
                        <td><label for="Password">Password</label><br>
                            <input type="password" id="password" name="password"
value=""required><br>
                        </td>
                    </tr>
                    <tr>
                        <td><br><br><input type="submit" value="Login" ></td>
                    </tr>
                </table>
            </fieldset>
        </form>
    </div>
    <div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>

```

</body>

<body>

Login-handel page

```
<jsp:include page="header.jsp"></jsp:include>
<%

String email = request.getParameter("Email");
String password = request.getParameter("password");

List<Admin> list = operations.check(email, password);

if(email== null || email.equals("") || password== null ||
password.equals("")) {
    out.print("<h3> Login Failure ! due to invliad fields</h3>");
} else{
    for(Admin admin :list){
        if(email.equals(admin.getA_Email()) &&
password.equals(admin.getA_Password())){
            out.print("<h3> Login Successful ! </h3>");

            int HOUR24 = 60*60*24;
            session.setAttribute("email", email);
            session.setMaxInactiveInterval(HOUR24);

            response.sendRedirect("AdminHome.jsp");
        }else{
            out.print("<h3> Login Failure </h3>");
        }
    }
}
%>
```

Admin home page

</body>

<body>

```
<div class="div1" ><jsp:include page="header.jsp"></jsp:include></div>

<div class="div3">
<%
String email =(String) session.getAttribute("email");
if(email != null){

    out.print("<h4 style='color:black;text-align:center'>Hello user
"+email+"<h4>");
}else{
    out.print("<h3> somthing wrong ! </h3>");
}
%>

</div>

<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
```

</body>

Header page

<body>

<nav>

```
<a href="AdminHome.jsp">Admin Home</a> ||
<a href="changePassword.jsp">Change Password</a> ||
```



```

        <a href="FlightList.jsp">list of flights</a> ||
        <a href="AirlineList.jsp">list of airlines</a> ||
        <a href="DepartureDestinationList.jsp">list of Departure-
Destination</a> ||
        <a href="Logout.jsp">Logout</a> ||
    </nav>
</body>

```

Output

The image shows two screenshots of a web application. The top screenshot is the login page, which has a 'Login' tab. It contains an 'Email' field with 'admin@gmail.com' and a 'Password' field with four asterisks. A 'Login' button is at the bottom. Below the form is a copyright notice: '© 2021 All rights are reserved by FlyAway.com'. The bottom screenshot shows the page after a successful login. It features a navigation bar with links: 'Admin Home', 'Change Password', 'list of flights', 'list of airlines', 'list of Departure-Destination', and 'Logout'. The main content area displays 'Hello user admin@gmail.com' and the same copyright notice at the bottom.

Step 3.2: Writing page for change Password

```
<body>
```

```

<div class="div1" ><jsp:include page="header.jsp"></jsp:include></div>
<div class="div3">

```

```

    <form method="post" action="changePassword" >
    <fieldset>
        <legend>Change Password</legend>
        <tr>
            <td><label for="Email">Current password:</label><br>
            <input type="password" name="Currentpassword"
id="Currentpassword" required><br>
            </td>
        </tr>
        <tr>
            <td><label for="Password">New password:</label><br>

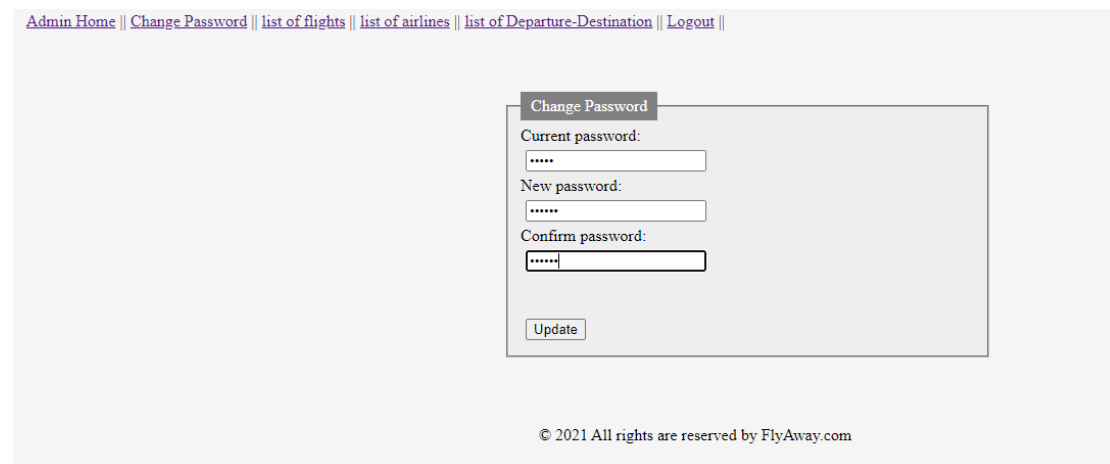
```

```

        <input type="password" name="Newpassword"
id="Newpassword"required> <br>
    </td>
</tr>
<tr>
    <td><label for="Password">Confirm password: </label><br>
        <input type="password" name="Confirmpassword"
id="Confirmpassword"required> <br>
    </td>
</tr>
<tr>
    <td><br><br><input type="submit" value="Update"></td>
</tr>
</fieldset>
</div>
<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

Output



Step 3.3: Writing page for a list of flight and add flight inside it.

List of flight

```

<body>
<div class="div1" ><jsp:include page="header.jsp"></jsp:include></div>
<div class="div3">

<h1 style="text-align:center;"> Flight List</h1>
<form action="Addflights.jsp">
    <%
        List<Flight> list = operations.listOfFlight();
        request.setAttribute("FlightList", list);
    %>

    <table>
    <tr>

        <th>FlightNumber</th>
        <th>flight Date</th>
        <th>Departure City</th>
        <th>Destination City</th>
        <th>Departure Time</th>
        <th>Arrival Time</th>
        <th>Airline Name</th>

```

```

        <th>Category of Aircraft</th>
        <th>flight Gate</th>
        <th>Price Of Ticket</th>

</tr>
<core:forEach items="${FlightList}" var="fli">
<tr>

        <td>${fli.getF_Number()}</td>
        <td>${fli.getF_Date()}</td>
        <td>${fli.getF_Departure()}</td>
        <td>${fli.getF_Destination()}</td>
        <td>${fli.getF_DepartureTime()}</td>
        <td>${fli.getF_ArrivalTime()}</td>
        <td>${fli.getAirlines().getA_Name()}</td>
        <td>${fli.getAirlines().getA_Category()}</td>
        <td>${fli.getF_Gate()}</td>
        <td>${fli.getF_Price()}SR</td>

</tr>
</core:forEach>
</table>
<input type="submit" value="Add Flight" >
</form>
</div>

<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>
<body>

```

```
<div class="div3">
```

Add flight page

```

<h1>Add Flight</h1>
<form method="post" action="Addflights" >
<fieldset>
<legend>Flight Information</legend>
flight Number: <input type="text" name="flightNumber" id="flightNumber">
<br><br>
flight Date: <input type="date" name="flightDate" id="flightDate"> <br><br>
Departure City: <select id="Departurecity" name="Departurecity">
<option value="Jeddah">Jeddah</option>
<option value="Riyadh">Riyadh</option>
<option value="Abha">Abha</option>
<option value="Dammam">Dammam</option>
<option value="altaif">Al-Taif</option>
<option value="Neom">Neom</option>
</select> <br><br>
Destination City: <select id="Destinationcity" name="Destinationcity">
<option value="Jeddah">Jeddah</option>
<option value="Riyadh">Riyadh</option>
<option value="Abha">Abha</option>
<option value="Dammam">Dammam</option>
<option value="altaif">Al-Taif</option>
<option value="Neom">Neom</option>
</select> <br><br>

Departure Time: <input type="time" name="DepartureTime" id="DepartureTime">
<br><br>

```

```

Arrival Time: <input type="time" name="ArrivalTime" id="ArrivalTime">
<br><br>
Airline Name:<select id="AirlineName" name="AirlineName">
<option value="Saudi Airlines">Saudi Airlines</option>
<option value="Flynas">Flynas</option>
<option value="Flyadeal">Flyadeal</option>
</select> <br><br>

Category of Aircraft: <select id="CategoryAircraft"
name="CategoryAircraft">
<option value="Airbus A330">Airbus A330</option>
<option value="Airbus A380">Airbus A380</option>
<option value="Boeing 737">Boeing 737</option>
<option value="Boeing 777">Boeing 777</option>
</select> <br><br>
flight Price: <input type="text" name="flightPrice" id="flightPrice">
<br><br>
flight Gate :<input type="text" name="flightGate" id="flightGate"> <br><br>

<input type="submit" value="Add">
</fieldset>
</div>
<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

Add flight Servlet

```

protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.setContentType("text/html");

    request.getRequestDispatcher("AdminHome.jsp").include(request,
response);

    request.getRequestDispatcher("Addflights.jsp").include(request,
response);
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    request.getRequestDispatcher("AdminHome.jsp").include(request,
response);

    String firstNumber = request.getParameter("flightNumber");
    String flightDate = request.getParameter("flightDate");
    String DepartureCity = request.getParameter("Departurecity");
    String DestinationCity =
request.getParameter("Destinationcity");
    String DepartureTime = request.getParameter("DepartureTime");
    String ArrivalTime = request.getParameter("ArrivalTime");

```

```

        String AirlineName = request.getParameter("AirlineName");
        String CategoryAircraft =
request.getParameter("CategoryAircraft");
        double flightPrice =
Double.parseDouble(request.getParameter("flightPrice"));
        String flightGate = request.getParameter("flightGate");

        try {
            // 1. load session factory
            SessionFactory factory =
HibernateSessionUtil.buildSessionFactory();

            // 2. create a session
            Session session = factory.openSession();

            // 3. begin transaction
            Transaction tx = session.beginTransaction();

            // 4. create persistence object / add product
            Airline airline= new
Airline(AirlineName,CategoryAircraft);

            Flight flight =new Flight(firstNumber, DepartureCity,
DestinationCity, DepartureTime, ArrivalTime,
                flightPrice, flightDate, flightGate);
            flight.setAirlines(airline);

            // 5. save product
            session.save(flight);
            session.save(airline);

            // 6. commit transaction
            tx.commit();

            out.print("<h3 style='color:green'> Flight is created
with Airline successfully !<h3>");
            response.sendRedirect("FlightList.jsp");
            // 3. close session
            session.close();
        } catch (Exception e) {
            out.print("<h3 style='color:red'> Create Flight failed
! <h3>" + e);
        }
    }
}

```

Output

Flight List

FlightNumber	flight Date	Departure City	Destination City	Departure Time	Arrival Time	Airline Name	Category of Aircraft	flight Gate	Price Of Ticket
R-98	2021-06-30	Jeddah	Riyadh	18:24	21:27	Saudi Airlines	Airbus A380	G-1	300.0SR
R-99	2021-07-01	Riyadh	Jeddah	09:00	00:00	Saudi Airlines	Airbus A330	G-2	300.0SR
R-100	2021-07-02	Jeddah	Neom	21:30	00:00	Saudi Airlines	Boeing 777	G-3	800.0SR
R-101	2021-07-04	Dammam	altaif	14:30	16:30	Saudi Airlines	Airbus A380	G-4	280.0SR
R-102	2021-07-06	Neom	Riyadh	06:30	10:30	Saudi Airlines	Airbus A330	G-6	800.0SR
R-103	2021-07-06	Jeddah	altaif	12:35	13:59	Flyadeal	Airbus A330	G-7	150.0SR
R-104	2021-07-08	altaif	Riyadh	09:37	12:34	Flynas	Boeing 737	G-8	200.0SR

[Add Flight](#)

Add Flight

Flight Information

flight Number:

flight Date:

Departure City:

Destination City:

Departure Time:

Arrival Time:

Airline Name:

Category of Aircraft:

flight Price:

flight Gate:

[Add](#)

Step 3.4: Writing page for a list of airlines.

```
<body>
<div class="div1" ><jsp:include page="header.jsp"></jsp:include></div>
<div class="div3">

<h1 style="text-align:center;"> Flight List</h1>
<form action="Addflights.jsp">
    <%
        List<Flight> list = operations.listOfFlight();
        request.setAttribute("FlightList", list);
    %>

    <table>
        <tr>

            <th>FlightNumber</th>
            <th>flight Date</th>
            <th>Departure City</th>
            <th>Destination City</th>
            <th>Departure Time</th>
            <th>Arrival Time</th>
            <th>Airline Name</th>
            <th>Category of Aircraft</th>
```

```

        <th>flight Gate</th>
        <th>Price Of Ticket</th>

</tr>
<core:forEach items="${FlightList}" var="fli">
<tr>

        <td>${fli.getF_Number()}</td>
        <td>${fli. getF_Date()}</td>
        <td>${fli.getF_Departure()}</td>
        <td>${fli.getF_Destination()}</td>
        <td>${fli.getF_DepartureTime()}</td>
        <td>${fli.getF_ArrivalTime()}</td>
        <td>${fli.getAirlines().getA_Name()}</td>
        <td>${fli.getAirlines().getA_Category()}</td>
        <td>${fli.getF_Gate()}</td>
        <td>${fli.getF_Price()}SR</td>

</tr>
</core:forEach>
</table>
<input type="submit" value="Add Flight" >
</form>
</div>

<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

Output

[Admin Home](#) || [Change Password](#) || [list of flights](#) || [list of airlines](#) || [list of Departure-Destination](#) || [Logout](#) ||

Airline List

Airline Name	Aircraft category
Saudi Airlines	Airbus A380
Saudi Airlines	Airbus A330
Saudi Airlines	Boeing 777
Saudi Airlines	Airbus A380
Saudi Airlines	Airbus A330
Flyadeal	Airbus A330
Flynas	Boeing 737

© 2021 All rights are reserved by FlyAway.com

Step 3.5: Writing Displaying page for a list of Departure Destination.

```

<body>

<div class="div1"><jsp:include page="header.jsp"></jsp:include></div>
<div class="div3">

<h1 style="text-align:center;"> Departure Destination List</h1>

```

```

<form action="FlightList.jsp">
    <%
        List<Flight> list = operations.listOfFlight();
        request.setAttribute("FlightList", list);

    %>
    <table>
    <tr>

        <th>Departure City</th>
        <th>Destination City</th>

    </tr>
    <core:forEach items="${FlightList}" var="fli">
    <tr>

        <td>${fli.getF_Departure()}</td>
        <td>${fli.getF_Destination()}</td>

    </tr>
    </core:forEach>
    </table>

</form>
</div>
<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

Output

Admin Home || [Change Password](#) || [list of flights](#) || [list of airlines](#) || [list of Departure-Destination](#) || [Logout](#) ||

Departure Destination List	
Departure City	Destination City
Jeddah	Riyadh
Riyadh	Jeddah
Jeddah	Neom
Dammam	altaif
Neom	Riyadh
Jeddah	altaif
altaif	Riyadh

© 2021 All rights are reserved by FlyAway.com

Step 3.6: logout from admin side.

```

<body>

    <%
        session.invalidate();
        response.sendRedirect("index.html");
    %>
</body>

```


Step 4: Writing a web pages in Eclipse for the passenger operations of the website to allow book flight.

- Create page it called search flight.
- for Book a flight consists sequential pages:-
 - 4.1. Creating search flight page .
 - 4.2. Creating a page for showing available flight.
 - 4.3. Creating a page to allow user to add their information.
 - 4.4. Creating a page for showing booking information.
 - 4.5. Creating a page to allow user to pay.

Step 4.1: Writing page to create search flight page.

```
<body>
<div class="div3">
<form action="flights.jsp" method="post">
  <fieldset>
    <legend>Book a Flight</legend>
    <table >

      <tr>
        <td>From</td>
        <td>To</td>
        <td>Departure Date</td>
        <td>No.of people</td>
      </tr>
      <tr>
        <td>
          <select id="FromCity" name="FromCity">
            <option value="Jeddah">Jeddah</option>
            <option value="Riyadh">Riyadh</option>
            <option value="Abha">Abha</option>
            <option value="Dammam">Dammam</option>
            <option value="altaif">Al-Taif</option>
            <option value="Neom">Neom</option>
          </select><br> </td>
        <td>
          <select id="ToCity" name="ToCity">
            <option value="Jeddah">Jeddah</option>
            <option value="Riyadh">Riyadh</option>
            <option value="Abha">Abha</option>
            <option value="Dammam">Dammam</option>
            <option value="altaif">Al-Taif</option>
            <option value="Neom">Neom</option>
          </select> </td>
        <td><input type="date" id="DepartureDate" name="DepartureDate" ><br>
      </td>
        <td><select id="NoOfPeople" name="NoOfPeople">
          <option value="1">1</option>
          <option value="2">2</option>
          <option value="3">3</option>
        </select> </td>
      </tr>
    </tr>
  </table>
</div>
</body>
```

```

        <td ></td>
    </tr>
</table>

<input type="submit" value="Search Flights" >
</fieldset>
</form>
</div>
<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

Output

Step 4.2: Writing page for showing available flight.

```

<body>
<%
String DepartureCity=request.getParameter("FromCity");
String DestinationCity=request.getParameter("ToCity");
String Date=request.getParameter("DepartureDate");
String persons=request.getParameter("NoOfPeople");
%>
<div class="div3">
    <form action="PassengerDetails.jsp">
        <%
            List<Flight> list =
operations.search(DepartureCity, DestinationCity, Date);
            request.setAttribute("FlightList", list);
        %>

        <table>
            <tr>
                <th>FlightNumber</th>
                <th>flight Date</th>
                <th>Departure City</th>
                <th>Destination City</th>
                <th>Departure Time</th>
                <th>Arrival Time</th>
                <th>Airline Name</th>
                <th>flight Gate</th>
                <th>Price Of Ticket</th>
                <th>No.of people</th>
                <th></th>
            </tr>
            <core:forEach items="${FlightList}" var="fli">
                <tr>
                    <td><input type="hidden" id="F_Id" name="F_Id"
value="${fli.getF_Id()}" >
                        <td>${fli.getF_Number()}</td>

```

```

        <td>${fli.getF_Date()}</td>
        <td>${fli.getF_Departure()}</td>
        <td>${fli.getF_Destination()}</td>
        <td>${fli.getF_DepartureTime()}</td>
        <td>${fli.getF_ArrivalTime()}</td>
        <td>${fli.getAirlines().getA_Name()}</td>
        <td>${fli.getF_Gate()}</td>
        <td>${fli.getF_Price()}</td>
        <td> <%=persons %><input type="hidden" id="NoOfPeople"
name="NoOfPeople" value="<%=persons %>" ></td>
        <td><input type="submit" value="select" ></td>
    </tr>
</core:forEach>
</table>
</form>
</div>
<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

Output

FlightNumber	flight Date	Departure City	Destination City	Departure Time	Arrival Time	Airline Name	flight Gate	Price Of Ticket	No.of people	
R-98	2021-06-30	Jeddah	Riyadh	18:24	21:27	Saudi Airlines	G-1	300.0	3	<input type="button" value="select"/>
R-105	2021-06-30	Jeddah	Riyadh	00:45	03:35	Flyadeal	G-8	250.0	3	<input type="button" value="select"/>

© 2021 All rights are reserved by FlyAway.com

Step 4.3: Writing page to allow user to add their information.

```

<body>

<%
    int fId1 = Integer.parseInt(request.getParameter("F_Id"));
    List<Flight> list1 = operations.search(fId1);
    request.setAttribute("FlightList", list1);

    %>

<%String persons=request.getParameter("NoOfPeople");
String F_Id=request.getParameter("F_Id");
int fId = Integer.parseInt(request.getParameter("F_Id"));

%>

<div class="div3">
<fieldset>
    <legend>Enter your information</legend>
    <form action="PassengerDetails" method="post">
        <input type="hidden" id="NoOfPeople" name="NoOfPeople" value="<%=persons
%>" >
        <input type="hidden" id="F_Id" name="F_Id" value="<%=F_Id %>" >
        <core:forEach items="${FlightList}" var="fli">
        <input type="hidden" id="price" name="price" value="${fli.getF_Price() }" >
        </core:forEach>
        <table>
            <tr>
                <td>Name</td>

```

```

        <td>Date of birth</td>
    </tr>
    <tr>
        <td><input type="text" id="passengerName" name="passengerName"
value=""required></td>
        <td><input type="date" id="DateOfBirth" name="DateOfBirth"
value=""required></td>
    </tr>
    <tr>
        <td>Nationality</td>
        <td>Passport Number</td>
    </tr>
    <tr>
        <td>
            <select id="Nationality" name="Nationality">
                <option value="Saudi Arabia">Saudi Arabia</option>
                <option value="India">India</option>
                <option value="Kuwait">Kuwait</option>
                <option value="Bahrain">Bahrain</option>
                <option value="Oman">Oman</option>
            </select>
        </td>
        <td><input type="text" id="PassportNumber" name="PassportNumber"
value=""required></td>
    </tr>
    <tr>
        <td>Address</td>
        <td>Phone</td>
    </tr>
    <tr>
        <td><input type="text" id="Address" name="Address"
value=""required></td>
        <td><input type="text" id="Phone" name="Phone" value=""required></td>
    </tr>
    <tr>
        <td>Email</td>
        <td><input type="email" id="email" name="email"required></td>
    </tr>
</table>

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

</form>
</fieldset>
</div>

<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

```

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

</form>
</fieldset>
</div>

<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

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

    int fId = Integer.parseInt(request.getParameter("F_Id"));
    String passengerName = request.getParameter("passengerName");

```

Add
passenger
Servlet

```

        String DateOfBirth = request.getParameter("DateOfBirth");
        String Nationality = request.getParameter("Nationality");
        String PassportNumber =
request.getParameter("PassportNumber");
        String Address = request.getParameter("Address");
        String Phone = request.getParameter("Phone");
        String email = request.getParameter("email");

        try {
            // 1. load session factory
            SessionFactory factory =
HibernateSessionUtil.buildSessionFactory();

            // 2. create a session
            Session session = factory.openSession();

            // 3. begin transaction
            Transaction tx = session.beginTransaction();

            // 4. create persistence object / add product
            List<Flight> list = operations.search(fId);
            request.setAttribute("FlightList", list);
            int id=0;
            String f_Number="";
            String f_Departure="";
            String f_Destination="";
            String f_DepartureTime="";
            String f_ArrivalTime="";
            double f_Price=0.0;
            String f_Date="";
            String f_Gate="";
            Airline airlines = null;
            for (Flight i : list) {
                id=i.getF_Id();
                f_Number=i.getF_Number();
                f_Departure=i.getF_Departure();
                f_Destination=i.getF_Destination();
                f_DepartureTime=i.getF_DepartureTime();
                f_ArrivalTime=i.getF_ArrivalTime();
                f_Price=i.getF_Price();
                f_Date=i.getF_Date();
                f_Gate=i.getF_Gate();
                airlines=i.getAirlines();
            }

            Set<Flight> Flights =new HashSet<>();
            Flight f1=new Flight( id, f_Number, f_Departure,
f_Destination, f_DepartureTime,
                                f_ArrivalTime, f_Price, f_Date,
f_Gate, airlines);

            Flights.add(f1);
            Passenger passenger = new
Passenger(email,passengerName, Nationality,PassportNumber, DateOfBirth,
Phone,Address);

            passenger.setFlights(Flights);
            // 5. save product

```

```

        session.save(passenger);
        // 6. commit transaction
        tx.commit();
        out.print("<h3 style='color:green'> Passenger is
created successfully !<h3>");

        request.getRequestDispatcher("bookingInfo.jsp").include(request,
response);

        // 3. close session
        session.close();
    } catch (Exception e) {
        out.print("<h3 style='color:red'> Create Passenger
failed ! <h3>" + e);
    }
}
}

```

Output

Step 4.4: Writing page for showing booking information.

```

<body>
<div class="div3">
    <%
    int fId = Integer.parseInt(request.getParameter("F_Id"));
    List<Flight> list1 = operations.search(fId);
    request.setAttribute("FlightList", list1);
    String price = "";

    %>
    <%
    String email=request.getParameter("email");
    List<Passenger> list = operations.listOfPassenger(email);
    request.setAttribute("PassengerList", list);

    Double fprice
=Double.parseDouble(request.getParameter("price"));

    Double
persons=Double.parseDouble(request.getParameter("NoOfPeople"));
    Double Total = 0.0;

    %>
    <fieldset>
        <legend> your booking information</legend>

```

```

<table>
<tr>
<th>Name</th>
<th>Date of birth</th>
<th>Nationality</th>
<th>Passport Number</th>
<th>Address</th>
<th>Email</th>
<th>Phone</th>
<th>Flight Number</th>
<th>flight Gate</th>
</tr>
<core:forEach items="${PassengerList}" var="Pas">
<tr>
<td>${Pas.getP_Name()}</td>
<td>${Pas.getP_DateOfBirth()}</td>
<td>${Pas.getP_Nationality()}</td>
<td>${Pas.getP_passport()}</td>
<td>${Pas.getP_Address()}</td>
<td>${Pas.getP_Email()}</td>
<td>${Pas.getP_Phon()}</td>
<td>${fli.getF_Number()}</td>
<td>${fli.getF_Gate()}</td>
</tr>
</core:forEach>
</tr>
</table>

<br>
<br>
<br>
<form action="paymentInfo.jsp" method="post">
<table>
<tr>
<td>Price Of Ticket</td>
<td>Total Price</td>
<td></td>
</tr>

<tr>

<td><%=fprice %></td>

<td> <%=Total=persons*fprice%> </td>

<td> <input type="submit" value="Complete to pay" ></td>
</tr>
</table>
</form>
</fieldset>
</div>
<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>

```

Output

your booking information								
Name	Date of birth	Nationality	Passport Number	Address	Email	Phone	Flight Number	flight Gate
reema	1996-03-24	Saudi Arabia	m-24996	jeddah-8832	reema@gmail.com	0555553996	R-987	G-7
Price Of Ticket			Total Price					
280.0			560.0					
<input type="button" value="Complete to pay"/>								

© 2021 All rights are reserved by FlyAway.com

Step 4.5: Writing a page to allow user to pay.

```
<body>

<div class="div3">
<fieldset>
    <legend>Your are using Credit/Debit Card for payment</legend>
<form action="confirmpage.jsp" method="post">

<table>

    <tr>
        <td><label for="CardNumber">Card Number</label><br>
            <input type="text" id="CardNumber" name="CardNumber" required>
        </td>
        <td><label for="CardHolderName">Card Holder Name</label><br>
            <input type="text" id="CardHolderName" name="CardHolderName"
required>
        </td>
    </tr>
    <tr>
        <td><label for="ExpiryDate">Expiry Date</label><br>
            <input type="Date" id="ExpiryDate" name="ExpiryDate" required>
        </td>
        <td><label for="CW2/CV2Number">CW2/CV2 Number</label><br>
            <input type="text" id="CW2/CV2Number" name="CW2/CV2Number"
required>
        </td>
    </tr>

</table>
<input type="submit" value="Confirm" >
</form>
</fieldset>
</div>
<div class="div2"><jsp:include page="footer.jsp"></jsp:include></div>
</body>
```

Output

Your are using Credit/Debit Card for payment	
Card Number <input type="text"/>	Card Holder Name <input type="text"/>
Expiry Date <input type="text"/> / <input type="text"/>	CW2/CV2 Number <input type="text"/>
<input type="button" value="Confirm"/>	

© 2021 All rights are reserved by FlyAway.com

Step 6: Pushing the code to GitHub repository

```
cd C:\Users\toshiba\eclipse-workspace\Project2
```

```
git init
```

```
git add .
```

```
git commit . -m "push the project"
```

```
git push -u origin master
```

Unique Selling Points of the Application

1. Even if there are exceptions, the program is meant to keep operating and accepting user input..
2. The website can take any date as input for searching flight or add flight..
3. The admin is also given the option of change password if they so desire.
4. Even after doing any essential function such as change password , showing list of flight, showing list of airline and showing list of airline Departure -Destination the admin may easily transition between menu bar.

Conclusions

The application may be improved further, for example:

- Allowing user book round-trip.
- Allowing user to check reservation by using their email.
- Sending notifications to user for remember about their reservation.