

**Project Specification Document**

**FlyAway An Airline Booking Portal**

End of Phase 2-Become a back-end expert  
Project

**Student: Thirumavalaven**

[vthiru97@gmail.com](mailto:vthiru97@gmail.com)

Full Stack Java Developer

Master's Program

## Project Objective

As a Full Stack Developer, design and develop an airline booking portal named as FlyAway

## Developer Details

The project is developed by Thirumavalaven. I worked as a Service Delivery Specialist at IBM India Pvt. Ltd.

The code for this project is hosted at  
<https://github.com/Thiru97/Simplilearn2023-FlyAway>

## Project Details

FlyAway is a ticket-booking portal that lets people book flights on their website. As a Full Stack Developer, design and develop an airline booking portal named as FlyAway. Use the GitHub repository to manage the project artefacts.

The Website should meet the following requirements

- 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.

## Sprint Planning and Tasks Achieved

The project is planned to be completed in 1 sprint. Tasks assumed to be completed in the sprint are:

- Creating the flow of the application
- Initializing git repository to track changes as development progresses.
- Writing the Java program to fulfill the requirements of the project.
- Testing the Java program with different kinds of User input
- Pushing code to GitHub.
- Creating this specification document highlighting application capabilities, appearance, and user interactions.

## Project Overview

The main objectives of this Projects are

- To gain an understanding of core concepts of the Java Programming Language (abstraction, polymorphism, inheritance, and encapsulation),
- Embrace the Eclipse Integrated Development Environment (IDE),
- Understand the Agile software development life cycle
- Gain familiarity with Java data structures for object-oriented applications.
- Familiarize with concepts of Servlets , Servlet Filter, HTTP Methods
- Familiarize with concepts of Web Application Development using Servlets
- Learn Maven, A build Automation tool and handle dependencies
- To learn SQL and do CRUD Operation(Create, Read, Update, Delete)
- To learn Hibernate and design a Web Application with Servlets and perform CRUD Operation (Create, Read, Update, Delete).
- To learn Hibernate's Association and perform @OnetoOne , @OnetoMany @ManytoMany Associations with different tables of a database
- Learn Cookie and HTTP Session Management and Perform User's Session Management in a Web Application
- Learn MVC Architecture and build a Web Application using Servlets, JSP, Hibernate and perform CRUD Operations

## Implemented Java Concepts

This section will highlight the Java concepts used to create the FlyAway Airline website. HTML, CSS, JSP,JSTL, Servlets, Servlet Filters, HTTP Methods, SQL, JDBC Connection for SQL , Java Persistence API, Hibernate, HTTP Sessions and Cookie , Servlet MVC Architecture ,Collections framework, Flow Control, Exception Handling, are the core concepts used in this program. ***The entire Application was built in JAVA 19***

## Packages

I chose to create a package dedicated to the FlyAway Airlines as per the naming standards.

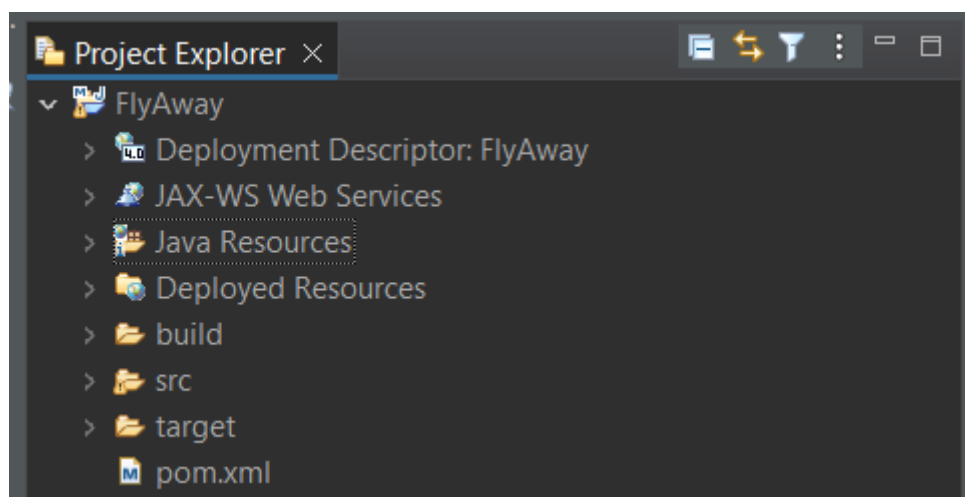
```
> com.flyaway.model
```

## Maven

Maven is an open-source build tool from Apache Group. It is a widely-used and popular tool to build, publish, and deploy software development projects. It is used to build and manage diverse Java-based projects. A powerful project management tool, it is based on the project object model (POM) concept.

STEP 1:

Create a new Maven project via **File >New> Other> Maven> Maven Project**. Project name as FlyAway. Once Done Maven will build the project and include a pom.xml file



## STEP 2:

Add the Following dependencies in pom.xml

```
<dependencies>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.0.1</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
  <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>
</dependencies>
```

## Servlet MVC Architecture

We will create our web application that implements the Model View Controller (MVC) design pattern, using basic Servlets and JSPs.

Model-View-Controller (MVC) is a pattern used in software engineering to separate the application logic from the user interface. As the name implies, the MVC pattern has three layers.

**The Model defines the business layer of the application, the Controller manages the flow of the application, and the View defines the presentation layer of the application.**

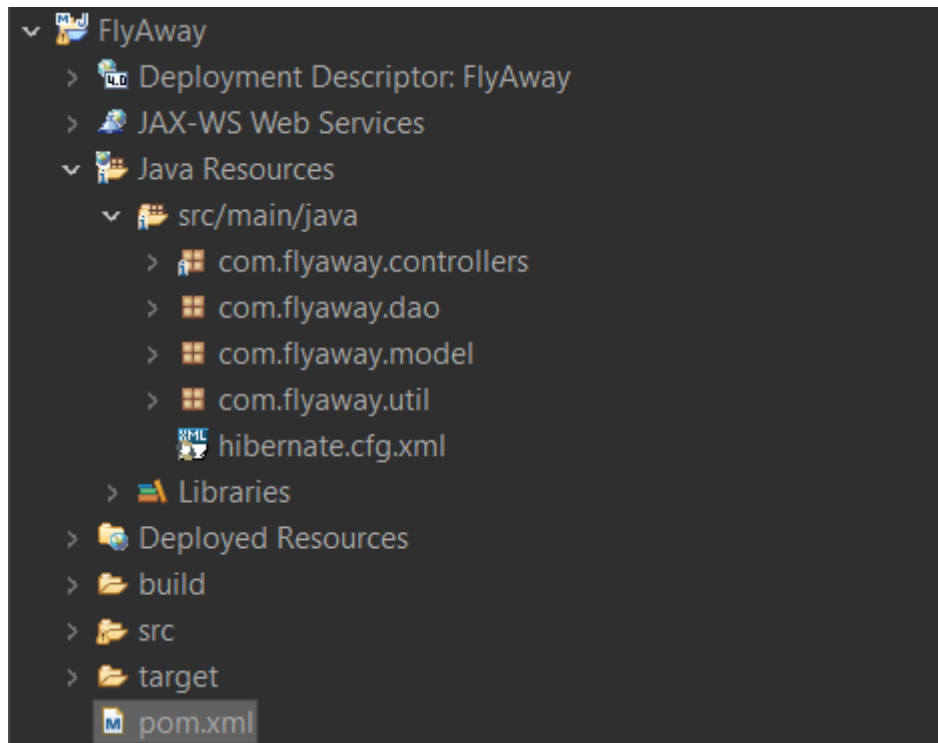
Although the MVC pattern isn't specific to web applications, it fits very well in this type of applications. In a Java context, the Model consists of simple Java classes, the Controller consists of servlets and the View consists of JSP pages.

Here're some key features of the pattern:

- It separates the presentation layer from the business layer
- The Controller performs the action of invoking the Model and sending data to View
- The Model is not even aware that it is used by some web application or a desktop application

## FlyAway -An Airline Booking Portal

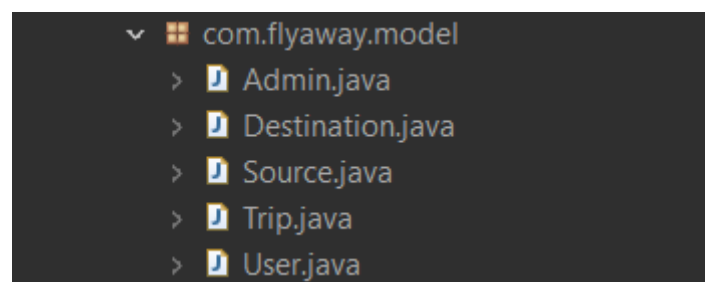
Below is the Maven and MVC Architecture based package Structure of FlyAway Web Application



Let's have a look at each layer.

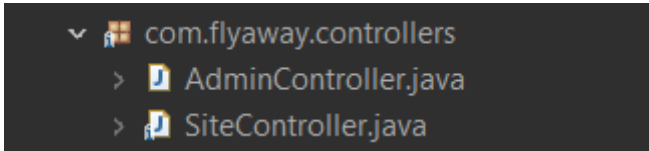
### The Model Layer

This is the data layer which contains business logic of the system, and also represents the state of the application. It's independent of the presentation layer, the controller fetches the data from the Model layer and sends it to the View layer.



## The Controller Layer

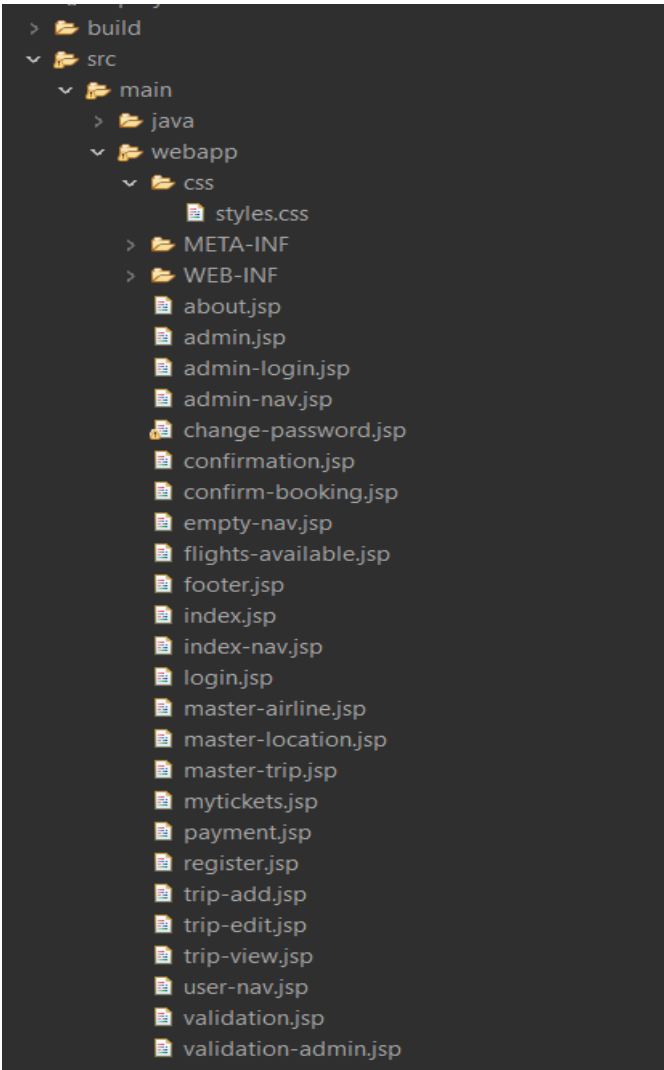
Controller layer acts as an interface between View and Model. It receives requests from the View layer and processes them, including the necessary validations. The requests are further sent to Model layer for data processing, and once they are processed, the data is sent back to the Controller and then displayed on the View.



```
com.flyaway.controllers
├── AdminController.java
└── SiteController.java
```

## The View Layer

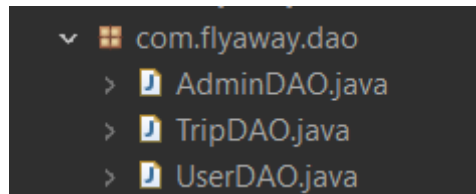
This layer represents the output of the application, usually some form of UI. The presentation layer is used to display the Model data fetched by the Controller.



```
build
src
├── main
│   ├── java
│   └── webapp
│       ├── css
│       │   └── styles.css
│       ├── META-INF
│       ├── WEB-INF
│       ├── about.jsp
│       ├── admin.jsp
│       ├── admin-login.jsp
│       ├── admin-nav.jsp
│       ├── change-password.jsp
│       ├── confirmation.jsp
│       ├── confirm-booking.jsp
│       ├── empty-nav.jsp
│       ├── flights-available.jsp
│       ├── footer.jsp
│       ├── index.jsp
│       ├── index-nav.jsp
│       ├── login.jsp
│       ├── master-airline.jsp
│       ├── master-location.jsp
│       ├── master-trip.jsp
│       ├── mytickets.jsp
│       ├── payment.jsp
│       ├── register.jsp
│       ├── trip-add.jsp
│       ├── trip-edit.jsp
│       ├── trip-view.jsp
│       ├── user-nav.jsp
│       ├── validation.jsp
│       └── validation-admin.jsp
```

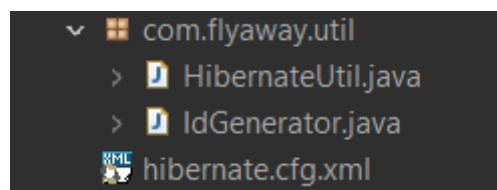
## The DAO Layer

DAO stands for Data Access Object. DAO Design Pattern is used to separate the data persistence logic in a separate layer. This way, the service remains completely in dark about how the low-level operations to access the database is done. This is known as the principle of **Separation of Logic**.



## The Util Package

The Util package consists of helper methods that our web application needs. We have a HibernateUtil.java files which creates a session factory with the help of hibernate.cfg.xml. The xml file hibernate.cfg.xml contains the necessary configuration needed for our hibernate to run. The IdGenerator.java files create a random customized ID number for our flights and Trips



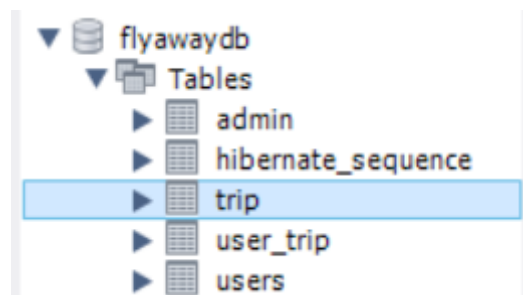
## Database

I have used MySQL database. I also used an ORM (Object Relational Mapping) tool like Hibernate. Hibernate makes our project loosely coupled with respect to database. In future we can easily change from MySQL database to other database like Postgres Database. All we need to do is change the configuration file

## MySQL

MySQL is a widely used relational database management system (RDBMS).MySQL is free and open-source. MySQL is ideal for both small and large applications.

I have used MySQL 8.0.33 in this application and below is the schema





## Hibernate

Hibernate is an open source Object-Relational Persistence and Query service for any Java Application. Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieves the developer from most common data persistence related programming tasks. Hibernate sits between traditional Java objects and database server to handle all the works in persisting those objects based on the appropriate O/R mechanisms and patterns.

Hibernate reduces lines of code by maintaining object-table mapping itself and returns result to application in form of Java objects. It relieves programmer from manual handling of persistent data, hence reducing the development time and maintenance cost.

In order to connect our database with Hibernate we need to define the configurations which includes the database port number, username and password. We define these configuration in hibernatecfg.xml file

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE hibernate-configuration PUBLIC
3    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
5
6  <hibernate-configuration>
7    <session-factory>
8      <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
9      <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/flyawayDB</property>
10     <property name="hibernate.connection.username">root</property>
11     <property name="hibernate.connection.password">root@thiru1234</property>
12
13     <property name="hibernate.show_sql">true</property>
14     <property name="hibernate.format_sql">true</property>
15     <property name="hbm2ddl.auto">update</property>
16   </session-factory>
17 </hibernate-configuration>
```

## Hibernate Association

Association in hibernate tells the relationship between the objects of POJO classes i.e. how the entities are related to each other. Association or entities relationship can be unidirectional or bidirectional.

Below is the associations used by Flyaway project

### @ManyToMany Association

Two items are said to be in Many-to-Many relationship if many occurrence of item are belong to the many occurrences of other item and vice versa

In this application Trip and Users are mapped with many to many association as user can book multiple tickets

## FlyAway -An Airline Booking Portal

```
@ManyToMany(fetch = FetchType.EAGER)
@JoinTable(name = "user_trip",
joinColumns = @JoinColumn(name = "trip_id"), inverseJoinColumns = @JoinColumn(name = "user_id"))
List<User> users = new ArrayList<User>();
```

```
@ManyToMany(mappedBy = "users", fetch = FetchType.EAGER)
List<Trip> trips = new ArrayList<Trip>();
```

### @Embeddable and @Embedded

The @Embeddable and @Embedded annotations in Hibernate are used to map an object's properties to columns in a database table. These annotations are used in combination to allow the properties of one class to be included as a value type in another class and then be persisted in the database as part of the containing class.

```
@Embedded
private Source source;

@Embedded
private Destination destination;
```

```
5 @Embeddable
6 public class Destination {
7     @Column(name = "destination_country")
8     protected String destinationCountryName;
9
10    @Column(name = "destination_city")
11    protected String destinationCityName;
12
13    @Column(name = "destination_airport")
14    protected String destinationAirportName;
15
16    public String getDestinationCountryName() {
17        return destinationCountryName;
18    }
19 }
```

```
5
6 @Embeddable
7 public class Source {
8
9     @Column(name = "source_country")
10    protected String sourceCountryName;
11
12    @Column(name = "source_city")
13    protected String sourceCityName;
14
15    @Column(name = "source_airport")
16    protected String sourceAirportName;
17 }
```

## Servlets

Servlets are the Java programs that run on the Java-enabled web server or application server. They are used to handle the request obtained from the webserver, process the request, produce the response, and then send a response back to the webserver.

Properties of Servlets are as follows:

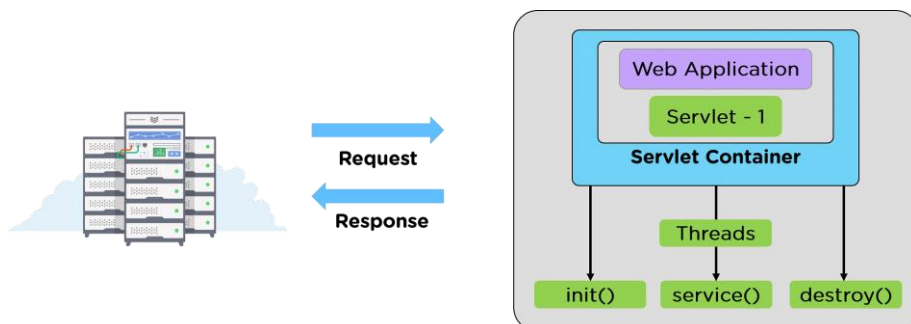
- Servlets work on the server-side.
- Servlets are capable of handling complex requests obtained from the webserver.

Execution of Servlets basically involves six basic steps:

- The clients send the request to the webserver.
- The web server receives the request.
- The web server passes the request to the corresponding servlet.
- The servlet processes the request and generates the response in the form of output.
- The servlet sends the response back to the webserver.
- The web server sends the response back to the client and the client browser displays it on the screen.

There are several varieties of interfaces and classes available in the Servlet API. Some of them are as follows:

- HTTP Servlet
- Generic Servlet
- Servlet Request
- Servlet Response



To write a Servlet, the user needs first to implement the Servlet Interface, directly or indirectly, using the following import command.

```
import javax.servlet.*;
```

Once the Servlet interface is imported, and we inherit the HTTP Class, we begin with the Java Servlet's life cycle.

In the life cycle of a servlet, we have mainly three stages, which are mentioned below.

- `init()`
- `service()`
- `destroy()`

We call these methods at their respective stages. The methods are resolved by generating the essential threads for the process to get executed.

The `service()` method is the heart of the life cycle of a Java Servlet. Right after the Servlet's initialization, it encounters the service requests from the client end.

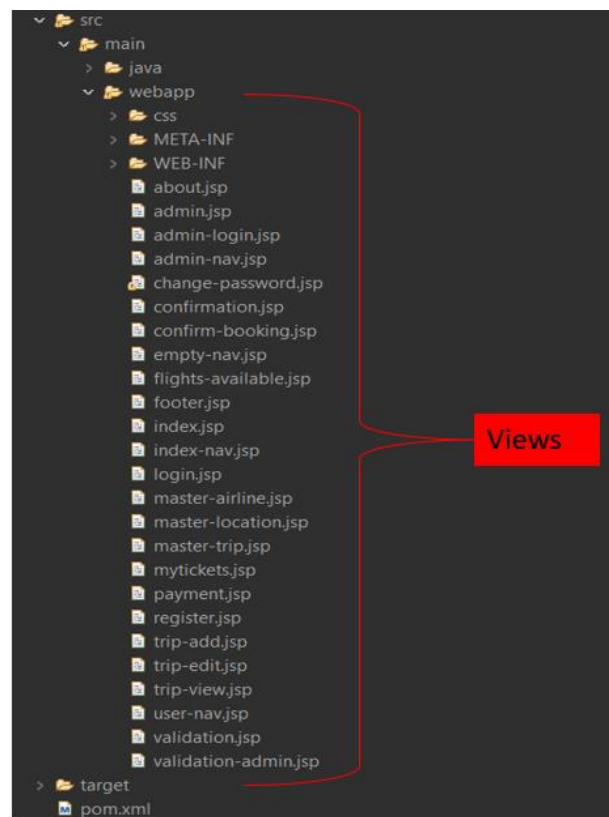
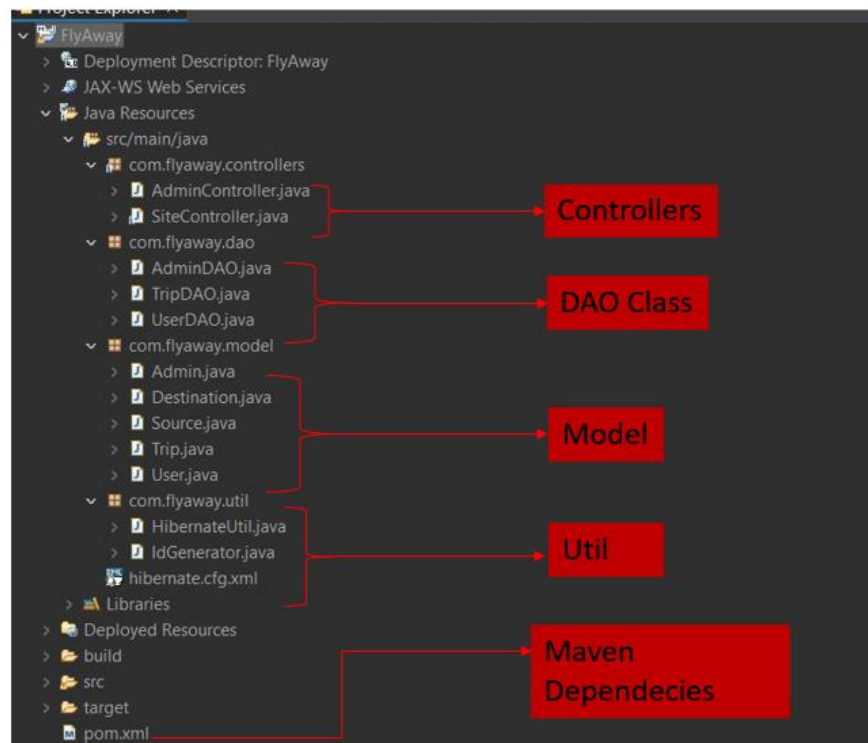
The client may request various services like:

- GET
- PUT
- UPDATE
- DELETE

The `service()` method takes responsibility to check the type of request received from the client and respond accordingly by generating a new thread or a set of threads per the requirement and implementing the operation through the following methods.

- `doGet()` for GET
- `doPut()` for PUT
- `doUpdate()` for UPDATE
- `doDelete()` for DELETE

## Project Structure

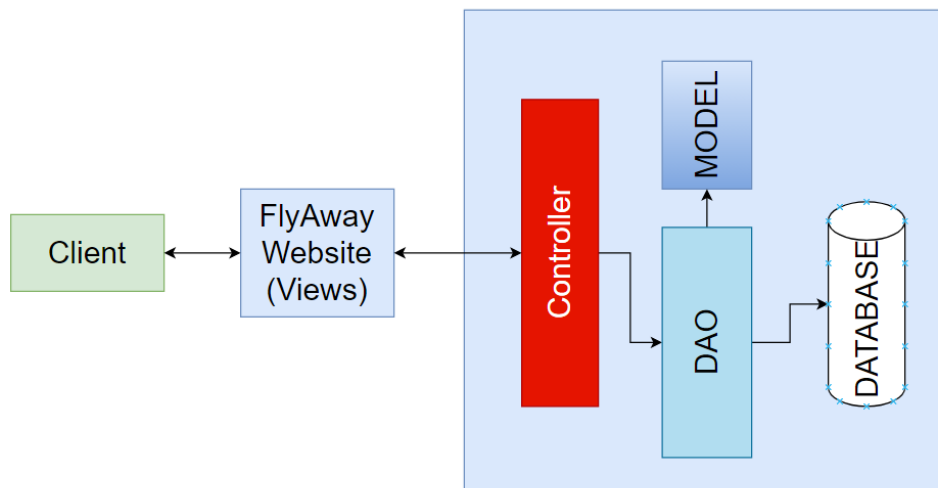


## Control Statements

The program utilizes the following control statements to direct the desired logic:

- *while* loop – Controls the program flow by prompting the User for main menu and the business options sub-menu, performing the desired operations, and terminates when the User wishes to quite the program.
- *Switch* statement Executes the desired code statements associated with the main menu and the business level options sub-menu based on the value entered by the user.

## Flow Chart



*Application Flow Chart (page 3:3)*

### Future Improvement Areas

- Passenger details can also be added by making it as a @onetoMany association with User and Trip models
- User can also change password same as the admin
- As of now user can only select from the list of cities available in the menu in future we can dependent dropdown with the help of JavaScript.
- As of now user can only select a one way trip, Round trip can also be added as a feature in the future.

### GitHub Repository

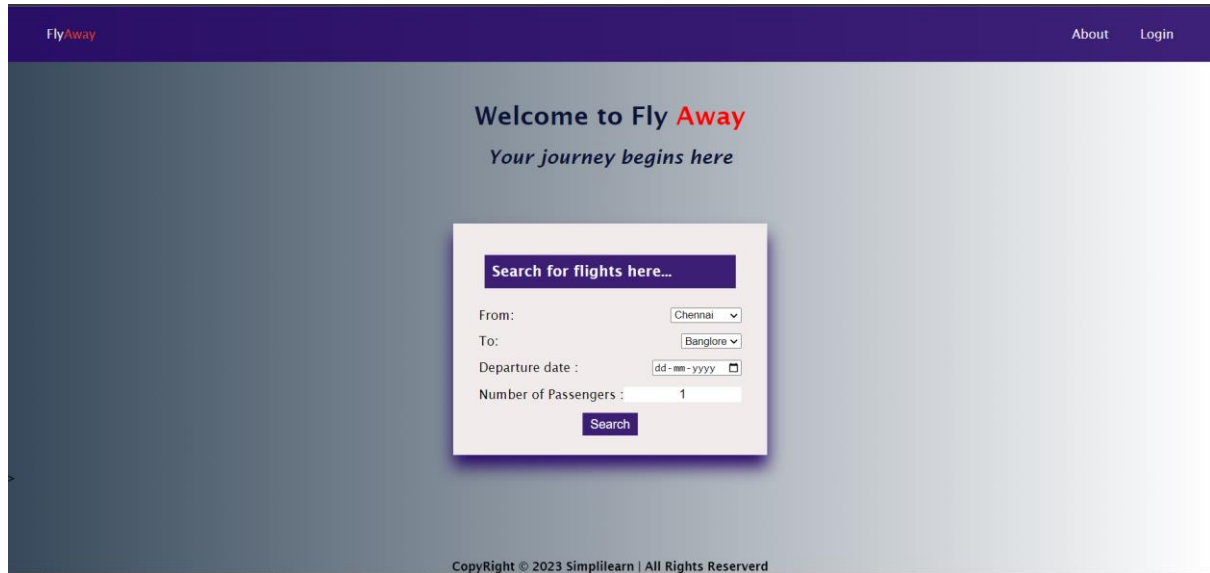
I have pushed my code and associated documentation to the following GitHub repository:

<https://github.com/Thiru97/Simplilearn2023.git>

# FlyAway -An Airline Booking Portal

## Web Application Screenshots

### Home Page



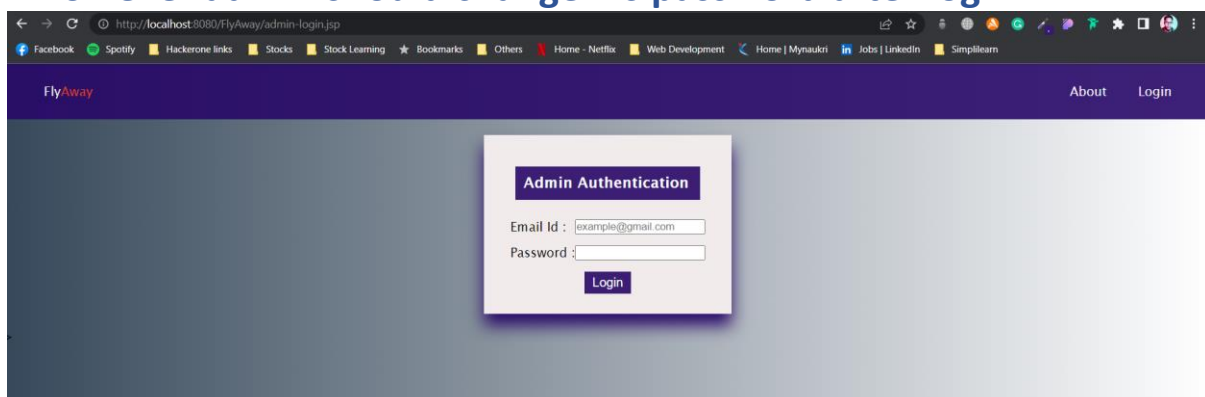
Admin can login by changing the url as  
**localhost:8080/FlyAway/admin-login.jsp**

Where he has to enter his credentials

**Default Email Id: admin@flyaway.com**

**Default Password: admin**

**However admin should change his password after login**

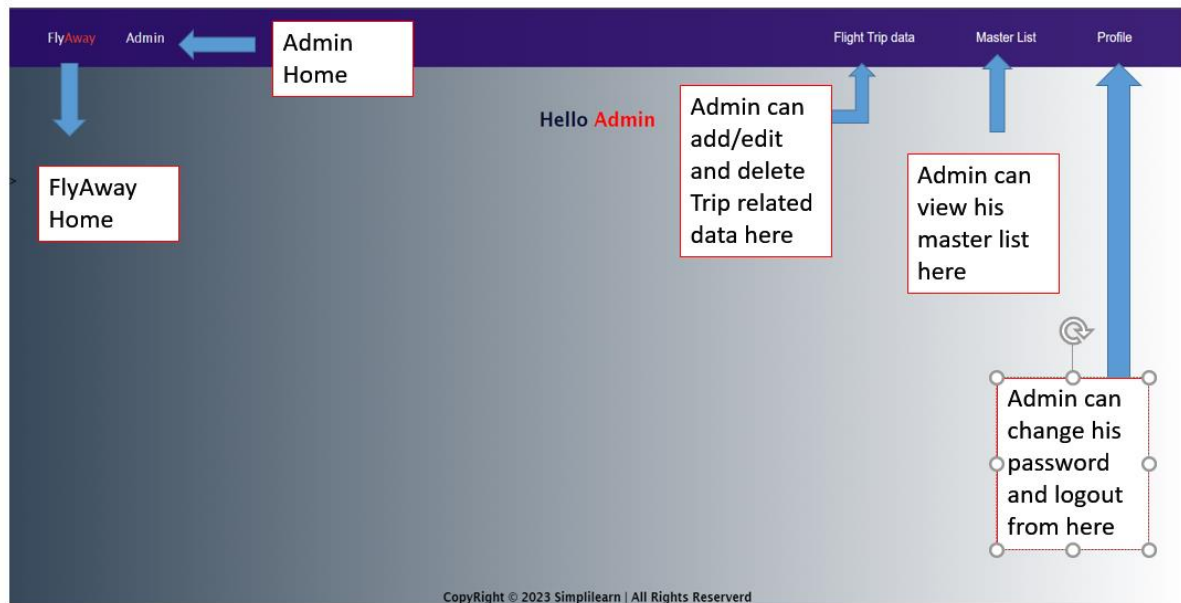




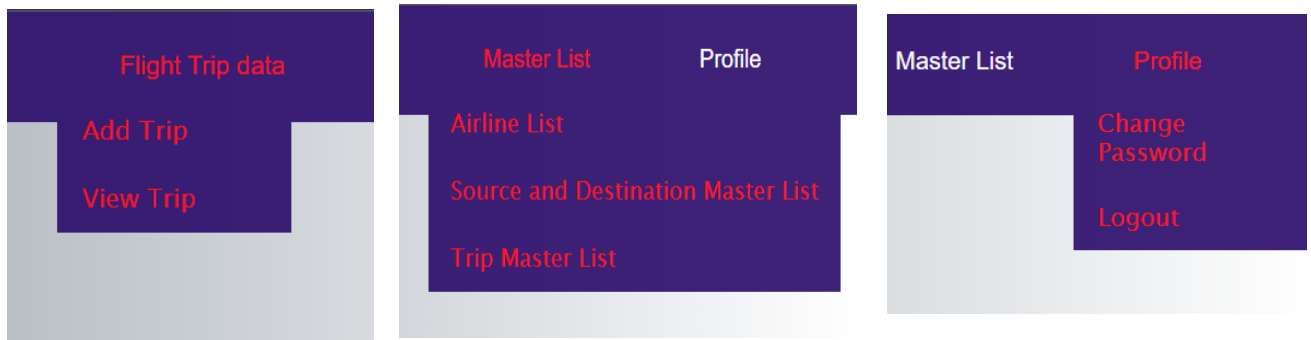
# FlyAway -An Airline Booking Portal

## Admin Dashboard

NOTE: If session is inactive for more than 5 minutes admin will be logged out



## Admin Navigation Bar Drop Down menu



## FlyAway -An Airline Booking Portal

### Flight Trip Data

Admin can add trip details here:

**Note: Trip ID and Flight number are generated by IdGenerator.java**

#### Add Trip Details

Source :

Country Name:

City Name:

Airport Name:

Departure date:

Departure time:

Destination :

Country Name:

City Name:

Airport Name:

Arrival date:

Arrival time:

Airline Details :

Airline Name:

Aircraft Manufacturer:

Max Passengers:

Fare per Passenger:

Add Trip

### View Trip data

Admin can edit / delete trips from here. He can also use search by to narrow the results

Trip related operations						
Search by : <input type="text" value="Select"/> Please enter value : <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Refresh List"/> <input type="button" value="Add More Trip"/>						
TripId	Actions	Source City Name	Destination City Name	Airline	Max Passengers	Ticket Price
TRIP3714	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>	Chennai	Banglore	Indigo	30	2000
TRIP8834	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>	Chennai	Banglore	Spice Jet	30	1750
TRIP2107	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>	Chennai	Banglore	Akasa Air	30	1850
TRIP6519	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>	Banglore	Chennai	Indigo	30	2000
TRIP656	<input type="button" value="EDIT"/> <input type="button" value="DELETE"/>	Banglore	Chennai	Spice Jet	30	1750

## Edit Trip data

If admin clicks on edit respective trip data will be retrieved and can be edited

### Edit Trip Details

Source :

Country Name:

City Name:

Airport Name:

Departure date:

Departure time:

Destination :

Country Name:

City Name:

Airport Name:

Arrival date:

Arrival time:

Airline Details :

Airline Name:

Max Passengers :

Aircraft Manufacturer :

Fare per Passenger :

Save

## Airline Master List

Flight Number	Airline Name	Aircraft Manufacturer	Max Passengers	Ticket Price
IN9295	Indigo	Airbus	30	2000
SP9814	Spice Jet	Bombardier	30	1750
AK377	Akasa Air	Boeing	30	1850
IN6588	Indigo	Airbus	30	2000
SP8058	Spice Jet	Bombardier	30	1750
IN2816	Indigo	Airbus	35	5000
IN6244	Indigo	Airbus	45	4500
AI6340	AirIndi	Bombardier	25	3500
SP4178	SpiceJet	Bombardier	30	4500
IN5636	Indigo	Boeing	30	4000
EM6233	Emirates	Airbus	50	25000
EM5594	Emirates	Airbus	50	25000
QA7097	Qatar	Airbus	60	100000
QA6558	Qatar	Airbus	60	100000

## Source and Destination Master List

FlyAway

Admin

Flight Trip data

Master List

Profile

Source and Destination data

Trip Id	Source Country	Source city	Source Airport	Destination Country	Destination City	Destination Airport
TRIP3714	India	Chennai	Chennai International Airport	India	Banglore	Banglore International Airport
TRIP8834	India	Chennai	Chennai International Airport	India	Banglore	Banglore International Airport
TRIP2107	India	Chennai	Chennai International Airport	India	Banglore	Banglore International Airport
TRIP6519	India	Banglore	Banglore International Airport	India	Chennai	Chennai International Airport
TRIP656	India	Banglore	Banglore International Airport	India	Chennai	Chennai International Airport
TRIP3773	India	Banglore	Banglore International Airport	India	Delhi	Delhi International Airport
TRIP3824	India	Mumbai	Mumbai International Airport	India	Chennai	Chennai International Airport
TRIP9957	India	Mumbai	Mumbai International Airport	India	Banglore	Banglore International Airport
TRIP7893	India	Delhi	Delhi International Airport	India	Chennai	Chennai International Airport
TRIP9165	India	Trivandrum	Trivandrum International Airport	India	Chennai	Chennai International Airport
TRIP9335	India	Chennai	Chennai International Airport	United Arab Emirates	Dubai	Dubai International Airport
TRIP631	United Arab Emirates	Dubai	Dubai International Airport	India	Chennai	Chennai International Airport
TRIP7801	United Arab Emirates	Dubai	Dubai International Airport	United States	Seattle	Seattle International Airport
TRIP5571	United States	Seattle	Seattle International Airport	United Arab Emirates	Dubai	Dubai International Airport

CopyRight © 2023 Simplilearn | All Rights Reserved

## Trip Master List

FlyAway

Admin

Flight Trip data

Master List

Profile

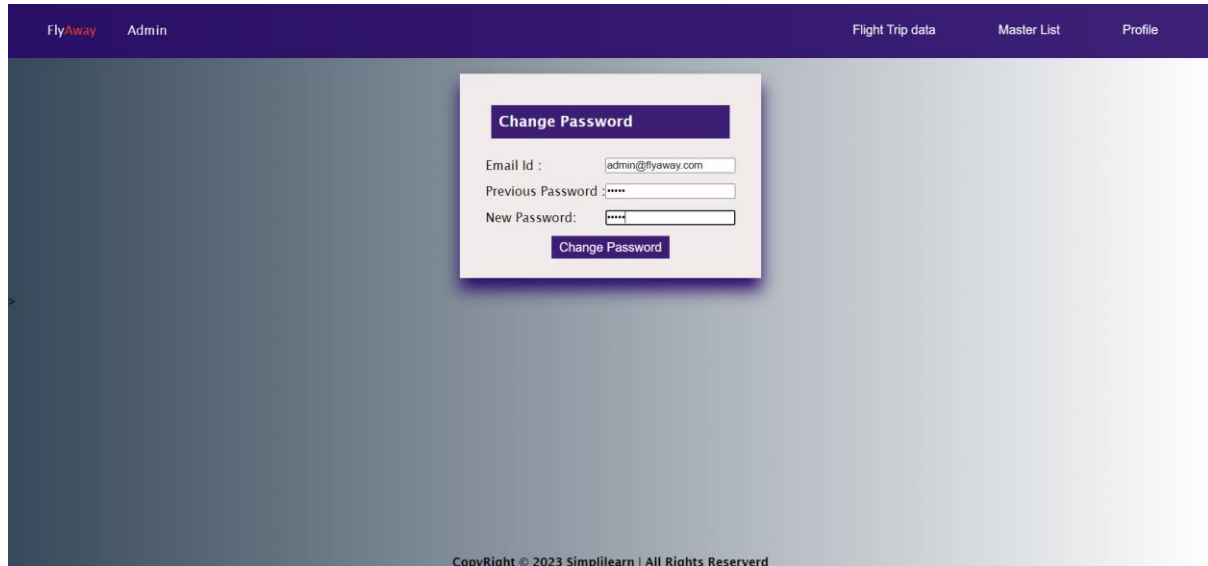
Trip data

Trip Id	Flight Number	Airline	Source Country Name	Source City Name	Source Airport Name	Departure Date	Departure Time	Destination Country Name	Destination City Name	Destination Airport Name	Arrival Date	Arrival Time	Max Passengers
TRIP3714	IN9295	Indigo	India	Chennai	Chennai International Airport	22-05-2023	07:00	India	Banglore	Banglore International Airport	22-05-2023	08:00	30
TRIP8834	SP9814	Spice Jet	India	Chennai	Chennai International Airport	22-05-2023	07:15	India	Banglore	Banglore International Airport	22-05-2023	08:15	30
TRIP2107	AK377	Akasa Air	India	Chennai	Chennai International Airport	22-05-2023	07:30	India	Banglore	Banglore International Airport	22-05-2023	08:30	30
TRIP6519	IN6588	Indigo	India	Banglore	Banglore International Airport	22-05-2023	08:30	India	Chennai	Chennai International Airport	22-05-2023	09:30	30
TRIP656	SP8058	Spice Jet	India	Banglore	Banglore International Airport	22-05-2023	08:45	India	Chennai	Chennai International Airport	22-05-2023	09:45	30
TRIP3773	IN2816	Indigo	India	Banglore	Banglore International	22-05-2023	08:00	India	Delhi	Delhi International	22-05-2023	11:30	35

CopyRight © 2023 Simplilearn | All Rights Reserved

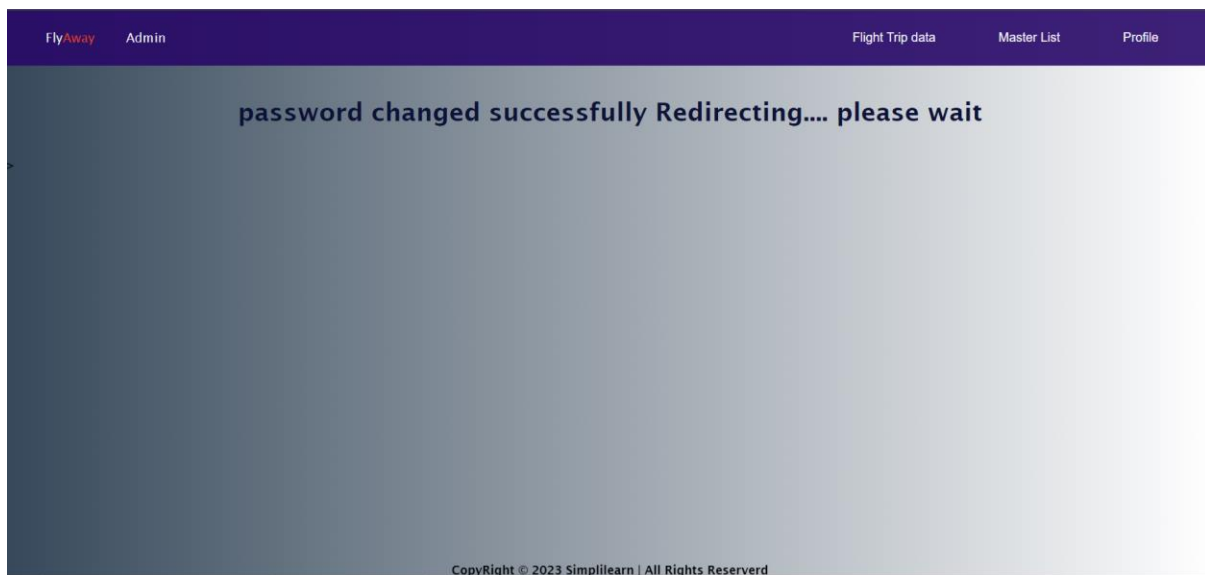
### Change Password

If the admin changes his password he will be logged out and prompted that he is logged out and redirected to login page within 3 seconds of delay and asked to login again



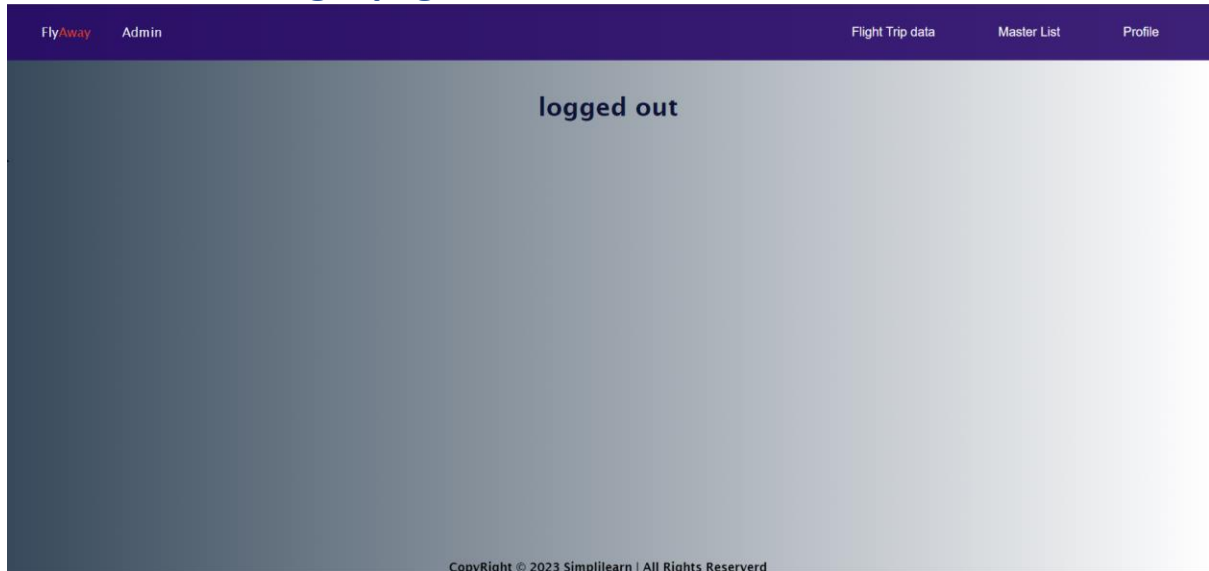
The screenshot shows the FlyAway Admin interface. The top navigation bar is dark purple with the FlyAway logo and 'Admin' on the left, and 'Flight Trip data', 'Master List', and 'Profile' on the right. The main content area is a light gray gradient. In the center, there is a white modal box titled 'Change Password'. Inside the modal, there are three input fields: 'Email Id' with the value 'admin@flyaway.com', 'Previous Password' with masked characters '\*\*\*\*', and 'New Password' with masked characters '\*\*\*\*'. Below these fields is a purple button labeled 'Change Password'. At the bottom of the main content area, there is a small copyright notice: 'CopyRight © 2023 Simplilearn | All Rights Reserved'.

### After Changing Password

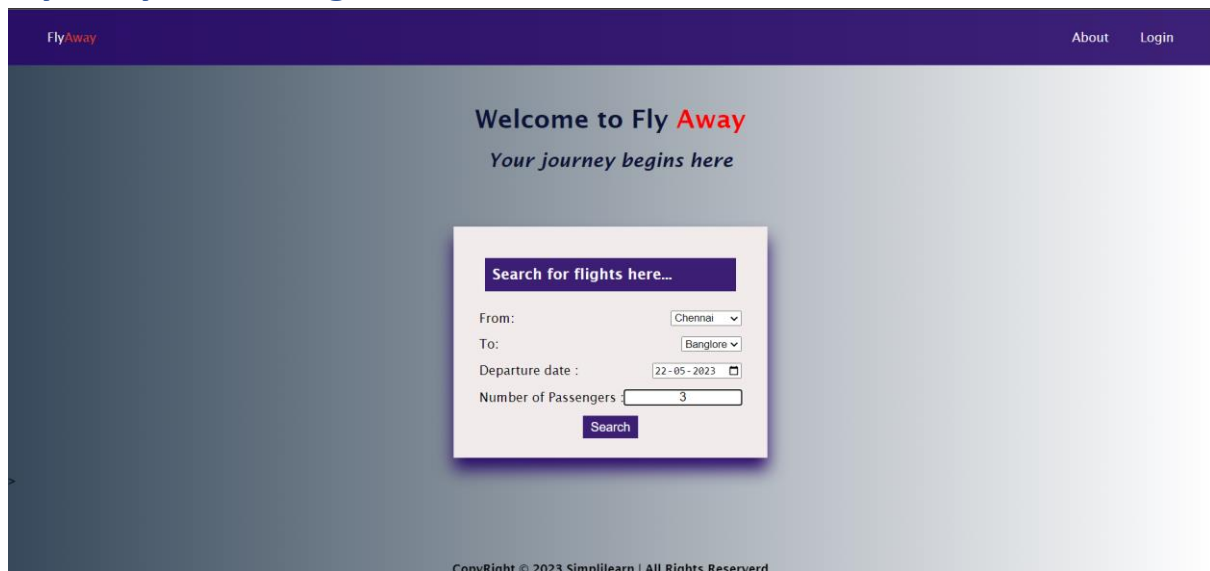


## Logout

If admin clicks logout he will be prompted and automatically redirected to login page within 3 sec timer



## FlyAway Home Page



## FlyAway Search Results

Flights from... Chennai to Bangalore									
Flight Number	Airline	Departure	Departure Date	Departure Time	Arrival	Arrival Date	Arrival time	Fare	Book
IN9295	Indigo	Chennai	22-05-2023	07:00	Banglore	22-05-2023	08:00	2000	<a href="#">Buy</a>
SP9814	Spice Jet	Chennai	22-05-2023	07:15	Banglore	22-05-2023	08:15	1750	<a href="#">Buy</a>
AK377	Akasa Air	Chennai	22-05-2023	07:30	Banglore	22-05-2023	08:30	1850	<a href="#">Buy</a>

Buying tickets = when user clicks “Buy” he will be automatically asked to either login or register himself. I am selecting Trip 1 where Airline operator is Indigo

Please Login to continue									
Email Id : <input type="text" value="example@gmail.com"/>									
Password : <input type="password"/>									
<a href="#">Login</a> <a href="#">New User? Register</a>									

## Register

### Registering myself as John Doe

The screenshot shows the FlyAway website's registration form. The form is titled 'Register' and is located in the center of the page. It contains the following fields: First Name (John), Last Name (Doe), Age (20), Email Address (john@gmail.com), Password (\*\*\*\*\*), and Confirm Password (\*\*\*\*\*). A 'Submit' button is located at the bottom of the form. The website's navigation bar at the top includes the FlyAway logo, 'About', and 'Login' links. The footer contains the text 'Copyright © 2023 Simplilearn | All Rights Reserved'.

Flight Number	Airline	Departure	Departure Date and time	Arrival	Arrival date and time	Number of passengers
IN9295	Indigo	Chennai International Airport	22-05-2023 07:00	Banglore International Airport	22-05-2023 08:00	3

Fare Per Passenger	No of Passenger	Total Amount
2000	3	6000

Confirm Cancel

When I click Register I will be logged in with the same credentials and redirected to proceed with my ticket booking. As I am logged in now please note the navigation bar has changed

The screenshot shows the FlyAway website's confirm booking page. The navigation bar at the top includes the FlyAway logo, 'About', 'My Tickets', and 'Logout' links. A blue arrow points to the 'My Tickets' link. The main content area is titled 'Confirm Booking' and contains two tables. The first table shows flight details: Flight Number (IN9295), Airline (Indigo), Departure (Chennai International Airport), Departure Date and time (22-05-2023 07:00), Arrival (Banglore International Airport), Arrival date and time (22-05-2023 08:00), and Number of passengers (3). The second table shows fare details: Fare Per Passenger (2000), No of Passenger (3), and Total Amount (6000). At the bottom of the page, there are 'Confirm' and 'Cancel' buttons. The footer contains the text 'Copyright © 2023 Simplilearn | All Rights Reserved'.

Flight Number	Airline	Departure	Departure Date and time	Arrival	Arrival date and time	Number of passengers
IN9295	Indigo	Chennai International Airport	22-05-2023 07:00	Banglore International Airport	22-05-2023 08:00	3

Fare Per Passenger	No of Passenger	Total Amount
2000	3	6000

Confirm Cancel

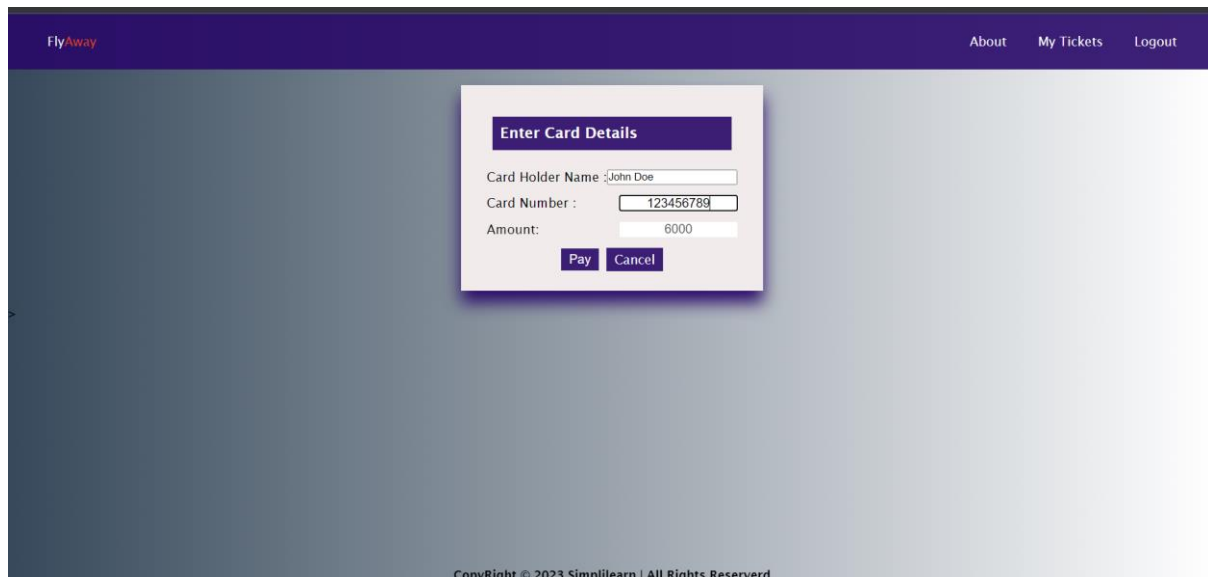


## FlyAway -An Airline Booking Portal

I have chosen no of passengers as 3 in the search menu so I am asked to pay ticket price for all the passengers

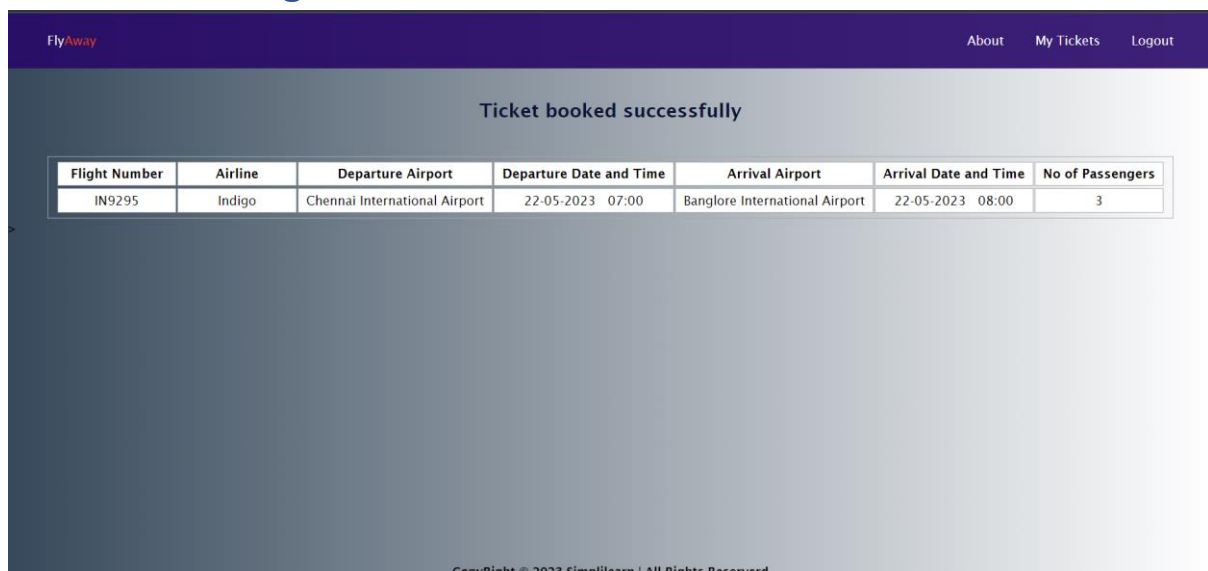
### Register

### Dummy Payment Gateway



The screenshot shows the FlyAway website with a dark purple header containing the logo and links for 'About', 'My Tickets', and 'Logout'. A modal window titled 'Enter Card Details' is centered on the screen. It contains three input fields: 'Card Holder Name' with the value 'John Doe', 'Card Number' with the value '123456789', and 'Amount' with the value '6000'. Below these fields are two buttons: 'Pay' and 'Cancel'. The footer of the page reads 'Copyright © 2023 Simplilearn | All Rights Reserved'.

### Confirm booking and confirmation status



The screenshot shows the FlyAway website with a dark purple header containing the logo and links for 'About', 'My Tickets', and 'Logout'. The main content area displays the message 'Ticket booked successfully'. Below this message is a table with the following data:

Flight Number	Airline	Departure Airport	Departure Date and Time	Arrival Airport	Arrival Date and Time	No of Passengers
IN9295	Indigo	Chennai International Airport	22-05-2023 07:00	Banglore International Airport	22-05-2023 08:00	3

The footer of the page reads 'Copyright © 2023 Simplilearn | All Rights Reserved'.

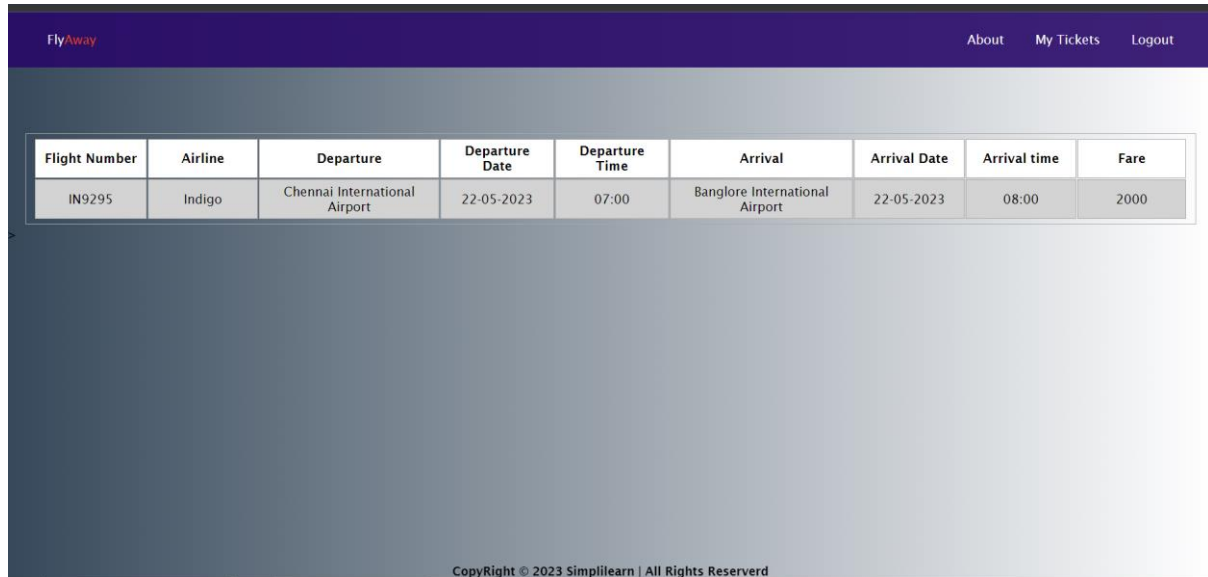
@ManytoMany trip details are also successfully updated in the database

## FlyAway -An Airline Booking Portal



	trip_id	user_id
▶	9	8
	9	9

User can also see the ticket details reflected in his My Tickets page

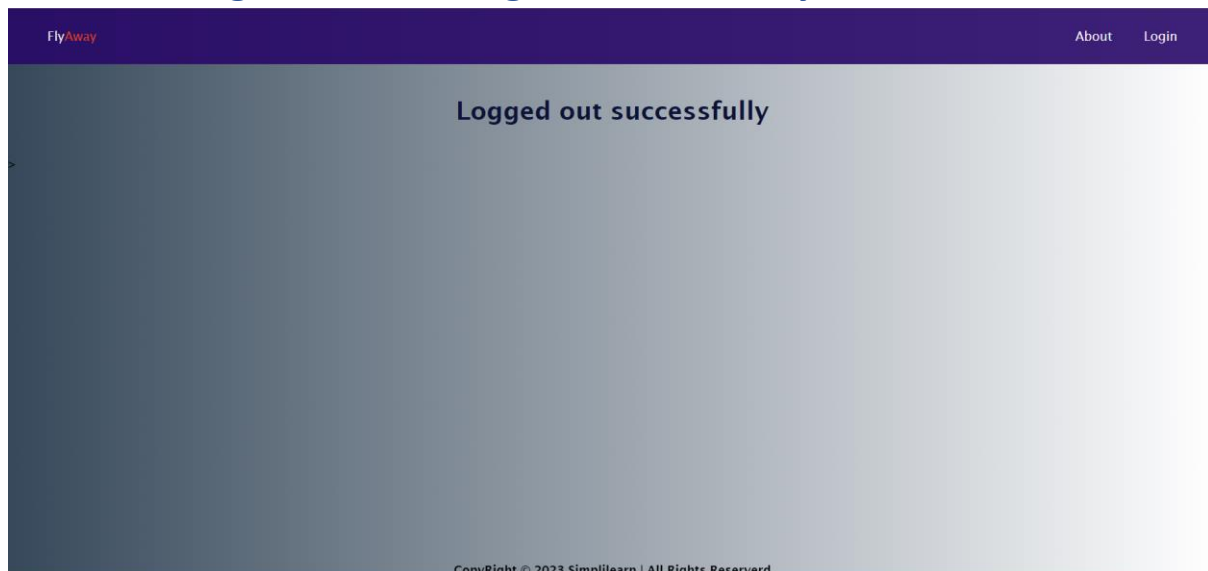


Flight Number	Airline	Departure	Departure Date	Departure Time	Arrival	Arrival Date	Arrival time	Fare
IN9295	Indigo	Chennai International Airport	22-05-2023	07:00	Banglore International Airport	22-05-2023	08:00	2000

Copyright © 2023 Simplilearn | All Rights Reserved

## Logout

When user clicks logout he will be logged out and prompted and the navigation bar changes automatically



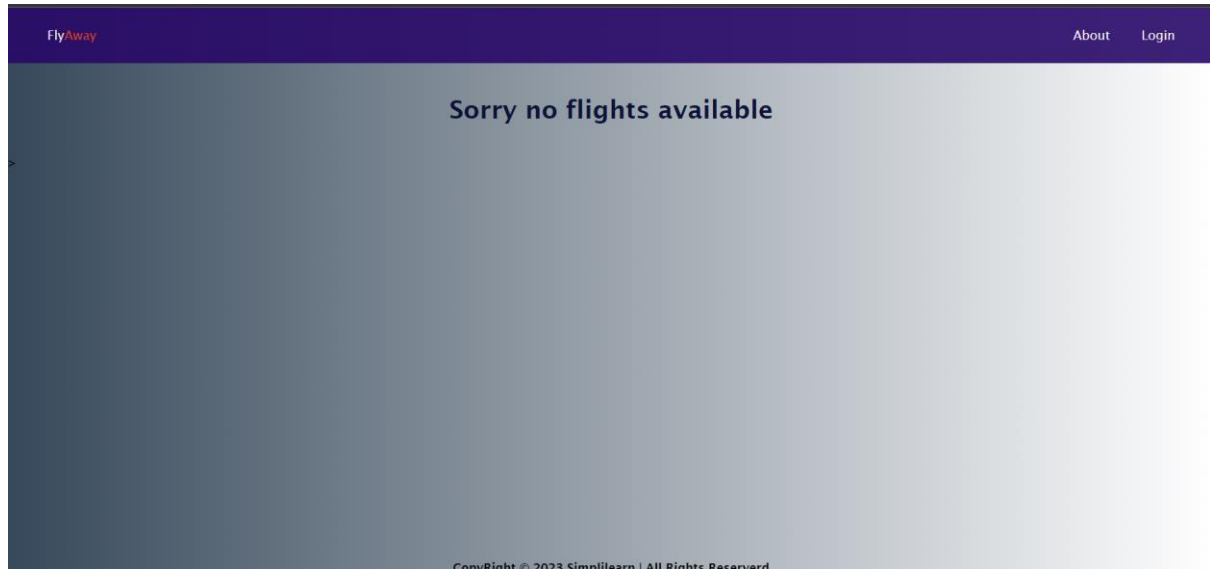
Flight Number	Airline	Departure	Departure Date	Departure Time	Arrival	Arrival Date	Arrival time	Fare
IN9295	Indigo	Chennai International Airport	22-05-2023	07:00	Banglore International Airport	22-05-2023	08:00	2000

Logged out successfully

Copyright © 2023 Simplilearn | All Rights Reserved

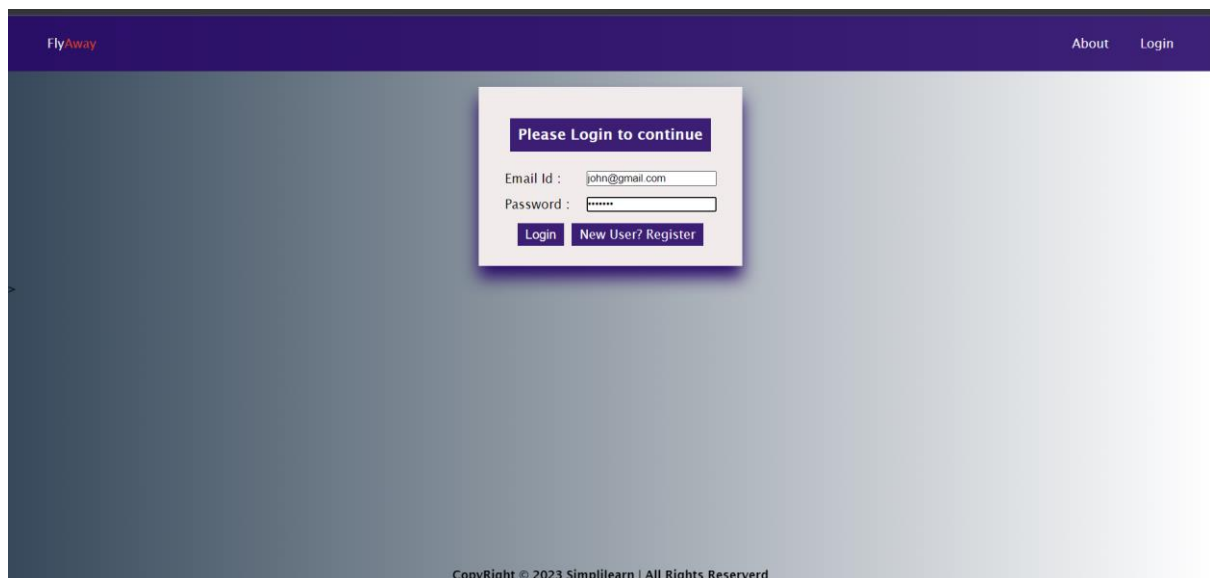
## Miscellaneous and Error Pages

If user query flight details are not available



While booking flight user can also log in

To show this I again logged In as John Doe and booked a new ticket



## John Doe's Ticket History

FlyAway		About My Tickets Logout						
Flight Number	Airline	Departure	Departure Date	Departure Time	Arrival	Arrival Date	Arrival time	Fare
IN9295	Indigo	Chennai International Airport	22-05-2023	07:00	Banglore International Airport	22-05-2023	08:00	2000
SP9814	Spice Jet	Chennai International Airport	22-05-2023	07:15	Banglore International Airport	22-05-2023	08:15	1750

CopyRight © 2023 Simplilearn | All Rights Reserved

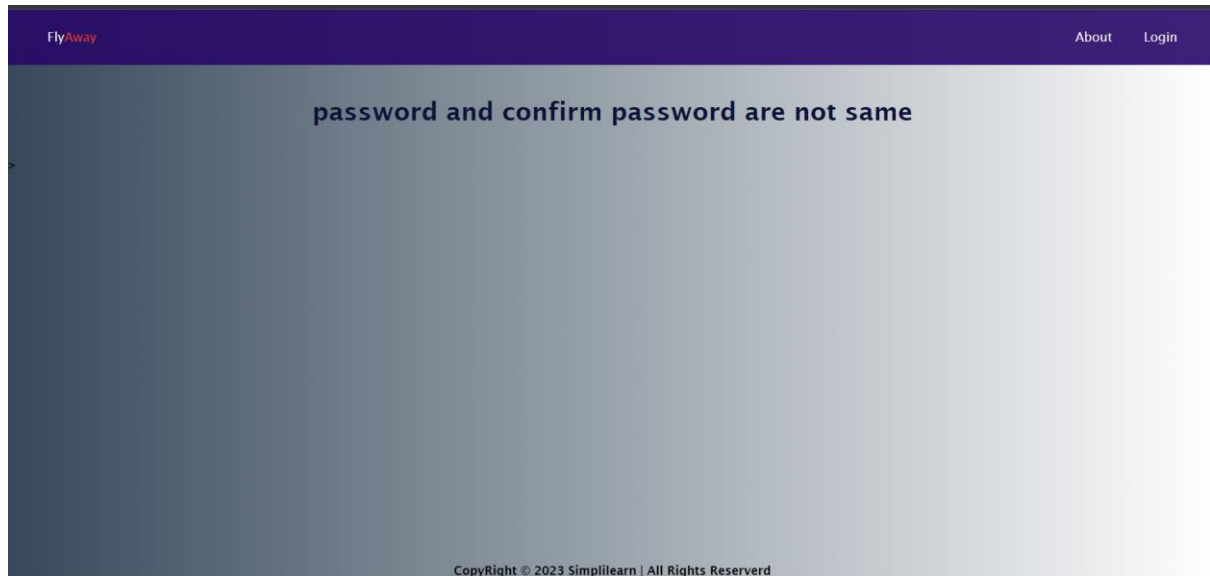
While login in user can enter a wrong password he will be prompted the same

FlyAway		About Login	
Username or password is wrong			

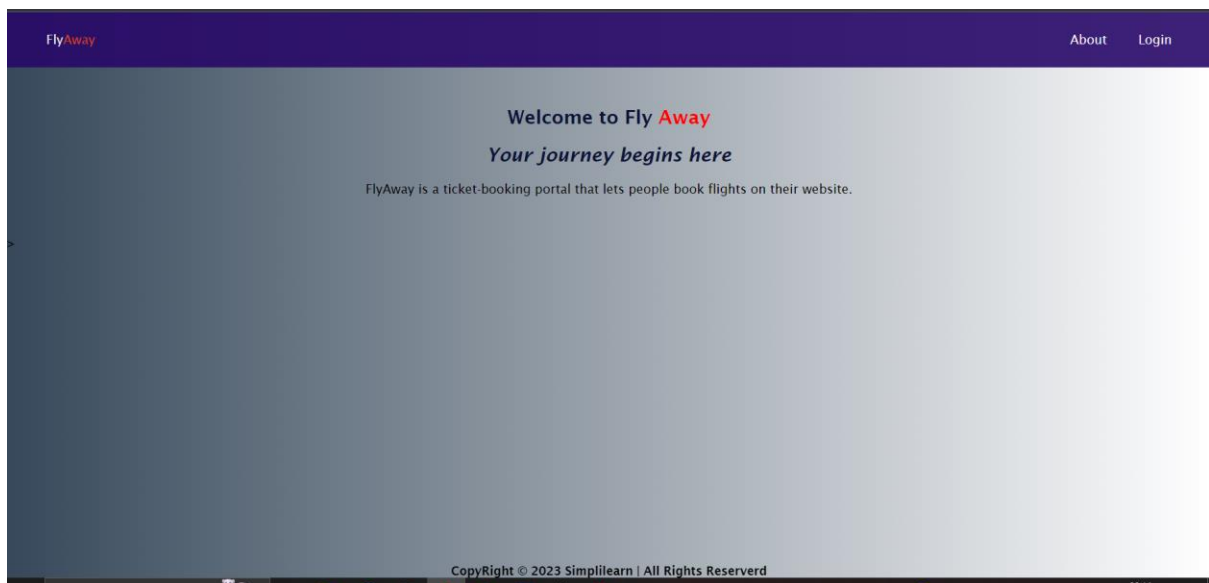
CopyRight © 2023 Simplilearn | All Rights Reserved

## FlyAway -An Airline Booking Portal

While registering user can enter password and confirm password different

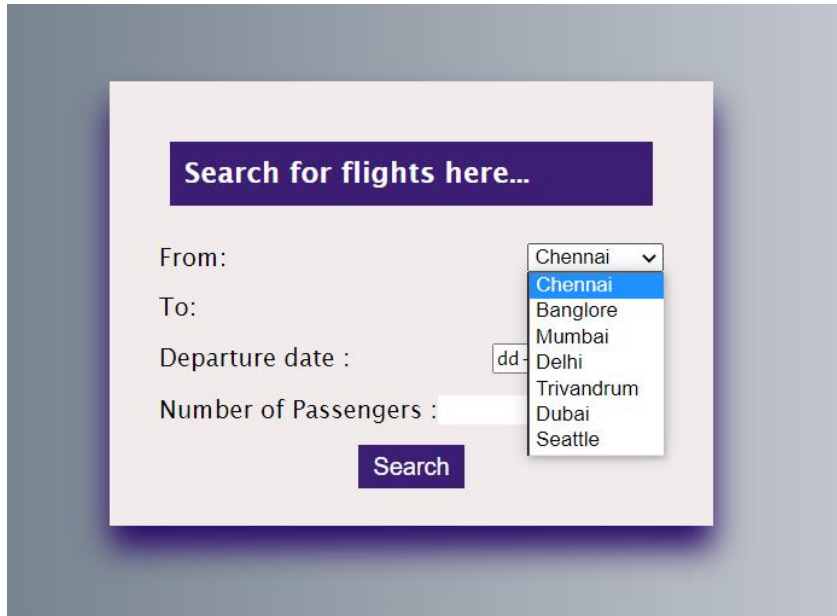


## About Page



### Search Query

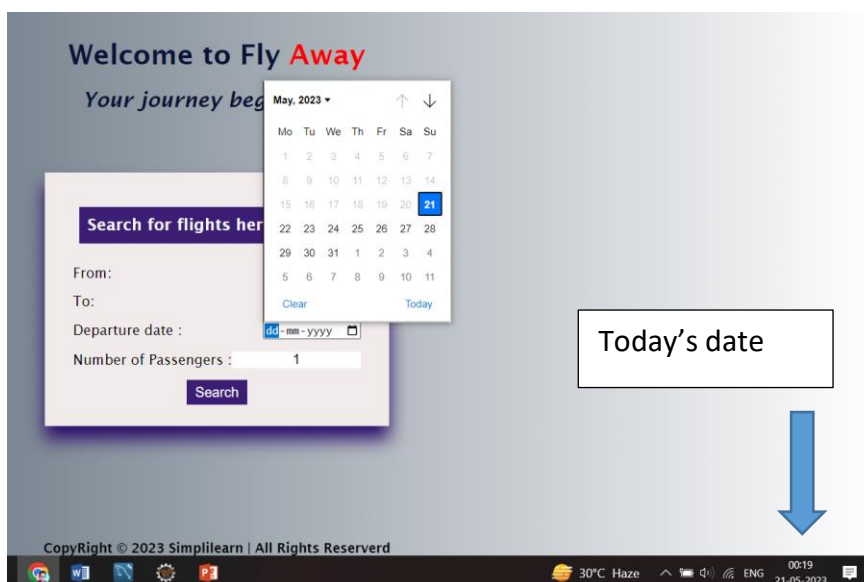
When Admin adds Trip data in the backend distinct cities are displayed respectively



The screenshot shows a search form titled "Search for flights here...". It includes fields for "From:", "To:", "Departure date :", and "Number of Passengers :". A dropdown menu is open next to the "To:" field, displaying a list of cities: Chennai, Bangalore, Mumbai, Delhi, Trivandrum, Dubai, and Seattle. The "Search" button is located at the bottom right of the form.

### Form Date field:

User and Admin can only use present date, past dates are greyed out



The screenshot shows the same search form as before, but with a date picker open for the "Departure date :" field. The date picker displays a calendar for May 2023. A callout box labeled "Today's date" points to the date 21-05-2023 in the system tray at the bottom right of the screen. The system tray also shows the current time as 00:19 and the date as 21-05-2023.

## Search Operations in Admin field

FlyAwayAdmin

Flight Trip dataMaster ListProfile

### Trip related operations

Search by : SourcePlease enter value : MumbaiSearchRefresh ListAdd More Trip

Tripld	Actions	Source City Name	Destination City Name	Airline	Max Passengers	Ticket Price
TRIP3824	<a href="#">EDIT</a> <a href="#">DELETE</a>	Mumbai	Chennai	Indigo	45	4500
TRIP9957	<a href="#">EDIT</a> <a href="#">DELETE</a>	Mumbai	Banglore	AirIndi	25	3500

CopyRight © 2023 Simplilearn | All Rights Reserved