

# Screenshoots\_for\_capstone\_project

## LoginPage:

FLYAWAY - MEMBER LOGIN

Home | Login/Signup |

Email id*	lokeshd@gmail.com
Password*	
<input type="button" value="Login"/> <a href="#">Not a member? Signup</a>	

(C) FlyAway 2019



## HOME:

FLYAWAY HOMEPAGE

Home | Dashboard | Edit Profile | Your Bookings | Logout

Source	Destination	Airline	Duration	Cost	
Bangalore	Chennai	AirAsia	Dep.8 pm Arr.9.30pm (1.30 hours)	4500.00	<a href="#">Book Flight</a>
Bangalore	Hyderabad	Vistara	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00	<a href="#">Book Flight</a>

(C) FlyAway 2019



From: Bangalore To: Chennai

Source	Destination	Airline	Duration	Cost	
Bangalore	Chennai	AirAsia	Dep.8 pm Arr.9.30pm (1.30 hours)	4500.00	<a href="#">Book Flight</a>

(C) FlyAway 2019



## Destination \_confirmed:

23°C Mostly cloudy

Search

Home | Dashboard | Edit Profile | Your Bookings | Logout

Your Flight by Vistara , from Bangalore to Hyderabad, on Thu Mar 30 17:16:18 UTC 2023 has been booked for 2500.00.

Check your [See Your Bookings](#)

(C) FlyAway 2019



## For\_completing\_booking:

Your card will be charged an amount of 2500.00

Please confirm your flight details below:

**Date:** Thu Mar 30 17:16:18 UTC 2023  
**Airline:** Vistara  
**Sector:** Bangalore - Hyderabad  
**Departure:** 8 AM  
**Arrival:** 10:30 AMx  
**Duration:** 2.5 HOURS

[Click to complete booking](#)

(C) FlyAway 2019

## Final\_flight\_booking:

Total 16 bookings

Date Of Booking	Source	Destination	Airline	Cost
2023-03-30 17:16:18.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 17:14:29.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 17:12:14.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 16:59:39.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 16:51:34.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 16:49:04.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 16:25:17.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 16:23:59.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 16:20:31.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 15:30:36.0	Bangalore	Hyderabad	Dep.8 AM Arr.10:30 AMx (2.5 HOURS)	2500.00
2023-03-30 15:02:46.0	Bangalore	Chennai	Dep.8 pm Arr.9.30pm (1.30 hours)	4500.00
2023-03-30 06:47:42.0	Bangalore	Chennai	Dep.8 pm Arr.9.30pm (1.30 hours)	4500.00
2023-02-20 17:10:06.0	Bangalore	Chennai	Dep.8 pm Arr.9.30pm (1.30 hours)	4500.00

[Click to complete booking](#)

(C) FlyAway 2019

## JDBC\_user\_inputs:

A screenshot of a Windows desktop environment. At the top, there is a taskbar with several icons. In the center, a Microsoft Edge browser window is open, displaying a simple HTML page with an 'h3' tag containing 'FLYAWAY HOMEPAGE'. Above the browser, an Eclipse IDE window is visible, showing the results of a TestNG test named 'Testrunn'. The console output shows the following information:

```
<terminated> Testrunn [TestNG] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 30, 2023, 5:25:45 PM)
[RemoteTestNG] detected TestNG version 7.4.0
Lokesh JD
bangalore
lokeshjd@gmail.com
lokii12345
Status_code _for_home:200

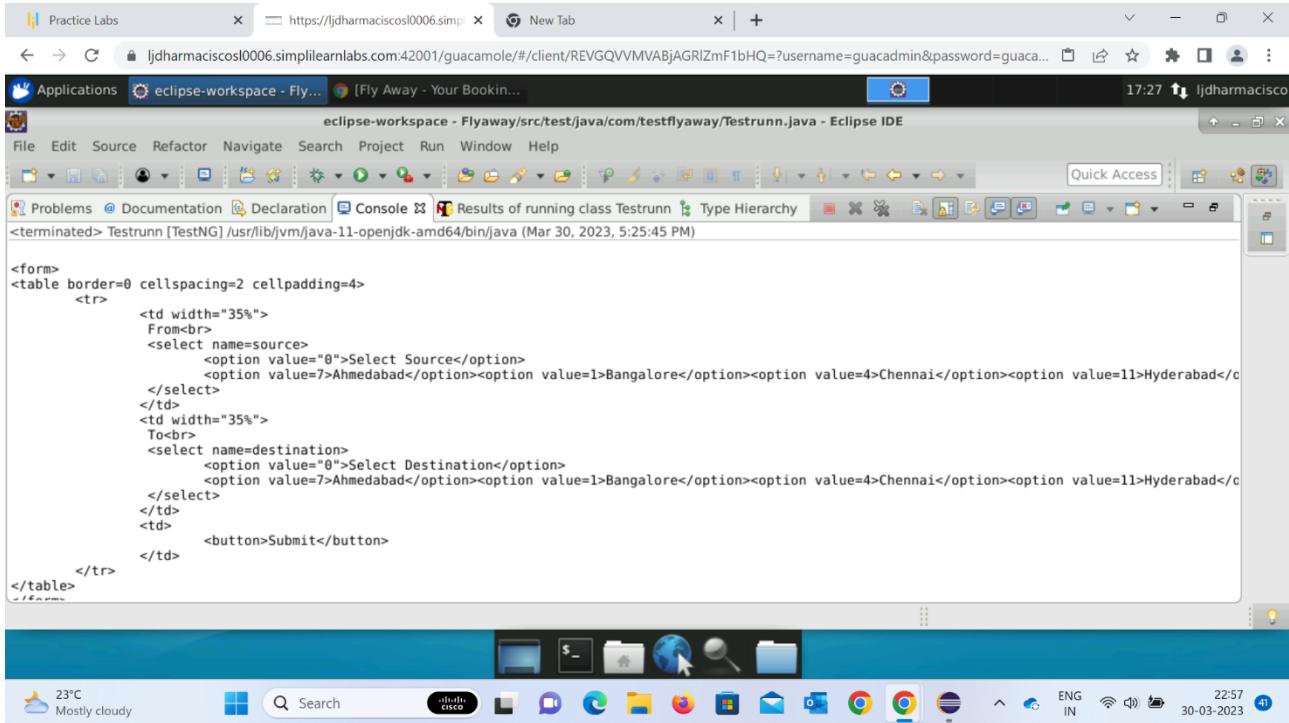
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Fly Away </title>
</head>
<body>

<h3> FLYAWAY HOMEPAGE </h3>
```

A screenshot of a Windows desktop environment, similar to the one above. A Microsoft Edge browser window shows a more complex HTML page with multiple rows of flight information. Each row contains five columns: destination, source, airline, departure time, arrival time, and price. Each row also includes a link to book the flight. The Eclipse IDE window above it displays the same TestNG test results as the previous screenshot.

```
<terminated> Testrunn [TestNG] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Mar 30, 2023, 5:25:45 PM)
</tr>
<tr>
<td>Bangalore</td>
<td>Chennai</td>
<td>AirAsia</td>
<td>Dep. 8 pm Arr. 9.30pm (1.30 hours)</td>
<td>4500.00</td>
<td>
<a href="bookflight?id=3">Book Flight</a>
</td>
</tr>
<tr>
<td>Bangalore</td>
<td>Hyderabad</td>
<td>Vistara</td>
<td>Dep. 8 AM Arr. 10:30 AMx (2.5 HOURS)</td>
<td>2500.00</td>
<td>
<a href="bookflight?id=7">Book Flight</a>
</td>
</tr>
```

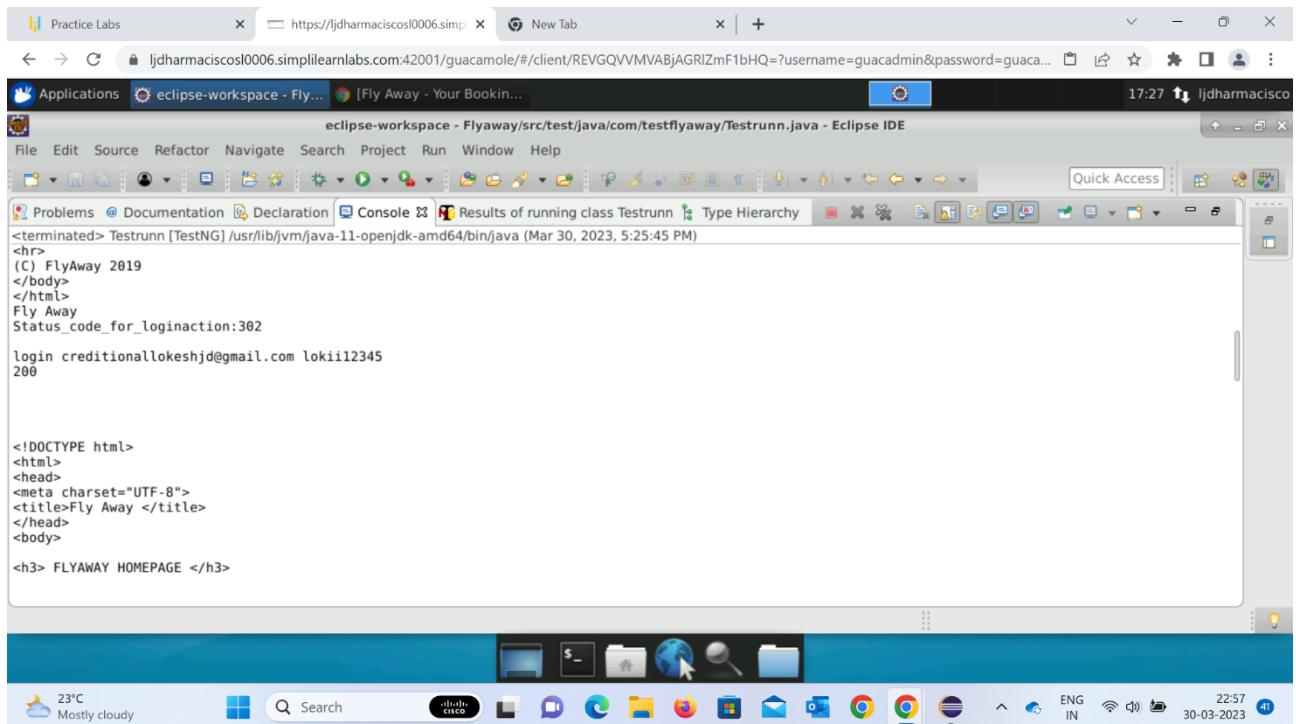
## Source\_and\_Destination:



The screenshot shows a web browser window with the URL <https://ljdarmacisco0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guaca...>. The page displays an HTML form for selecting a source and destination city from dropdown menus. The source dropdown has options: "Select Source", "Ahmedabad", "Bangalore", "Chennai", "Hyderabad". The destination dropdown has options: "Select Destination", "Ahmedabad", "Bangalore", "Chennai", "Hyderabad". A "Submit" button is at the bottom. Below the form, the Eclipse IDE interface is visible, showing the code for a Java test class named Testrunn.java.

```
<form>
<table border=0 cellspacing=2 cellpadding=4>
    <tr>
        <td width="35%">
            From<br>
            <select name=source>
                <option value=0>Select Source</option>
                <option value=7>Ahmedabad</option><option value=1>Bangalore</option><option value=4>Chennai</option><option value=11>Hyderabad</option>
            </select>
        </td>
        <td width="35%">
            To<br>
            <select name=destination>
                <option value=0>Select Destination</option>
                <option value=7>Ahmedabad</option><option value=1>Bangalore</option><option value=4>Chennai</option><option value=11>Hyderabad</option>
            </select>
        </td>
        <td>
            <button>Submit</button>
        </td>
    </tr>
</table>
</form>
```

## Login\_credentials:



The screenshot shows a web browser window with the same URL as the previous screenshot. The page now displays the results of running the Testrunn.java test. It shows the FlyAway 2019 logo and the status code for the login action: 302. Below this, it shows the login credentials used: cedula@lokedshd@gmail.com and lokii12345. At the bottom, the FLYAWAY HOMEPAGE is displayed. The Eclipse IDE interface is visible below the browser.

```
<hr>
(C) FlyAway 2019
</body>
</html>
Fly Away
Status_code_for_loginaction:302
login cedula@lokedshd@gmail.com lokii12345
200

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Fly Away </title>
</head>
<body>
<h3> FLYAWAY HOMEPAGE </h3>
```

## Flight\_booking success:

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with various icons. In the center, a web browser window is open to a URL starting with "https://ljdarmaciscosl0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guaca...". Below the browser, an Eclipse IDE window is active, titled "eclipse-workspace - Flyaway/src/test/java/com/testflyaway/Testrunn.java - Eclipse IDE". The Eclipse interface includes a toolbar, a menu bar, and several tabs like "Problems", "Documentation", "Declaration", "Console", and "Results of running class Testrunn". The "Console" tab displays the output of a TestNG test run, which includes HTML code for a flight booking confirmation page. The confirmation page content is as follows:

```
<a href="home">Home</a>
<a href="login">Login/Signup</a>

<br><br>
Your Flight by , from to , on has been booked for .<br><br>
Check your <A href="memberbookings">See Your Bookings</A>

<hr>
(C) FlyAway 2019
</body>
</html>
```

## ALL test cases are passed:

The screenshot shows a Windows desktop environment. At the top, there is a taskbar with various icons. In the center, a web browser window is open to a URL starting with "https://ljdarmaciscosl0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guaca...". Below the browser, an Eclipse IDE window is active, titled "eclipse-workspace - Flyaway/src/test/java/com/testflyaway/Testrunn.java - Eclipse IDE". The Eclipse interface includes a toolbar, a menu bar, and several tabs like "Problems", "Documentation", "Declaration", "Console", and "Results of running class Testrunn". The "Console" tab displays the output of a TestNG test run, which includes the following text indicating all tests passed:

```
login_successful
Home_page_validated
source_and_destination_setting_done
flight_Booking_validation done
flight_Booking_conformed
PASSED: test1
PASSED: test4
PASSED: test2
PASSED: test3
PASSED: test5
=====
 Default test
 Tests run: 5, Failures: 0, Skips: 0
=====
Default+ exit
```

# Postman API testing:

## GET Request loginpage:

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'History' and 'Collections'. Under 'Collections', 'LOGIN' is selected, and 'LoginPage' is highlighted. The main workspace shows a GET request for 'LoginPage' with the URL {{url}}/login. The 'Params' tab is active, showing a single query parameter 'Key' with 'Value' as 'Value'. Below the table is a 'Response' section.

## Login page html body;

The screenshot shows the Postman application interface. The 'HTML' tab is selected in the response preview. The code is as follows:

```
26<form name=frmLogin action="loginaction" method="post">
27    <table border=1 cellspacing=2 cellpadding=4>
28        <tr>
29            <td width=25%>Email id*</td>
30            <td><input name=email_id maxlength=50 type="email"></td>
31        </tr>
32        <tr>
33            <td width=25%>Password*</td>
34            <td><input name=pwd maxlength=10 type="password"></td>
35        </tr>
36    </table>
37</form>
38
```

## Test cases for loginpage:

The screenshot shows the Postman application interface. In the center, there is a table titled "Test Results (2/2)" which contains two rows of test results. The first row has a green "PASS" status with the message "Status code is 200". The second row also has a green "PASS" status with the message "ckeckTitle". The table has columns for KEY, VALUE, and DESCRIPTION.

KEY	VALUE	DESCRIPTION
Key	Value	Description

At the bottom of the table, it says "Status: 200 OK Time: 29 ms Size: 1006 B Save Response". On the left side of the interface, there is a sidebar with a "Collections" tab selected, showing a list of requests under the "LOGIN" collection. One request, "LoginPage", is highlighted in green.

## TEST Cases for login:

The screenshot shows the Postman application interface. In the center, there is a table titled "Test Results (3/3)" which contains three rows of test results. The first two rows have green "PASS" status with messages "Status code is 200" and "ChekTitle". The third row has a green "PASS" status with the message "ResponseShouldNotContainLoginFail". The table has columns for BODY, COOKIES, HEADERS, and TEST RESULTS.

BODY	COOKIES	HEADERS	TEST RESULTS
PASS Status code is 200			
PASS ChekTitle			
PASS ResponseShouldNotContainLoginFail			

At the bottom of the table, it says "Status: 200 OK Time: 10 ms Size: 591 B Save Response". On the left side of the interface, there is a sidebar with a "Collections" tab selected, showing a list of requests under the "LOGIN" collection. One request, "Login", is highlighted in green.

## GET Request:LoginPage

The screenshot shows the Postman application interface. In the left sidebar, under the 'LOGIN' collection, there is a 'LoginPage' item. The main workspace shows a GET request for 'HomePage'. The 'Tests' tab contains the following JavaScript code:

```
const $ = cheerio.load(pm.response.text());
console.log($('title').text());

console.log($('select[name="source"] option:nth-child(3)').attr('value'));

pm.environment.set("s_ID", $('select[name="source"] option:nth-child(3)').attr('value'));
pm.environment.set("d_ID", $('select[name="destination"] option:nth-child(3)').attr('value'));
```

The status bar at the bottom indicates it's 22°C, Mostly cloudy, and the date is 30-03-2023.

## GET Request :search\_flight

The screenshot shows the Postman application interface. In the left sidebar, under the 'LOGIN' collection, there is a 'search\_flight' item. The main workspace shows a GET request for 'HomePage' with parameters 's\_ID={{(s\_ID)}}&d\_ID={{(d\_ID)}}'. The 'Pretty' tab displays the generated HTML code:

```
</select>
</td>
<td width="35%">
To:<br>
<select name=destination>
<option value="0">Select Destination</option>
<option value="7">Ahmedabad</option><option value=1>Bangalore</option><option value=4>Chennai</option><option value=11>Hyderabad</option><option value=6>Kolkata</option><option value=2>Mumbai</option><option value=3>New Delhi</option><option value=10>tomato</option>
</select>
</td>
<td>
<button>Submit</button>
```

The status bar at the bottom indicates it's 22°C, Mostly cloudy, and the date is 30-03-2023.

## GET Request:Flight\_booking\_validation

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for 'Practice Labs' and 'New Tab'. Below the tabs, the URL is https://ljdarmaciscosl0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guaca... . The main workspace shows a collection named 'LOGIN' with several requests listed:

- GET LoginPage
- POST Login
- GET HomePage
- GET HomePage Copy
- GET search\_flight
- GET Book\_flight
- GET Complete\_booking

The 'Complete\_booking' request is selected and expanded. Its details are shown in the central pane:

```
36 <b>Airline:</b> AirAsia<br>
37 <b>Sector:</b> Bangalore - Chennai<br>
38 <b>Departure:</b> 8 pm<br>
39 <b>Arrival:</b> 9.30pm<br>
40 <b>Duration:</b> 1.30 hours<br><br>
41 <a href="completerequest">Click to complete booking</a>
42
43
44 <hr>
45 (C) FlyAway 2019
46
47
48 </html>
```

The bottom of the screen shows a Windows taskbar with various icons and system status.

## GET REQUEST :Complete\_flight\_booking

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for 'Practice Labs' and 'New Tab'. Below the tabs, the URL is https://ljdarmaciscosl0006.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guaca... . The main workspace shows a collection named 'LOGIN' with several requests listed:

- GET LoginPage
- POST Login
- GET HomePage
- GET HomePage Copy
- GET search\_flight
- GET Book\_flight
- GET Complete\_booking

The 'Complete\_booking' request is selected and expanded. Its details are shown in the central pane:

```
27 <br><br>
28
29
30
31 <br><br>
32 Your Flight by AirAsia , from Bangalore to Chennai, on Thu Mar 30 17:57:50 UTC 2023 has been booked for 4500.00.<br><br>
33 Check your <A href="memberbookings">See Your Bookings</A>
34
35
36 <hr>
37 (C) FlyAway 2019
38
39 </html>
```

The bottom of the screen shows a Windows taskbar with various icons and system status.

