FlyAway flight booking engine application

flyaway flights | https://github.com/malinimurthy123/FlyAwayFlightsProjectwithHibernate.git

flyaway flight booking

Malini Murthy

2021

https://github.com/malinimurthy123/FlyAwayFlightsProjectwithHibernate.git

Github link:

<https://github.com/malinimurthy123/FlyAwayFlightsProjectwithHibernate.git>

**Project objective:**

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

**Background of the problem statement:**

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

**The website needs to have the following features:**

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

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

● An admin login page where the admin can change the password after login, if he wishes  
● A master list of places for source and destination  
● A master list of airlines  
● A list of flights where each flight has a source, destination, airline, and ticket price  
  
The goal of the company is to deliver a high-end quality product as early as possible. 

**The flow and features of the application:**

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

**Software Requirements**

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

|  |  |
| --- | --- |
| IntelliJ |  |
| Java |  |
| Maven |  |
| Git |  |
| GitHub |  |
| Scrum | JiRA |
| Search and Sort techniques | List, arrays |
| Paypay integration using their API | https://www.codejava.net/coding/how-to-integrate-paypal-payment-into-java-web-application |

Dependencies in Maven POM.XML:

* hibernate-core - 5.4.10.Final
* hibernate-validator - 6.1.6.Final
* log4j - 1.2.17
* javax.servlet-api - 4.0.1
* persistence-api - 1.0.2
* mysql-connector-java - 8.0.23
* junit
* maven-war-plugin

Integration to payment gateway using sandbox paypal:

<https://www.codejava.net/coding/how-to-integrate-paypal-payment-into-java-web-application>

**Screen Shots of Fly Away (An airline booking portal)-Simplilearn-Phase2 Project**

**Git Link-**

**https://github.com/malinimurthy123/FlyAwayFlightsProjectwithHibernate.git**

**Project objective:**

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

**Background of the problem statement:**

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

**The website needs to have the following features:**

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

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

● An admin login page where the admin can change the password after login, if he wishes  
● A master list of places for source and destination  
● A master list of airlines  
● A list of flights where each flight has a source, destination, airline, and ticket price  
  
The goal of the company is to deliver a high-end quality product as early as possible. 

**The flow and features of the application:**

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

**Software Requirements**

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

**Functional Requirements**

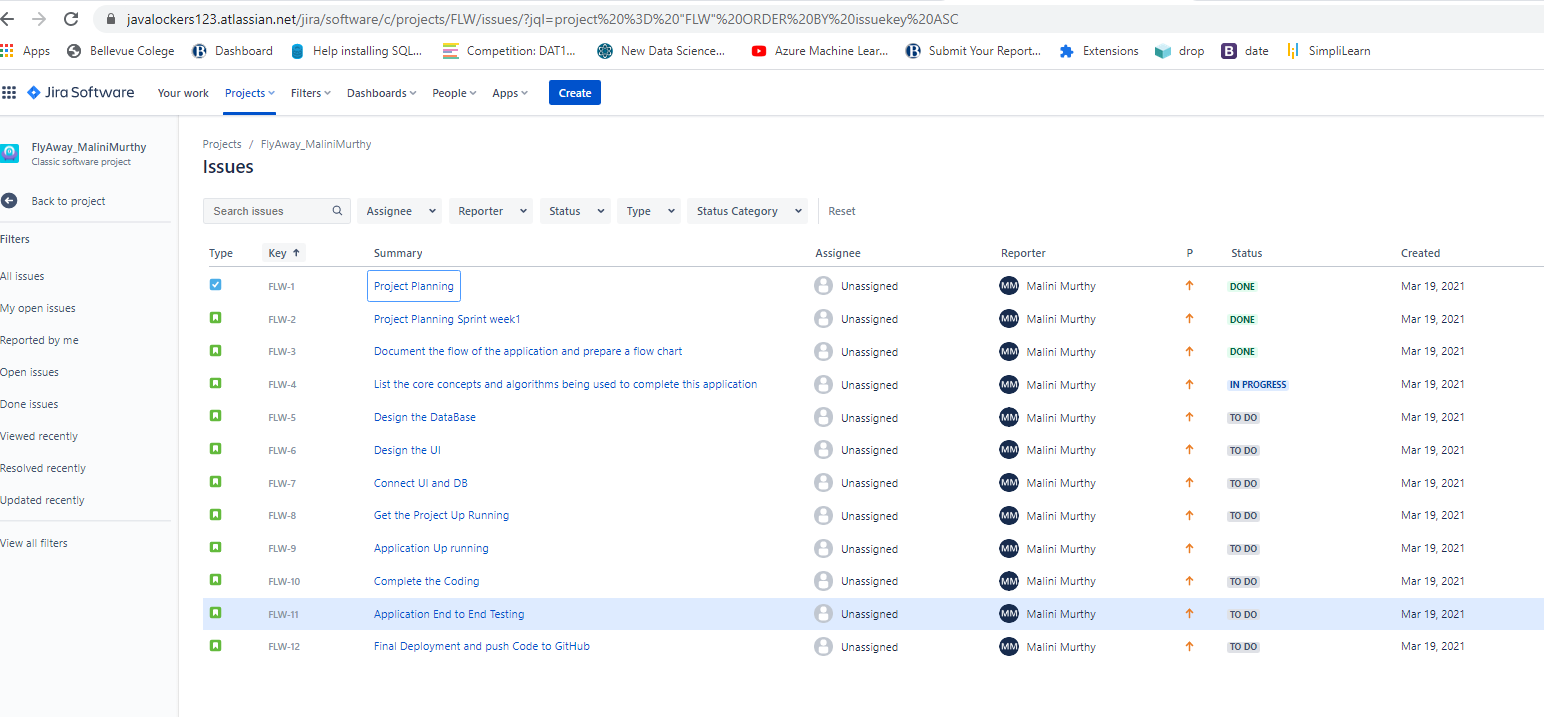
● An admin login page where the admin can change the password after login, if he wishes  
● A master list of places for source and destination  
● A master list of airlines  
● A list of flights where each flight has a source, destination, airline, and ticket price

[FlyAwayAirWays](https://javalockers123.atlassian.net/browse/FLYAW)

Project Creation in Jira under SCRUM

[Projects](https://javalockers123.atlassian.net/secure/BrowseProjects.jspa)-[FlyAway\_MaliniMurthy](https://javalockers123.atlassian.net/browse/FLW" \t "_self)

<https://javalockers123.atlassian.net/jira/software/c/projects/FLW/issues/?jql=project%20%3D%20%22FLW%22%20ORDER%20BY%20issuekey%20ASC>



DESCRIPTION

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

**Background of the problem statement:**

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

**The website needs to have the following features:**

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

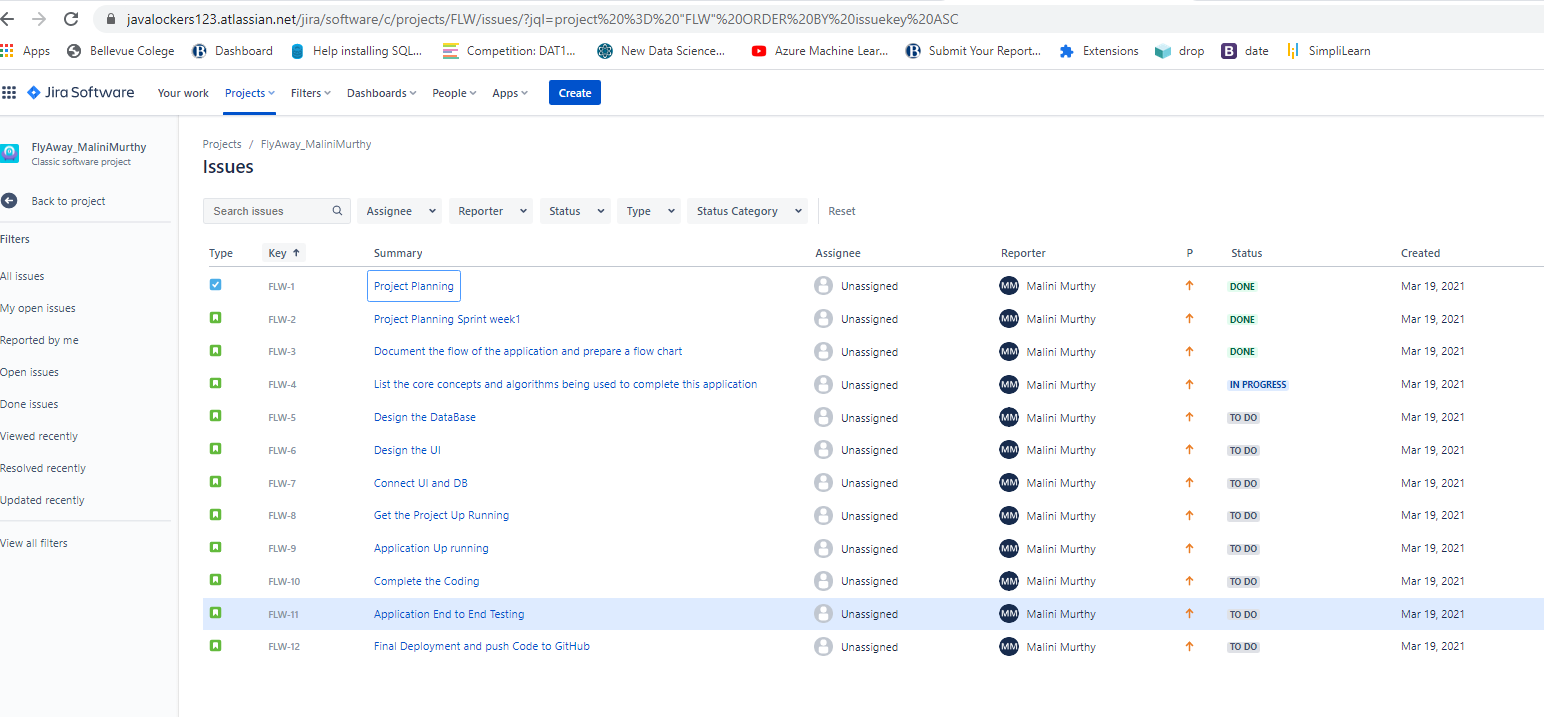
For the above features to work, there will be an admin backend with the following features:

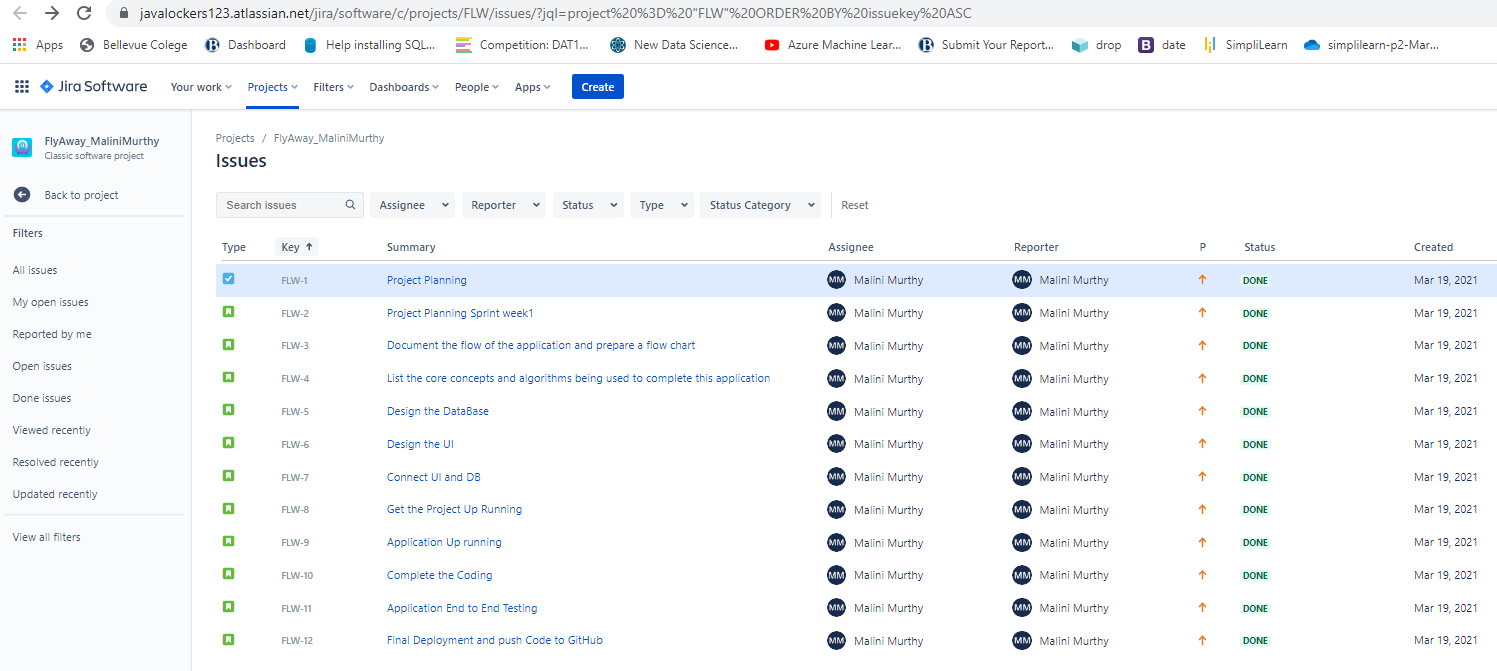
● An admin login page where the admin can change the password after login, if he wishes  
● A master list of places for source and destination  
● A master list of airlines  
● A list of flights where each flight has a source, destination, airline, and ticket price  
  
The goal of the company is to deliver a high-end quality product as early as possible. 

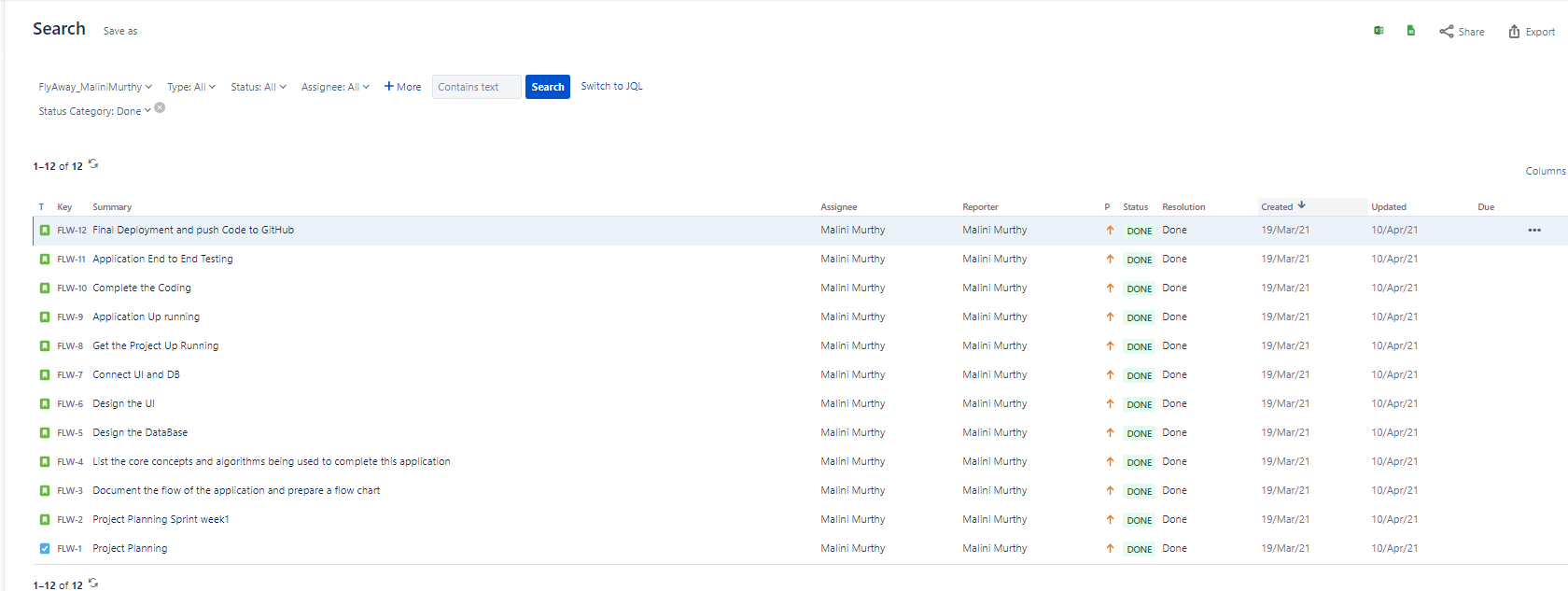
**The flow and features of the application:**

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

The Project is completed in two sprints and below are screen shots.







Database tables;

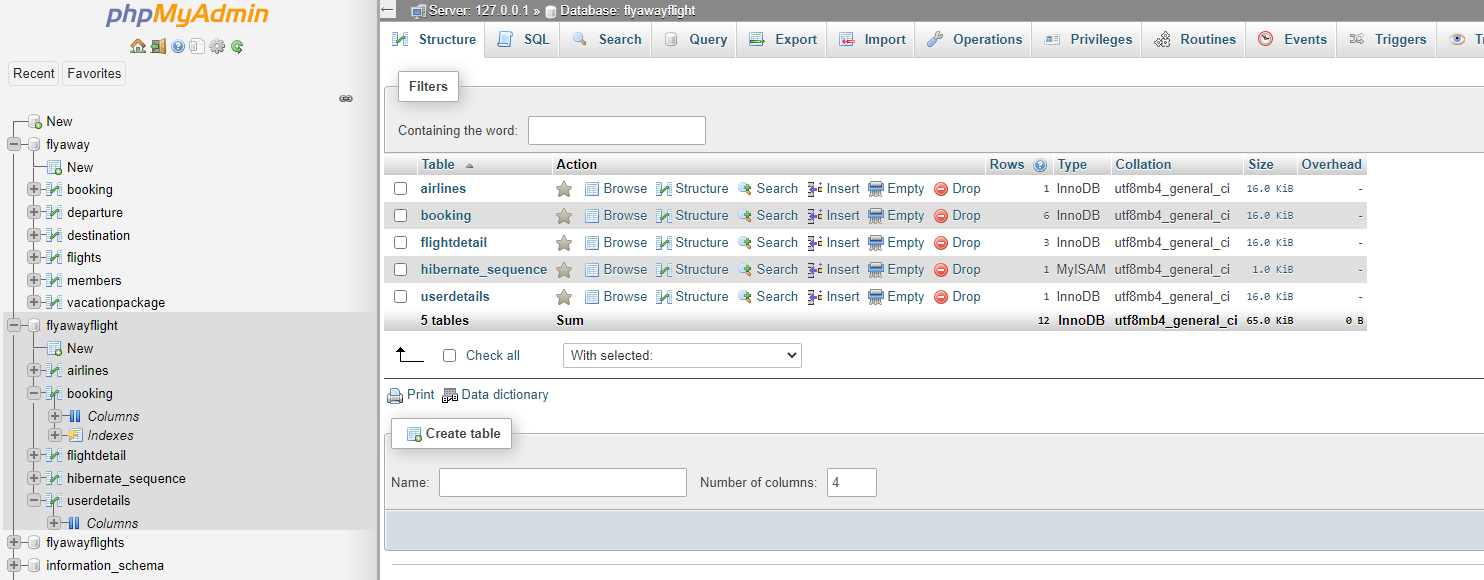
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| user |  | Flights |  | Destination |  | Departure |
| membershipnumber | int,primary key | flightid | int,primary | destinationcode | varchar,primary | departurecode |
| membername | varchar | flightnumber | varchar | VacationpackageIid | varchar | departureflyingfrom |
| Password | varchar | flightname | varchar | destinationname | varchar |  |
|  |  | fromcity | varchar | flightid | int |  |
|  |  | tocity | varchar |  |  |  |
|  |  | traveldate | date |  |  |  |
|  |  | description | varchar |  |  |  |
|  |  | time | time |  |  |  |
|  |  | travelduration | varchar |  |  |  |
|  |  | ticketprice | float |  |  |  |
|  |  | airportname | varchar |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VacationPackage | |  | Booking | |  |
| vacationpackageid | | varchar,primary | Bookingnumber | | varchar,primary |
| vacationname | | varchar | flightid | | int |
|  | |  | flightname | | vachar |
|  | |  | primarypassengername | | varchar |
|  | |  | bookingdate | | date |
|  | |  | emailid | | varchar |
|  | |  | address | | varchar |
|  | |  | totalpassengers | | int |
|  | |  | bookingprice | | float |
|  | |  | bookingfinalprice | |  |
|  | |  |  | |  |
| departure |  | | |
| D01 | SeattleTacoma | | |
| D02 | Newark | | |
| D03 | NewYork | | |
| D04 | WashingtonDC | | |

Some examples of the data assumption that can change at time and this is not fixed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| members | | |  | |
| 1001 | | | admin | |
| V001 | | | seattle | |
| V002 | | | Hawaii | |
| V003 | | | Mauii | |
| V004 | | | Florida | |
| V005 | | | NewYork | |
| destinaton |  |  | |  | |
| D01 | V001 | Seattle | | 101 | |
| D02 | V002 | Hawaii | | 102 | |
| D03 | V003 | Maui | | 103 | |
| D04 | V004 | Florida | | 104 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| flights |  |  |  |  |
| flightid | 101 | 102 | 103 | 104 |
| flightnumber | AA01 | SW02 | DE03 | SA04 |
| flightname | AmericanAirlines | SouthWest | Delta | SpiritAirlines |
| fromcity | Seattle | Hawaii | Maui | Florida |
| tocity | Florida | Seattle | Hawaii | Seattle |
| traveldate | 9/15/2020 | 10/15/2020 | 10/15/2021 | 7/14/2021 |
| description | AmericanAirlines | SouthWest | Delta | SpiritAirlines |
| time | 7:45:00 | 9:45:00 | 9:45:00 | 18:00:00 |
| travelduration | 4 | 5 | 6 | 7 |
| ticketprice | 500 | 700 | 200 | 500 |
| airportname | SeattleTacoma | DanielHNL | Kahului | OrlandoMCO |



**Admin Flow Chart Diagram**

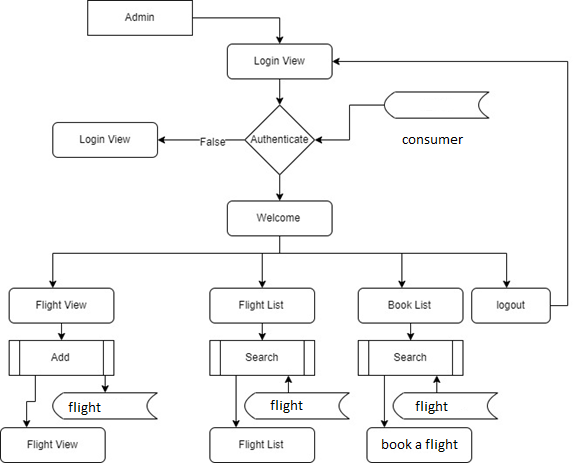
shown a confirmation page with the details of the booking.

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

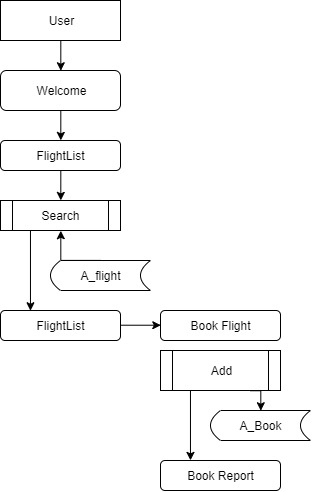
● An admin login page where the admin can change the password after login, if he wishes  
● A master list of places for source and destination  
● A master list of airlines  
● A list of flights where each flight has a source, destination, airline, and ticket price  
  
The goal of the company is to deliver a high-end quality product as early as possible. 

**The flow and features of the application:**

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

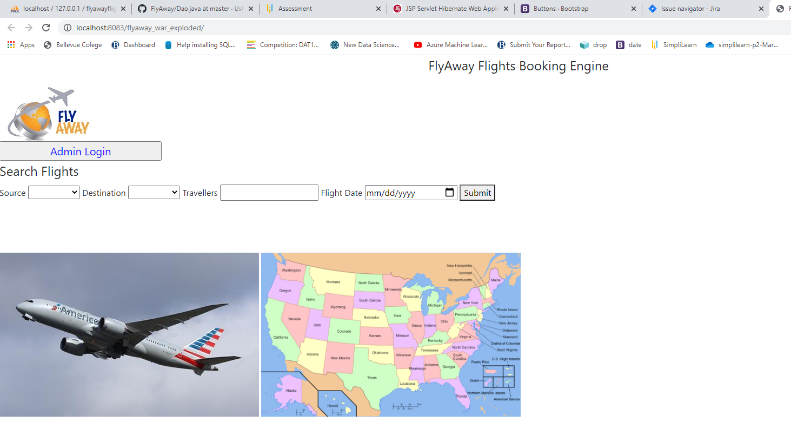


**User Flow Chart Diagram**

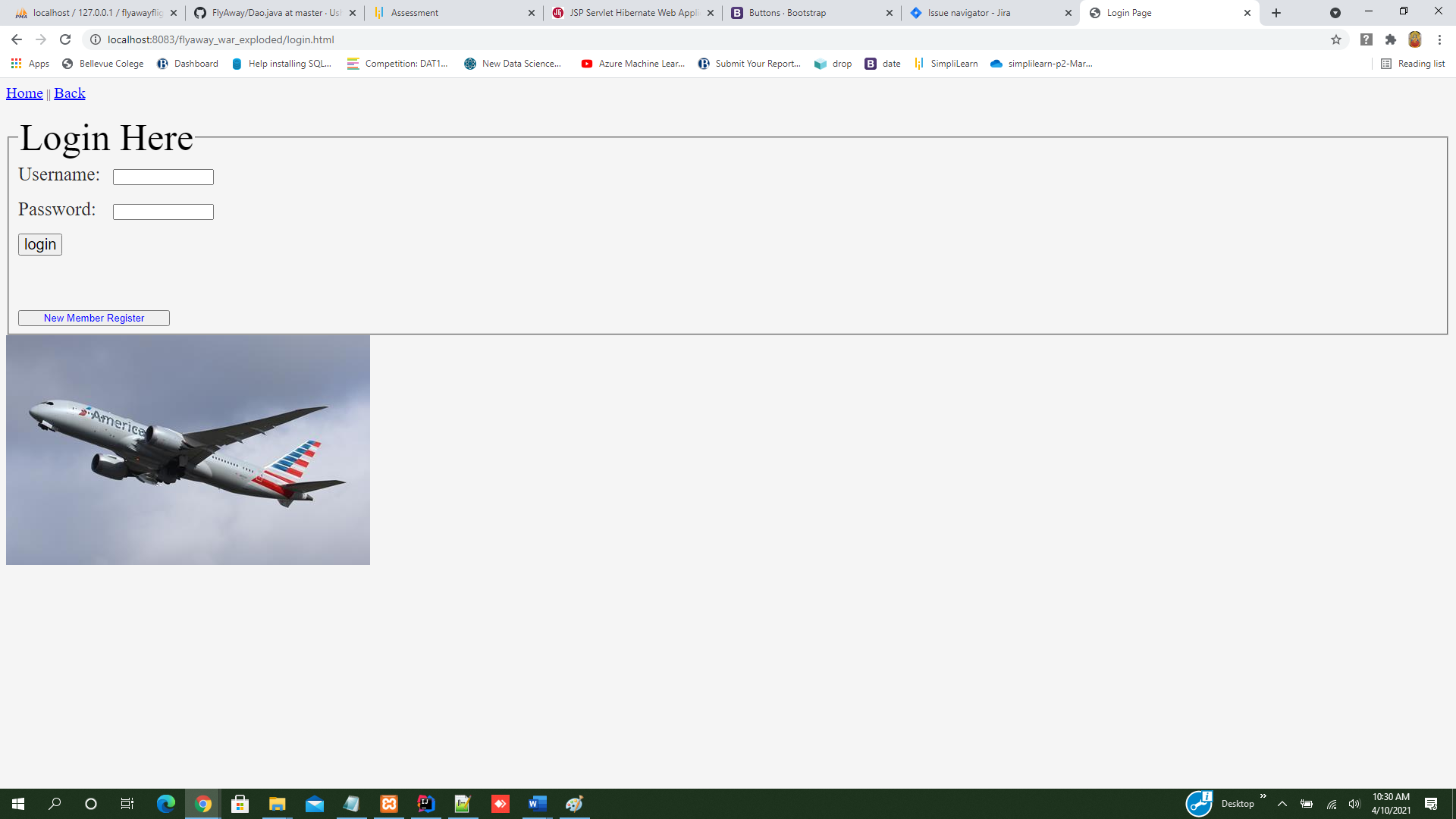


**Home page Screen**

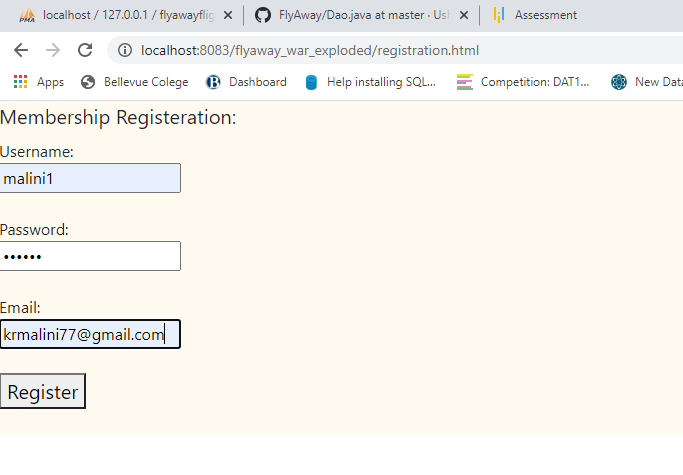
This is the index page with Login module and flight search options



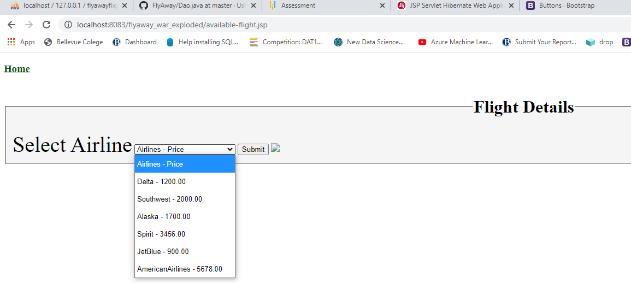
2. Do a login:



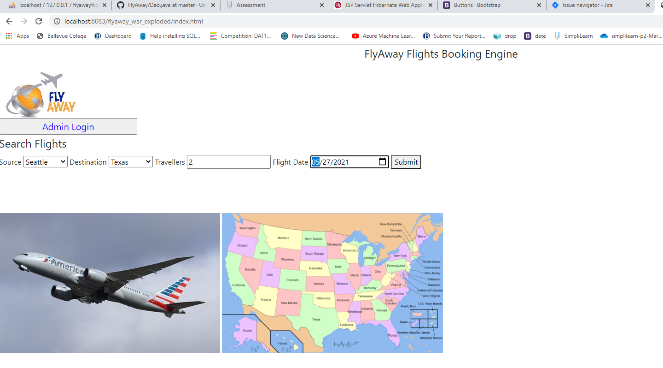
3. create a new member



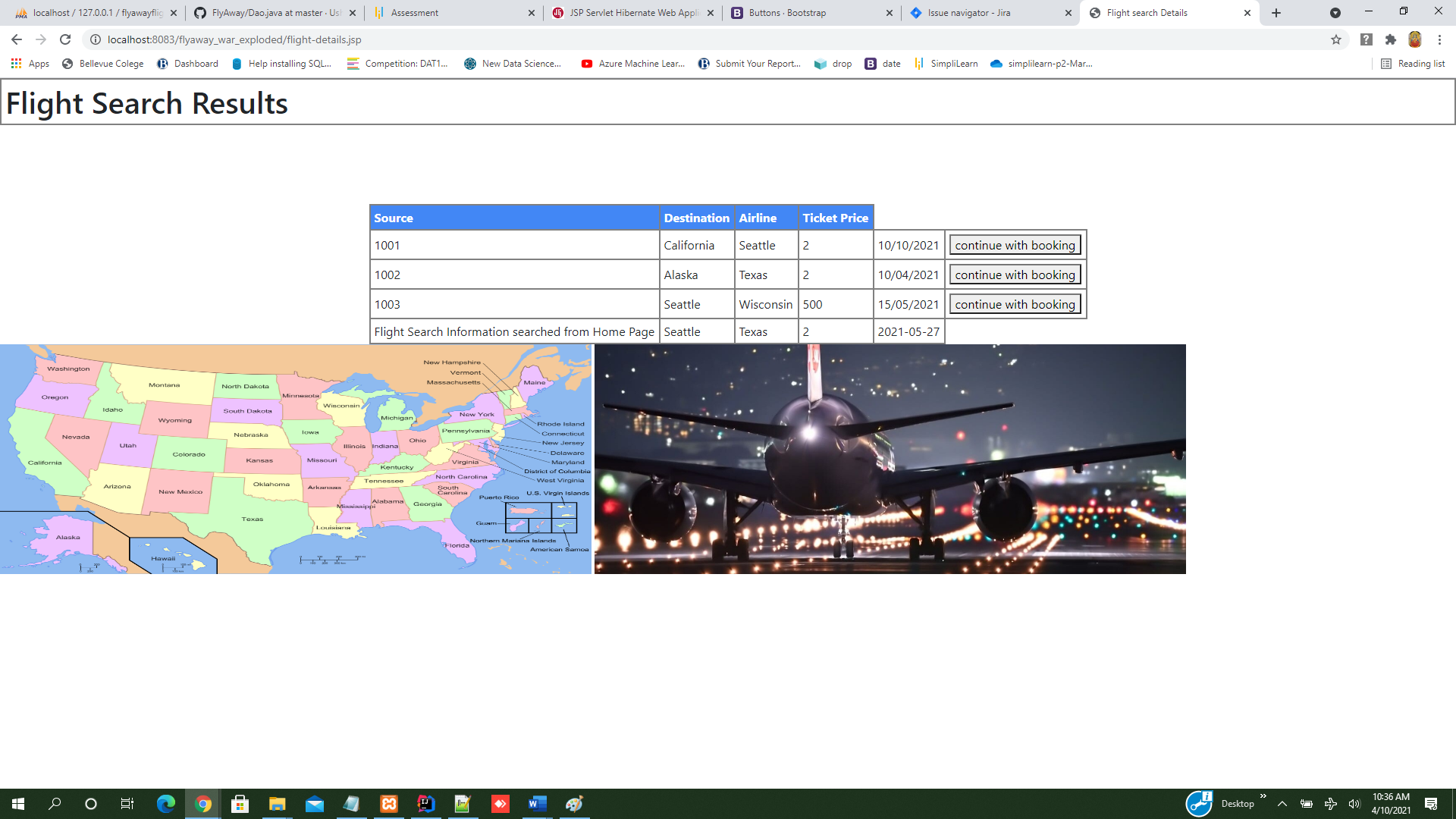
4. check airline details:



5. flight search after / before login



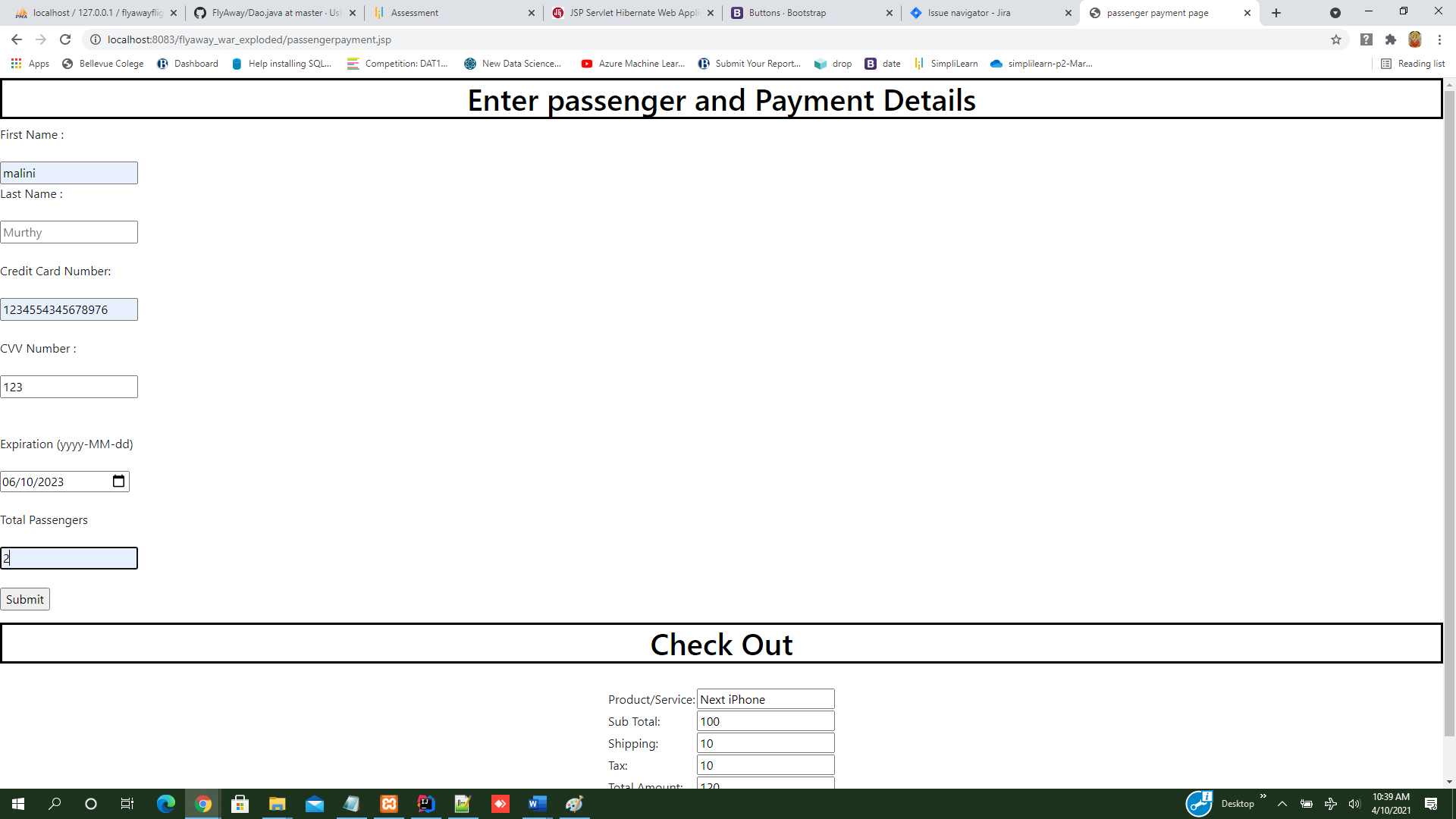
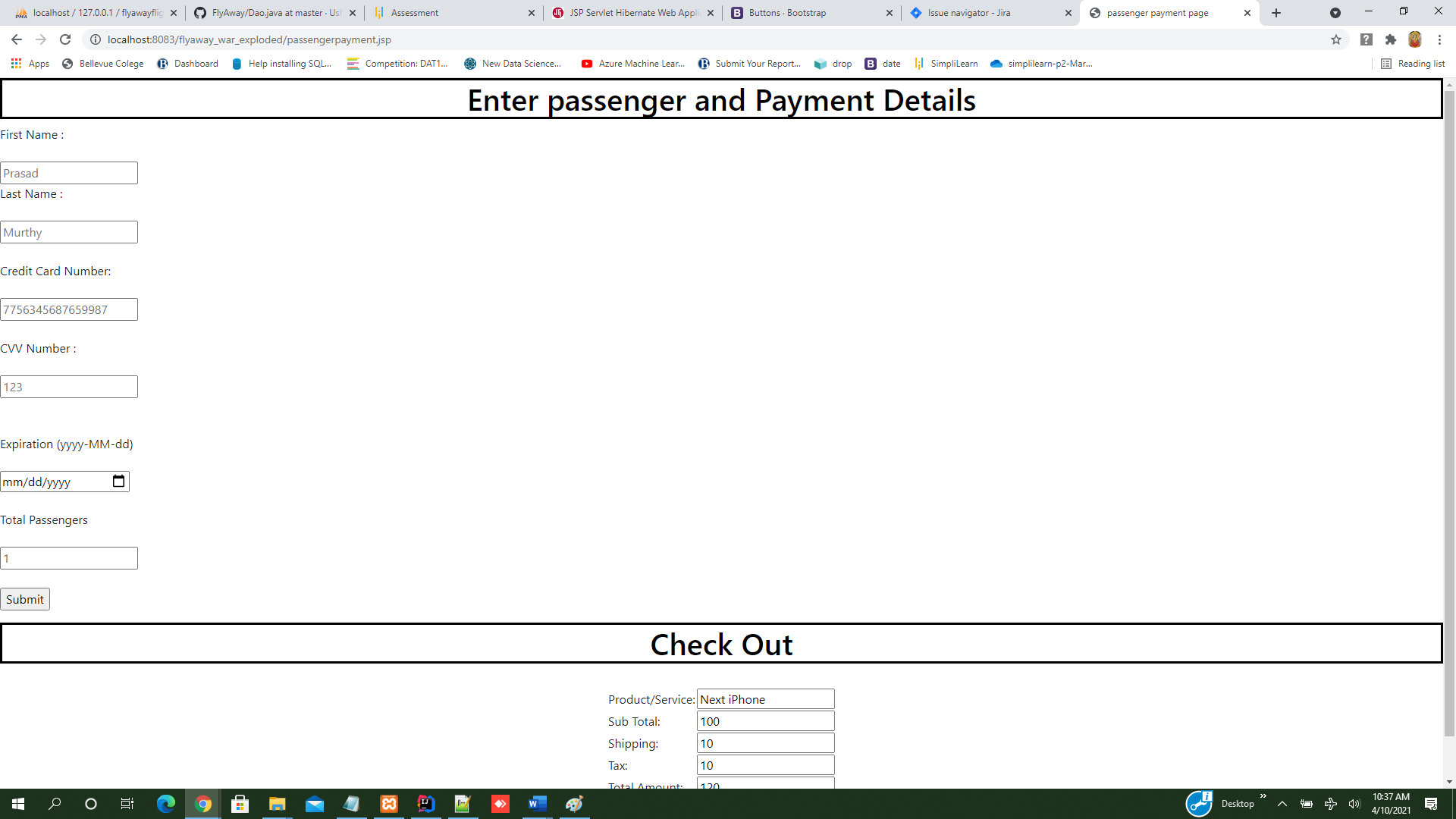
6. search landing screen

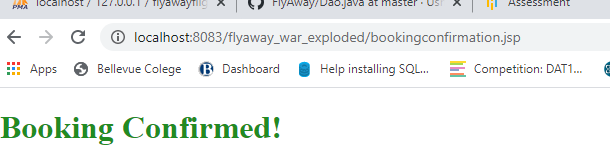


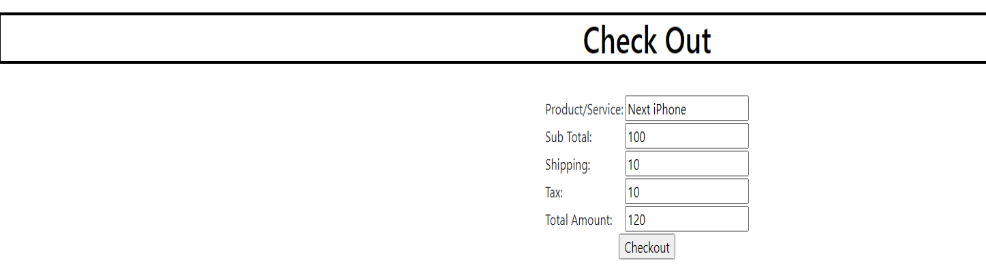
7. Select a package and continue:

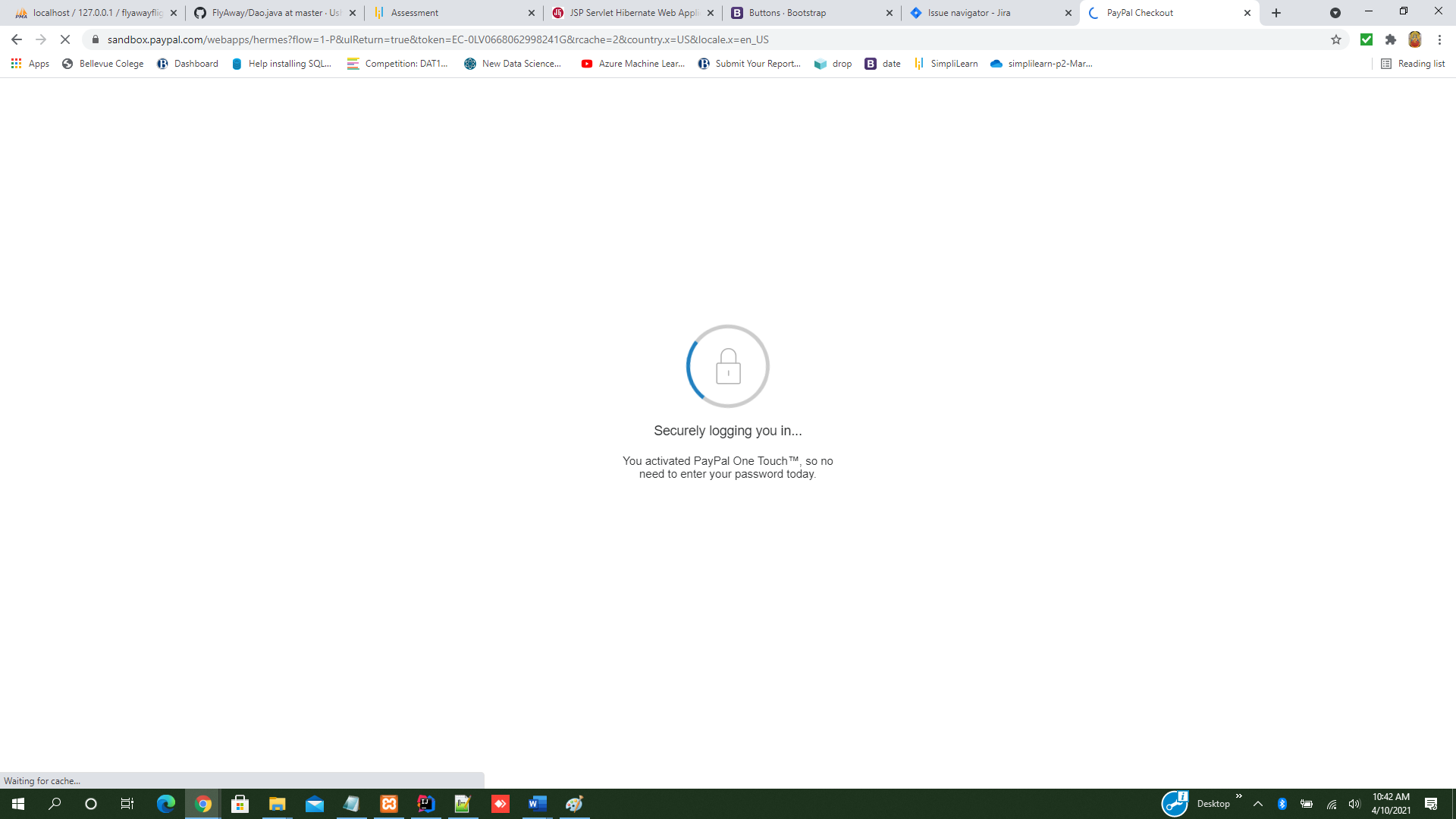
* 1. payment and booking confirmation saving to db
  2. check out using paypal sandbox testing API integration

booking is achieved through both ways:







=🡺 