Spring boot application

# Introduction

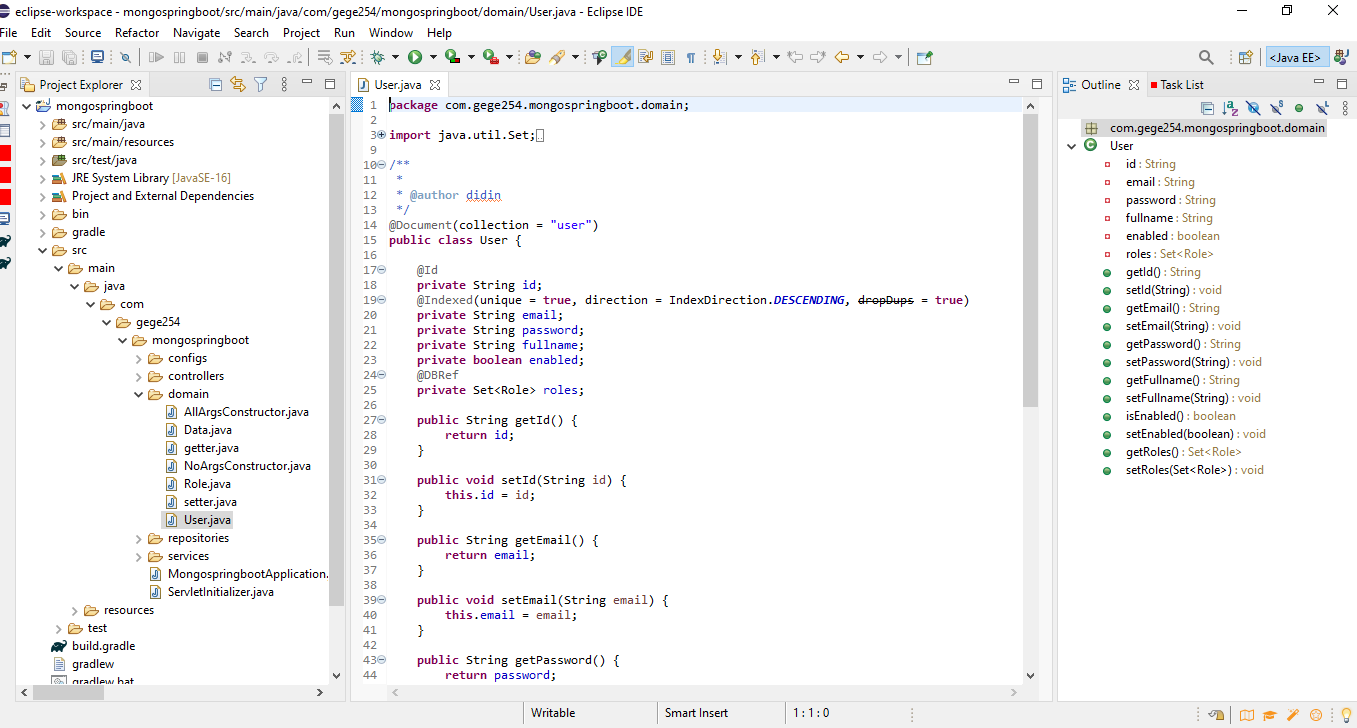
The purpose of this project is to make sure that spring boot application is connected to the database and one can successfully create a user to log into the dashboard after authentication process. The application also test the creation of data into the database and displaying it once it is saved successfully.

|  |  |
| --- | --- |
| **TASK FUNCTIONAL REQUIREMENTS** | |
| Note: Functional-Requirement (FR) 1 represents a primary requirement task  Non-Functional-Requirement (NFR) 1 is a non-functional requirement and not necessarily needed | |
| FR01 | The user should be able to login to the system |
| FR02 | The user should be provided with a unique session that prevents them from re-accessing critical server resources |
| NFR01 | The system should make use of CSS to improve user-experience |
| … |  |

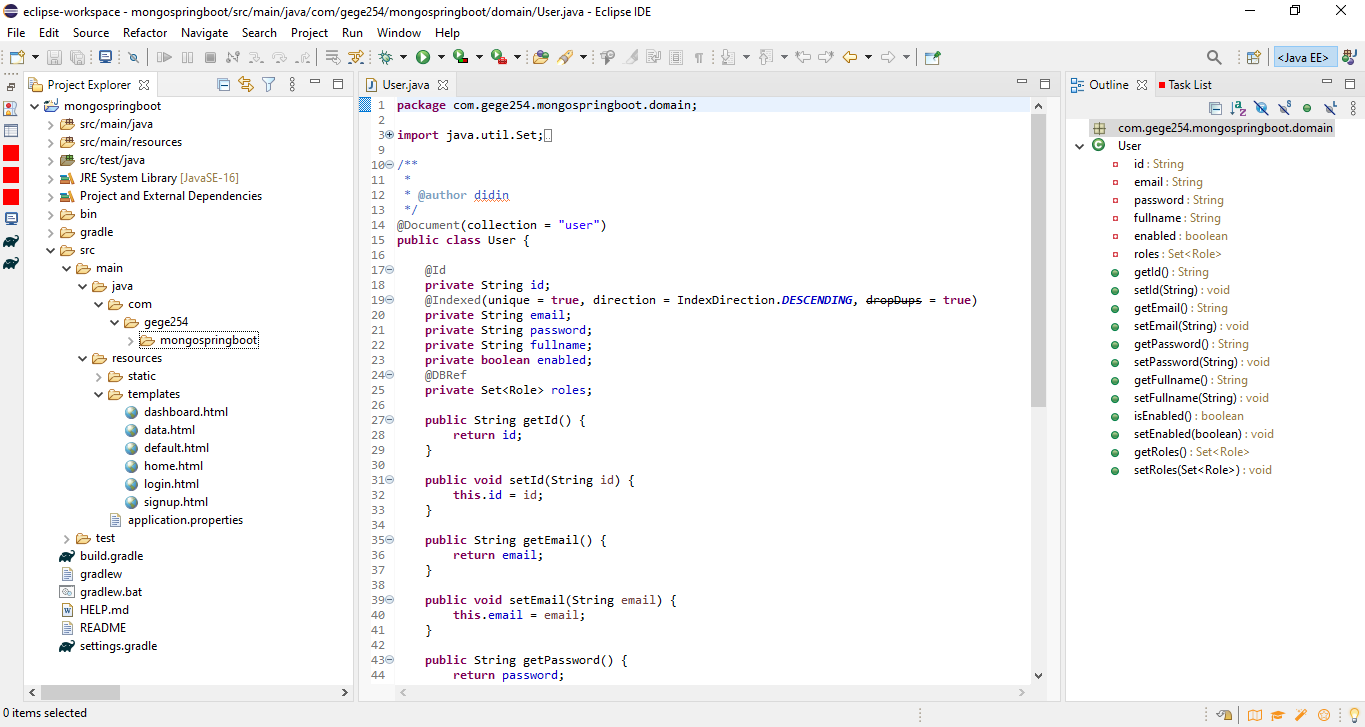
# Technical Code Description

The web application is built by using spring boot, a java framework, for backend to connect to the mongo dB database, with html template together with css. The spring boot application is mostly using the Model Views and Controls (MVC) to run the application. The model with mostly has the domain of the schema to be stored in the database, views are being the html template and css will be situated including JavaScript and finally the controllers is where they define the routes between the model and the views.

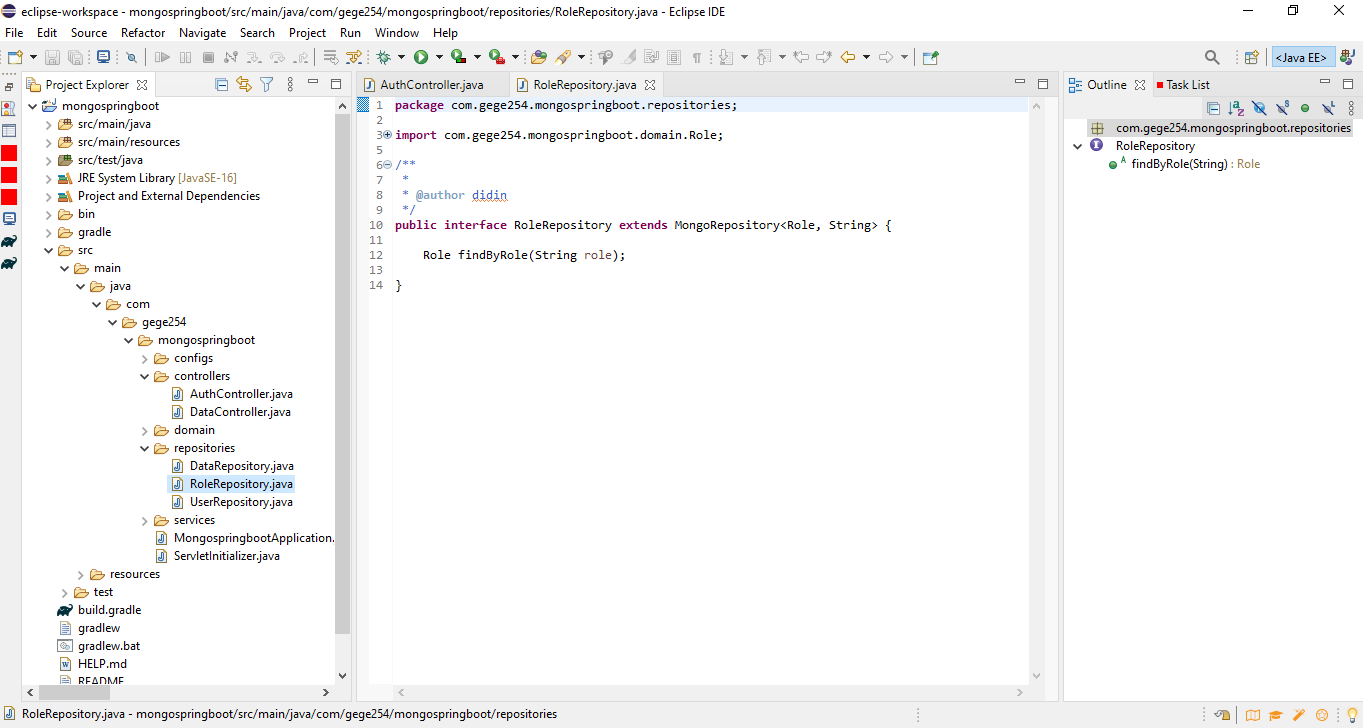
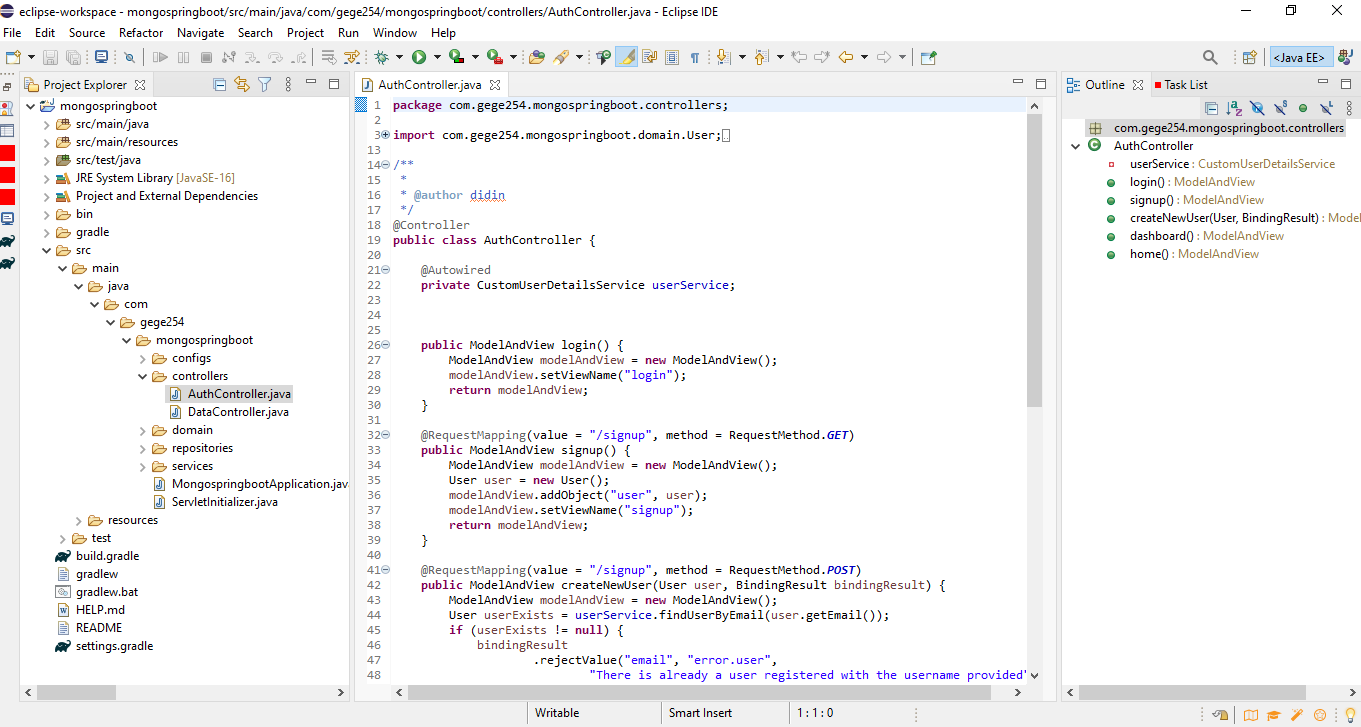
As shown in the diagram below the java files inside the domain package shows the code for the model where it defines even the collection on where the data in the schema will be created.



The views code will be found in the template package where it is carrying the html, css and JavaScript code.



Finally, we have the controllers where the routes are found and also the repository that carry the interface of the models to the controllers inside the repository package.



# Testing Procedure

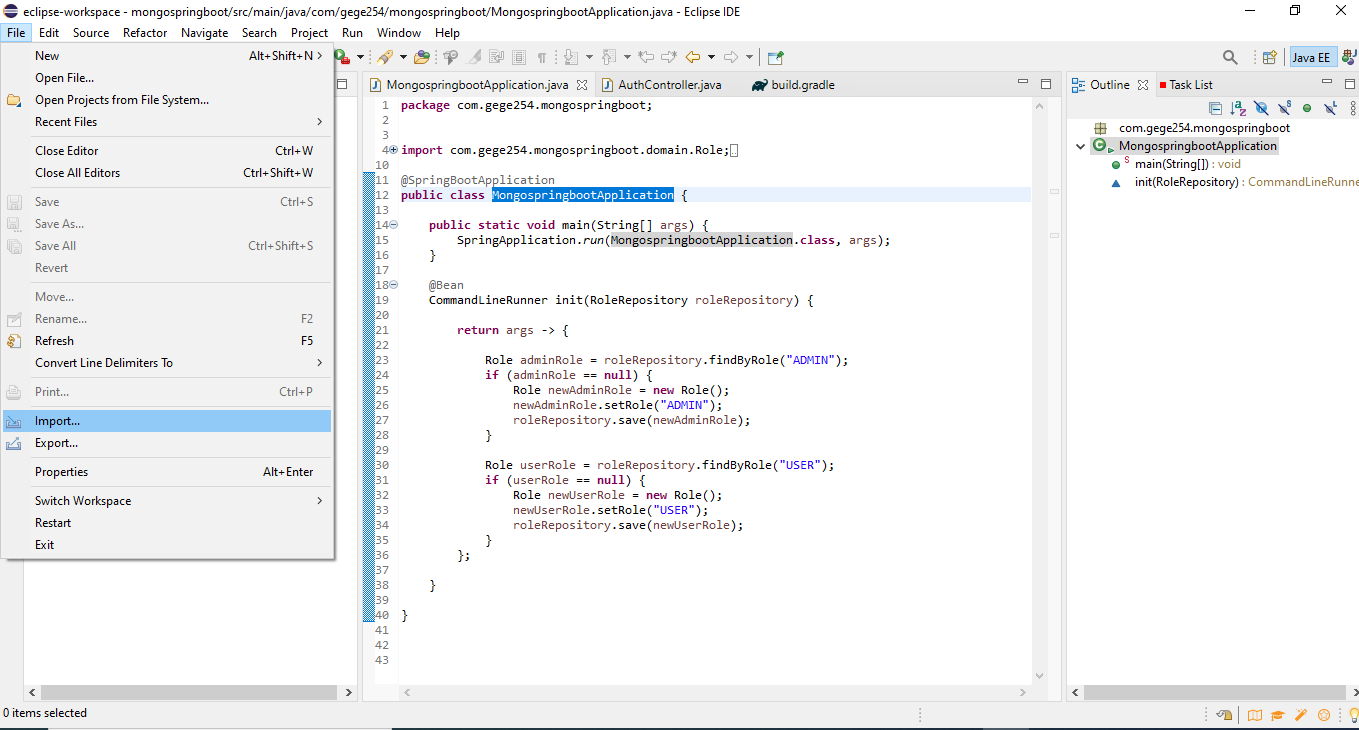
**Detailed steps on running the Spring Boot application**

Before importing the application on eclipse you need the following:

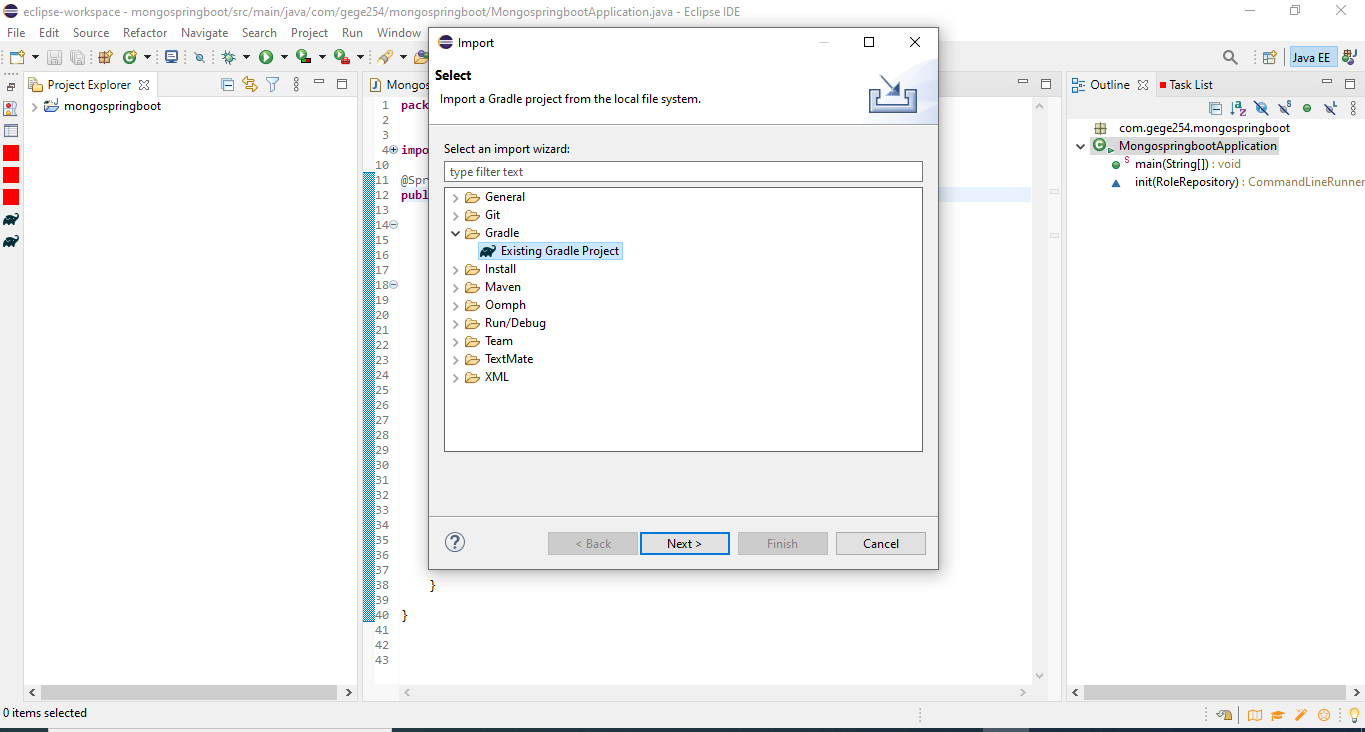
1. Mongo dB running on either locally or using the atlas mongo dB since this application will be using this NoSQL database.
2. Make sure your eclipse can run gradle since this application will was built on gradle.

Steps to run the spring boot application on eclipse

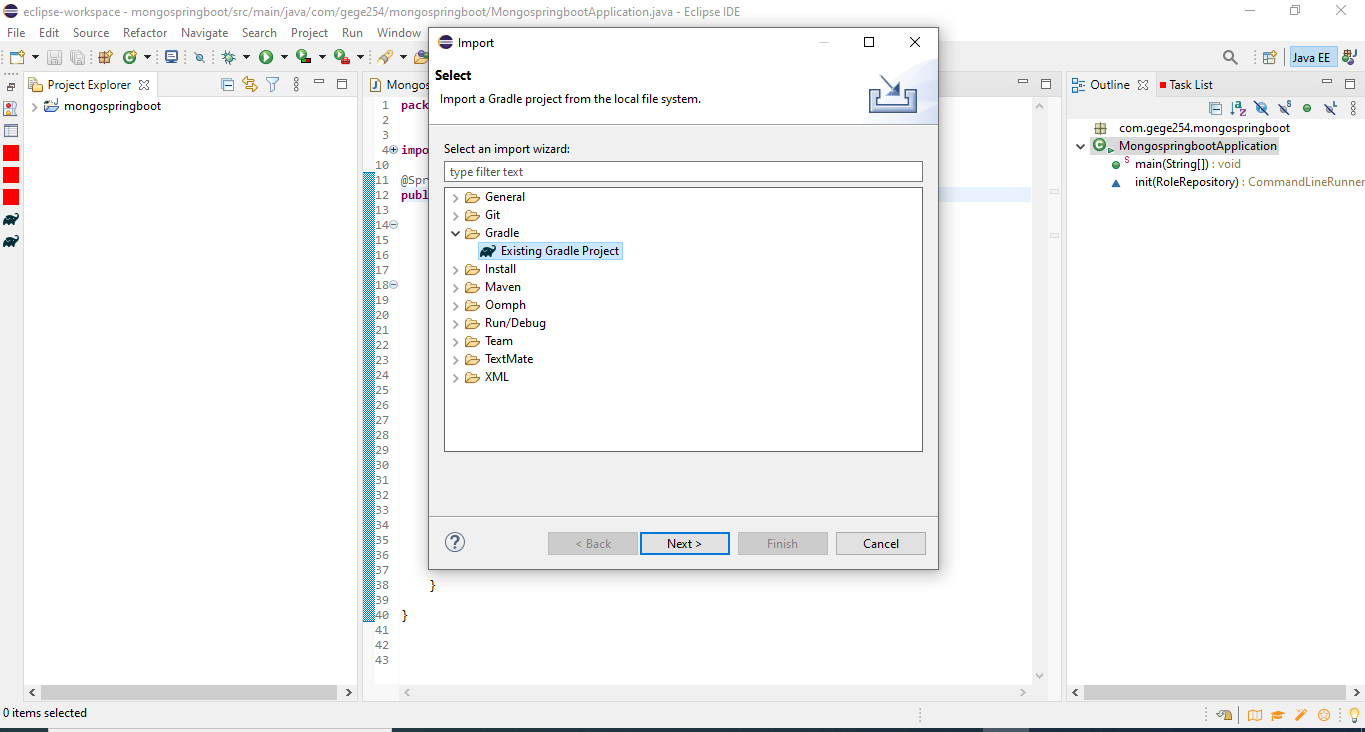
1. Open eclipse IDE and click on the file on the top left corner and head to the import and click to import the folder as shown on the image below.



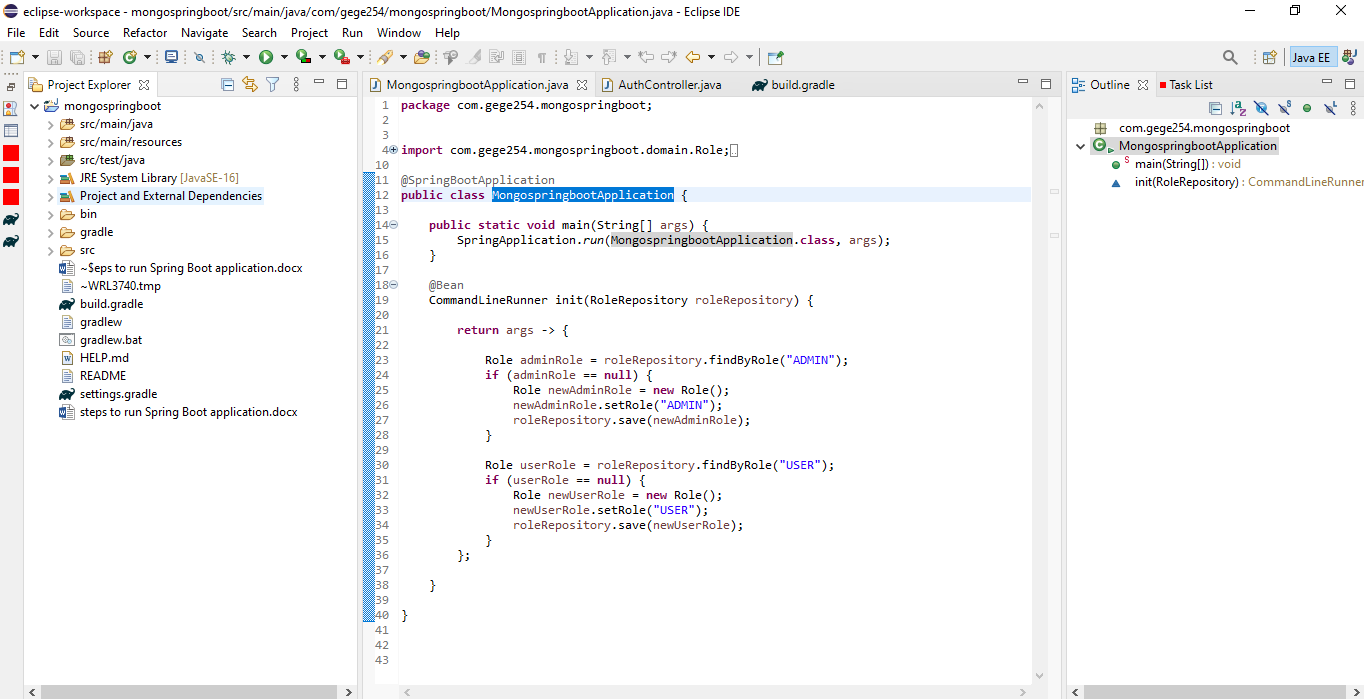
1. Once you have clicked the import button choose the gradle folder and when you expand it choose the existing gradle project and click next.



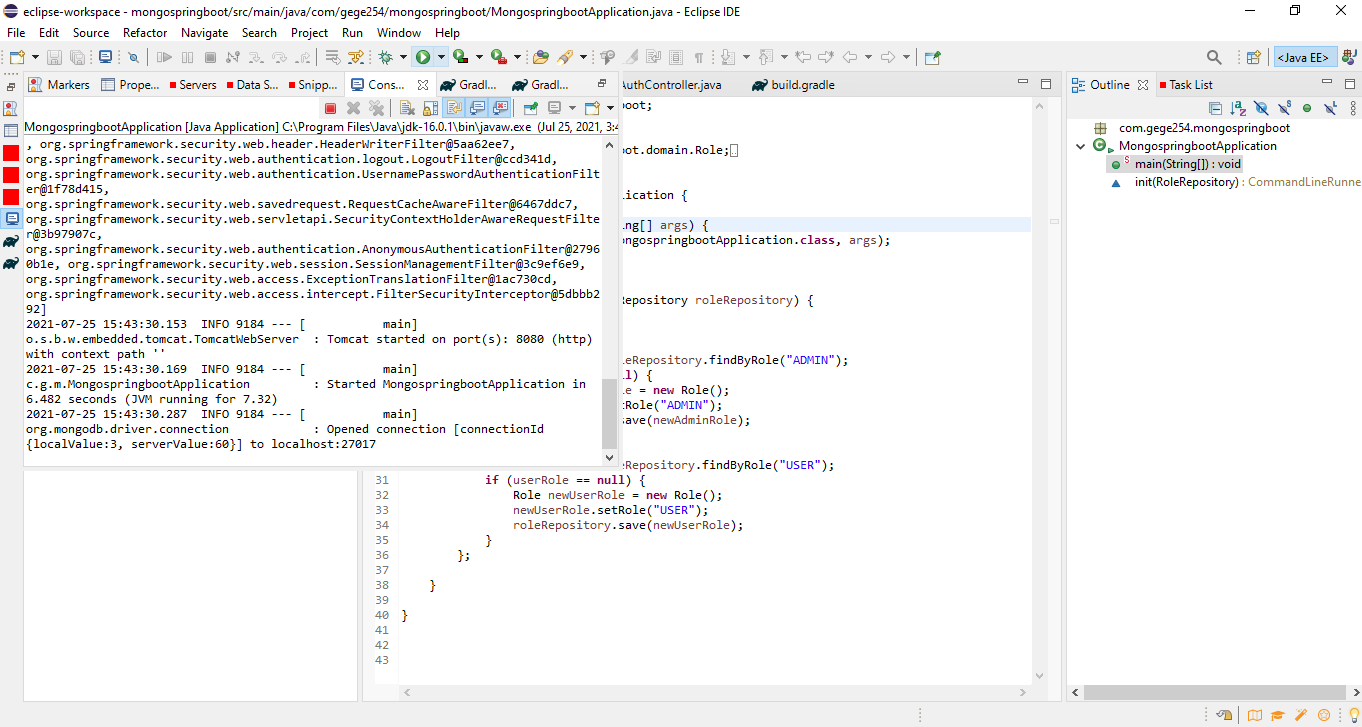
1. Click next until you find the window below to go browse the area where your spring boot folder is stored. After that click finish to open the project on your Eclipse IDE.



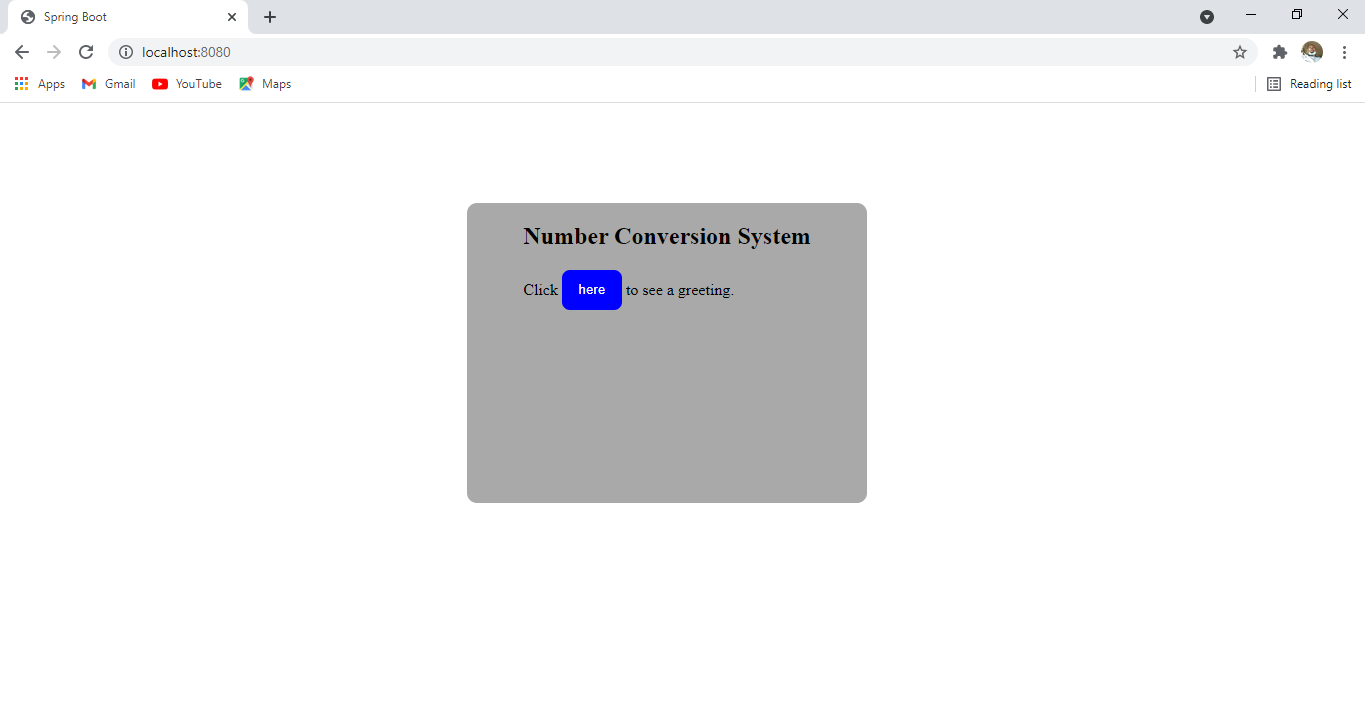
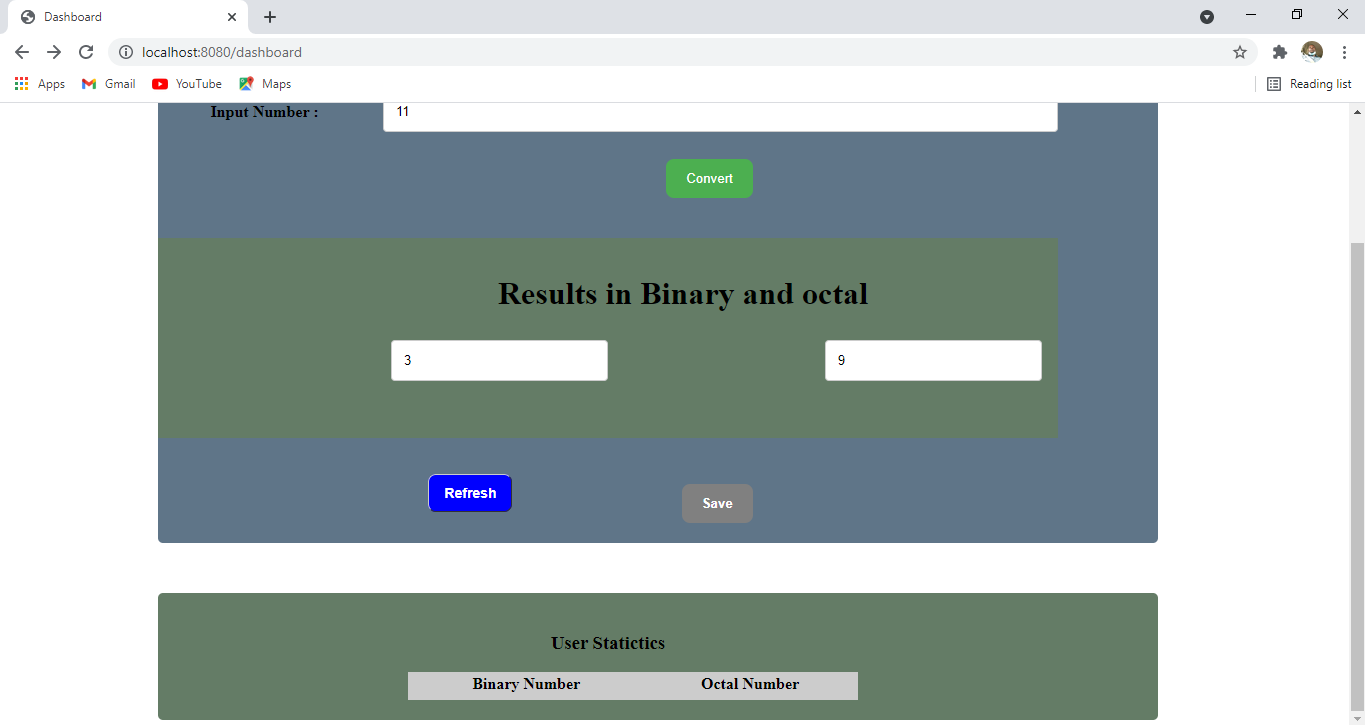
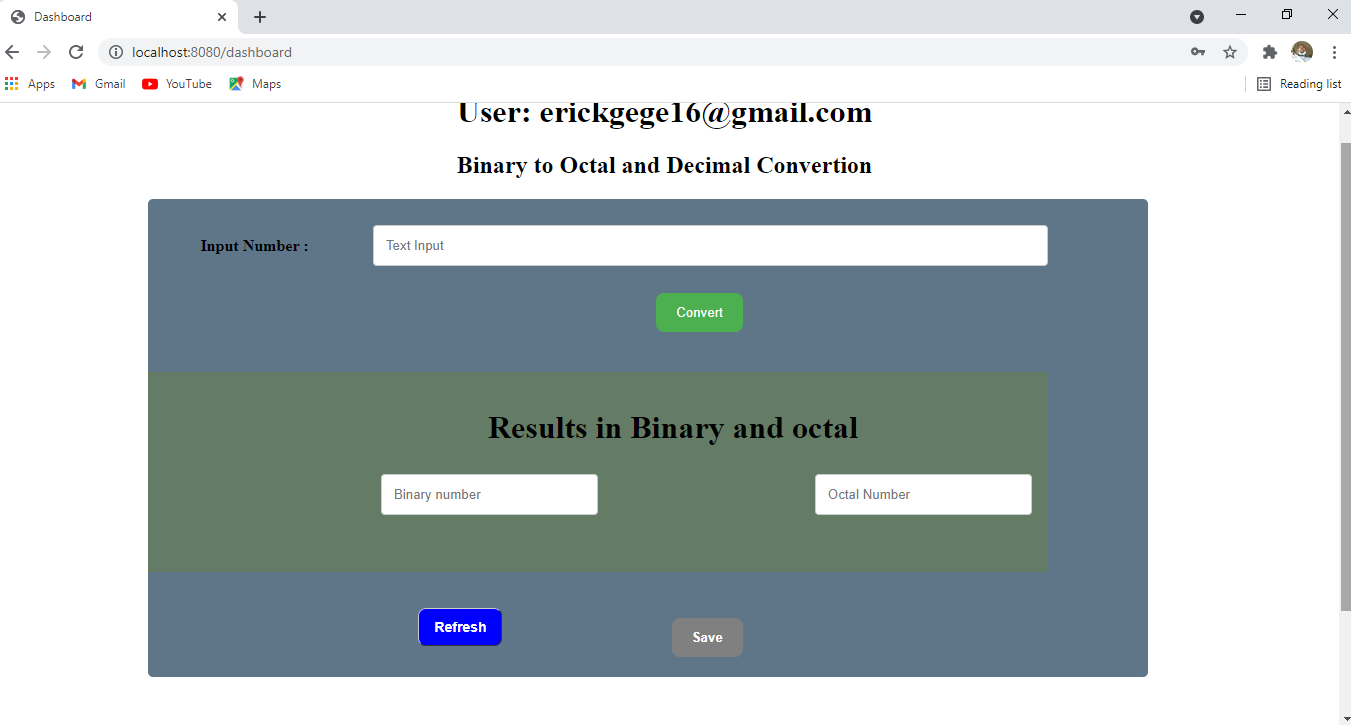
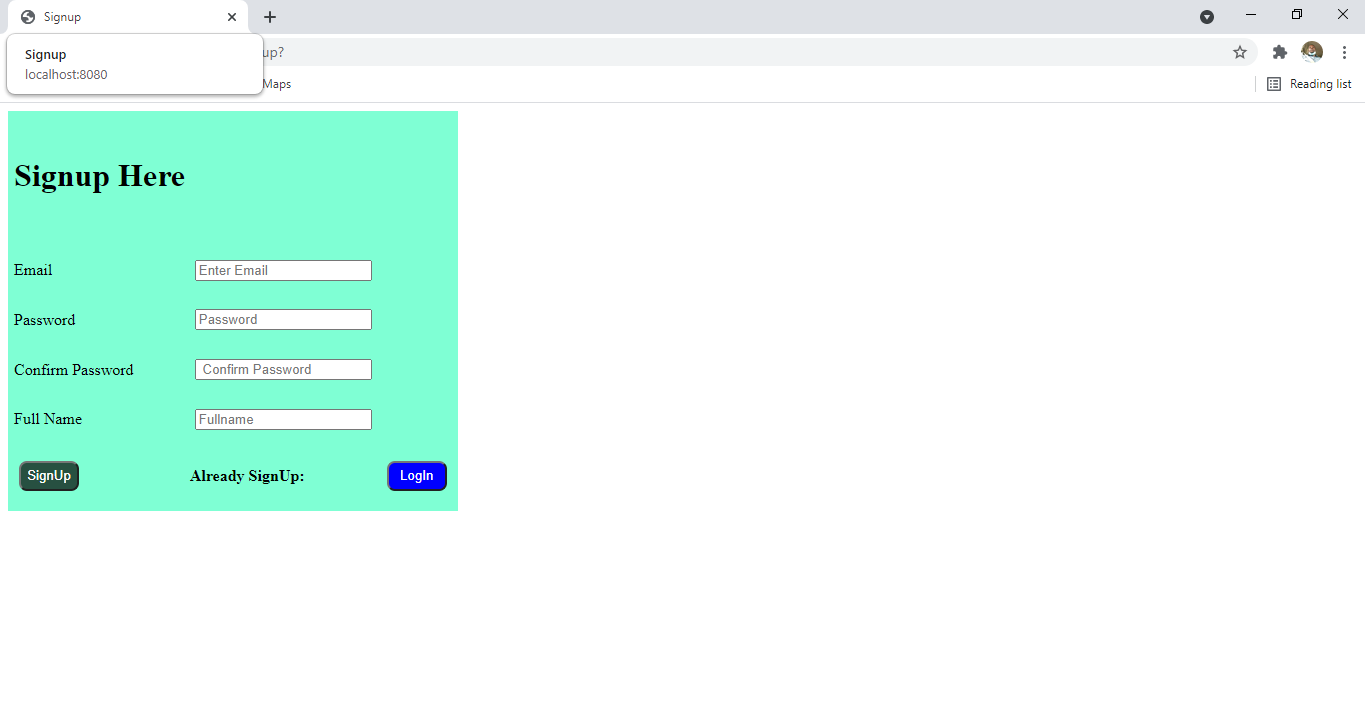
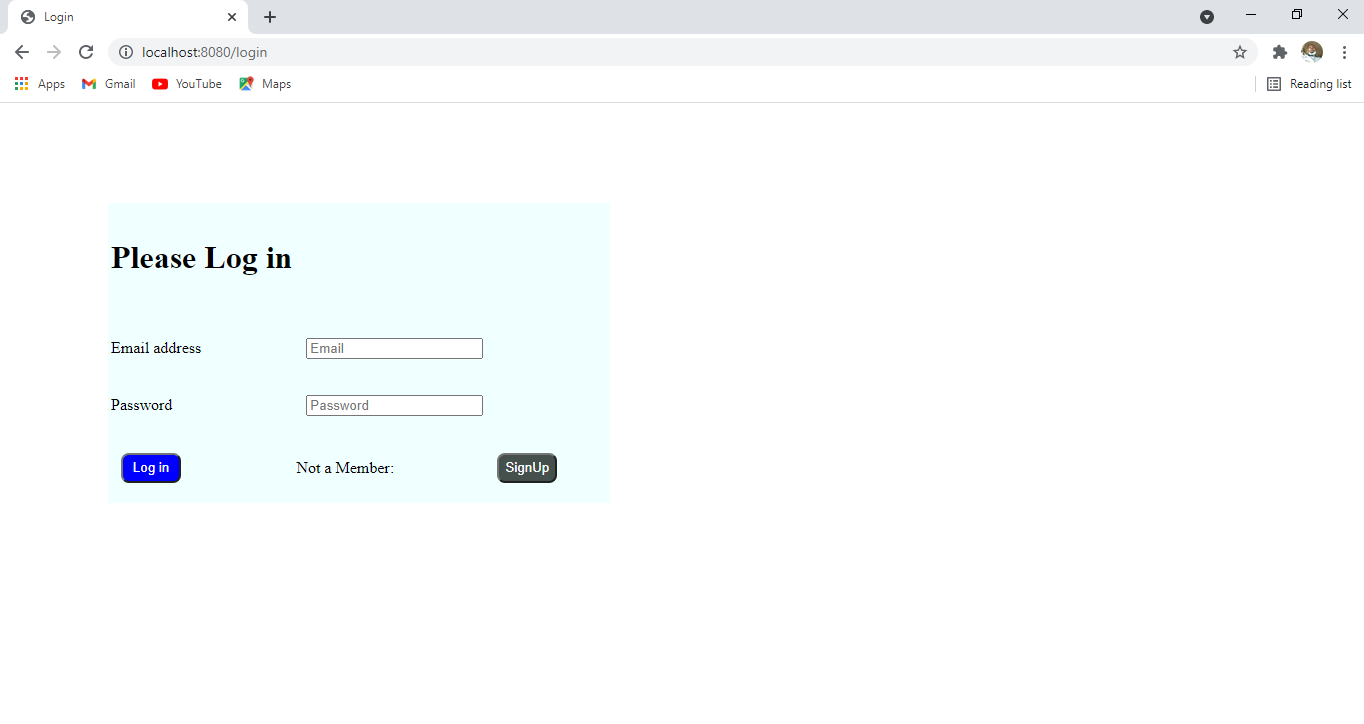
1. Make sure you have internet connection since the application will download the dependencies from the built gradle thus it time until all the gradle has built all the dependencies until you do not see any errors on your project as seen below. Now you are ready to run the application



1. Now click on the green arrow named run to run the application and the terminal console should bring something like the image below to show that the application started successfully. Kindly note the console may be below the IDE not necessary should be where mine is located.



1. Now launch any of your browser and type localhost:8080 as shown below to start the application on a browser as shown below. If your browser windows show you the interface below you are ready to use the application. Click the blue button with label here to either log in or sign into the application.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST PLAN (BLACK – BOX)** | | | | | |
| **Test** | **Description** | **Test Data** | **Expected Outcome** | **Actual Outcome** | **PASS/FAIL** |
| FR01 | Navigate to login page  Provide Valid Username  Provide Valid Password  Click on the Login Button | User = [example@gmail.com](mailto:example@gmail.com)  Password = 1234 | The user should be able to login | User is navigated to the dashboard with successful login | PASS |
| .. | .. | .. | .. | .. | .. |

|  |  |
| --- | --- |
| **POST CONDITIONS** | |
| FR01 | User is validated with database and successfully logged into account. The account session details are logged in the database |
|  |  |