Online Exam Management System

Installation Manual

# Table of Contents

[**Table of Contents 1**](#_heading=h.30j0zll)

[**Installing the Program for Users 2**](#_heading=h.3znysh7)

[Importing the Project from GitHub 2](#_heading=h.2et92p0)

[Before Running the Program 6](#_heading=h.ck6gvrycfvyy)

[Configuring the Build Path 9](#_heading=h.tyjcwt)

[**Installing the Program in Eclipse as a Programmer 14**](#_heading=h.4d34og8)

[Importing the Project with Projects from Git (smart import) 14](#_heading=h.2s8eyo1)

[Importing the Project Manually 16](#_heading=h.26in1rg)

[Setting Up the Database 19](#_heading=h.qnt1f3yxwm4u)

[The Program’s Database 20](#_heading=h.2bn6wsx)

[Running the Program after Setup 21](#_heading=h.tyaqzy1rgi9n)

[Viewing the Packages 25](#_heading=h.4i7ojhp)

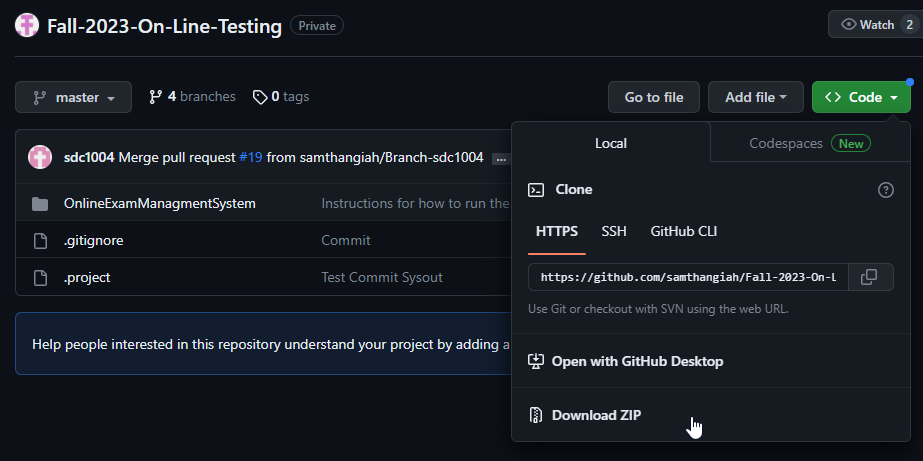
# 

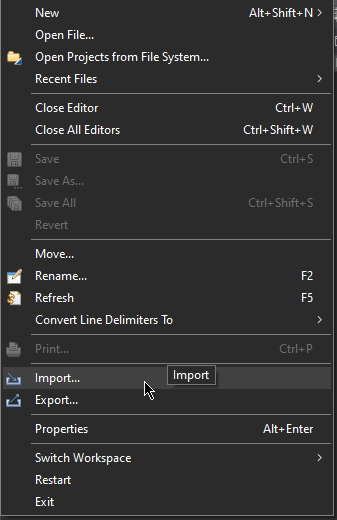
# Installing the Program for Users

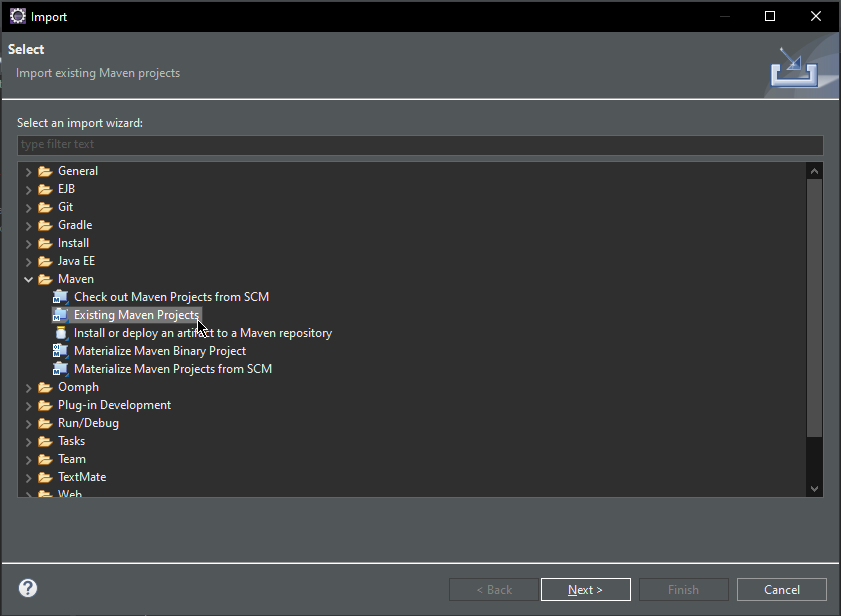
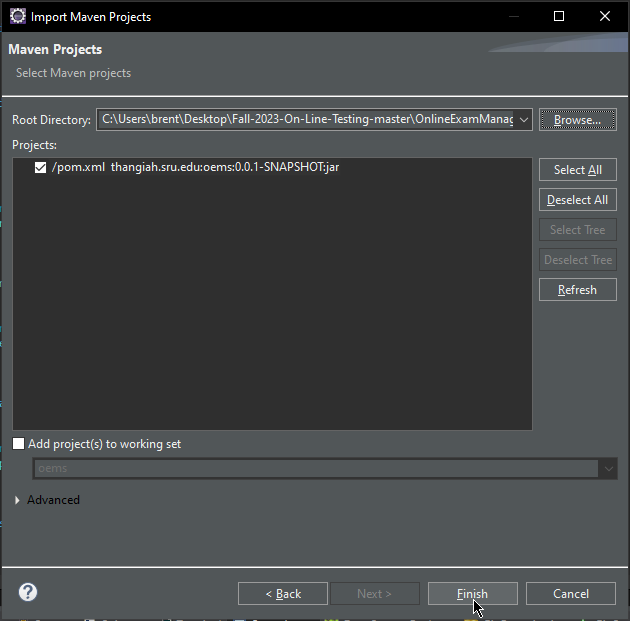
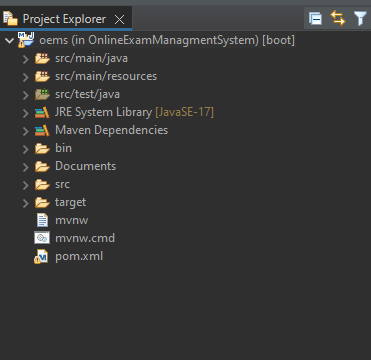
This section of the manual shows how to install and run the program by downloading it from the GitHub repository. This section should **not** be used by programmers, as it does not demonstrate how to set up the local repository for adding to the source code.

## Importing the Project from GitHub

The following steps will outline how to import the project into Eclipse for running the program:

1. From the GitHub repository page, click on **Code** and then **Download ZIP.** Extract the folder to a desired location.
2. While in Eclipse, you’ll need to import the project by choosing **File > Import**

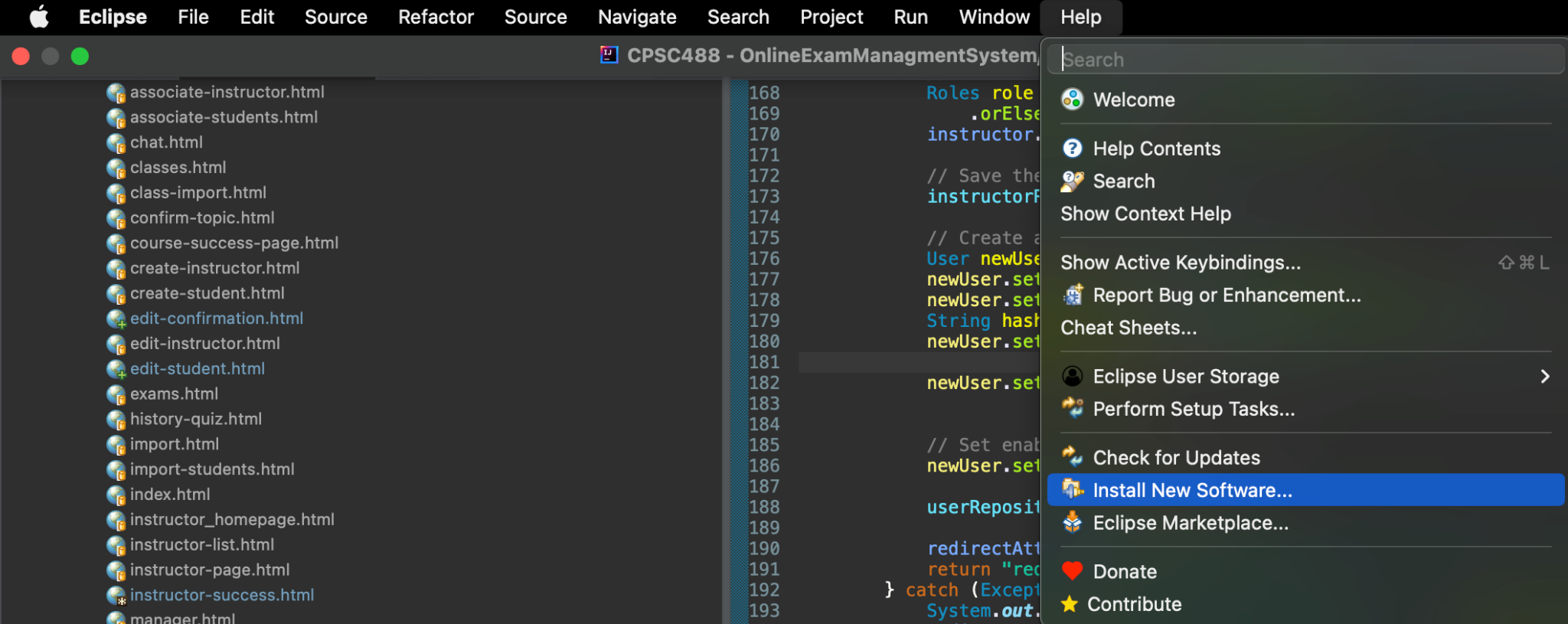


1. After selecting import, choose **Maven > Existing Maven Projects** and then hit **Next**  
   
2. Locate the directory of where you extracted the program by clicking on **Browse**. Once you find it, open the folder that is named **OnlineExamManagementSystem** for the import. Then click **Finish**.  
     
   
3. The project is now imported into the Eclipse workspace.  
   

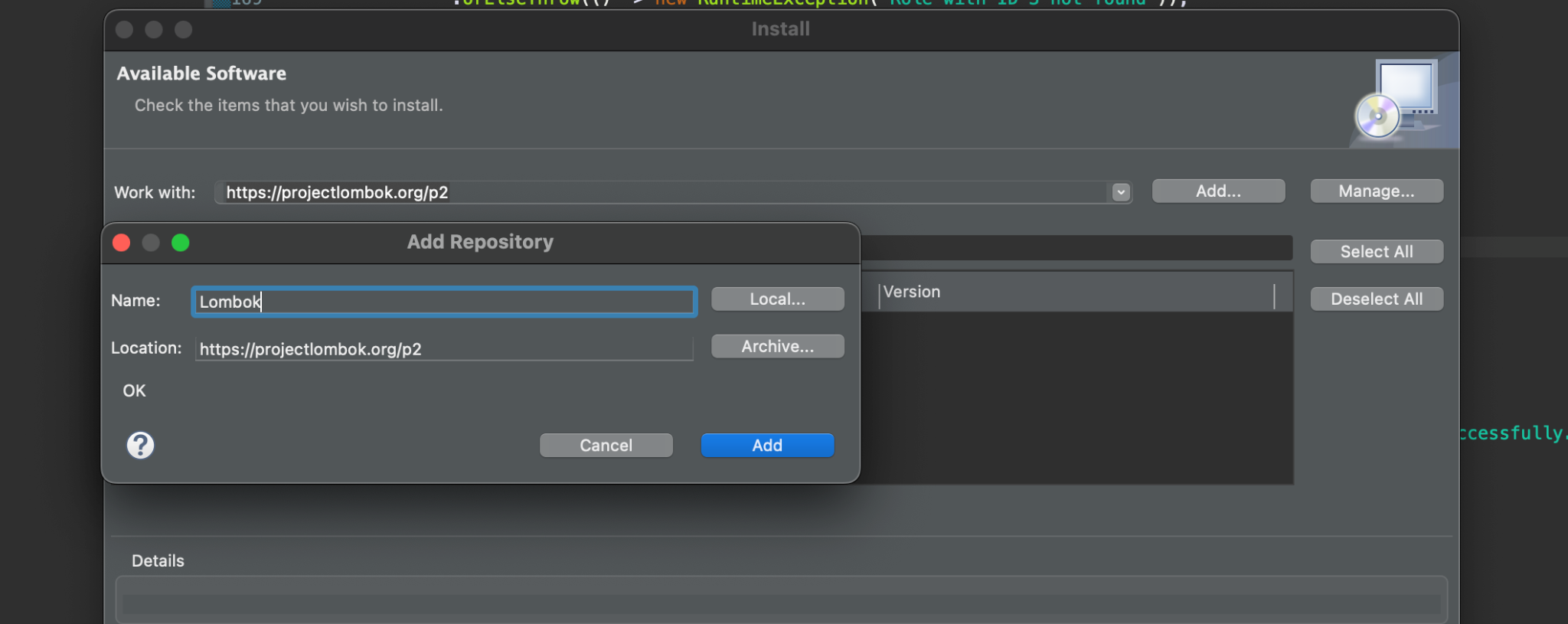
## Before Running the Program

After getting all the program files open in Eclipse, you are likely to be faced with a lot of errors in the code, that will prevent you from running the program. This is due to a common issue implementing the lombok library with the eclipse IDE. You only need to do this once.

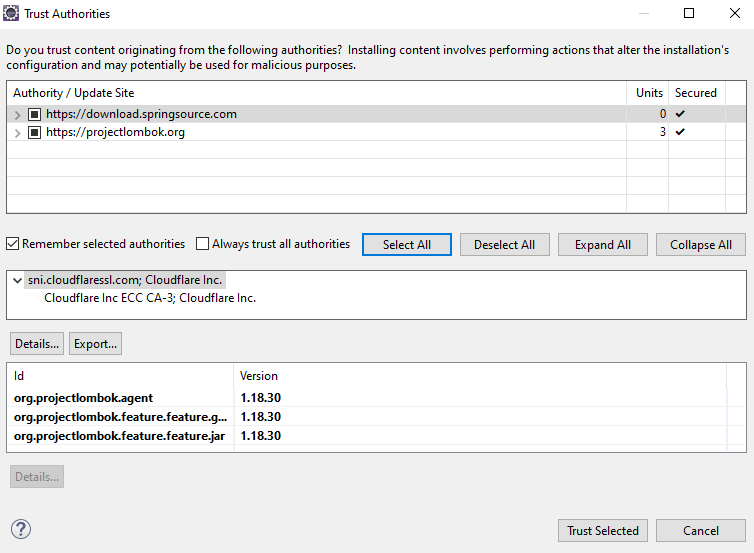
1. You’ll need to go to the “Help” option in the top navbar, then “Install New Software”



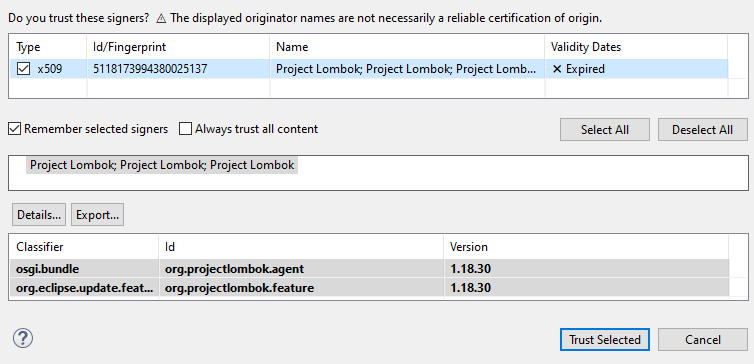
1. In the “Available Software” input “<https://projectlombok.org/p2>” in the text box labeled as “Work With” and press “Add”



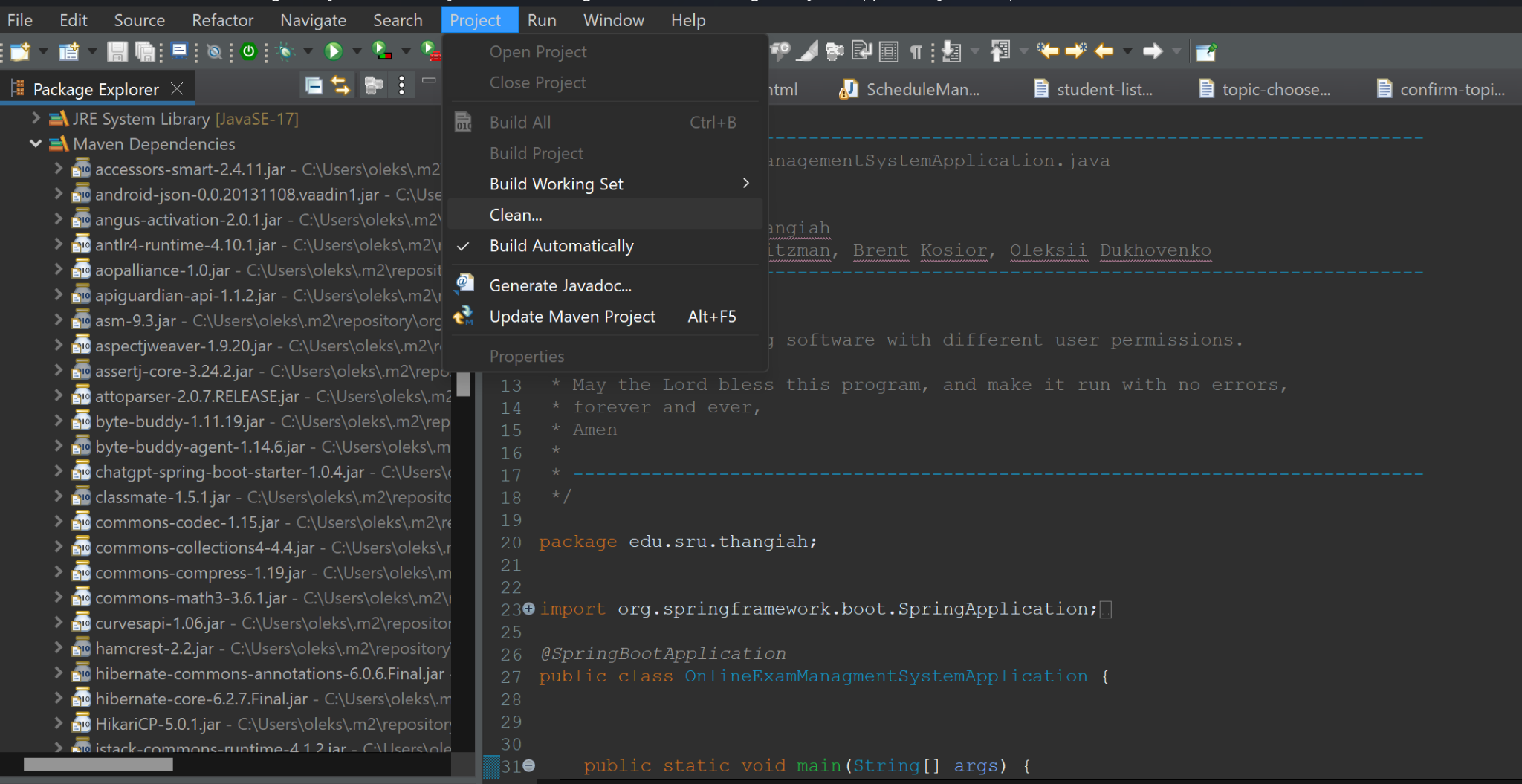
1. You will be prompted to add a repository, name the repository “Lombok” and use the default URL provided and press “Add.”
2. A Lombok dependency should pop up in the box below, press “Select All” then “Next” at the bottom.
3. Lomboks “Install details” will be provided, you just need to press “Next” and accept the Lombok Agreement.
4. Press “Finish”
5. Next Eclipse IDE will ask for trusted certificates, press “Select All” and “Trust Selected”



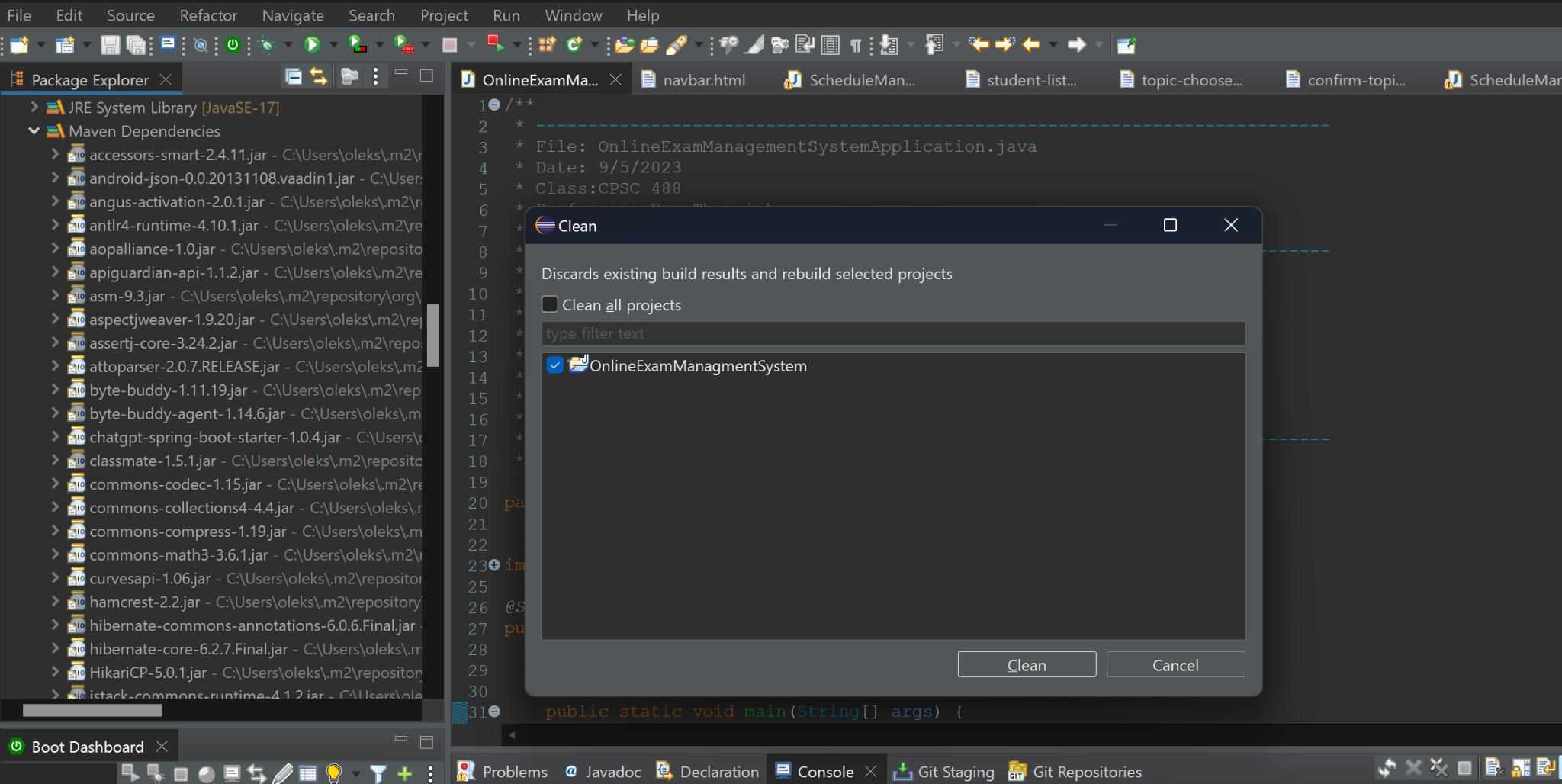
1. It will ask if you trust the signers, “Select All” and “Trust Selected” once again.



1. Eclipse will then prompt the user to restart Eclipse to install the lombok library.
2. After Eclipse restarts, Go to **Project**→ and click **Clean**

****

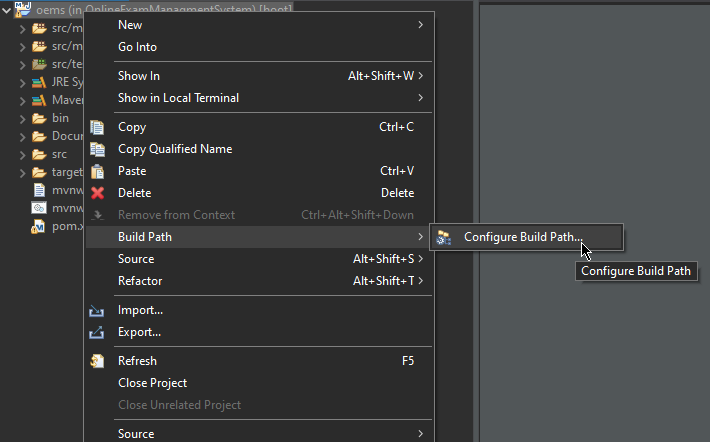
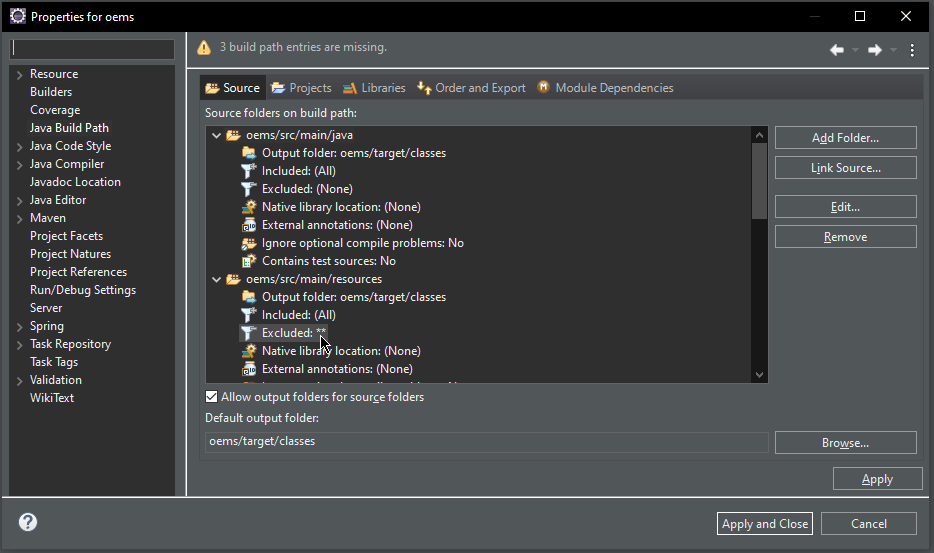
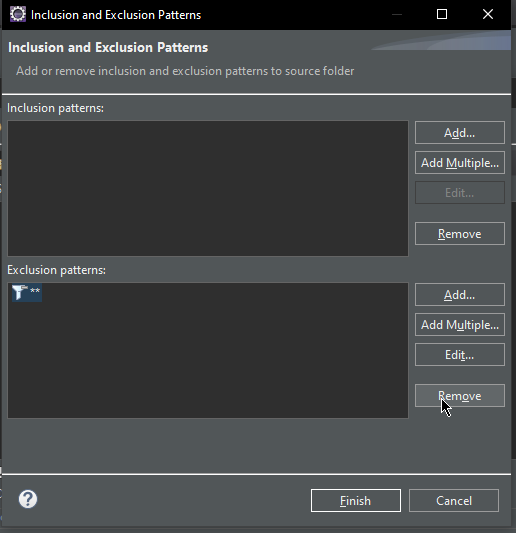
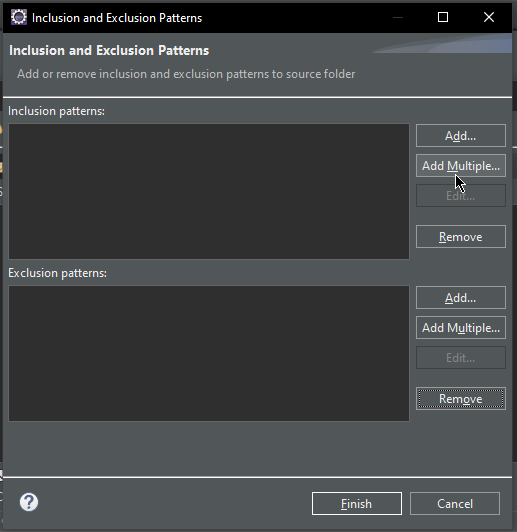
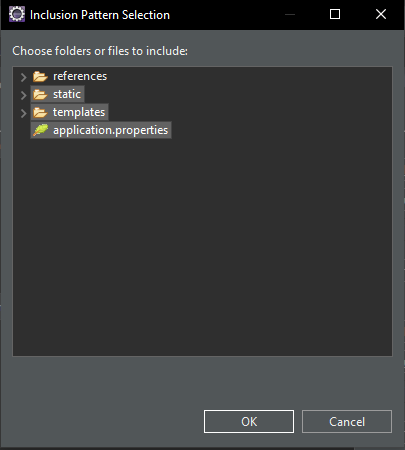
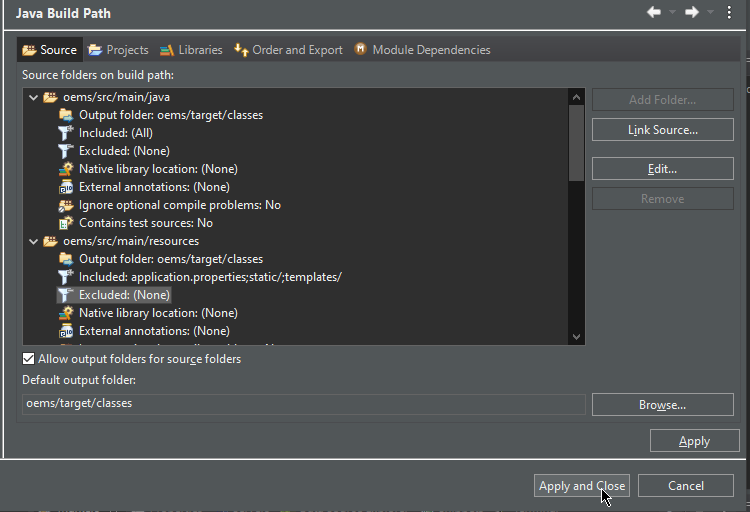
1. Uncheck ‘**Clean all projects**’ , select the **OnlineExamManagementSystem** project and click on **Clean**

****

1. Now, the program should be good to run with no errors.

## Configuring the Build Path

Eclipse is unable to find the folder that contains all of the HTML pages of the program by default. In order to fix this, the build path needs to be configured using the following steps:

1. After Importing the Maven project into Eclipse, right click on the main folder and choose **Build Path > Configure Build Path**  
     
   
2. Once in the configuration menu, double click on “Excluded” under the oems/src/main/resources folder  
     
   
3. Click on the item in the Exclusion patterns box, and then click “Remove”  
     
   
4. In the Inclusion Patterns section, select “Add Multiple”  
     
   
5. Select the templates folder, static folder, and application.properties and then click “OK”  
   
6. Click “Finish”
7. Click “Apply and Close”.  
   

# 

# 

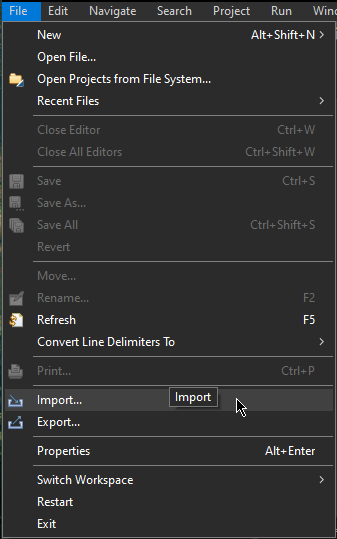
# 

# Installing the Program in Eclipse as a Programmer

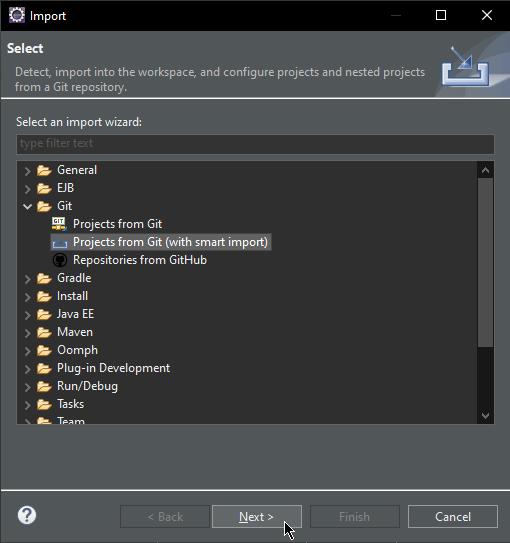
This section should only be referred to when attempting to set up a local git repository to work on and use the program. Otherwise, for normal use and just running the program, the prior method of installing the program should be used.

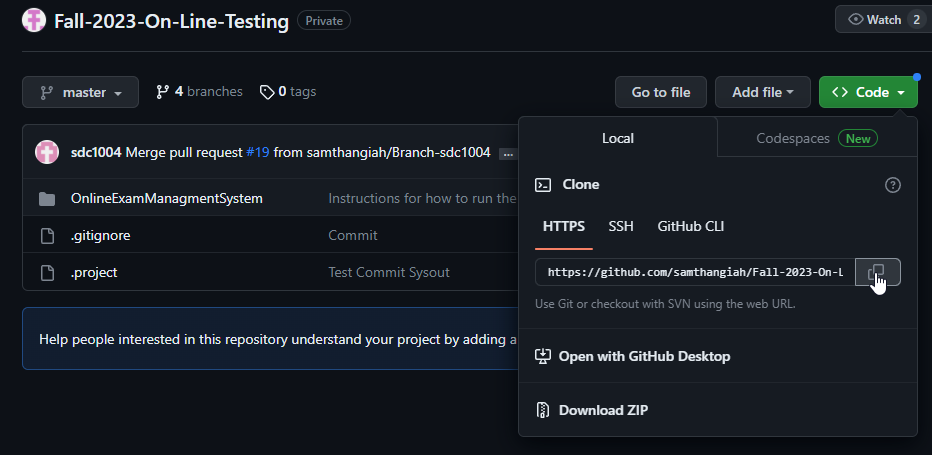
## Importing the Project with Projects from Git (smart import)

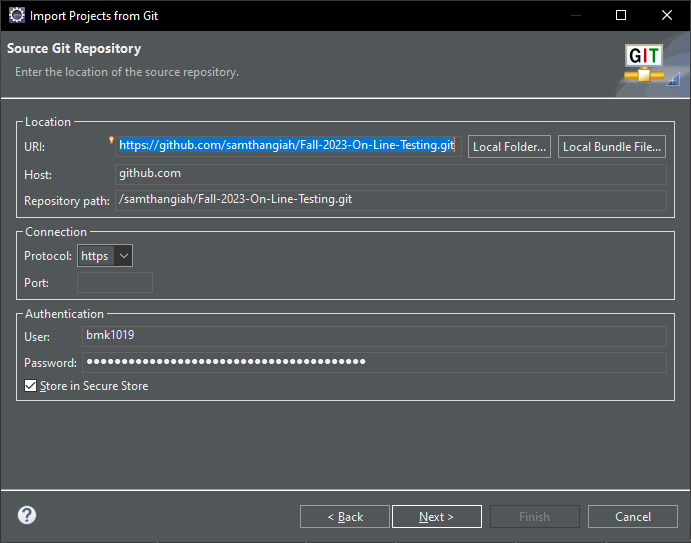
1. After installing the program requirements, launch the Eclipse IDE and select/create a location for your workspace to be stored with the **Browse...** option and then select **Launch** when the workspace has been set.
2. Navigate to the top toolbar to the **File** option and select **Import…** as shown below.

****

1. Once you’re in the import window, click the **Git** folder arrow to expand what’s inside and select **Projects from Git (with smart import)** as seen below.



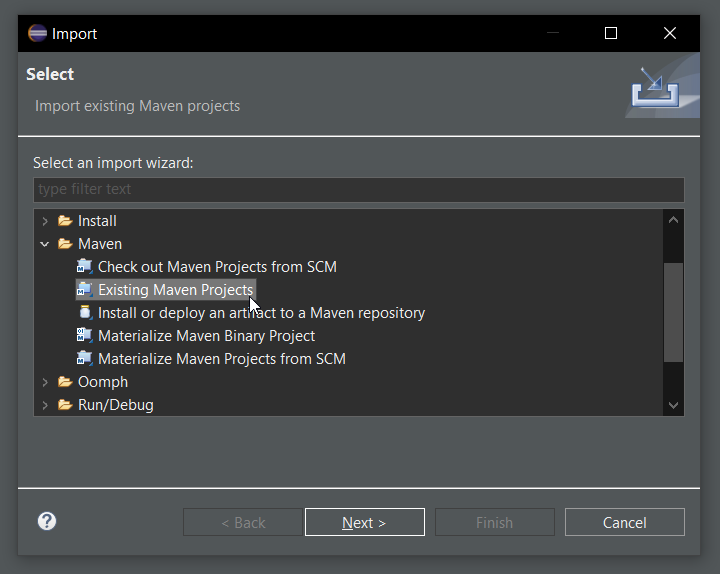
1. In the **Import Projects from Git** windows, select **Clone URI** and press next
2. In the next section, navigate to the GitHub, login, generate a code, and go to the project’s specific page and press on the green **Code** button and copy the URI from there. A visual of what to see is shown below
3. Next, take that URI and paste it into the URI text box and the rest will automatically fill in, an example of what it should look like is displayed below.



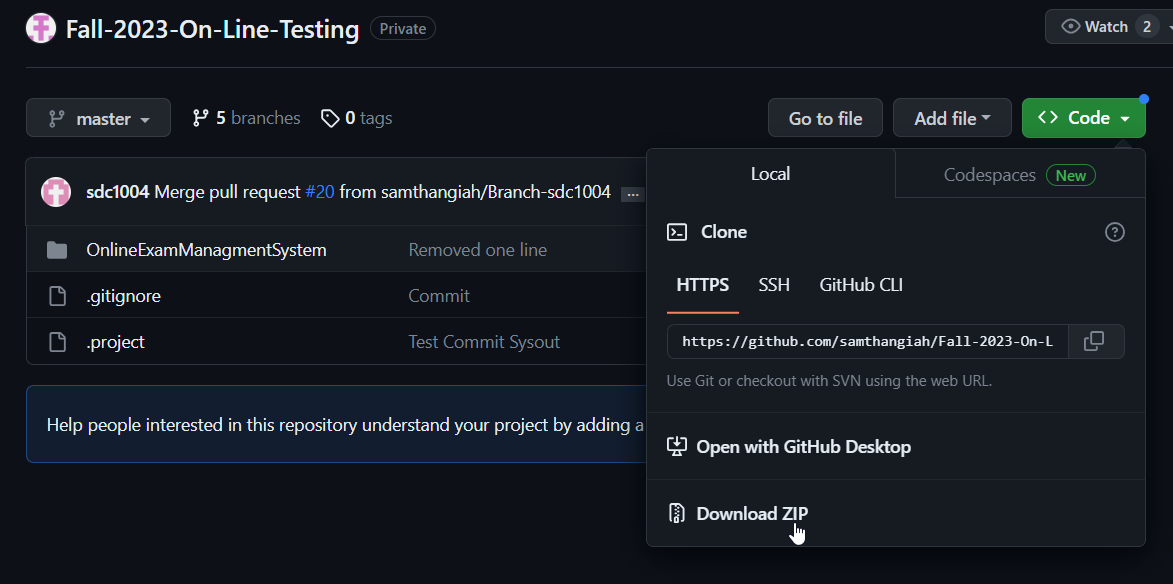
1. Select **Next** and on the new page, search for the branch **master** and select **Next** once more. You’ll be taken to a page where you must select the directory of the destination of the GitHub files. Either leave it as the default location or user **Browse** to manually choose where to place the files.
2. After having a directory for the GitHub files selected, continue with the **Next** button and wait for everything to be downloaded. Select **Finish** once you have the opportunity to do so.

## Importing the Project Manually

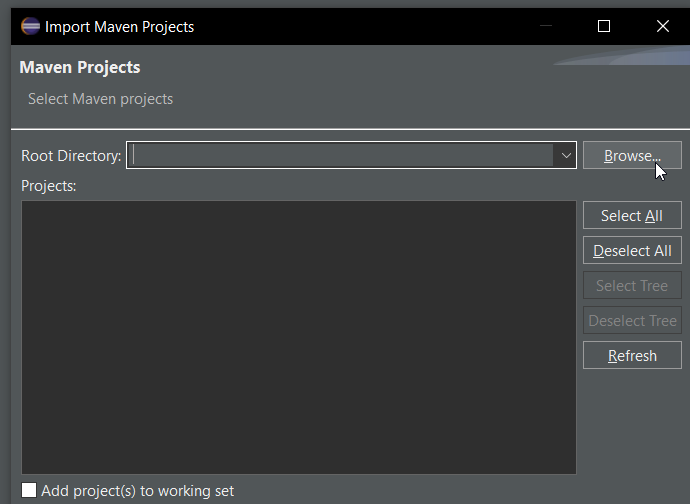
1. Once you’re in the import window, click the **Maven** folder arrow to expand what’s inside and select **Existing Maven Projects** and press **Next** as seen below.

****

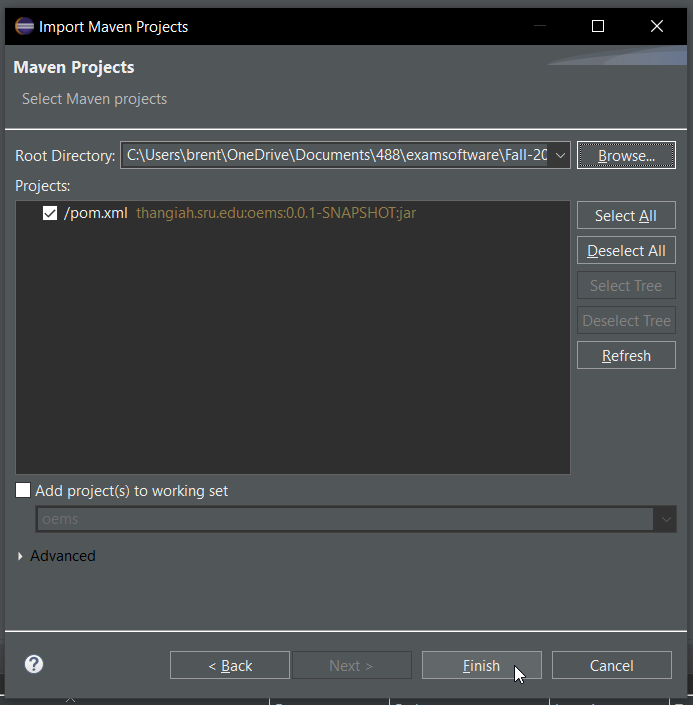
1. You must download the zip file of the master branch of the project by going to the GitHub page for the project and clicking on the green **Code** button and pressing **Download ZIP** button. Once the download is complete, you must unzip the file into a folder with your decompressing program of choice (Winrar, 7zip, etc). There is a built-in unzipping ability Windows has, but look elsewhere for more information on how to unzip a file on the internet.



1. On the next page, you’ll need to press the **Browse…** next, navigate to the unzipped folder that you previously downloaded from the GitHub.



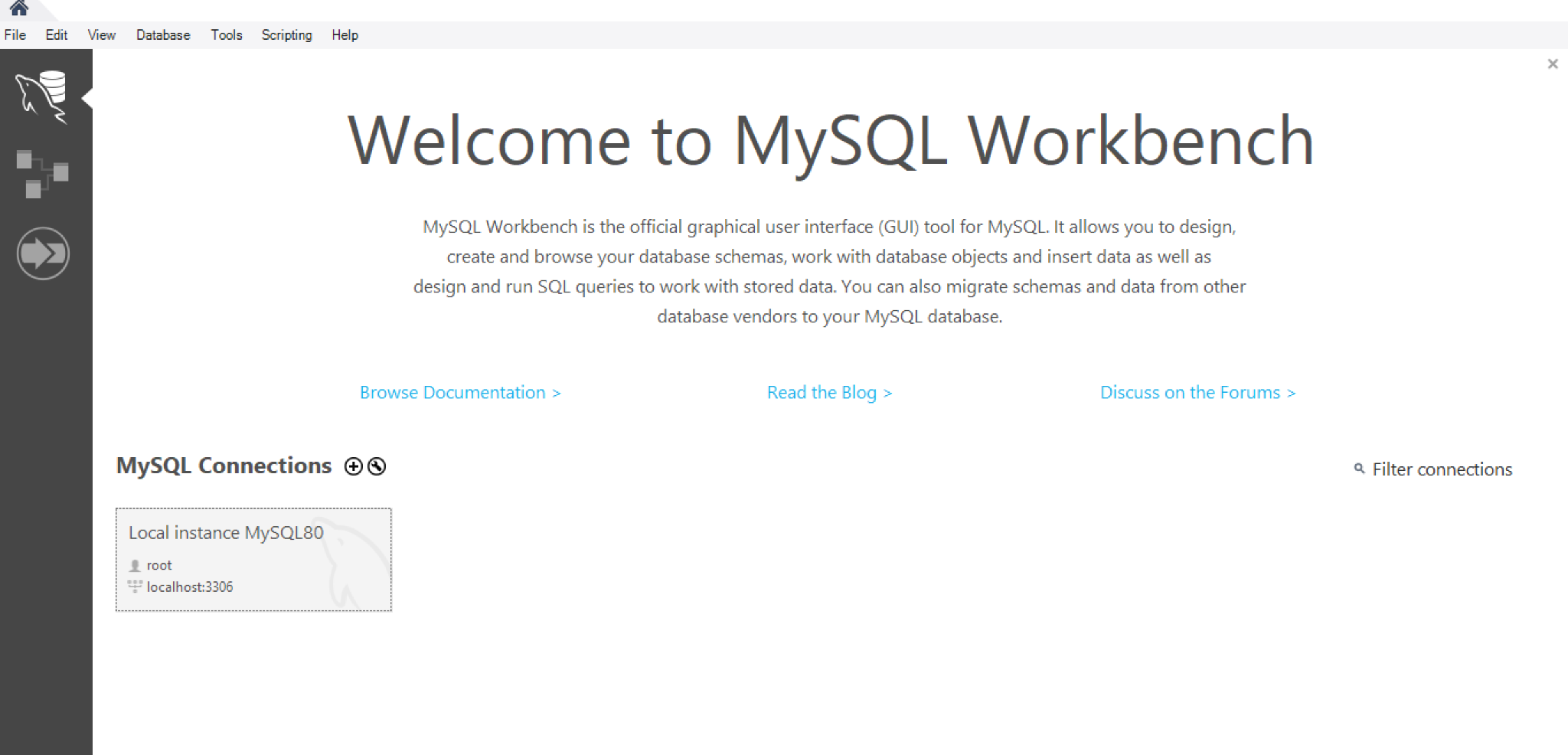
1. You will then select the pom.xml file that is displayed in Projects, then click finish.

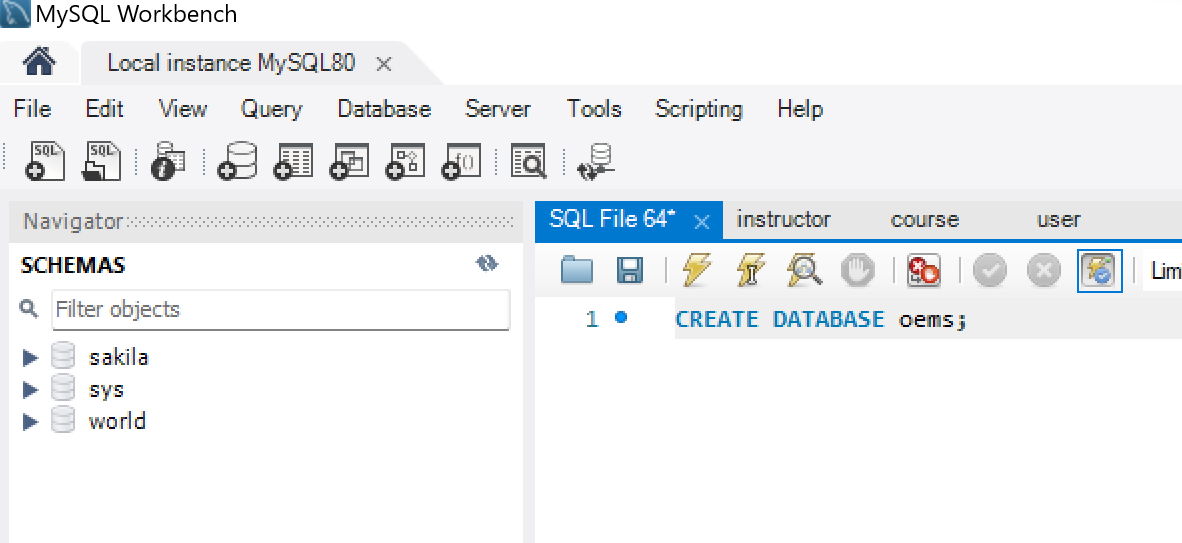


## Setting Up the Database

After getting all the program files open in Eclipse you’ll want to set up the database.

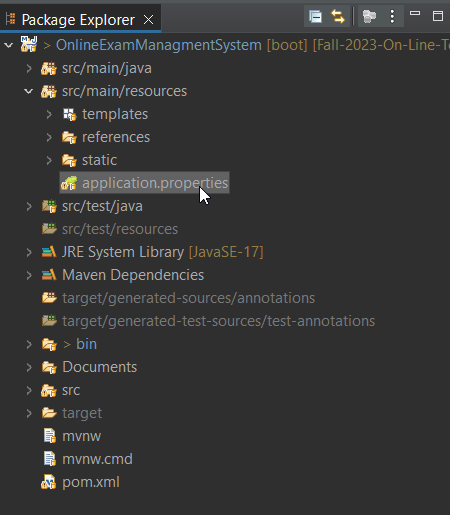
1. You’ll need to go to your mySQL Workbench



1. Open mySQL instance and write the password, which is **‘software’**
2. IF you have ‘oems’ databases from any previous versions of our software - go ahead and delete them.
3. Create a new schema
4. Write an SQL command to create a new database called ‘oems’, as such:
   1. 
5. Execute

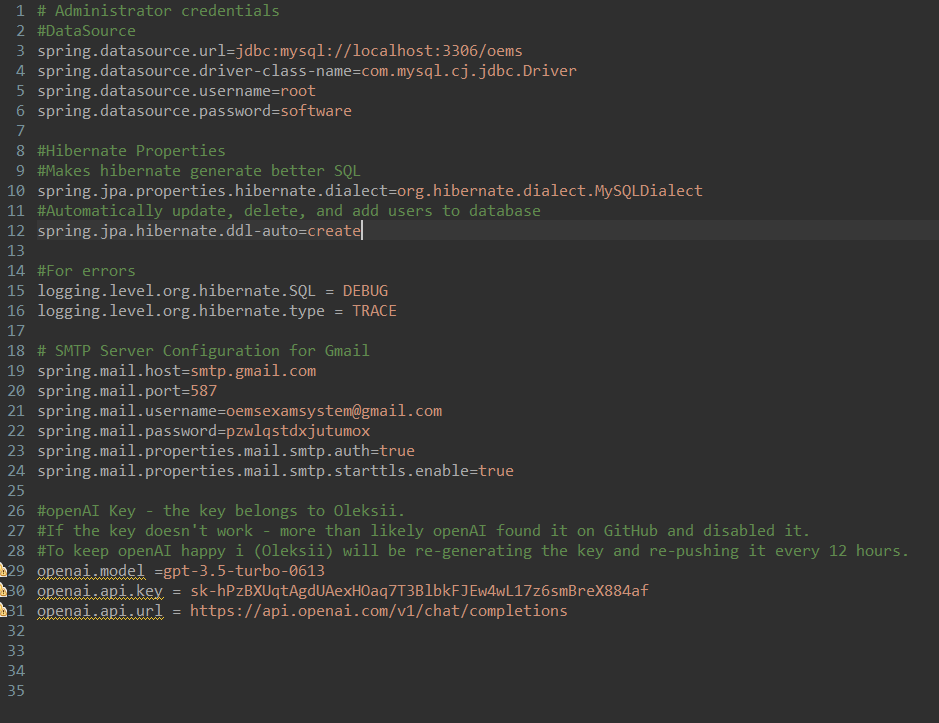
## The Program’s Database

1. After setting up the MySQL database, you need to go into the Java project and go to **src/main/resources** folder (or into **src** then **main** then **resources)** to edit the **applications.properties** file.



1. After opening the file be sure to check the highlighted line to ensure it is set to:

spring.jpa.hibernate.ddl-auto=create

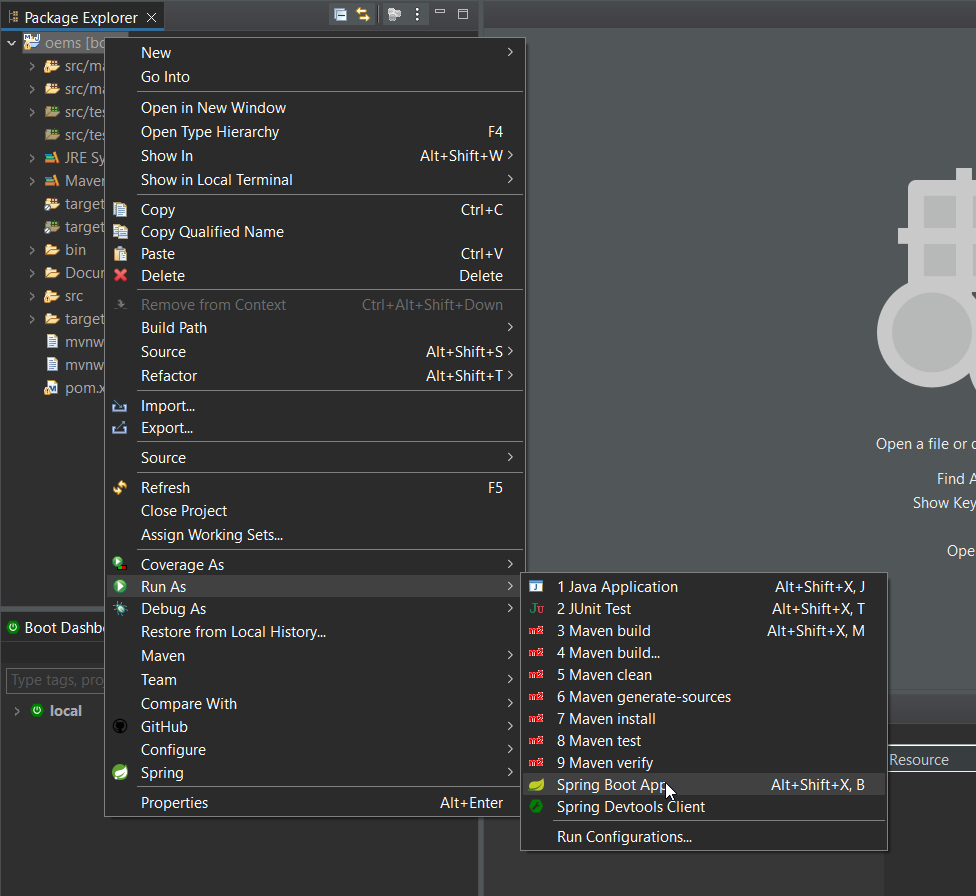


1. If needed you can change the spring.datasource.username and spring.datasource.password to reflect your respected MySQL server’s username and password and save the changes.

## Running the Program after Setup

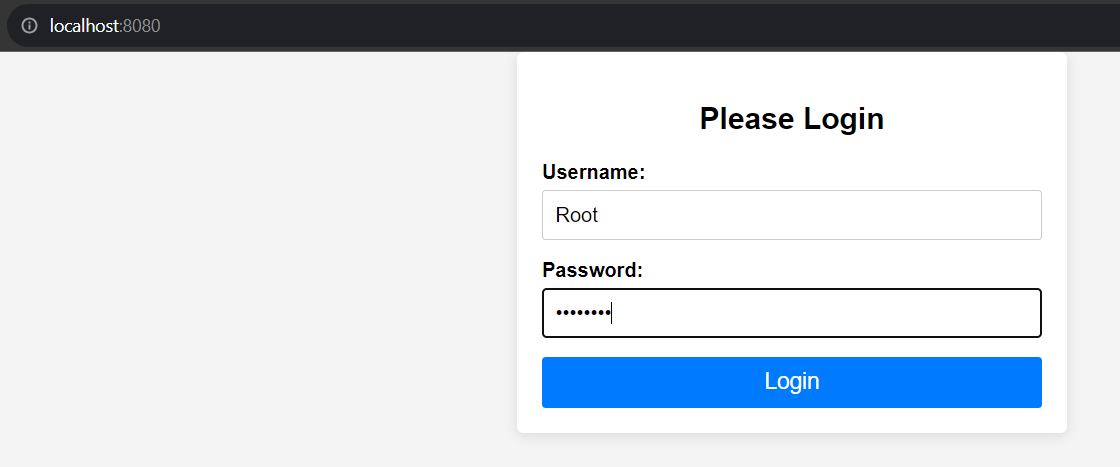
After getting all the program files open in Eclipse you’ll want to run the program.

1. You’ll need to right click on the top of the files tree, hover over **Run As**, then click Spring Boot Apps

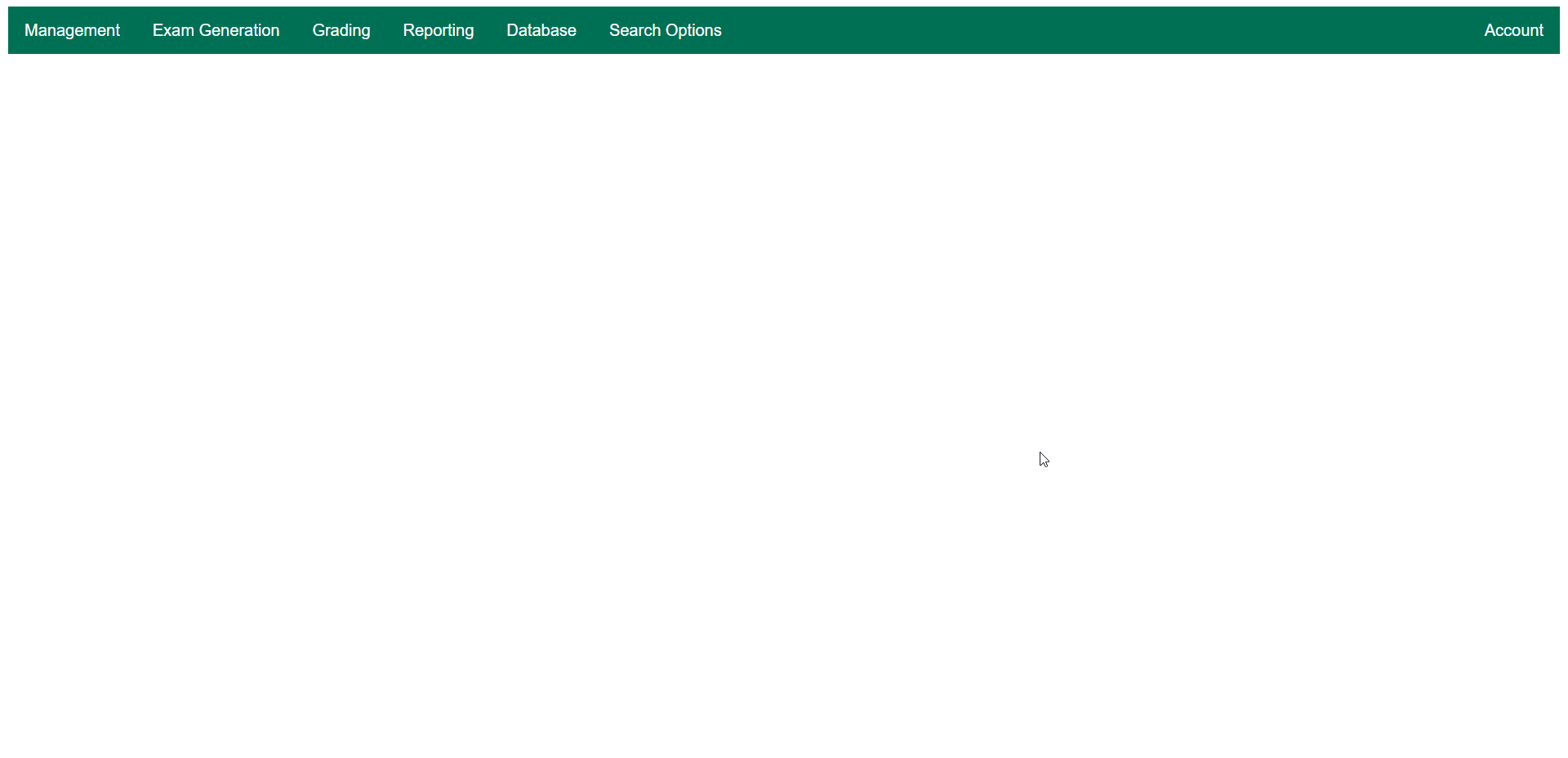


1. You will then need to navigate to your browser of choice and type *localhost:8080* into the url bar. The login page will be displayed.

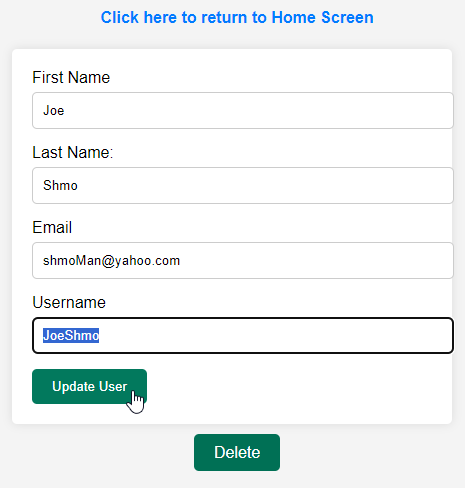
*user: Root, pass: software*



1. After Logging in you will be presented with the Main Screen for an instructor.



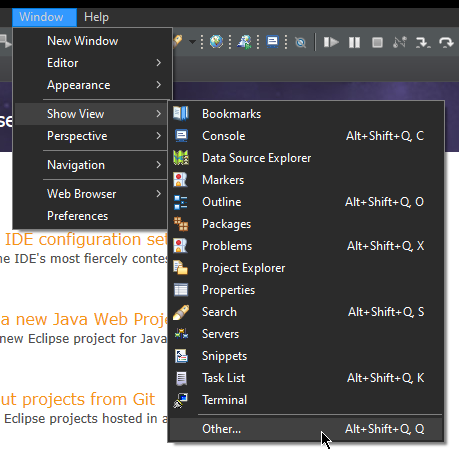
1. You have a navigation bar that has multiple options, the current options that are available are Management, Computer Literacy, Grading, Reporting, Database, Search, Options.
   1. Management has 5 options under it
      1. Accounts - ***Currently Not Functioning***
      2. Create Account - This option will take you to the Registration Form page. You are able to enter user information and click finish to register that user to the database.
      3. Classes - Dropdown menu with classes will appear. The chosen class will be redirected to an examination page. This is still ***under development***
      4. Create Schedule Manager, this form allows administrators to create a schedule manager, who can login.
      5. Chapters - This lets the user choose between chapters and different examination forms. This is ***under Development***
      6. Exams - Allows the user to pick a class and get redirected to the exam
   2. Exam Generation
      1. Exam Generator - A full-fledged AI-Exam generator. Simply pick a topic and it will generate an exam for you. See ‘A.I. Exam Generation’ for in-depth user manual information.
   3. Grading
      1. Exam Grading - ***Currently Not Functioning***
   4. Reporting
      1. Exam information - ***Currently Not Functioning***
      2. IP Address - ***Currently Not Functioning***
   5. Database
      1. View All Students - Routes to an HTML page that shows all students in the database.
         1. There is an edit feature when viewing all students as well, you can click edit to the right hand side of any student currently enrolled and it will route you to a new page to edit the students details, or delete the student all together.



* + 1. Add Students - Routes to an HTML page that allows the admin to add students to the database.
    2. Add Instructors - Routes to an HTML page that allows the admin to add instructors to the database.
    3. Add Courses- Routes to an HTML page that allows the admin to add courses to the database.
    4. Assign Students - Allows the admin to assign existing students to existing classes
       1. The association will be recorded and uploaded to the student\_and\_courses table
    5. Assign Instructors- Allows the admin to assign existing instructors to existing classes
       1. The association will be recorded and uploaded to the instructor\_courses table
    6. Export to Excel - Exports the students from the database to your local drive as an excel file. By default, it should be downloaded to your ‘Downloads’ folder
    7. Import with Excel - Allows an import of an Excel File that uploads students into the database. Checks if usernames are unique.
  1. Search Options
     1. View Details - ***Currently Not Functioning***
  2. Account
     1. Change Password - ***Currently Not Functioning***
     2. Logout - This option will take you back to the login page.

## Viewing the Packages

1. In order to view the project’s packages, press the **X** right next to the **Welcome** tab in the top left corner or go to the top toolbar and press **Window** then **Show View** then **Other** in order to bring up a window.



1. A **Show View** window will display, and you must search for a folder named **Java** and expand it with the arrow right next to it to display the folder’s contents. Select **Package Explorer** and press the **Open** button.

## 