Web-Based Evaluations

Installation Manual

Revised by Chris Thornton

9/7/2023

# Table of Contents

[**Table of Contents 2**](#_1bz3ym1z0dt3)

[**Installing the Program as a User 3**](#_9wadzcmk5h96)

[Importing the Project from GitHub 3](#_pt7mpzj1tvta)

[Configuring the Build Path 6](#_hfyc63t1tcaz)

[Configuring the MySQL Database 10](#_bedjirl52hbm)

[**Installing the Program in Eclipse as a Programmer 12**](#_fvl7txie6lkr)

[Importing the Project Start 12](#_m3il7st4b7nv)

[Importing the Project with Smart Import 13](#_2t4bjprgjnr0)

[Importing the Project Manually 15](#_finsz8jugr0z)

[Running the Program with Data Persistence 17](#_xn6dd17u1nlq)

[Drop Old SQL Schema 17](#_iub547jnv1hy)

[Running the Program Once Using the Create Schema 18](#_9d9ovkw9assg)

[Changes to Code 18](#_7tdo7bp9pz21)

[WebBasedEvaluationsApplication.java 18](#_ulzft6rivsjf)

[application.properties 19](#_vczdbwi4cb6m)

[Viewing the Packages 19](#_ql9500kd52e2)

[Resolving Build Errors 21](#_2tnc1znyvxp)

[Missing Libraries Problems 24](#_3l36yi49ju5f)

[The Program’s Database 26](#_fiybbo4xldtl)

[Login Information 27](#_slhylwwbqsjq)

[Continued Maintenance](#_t9klsywn2wus) 28

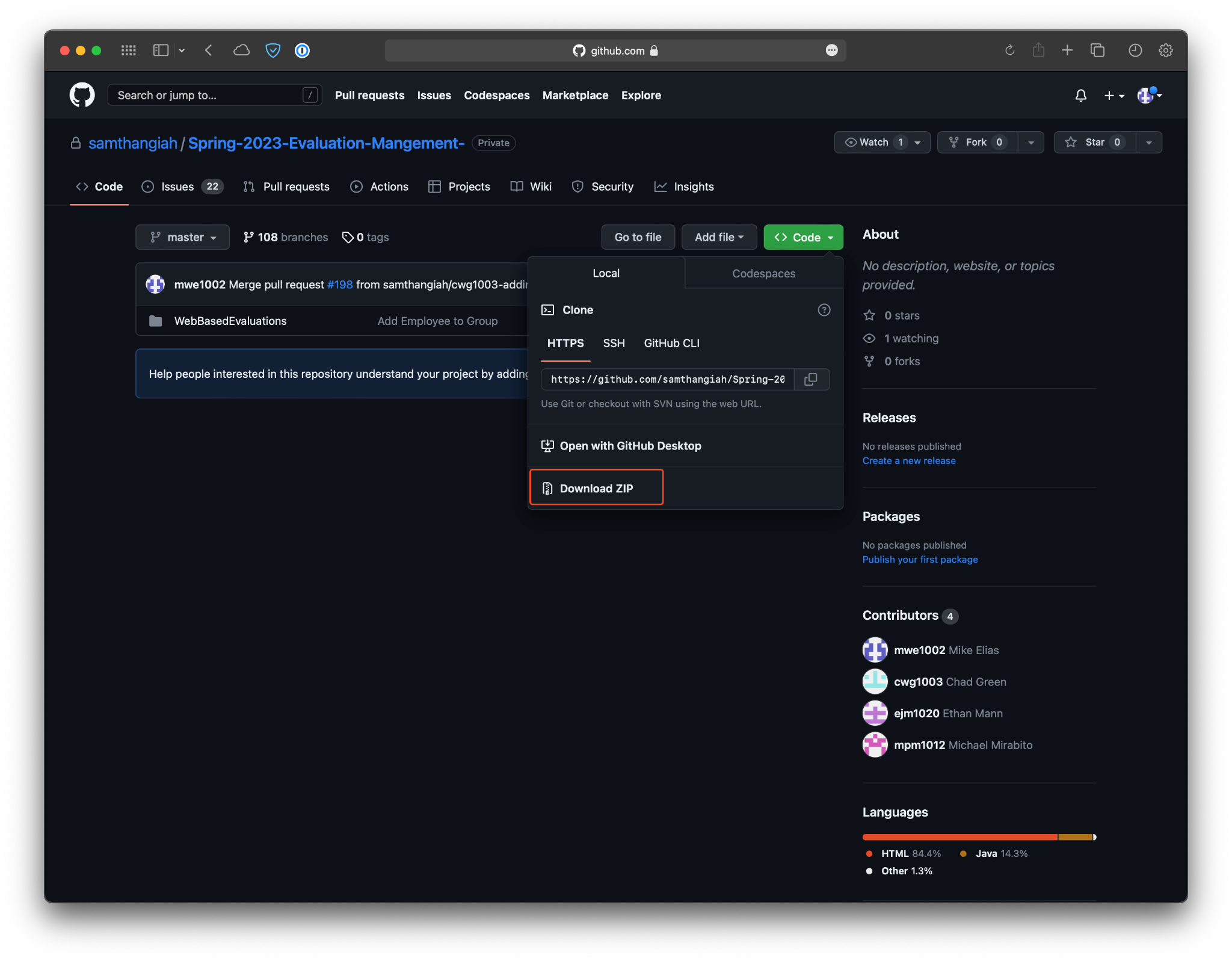
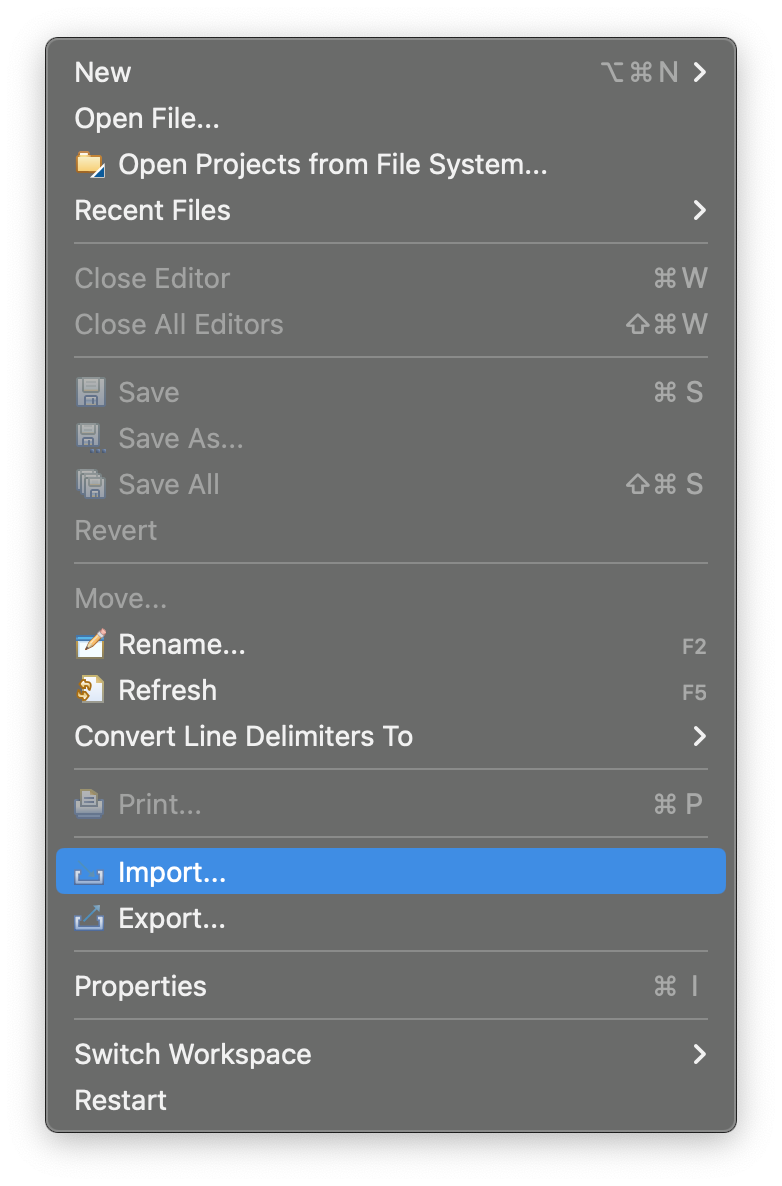
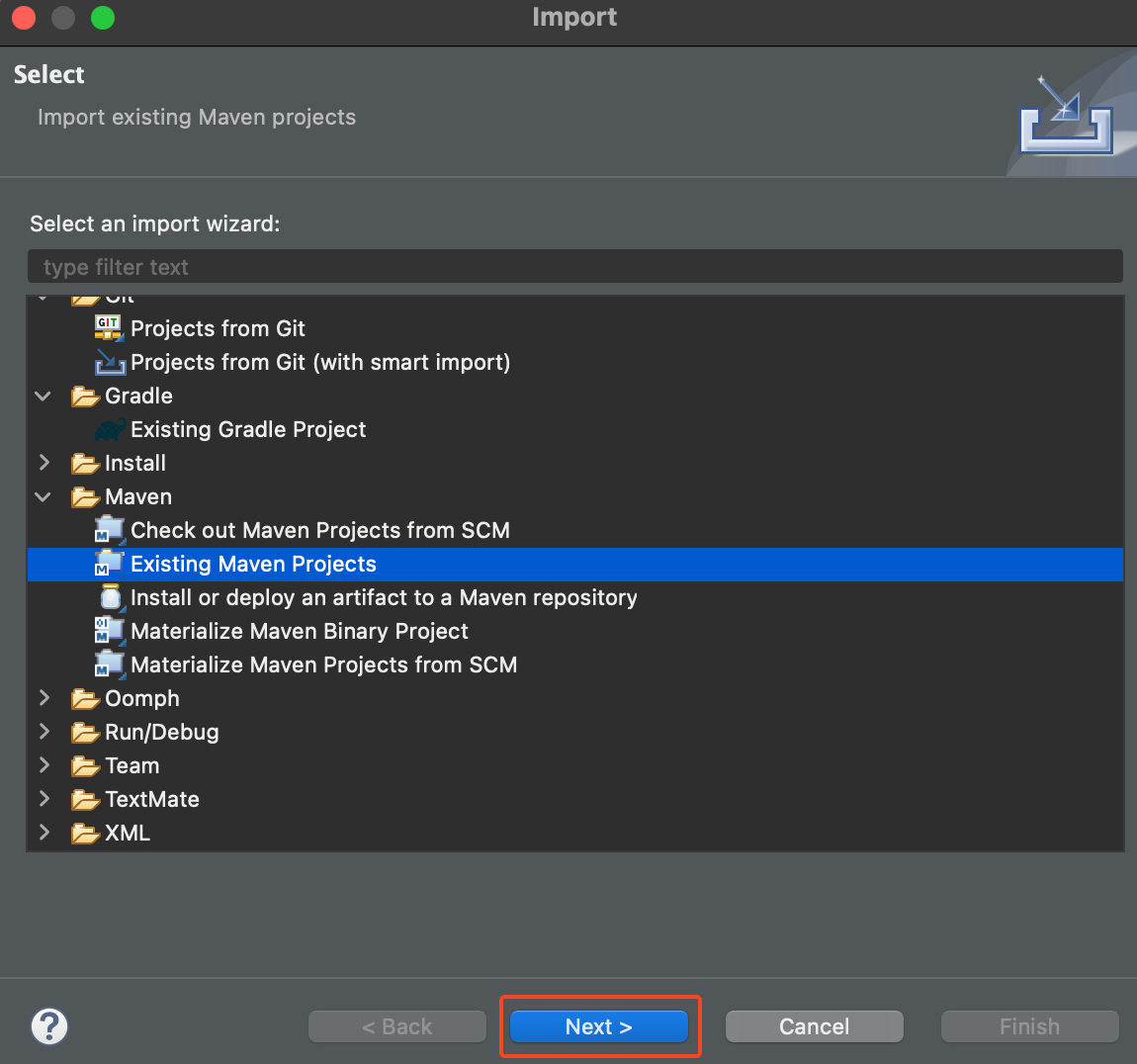
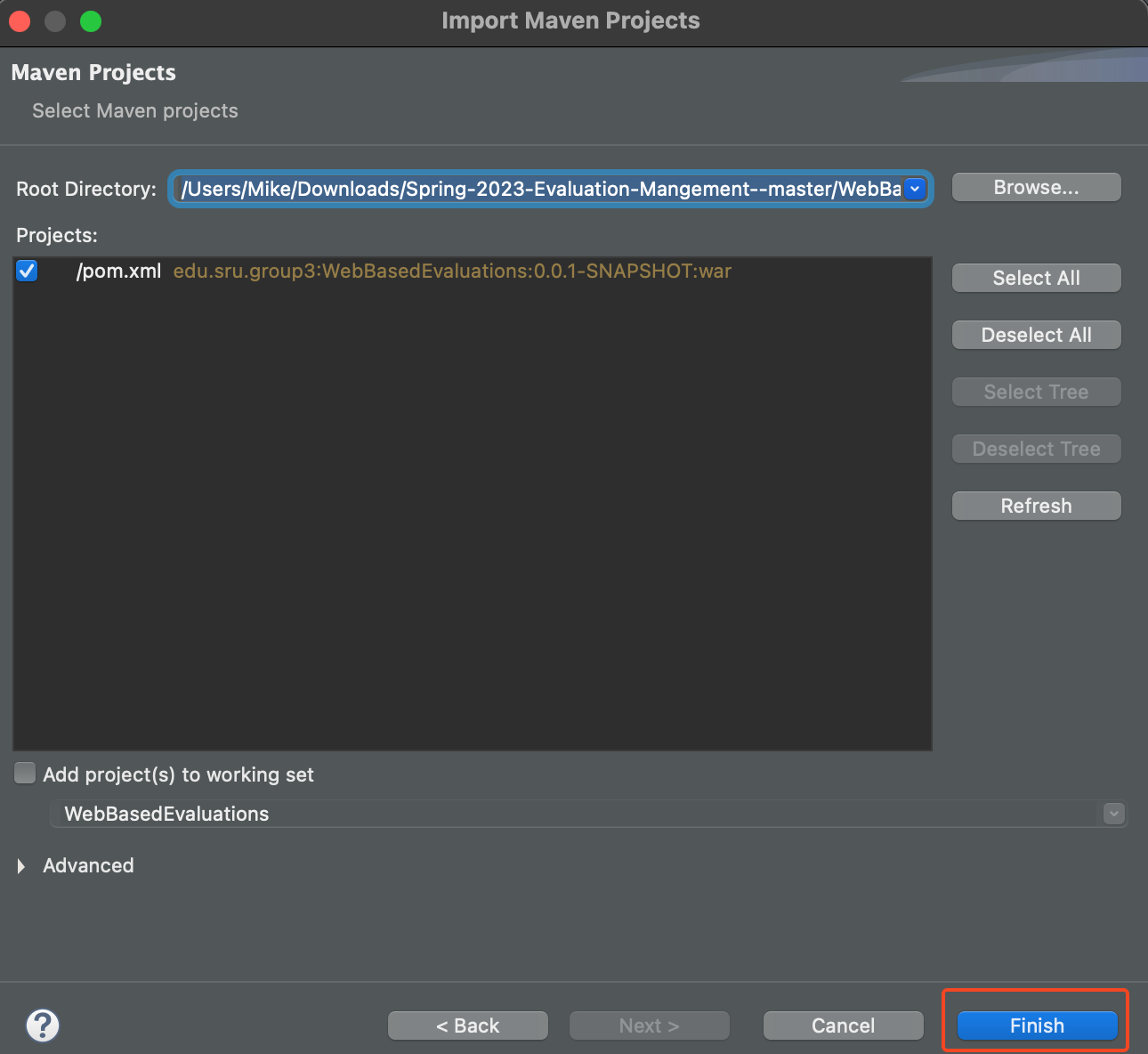
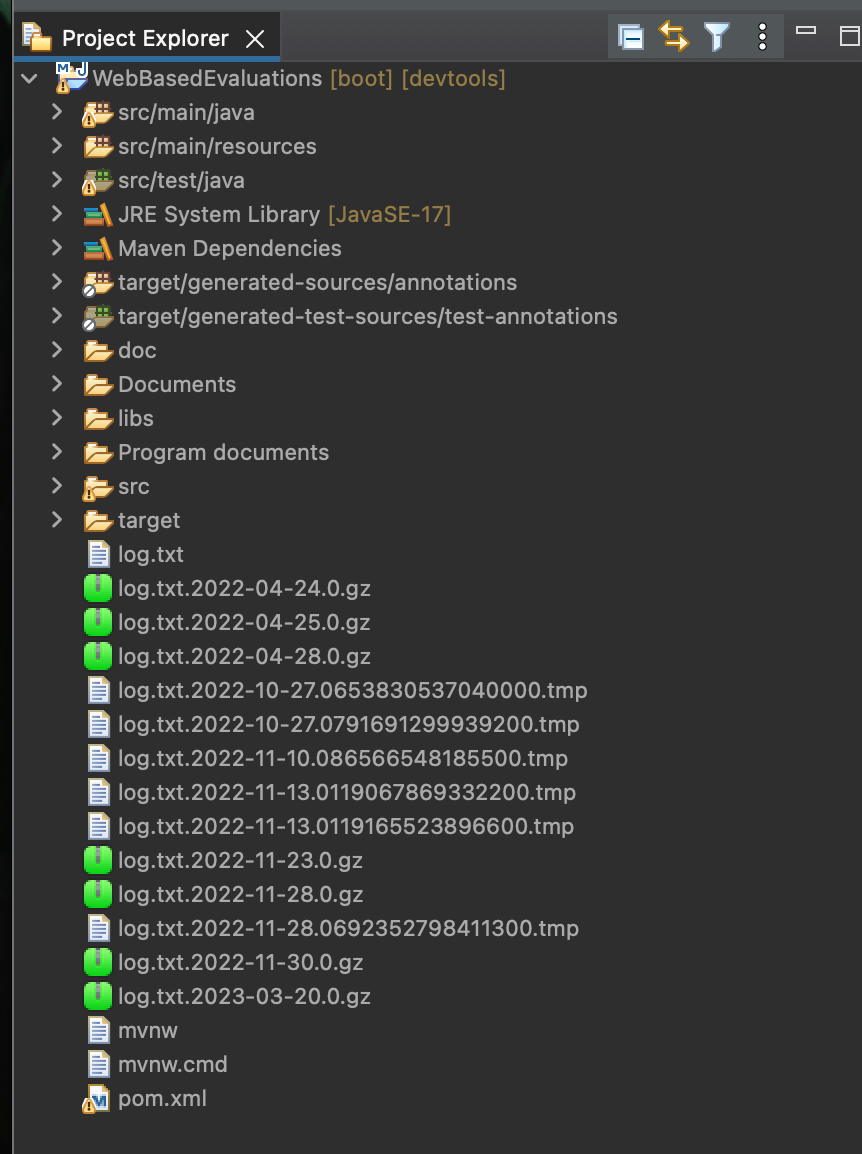
# 

# Installing the Program as a User

This section will outline how to install and run the program by just downloading the project’s source files from the GitHub repository. This section should NOT be used by programmers, as it does not illustrate the proper methods to set up a local repository for contributing to the code.

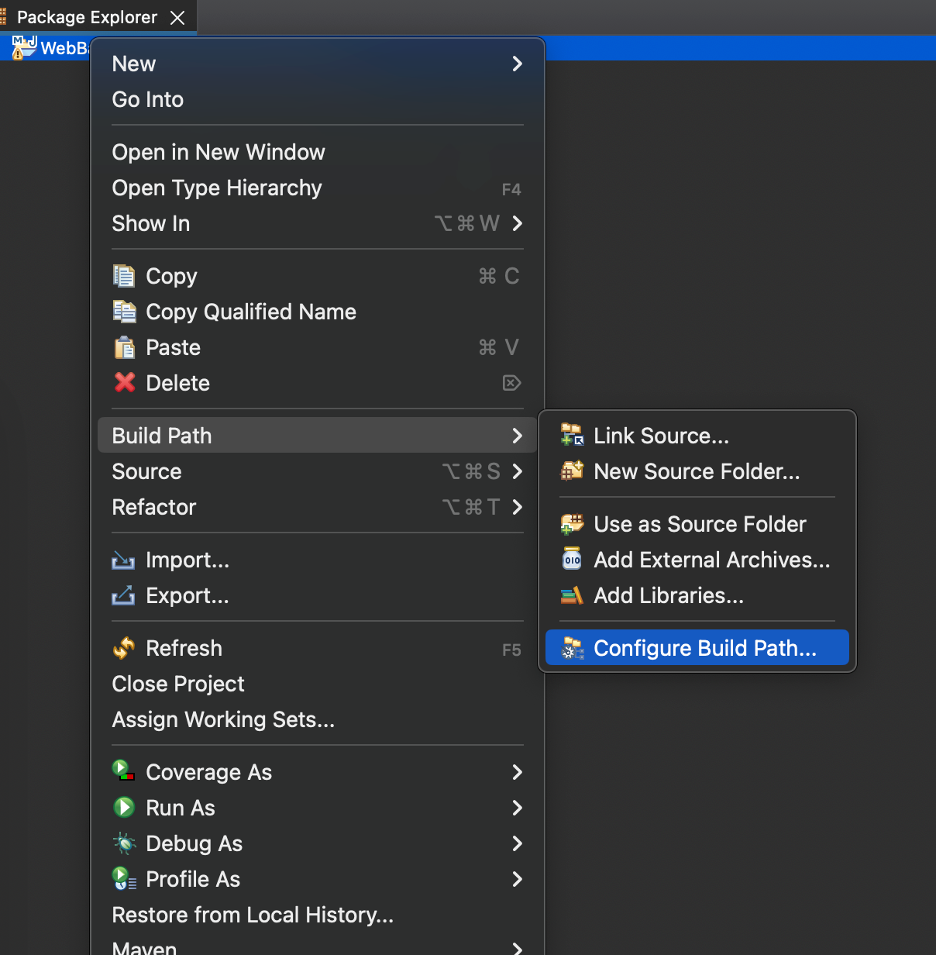
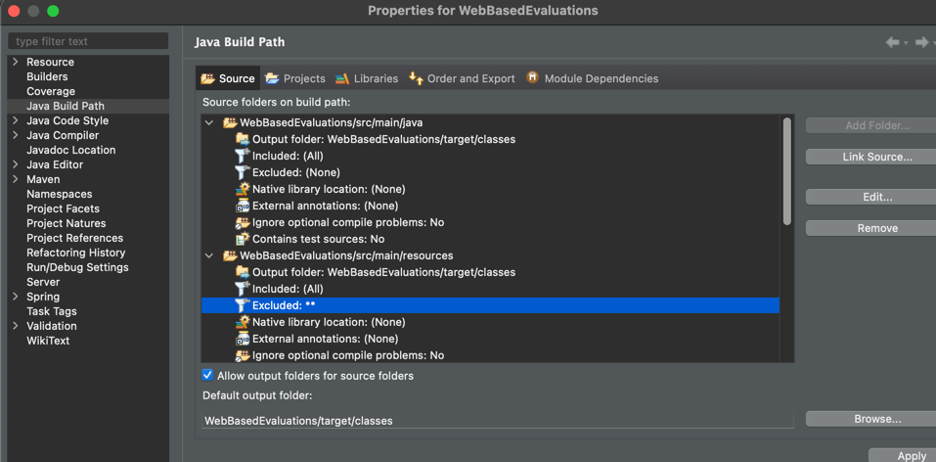
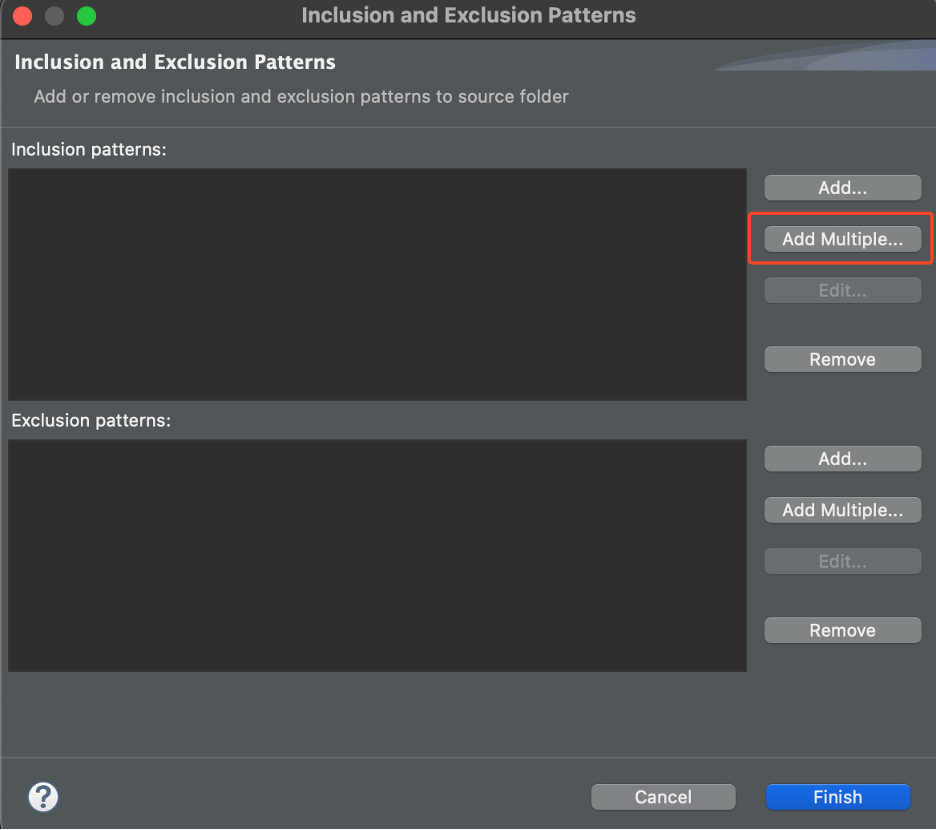
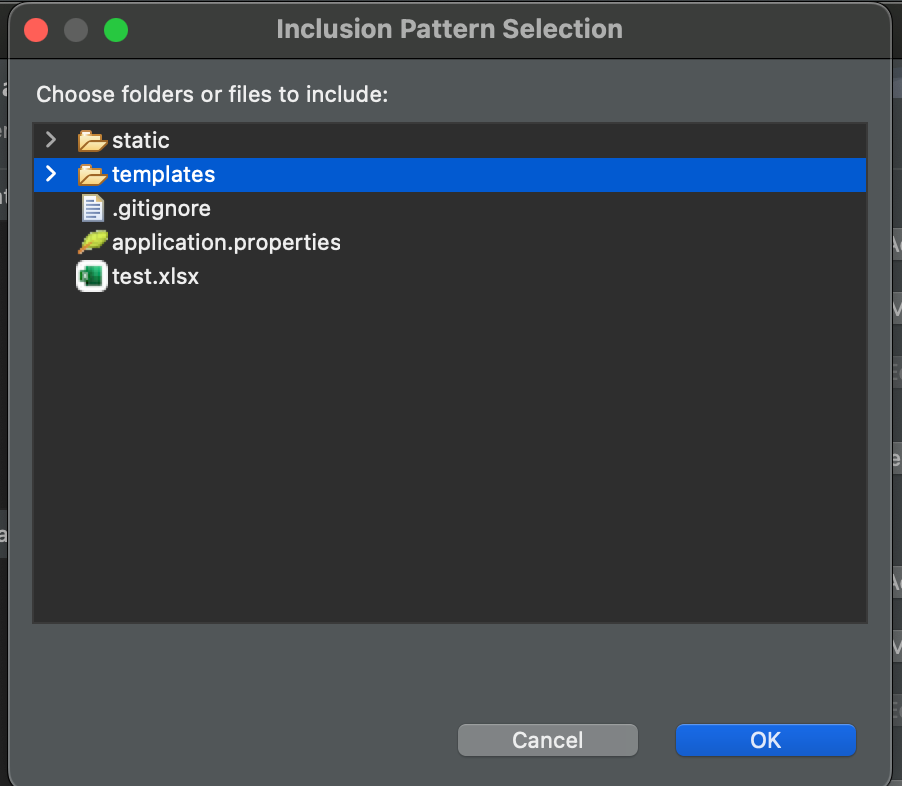
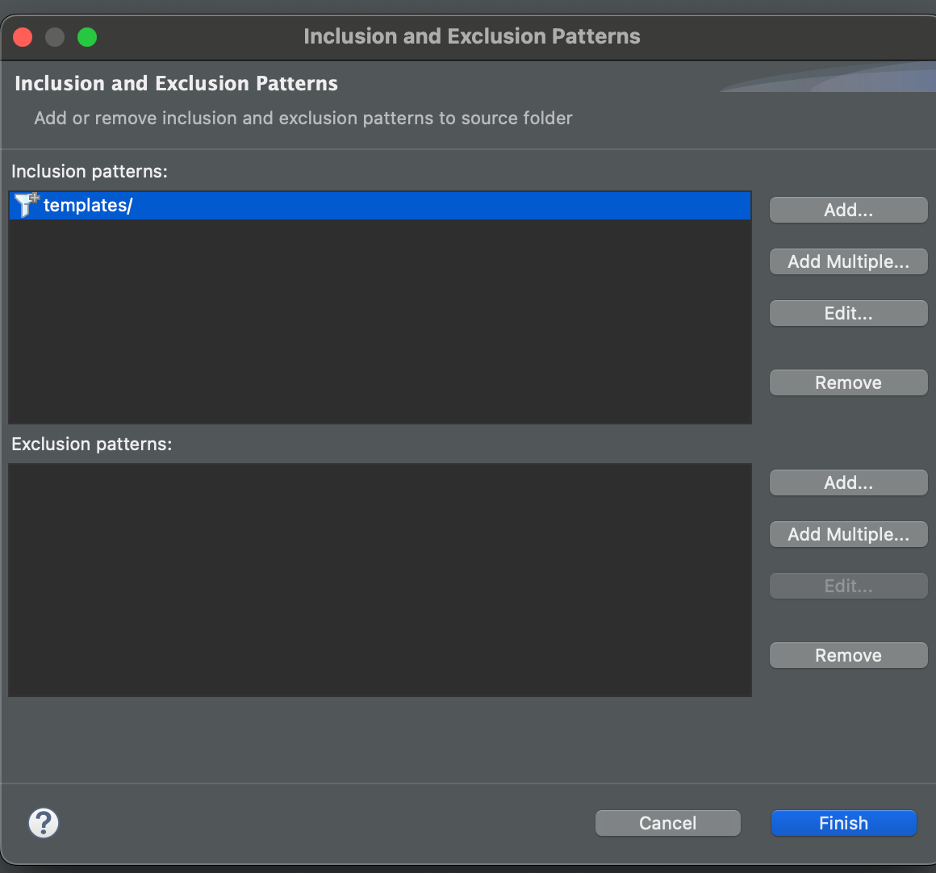
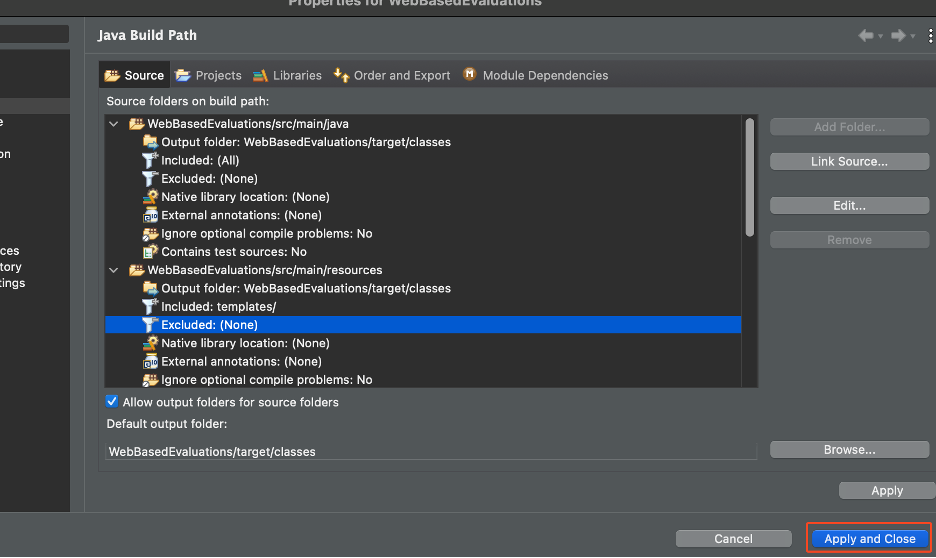
## Importing the Project from GitHub

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. In Eclipse, begin the import process by choosing **File > Import**  
     
   
3. Choose **Maven > Existing Maven Projects** and then hit **Next**  
     
   
4. Locate the directory of where you extracted the program by clicking on **Browse**. Once you find it, open the folder that is named **WebBasedEvaluations** for the import. Then click **Finish**.  
     
   
5. The project is now imported into the Eclipse workspace.  
     
   

## Configuring the Build Path

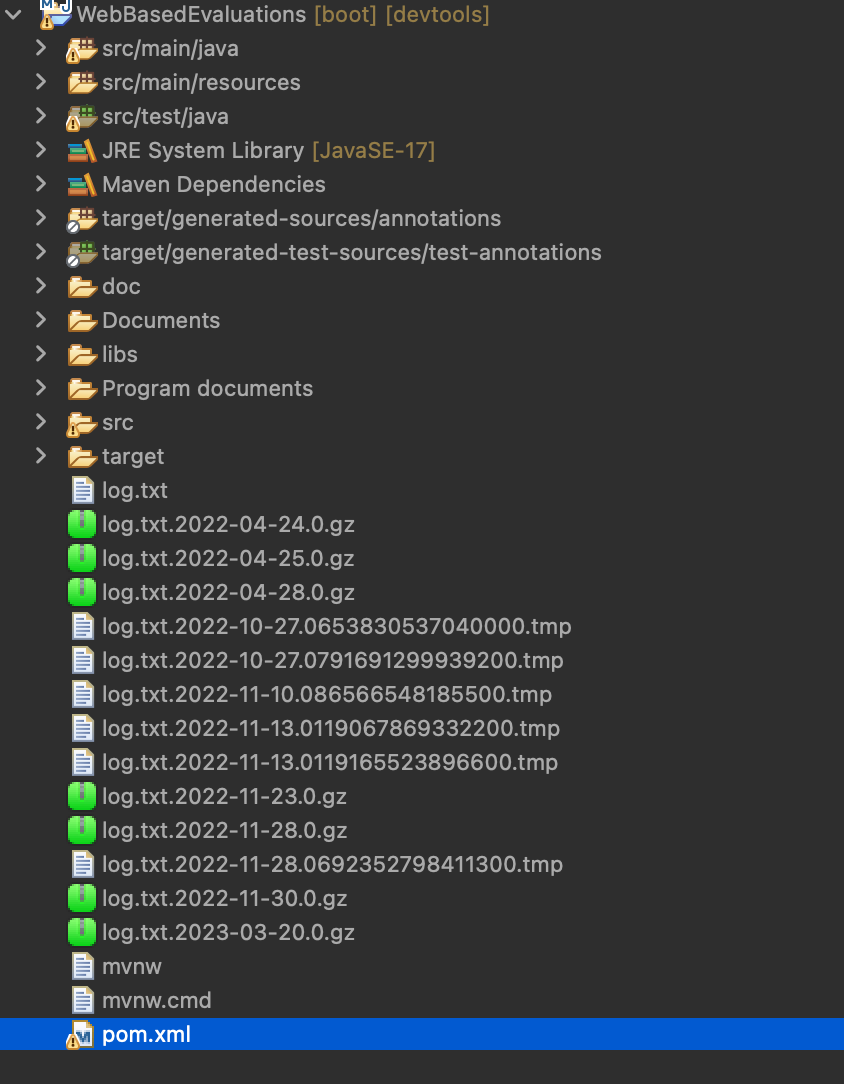
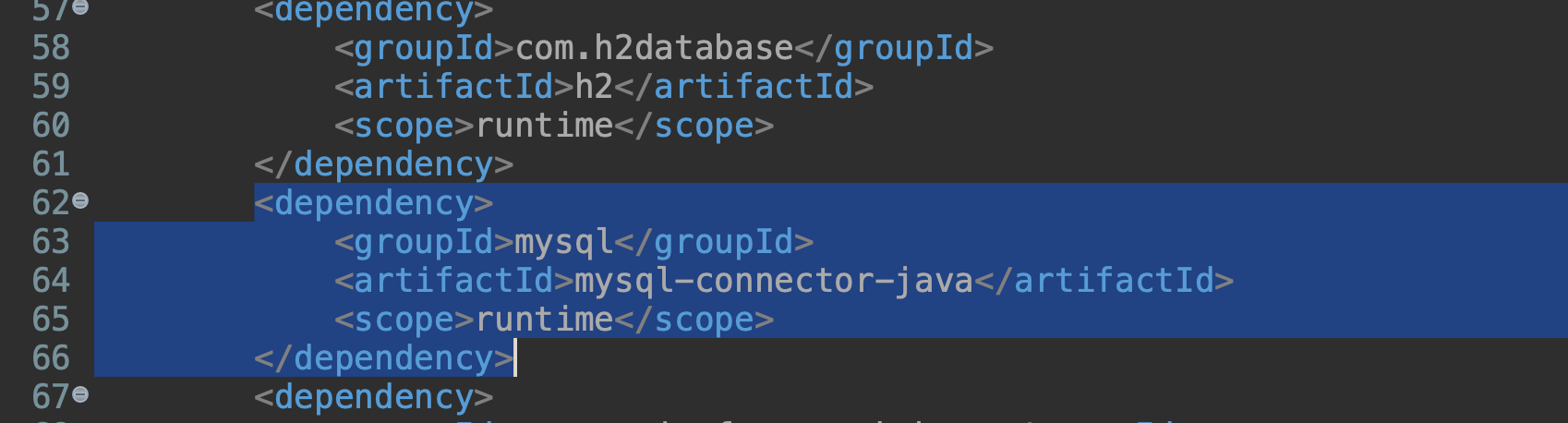
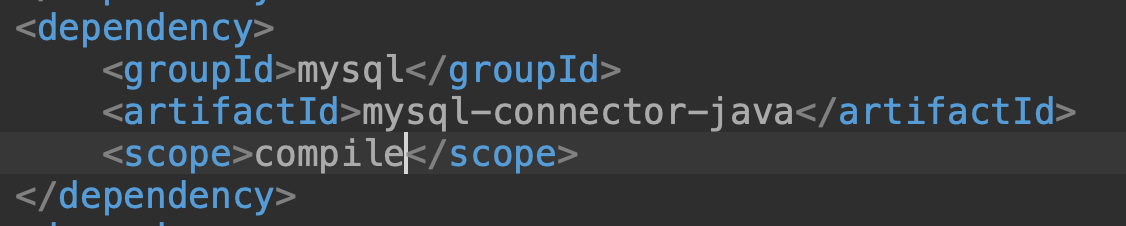
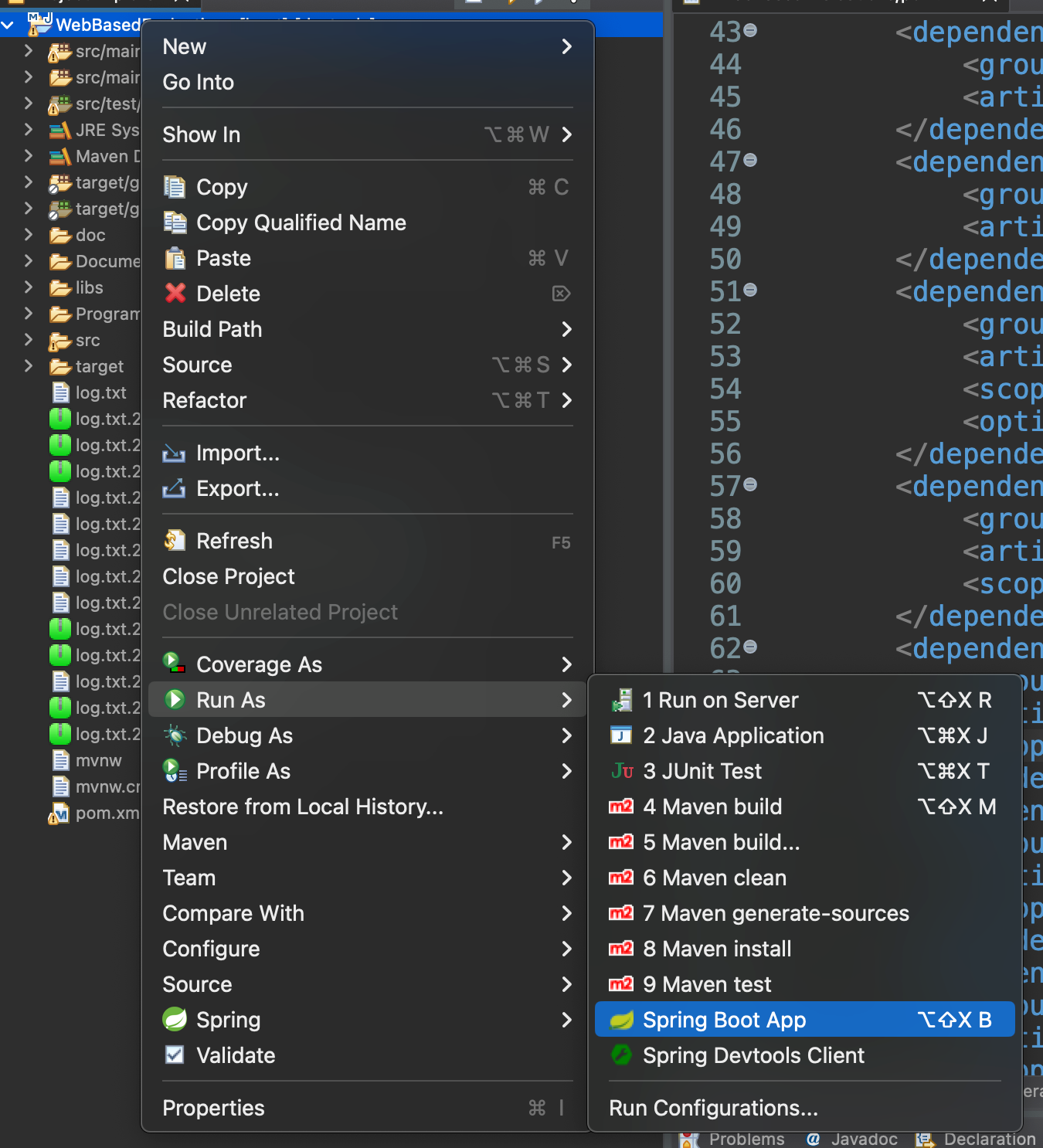
Eclipse is unable to find the folder that contains all of the HTML pages of the program by default. In order to fix this, the build path needs to be configured using the following steps:

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 WebBasedEvaluations/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, and then click “OK”  
     
   
6. Click “Finish”   
     
   
7. Click “Apply and Close”. The build path is now configured for running the program.  
     
   

## 

## Configuring the MySQL Database

At this point, the program will run, but the database will not appear in MySQL workbench. This also means that data will not be saved when running the program in data persistence mode. To fix this, a quick adjustment need to be made to the **pom.xml** file.

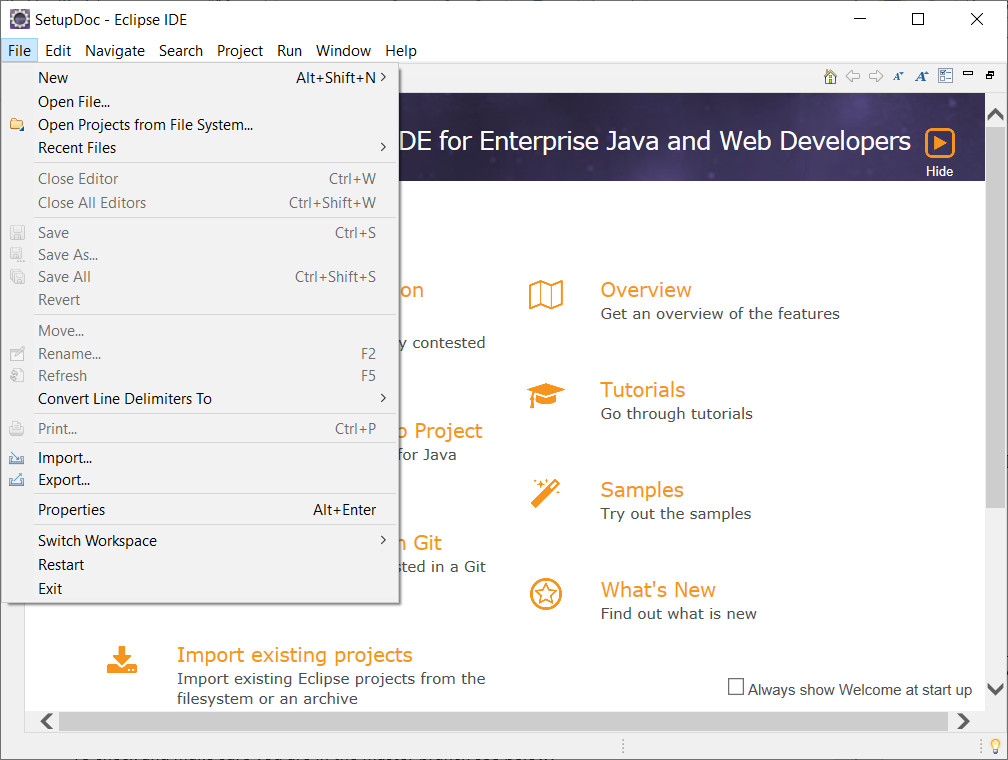
1. In the project directory in Eclipse, locate the **pom.xml** file and open it. It should be at the bottom.  
     
   
2. Locate the dependency that is created for MySQL. It should be near line 63.  
     
   
3. Change the value for **scope** to be “compile” instead of “runtime.” This will ensure that the database is created correctly upon running the program.   
     
   
4. The program can now be ran in Eclipse by right clicking the main project folder and then selecting **Run As > Spring Boot App**.  
     
     
   

# 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 Start

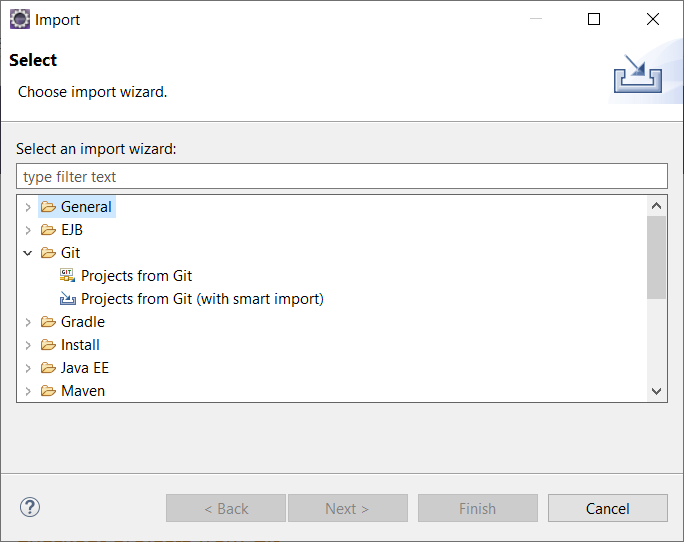
1. After installing the program requirements, launch the Eclipse IDE and select/create a location for your workspace to be stored with the **Browse...** option and then select **Launch** when the workspace has been set.
2. Navigate to the top tool bar to the **File** option and select **Import…** as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



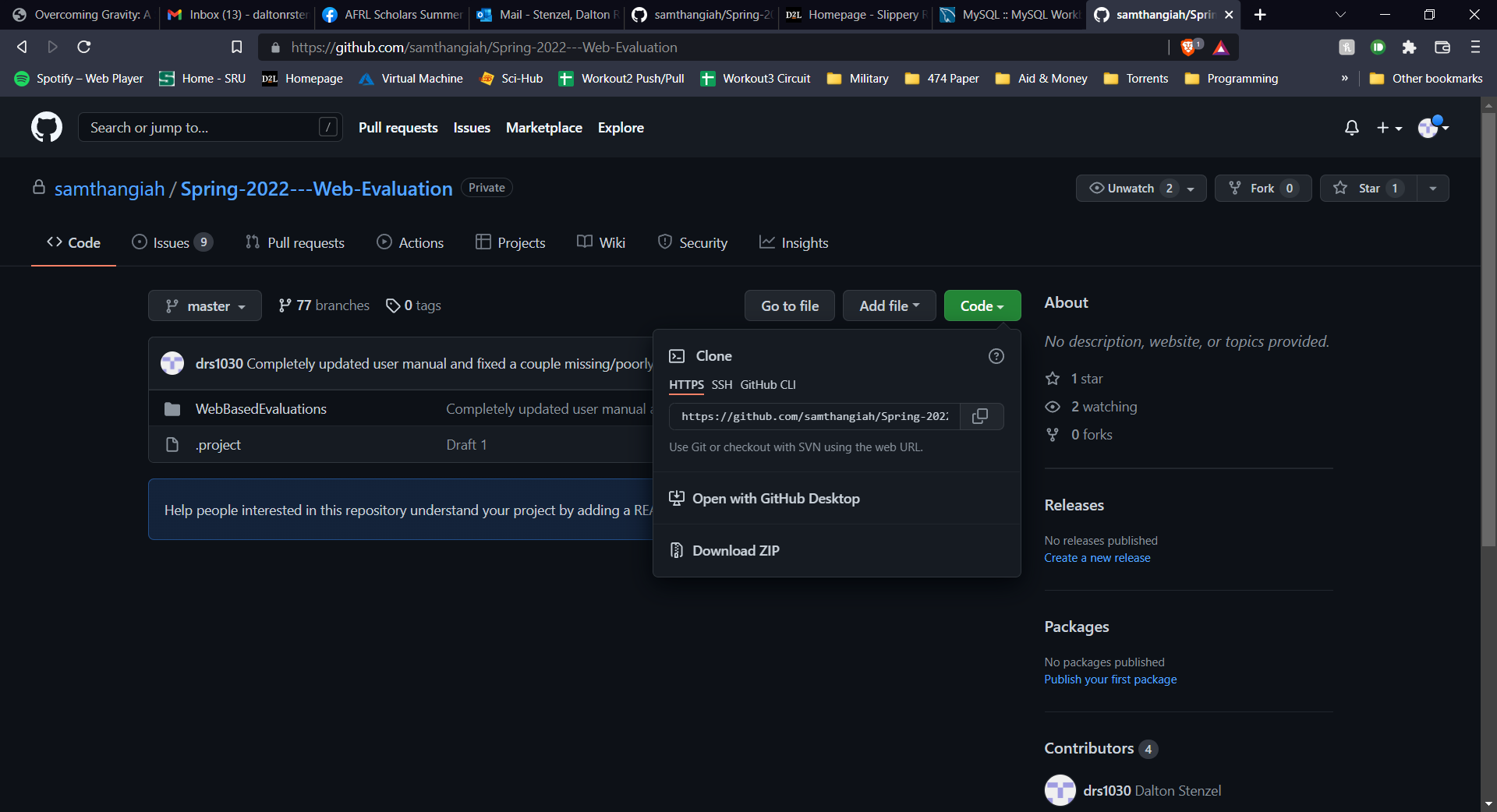
## 

## Importing the Project with Smart Import

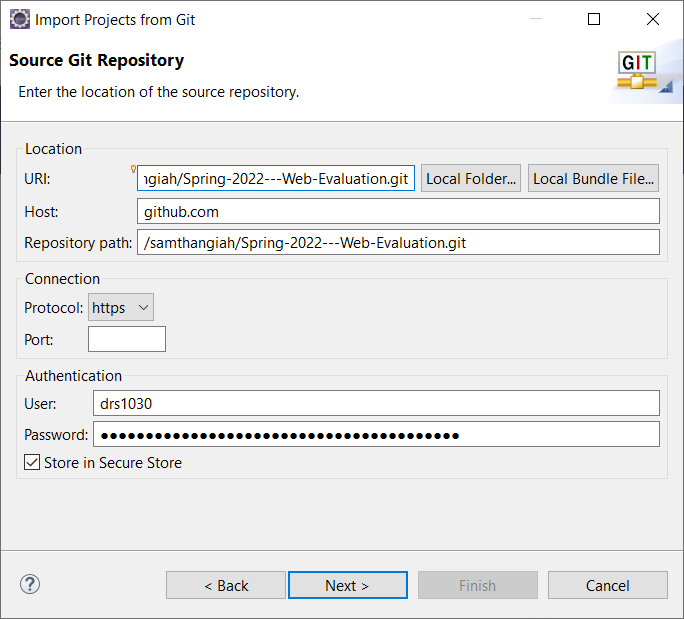
1. Once you’re in the import window, click the **Git** folder arrow to expand what’s inside and select **Projects from Git (with smart import)** as seen below.



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.



1. 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.



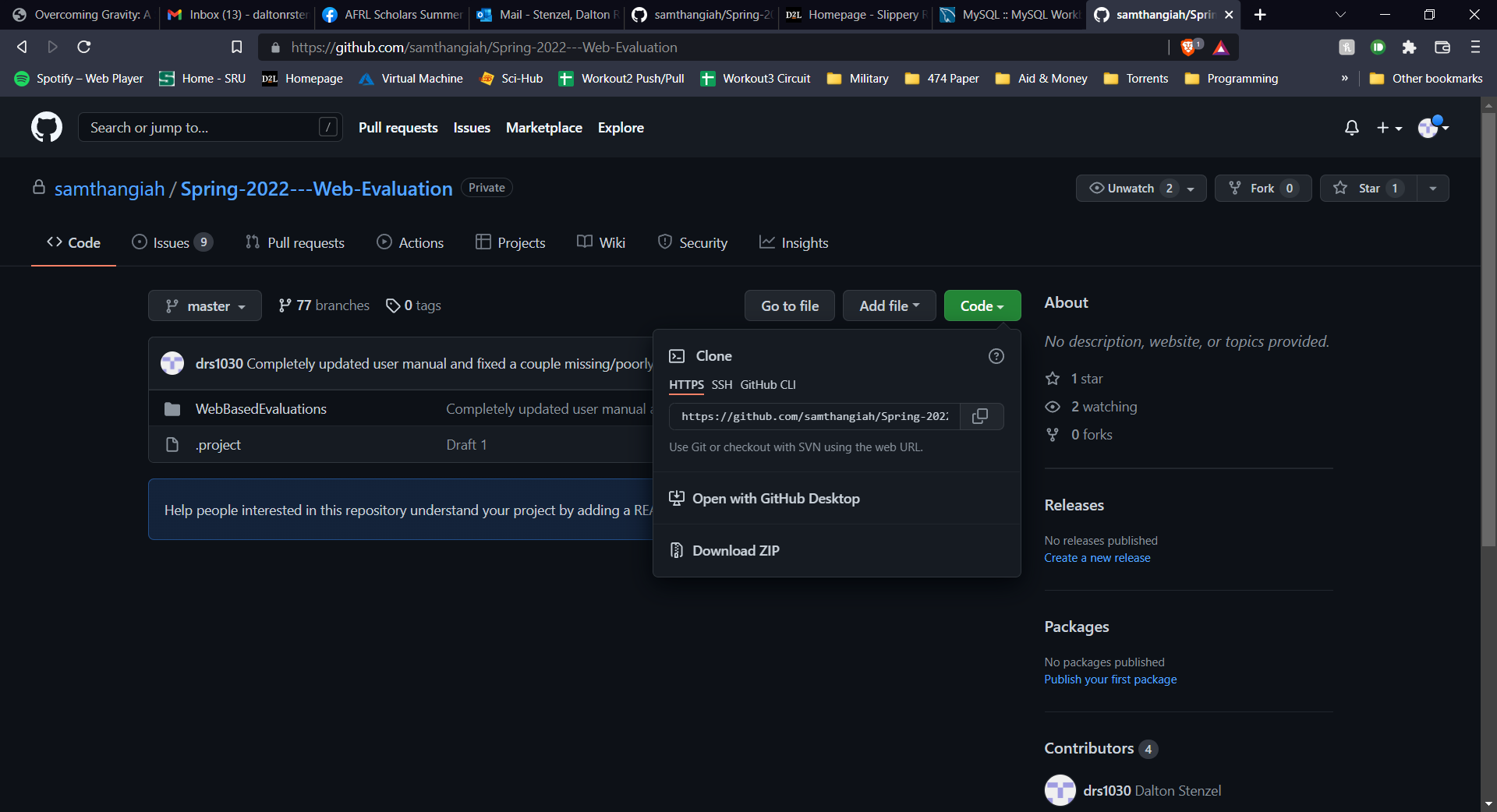
1. Select **Next** and on the new page, search for the branch **master** and select **Next** once more. You’ll be taken to a page where you must select the directory of the destination of the GitHub files. Either leave it as the default location or user **Browse** to manually choose where to place the 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 to the **Root directory** box in order to choose the unzipped folder that you previously downloaded from the GitHub. Once that the project folder has been selected, press **Finish**

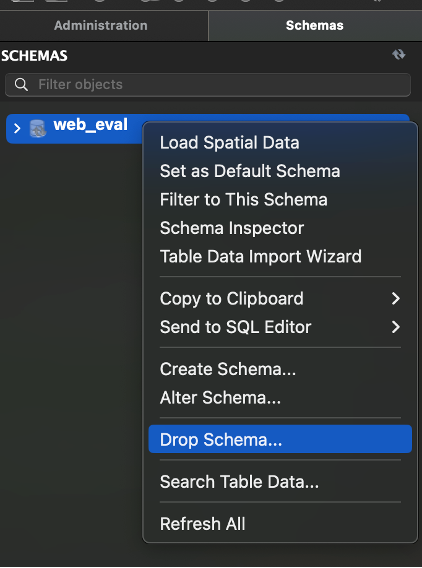
## 

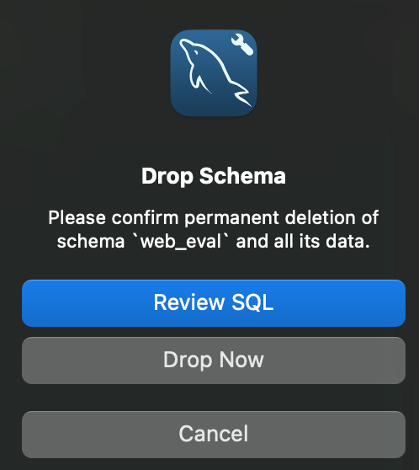
## Running the Program with Data Persistence

By default, the program does not save any of the data uploaded or entered. This can be changed to preserve data after program termination using these outlined steps.

### Drop Old SQL Schema

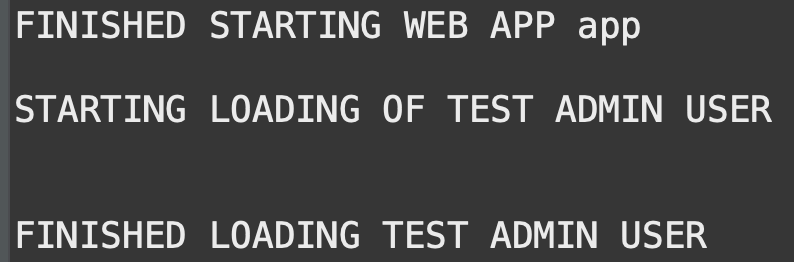
Before using Data Persistence mode for the first time, you must drop any existing schemas in MySQL Workbench. Do this by right clicking on the **web\_eval** schema, and selecting **Drop Schema**. Then in the dialog click on **Drop Now**.



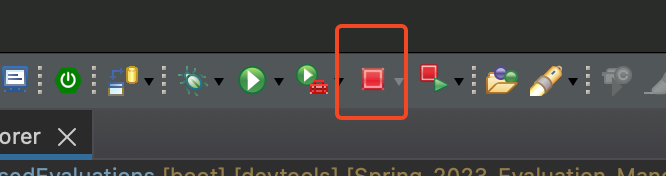


### Running the Program Once Using the Create Schema

After ensuring the old SQL schema does not exist, run the program ONCE to ensure that the tables are created. This can be verified by the logins created, which are logged in the console.



At this point, you can terminate the program in Eclipse and proceed.



### Changes to Code

Now that the tables have been created, some further changes will be necessary in order to make the database save new data.

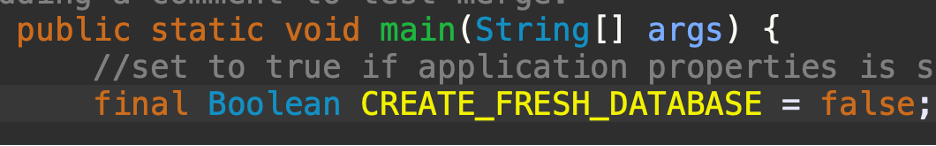
### 

#### WebBasedEvaluationsApplication.java

This file can be located in the following path:

**src/main/java/edu/sru/WebBasedEvaluations/WebBasedEvaluationsApplication.java**

Inside the main method, you will need to set the Boolean value CREATE\_FRESH\_DATABASE to be equal to **false**.

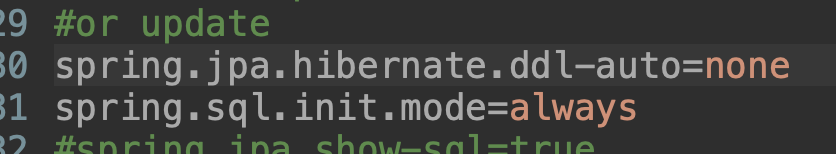


#### application.properties

This file can be located in the following path:

**src/main/resources/application.properties**

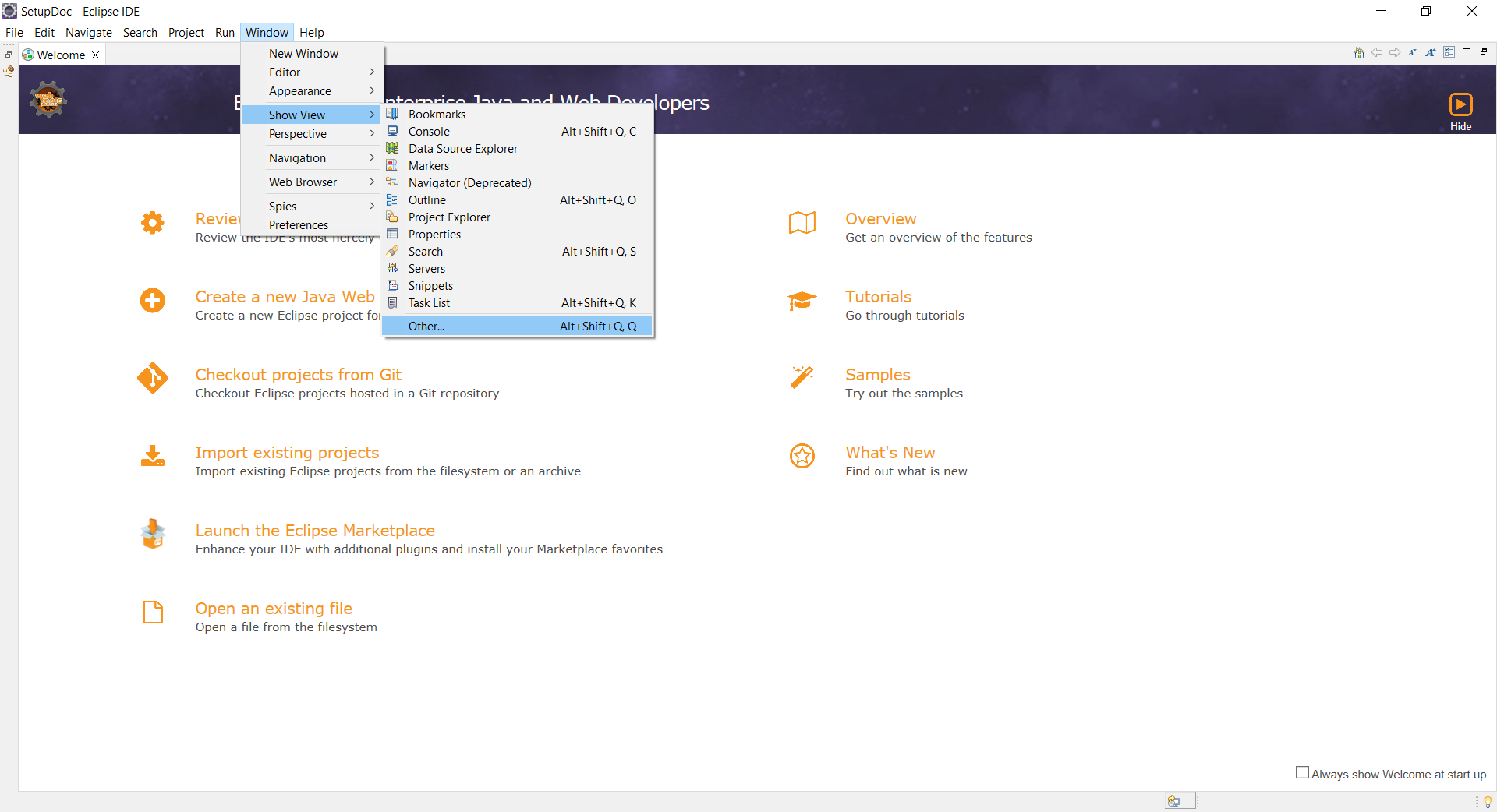
Locate the line that says “spring.jpa.hibernate.ddl-auto” and set it to be equal to **none**.



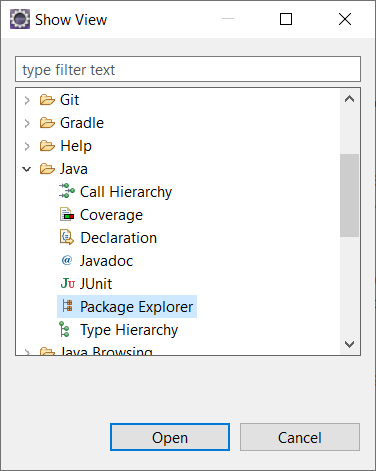
Now, when you upload files while running the program, they will stay in the database after the program is terminated.

## 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 tool bar and press **Window** then **Show View** then **Other** in order to bring up a window.



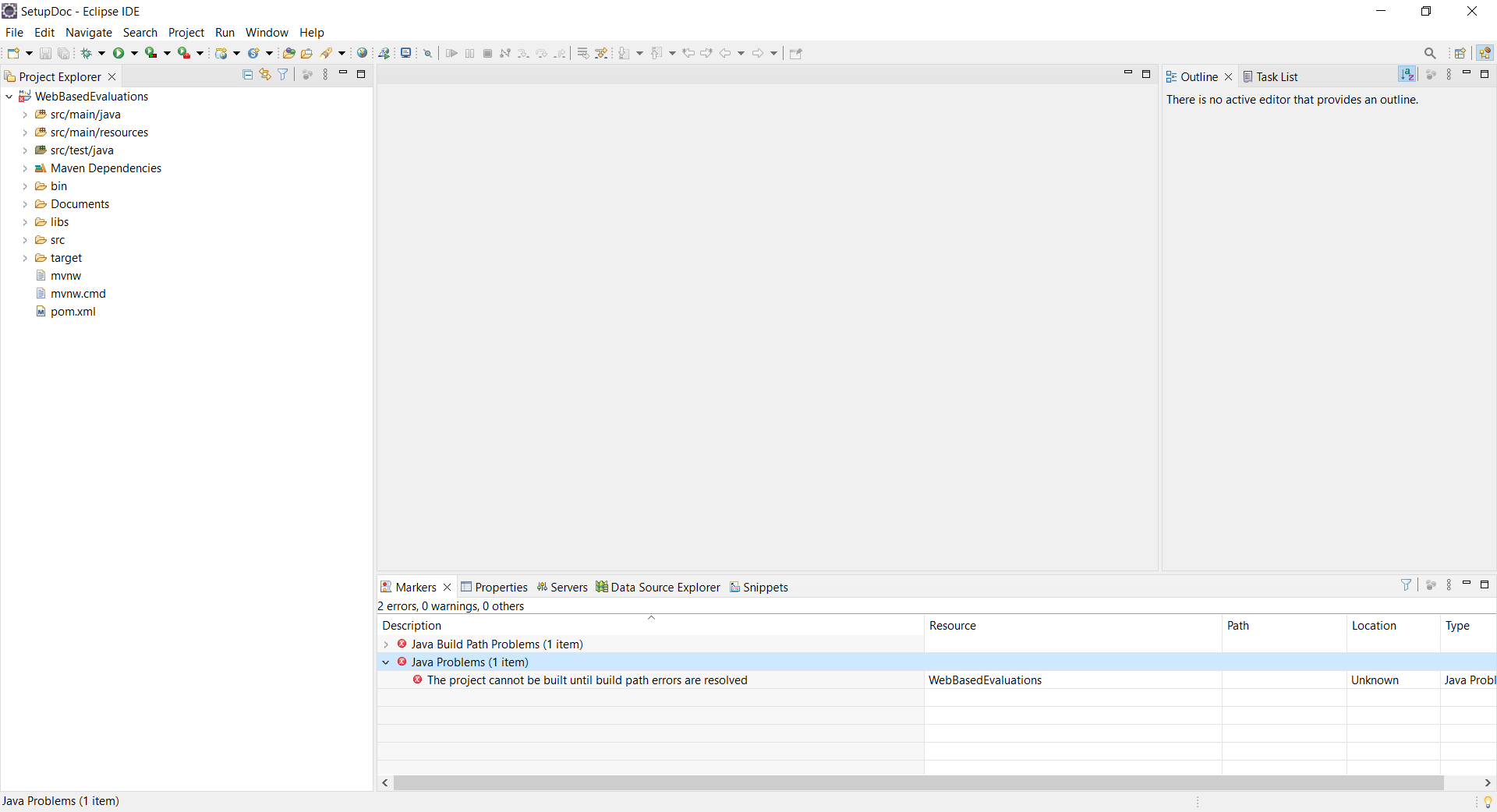
1. A **Show View** window will display, and you must search for a folder named **Java** and expand it with the arrow right next to it to display the folder’s contents. Select **Package Explorer** and press the **Open** button.



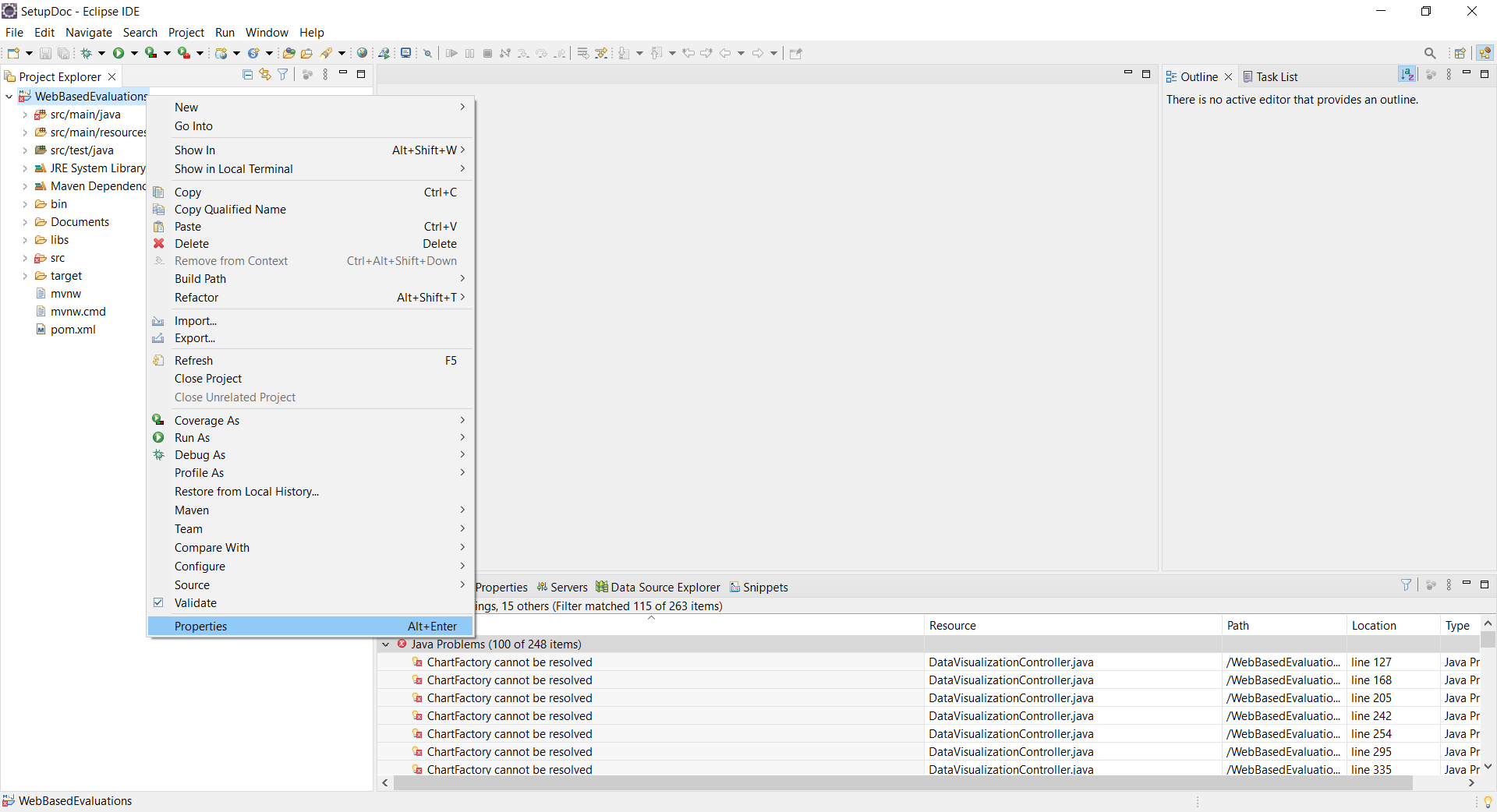
## 

## Resolving Build Errors

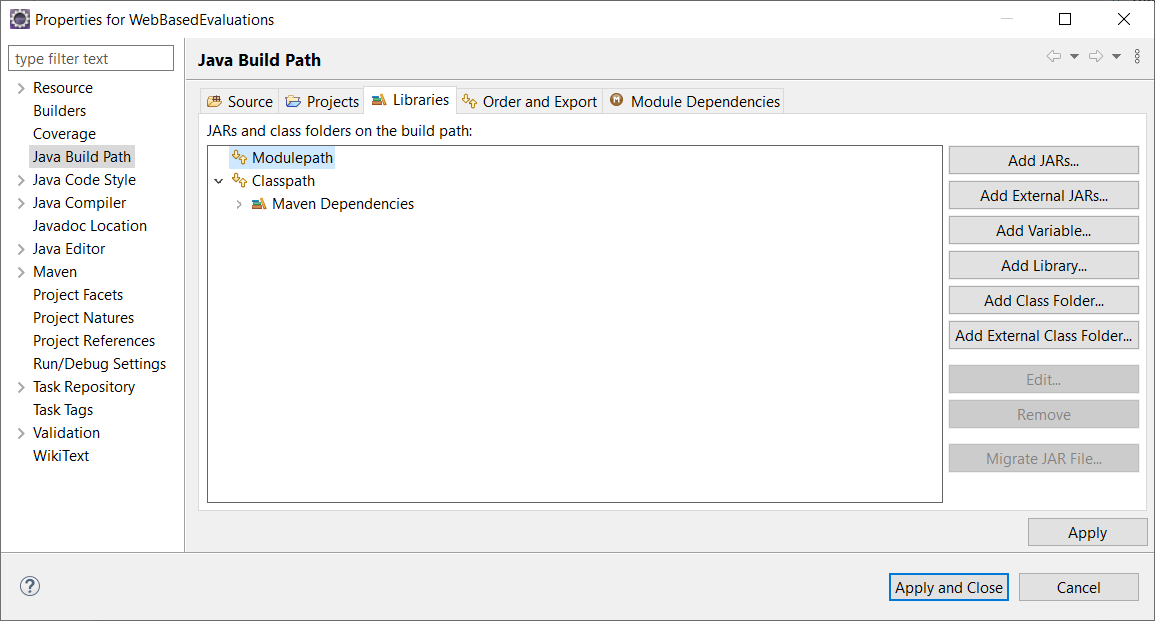
If you come across errors in the **Markers** tab located near the bottom of the program, that’s like due to the fact that Java SE 17 wasn’t selected as the default version.



1. Right click on the project in either the **Project Explorer** or **Package Explorer** and select **Properties.**



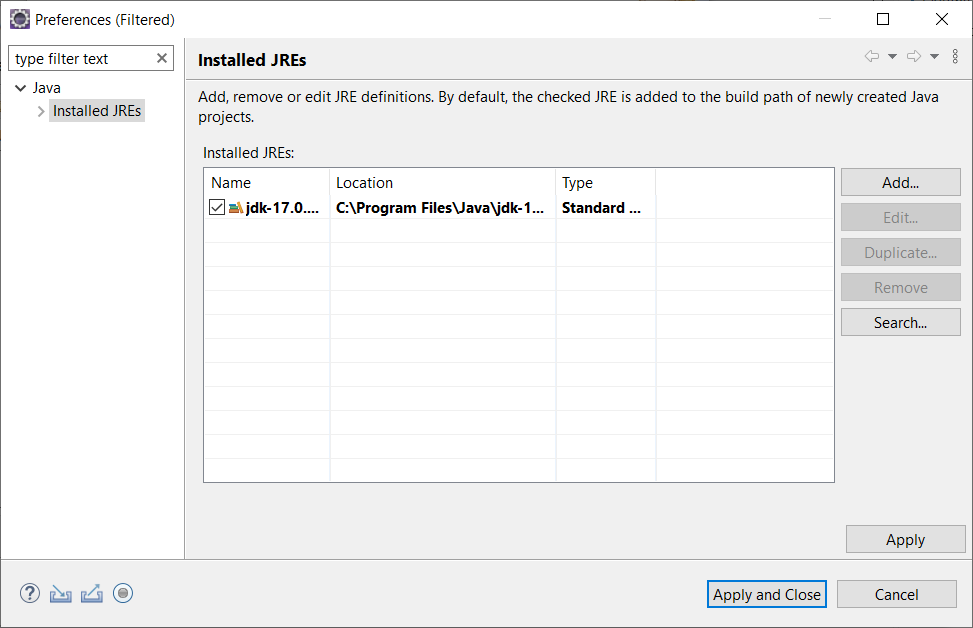
1. Then on the right side, select **Java Build Path** and from within that newly opened area press **Libraries.**



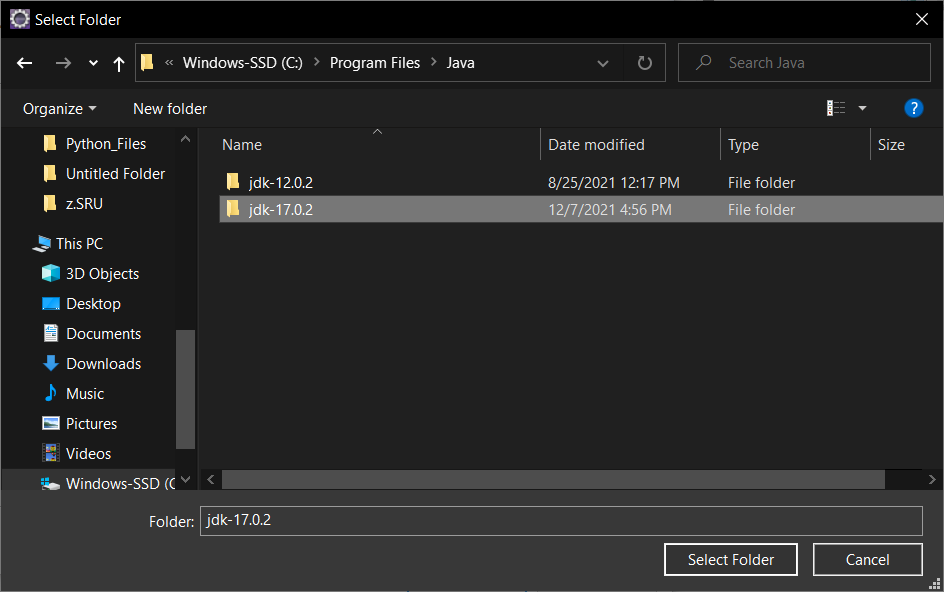
1. From with the **Libraries** tab, select **Modulepath** and press **Add Library…** located to the left to bring up a new window.
2. From the **Add Library** window, select **JRE System Library** and press **Next.**



1. Once you’re on the **JRE System Library** part, select **Alternate JRE:** and press the **Installed JREs…** button to the right of it. That will bring up a new window.



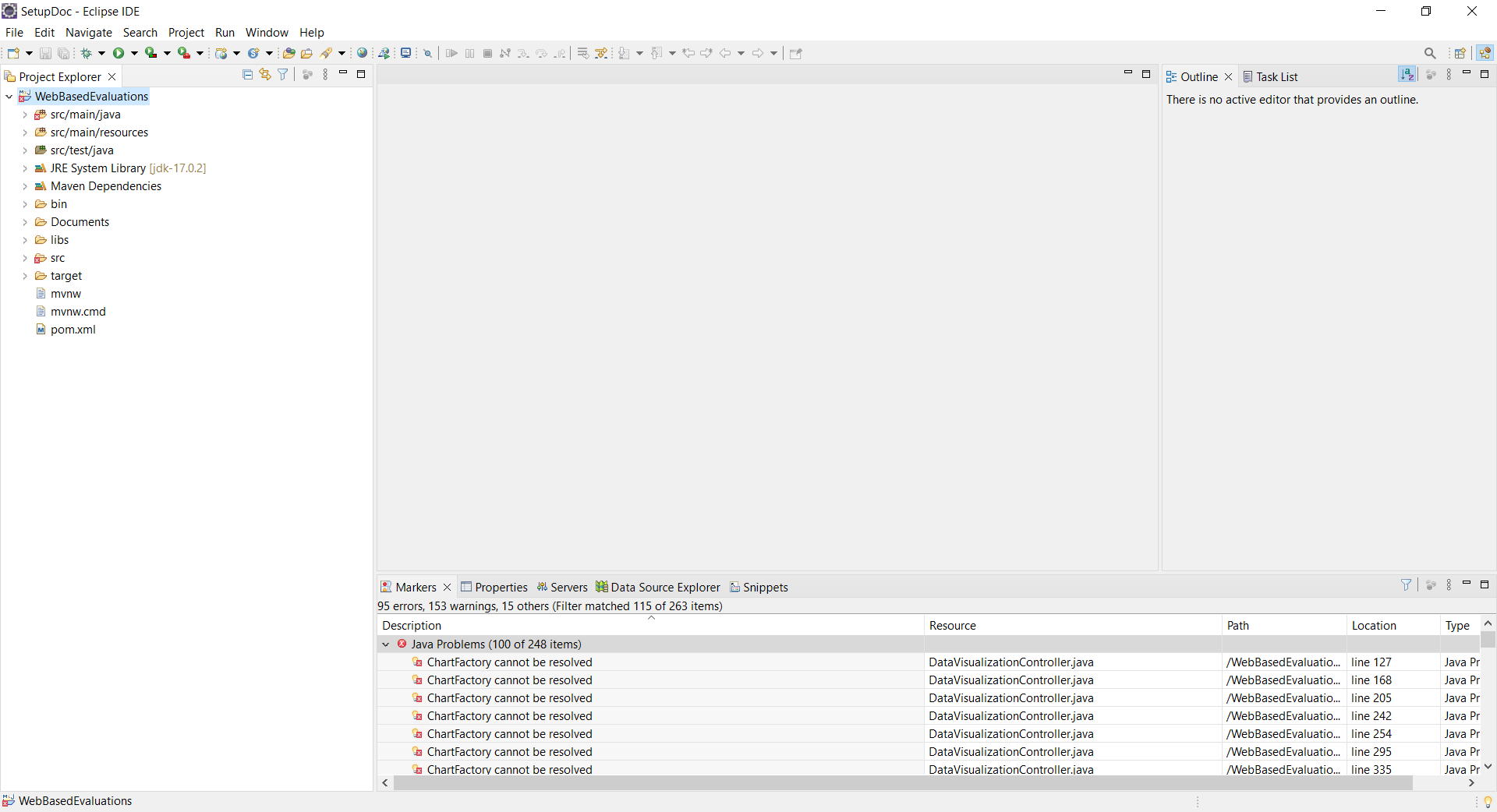
1. From this new window, press the **Add…** button located on the right. A new window will pop up, named **Add JRE** where you will select **Standard VM** and press N**ext.**
2. After pressing the next button, a you’ll see a **JRE home** box with a **Directory…** button to the right of it, press the **Directory…** button and select the folder that contains jdk-17. The location of it can vary, but if the jdk is installed from an installer, then it will like be in C:\Program Files\Java.



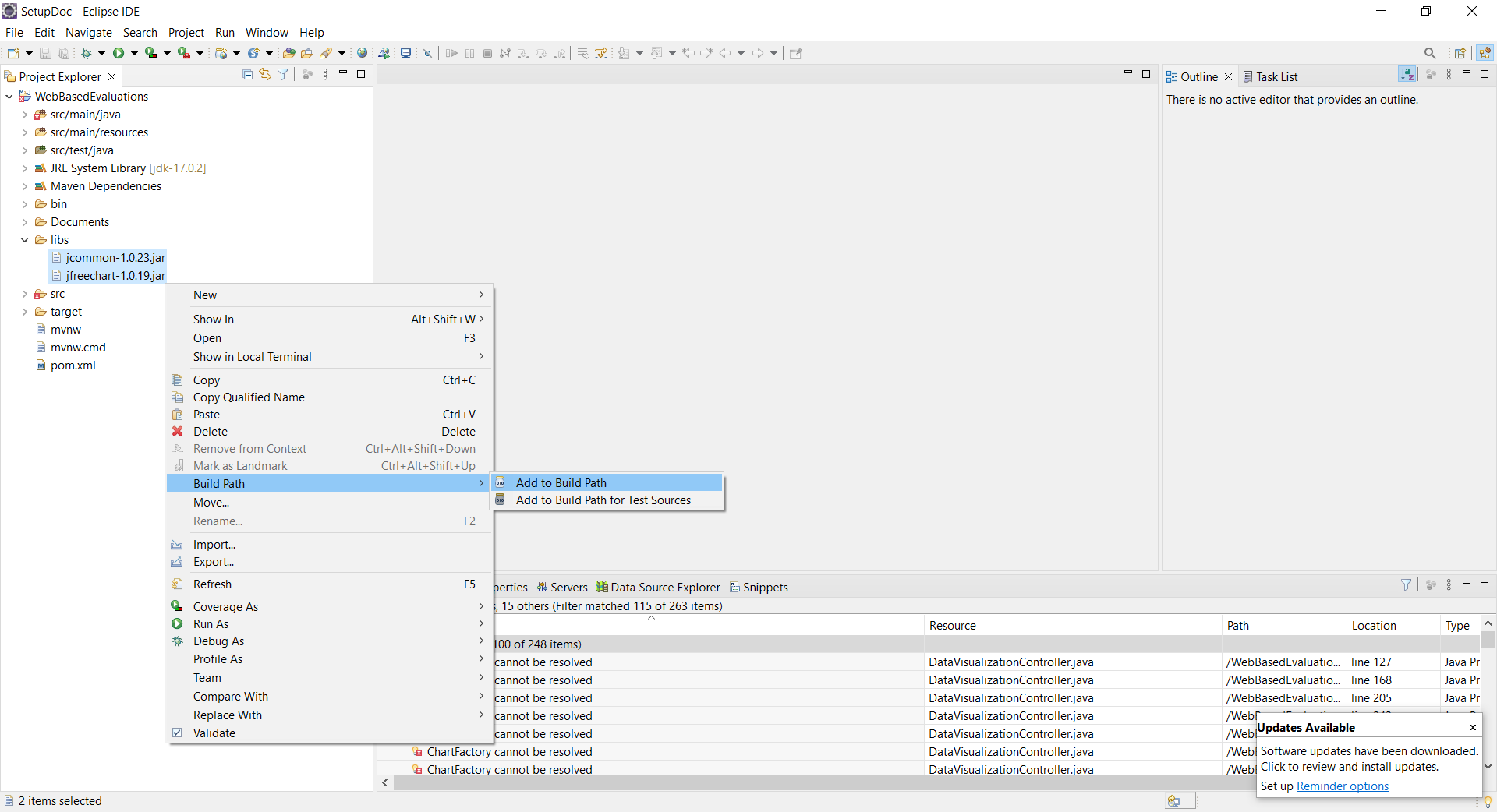
1. After selecting the folder with the **Select Folder** button, you should be able to close out of everything with a series of **Finish** and **Apply and close** buttons.

## Missing Libraries Problems

You’ll likely see errors relating to “ChartFactory cannot be resolved” in the **Markers** window located near the bottom which come from missing links to additional libraries required.

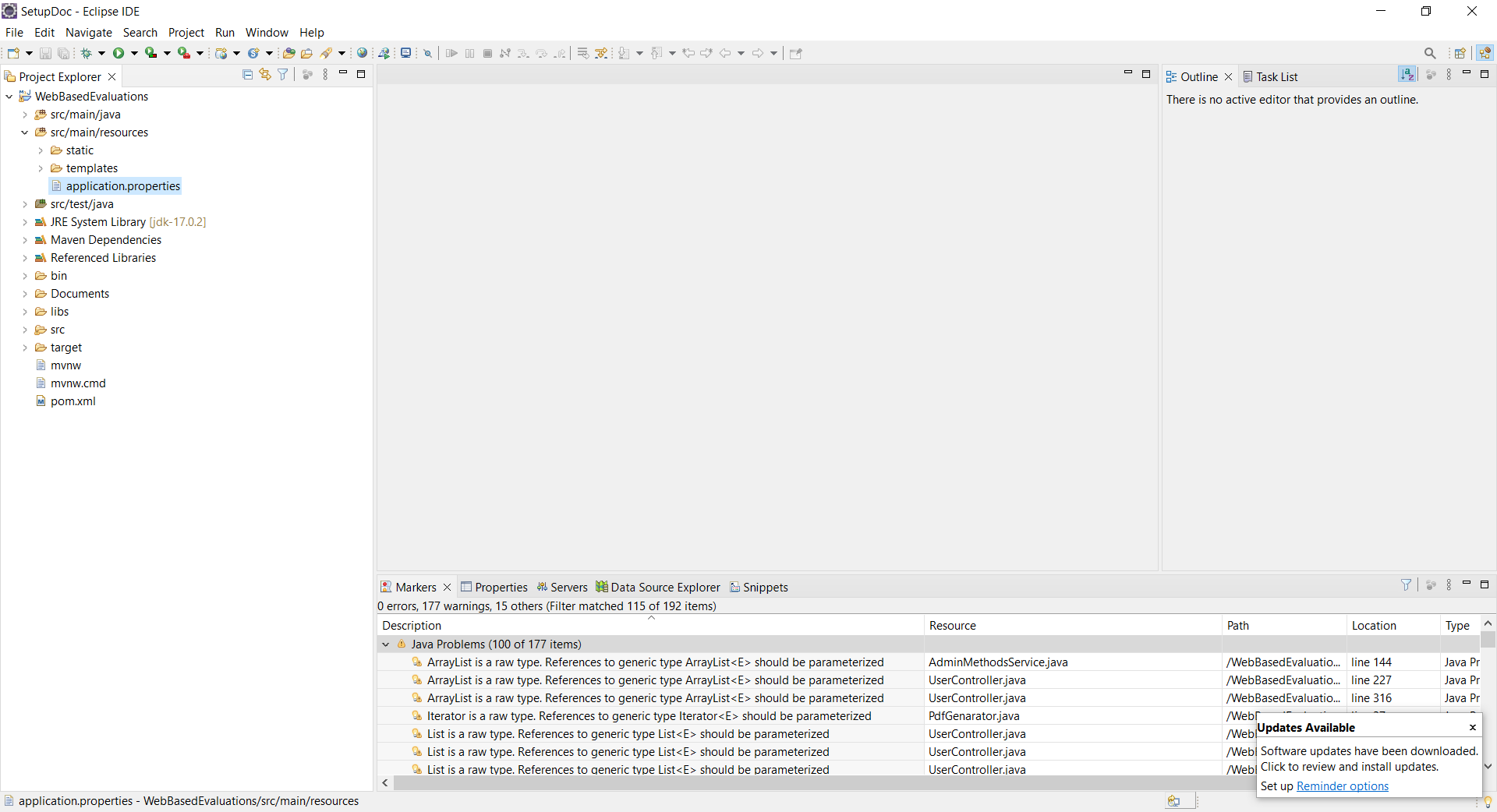


1. Under the Project folder, find the **libs** folder and select both **jcommon-1.0.23.jar** and **jfreechart-1.0.19.jar** and right click them. From there, go to **Build Path** and then **Add to Build Path** to resolve the issues.



## 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. Upon opening the file, which can be done by double clicking it the following lines must be changed:

Text

Description automatically generated

1. Set the spring.datasource.username and spring.datasource.password to your respected MySQL server’s username and password and save the changes by pressing “ctrl” and “s” at the same time or holding “ctrl” down and pressing “s”.
2. Set the spring.jpa.hibernate.ddl-auto setting to = create, =create-drop or =update

## Login Information

Use the following default usernames and passwords to traverse through the program:

The first has access to the Thangiah Manufacturing LLC company and the other to Test Company 2

The permissions can be shown as only the Thangiah LLC user can upload company1 users or groups and the Test co2 user can upload company2 users/groups. Both should be working.

[admin@gmail.com](mailto:admin@gmail.com) test

[admin2@gmail.com](mailto:admin2@gmail.com) test

The database will only contain a single super-superuser. The user has the email [admin@gmail.com](mailto:admin@gmail.com), and admin has “ADMIN” privileges. The “ADMIN” privilege is one that grants the ability to add all sorts of users with a variation of five types of roles to choose from: “ADMIN”, “EVAL\_ADMIN”, “EVALUATOR”, “EVALUATOR\_EVEL”, and “USER”. The five privilege types have access to different things as noted in the User Manual. In order to take full advantage of the program, the administrative user will need to provide other users, beginning with an admin for each department. Such details are also covered in the user manual.

If this does not work make sure that you drop the old schema and have the spring.jpa.hibernate.ddl-auto set to update as sometimes the automatic dropping of tables with create/create-drop will not work correctly after changes have been made to the tables in the java code.

If several “errors” pop up says XX tables doesn’t exist they can be ignored as this happens when create-drop is enabled, the “errors” do not have any effect.

## Continued Maintenance

Ensure files are uploaded to the system in this order:

Company > Roles > Users