**Requirements**:

* MySQL Workbench (<https://www.mysql.com/products/workbench/>)
* Java SE 17\* (<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>)
* Eclipse IDE for Enterprise Java and Web Developers\*\* (https://www.eclipse.org/downloads/packages/release/2021-12/r/eclipse-ide-enterprise-java-and-web-developers)
* GitHub Account

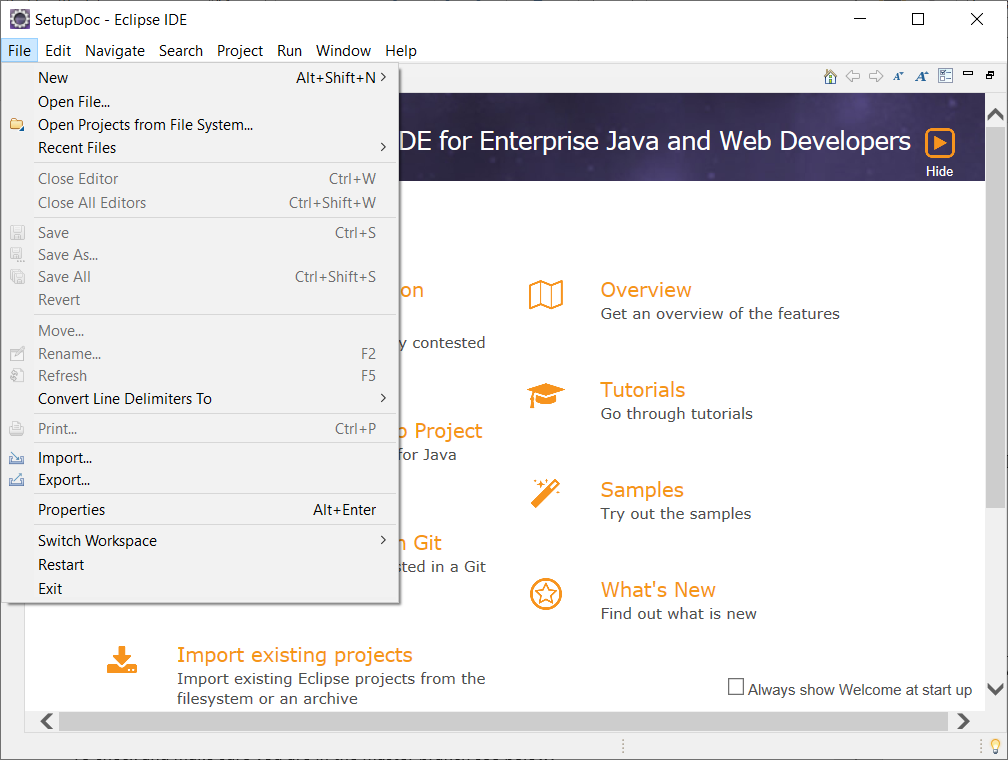
\*note version 17.0.2 was used in this example, but other versions of 17 might work

\*\*note version 2021-12 was used in this example but other versions around the same version or newer might work

**Post Installation of Requirements**

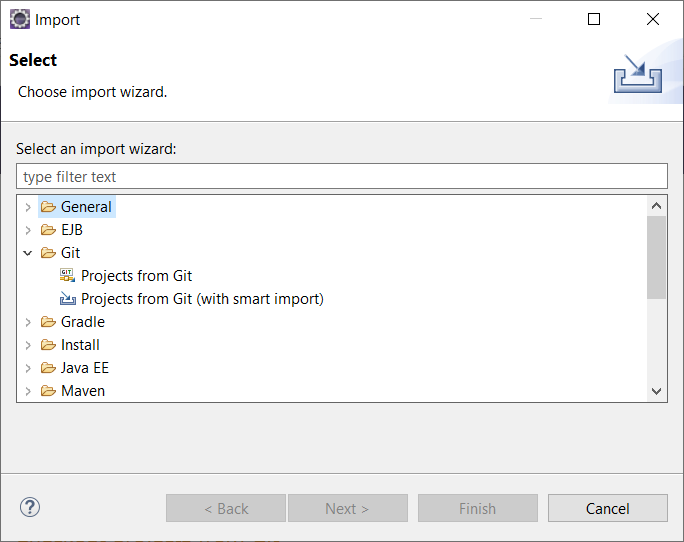
**Importing the Project Start**

* 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.
* Navigate to the top tool bar to the **File** option and select **Import…** as shown below.

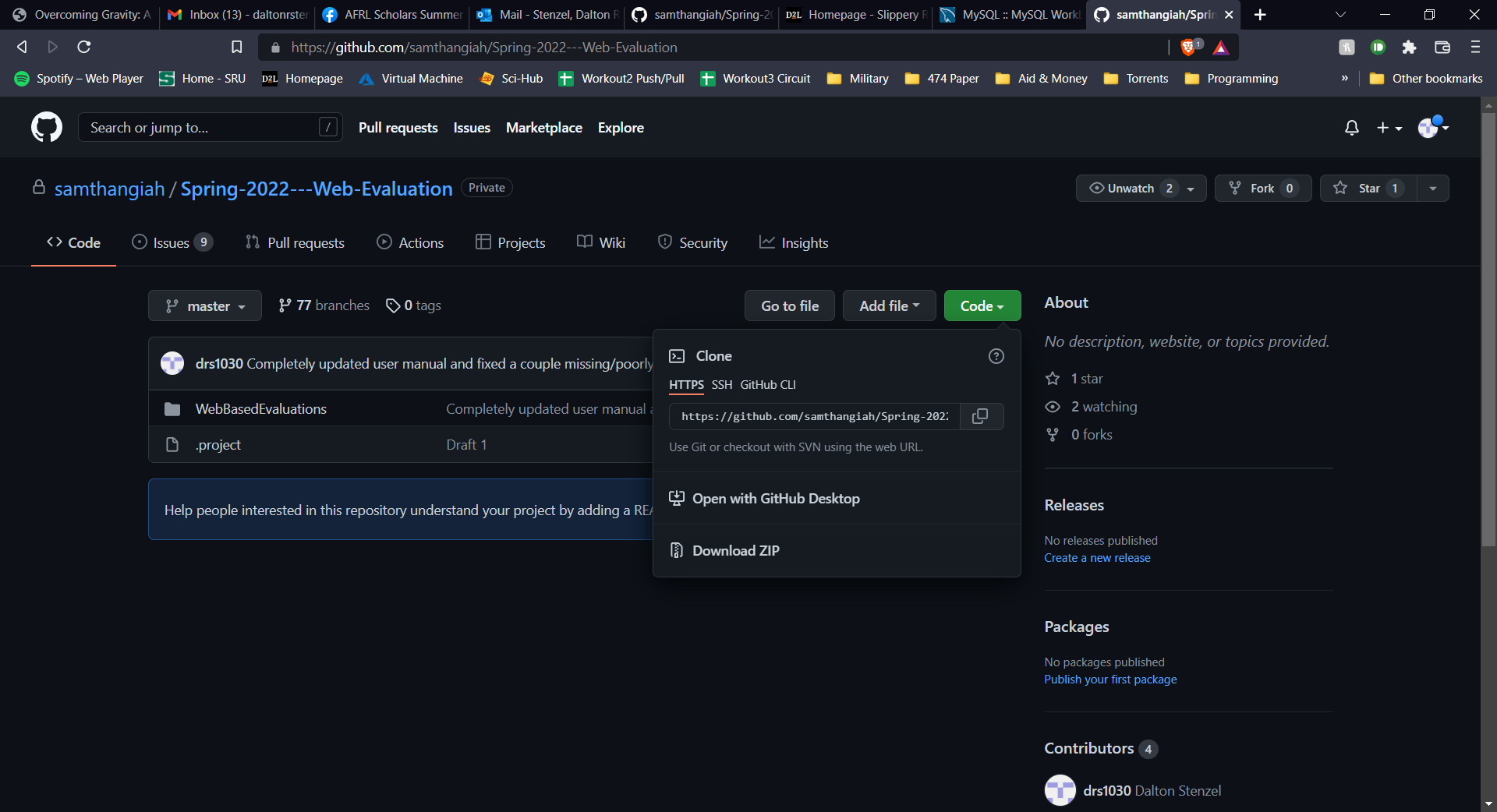


**Importing the Project with Smart Import**

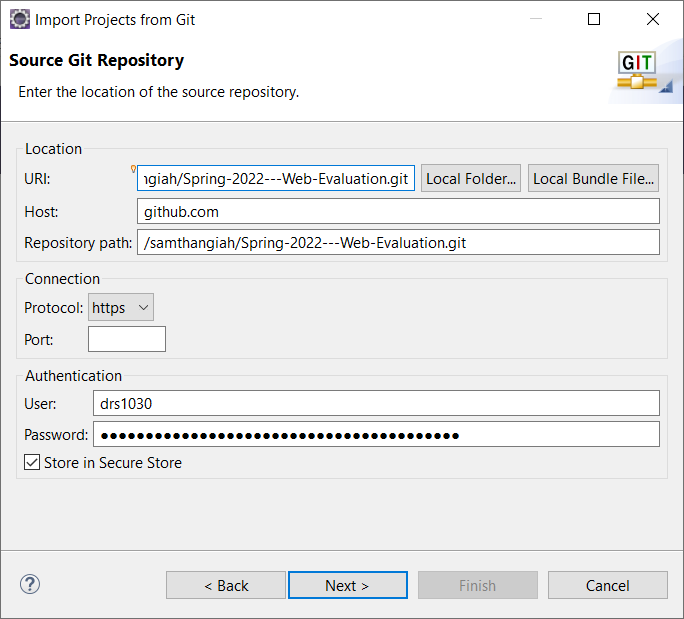
* Once you’re in the import window, click the **Git** folder arrow to expand what’s inside and select **Projects from Git (with smart import)** as seen below.



* In the **Import Projects from Git** windows, select **Clone URI** and press next
* 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.

­­­­­­­­­­

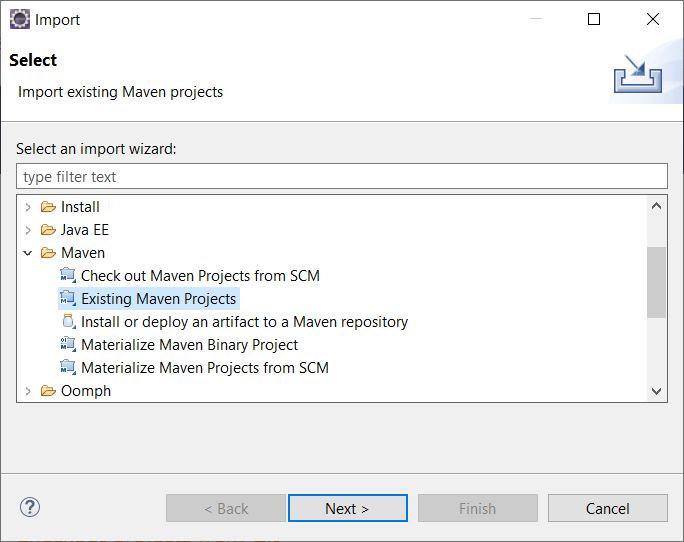
* Next, take that URI and paste it into the URI text box and the rest will automatically fill in, an example of what it should look like is displayed below.



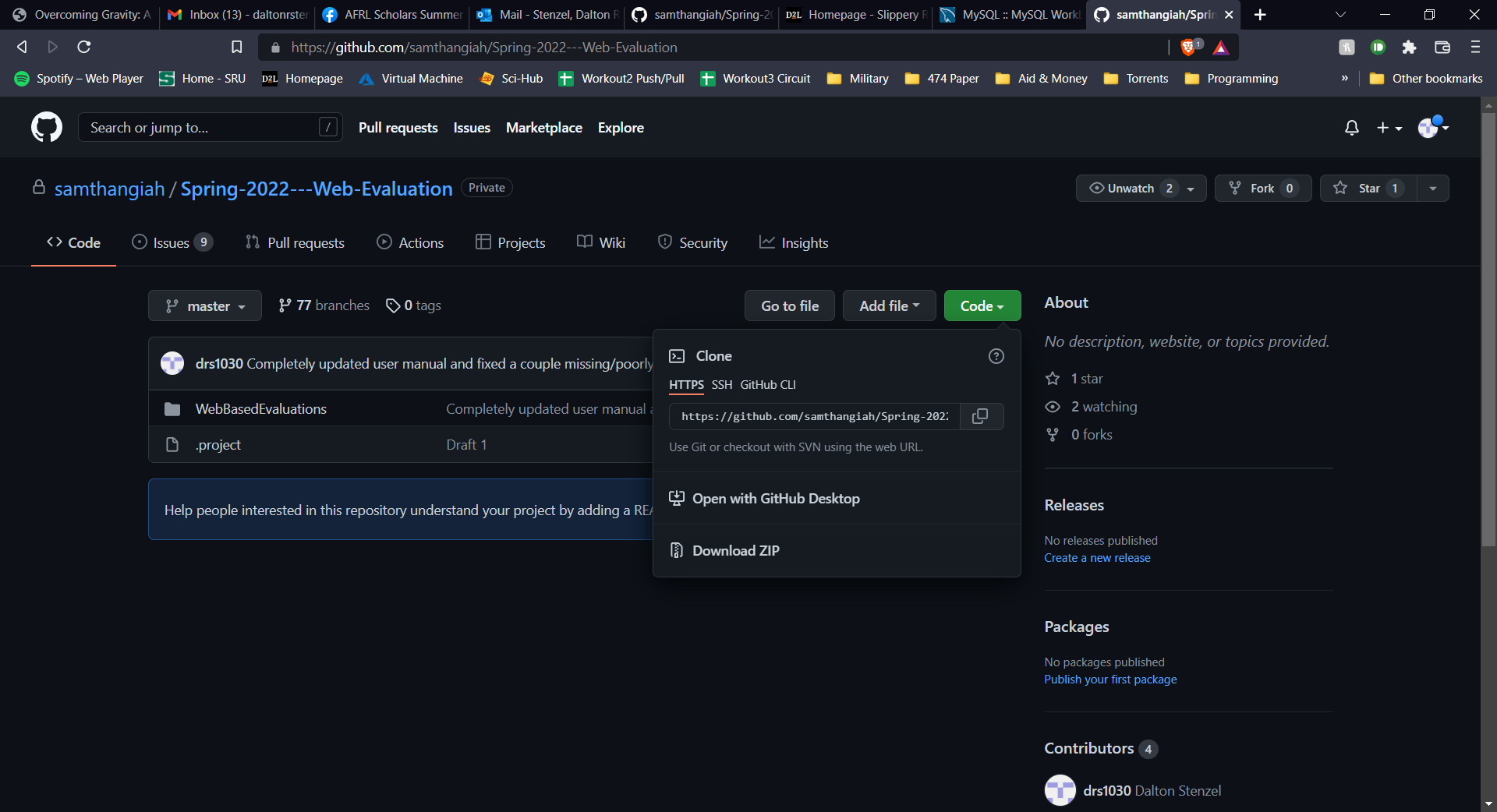
* Select **Next** and on the new page, search for the branch **master** and select **Next** once more. You’ll be taken to a page where you must select the directory of the destination of the GitHub files. Either leave it as the default location or user **Browse** to manually choose where to place the files.
* 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**

* 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.



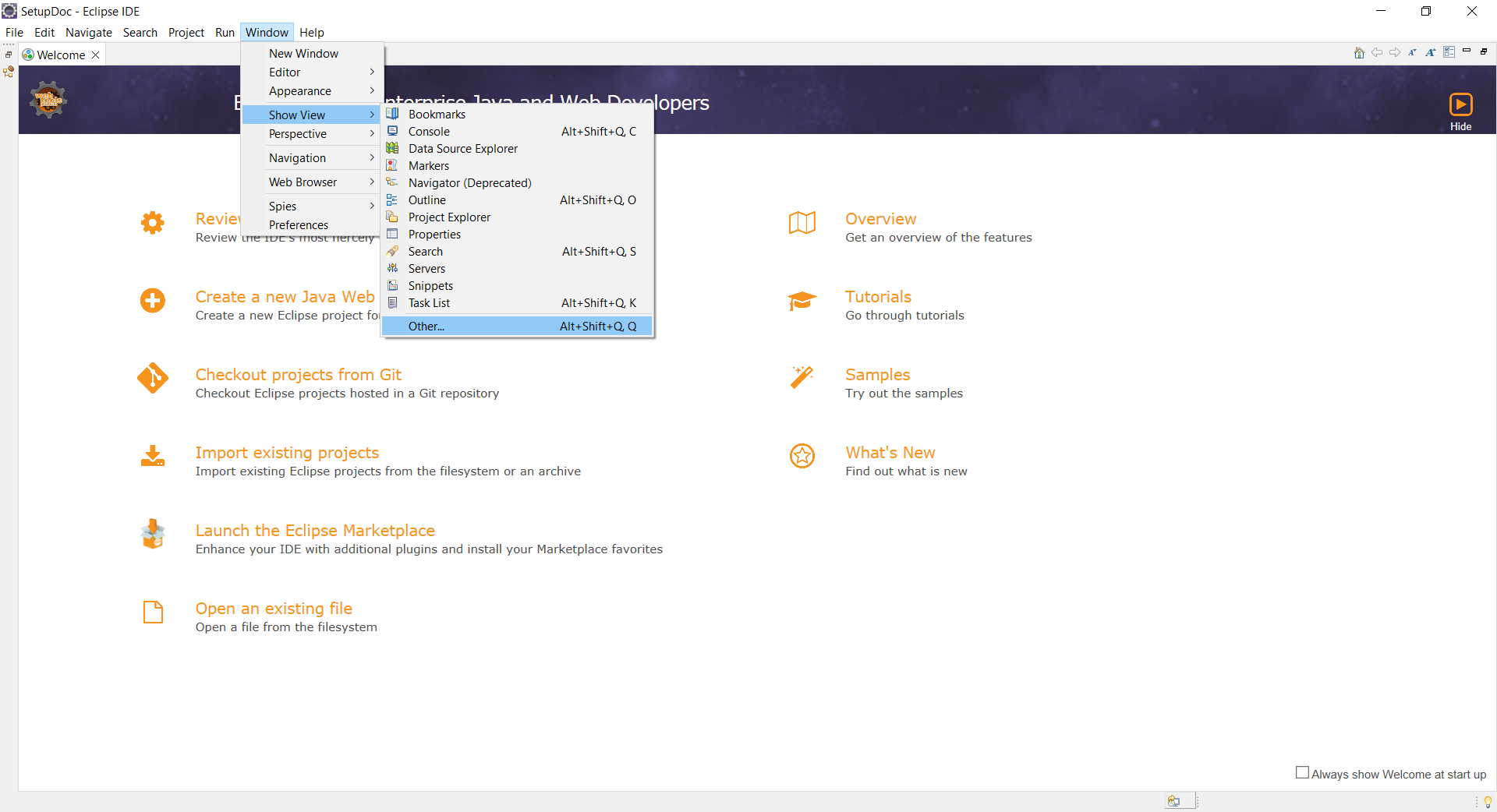
* You must download the zip file of the master branch of the project by going to the GitHub page for the project and clicking on the green **Code** button and pressing **Download ZIP** button. Once the download is complete, you must unzip the file into a folder with your decompressing program of choice (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.



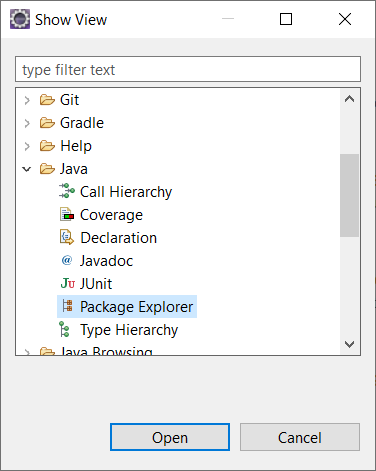
* On the next page, you’ll need to press the **Browse…** next to the **Root directory** box in order to choose the unzipped folder that you previously downloaded from the GitHub. Once that the project folder has been selected, press **Finish**

**Viewing the Packages**

* 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.

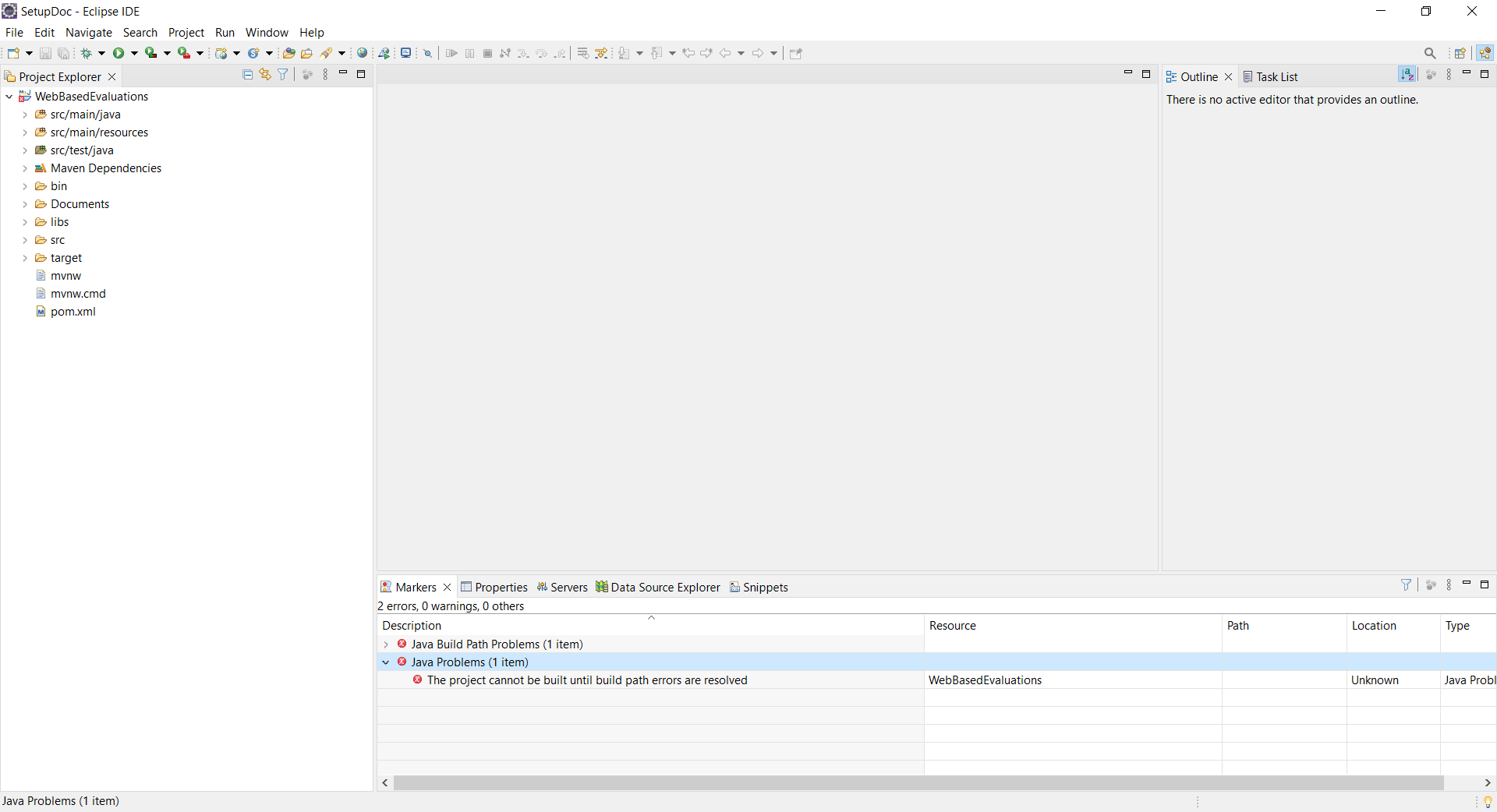


* A **Show View** window will display, and you must search for a folder named **Java** and expand it with the arrow right next to it to display the folder’s contents. Select **Package Explorer** and press the **Open** button.

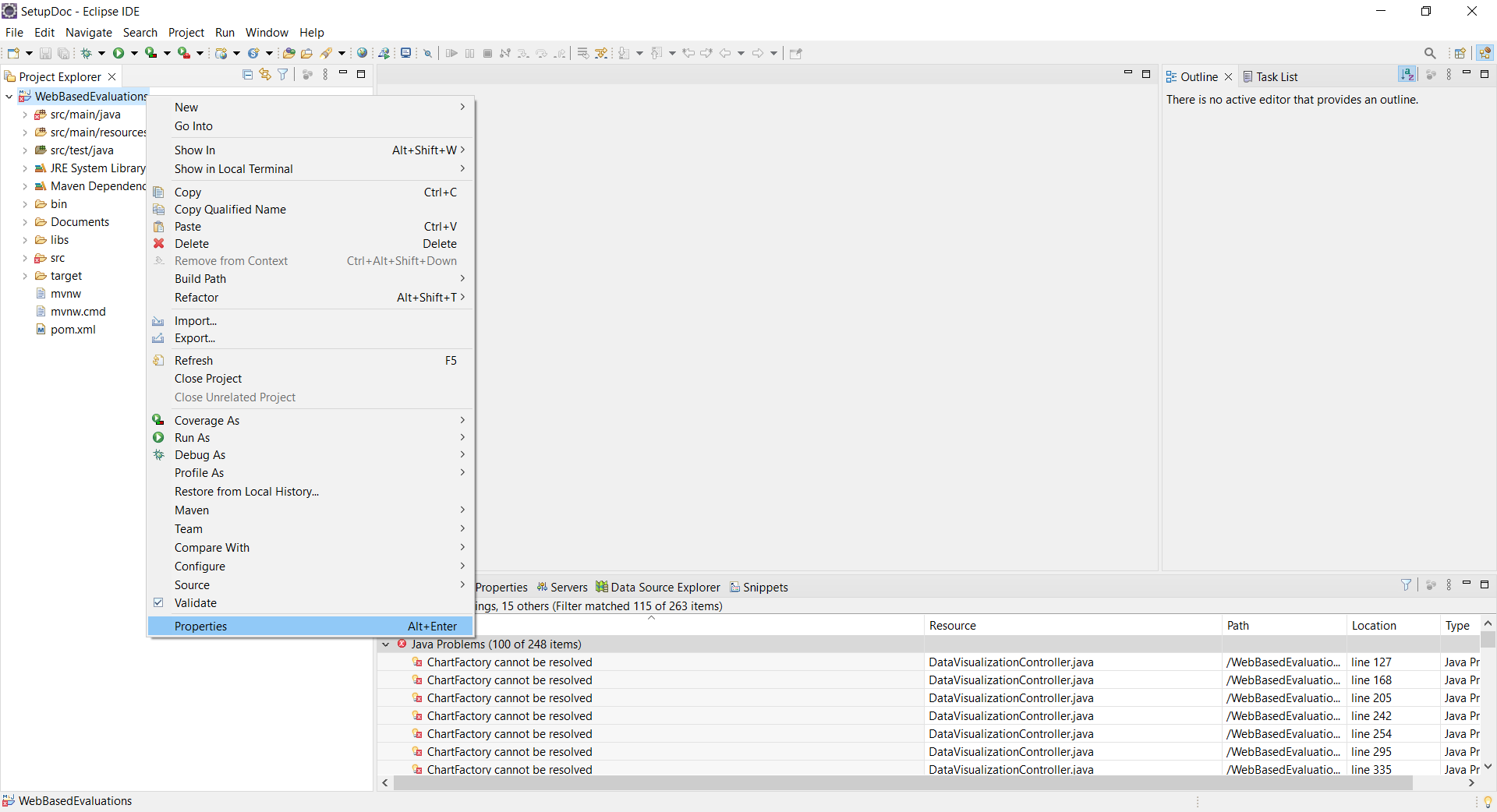


**Resolving Build Errors**

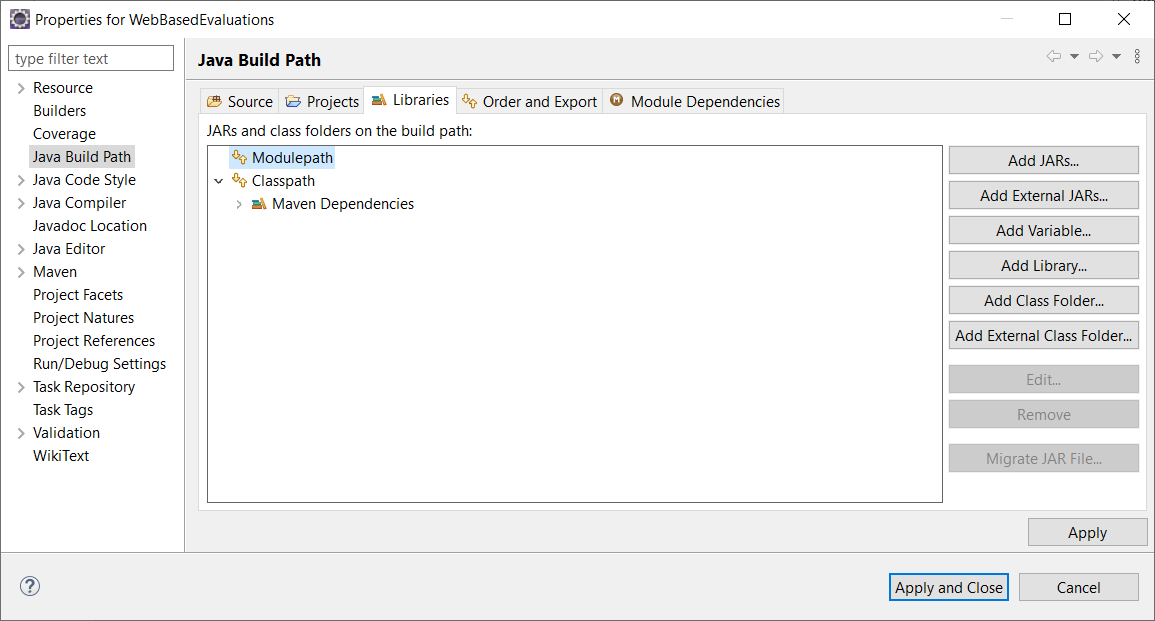
* If you come across errors in the **Markers** tab located near the bottom of the program, that’s like due to the fact that Java SE 17 wasn’t selected as the default version.



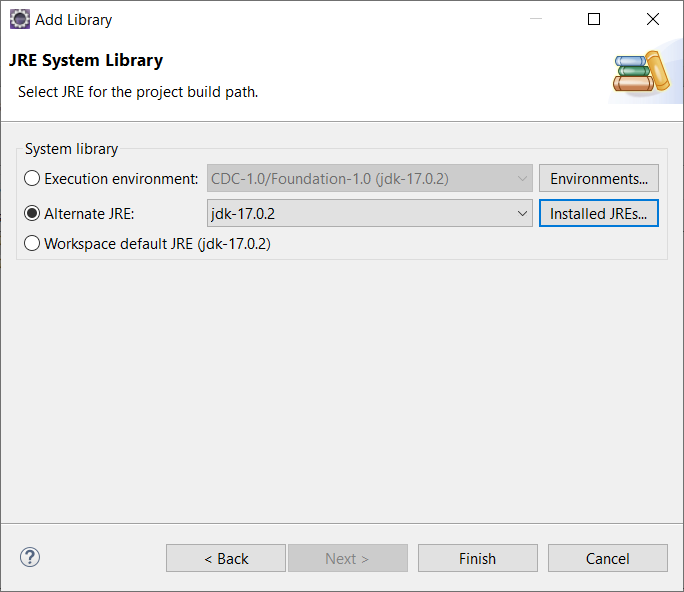
* Right click on the project in either the **Project Explorer** or **Package Explorer** and select **Properties.**



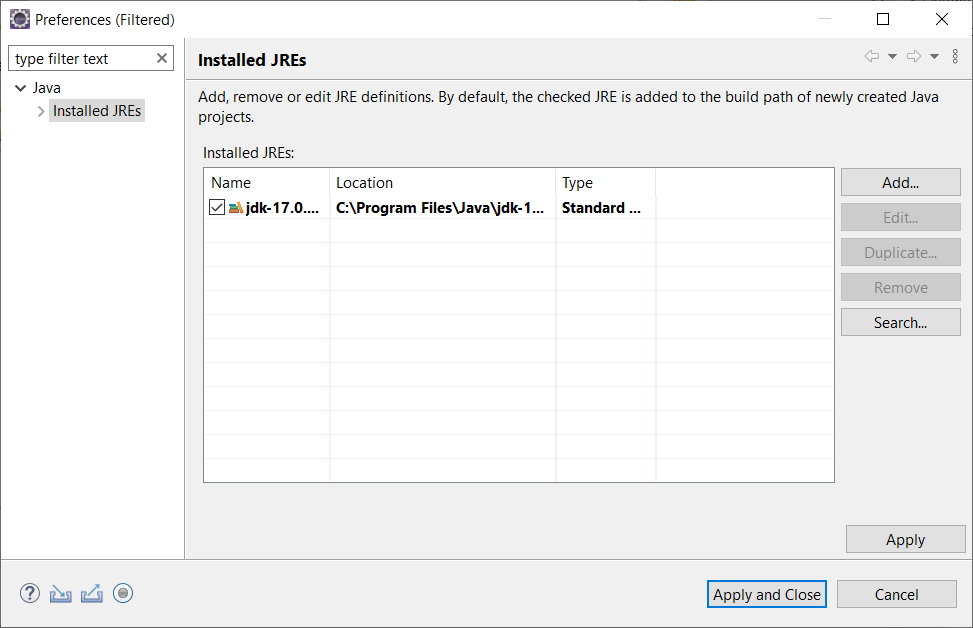
* Then on the right side, select **Java Build Path** and from within that newly opened area press **Libraries.**



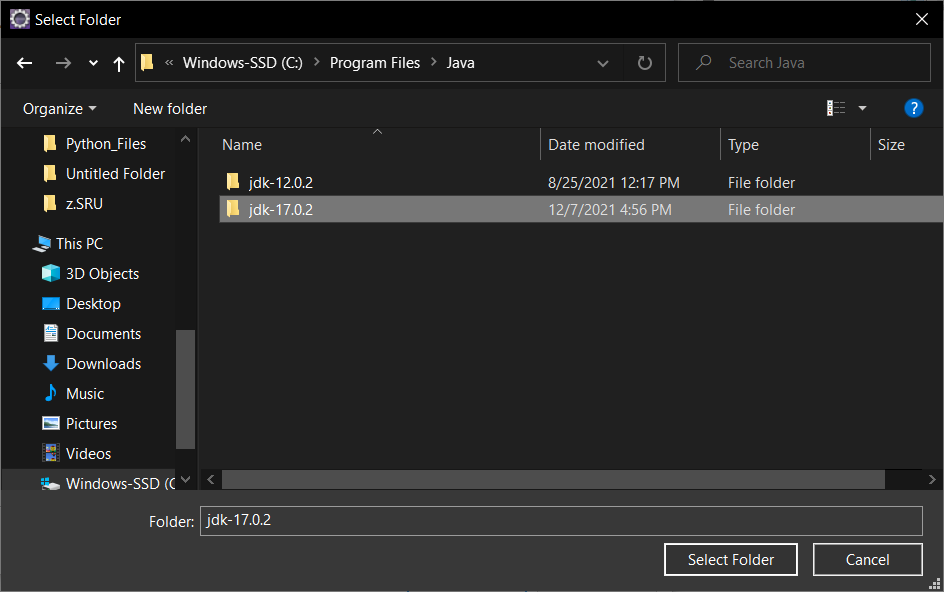
* From with the **Libraries** tab, select **Modulepath** and press **Add Library…** located to the left to bring up a new window.
* From the **Add Library** window, select **JRE System Library** and press **Next.**



* 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.



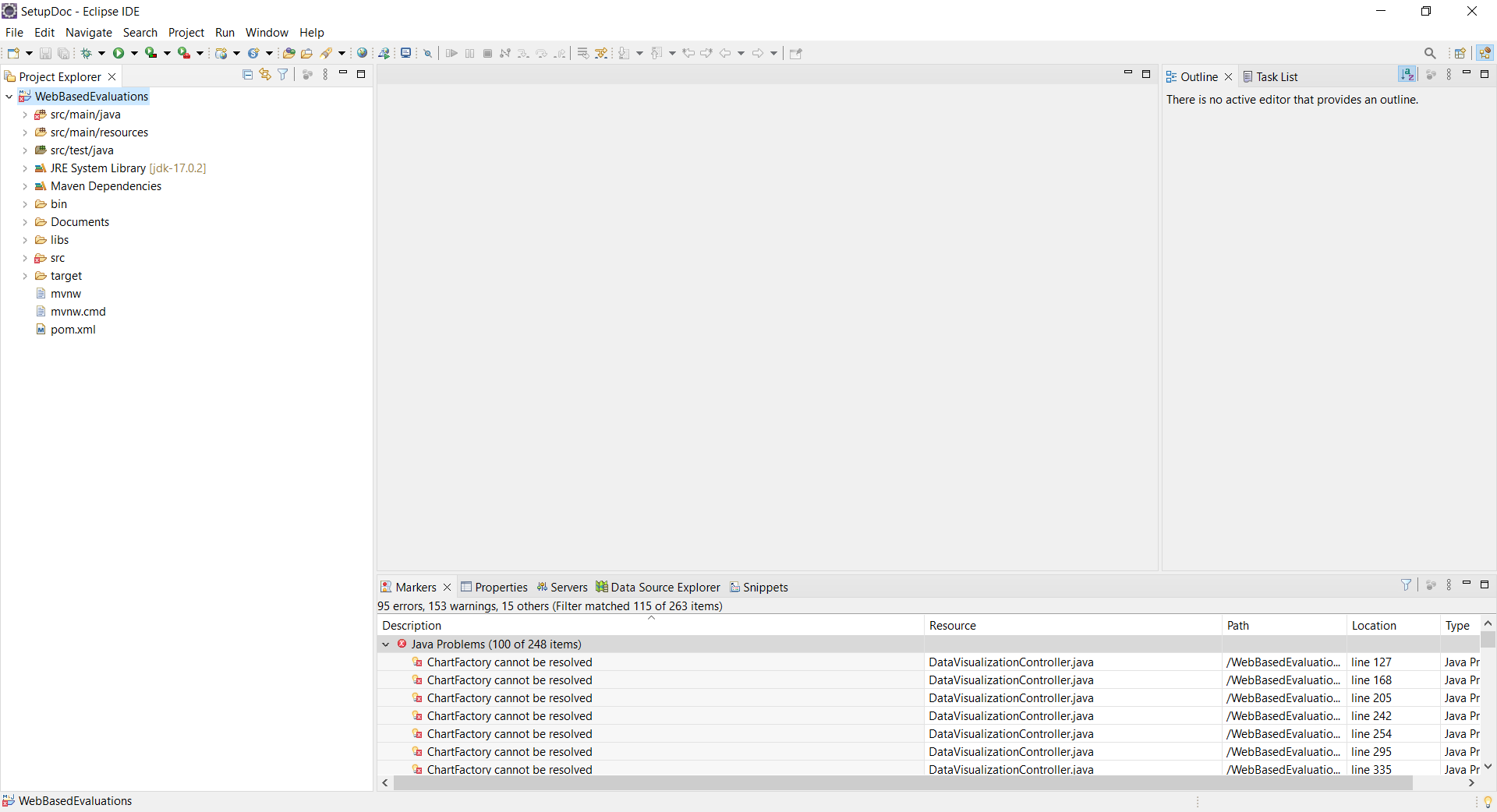
* From this new window, press the **Add…** button located on the right. A new window will pop up, named **Add JRE** where you will select **Standard VM** and press N**ext.**
* After pressing the next button, a you’ll see a **JRE home** box with a **Directory…** button to the right of it, press the **Directory…** 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.



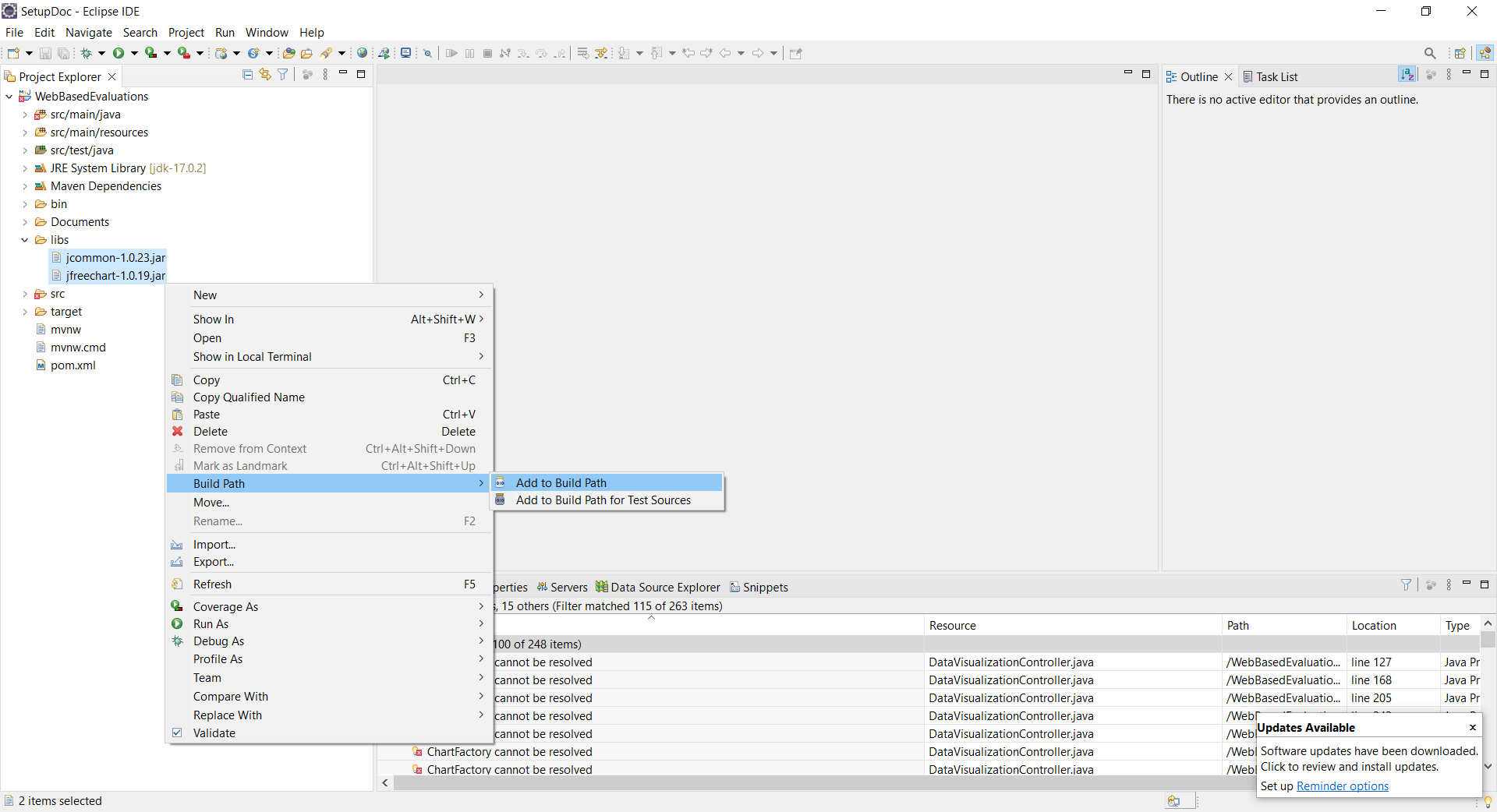
* After selecting the folder with the **Select Folder** button, you should be able to close out of everything with a series of **Finish** and **Apply and close** buttons.

**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.

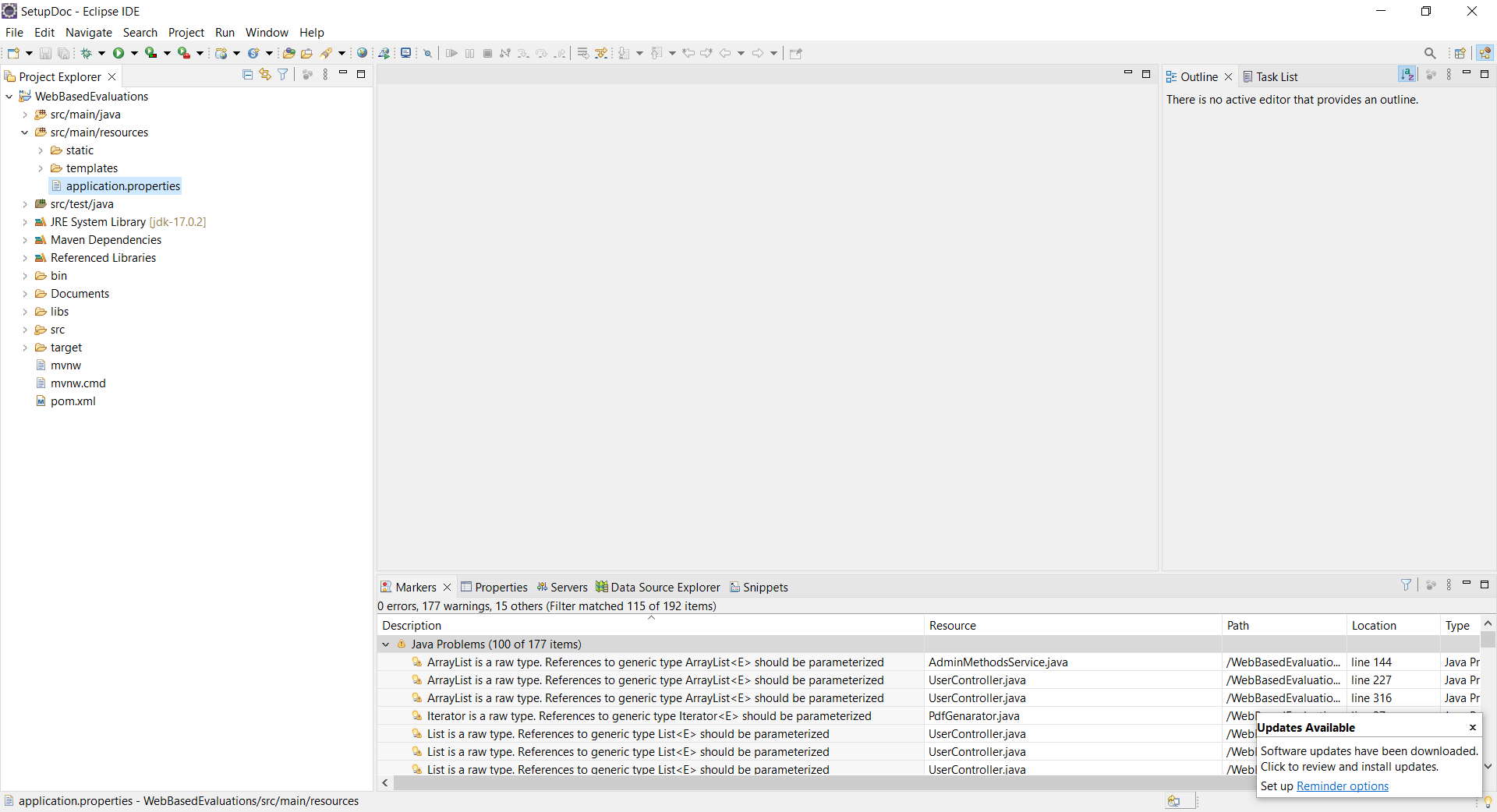


* Under the Project folder, find the **libs** folder and select both **jcommon-1.0.23.jar** and **jfreechart-1.0.19.jar** and right click them. From there, go to **Build Path** and then **Add to Build Path** to resolve the issues.



**the Program’s Database**

* 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.



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

Text

Description automatically generated

* 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”.
* 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 user. 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. 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.