WEB ROUTING

INSTALL CONFIGURATION MANUAL

SLIPPERY ROCK UNIVERSITY

Thomas Haley – [tjh1019@sru.edu](mailto:tjh1019@sru.edu)

Beth Orgovan – [bro0700@sru.edu](mailto:bro0700@sru.edu)

Dakota Myers – [drm1022@sru.edu](mailto:drm1022@sru.edu)

Sinchana Kori – [ssk1002@sru.edu](mailto:ssk1002@sru.edu)

TABLE OF Contents

[1 . Application Requirements 3](#_Toc133097880)

[2. Getting The Application VIA GITHUB / ECLIPSE 3](#_Toc133097881)

[2.1 Downloading Application via GitHub 3](#_Toc133097882)

[2.1.1 Locating The GitHub Repository 3](#_Toc133097883)

[2.1.2 DOWNLOADING THE PROJECT 3](#_Toc133097884)

[2.1.3 Accessing the Project in Eclipse 4](#_Toc133097885)

[2.2 Cloning Application Into ECLIPSE IDE 6](#_Toc133097886)

[2.2.1 Cloning the Application By URI 6](#_Toc133097887)

[2.2.2 Fetching Password for Import projects from Git screen 9](#_Toc133097888)

[3.Configuring the MYSQL DATABASE 12](#_Toc133097889)

[3.1 INSTALLING MYSQL 12](#_Toc133097890)

[3.2 Starting MySQL 13](#_Toc133097891)

[3.2.1 SERVER STATUS NOT RUNNING 14](#_Toc133097892)

[4. Setting Up Application Properties 14](#_Toc133097893)

[5. Starting the Webrouting Application 16](#_Toc133097894)

[5.1 Running the Project 16](#_Toc133097895)

[6. Accessing the Application 18](#_Toc133097896)

[6.1 Where to Access the Application 18](#_Toc133097897)

[7. Figures 19](#_Toc133097898)

# 1 . Application Requirements

**Required Computational Aspects necessary to run this project.**

1. **Java JDK:**  Java SE 17 - Available from<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>
2. **MySQL:** MySQL Server,My SQL Workbench & My SQL Shell **8.0.30** – Available from:<https://dev.mysql.com/downloads/mysql/>
   1. Reference Section 3 for more information
3. **Java IDE:** 
   1. **Recommended IDE**: 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>
      1. **Version Eclipse 2021-12R**

**4. Updated Internet Browser**

# 2. Getting The Application VIA GITHUB / ECLIPSE

To set up the application it must be downloaded into your system. This section will highlight how to accomplish this.

## 2.1 Downloading Application via GitHub

This sub-section will walk you through how to download a zip folder from GitHub containing all relevant material to the application. Then opening the folder into the Eclipse IDE(See Section 2.1.3).

### 2.1.1 Locating The GitHub Repository

Finding the GitHub Repository where the application is located can be done by following the link below:

Link: <https://github.com/samthangiah/Spring-2023-Web-Routing-Auctioning>

**You might need a GitHub Account associated with your school account to access/view or download from this repository**

### 2.1.2 DOWNLOADING THE PROJECT

Once reaching the GitHub repository you will want to be on the **Code** tab seen at the top of the page (Figure 1).



Figure 1 – Code Tab

Assuring you are on the Code tab you should see a **Code button** this time green with a drop-down arrow on it (Figure 2).

Graphical user interface, application

Description automatically generated

Figure 2 – Code button

Once locating the button click on it and select Download ZIP from the drop-down menu(Figure 3).

Graphical user interface, application

Description automatically generated

Figure 3 – Download ZIP Button

After clicking this button, a download should start in your browser for a zip file.

Assure that the Download has taken place and finished on the system before moving to section 2.1.3.

### 2.1.3 Accessing the Project in Eclipse

**STOP!**

Before continuing with this section make sure you followed section 2.1.2 and that the project ZIP File is downloaded.

Upon successfully completing section 2.1.2 you will now have a zip file where you designated it to download, access it and extract the zip file downloaded in section 2.1.2. Once extracted there will be a folder titled: **Spring-2023-Group-4-Web-Auctioning-Routing-System-master**

Now you will open the Eclipse IDE. Eclipse will prompt you for a workspace, navigate to where you extracted the **Spring-2023-Group-4-Web-Auctioning-Routing-System-master** and have it selected as your folder as the workspace and select **Launch** (Figure 4).

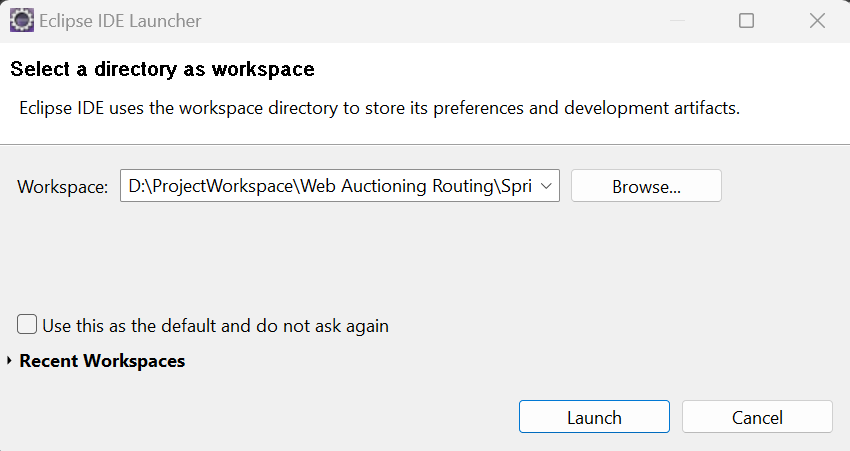


Figure 4 – Launch Eclipse with File as Workspace

Eclipse will launch and you will see the Web Auctioning - Routing System Application on the side in the Package Explorer (Figure 5). You will need to update the project’s dependencies before continuing (See Section 2.1.3.1 Below).

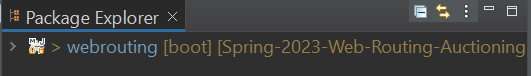


Figure 5 – Web Routing-Auctioning Application

#### 2.1.3.1 Updating Projects Dependencies

From section 2.1.3 you will right-click onto the **Web Routing-Auctioning** folder and select **Maven**, now you will select **Update Project** and press **OK** (Figure 6).

Text

Description automatically generated

Figure 6 – Updating Maven Project

Once this has completed you may now move onto Section 3.

## 2.2 Cloning Application Into ECLIPSE IDE

This section will walk you through the steps of cloning the Application from the GitHub repository directly into the Eclipse IDE. This section will outline how to achieve this with the Eclipse IDE and may not work for some Java IDE’s.

### 2.2.1 Cloning the Application By URI

Create a new empty Workspace folder in the desired location by right clicking and creating new folder.

Start the Eclipse IDE and launch the newly created workspace by clicking browse and click launch. In the top left corner of the Eclipse IDE Select **File**, next you will select **Import** (See Figure 7).

**Note-** See the Requirement Section on Which Java IDE to download (Section 1).

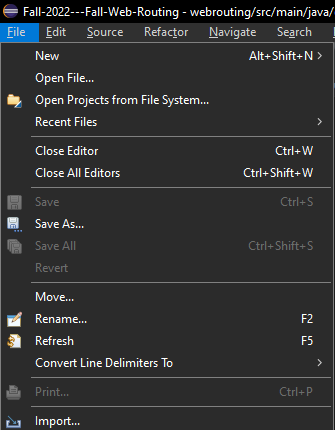


Figure 7 – Import Button

After selecting **Import** the **Import** window will be open on your screen. You will select Projects from the **Git** Folder

This will expand and you will see **Projects from Git** and **Projects from Git (with smart import)**. Select **Projects from Git (with smart import)** and select **Next** at the bottom of the **Import Menu** (See Figure 8). A screenshot of a computer

Description automatically generated with medium confidence

Figure 8 – Import Menu

After selecting **Next,** the **Import Projects from Git** window will open, select **Clone URI** then click **Next** (Figure 9).

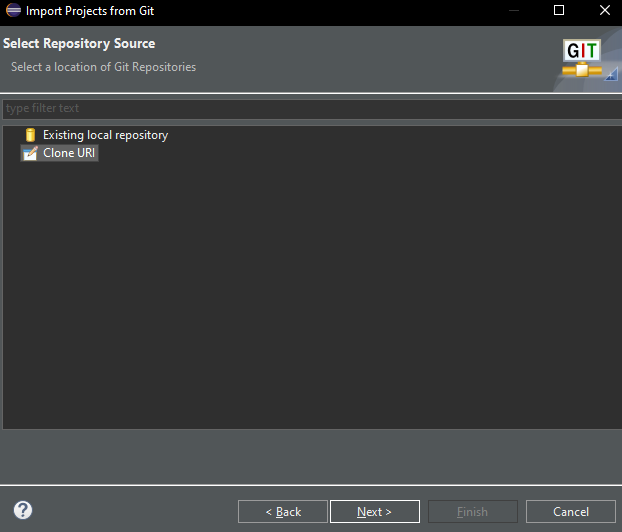


Figure 9 – Clone URI

This will prompt the Source Git Repository Window to open and, in the **URI** Field, enter the below **URI**.

**URI:** https://github.com/samthangiah/Spring-2023-Web-Routing-Auctioning.git

Once doing this the fields **Host** and **Repository path** will be filled out with the appropriate information. You will also see that the **Protocol** field is set to **HTTPS** and the **Port** field will remain blank. The only fields that need to be set now are the **User** and **Password** Fields, type in your SRU Email ID next to User field (same credentials you used while creating your GitHub account) and for the password follow the steps from section 2.2.2 (Figure 10).

Paste the personal token access (look for section 2.2.2 – below this section) generated in GitHub in the password field and make sure to check the Store in Secure Store checkbox. Your Import Projects from Git should look exactly like below except for the User field which will have your sru.edu email ID.

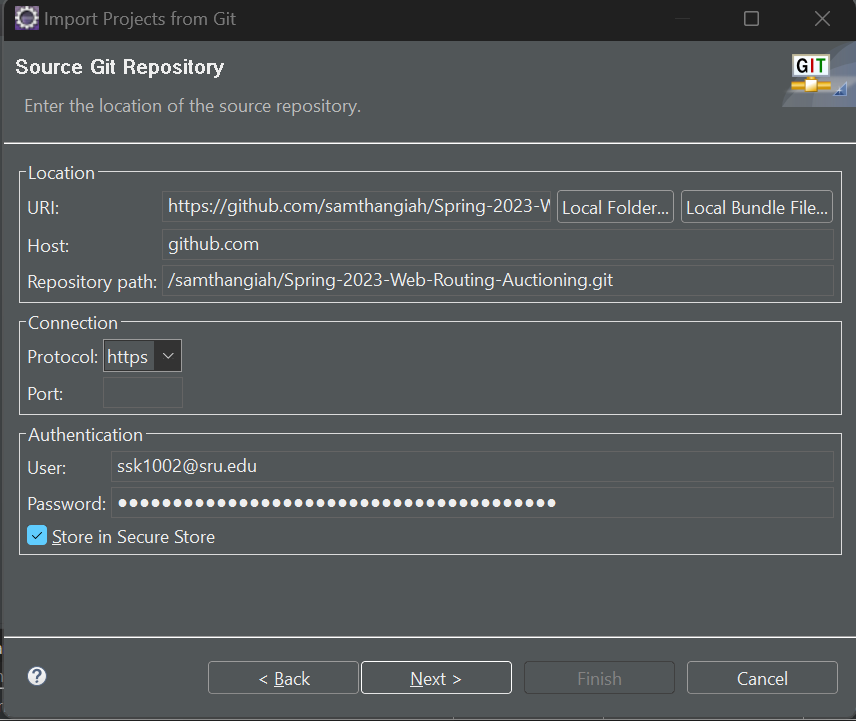


Figure 10 – Source Git Repository Window

After Clicking **Next** on the **Source Git Repository Window** (Figure 10). The **Branch Selection Window** will open, under that, make sure all branches have been selected and make sure “**When fetching a commit, also fetch tags**” is checked out and click **Next**.

On the **Local Destination** page, choose the file directory where the repository will be saved in and select **Next**.

The project will copy itself to the destination that was selected. Verify that **Spring-2023-Group-4-Web-Auctioning-Routing-System** is selected and choose **Finish**.

In the **Package** **Explorer** you should see a project. To access the application, choose **webrouting**. It is recommended to update the Maven dependencies to do to so See Section 2.1.3.1.

### 2.2.2 Fetching Password for Import projects from Git screen

Open your GitHub account, click on the **profile icon** on the top right corner of your GitHub screen, it should show you a drop-down menu.



Figure 11 – Profile Icon in GitHub

In the drop-down menu, click on the **settings**. The settings page should appear when click on settings. In the same page, scroll down until you find **< > Developer Settings** Tab towards your left side of the screen as shown below.

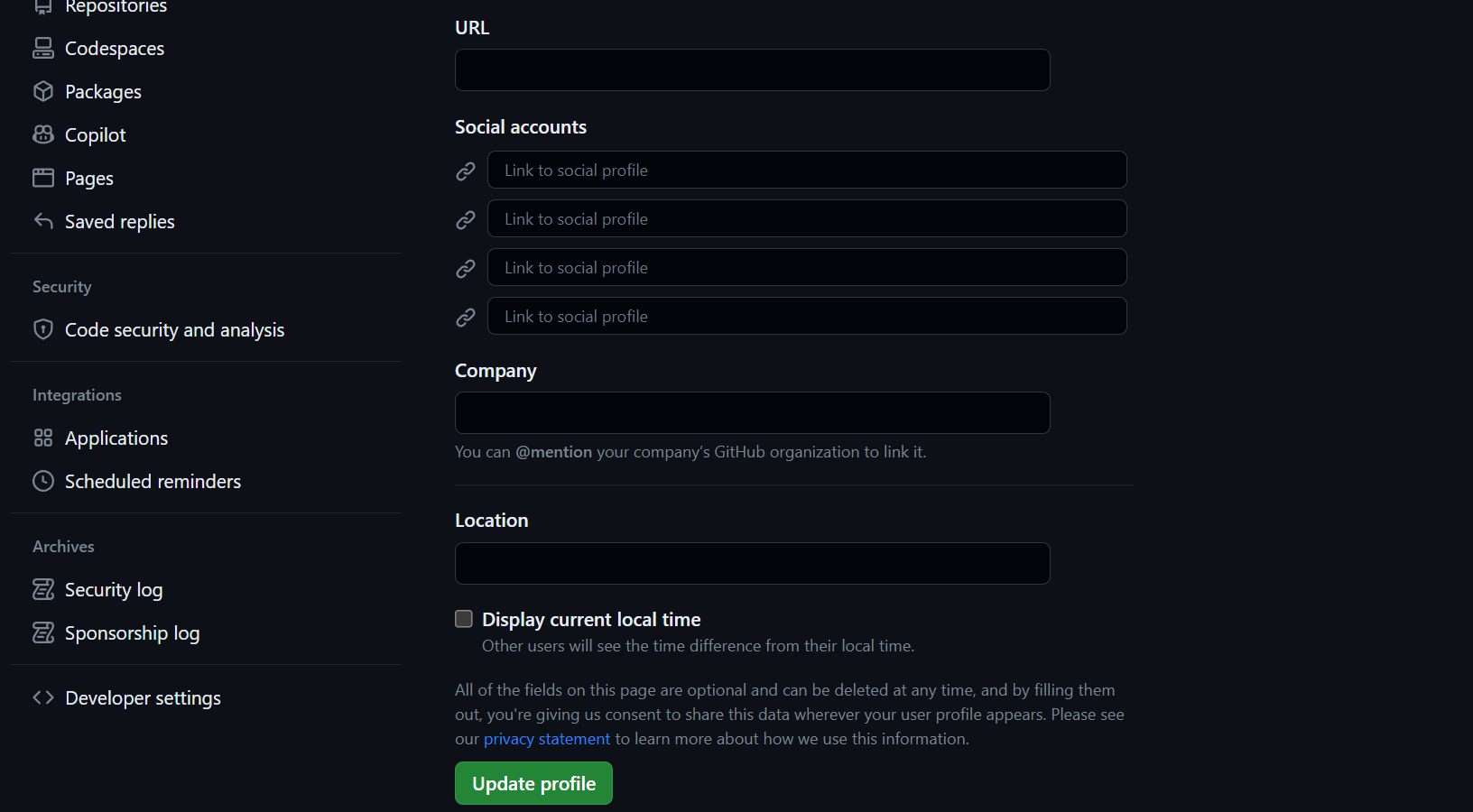


Figure 12 – Developer settings under settings in Profile icon drop-down menu

Click on the **< > Developer settings** tab and click on **Personal access tokens** with a key symbol next to it as shown below. After clicking that, choose **Tokens (classic)** option. All these options are present towards the left of the screen.

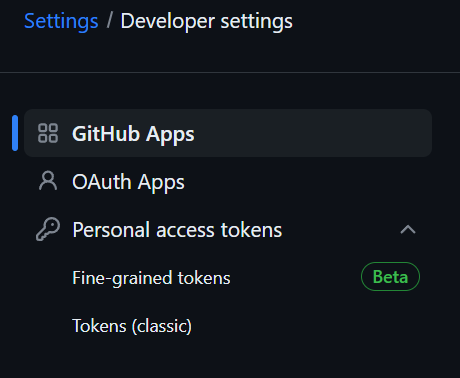


Figure 13 – List under Developer Settings

After clicking on the **Tokens (classic)** you should be able to see the screen as shown below.

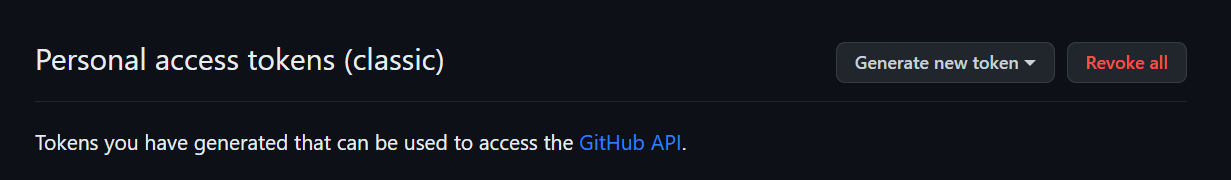


Figure 14 – Personal access token (classic) page

Click on **Generate new token** in the first list of the above figure, and click on **Generate new token (classic)**.

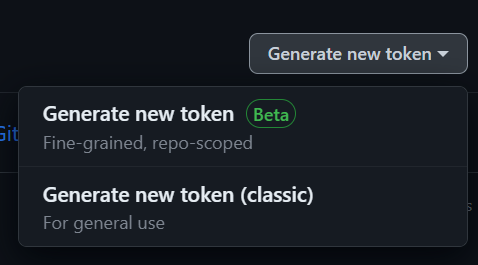


Figure 15 – Generate new token drop-down menu

After clicking on **Generate new token (classic)** option it will ask you to **confirm the access** by entering your password. After entering your password and confirming the access you will end up in **New personal access token (classic) page** that looks like below:

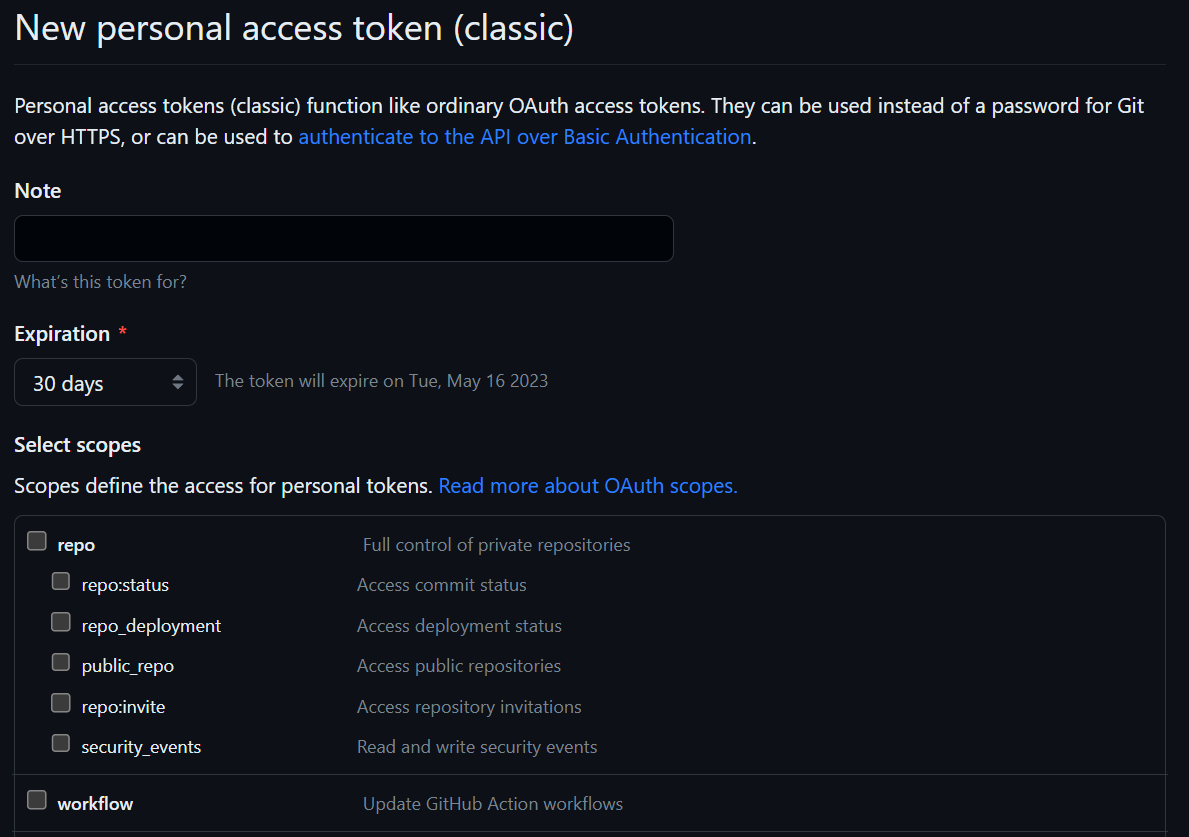


Figure 16 – Creating New token in New personal access token (classic) page

In Note, you can name the token as per your wish but the recommended token note for this application is “**Web Auctioning – Routing System**”. Set the expiration to **90 days**. In Select Scopes, make sure to check **repo check box** (it should check out everything starting from repo:status to security\_events).

Make sure you have completed all the above three main steps in here and then click on “**Generate Token**” by scrolling your screen downwards to the end.

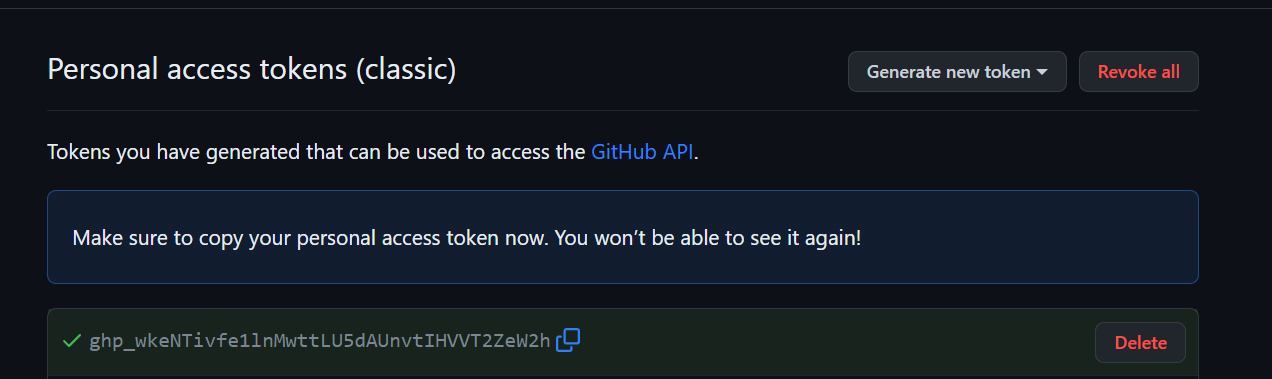
Type in your password for **confirming the access**. After you confirm the access by typing in the password, the below screen should appear.

Figure 17 – Personal access token successfully created page

This will be your **password** what you will be entering under password in the Import Projects from Git screen.

Make sure to copy and paste it in your notepad and save it, after it disappears on this GitHub screen you will not be able to access it again.

Now, copy this token and follow the next steps after entering **User** under section 2.2.1.

# 3.Configuring the MYSQL DATABASE

This project requires a local MySQL database. This section will highlight the steps needed to set up and install the MySQL Workbench and MySQL.

## 3.1 INSTALLING MYSQL

Download the MySQL installer here: <https://dev.mysql.com/downloads/>

Select Version 8.0.30 and ensure that it is downloaded.

Once the download has completed Run the installer and select Custom for setup type.

This will open the Select Products page. You will need to Select **MySQL Server 8.0.30, MySQL Workbench 8.0.30, and MySQL Shell 8.0.30.** Next you will need to define your **Port**, Your **Root** Username and the **Password** associated with the **Root User**. For more information a guide from MySQL on the Installation can be found here: <https://www.simplilearn.com/tutorials/mysql-tutorial/mysql-workbench-installation>

**Note:** The default port is **3306** for MySQL. This port can vary for installers due to port conflicts. The application will dynamically find your MySQL port.

**Note:** Under development it is recommended to define **Root** as “root” and **Password** as “password”.

## 3.2 Starting MySQL

To run the Application, you will need to verify that your **MySQL server** status is running.

Start by starting the **MySQL Workbench** Application. Login with the credentials you set as **Root** and **Password**. Now you will Open your **local Instance/Connections**. (Figure 11).

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 18 – MySQL Local Instance/Connection

After Opening the Connection, you need to verify that the server is running to do so look at the left side of the Application you will select the **Server Status** button (Figure 12).



Figure 19 – Server Status Button

After selecting the **Server Status** button, a window Called **Administration – Server Status** (Figure 13) will open. You will see a **Server Status** symbol on the right-hand side of the window.

Graphical user interface

Description automatically generated

Figure 20 – Administration – Server Status

MySQL default is that your server is running. If it is not running, you will verbosely see that it is not running.

**If your server is not running, see section 3.2.1.**

Otherwise, continue to section 3.3.

### 3.2.1 SERVER STATUS NOT RUNNING

Navigate to the Navigator on the left side and click **Startup / Shutdown**

This will Open a Page called **Administration – Startup / Shutdown**

Click on the button Called **Start Server**

# 4. Setting Up Application Properties

To access the **application.properties** file open your **Eclipse IDE** and navigate to the **src/main/resources** folder, underneath the **webrouting project file**. You will see a file called **“application.properties”** select it and open it.

In this file you will see two properties the first being **spring.datasource.username=root** and the second being

**Spring.datasource.username=password** in this file you will need to change the **root** to your **MySQL Root username** and the **password** to the **password** you chose in section 3.1 (Figure 14).

Text

Description automatically generated

Figure 21 – Application Properties file

If you get an error related to the port 8080 or re-captcha, please change your spring.datasource.url=jdbc://mysql://locahost:3360/webrouting. Replace **localhost with locahost:3360**. The screen should look like below:

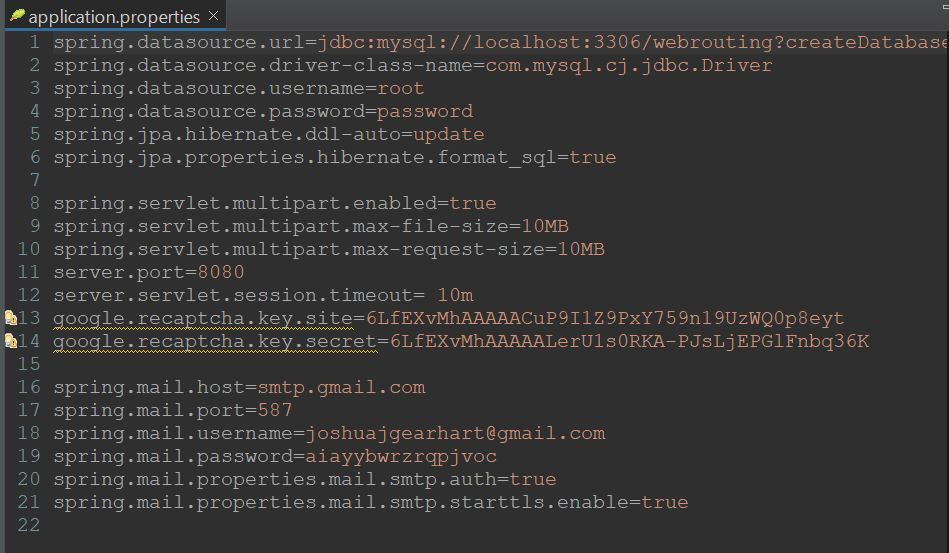


Figure 22 – Application properties after changing localhost into localhost:3306 in the first line

Incase, if the above error does not get fixed by changing localhost to localhost:3360, then, Right click on the **webrouting** (refer figure 5) and click build path and then click on configure build path. Go to the Source Tab, make sure that all Excluded: says None. See the below picture for make sure you are doing it correct.

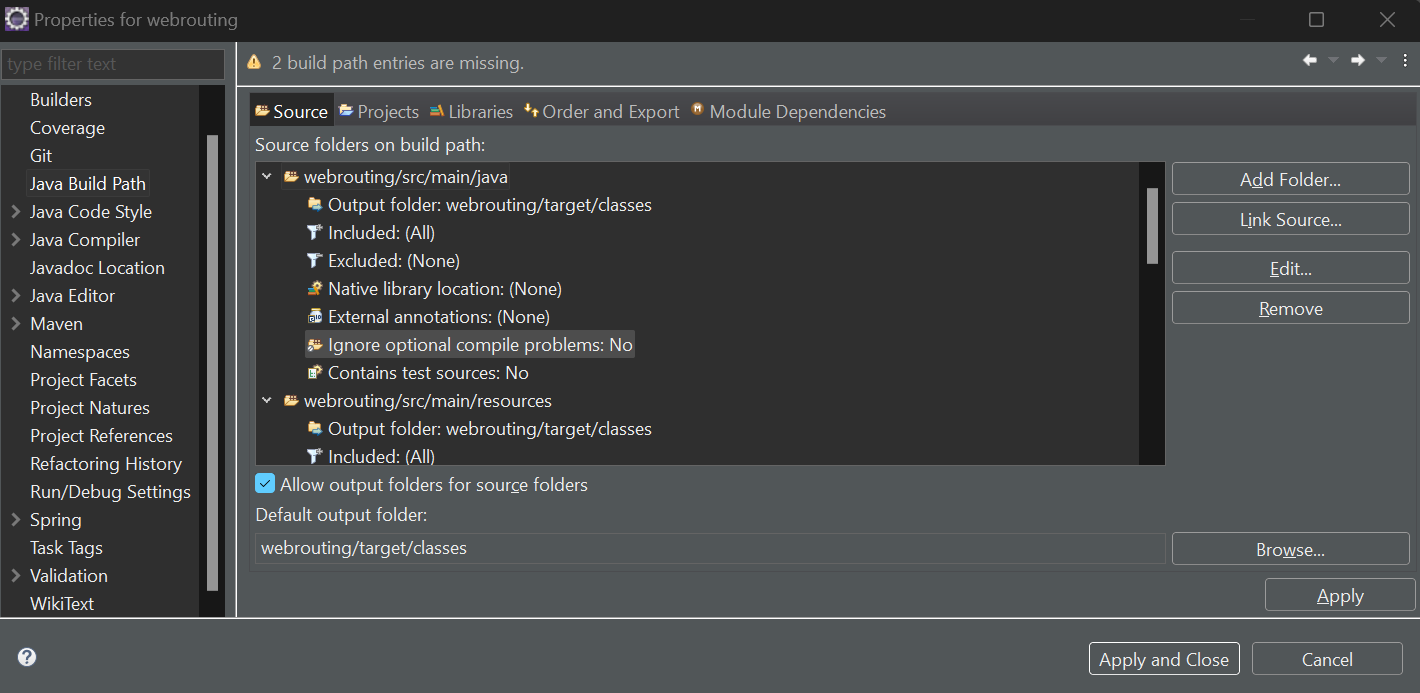


Figure 23 – Build Path by right clicking on webrouting

After changing the properties, the application will be able to successfully start.

# 5. Starting the Webrouting Application

This Section will highlight how to start the application properly.

## 5.1 Running the Project

In the **Eclipse IDE**, navigate to **src/main/java/edu.sru.thangiah.webrouting/WebroutingApplication.java**

Right-click on that file and select **Run As** then choose **Java Application** (Figure 15).

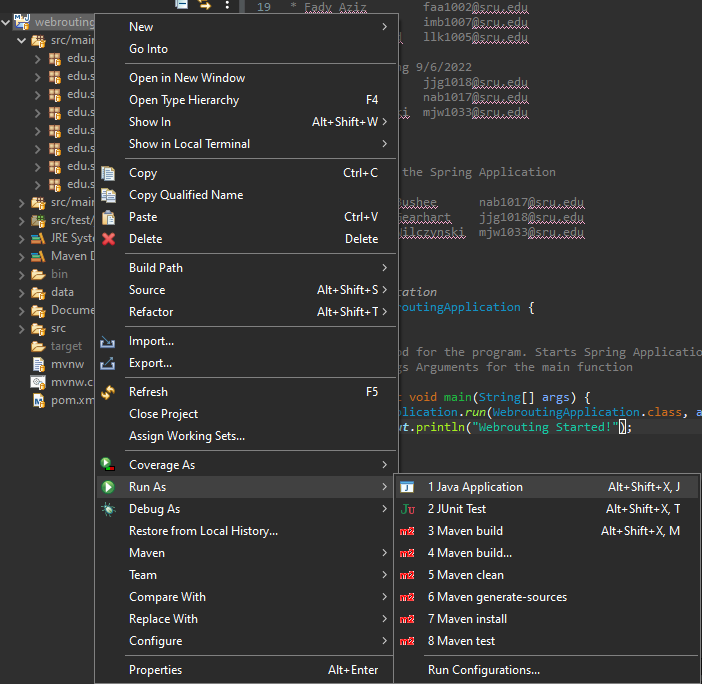


Figure 24 – Run Webrouting Application

After selecting this the application will begin to run at the bottom of your **Eclipse IDE** the **Console** should appear. Watch this **Console**, this may take a few minutes.

Wait until you See “**Webrouting Started!”** at the bottom of the console (See Figure 16).

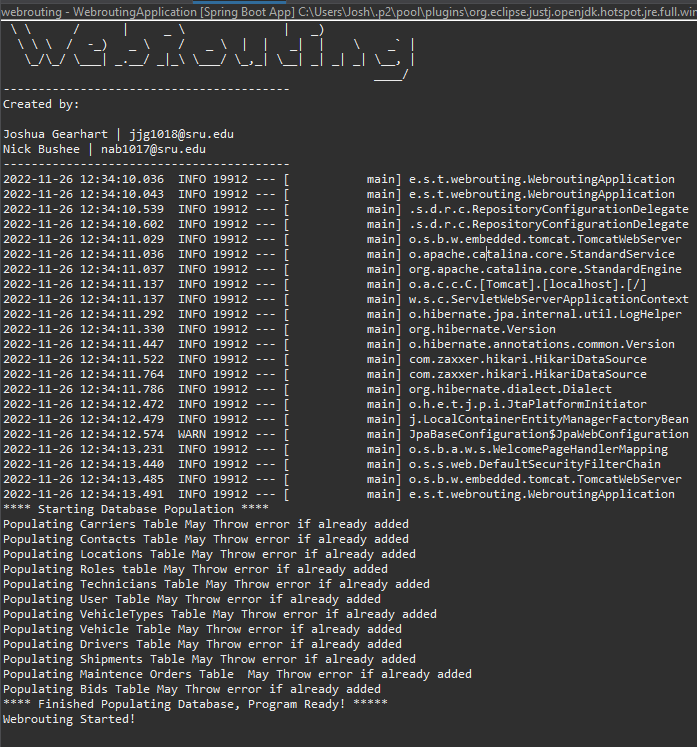


Figure 25 – Webrouting Console

Once verifying that the **Webrouting Started!** Message has been displayed you may now move on to accessing the Application (See section 6).

# 6. Accessing the Application

This section will show you how to access the Web Routing – Auctioning Application once it’s started.

## 6.1 Where to Access the Application

To open the application an internet browser is required. Open any internet browser of your choice once the application is running type <http://localhost:8080/> into the search bar. This will you take you to the **Enterprise Routing System** home page (Figure 17).

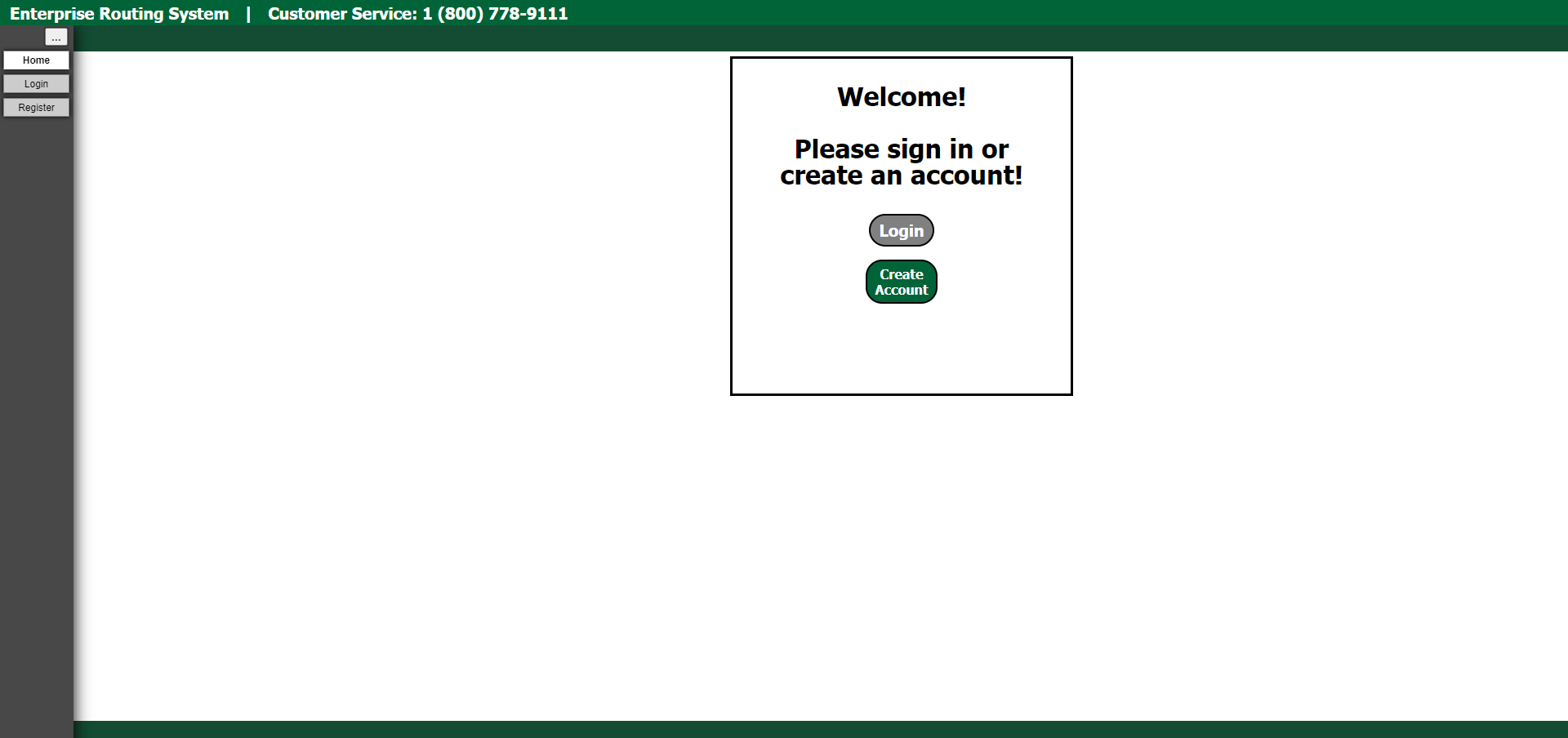


Figure 26 - Enterprise Routing Home Page

For more information and detailed guide on using the program please refer to the **User Manual.**

# 7. Figures

Figure 1 - Code Tab 3

Figure 2 – Code Button 4

Figure 3 – Download ZIP Button 4

Figure 4 – Launch Eclipse with File as Workspace 5

Figure 5 – Webrouting Application 5

Figure 6 – Updating Maven Project 6

Figure 7 – Import Button 7

Figure 8 – Import Menu 7

Figure 9 – Clone URI 8

Figure 10 – Source Git Repository Window 9

Figure 11 – Profile Icon in GitHub 9

Figure 12 – Developer settings under settings in Profile icon drop-down menu 10

Figure 13 – List under Developer Settings 10

Figure 14 – Personal access token (classic) page 11

Figure 15 – Generate new token drop-down menu 11

Figure 16 – Creating New token in New personal access token (classic) page 11

Figure 17 – Personal access token successfully created page 12

Figure 18 – MySQL Local Instance/Connection 13

Figure 19 – Server Status Button 13

Figure 20 – Administration – Server Status 14

Figure 21 – Application Properties file 15

Figure 22 – Application properties after changing localhost into localhost:3306 in the first line 15

Figure 23 – Build Path by right clicking on webrouting 16

Figure 24 – Run Webrouting Application 17

Figure 25 – Webrouting Console 18

Figure 26 - Enterprise Routing Home Page 19