Green Works Eco-map

Project Members:

Mariadelly Esparza

Marcos Santana

Derek Sheppard

Cason Fluharty

Kashai Bingham

Adrian Batista

Table of Contents

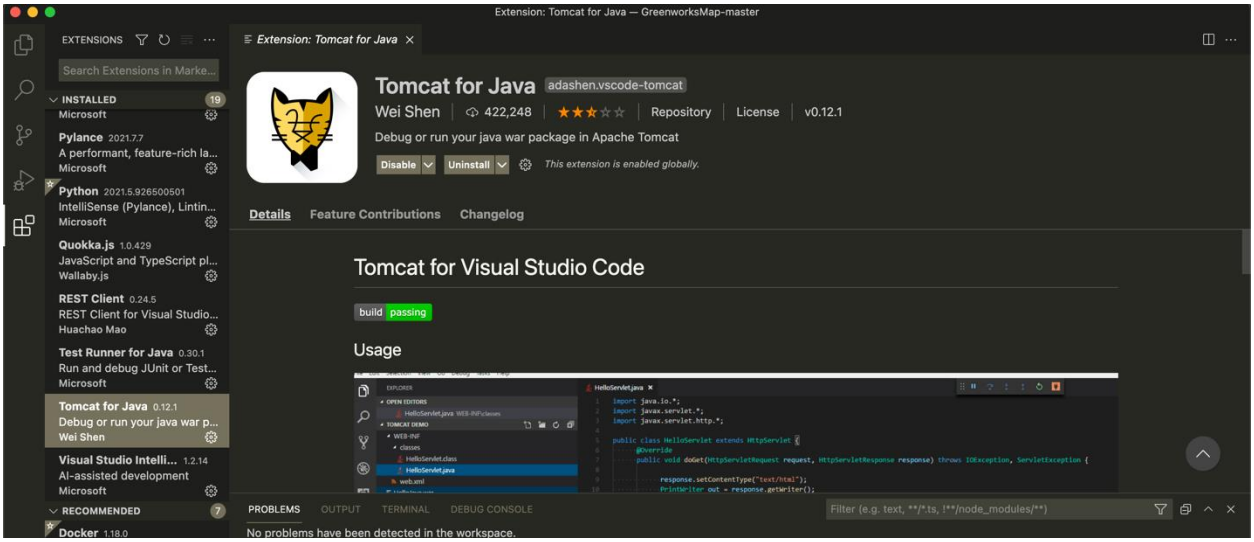
Development Environment

IDE

The IDE we used for this project is Visual Studio Code (VS Code).

IDE Extensions

We decided to add the Tomcat Server extension that came in VS Code. Here is a screenshot below.



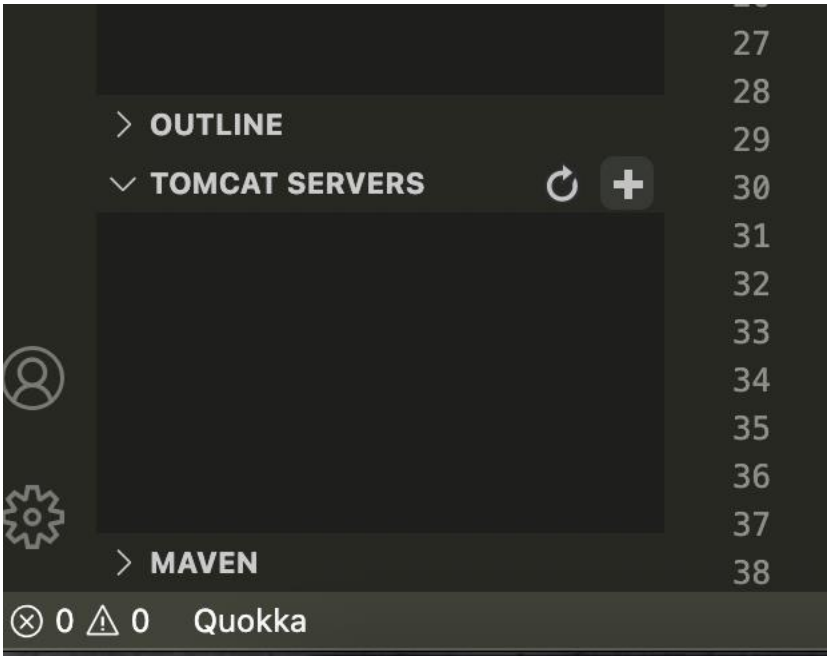
This allows the Green Works web app that we must run on a web server. For this to be functional, you must install Tomcat to your computer. You can find the link here: https://tomcat.apache.org/download-10.cgi.

Under the Core section, you have several links that you can download, depending on your OS.

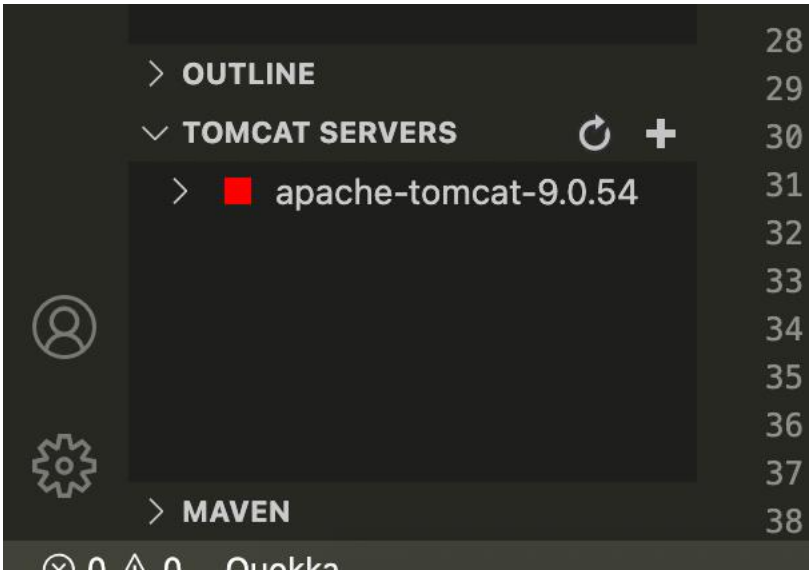
If you have the 32-bit for windows, download the 32-bit zip. Same goes for 64-bit file.

For Mac, download the tar.gz file.

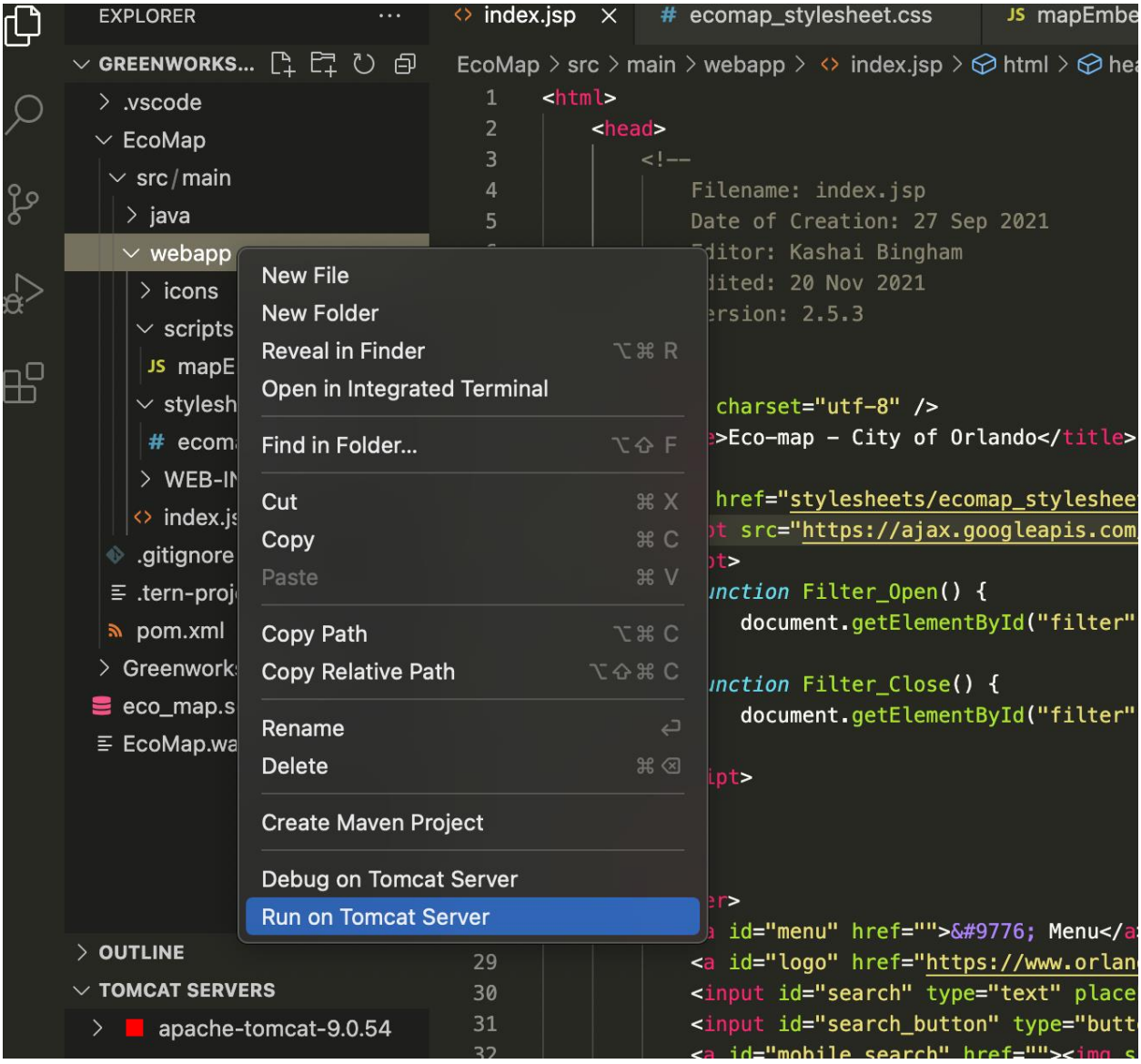
After downloading this pre-requisite, install the Tomcat for Java extension on VS Code, and follow these steps.



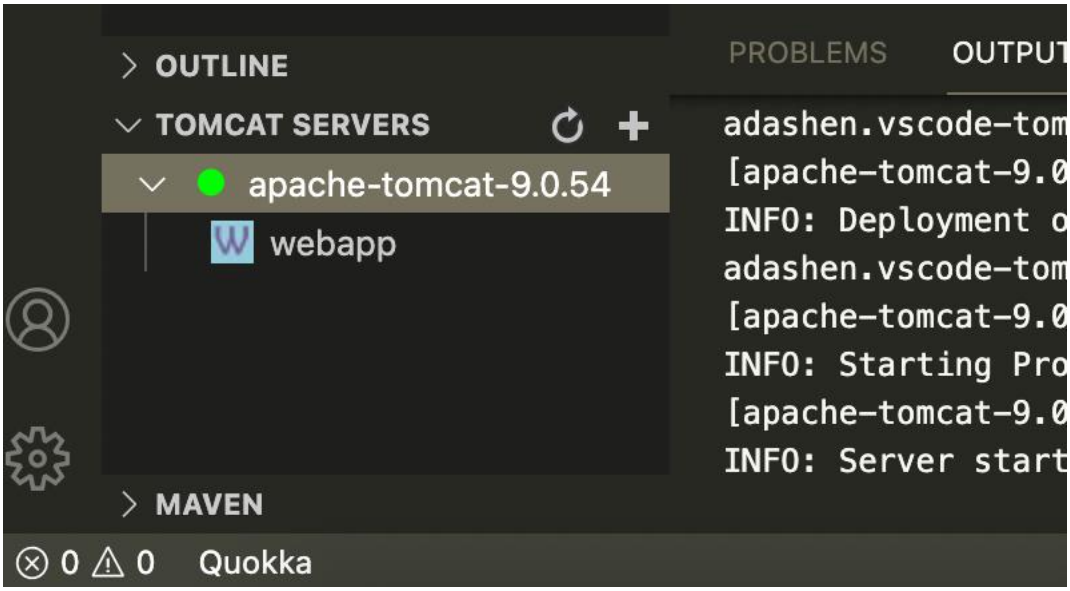
Add a new server, where the plus sign shows. Then you are going to locate the folder of Tomcat that you installed to your computer. It should look like this.



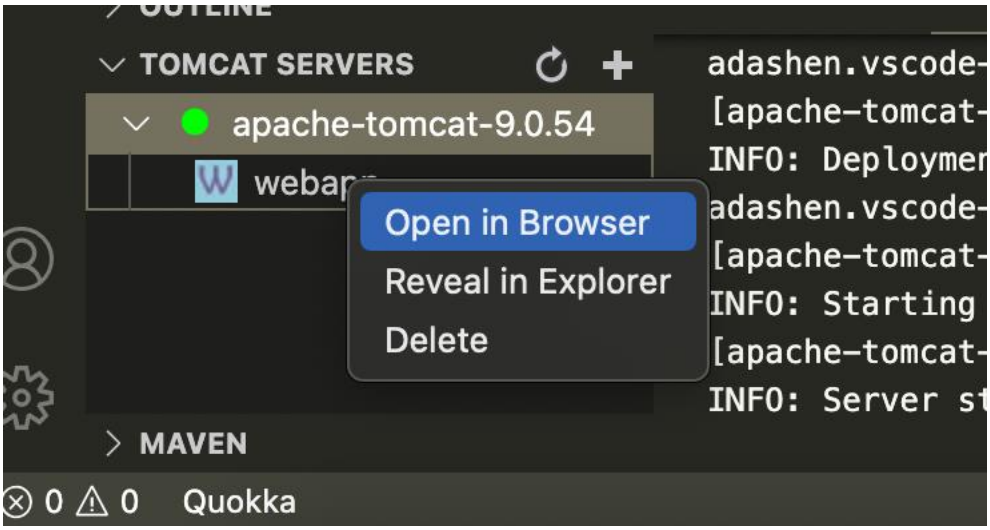
From here you must go to the folder that says “webapp” which is located on the project explorer on the left-hand side, and right click it. A side menu will pop up and you must select the one that says, “Run on Tomcat Server”. It should look like this.



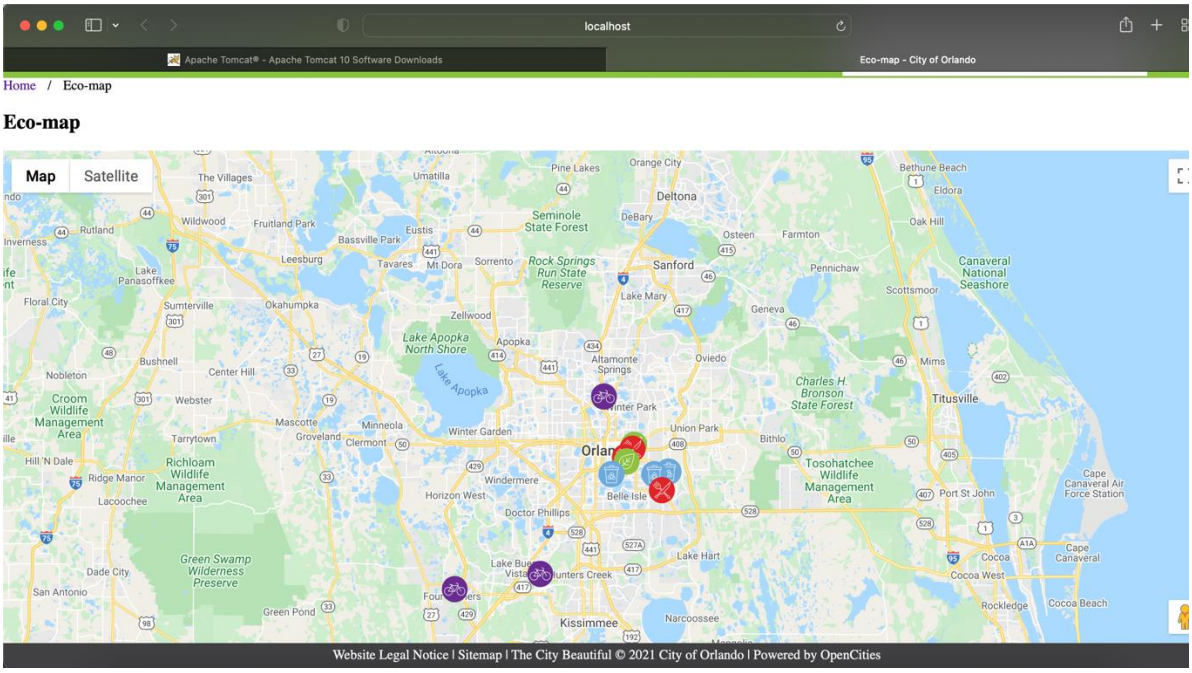
If done correctly, the red square on the Tomcat Servers section should turn green and it should show a new item under it that says “webapp” like this.



From here you can right click the webapp and open in a new browser.

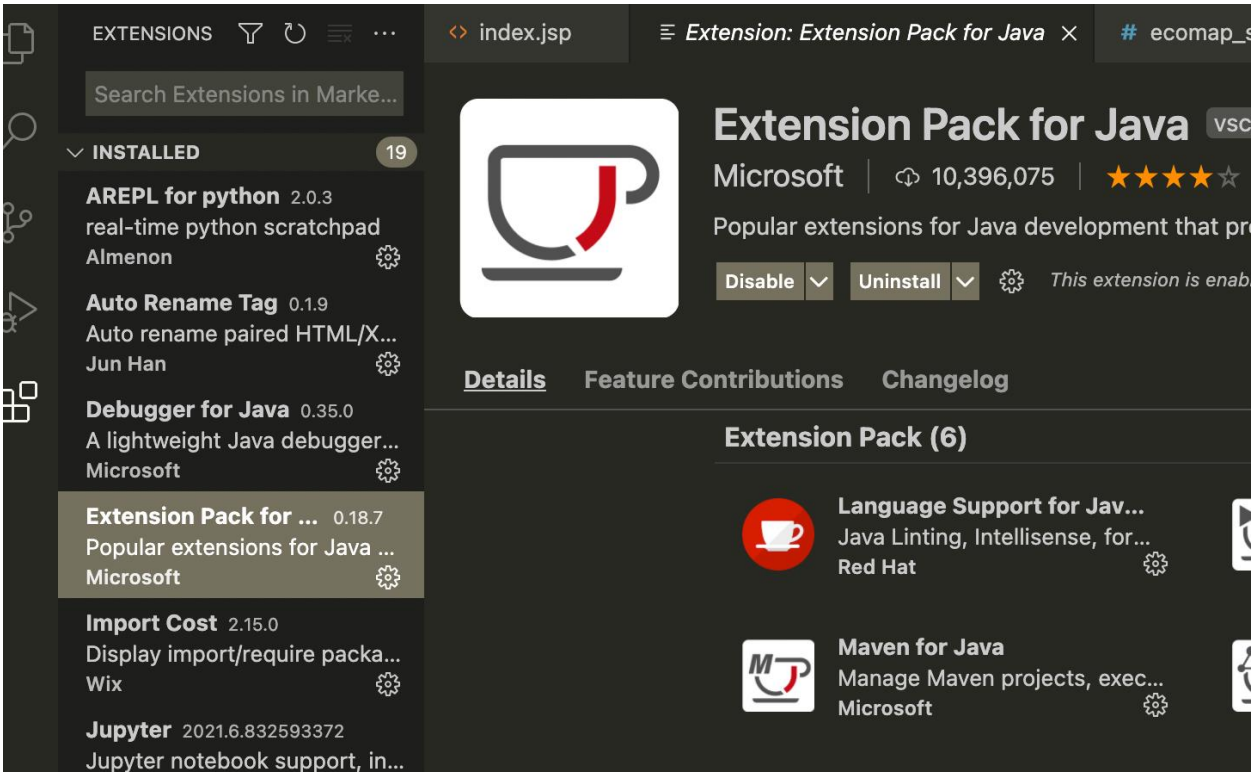


The result should be this! The port for the localhost is 8080.



Extension Pack for Java

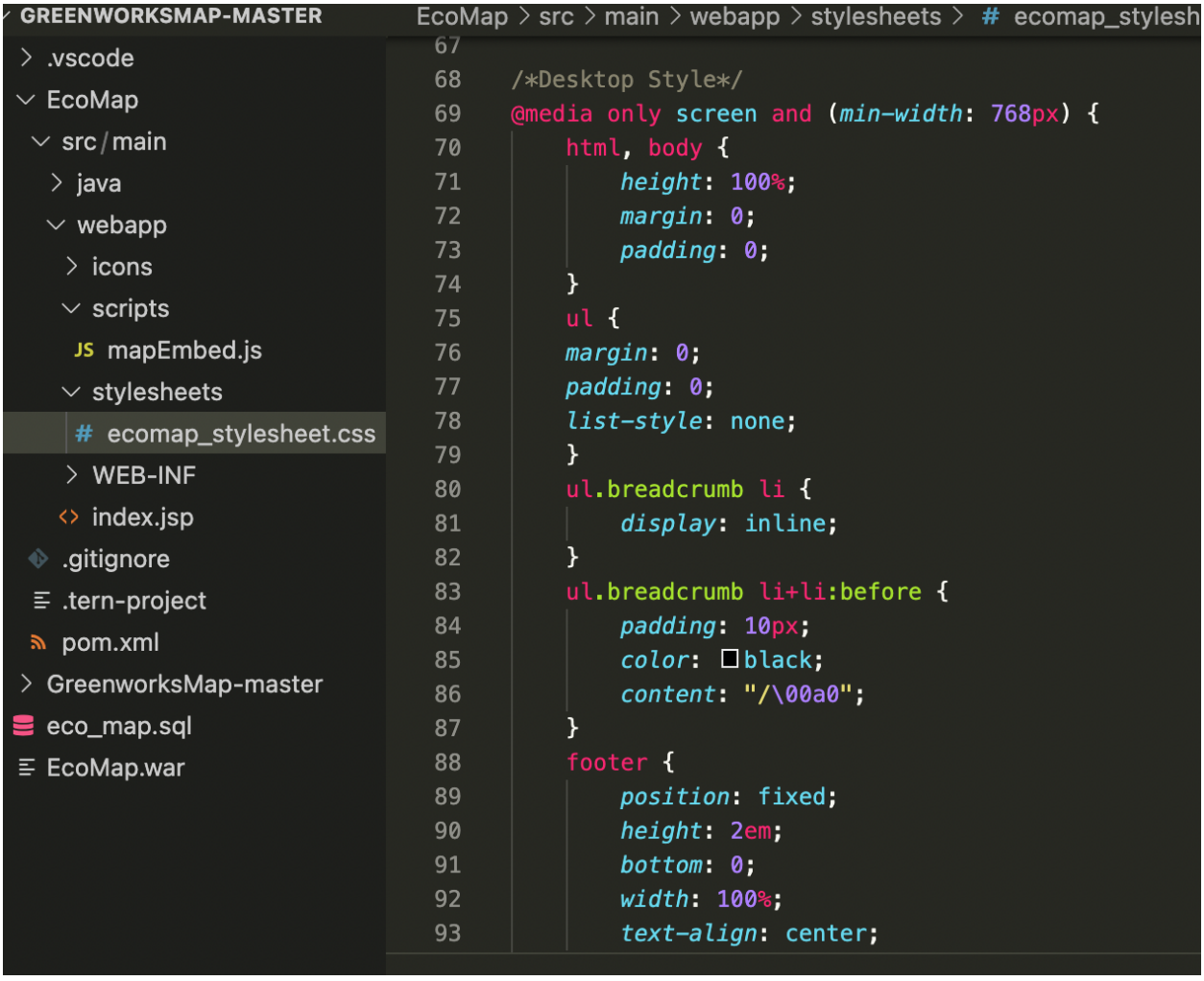
Another extension we used for this project is the extension pack for Java. You can find that under the extensions tab.

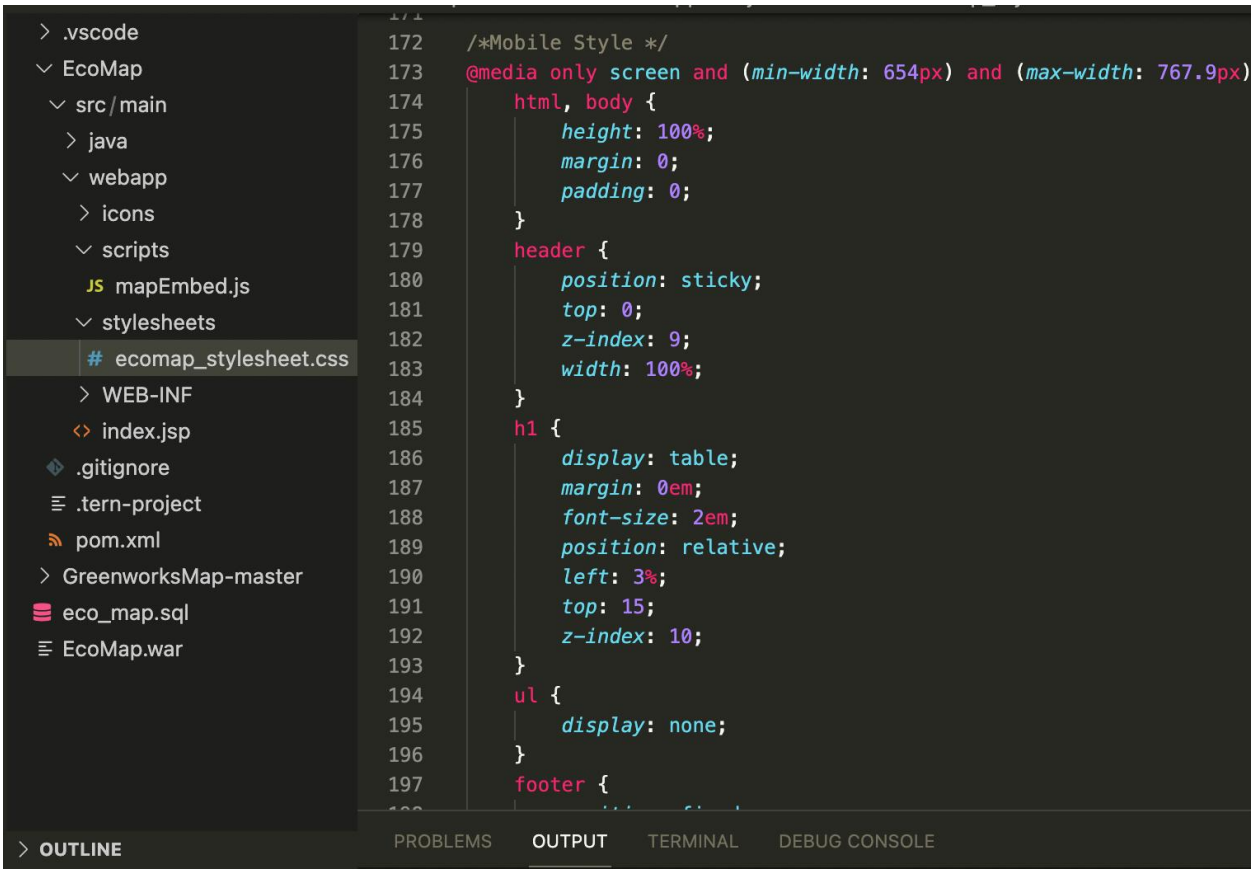


Installing this extension saves a lot of time for all of us that collaborated in this project. This allows us to debug, run test cases, and have project management. This extension is also compatible with Tomcat. It is a requirement to have this extension if wanting to use the Tomcat extension.

Styling

The styling that we are using is CSS. Our web application supports both desktop and mobile versions. In our code, you can see the separated sections for each.





A comment is made by typing the following for CSS and JavaScript: /* This is a comment */

Google Maps

Utilizing the JS SDK Embedded pre-set map that was available on their website.

The Maps Embed API overview

The Maps Embed API lets you place an interactive map, or Street View panorama on your web page with JavaScript.

You can set the Maps Embed API by typing the following into your JavaScript file.

```
//console.log(locations);
function initMap() {
    var orlando = {lat: 28.5384,lng:-81.3789};
    var settings = {
        zoom:10,
        center:orlando
    };
```

You can adjust the latitude, longitude, and the settings such as the zoom and center.

Add the following to the index.jsp file.

```
<div id="map"></div>
<!--sets the parameters for the Google Maps API embed -->

<script async defer
    src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD9nc7U6UPDjbOZUwgTn1pdDxBqX49zjI4&callback=initMap"
></script>
<!--script renders the map under the breadcrumb nav -->
```

The DOM element “map” is what allows us to reserve a spot for the map to display on the web page.

The script tag with the “async” asks the browser to asynchronously download and execute the script. When the script is executed, it will call the function specified using the callback parameter. Doing this will get you access to the Google Maps API.

There is no cost for this tool, with unlimited usage. For more information, a link to the Google Maps API Docs is provided here:

https://developers.google.com/maps/documentation

MySQL

The version this project is utilizing is MySQL Server 8.0.22 for windows x86_64.

To install MySQL, you must go to the following link:

https://www.mysql.com/downloads/

Connecting to and Disconnecting from the Server

When you run MySQL, you'll be prompted for a MySQL username and, most likely, a password to connect to the server. You must also specify a host name if the server runs on a machine other than the one where you log in.

```
$> mysql -h host -u user -p
Enter password: ********
```

The host name where your MySQL server is running, and the username of your MySQL account are represented by host and user. Replace the values with those that are appropriate for your configuration. When MySQL shows the Enter password: prompt, enter the ******** as your password.

If that works, you should see the following information followed by a mysql> prompt:

```
$> mysql -h host -u user -p
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25338 to server version: 8.0.27-standard

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

The mysql> prompt indicates that MySQL is ready for SQL commands to be entered.
If you're logging in on the same system as MySQL, you can skip the host and use the following instead:

```
$> mysql -u user -p
```

Some MySQL configurations allow users to connect to the server operating on the local host as the anonymous (unnamed) user. If this is the case on your PC, you should be able to connect to the server simply by typing mysql:

```
$> mysql
```

You can disconnect at any time after connecting successfully by typing QUIT (or q) at the mysql> prompt:

```
mysql> QUIT
Bye
```