HTML/CSS Project 1: Company Landing Page

# Part 1: Preparing your environment

Now that you have successfully broken down the project into smaller, more manageable bites, you can now move on to gathering resources for your project and set up your environment.

## **Github**

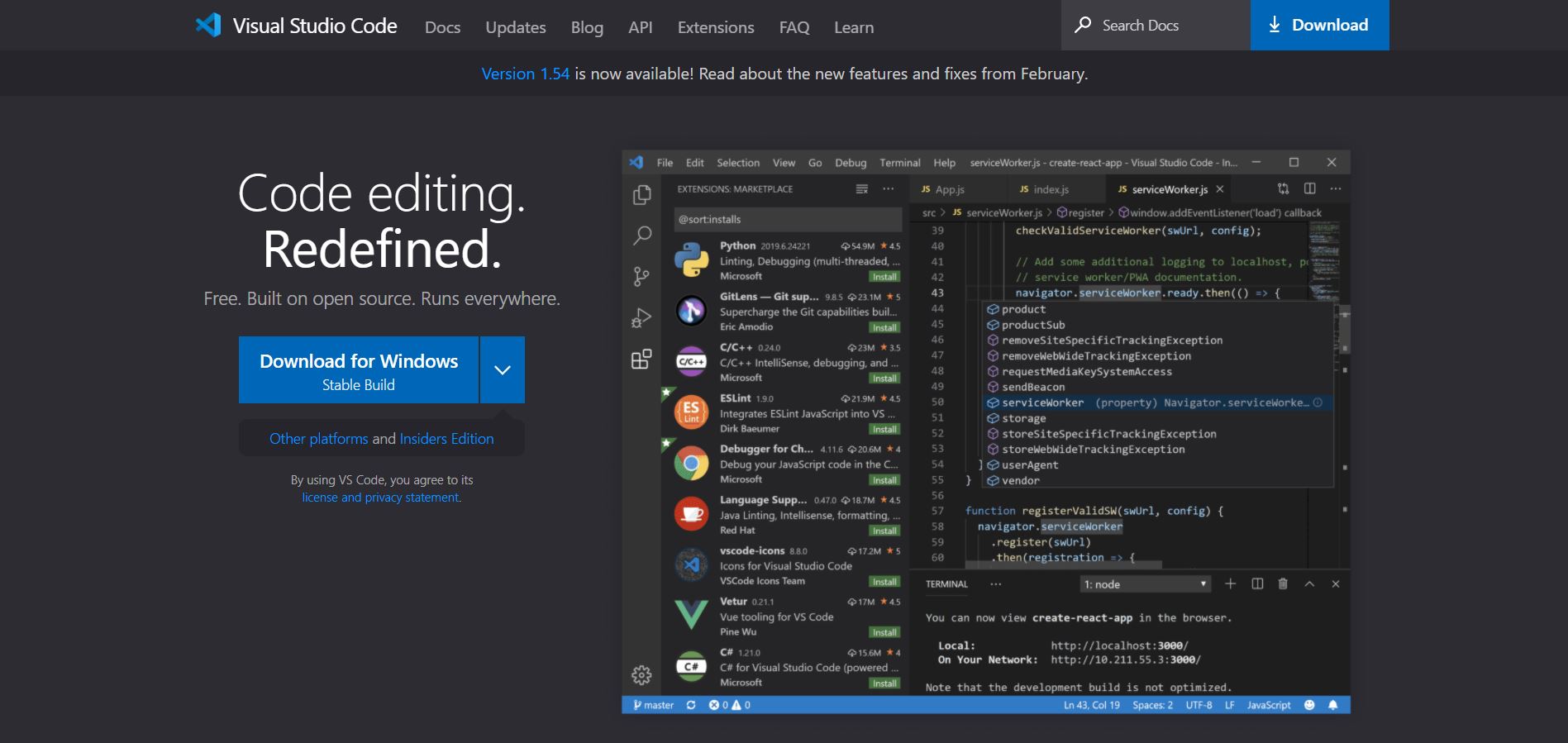
The first thing you will need to do is to create a [Github](https://github.com/) account, if you do not have one already. **Github** is a code hosting service that focuses around *version control*. Why this is useful to developers, is that if their local machine happens to crash, and the harddrive fails, the code will still be safe, being stored elsewhere.

It also allows you to create ***branches***, which is a way for us to keep our code and projects neat, by creating a separate ***branch*** which enables us to restore old code if we make a mistake, and lets us work on a single feature of our project at a time.

For these reasons, we will be using Github as a place to store and manage our code, whilst keeping our code safe from hardware issues.

## **Code Editor**

The next step is to ensure that we have a code editor available to us, so that we can start coding our project. The code editor of choice for this course is [Visual Studio Code](https://code.visualstudio.com/).

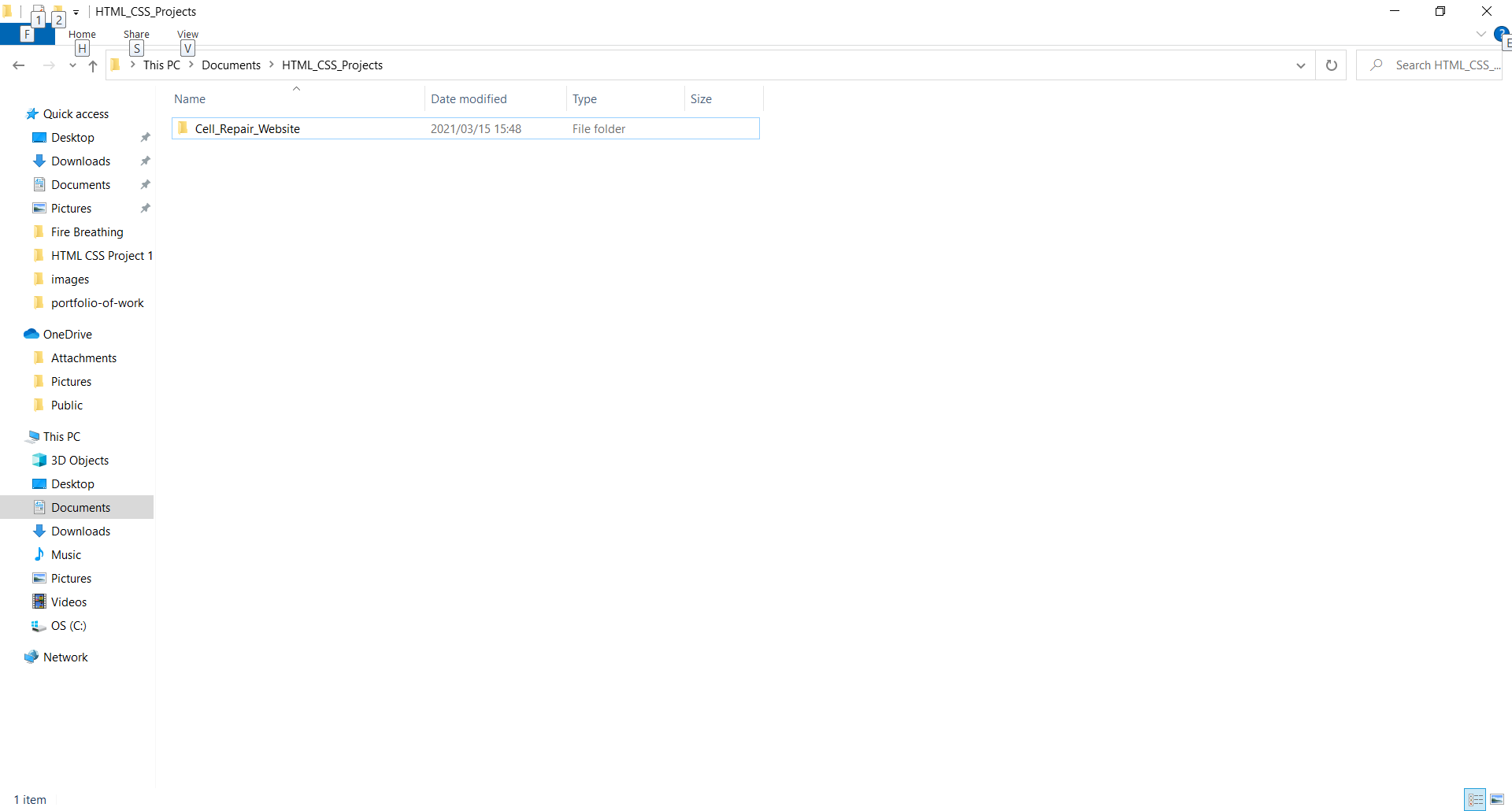


Download the installation and install VS Code on your machine. Make sure you enable the option to add VS Code to your **PATH** variable. This allows us to open folders in VS Code by right clicking.

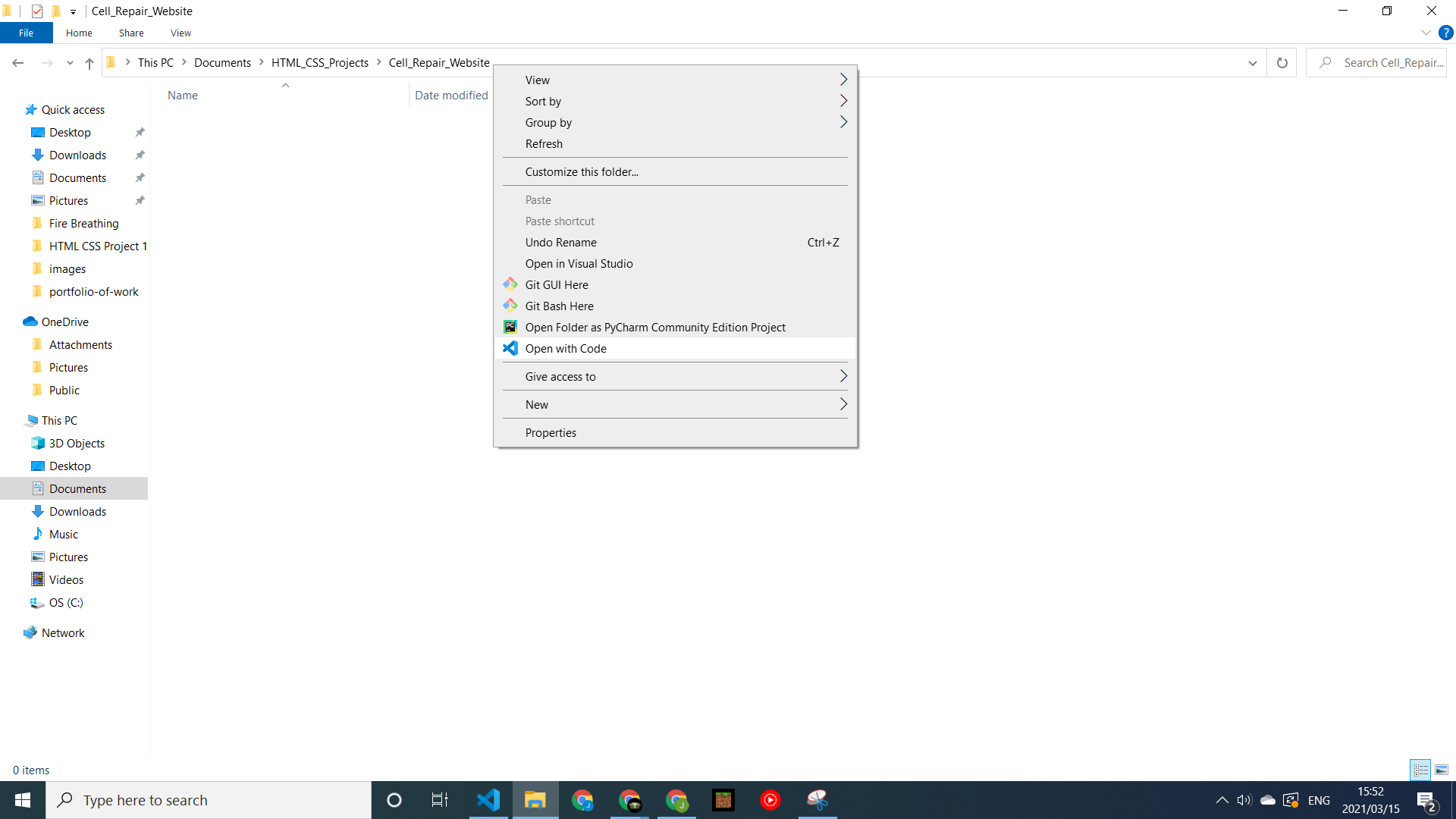
## **Initial Folder Structure**

In your **Documents** folder, create a new folder called **HTML\_CSS\_Projects**, then enter it.

Next, you will need to create a new folder called **Cell\_Repair\_Website.**

****

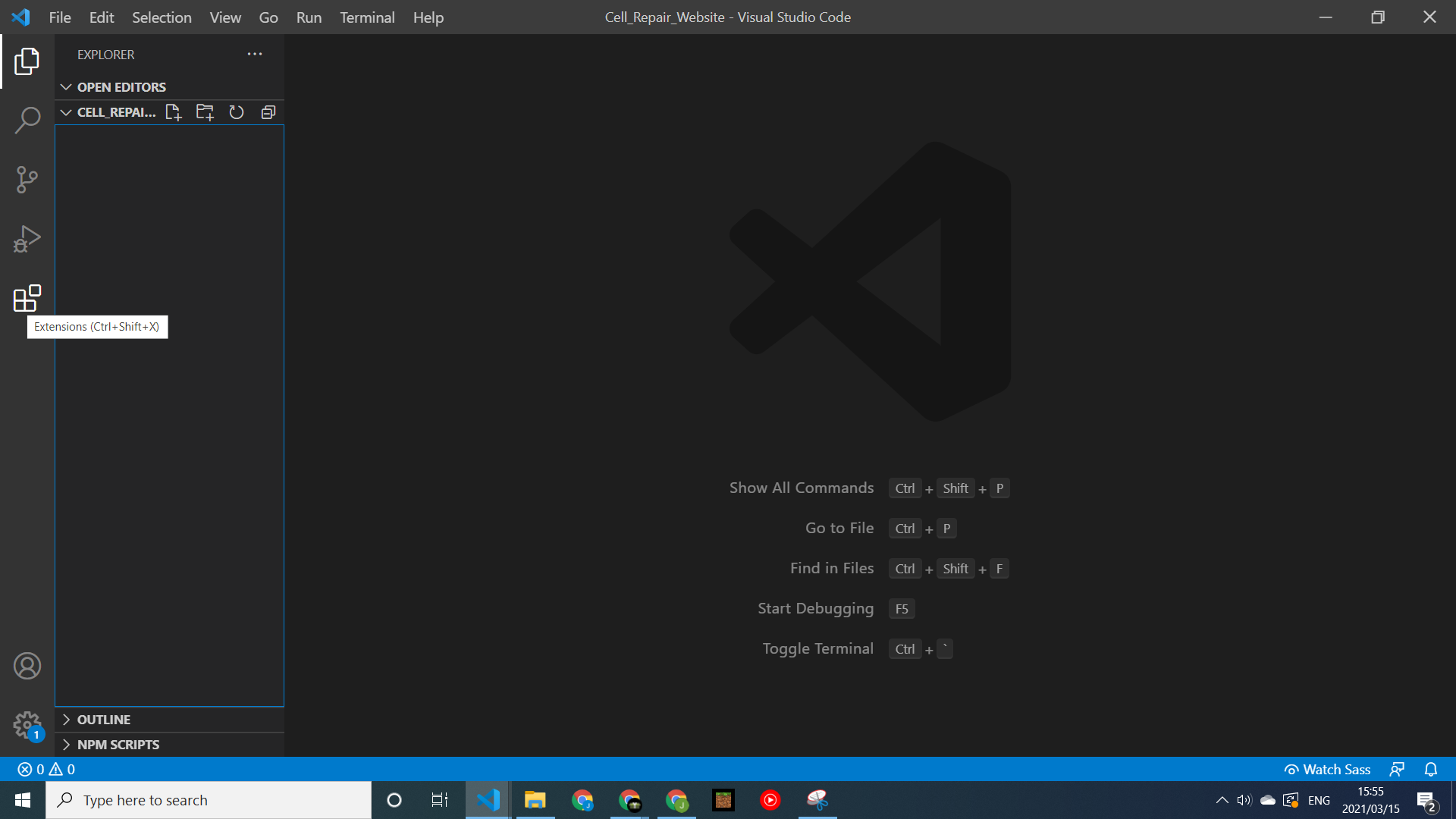
Next, go into that folder, right click, and select **Open with Code:**



Congratulations! You have just opened up your first project in the code editor!

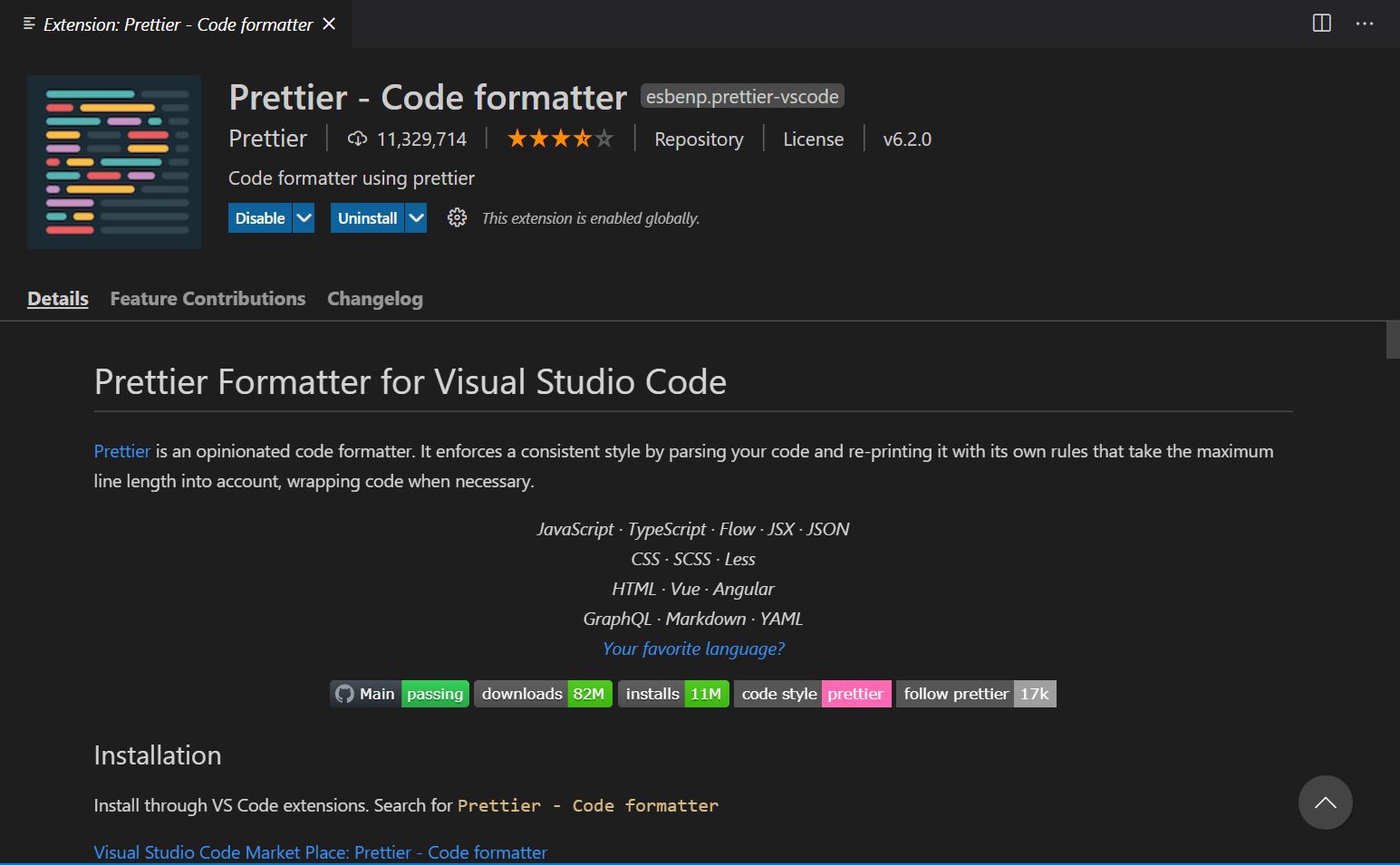
## **Text Editor Plugins**

The final step before we can start to code, is to ensure that our VS Code has been set up to help us code more efficiently. For this, we will need to go to the **extensions** section of VS Code:



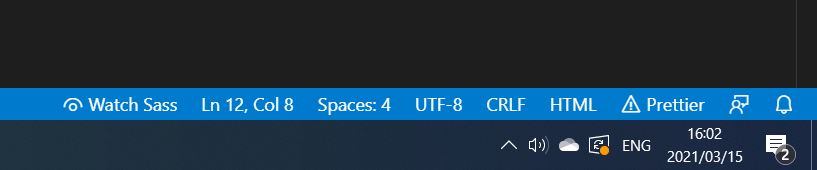
For now, we are only going to download 4 different extensions:

1. **Prettier**

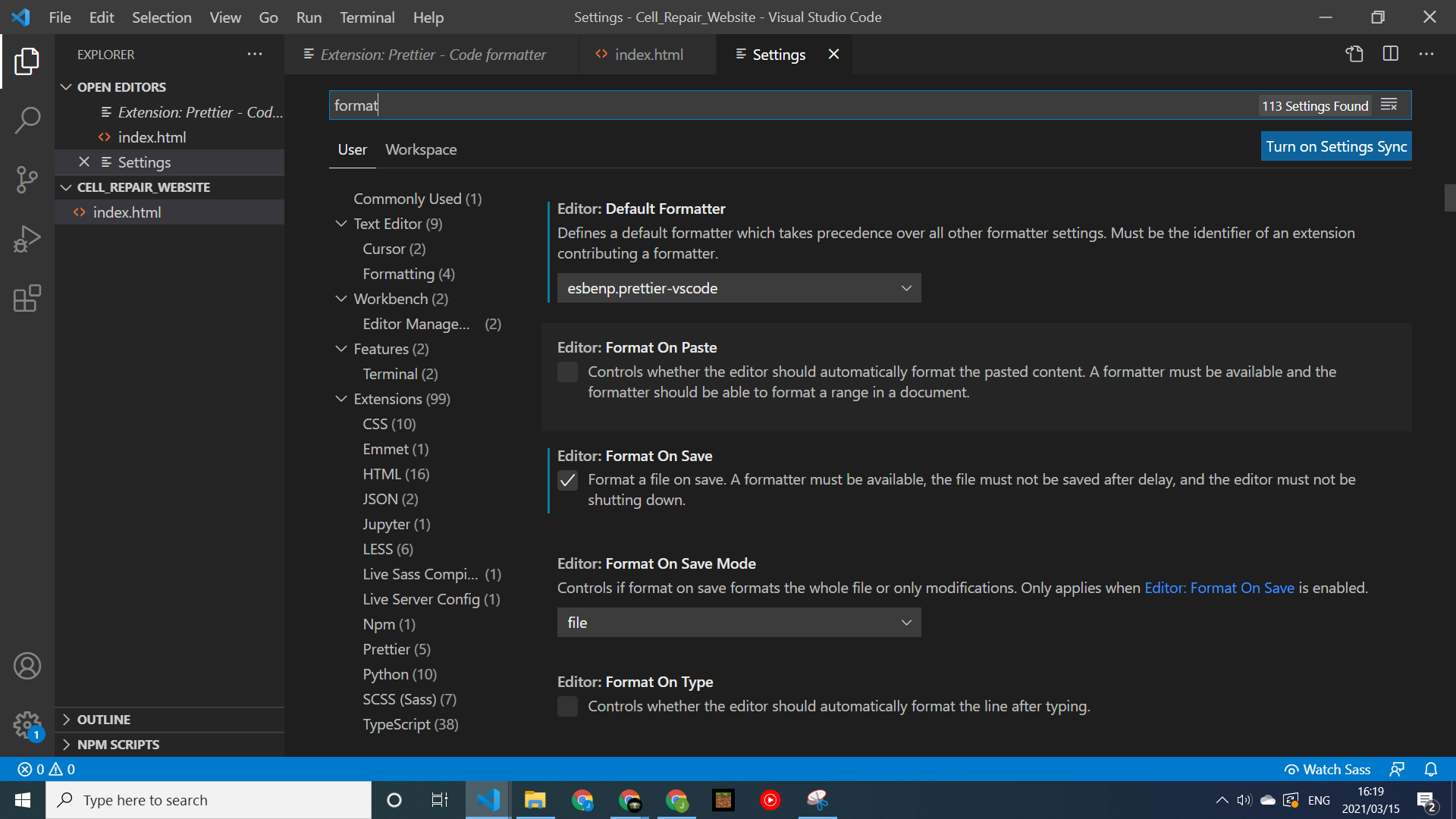


This extension (when set up correctly) will automatically indent and neaten our code, making it easier to read and navigate.

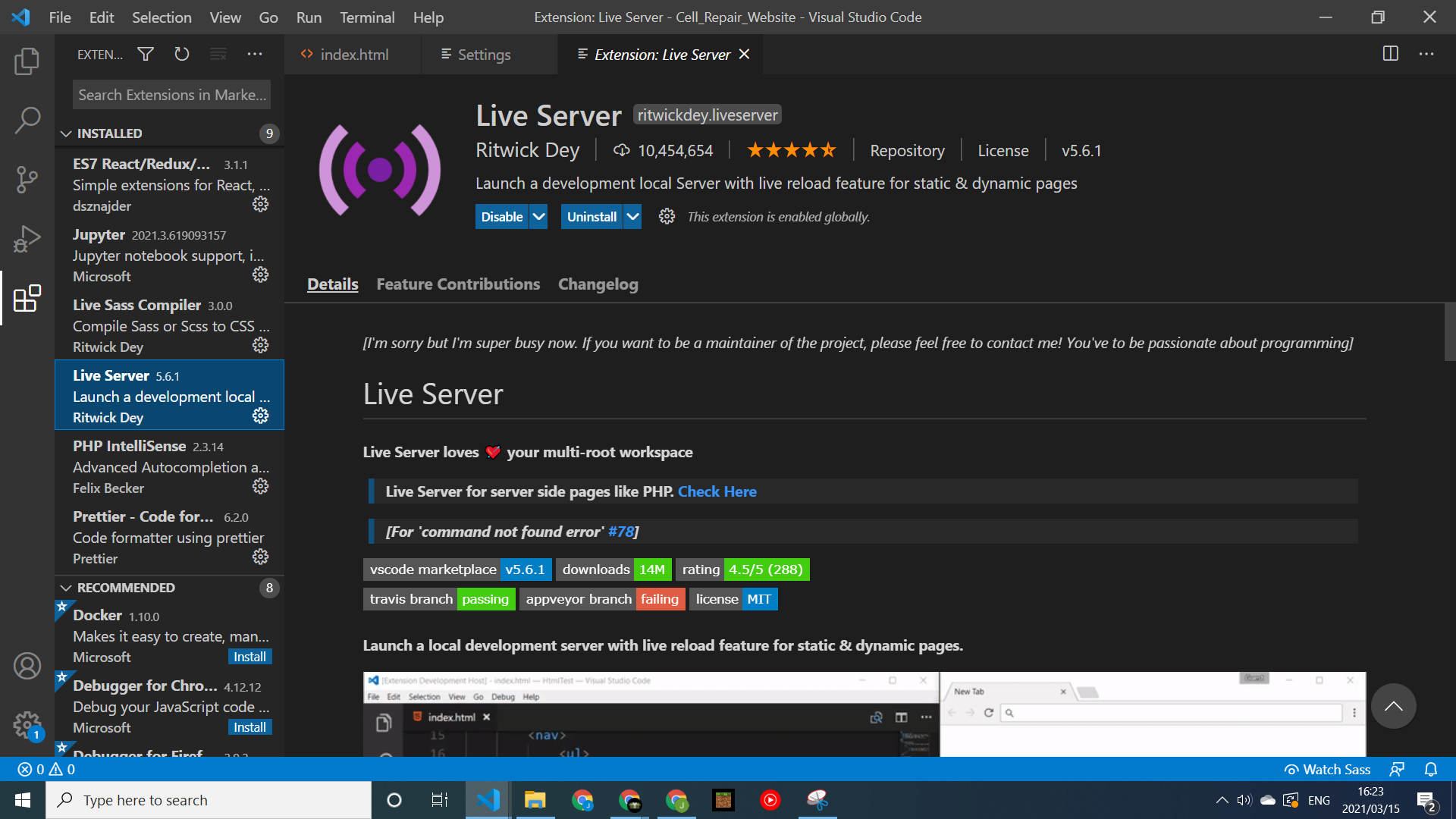
***As a tip, if you set the plugin to format your document on save, and after you save your code does not neaten up, then there is an error in how you have structured your code. You can get assistance in seeing where the issue is, by clicking on the warning icon with prettier, at the bottom of your code editor:***



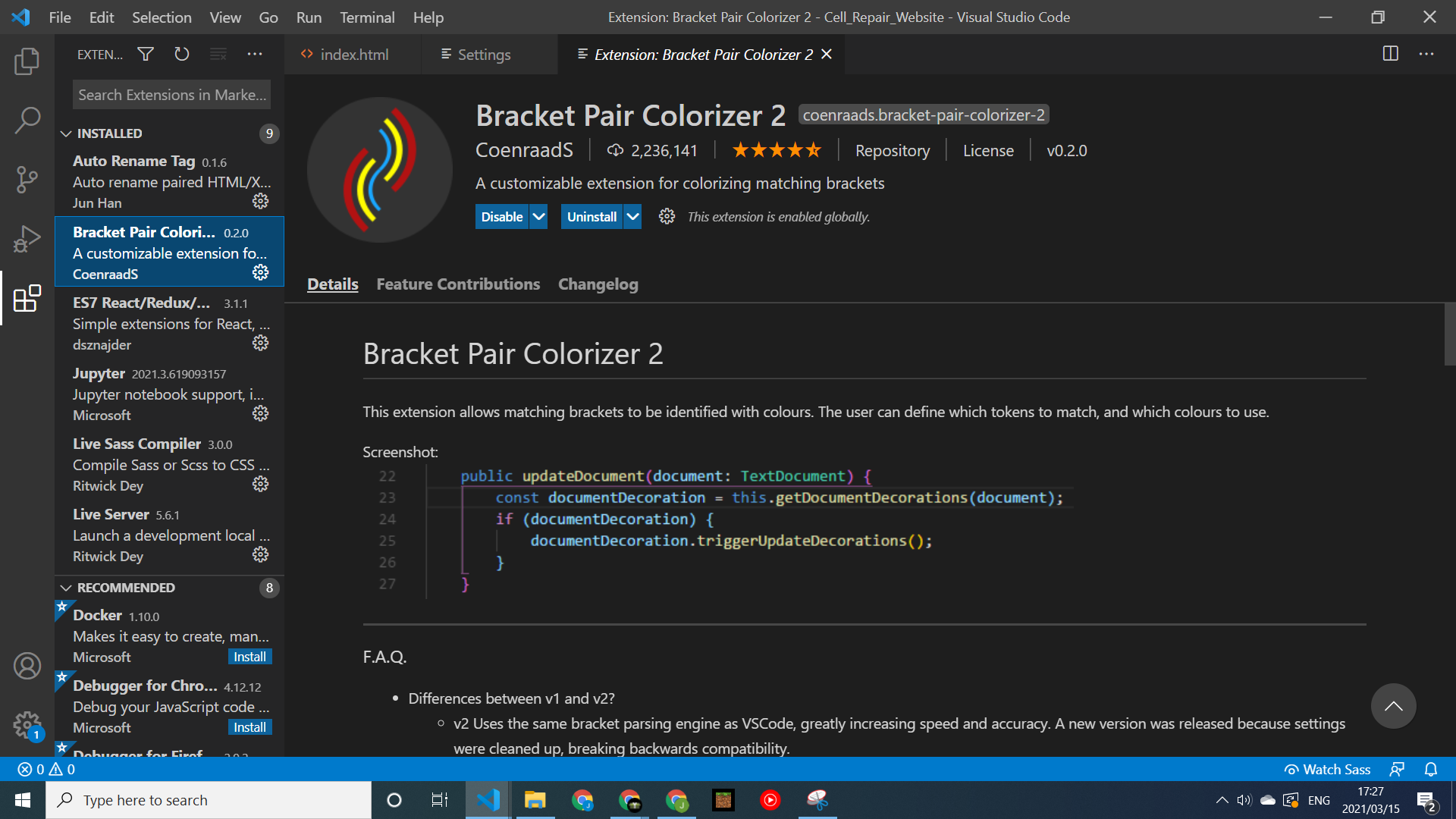
To set up *prettier*, you will need to go to the **settings** tab (bottom left of VS Code). Search for *format*, and make sure the following settings are enabled in the workspace:



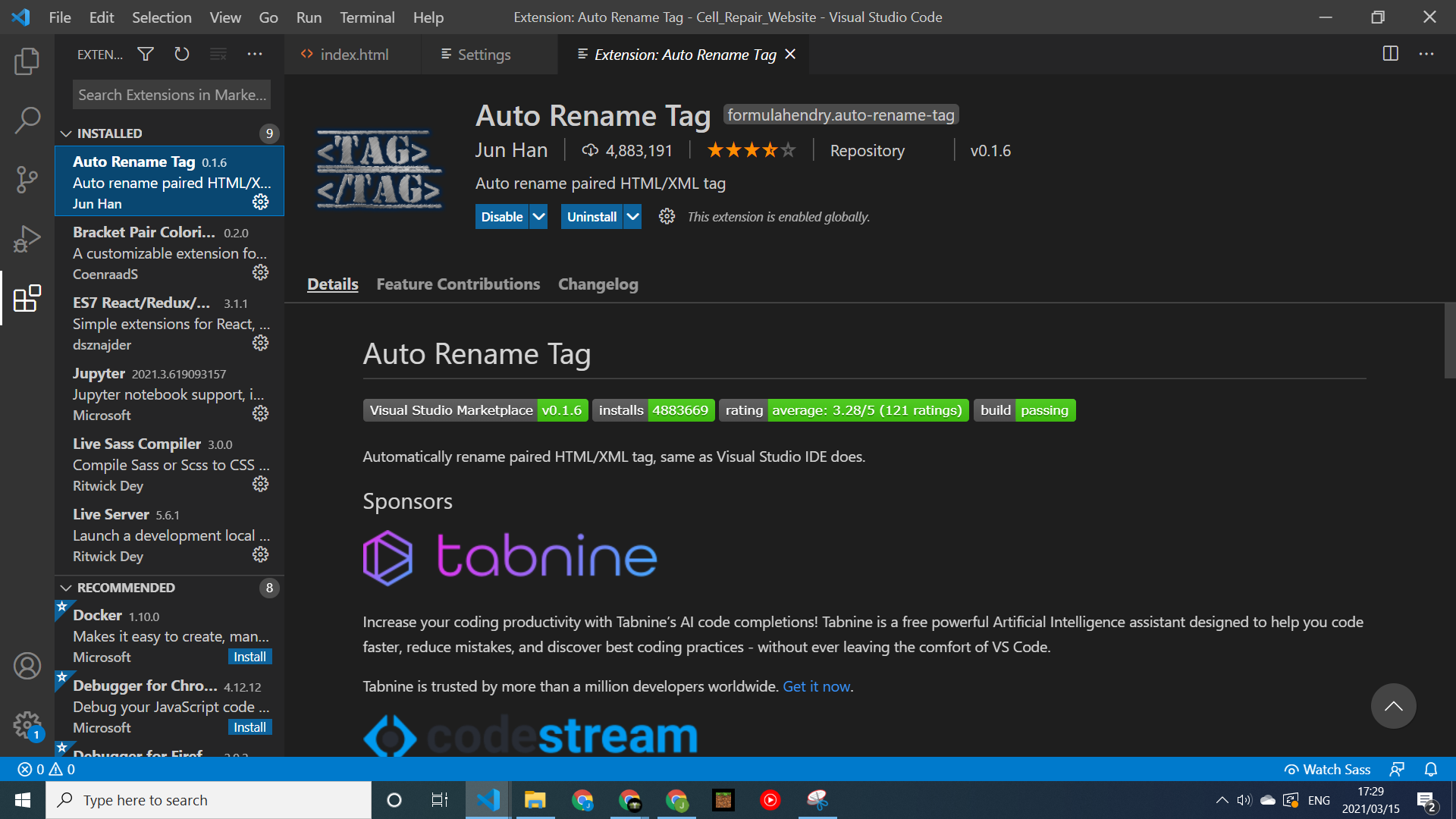
The first setting lets VS Code know how we want to format our code, and the third setting allows us to format our code any time we save our project.

1. **Live Server**

**Live Server** hosts the project on your machine, so that when you make changes to the project, it is automatically updated in your browser. This helps us so that we don’t have to refresh your browser to see the changes you have made in your code.

1. **Bracket Pair Colorizer**

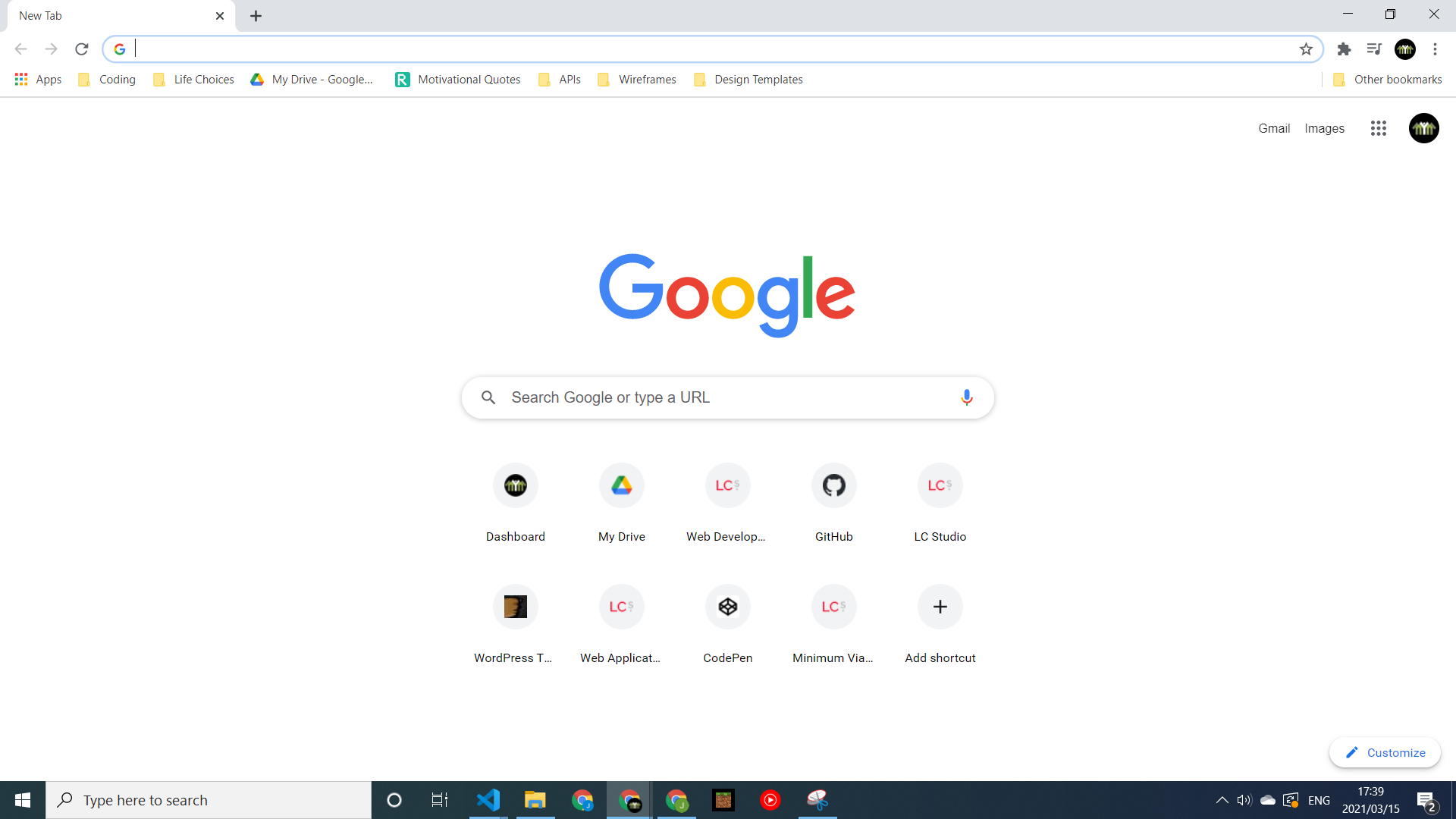
This extension color codes your brackets, so that you can read code more easily, and see where your brackets open and close. This helps you keep track of what your code is doing.

1. **Auto Rename Tag**

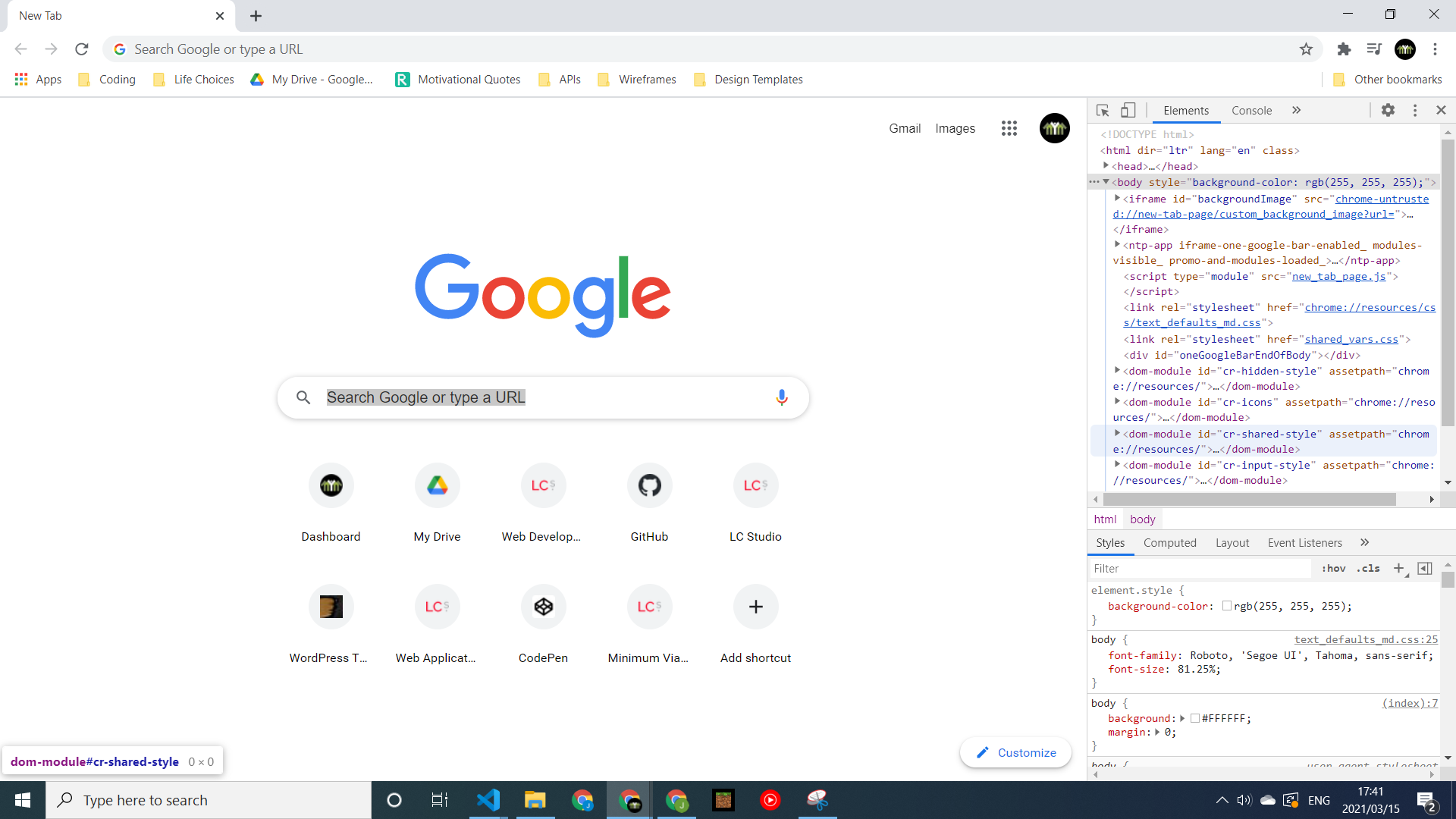
This extension helps us keep our tag names consistent. If you rename an opening HTML tag, it will automatically rename the closing tag, and vice versa.

## The Browser

The last piece of software needed is your web browser. The preferred browser for this course is **Google Chrome**, and all examples shown will be taken using Chrome:



The browser is where we will see how our application will look to everyone else on the web. This is where we also do the bulk of our testing and tweaking of our project, before we make any hard changes. The way we do this is with the **developer console** of the browser.



This lets us see what is going on under the hood of our browser. It allows us to edit and tweak what we see and do on the browser, without making any changes to your actual code, so you can feel free to read around and try to break the code. As soon as you refresh the tab, the page code will reset, and nothing is saved in your code.

*If this code looks strange now, that is perfectly fine. Over the next 5 projects, you will grasp the ins and outs of HTML and CSS!*

## **Task 1**: Exploring the developer tools

Now that we have our environment set up to start coding, we should explore our tools before diving into the project. Go to **Google Chrome** and open up the **developer console**. In **groups of 3**, read through some of the code that is shown, and see if you can decipher any patterns.

**Items to discuss:**

* Try to differentiate between which code is for content(HTML), and which is for style(CSS)
* After differentiating between content and styling, try to change the color of any component in the browser
* Change some content in between the tags in the HTML section, and see how it affects what is shown in the browser
* Refresh the browser and note the changes