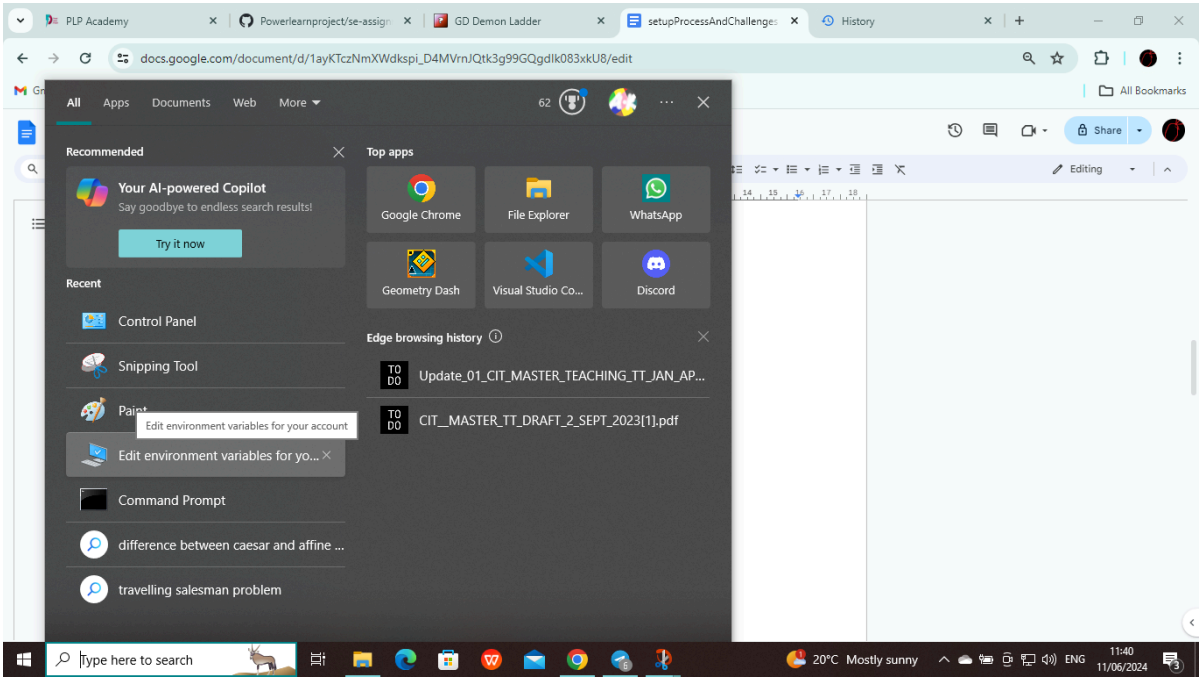
The development environment setup process

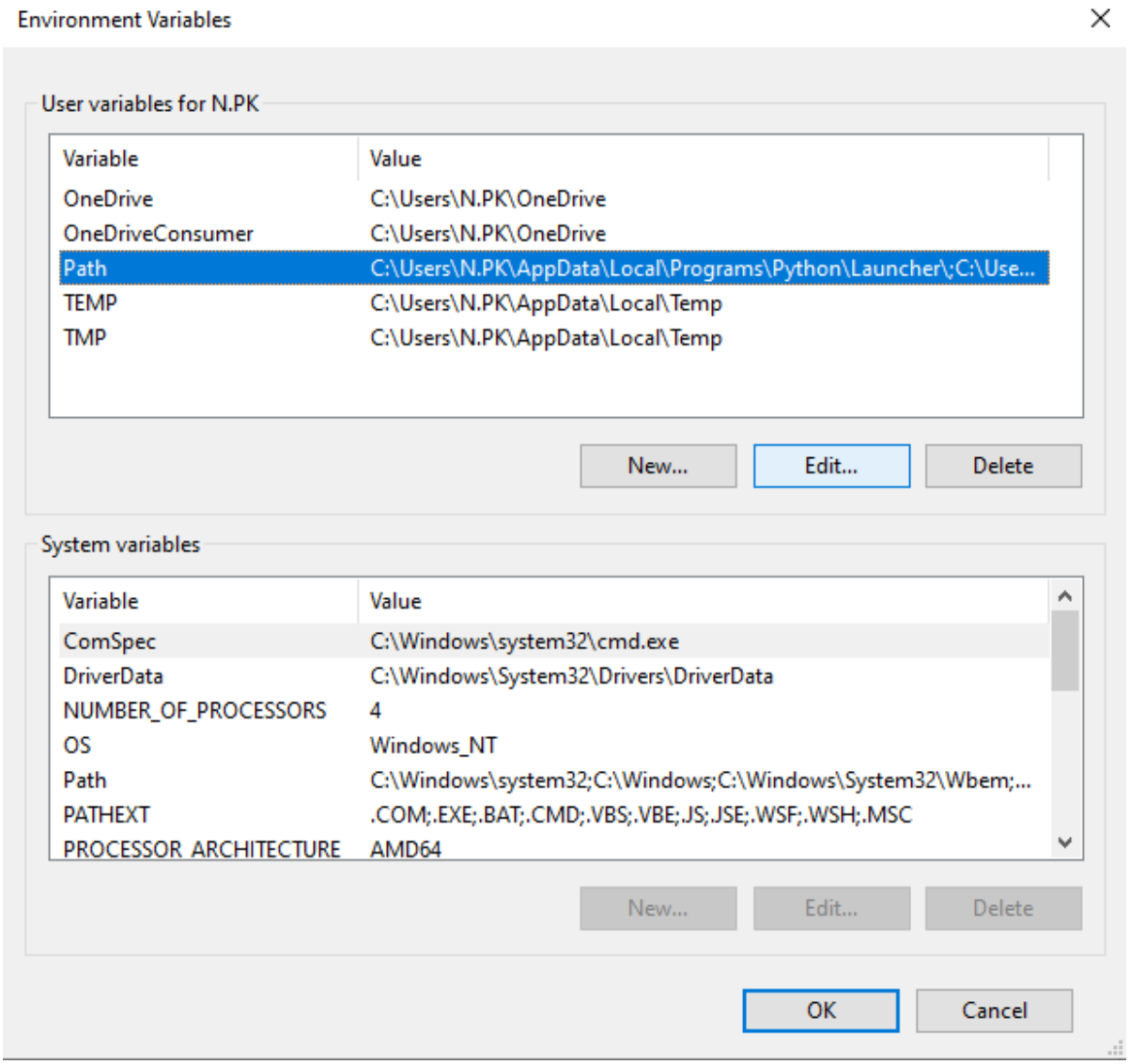
Most, if not all tools and configurations that are to be installed, have a similar installation process. Thus, I will give a general procedure on how to install and configure said tools and configurations. This procedure is what I used when configuring VS Code, MySQL, Python, Dart and Flutter.

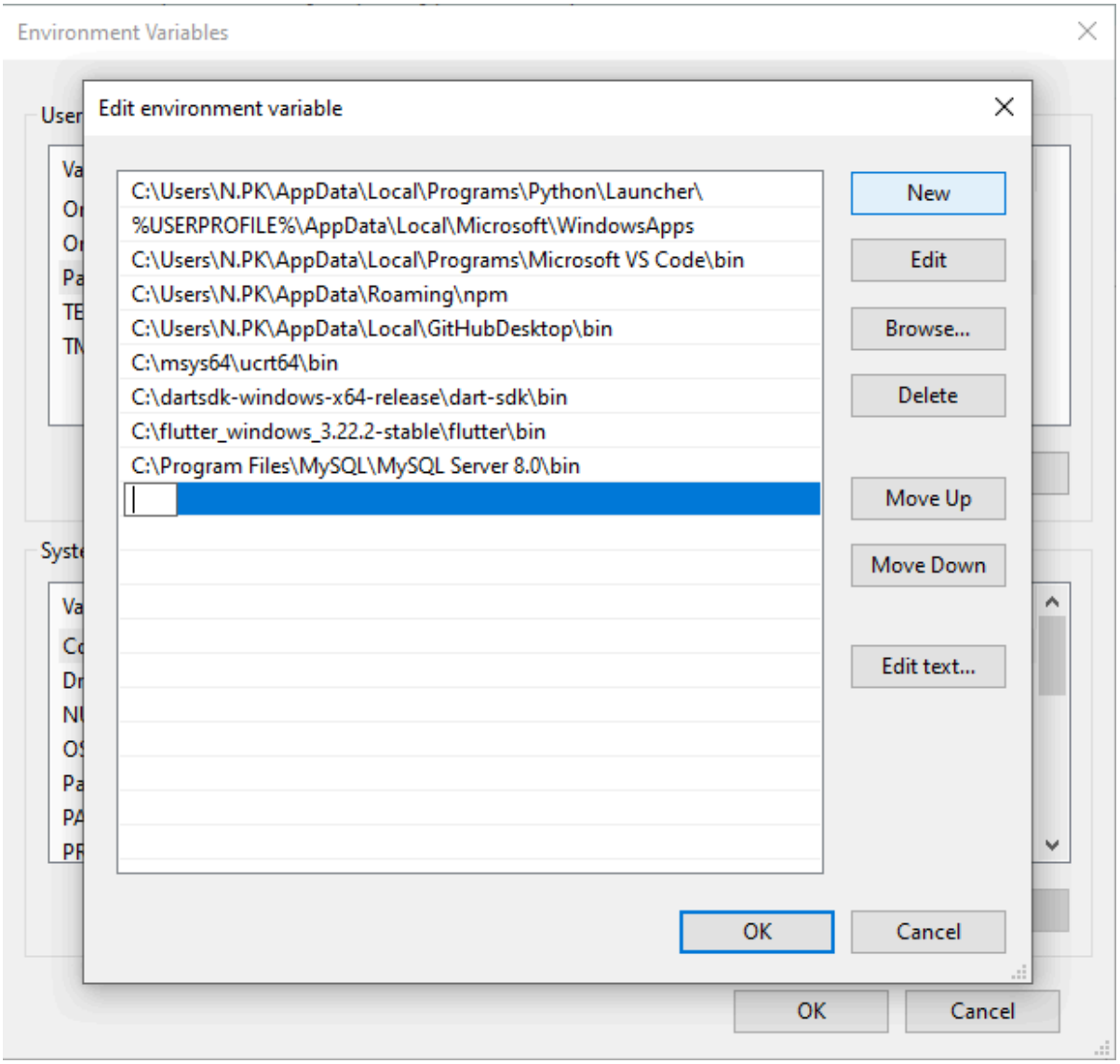
- I visited the official download website for said tool. These are the official download websites for VS Code, Git for Windows, GitHub Desktop,Python, MySQL, Dart and Flutter.
- I selected the operating system I am currently using and the architecture of my processor. I carefully read any instructions before progressing. For example, on the Flutter website, you are instructed to not put the downloaded zip file or extracted file in a directory whose name has a space or special character. I also ensured that my computer met the minimum requirements of the tool.
- I then clicked on the download button and waited for either the installer or ZIP file to be downloaded. I checked the progress of downloading on the 'Downloads' page of the browser I was using(Google Chrome).
- If the tool being downloaded had an installer(e.g. MySQL), I followed the steps provided by the installer and allowed any settings that the computer may have prompted me to allow. I then configured the app being installed to my desired requirements. The program folder was put inside the 'Program Files' directory in the local disk (C:).
- If the tool being downloaded was in a ZIP folder(e.g. Dart), then, I navigated to the 'Downloads' folder where the ZIP folder is usually placed once downloading is complete. I cut the ZIP folder and pasted it in the local disk folder. I ensured that I had a ZIP file/folder extractor. I personally used BreeZIP(free), but you can use others such as 7-ZIP(free) and WinRAR(monthly subscription). I then used the 'Extract Here' option to extract the folder in the same directory.
- I opened the extracted folder and redirected to the `bin` folder. I then copied the entire path. For example, in the case of Dart, this was the

path that I copied:
`C:\dartsdk-windows-x64-release\dart-sdk\bin`

- Using my computer's search bar, I searched for 'Environment Variables', then clicked on 'Edit Environment Variables'. I clicked on 'Path' in User Variables, then 'Edit', but you can also double-click 'Path'. I then clicked on 'New' to add a new path. I pasted the path I copied that led to the `bin` folder, then clicked 'OK'. I clicked 'OK' again on the initial popup window to ensure changes are saved. I had now installed the tool that I required.




Environment Variables
User variables for N.PK
Variable
Value
OneDrive
C:\Users\N.PK\OneDrive
OneDriveConsumer
C:\Users\N.PK\OneDrive
Path
C:\Users\N.PK\AppData\Local\Programs\Python\Launcher\;C:\Use...
TEMP
C:\Users\N.PK\AppData\Local\Temp
TMP
C:\Users\N.PK\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ComSpec
C:\Windows\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS
4
OS
Windows_NT
Path
C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE
AMD64
New...
Edit...
Delete
OK
Cancel



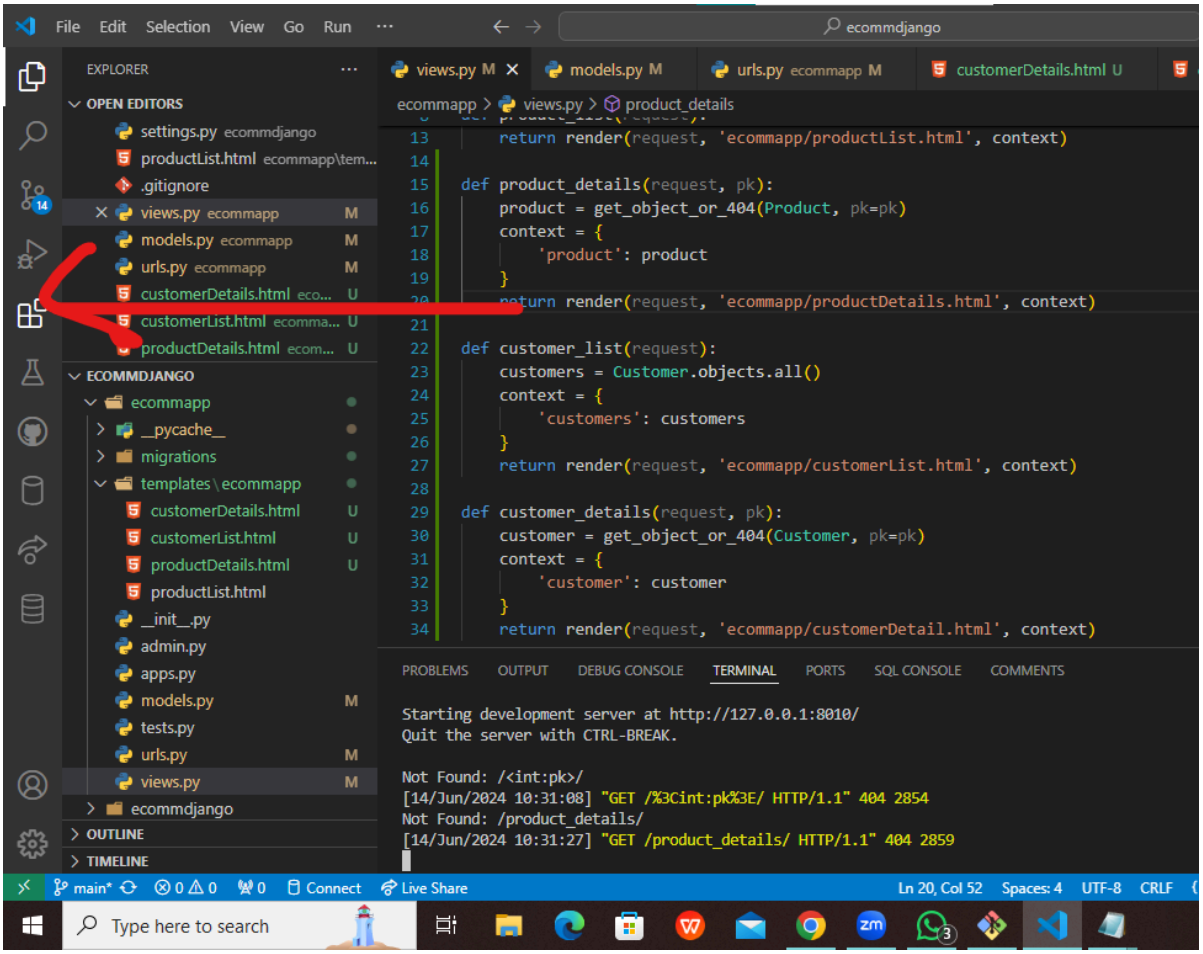
- To ensure that what I was installing has been installed correctly, I ran `tool –version` on either my Git Bash terminal, Command Line interface or Windows Powershell. You can use Kali Linux as well.

```
N.PK@DESKTOP-L1EPCE6 MINGW64 ~ (master)
$ mysql --version
C:\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe  Ver 8.0.37 for Win64 on x86_64 (MySQL Community Server - GPL)

N.PK@DESKTOP-L1EPCE6 MINGW64 ~ (master)
$
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\N.PK> dart --version
Dart SDK version: 3.4.3 (stable) (Tue Jun 4 19:51:39 2024 +0000) on "windows_x64"
PS C:\Users\N.PK>
```

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\N.PK>flutter --version
Flutter 3.22.2 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 761747bfc5 (6 days ago) • 2024-06-05 22:15:13 +0200
Engine • revision edd8546116
Tools • Dart 3.4.3 • DevTools 2.34.3

C:\Users\N.PK>
```

- In VS Code, I downloaded various extensions to aid me in my coding experience. These extensions are in the 'Extensions' window shown below.

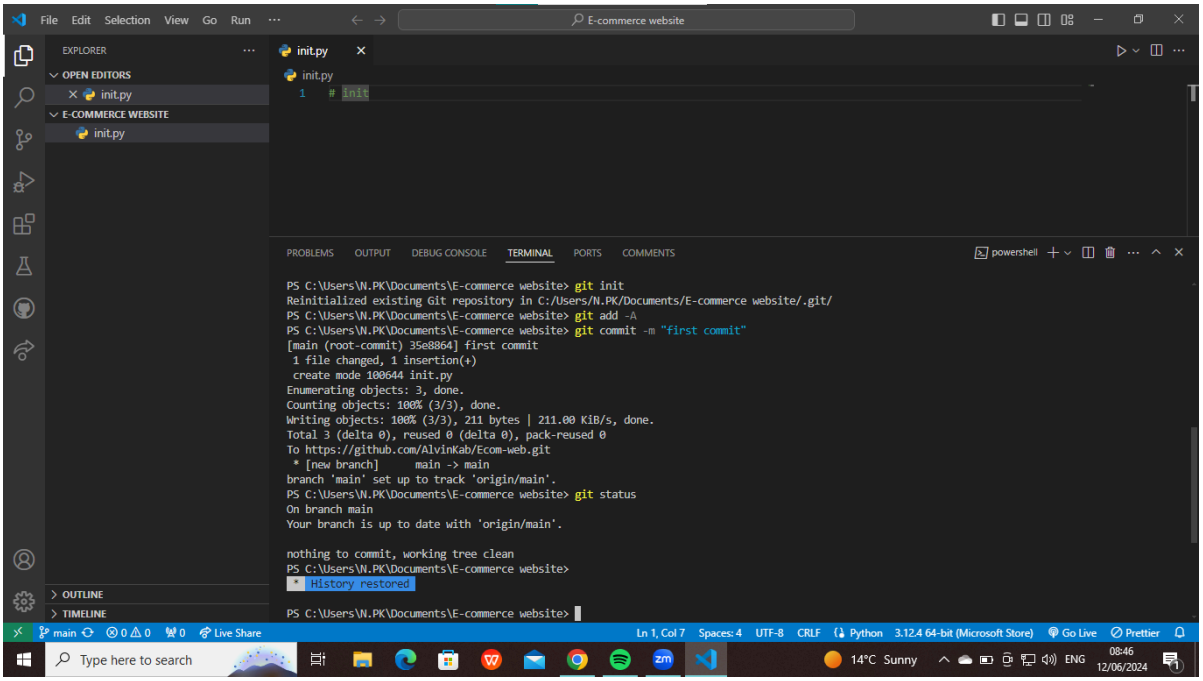


- Useful extensions that I downloaded include:
  - C/C++ Extension Pack
  - CMake, CMake Tools
  - Code Runner
  - Dart
  - Flutter
  - Flutter widget snippets
  - Jinja template support
  - Jupyter, Jupyter cell tags, Jupyter keymap, Jupyter slideshow, Jupyter Notebook
  - Live Server
  - Live Share
  - Multiple cursor case preserve
  - Prettier
  - Python, Pylance
  - SQLite Viewer
  - TailwindCSS IntelliSense

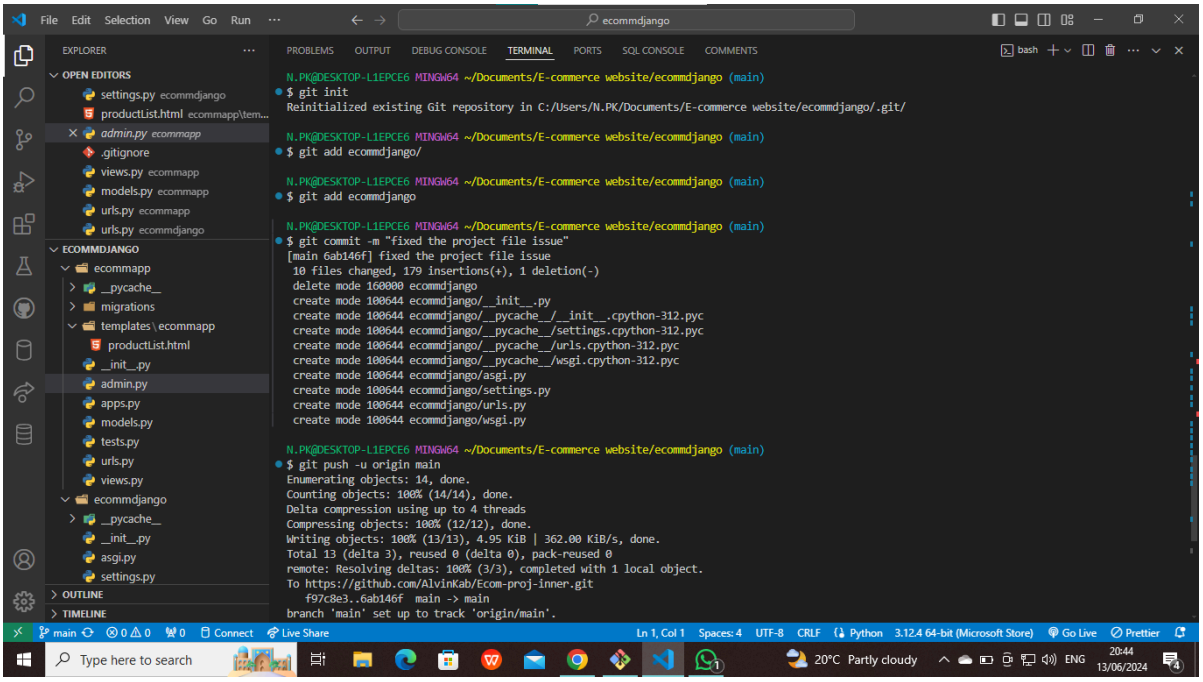
- vscode-icons

Sample project initialised using Git

The GitHub repository where this assignment was submitted through was initialised in a similar way to what is shown in the following screenshot:



The aforementioned repository should contain the sample project folder. If it does not, or the project folder cannot load, then use this repository to access the project:
https://github.com/AlvinKab/Ecom-proj-inner
This repository was initialised in a similar way to what is shown below:



Please note that in both cases, during the initial initialisation, `git branch -M main` was used to change the branch from master to main. `git remote add origin [repo link]` was used to connect the local repository to the remote GitHub repository before pushing.

Challenges faced and how I overcame them

Fortunately, I only faced one major challenge during installation, and it was the installation of MySQL. I first installed MySQL last July. I installed MySQL Server 5.7 and Shell 8.0. However, when I ran `mysql –version` on Git Bash, `mysql` was not recognised as a command, even though I had done all the processes above. Redownloading the installer didn't help either. To fix this, I searched 'How to safely uninstall MySQL' on Google. One of the top results directed me to this website. After following the procedures provided, I reinstalled and configured MySQL using the steps above, but I installed Server 8.0 to prevent more issues from popping up. Now running `mysql –version` gives me the correct version(8.0.37).