

NAME: MOHAMMED MUSA BODLE

SE-ASSIGNMENT-5

Installation and Navigation of Visual Studio Code (VS Code)

Instructions:

Answer the following questions based on your understanding of the installation and navigation of Visual Studio Code (VS Code). Provide detailed explanations and examples where appropriate.

**NAME: MOHAMMED MUSA BODLE**

Questions:

**1. Installation of VS Code:**

**Describe the steps to download and install Visual Studio Code on Windows 11 operating system. Include any prerequisites that might be needed.**

Below are the procedures or steps on how to download to and install Visual Studio Code on a Windows 11 operating system, follow;

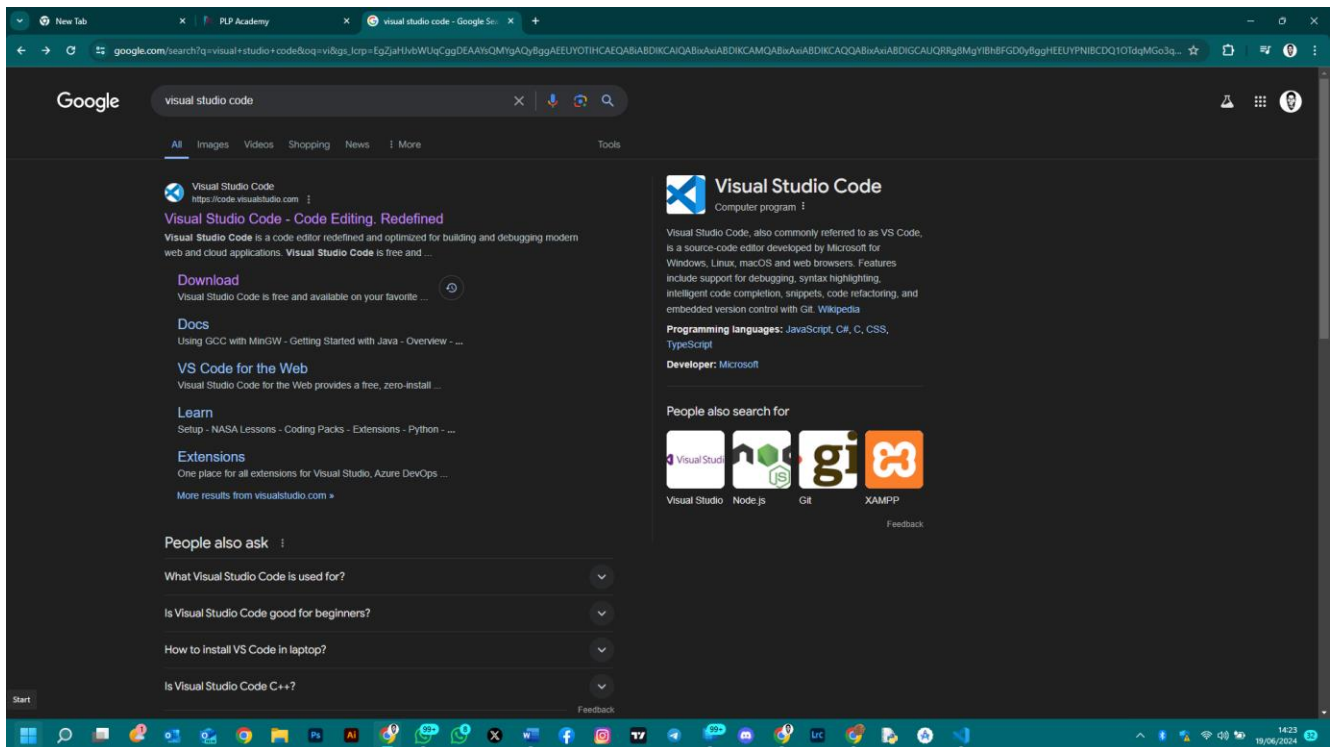
- a) Internet Connection: I ensured i had a stable internet connection to download the Visual Studio Code installation file.
- b) Minimum System Requirements: I made sure my system had met the minimum requirements to run Visual Studio Code.

**Steps and screenshots on how to Install Visual Studio Code on Windows 11:**

1. Download the Installer:

I googled to the official Visual Studio Code website at [<https://code.visualstudio.com/>]  
(<https://code.visualstudio.com/>).

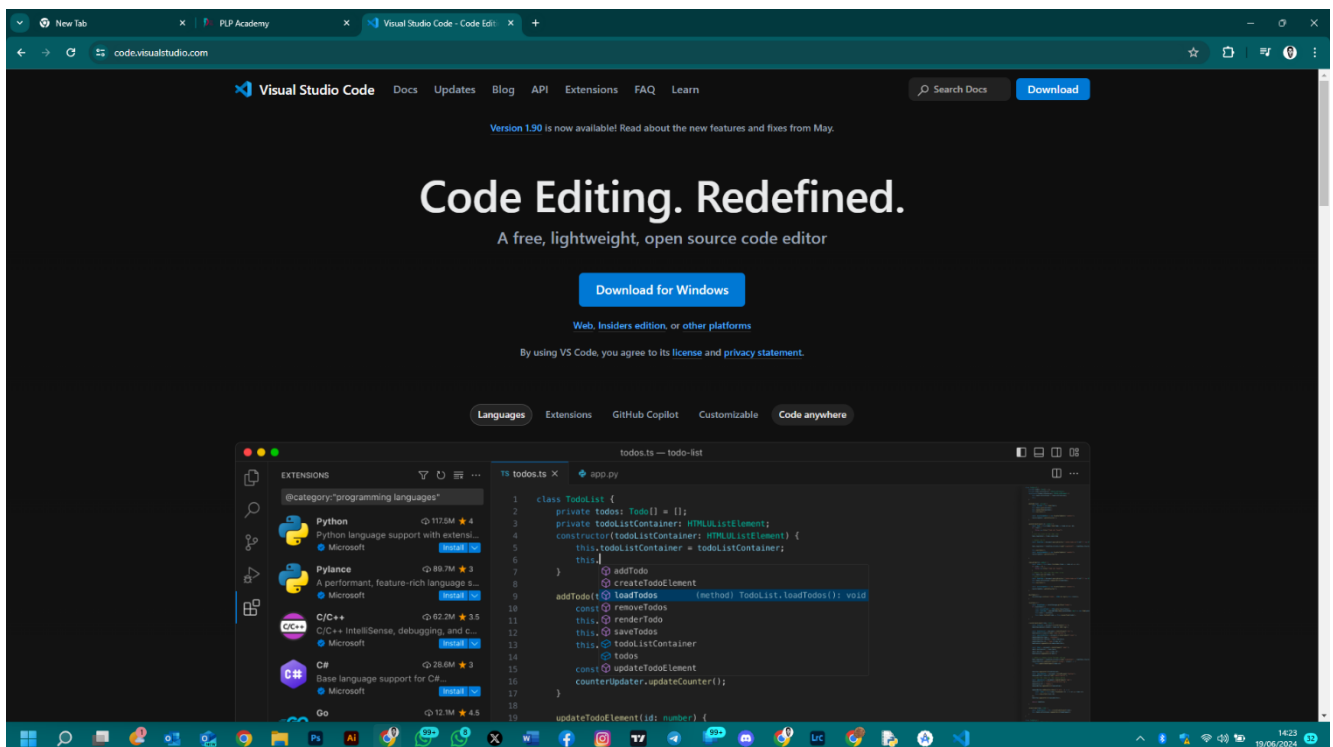
Clicked on the "Download for Windows" button to download the latest version of the installer.



## 2. Run the Installer:

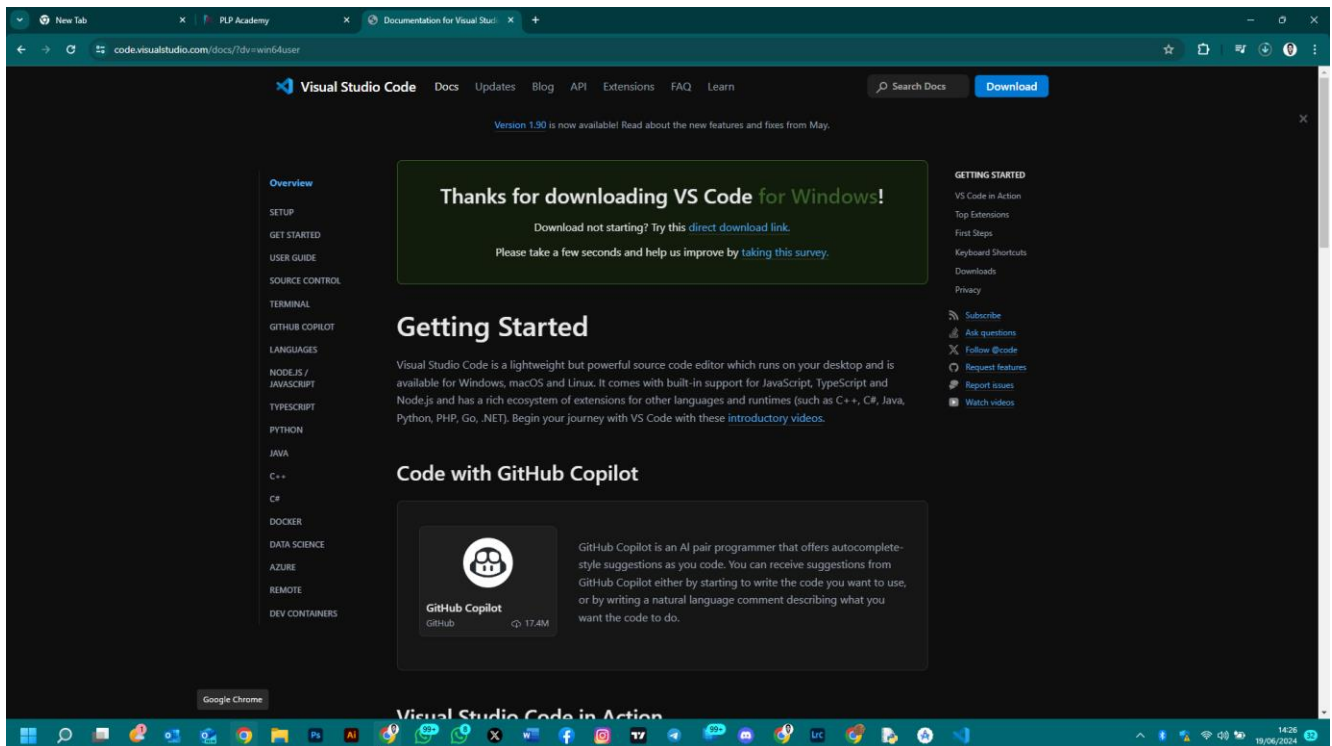
Once the download was complete, i located the downloaded installer file (named something like `VSCodeSetup-version.exe`).

I double-clicked on the installer file to start the installation process.



### 3. Begin Installation:

I was prompted by User Account Control (UAC) for permission to make changes to your system. Click "Yes" to proceed.

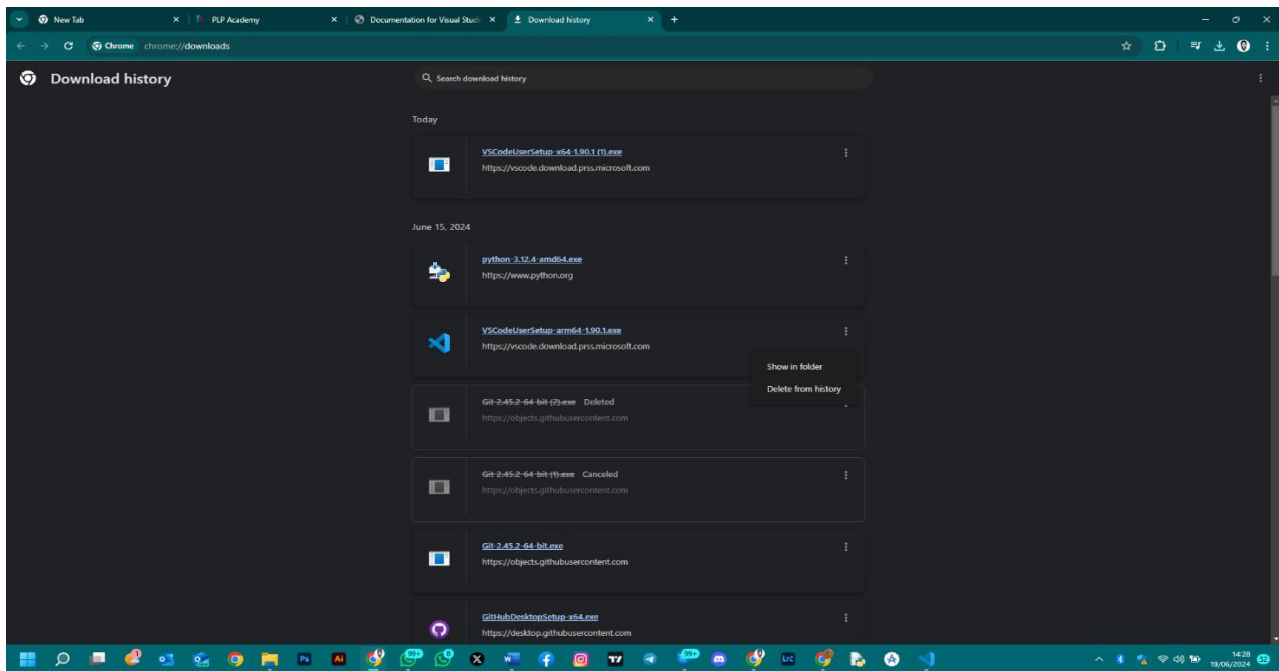


### 4. Select Installation Options/Select Workloads and Components:

I choose the destination folder for Visual Studio Code installation or leave the default setting.

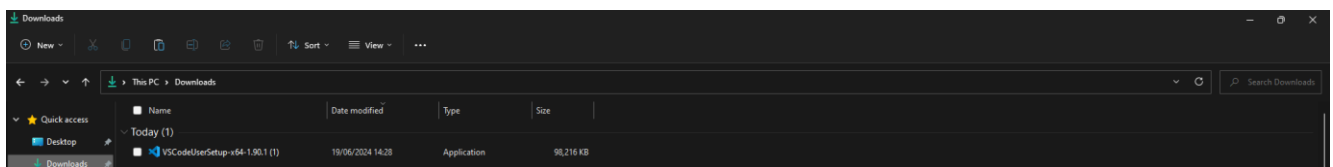
I choose additional tasks during the installation process, like adding VS Code to the PATH for easy access from the command line.

In this Visual Studio Installer, i selected the workloads and components i needed based on my development requirements. Common workloads included ".NET Desktop Development" or "Web Development."



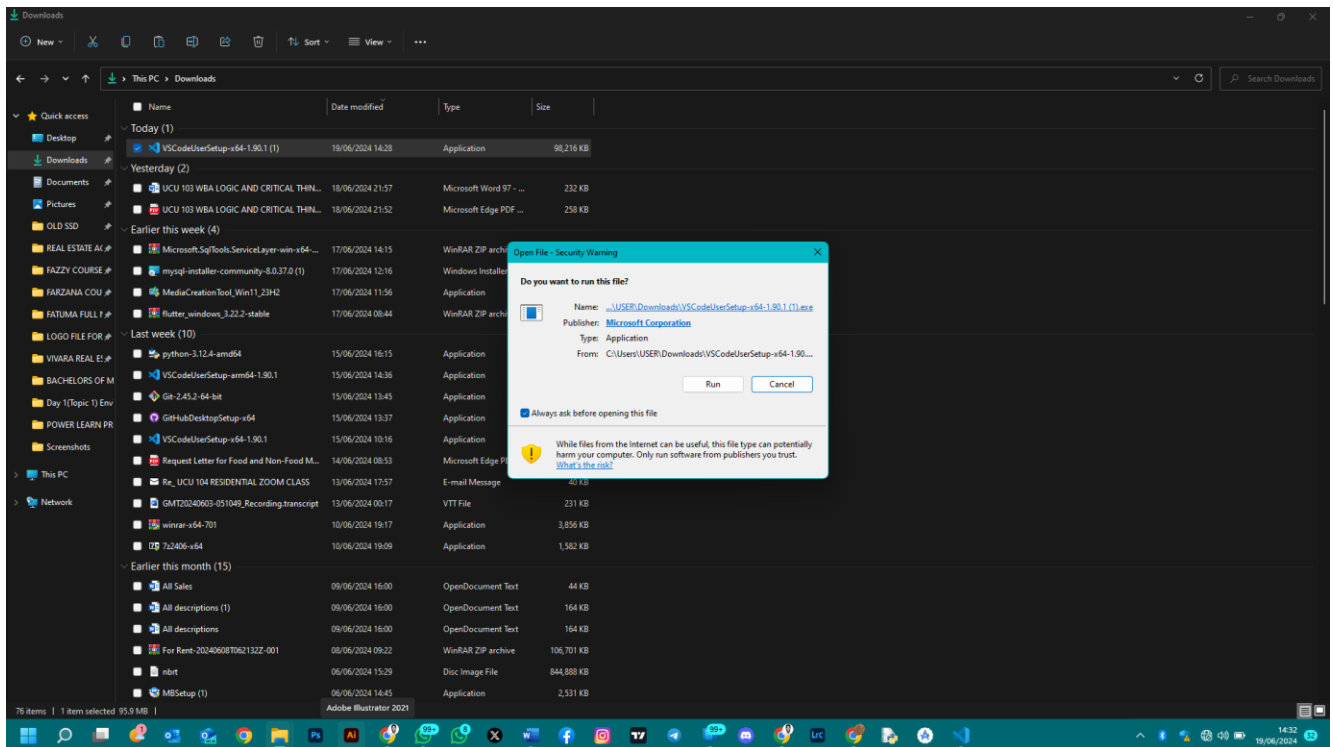
## 5. Install Visual Studio Code:

I clicked the "Next" button to begin the installation process. The installer extracted necessary files and set up Visual Studio Code on the system.



## 6. Launch Visual Studio Code:

Once the installation was complete, i launched Visual Studio Code by double-clicking the shortcut on my desktop, or by searching for it in the Start menu.



## 7. Optional Steps/Choose Development Environment:

**Extensions:** Customized my Visual Studio Code setup by installing extensions for different languages and functionalities. I did this within VS Code itself.

**Settings:** Configured preferences and settings by accessing the settings menu in VS Code. I chose "Development Settings" based on my preferred coding style.

## 8. Update Visual Studio Code:

Visual Studio Code receives regular updates with bug fixes and new features. It's recommended to keep installation up to date. I did this by going to the menu bar: 'Help > Check for Updates'.

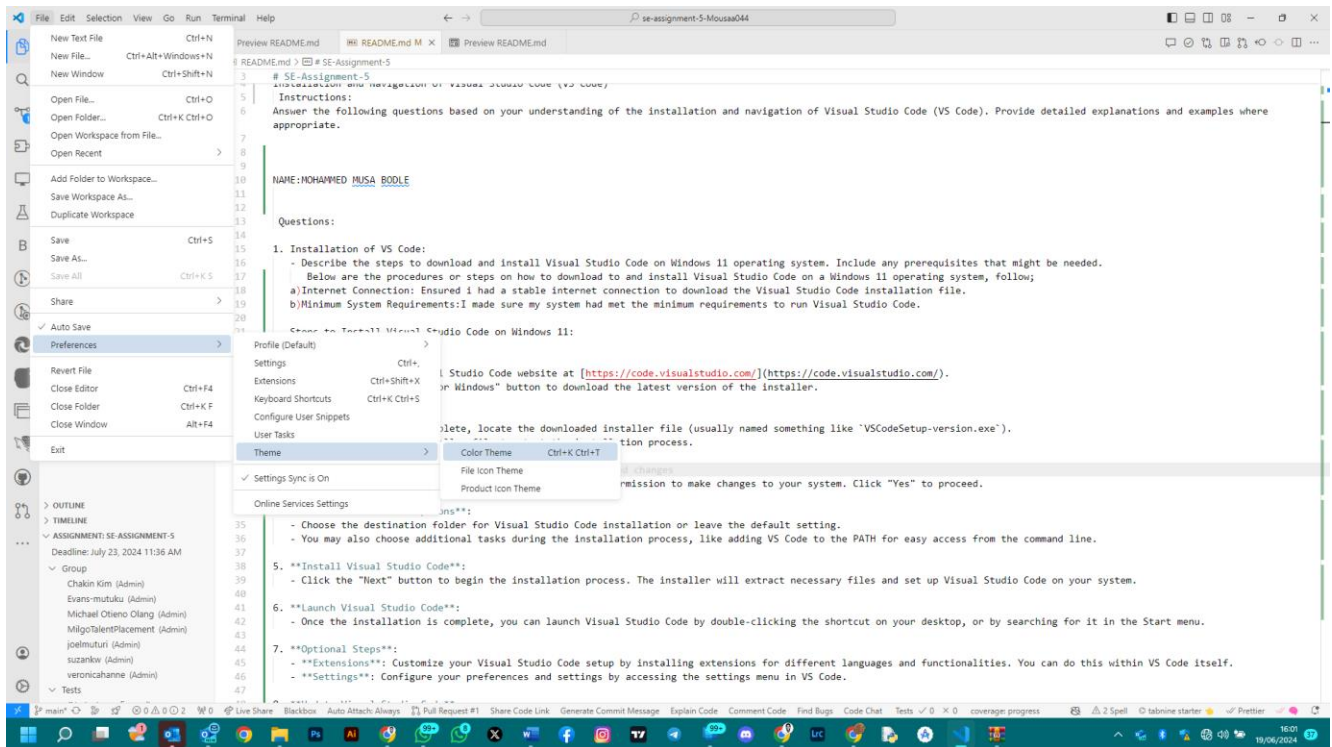
After following these steps, i had Visual Studio Code successfully installed and ready to use on my Windows 11 operating system.

## 2. First-time Setup:

After installing VS Code, what initial configurations and settings should be adjusted for an optimal coding environment? Mention any important settings or extensions.

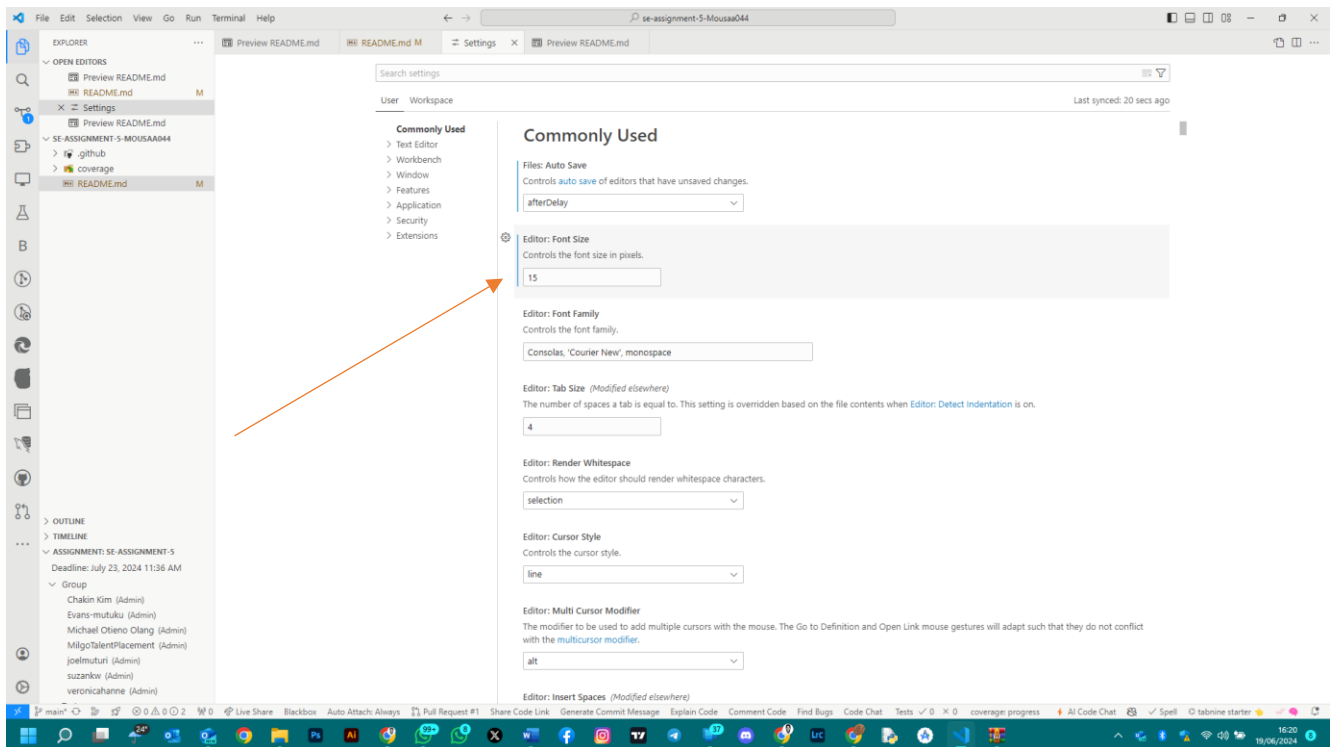
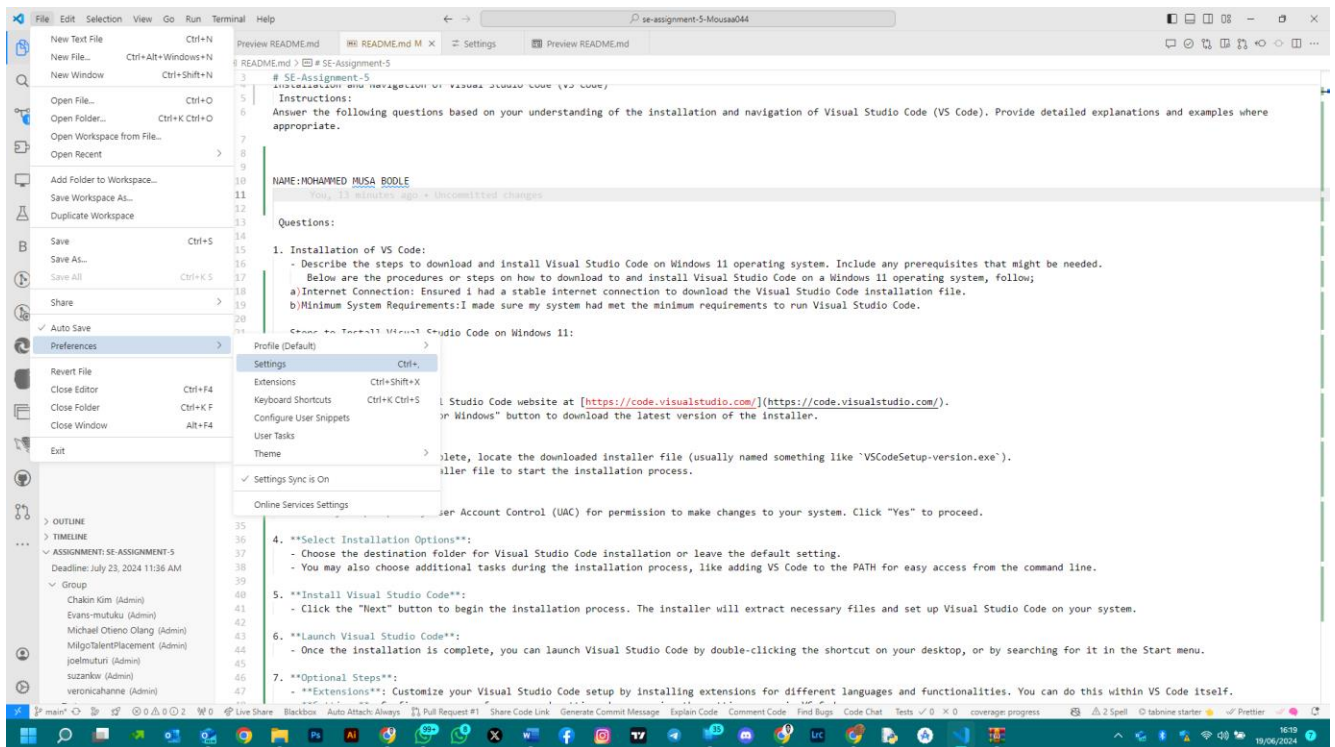
**Theme:**

Got to File > Preferences > Colour Theme and choose a theme that suited my preference. Popular themes include "Dark+ (default dark)" and "Light+ (default light)".



**Font Size:**

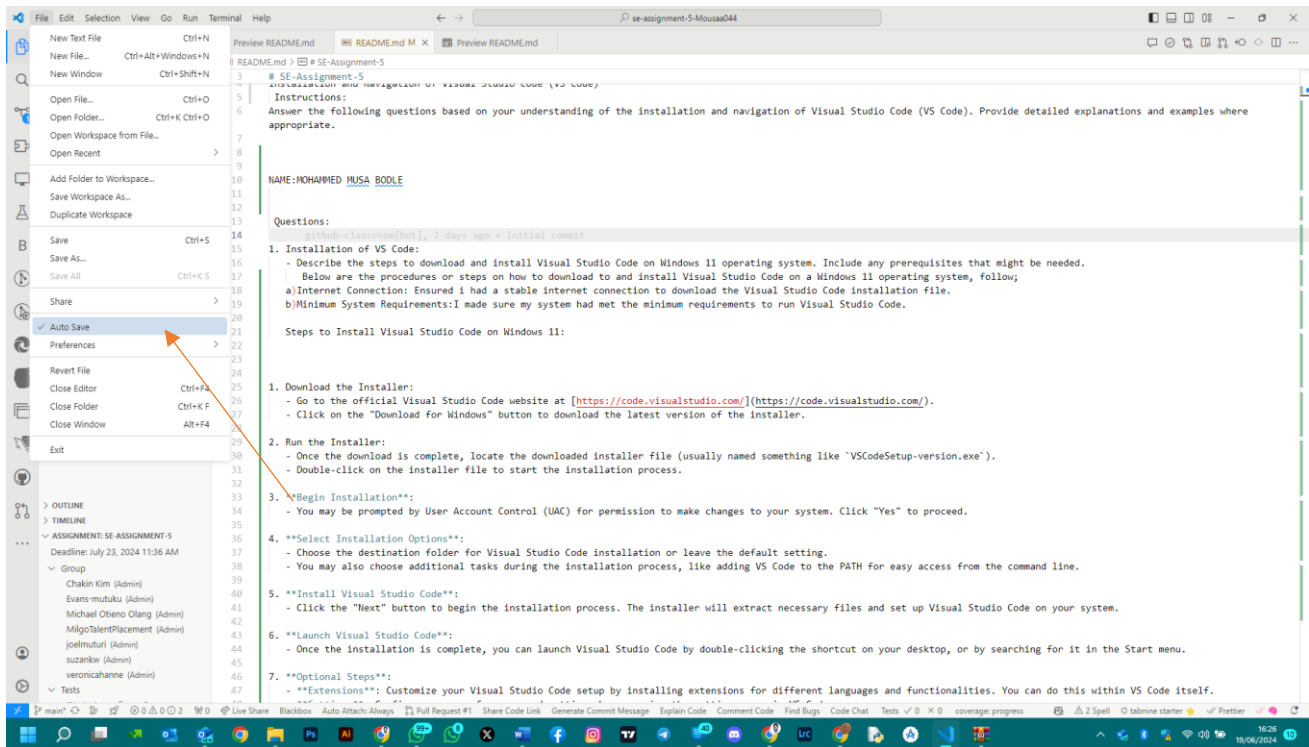
Adjusted the font size by going to File > Preferences > Settings, then searching for "font size" and setting your preferred size.





## Auto Save:

Enable auto-save by going to File > Preferences > Settings, searching for "auto save", and setting it to "afterDelay" or "onWindowChange".



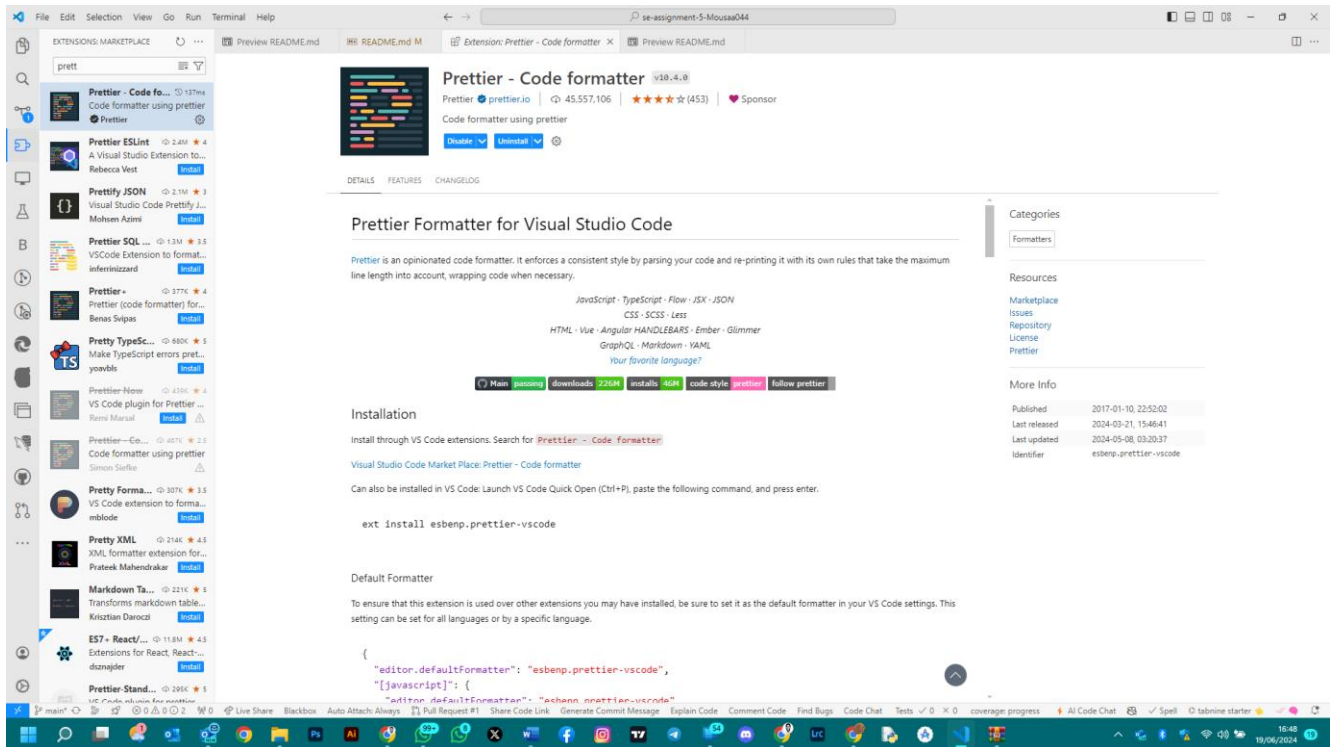


## Editor Format:

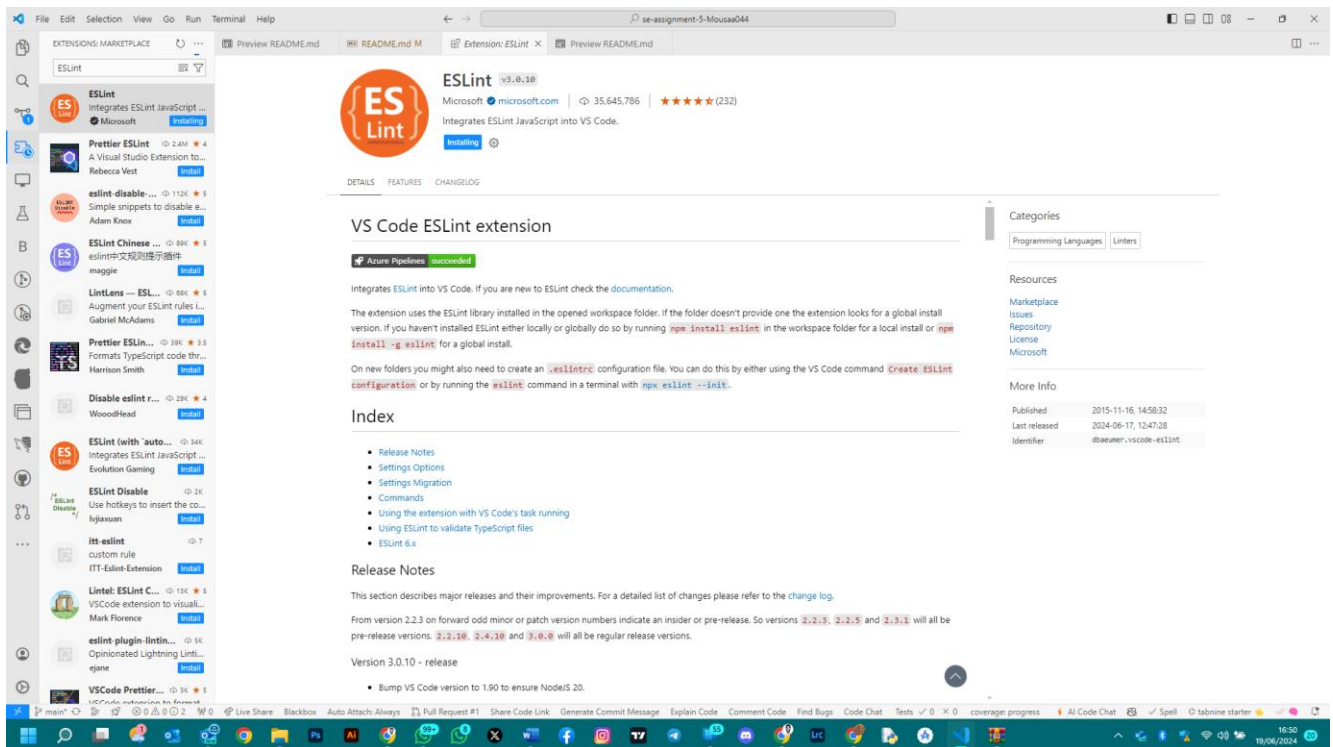
Set up automatic formatting by enabling format on save in File > Preferences > Settings, and searching for "format on save".

## Essential Extensions:

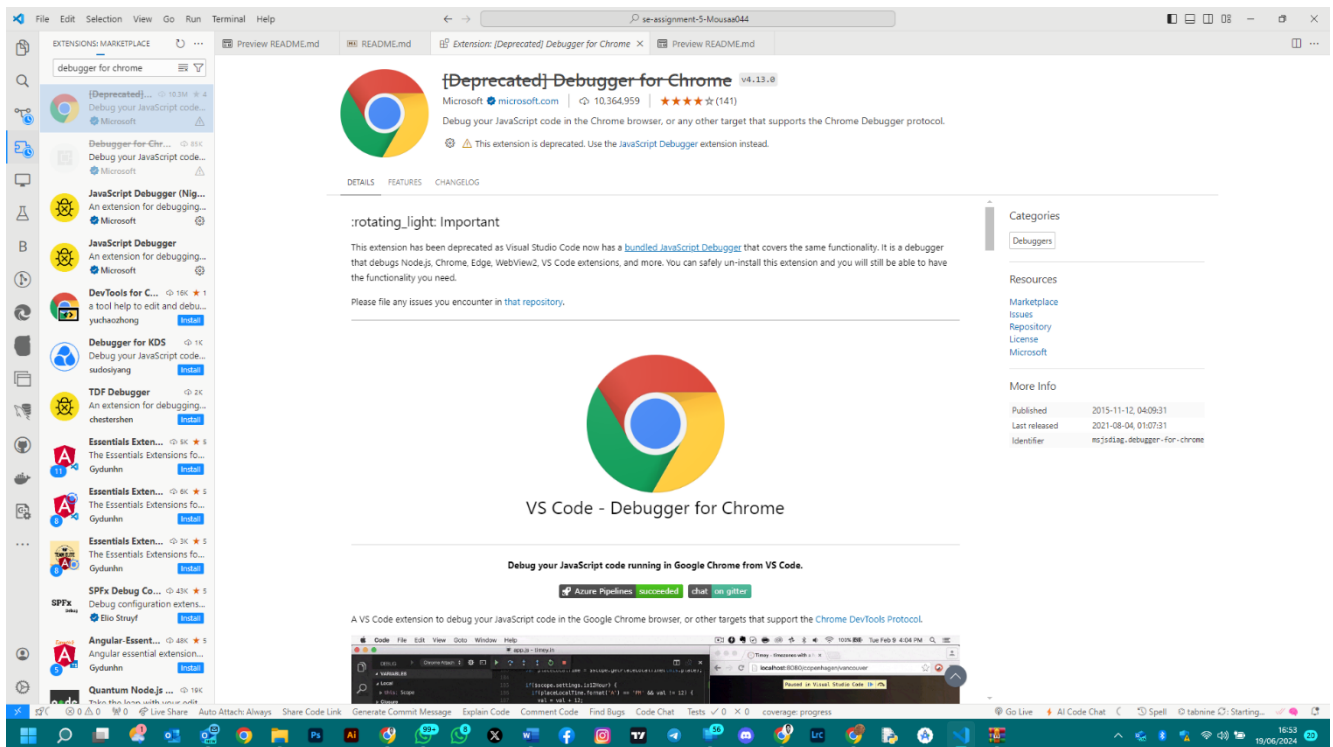
**Prettier - Code formatter:** Automatically formats your code to improve readability.

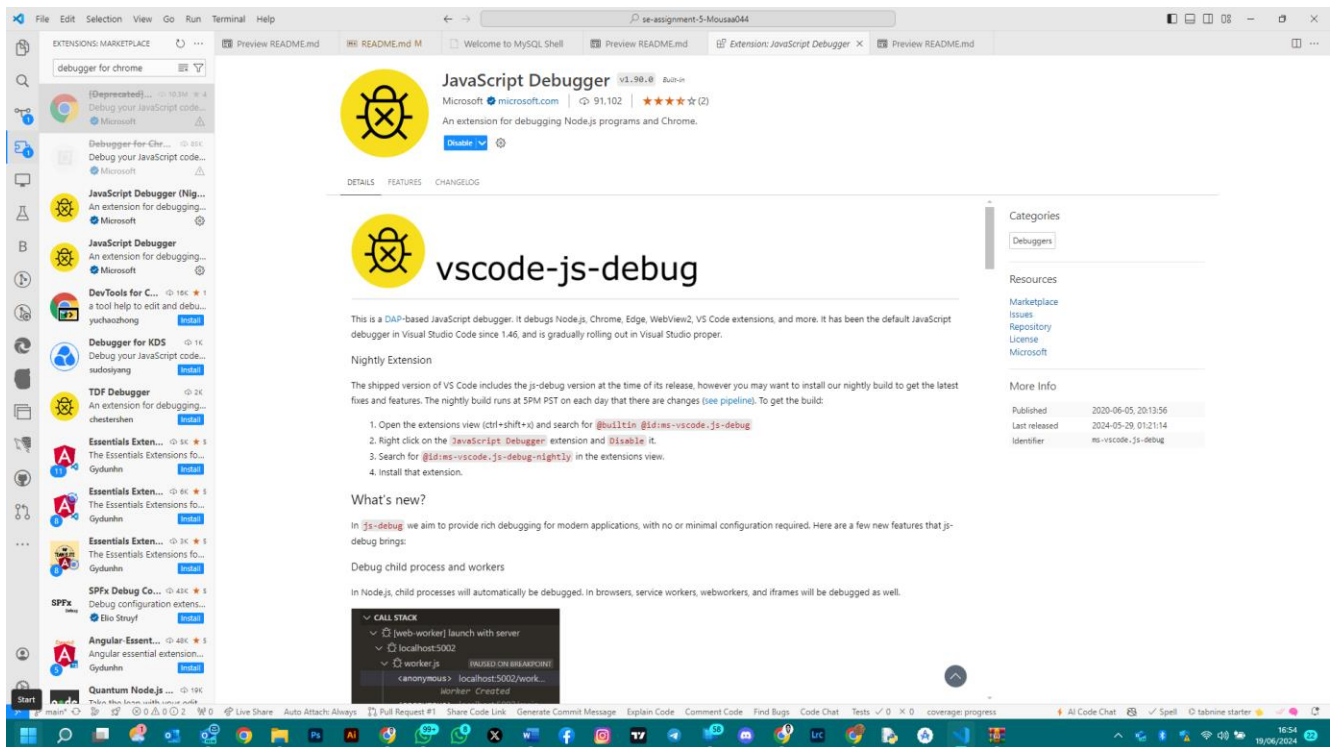


**ESLint:** Integrates ESLint into VS Code for JavaScript linting.

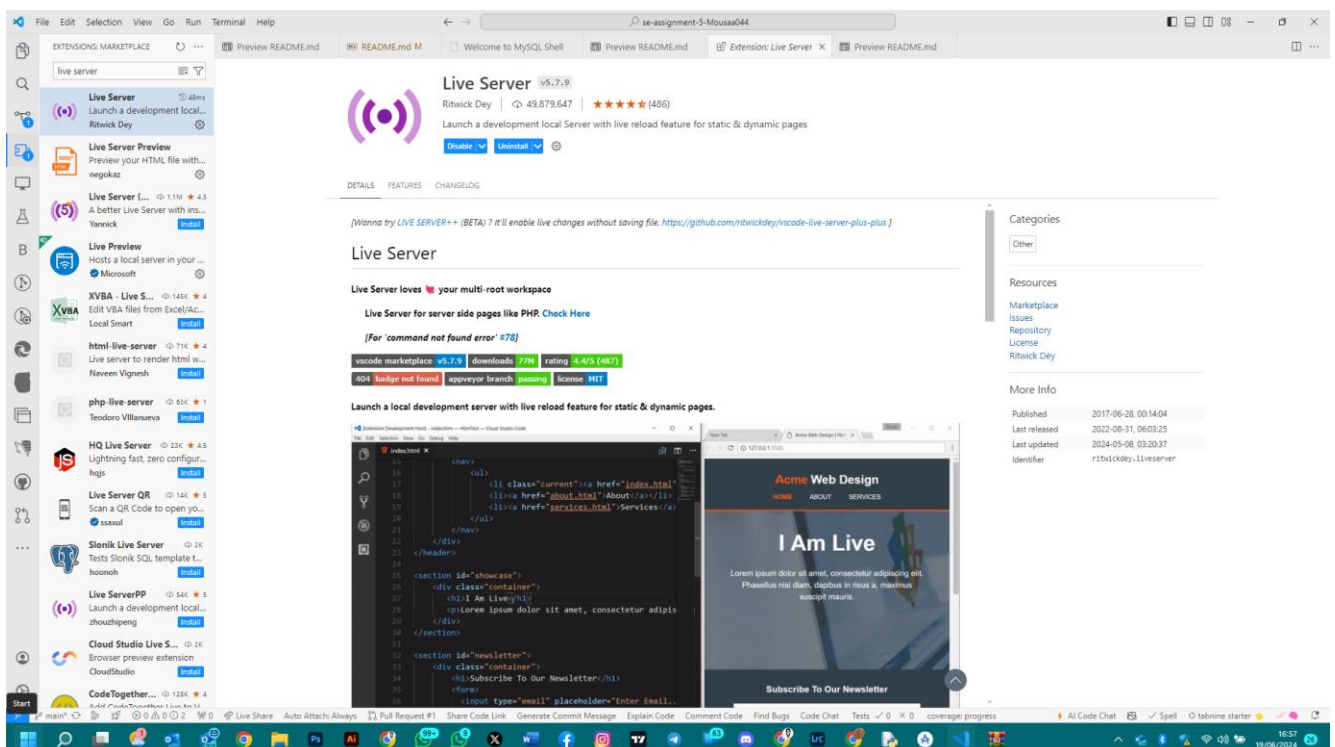


**Debugger for Chrome:** Enables debugging JavaScript code running in the Google Chrome browser.

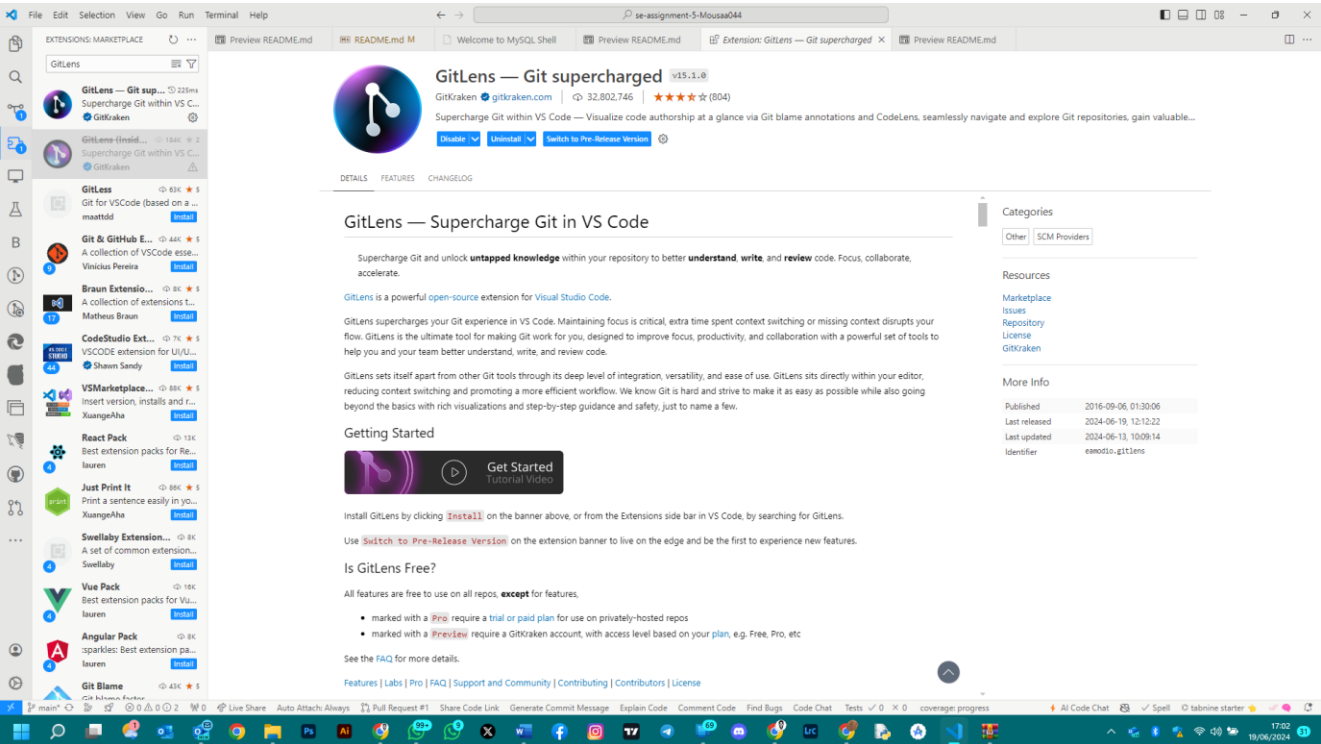
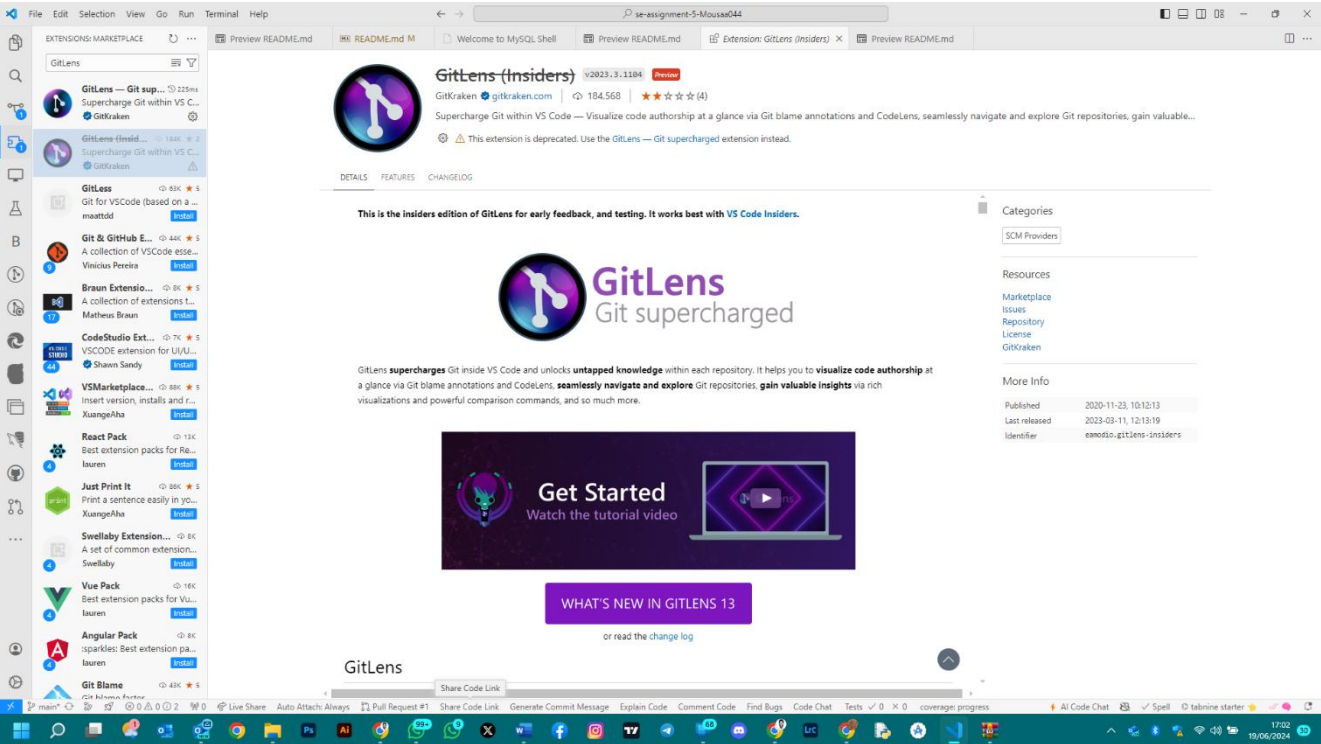




**Live Server:** Launches a development local server with live reload feature for static and dynamic pages.



# GitLens: Enhances the Git capabilities built into VS Code.



### 3. User Interface Overview:

**Explain the main components of the VS Code user interface. Identify and describe the purpose of the Activity Bar, Side Bar, Editor Group, and Status Bar.**

**Activity Bar:** Located on the far left, it allows you to switch between different views such as Explorer, Search, Source Control, Run and Debug, and Extensions.

**Side Bar:** Displays various views like Explorer (file and folder management), Source Control, and Extensions. It's located next to the Activity Bar.

**Editor Group:** The central area where you open, edit, and manage your files. You can split it into multiple groups for side-by-side editing.

**Status Bar:** Located at the bottom, it provides information about the current workspace, such as errors, warnings, the current branch in version control, and file encoding.

### 4. Command Palette:

**What is the Command Palette in VS Code, and how can it be accessed? Provide examples of common tasks that can be performed using the Command Palette.**

The Command Palette is a powerful tool in VS Code that provides access to various commands and settings.

In order to accessing the Command Palette:

Press Ctrl + Shift + P (Windows/Linux) or Cmd + Shift + P (Mac) to open the Command Palette.

Start typing the command you want to execute, and the Command Palette will filter the list of available commands.

#### **Examples of Common Tasks:**

**Opening Settings:** Type Preferences: Open Settings (UI) to open the settings UI.

**Installing Extensions:** Type Extensions: Install Extensions to open the Extensions view and install new extensions.

**Running Tasks:** Type Tasks: Run Task to see a list of available tasks and execute them.

## **5. Extensions in VS Code:**

**Discuss the role of extensions in VS Code. How can users find, install, and manage extensions? Provide examples of essential extensions for web development.**

Extensions enhance the functionality of VS Code. Users can find, install, and manage extensions via the Extensions view in the Activity Bar.

### **Finding and Installing Extensions:**

Click the Extensions icon in the Activity Bar.

Search for the desired extension.

Click "Install".

### **Managing Extensions:**

Click on the gear icon next to the installed extension to manage settings or disable/uninstall it.

### **Essential Extensions for Web Development:**

HTML Snippets

CSS IntelliSense

JavaScript (ES6) code snippets

Npm

## 6. Integrated Terminal:

**Describe how to open and use the integrated terminal in VS Code. What are the advantages of using the integrated terminal compared to an external terminal?**

The integrated terminal in VS Code allows you to run command-line tasks within the same window.

The integrated terminal in VS Code can be opened by pressing Ctrl+ (Windows/Linux) or Cmd + (Mac).

### **Advantages of integrated Terminal**

**Convenience:** It allows the user to run command-line tasks within the same window without leaving the VS-Code environment.

**Integration:** Direct access to the project directory and easy navigation.

**Customization:** Supports multiple terminal instances and shell types (e.g., PowerShell, Command Prompt, Git Bash).

## 7. File and Folder Management:

**Explain how to create, open, and manage files and folders in VS Code. How can users navigate between different files and directories efficiently?**

VS Code provides a user-friendly interface for creating, opening, and managing files and folders.

### **Creating and Opening Files and Folders:**

**Creating:** Right-click in the Explorer view and select "New File" or "New Folder".

**Opening:** Drag and drop files/folders into the Explorer or use File > Open Folder.

### **Efficient Navigation:**

Use Ctrl+P to quickly open files by typing their names. This is known as the Quick Open feature.

Use breadcrumbs in the editor (located above the tabs) to navigate through directories and files.



## **8. Settings and Preferences:**

**Where can users find and customize settings in VS Code? Provide examples of how to change the theme, font size, and key bindings.**

Settings can be customized to tailor VS Code to your specific needs.

### **Accessing Settings:**

Go to File > Preferences > Settings or press Ctrl+,..

The Settings UI provides a user-friendly interface to change various configurations.

### **Examples:**

#### **Change Theme:**

Navigate to File > Preferences > Colour Theme or use Ctrl+K Ctrl+T.

Choose from the list of available themes.

Change theme.

#### **Change Font Size:**

In Settings, search for "font size" and set "Editor: Font Size" to your preferred value.

Font size

#### **Change Key Bindings:**

Go to File > Preferences > Keyboard Shortcuts or press Ctrl+K Ctrl+S.

Modify existing key bindings or add new ones to fit your workflow.

Key bindings.

## **9. Debugging in VS Code:**

**Outline the steps to set up and start debugging a simple program in VS Code. What are some key debugging features available in VS Code?**

VS Code offers robust debugging capabilities for various programming languages.

### **Steps to Set Up and Start Debugging:**

Open your project in VS Code.

Go to the Run and Debug view in the Activity Bar or press Ctrl+Shift+D.

Click "create a launch.json file" to configure the debugger for your project.

Add breakpoints by clicking in the gutter next to the line numbers.

Start debugging by clicking the green play button.

### **Key Debugging Features are;**

**Breakpoints:** Pause execution at specific points.

**Watch variables:** Monitor the values of variables in real-time.

**Call stack:** View the call stack to understand the sequence of function calls.

**Debug console:** Evaluate expressions and execute commands during debugging.

## **10. Using Source Control:**

**How can users integrate Git with VS Code for version control? Describe the process of initializing a repository, making commits, and pushing changes to GitHub.**

VS Code integrates seamlessly with Git for version control, allowing you to manage your source code directly within the editor.

Integrating Git with VS Code:

### **Initialize a Repository:**

Open the Source Control view by clicking the Source Control icon in the Activity Bar or pressing Ctrl+Shift+G.

Click "Initialize Repository" to create a new Git repository in your project folder.

### **Making Commits:**

Stage changes by clicking the plus icon next to the files you want to commit.

Write a commit message in the text box at the top of the Source Control view.

Click the check mark to commit the changes.

**Pushing Changes to GitHub:**

Click the ellipsis (...) in the Source Control view and select "Push".

Follow the prompts to link your repository to a GitHub remote.

Enter your GitHub credentials if prompted.

**Submission Guidelines:**

- Your answers should be well-structured, concise, and to the point.

- Provide screenshots or step-by-step instructions where applicable.
- Cite any references or sources you use in your answers.
- Submit your completed assignment by 1st July