**INSTALLATION AND NAVIGATION OF VISUAL STUDIO CODE (VS CODE)**

**Question 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.

According to the steps outlined by geeksforgeeks (2022), the steps include:

1. **Prerequisites:** Before installing VS Code, ensure that your system meets the minimum hardware requirements. VS Code is a lightweight application, so most modern systems should be able to run it without any issues.
2. **Download VS Code:** Visit the official Visual Studio Code website (<https://code.visualstudio.com/>) and click on the "Download for Windows" button under the "Download" section.
3. **Run the Installer:** Once the download is complete, locate the installer file (e.g., VSCodeUserSetup-x64-1.xx.x.exe) and double-click on it to run the installer.
4. **Accept Licence Agreement:** The installer will start, and you'll be presented with a license agreement. Review the terms and conditions, and if you agree, select "I accept the agreement" and click "Next".
5. **Choose Installation Location:** By default, VS Code will be installed in the following location: C:\Users<username>\AppData\Local\Programs\Microsoft VS Code. If you want to change the installation location, click "Browse" and select your desired location. Click "Next" to proceed.
6. **Select Additional Tasks:** You can choose additional tasks during the installation process. These options include:

* Adding Open with Code context menu entry (recommended).
* Creating a desktop icon for Visual Studio Code (optional).
* Adding to the PATH environment variable (optional, but recommended for advanced users).

Select the desired options and click "Next".

1. **Install:** Click "Install" to begin the installation process. This may take a few minutes, depending on your system's performance.
2. **Finish Installation:** Once the installation is complete, you can choose to launch Visual Studio Code by checking the "Launch Visual Studio Code" option. Click "Finish" to complete the installation process.

**Question 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.

According to visualstudiocode (2020), the following are the initial configurations and settings and the extensions necessary for an optimal coding environment:

1. **Configure Git Integration:** If you plan to use VS Code for version control with Git, you should configure the Git integration. Go to "File" > "Preferences" > "Settings" (or press "Ctrl+," on Windows) and search for "git". You can set the path to your Git executable and enable features like Git blame annotations, commit preview, and more.
2. **Install Essential Extensions:** To install extensions, go to the Extensions view (Ctrl+Shift+X on Windows), search for the desired extension, and click "Install". The following are examples of essential extensions:

* Python: Provides rich Python language support for Visual Studio Code, including features like IntelliSense, linting, debugging, code navigation, and more.
* Pylance: A high-performance language server for Python that offers advanced type checking, code completion, and other language-related features.
* Code Runner: Allows you to run code snippets or entire files with a simple keyboard shortcut or command.
* Prettier: An opinionated code formatter that enforces a consistent code style across various languages, including JavaScript, TypeScript, CSS, HTML, and more.
* Beautify: Automatically formats and beautifies code with proper indentation and style.
* Bracket Pair Colorizer: Helps to visualize matching brackets with different colors.

1. **Customise User Settings:** You can customize various settings in VS Code to fit your preferences. Go to "File" > "Preferences" > "Settings" (or press "Ctrl+," on Windows) and explore the available options. Some important settings to consider:

* "editor.fontSize": Adjust the font size for better readability.
* "editor.fontFamily": Change the font family used in the editor.
* "editor.lineNumbers": Show or hide line numbers in the editor.
* "editor.wordWrap": Enable or disable word wrap in the editor.
* "files.autoSave": Configure auto-save behavior for files.

1. **Configure Keyboard Shortcuts:** VS Code allows you to customize keyboard shortcuts for various actions. Go to "File" > "Preferences" > "Keyboard Shortcuts" (or press "Ctrl+K Ctrl+S" on Windows) to view and modify keyboard shortcuts.
2. **Set Up Integrated Terminal:** VS Code has an integrated terminal that can be useful for running command-line tools and scripts. You can access it via "View" > "Terminal" (or press "Ctrl+" on Windows). Configure the integrated terminal to use your preferred shell (e.g., PowerShell, Bash, Zsh) by going to "File" > "Preferences" > "Settings" and searching for "terminal.integrated.shell".
3. **Configure Themes and File Icons:** VS Code offers a variety of themes and file icons to personalize the appearance of the editor. You can explore and install different themes and file icon themes from the Extensions view.
4. **Set Up Version Control Integration:** If you're working with version control systems like Git or SVN, you can configure their integration within VS Code. This allows you to perform common version control operations like committing, pushing, and pulling changes directly from the editor.

**Question 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.

As elaborated by visualstudiocode (2024), on their section on visual studio code user interface, the purpose of the activity bar, side bar, editor group, and status bar are as follows:

1. **Activity Bar:** The Activity Bar is the vertical bar located on the far left side of the VS Code window. It provides quick access to different views and functionalities within the editor. The main icons in the Activity Bar are:

* Explorer: Displays the file explorer, which allows you to navigate and manage your project files and folders.
* Search: Enables you to search for text within your project or open files.
* Source Control: Provides integration with version control systems like Git, allowing you to manage your code repositories.
* Run and Debug: Offers tools for running and debugging your code.
* Extensions: Allows you to browse, install, and manage extensions that add additional functionality to VS Code.

1. **Side Bar:** The Side Bar is the vertical panel on the left side of the editor window. It displays different views or panels based on the selected activity from the Activity Bar. For example, when the Explorer activity is selected, the Side Bar shows the file explorer view, allowing you to navigate your project's file structure.
2. **Editor Group:** The Editor Group is the main area in the center of the VS Code window where you edit and view your code files. To work with multiple files simultaneously, you can split the Editor Group into numerous panes, either horizontally or vertically. Each pane within the Editor Group is called an Editor.
3. **Status Bar:** The Status Bar is located at the bottom of the VS Code window. It provides valuable information and indicators related to your current project and open files. Some of the key elements in the Status Bar include:

* Branch name and current state (for version control systems like Git).
* Language mode and encoding of the active file.
* Line and column position of the cursor.
* Indentation settings (spaces or tabs).
* File format and line endings.
* Errors, warnings, and other notifications related to your code.

**Question 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.

**Definition:** The Command Palette in Visual Studio Code (VS Code) is a powerful tool that provides access to a wide range of commands and actions available within the editor. It allows you to quickly search and execute commands without having to navigate through menus or remember specific keyboard shortcuts.

To access the Command Palette, you can use one of the following methods:

* Keyboard Shortcut: Press Ctrl+Shift+P (Windows/Linux) or Cmd+Shift+P (macOS) to open the Command Palette.
* Menu: Go to "View" > "Command Palette" in the main menu bar.

Once the Command Palette is open, you can start typing the name of the command you want to execute or a brief description of the task you want to perform. VS Code will provide a list of matching commands, and you can select the desired command using the up and down arrow keys or by clicking on it with the mouse.

Here are some examples of common tasks that can be performed using the Command Palette:

1. **File Operations:**

* Create a new file: Type "new file" and select the "Create New File" command.
* Open a file: Type "open file" and select the "Open File" command to browse and open a file.
* Save a file: Type "save" and select the "Save" command to save the currently open file.

1. **Editing:**

* Format document: Type "format" and select the "Format Document" command to automatically format the code in the active editor.
* Toggle comments: Type "comment" and select the "Toggle Line Comment" command to comment or uncomment the selected lines of code.
* Rename a symbol: Type "rename" and select the "Rename Symbol" command to rename a variable, function, or other symbol across your codebase.

1. **Navigation:**

* Go to a specific line: Type "go to line" and select the "Go to Line..." command to jump to a specific line number in the active file.
* Go to a symbol: Type "go to symbol" and select the "Go to Symbol..." command to search for and navigate to a specific symbol (e.g., function, class, variable) within your project.

1. **Extensions:**

* Install an extension: Type "install extension" and select the "Extensions: Install Extension" command to search and install a new extension from the Visual Studio Marketplace.
* Manage extensions: Type "manage extensions" and select the "Extensions: Manage Extensions" command to view and manage your installed extensions.

1. **Customization:**

* Change color theme: Type "color theme" and select the "Preferences: Color Theme" command to browse and apply a different color theme to the editor.
* Open settings: Type "settings" and select the "Preferences: Open Settings (UI)" command to access and modify VS Code's settings.

1. **Version Control:**

* Initialize a Git repository: Type "git init" and select the "Git: Initialize Repository" command to create a new Git repository for your project.
* Commit changes: Type "commit" and select the "Git: Commit" command to commit your changes to the Git repository.

(visualstudiocode, 2020)

**Question 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.

Role of exensions in VS Code: Extensions play a crucial role in Visual Studio Code (VS Code) by providing additional functionality and enhancing the development experience. VS Code has a rich ecosystem of extensions that cater to various programming languages, frameworks, tools, and workflows. Extensions can add new features, integrate tools, customize the editor's appearance, and much more.

To find, install, and manage extensions in VS Code, you can follow these steps:

1. **Finding Extensions:**

* Open the Extensions view by clicking on the square icon in the Activity Bar (the fifth icon from the top) or by pressing Ctrl+Shift+X (Windows/Linux) or Cmd+Shift+X (macOS).
* In the Extensions view, you can search for extensions by typing keywords in the search bar at the top.
* You can browse through different categories, such as "Popular," "Most Popular," or specific languages and tools.
* Extensions are listed with descriptions, ratings, and the number of installations, which can help you evaluate their popularity and usefulness.

1. **Installing Extensions:**

* Once you've found an extension you want to install, simply click the "Install" button next to its listing.
* Some extensions may require you to reload VS Code after installation. Follow the prompts or reload manually by clicking the "Reload" button.

1. **Managing Extensions:**

* Installed extensions can be viewed and managed in the Extensions view.
* To update an extension, click the "Update" button next to the extension listing.
* To disable or uninstall an extension, click the gear icon next to the extension listing and select the appropriate option.
* You can also access extension settings and configurations by clicking the gear icon.

Here are some essential extensions for web development:

1. **Live Server:** Launches a local development server with live reload functionality for static and dynamic pages. It automatically reloads the page whenever you save changes to your code.
2. **Emmet:** A powerful tool for writing HTML and CSS code more efficiently. It provides shorthand syntax for generating markup and styles.
3. **Debugger for Chrome:** Allows you to debug your JavaScript code directly in the Chrome browser from within VS Code.
4. **ESLint:** Integrates the popular ESLint JavaScript linter into VS Code, enabling real-time code analysis and error checking.
5. **Prettier:** An opinionated code formatter that automatically formats your code according to predefined rules, ensuring consistent code style across your project.
6. **Auto Rename Tag:** Automatically renames matching HTML/XML tags when you rename one tag.
7. **CSS Peek:** Allows you to peek or go to the definition of CSS classes and IDs from HTML files.
8. **Path Intellisense:** Autocompletes filenames while typing relative paths in your code.
9. **Bracket Pair Colorizer:** Colorizes matching brackets, making it easier to identify nested code blocks.
10. **GitLens:** Supercharges the Git capabilities within VS Code, providing a rich visual experience for working with Git repositories.

(visualstudiocode, 2020)

**Question 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?

Here's how to open and use the integrated terminal in VS Code:

1. **Opening the Integrated Terminal:**

* You can open the integrated terminal using one of the following methods:
* Press Ctrl+ (backtick) on Windows/Linux or Cmd+ on macOS.
* Go to the main menu and select "Terminal" > "New Terminal".
* Use the Command Palette (Ctrl+Shift+P or Cmd+Shift+P) and search for "Terminal: Create New Integrated Terminal".

1. **Navigating Terminals:**

* If you have multiple terminal instances open, you can switch between them using the drop-down menu in the terminal panel or by pressing Ctrl+PageUp and Ctrl+PageDown on Windows/Linux, or Cmd+Option+Right and Cmd+Option+Left on macOS.
* You can split the terminal panel horizontally or vertically by right-clicking on the terminal panel and selecting "Split Terminal".

1. **Using the Integrated Terminal:**

* Once the integrated terminal is open, you can interact with it just like you would with any external terminal.
* The terminal starts in the context of your current workspace or project folder, so you don't need to navigate to your project's directory manually.
* You can run commands, scripts, and tools directly from the integrated terminal, and the output will be displayed in real-time within the terminal panel.

1. **Terminal Tasks:**

* VS Code allows you to define custom terminal tasks in a tasks.json file, which can automate common command-line operations or build processes.
* These tasks can be run from the Command Palette or assigned keyboard shortcuts for quick access.

Advantages of using the integrated terminal in VS Code:

**a) Seamless Integration:** The integrated terminal is tightly coupled with the editor, providing a unified and streamlined development experience. You can switch between coding and terminal tasks without leaving the editor.

**b) Context Awareness:** The integrated terminal is aware of your current workspace or project folder, making it easier to navigate and execute commands related to your project.

**c) Convenience:** You don't need to switch between different applications or windows to access the terminal. The integrated terminal is always available within the VS Code interface.

**d) Consistent Environment:** The integrated terminal inherits the same environment variables and settings as your external terminal, ensuring consistent behavior across different terminals.

**e) Terminal Splitting:** You can split the terminal panel horizontally or vertically, enabling you to run multiple terminal instances side by side.

**f) Debugging Integration:** The integrated terminal can be used in conjunction with VS Code's debugging features, allowing you to run and debug your applications directly from the editor.

(visualstudiocode, 2020)

**Question 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?

How you can manage files and folders in VS Code:

1. **Creating a New File:**

* To create a new file, open the Explorer panel by clicking the first icon in the Activity Bar or by pressing Ctrl+Shift+E (Windows/Linux) or Cmd+Shift+E (macOS).
* In the Explorer panel, right-click on the folder where you want to create the new file, and select "New File" from the context menu.
* Alternatively, you can use the Command Palette (Ctrl+Shift+P or Cmd+Shift+P) and search for "New File", then select the command.

1. **Opening an Existing File:**

* In the Explorer panel, navigate to the file you want to open, and click on it.
* You can also use the Ctrl+P (Windows/Linux) or Cmd+P (macOS) shortcut to open the "Quick Open" dialog, which allows you to search for and open files by their name or path.
* From the main menu, go to "File" > "Open..." to browse and open a file from your file system.

1. **Creating a New Folder:**

* In the Explorer panel, right-click on the parent folder where you want to create the new folder.
* Select "New Folder" from the context menu.
* Enter the desired name for the new folder.

1. **Renaming Files and Folders:**

* In the Explorer panel, right-click on the file or folder you want to rename.
* Select "Rename" from the context menu.
* Enter the new name for the file or folder.

1. **Deleting Files and Folders:**

* In the Explorer panel, right-click on the file or folder you want to delete.
* Select "Delete" from the context menu.
* Confirm the deletion when prompted.

1. **Navigating Between Files:**

* Use the Ctrl+Tab (Windows/Linux) or Cmd+Tab (macOS) shortcut to switch between open editor tabs.
* Click on the desired file in the Explorer panel to open it in the editor.
* Use the "Go to File..." command (Ctrl+P or Cmd+P) to search and open files by their name or path.

1. **Navigating Between Folders:**

* In the Explorer panel, click on the folder icons to expand or collapse folders.
* Use the breadcrumb navigation at the top of the Explorer panel to quickly navigate up or down the folder hierarchy.
* The "Open Folder" command (Ctrl+K Ctrl+O on Windows/Linux or Cmd+K Cmd+O on macOS) allows you to open a folder or project directory in the Explorer panel.

1. **Working with Multiple Folders:**

* VS Code supports working with multiple folders or projects simultaneously.
* Use the "Add Folder to Workspace..." command (Ctrl+K Ctrl+O on Windows/Linux or Cmd+K Cmd+O on macOS) to add additional folders to the current workspace.
* The Explorer panel will display all the folders and files in the current workspace, making it easy to navigate between them.

Additionally, VS Code provides several keyboard shortcuts and commands to streamline file and folder management tasks, such as moving files, copying paths, and revealing files in the Explorer or system file explorer.

(visualstudiocode, 2020)

**Question 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 keybindings.

How to access and modify settings in VS Code:

1. **Opening the Settings Editor:**

* You can open the Settings Editor by going to "File" > "Preferences" > "Settings" (or by pressing Ctrl+, on Windows/Linux or Cmd+, on macOS).
* Alternatively, you can use the Command Palette (Ctrl+Shift+P or Cmd+Shift+P) and search for "Preferences: Open Settings (UI)".

1. **Changing the Theme:**

* In the Settings Editor, search for "workbench.colorTheme" or "Preferences: Color Theme" in the Command Palette.
* Click on the "Edit in settings.json" link to open the settings.json file.
* Set the value of "workbench.colorTheme" to the desired theme name (e.g., "workbench.colorTheme": "Visual Studio Dark").
* You can also browse and preview available themes by clicking on the theme name in the Settings Editor.

1. **Adjusting the Font Size:**

* In the Settings Editor, search for "editor.fontSize" or "Editor: Font Size" in the Command Palette.
* Click on the "Edit in settings.json" link to open the settings.json file.
* Set the value of "editor.fontSize" to your desired font size (e.g., "editor.fontSize": 14).

1. **Modifying Keybindings:**

* To modify keybindings, go to "File" > "Preferences" > "Keyboard Shortcuts" (or use the Command Palette and search for "Preferences: Open Keyboard Shortcuts").
* In the Keyboard Shortcuts editor, you can search for a specific command or scroll through the list of available commands.
* To modify a keybinding, click on the pencil icon next to the command, and then press the desired key combination.
* You can also create a new keybinding by clicking on the plus icon at the top and entering the command and desired key combination.

1. **User Settings vs. Workspace Settings:**

* VS Code has two levels of settings: User Settings and Workspace Settings.
* User Settings are global settings that apply to all instances of VS Code on your machine.
* Workspace Settings are specific to a particular project or workspace and override User Settings when conflicts arise.
* To modify User Settings, open the settings.json file from the Command Palette by searching for "Preferences: Open Settings (JSON)".
* To modify Workspace Settings, open the .vscode/settings.json file within your project folder.

1. **Exploring Settings Options:**

* In the Settings Editor, you can explore various settings categories by clicking on the different sections in the left pane.
* Each setting has a description and provides information about its purpose and available values.
* You can also search for specific settings by typing in the search box at the top of the Settings Editor.

(visualstudiocode, 2020)

**Question 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?

Here are the steps to set up and start debugging a simple program in VS Code, along with some key debugging features:

1. **Set up a debugger configuration:**

* Go to the Command Palette (Ctrl+Shift+P or Cmd+Shift+P) and search for "Debug: Open launch.json".
* This will create a launch.json file in the .vscode folder of your project, which is used to configure the debugger.
* In the launch.json file, you'll find different configuration options based on the programming language or environment you're working with.
* For a simple program, you can typically use the provided configuration templates or make minor adjustments as needed.

1. **Set breakpoints:**

* Open the file you want to debug in the editor.
* Click on the left gutter (the narrow vertical area on the left side of the editor) next to the line of code where you want to set a breakpoint.
* Alternatively, you can place the cursor on the desired line and press F9 (Windows/Linux) or Cmd+F9 (macOS) to toggle a breakpoint.

1. **Start the debugging session:**

* In the Debug view (accessible by clicking the bug icon in the Activity Bar or by pressing Ctrl+Shift+D on Windows/Linux or Cmd+Shift+D on macOS), select the appropriate configuration from the drop-down list at the top.
* Click the green "Start Debugging" button (or press F5) to start the debugging session.
* The program will run until it hits the first breakpoint you set.

1. **Debug actions and features:**

* While the program is paused at a breakpoint, you can use various debugging actions and features:
* Step Over (F10): Execute the current line of code and move to the next line.
* Step Into (F11): Step into a function or method call if the line contains one.
* Step Out (Shift+F11): Step out of the current function or method.
* Restart (Ctrl+Shift+F5): Restart the debugging session from the beginning.
* Stop (Shift+F5): Stop the current debugging session.
* In the Debug view, you can inspect variables, watch expressions, and view the call stack.
* The Debug Console panel displays output from your program, such as console logs or error messages.

1. **Advanced debugging features:**

* VS Code provides additional debugging features, such as:
* **Conditional breakpoints:** Set breakpoints that only trigger under specific conditions.
* **Data breakpoints:** Pause execution when a variable or expression changes its value.
* **Logpoints:** Log messages to the console without breaking execution.
* **Multi-target debugging:** Debug multiple processes or targets simultaneously.
* **Debug configurations:** Create and switch between different debugging configurations for various scenarios.

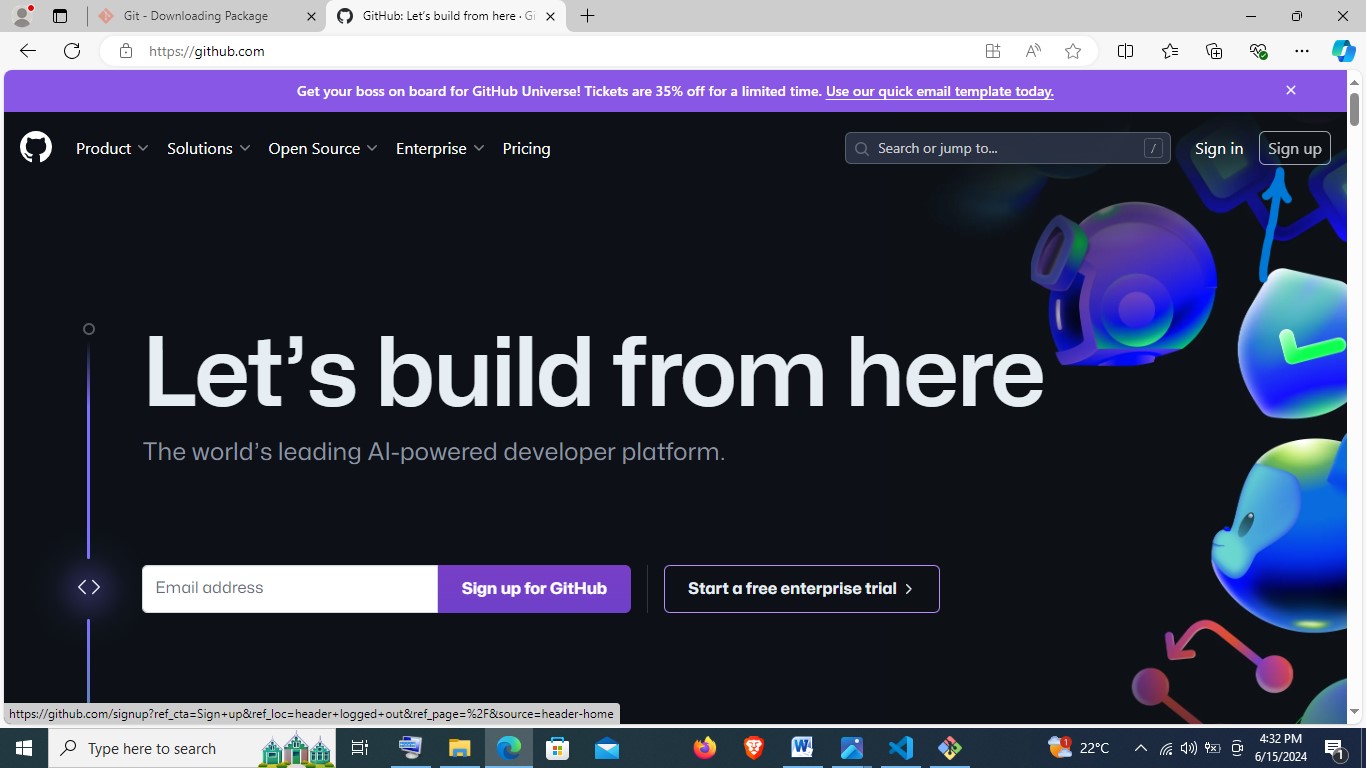
(visualstudiocode, 2020)

**Question 10: Using Source Control:**

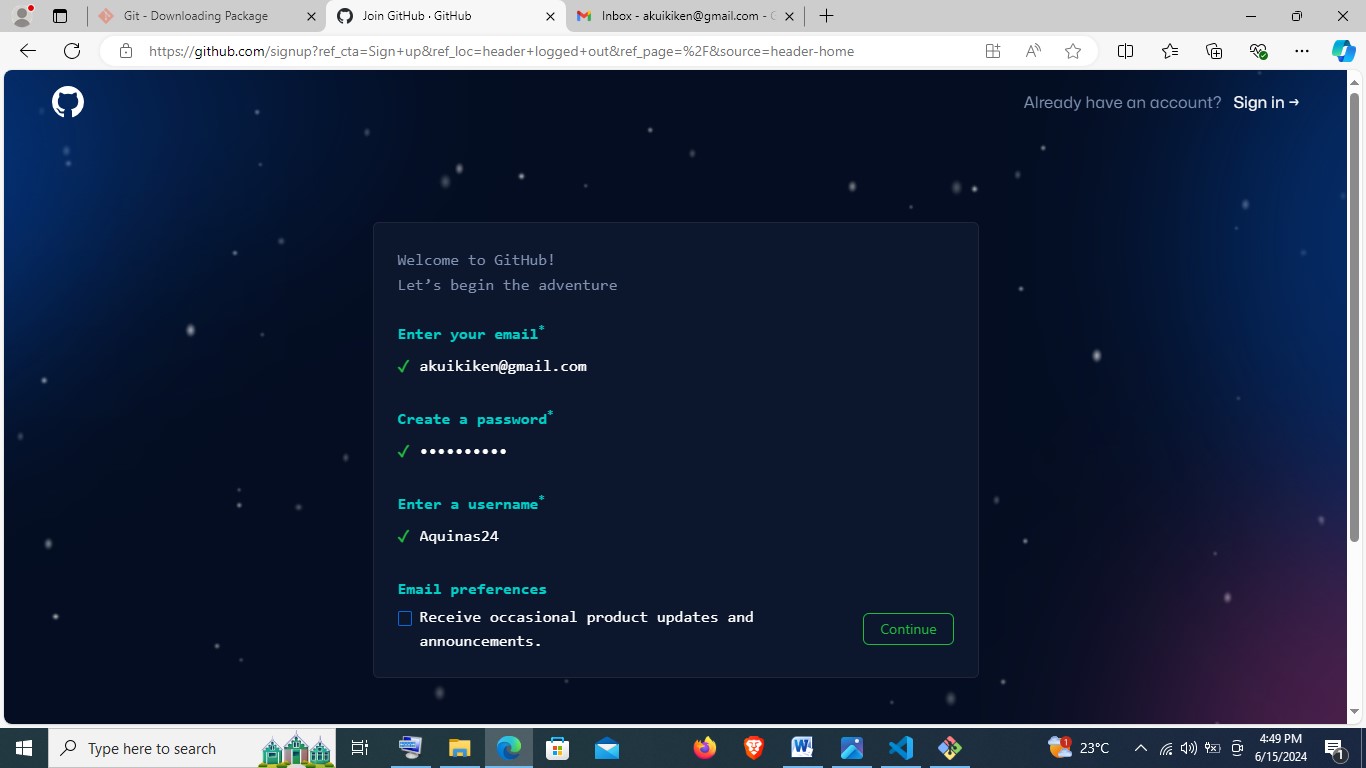
How can users integrate Git with VS Code for version control? Describe the process of initializing a repository, making commits, and pushing changes to GitHub.

**Creating a GitHub Account**

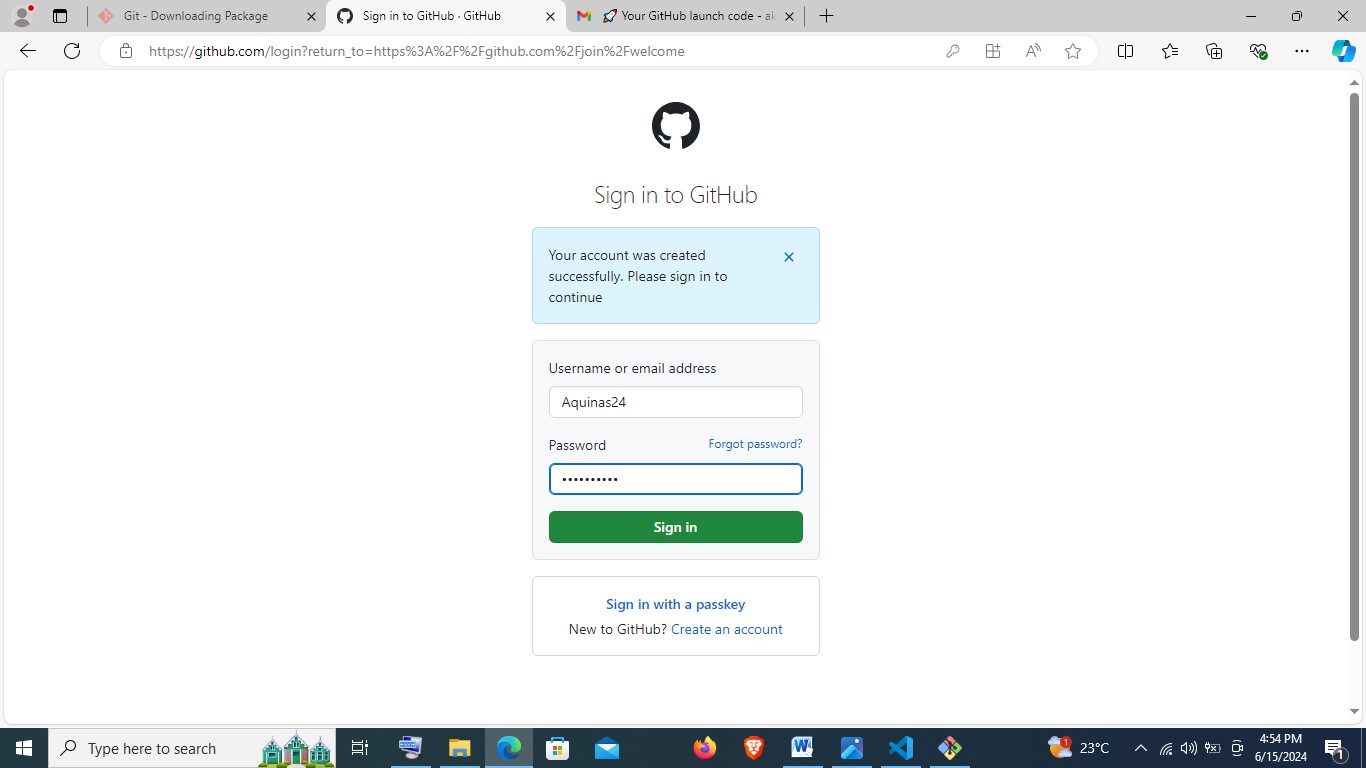
* Open the following link: <https://github.com> and click on **sign up**



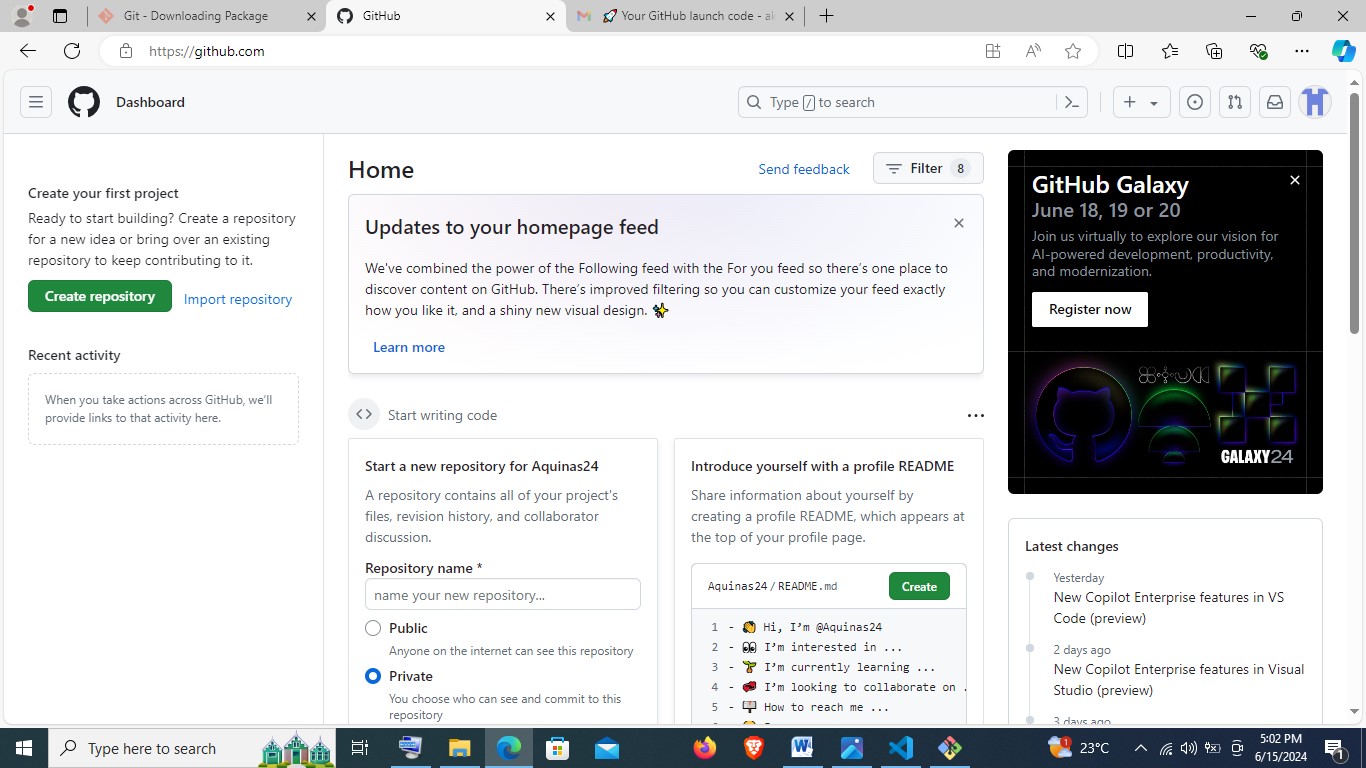
* Key in the required details for signing up as follows:



Then confirm email and sign in:



Once signed in, it will appear as below:

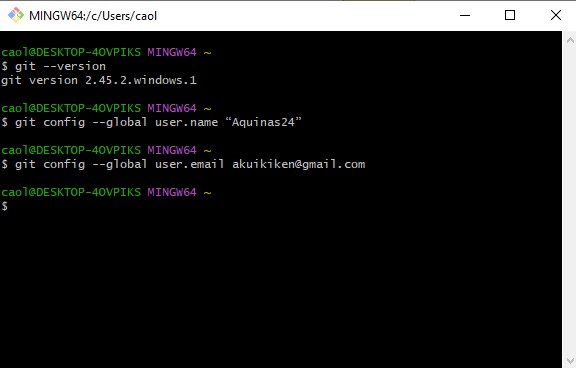


**Linking Git and GitHub**

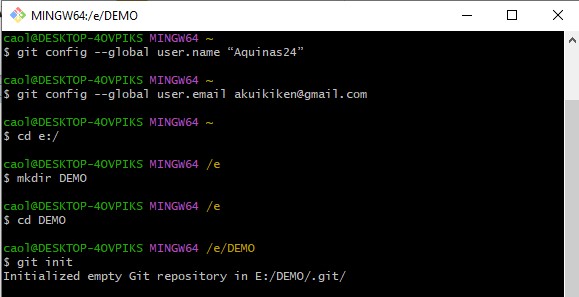
* Open Gitbash and type the following:

To configure my GitHub username and email:

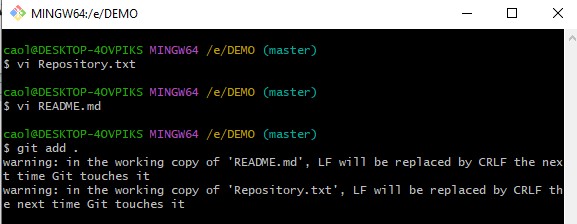
* git config - - global user.name “Aquinas24”
* git config - - global user.email [akuikiken@gmail.com](mailto:akuikiken@gmail.com)

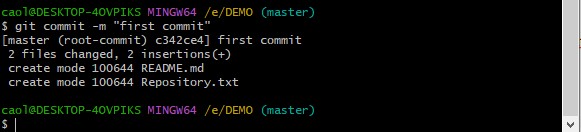


* Proceed to creating a local repository in the following steps:
* **cd e:/** to open the drive where I will create my repository.
* **mkdir DEMO** to create a new directory called **DEMO** that will contain my local repository.
* **cd DEMO** to open the directory.
* **git init** to initialize a local git repository in the **DEMO** directory.

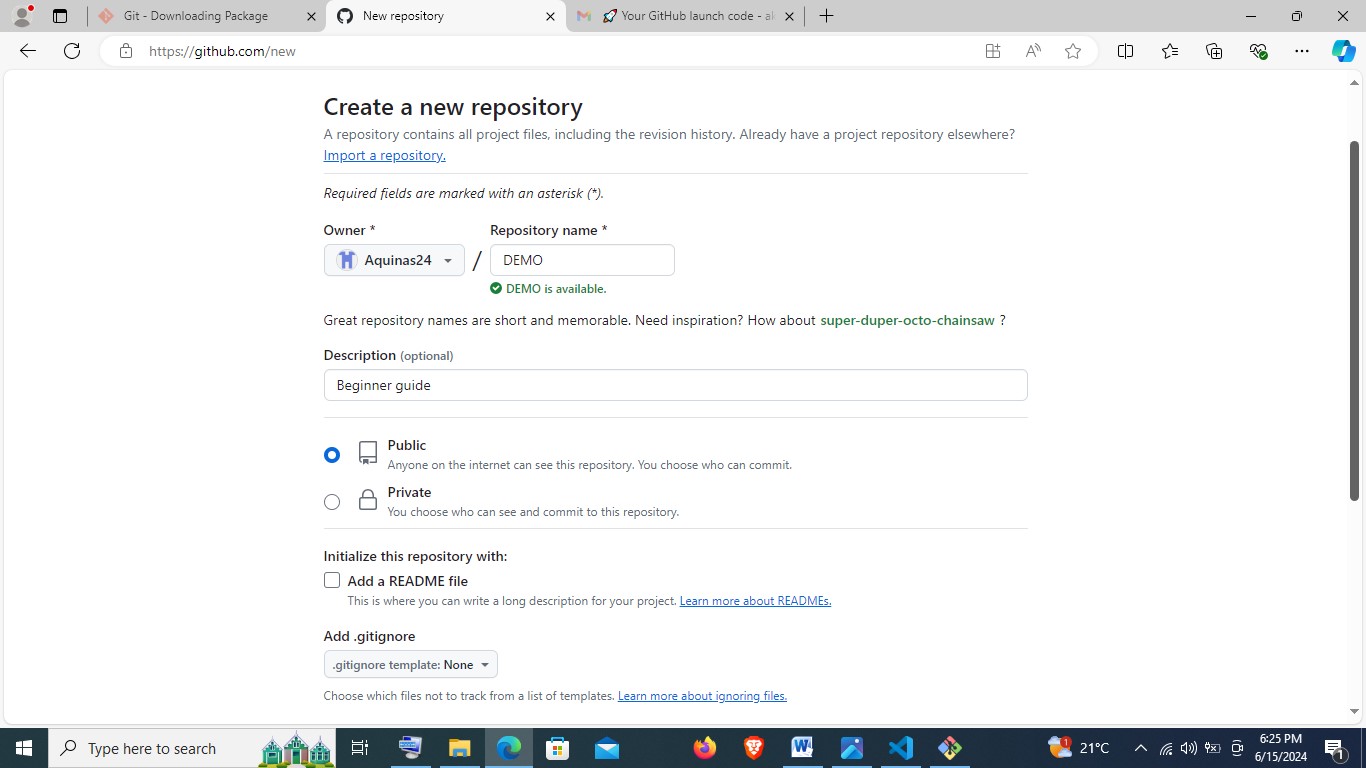


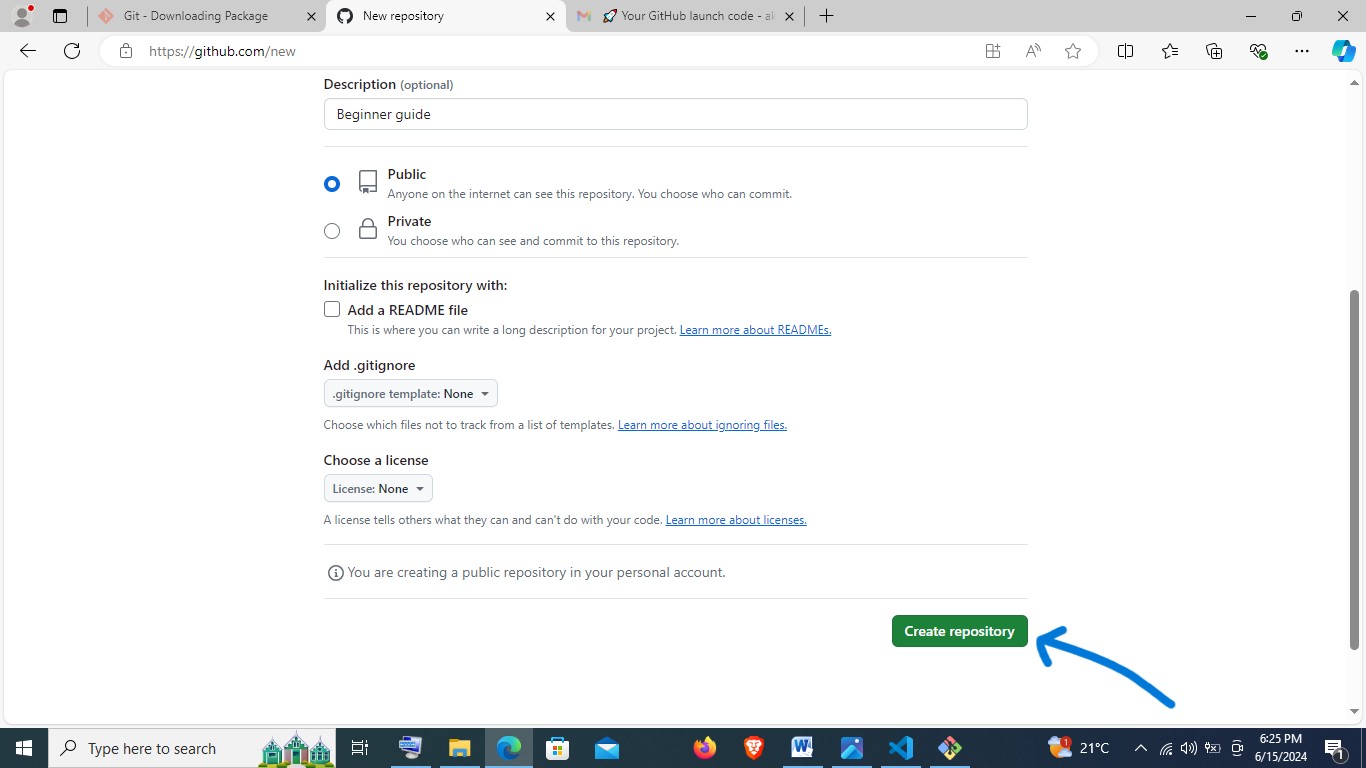
* **vi Repository.txt** to create a new file, which I can insert text by typing **i** and then close by pressing **escape, shift+colon, w, then q.**
* **vi README.md** to create a markdown file, which I can insert text by typing **i** and then close by pressing **escape, shift+colon, w, then q.**
* **git add .** to add the **Repository.txt** and **README.md**  file to git.
* **git commit –m “first commit”** to save the file on git.



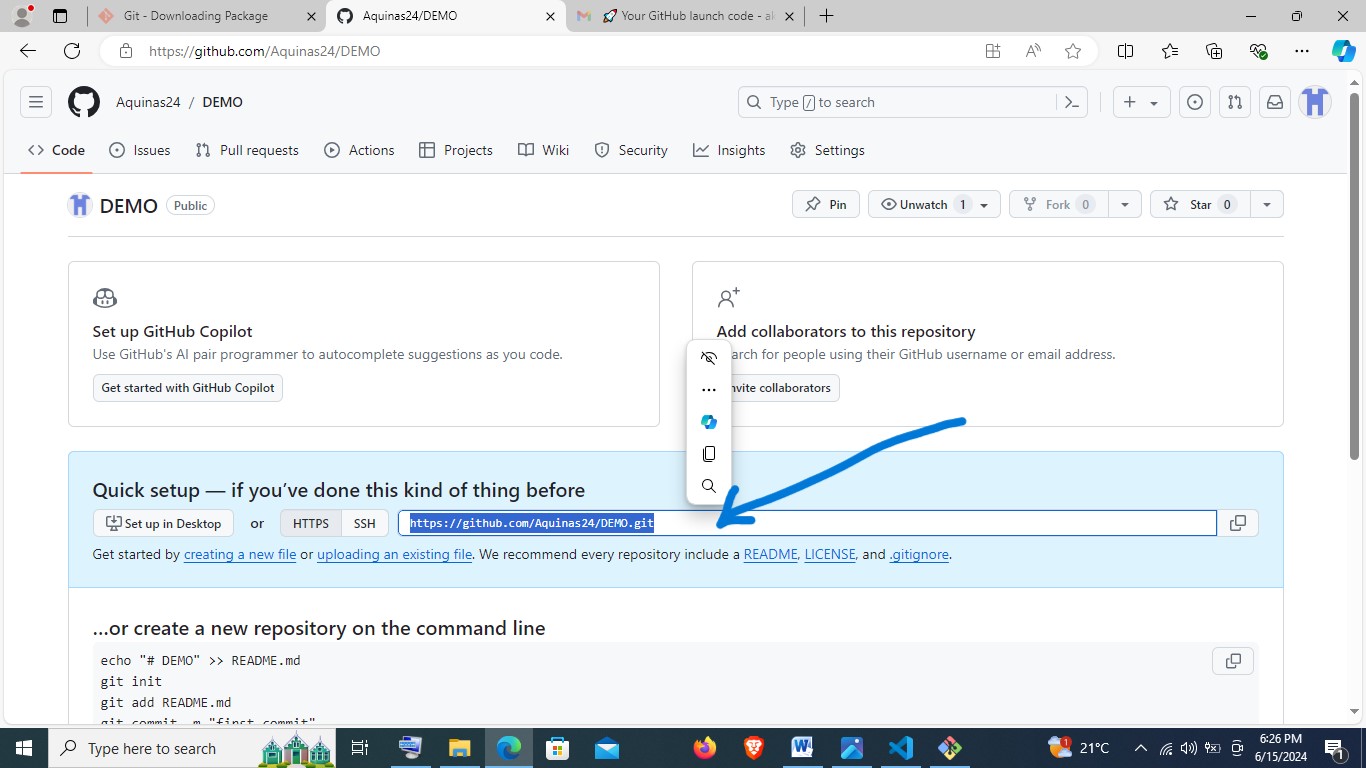


* Proceed to GitHub portal and create a new repository:

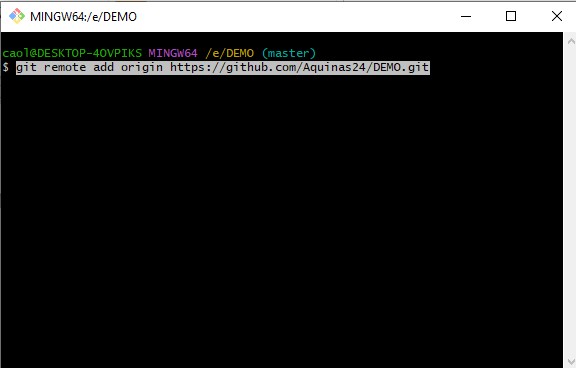




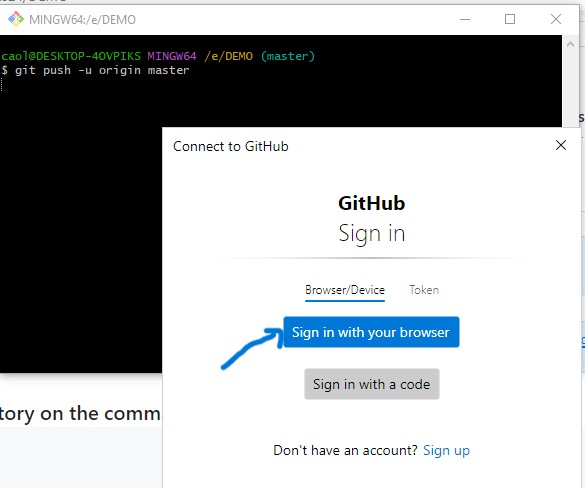
Copy the url in the empty repository as shown below:



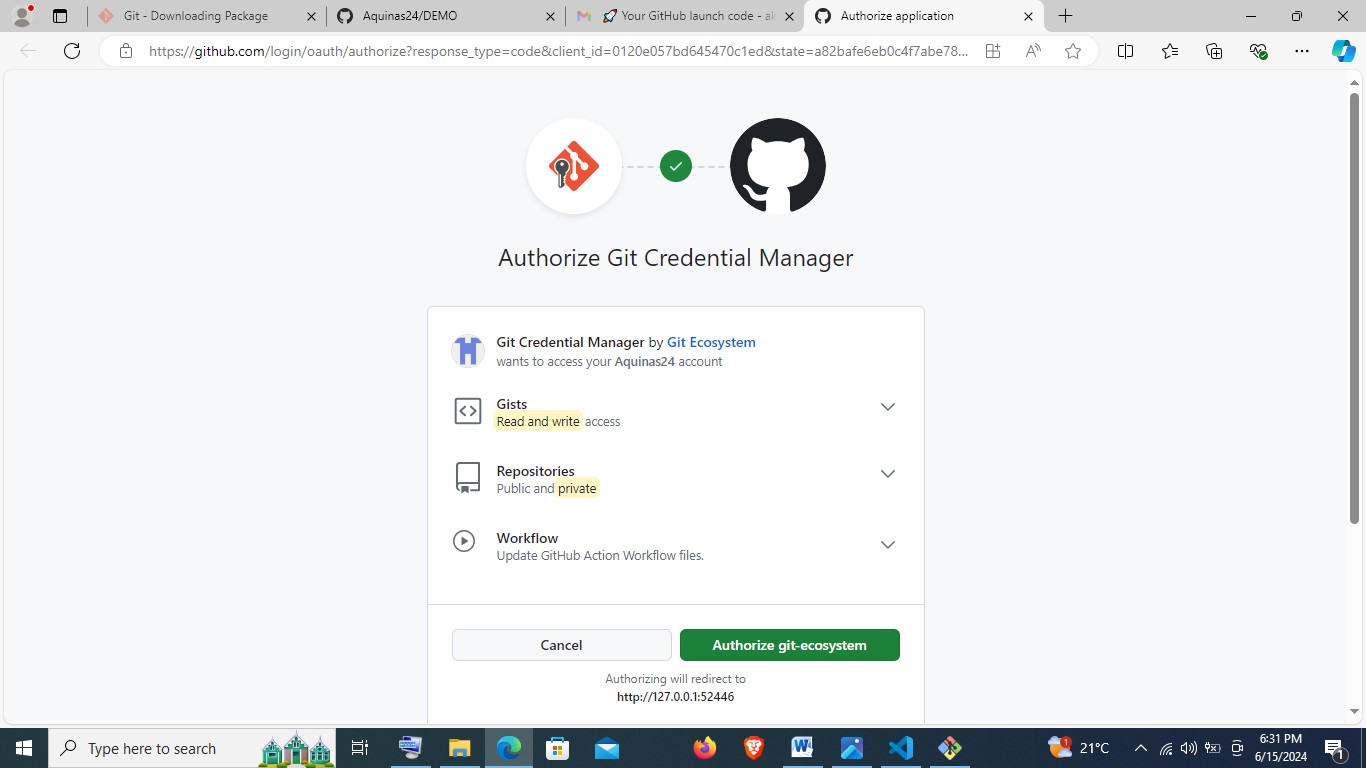
* Open Gitbash, and type the following:
* **git remote add origin** [**https://github.com/Aquinas24/DEMO.git**](https://github.com/Aquinas24/DEMO.git)
* **NB:** [**https://github.com/Aquinas24/DEMO.git**](https://github.com/Aquinas24/DEMO.git)is the link copied from the empty repository.

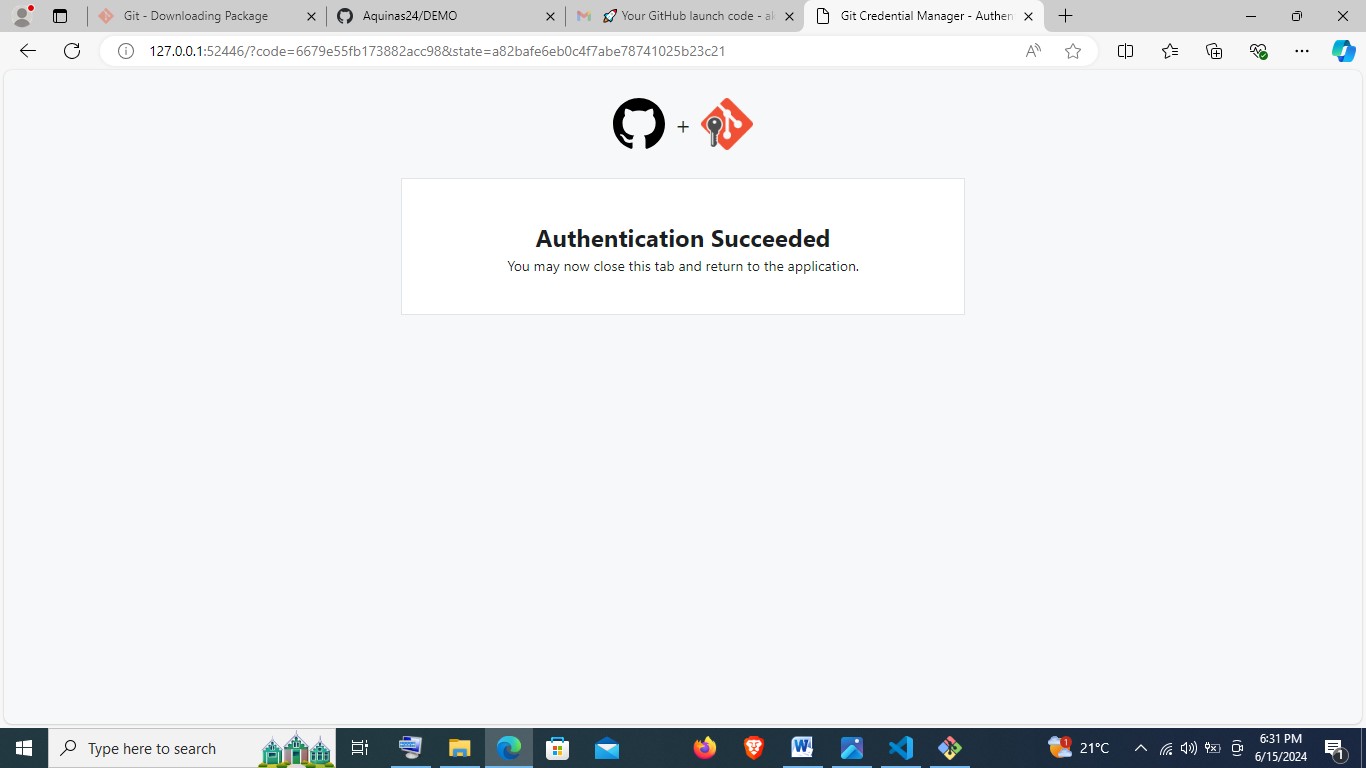


* Then proceed to type **git push -u origin master** given that my branch is master.
* This will prompt opening GitHub portal for signing in.

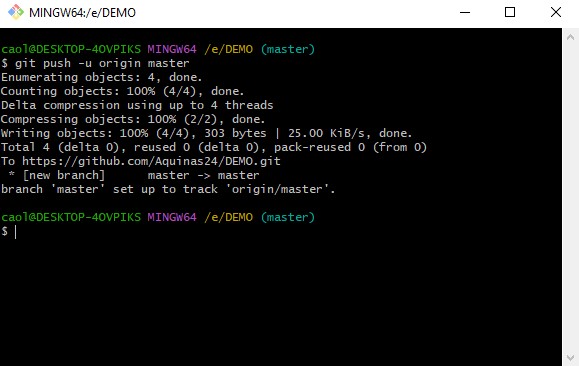


Proceed to authorize as follows:

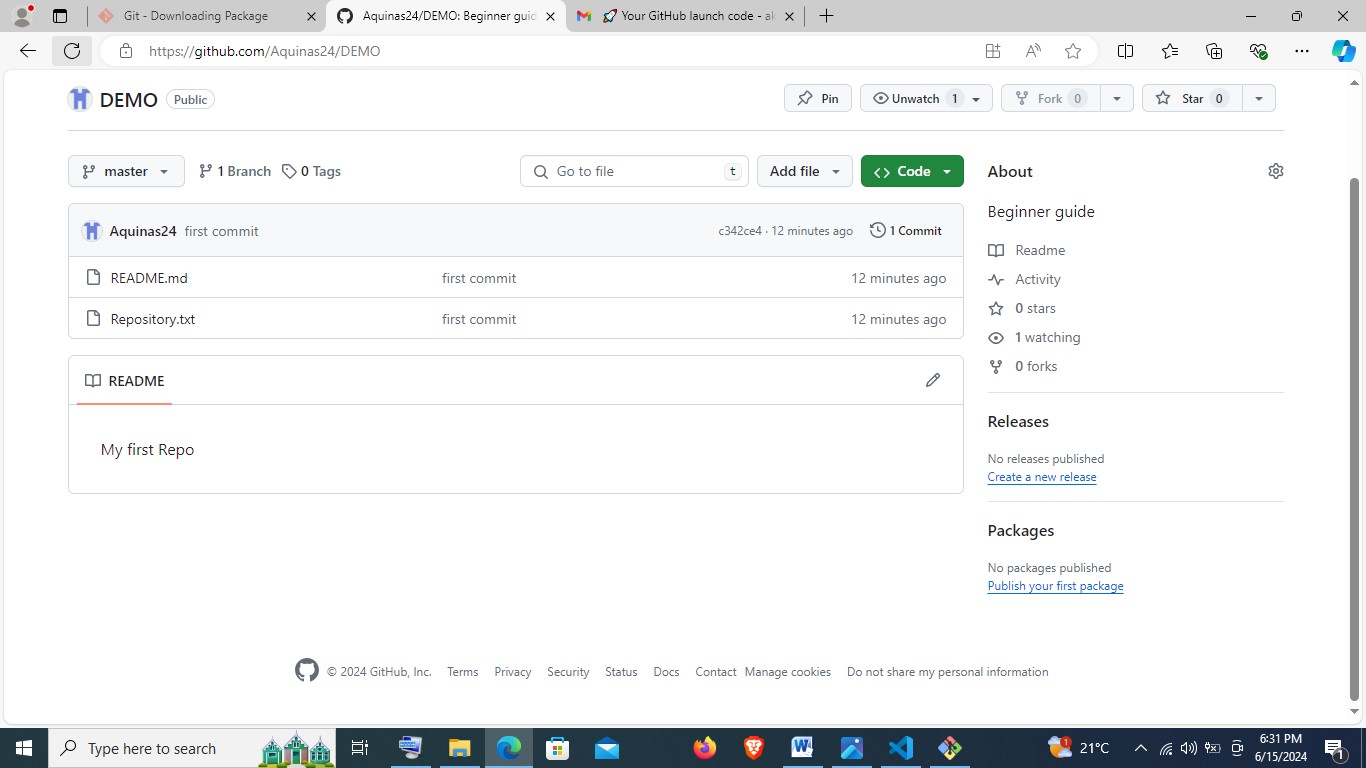




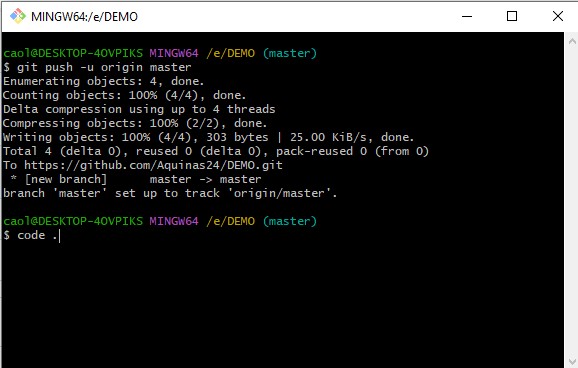
Back in Gitbash when the process is complete, it will show as below:



Refresh GitHub portal to observe the changes:



Type **code .** in Gitbash to view the contents on visual studio code:





**References**

geeksforgeeks (2022). *How to Install Visual Studio Code on Windows?* [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/how-to-install-visual-studio-code-on-windows/>.

visualstudiocode (2020). User and Workspace Settings. *Microsoft Visual Studio*. [online] 17 Aug. Available at: <https://code.visualstudio.com/docs/getstarted/settings>.

visualstudiocode (2024). *Visual Studio Code User Interface*. [online] code.visualstudio.com. Available at: <https://code.visualstudio.com/docs/getstarted/userinterface>.