**INSTALLATION OF VISUAL SOURCE CODE**

**Question 1**

1. **DOWNLOAD VS CODE**

Visit the visual source code official site so as to download a genuine app.

You click on the download button so as to get the installer

1. **RUN THE INSTALLER.**

Once the installer is downloaded, you should execute it.

Follow the prompts for successful installation.

1. **ACCEPT THE TERMS AND CONDITIONS**

Make sure you access the license agreement and read all the terms and conditions listed before accepting them.

1. **CHOOSE THE INSTALLATION LOCATION**

By default, the visual studio code installs under **c:\\users\\{username}\\App data\\local\\programs\\Microsoft vs code.**

You can specify the location which you want.

1. **ADDITIONAL TASKS**

The installer may ask the user to select additional task which he may want to be downloaded.

1. **READY TO INSTALL**

Click the install button.

1. **LAUNCH VS CODE**

Click the launch button to vs code.

**Question 2**

* **EXTENSIONS**

Install relevant extensions based on the programming language and the workflow. Examples are like:

* Docker: “Docker”, “Remote-containers”
* Python: “python”, pylance
* Git: “Gitiens”, “GitHub pull request and issues
* **SETTINGS**

Access settings by clicking the gear icon in the button left corner and select “settings”.

Important settings to consider:

* Font: set your preferred font family and size
* Editor: customize indention, line wrapping and other editor.
* **KEYBIDINGS**

Customize key bindings to match your workflow. Go to preferences or you can also use the keyboard shortcuts.

* **USER SNIPPEST**

Create custom code snip-pest for repetitive tasks. Go to preferences then go to user snip-pest and select a language.

* **INTERGRATED TERMINAL**

Set your preferred shell example Power-Shell, in the integrated terminal

* **VERSION CONTROL SYSTEM**

Initialize a git repository in your project folder if needed.

Configure your Git username and email using (git config).

* **WORKSPACE SETTINGS**

If you work on multiple projects, consider using workspace. You can access it through the following steps:

* + Go to specific settings
  + Create a vs code folder in your project
  + Add a setting. j son file

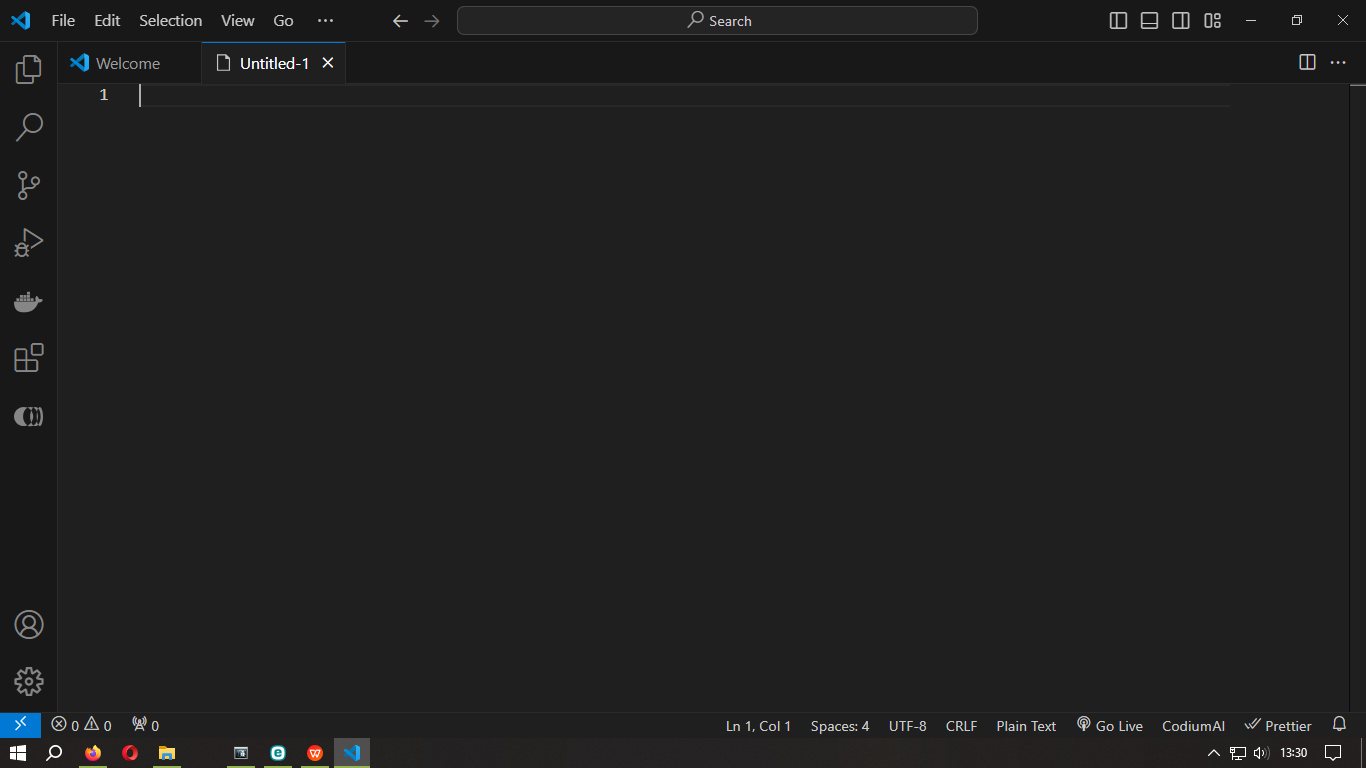
**QUESTION 3**

1. **EDITOR AREA**

The central area where you edit your files, you can open multiple editors side by side vertically.

Each editor displays the content of a file you have opened.

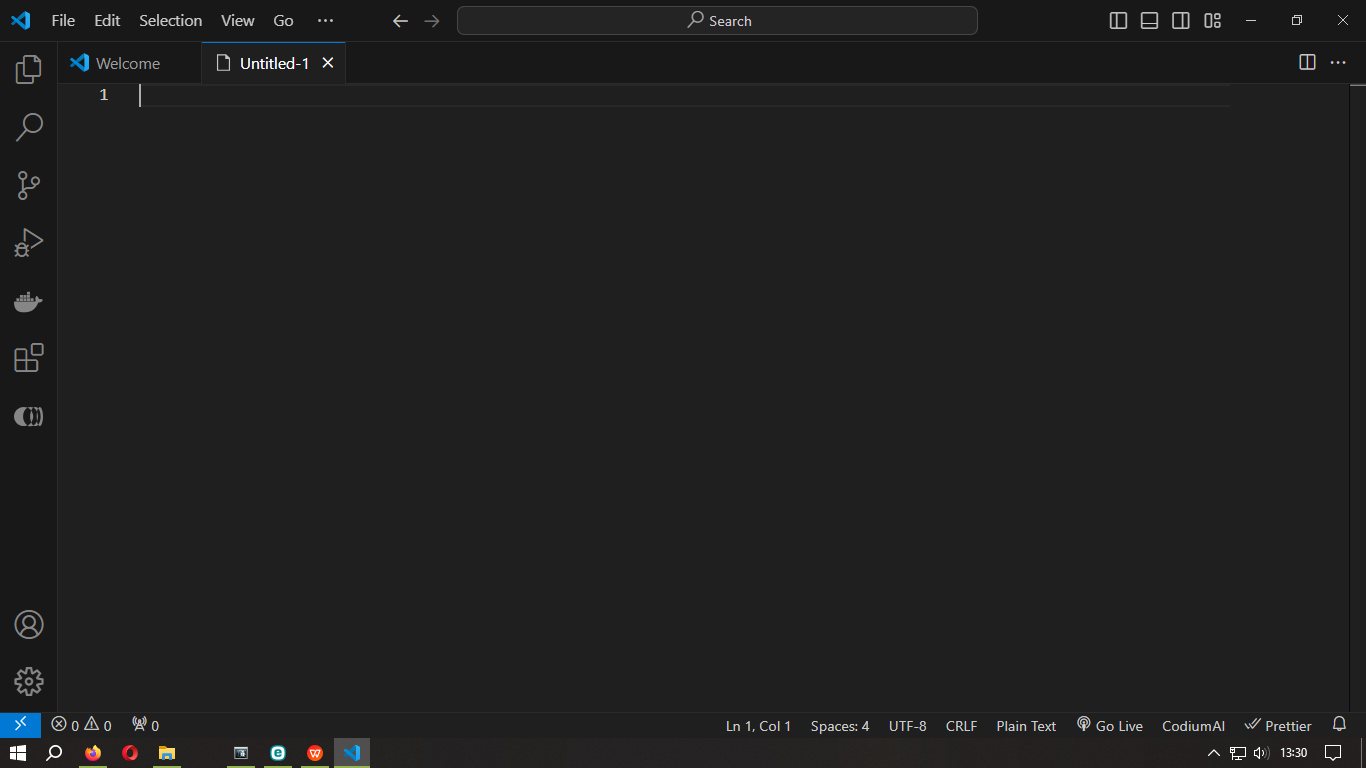
1. **SIDE BAR**
   * Located on the left hand side, the side bar contains various views to assist you while working on your project.
   * Key views include the explorer. It shows your projects files and folders and other relevant views like extensions search and source control.
2. **STATUS BAR**



* Located at the bottom, the status bar provides info about the opened project and the file you are editing.
* It displays details like the current line and column number git branch and may give extensions

**ACTIVITY BAR**





* Positioned on the far left hand side, the activity bar lies you switch between different views.
* It provides context specific indicators such as the number of ongoing changes when Git is enabled.

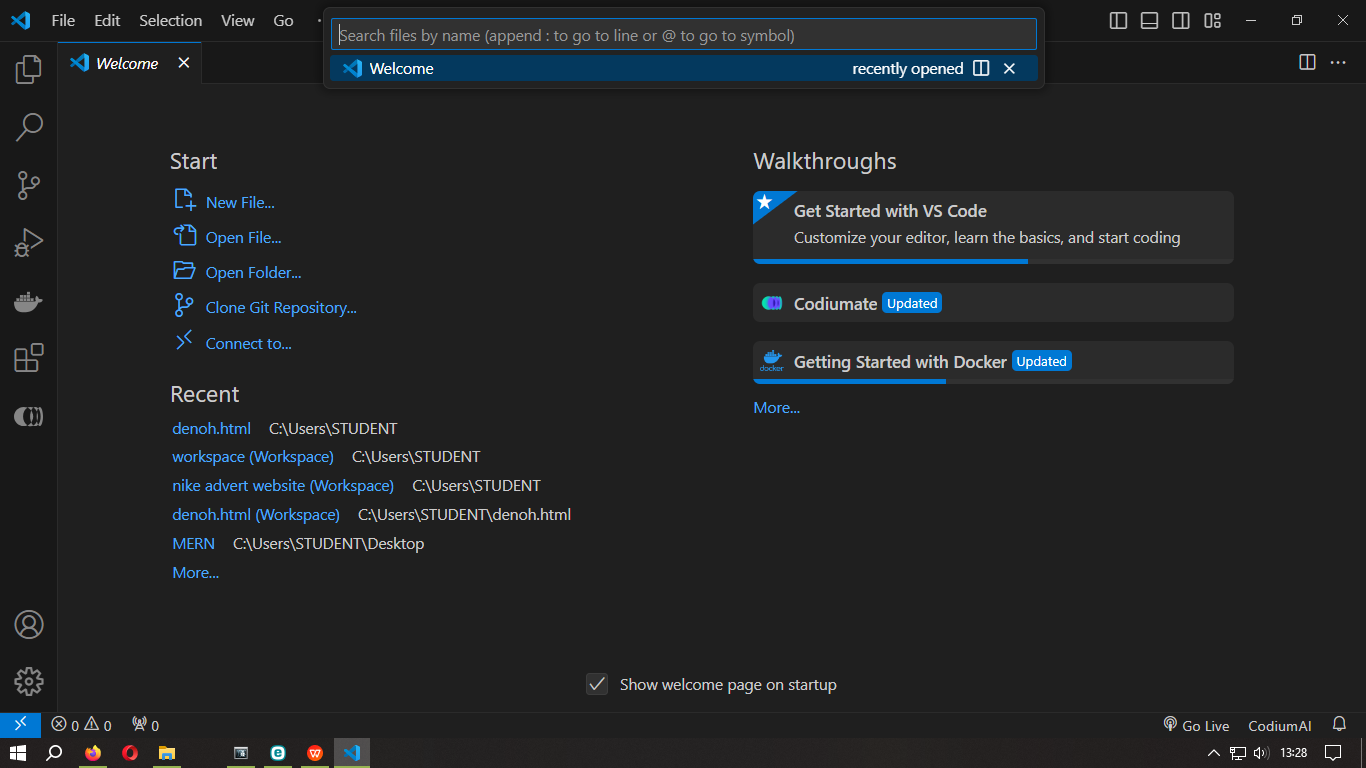
**PANEL**

* An additional space below the editor region.
* By default it contains output, debug info, errors warnings.
* You can move the panel to the left or right for more vertical space.

**QUESTION 4**

**COMMAND PALLET**

* It is a tool that allows you to access all commands. It can be accessed through the following short cut **ctrl + shift + p**



**OPEN FILE**

* Quickly open a specific file in the editor by typing in the name or part of it.

**SEARCH SYMBOLS**

* Find specific symbols (eg, functions) within your code base.

**RUN TASK**

* Execute predefined tasks such as building, debugging or running test for your project.

**MANAGE EXTENSIONS**

* Install, update or remove extensions directly from the command palette.

**QUESTION 5**

**ESSENTIAL EXTENSIONS FOR WEB DEV**

* Pretter – Automatically format your code for constituency.
* Live server – Launch a local dev server with live reloading.
* HTML CSS support – Provides auto completion and syntax highlighting for HTML and CSS.
* ESLint – Lint your JavaScript / Typescript code for that stream lines your coding workflows.

**QUESTION 6**

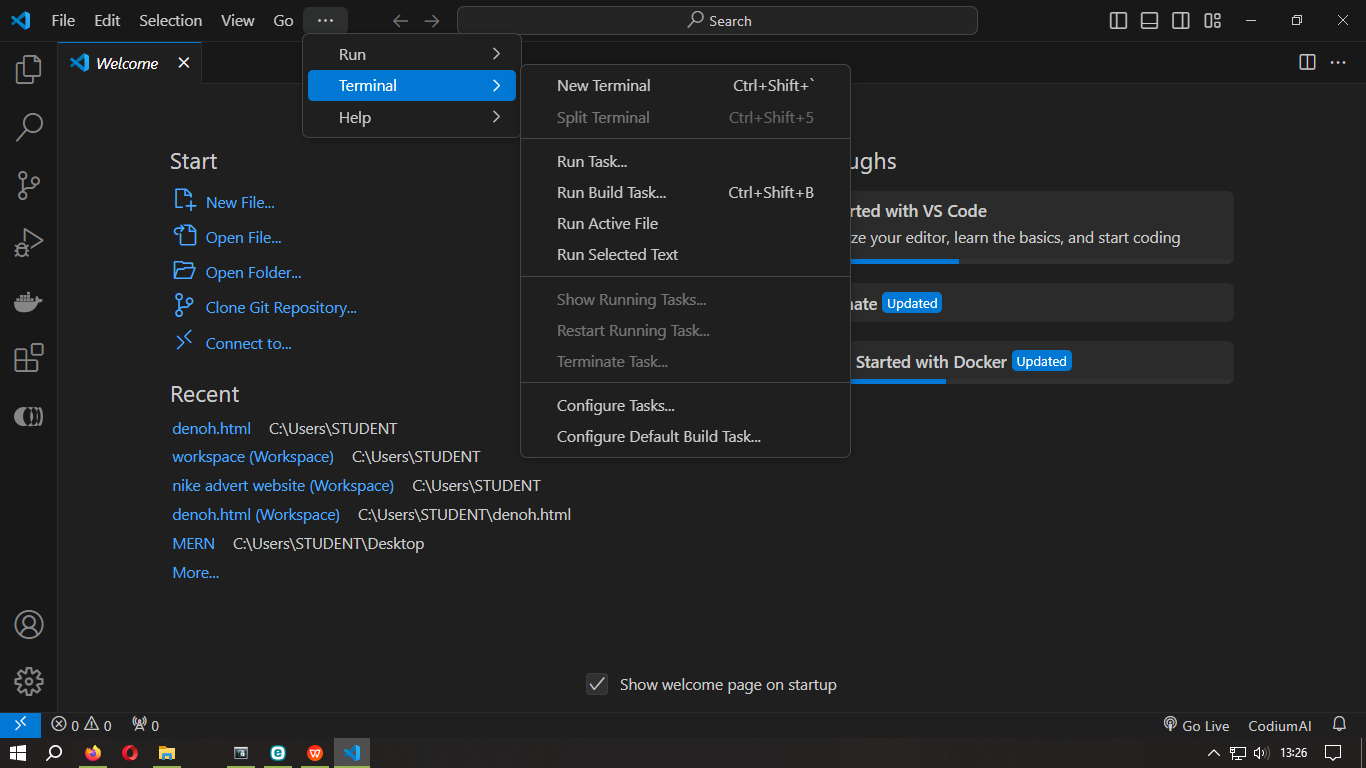
**INTERGRATED TERMINAL**

* Is a powerful tool that stream lines you are coding workflows.

**OPEN**

* You have a few options to open the integrated terminal.
* Use the keyword shortcut **shift+Back-tick**
* From the menu go the **terminal and then new click on the new terminal or view terminal.**







* In the explorer right click on the folder and choose open the interdenominational.

**ADVANTAGE**

* Unicode and emojis support. You can even use emojis in your terminal commands.
* Workspace context, It starts at the root of your workspace making it convenient for project specific tasks.
* Customization, Customize keybindings, appearance and other setting for personalized experience.
* Seamless integration, It is tightly integrated with the editor, you can easily switch between code and terminal without leaving the vs code window.

**QUESTIION 7**

**File and folder management.**

**OPENING A FOLDER AS WORKSPACE**

* To open a folder in vs code you should use the file and then click open folder and then menu.
* Alternatively, if you launch vs code from a terminal pass the folder path as the first argument to the code command.
* Vs code automatically keeps track of configurations for the folder you open.

**WORKSPACE TYPES**

* Single folder workspace:
* When you open single folder, it becomes a workshop space. No additional steps needed.
* Multi-root workspace

**MANAGING FILES AND FOLDERS**

Use the explorer to view

* Create new files and folders using the new file and new folder buttons.
* Delete and rename files and folders.
* Move files and folders via drag and drop

**EFFICIENT NAVIGATION**

Use the explorer view to quickly switch between files and directories.

Utilize keyword shortcuts like **ctrl+Tab** to cycle through open file.

Use the breadcrumbs at the top of the editor to navigate within a file.

**QUESTION 8**

**SETTING AND PREFERENCE**

1. **Open the setting editor**

Navigate to file > preferences> settings.

Keyboard shortcut **ctrl+**

1. **USER SETTING**

It is a personal preference for customizing vs code,

In the setting editor, you will find various categories like keybindings, themes and more.

**Example**

1. **Change Theme**

Search for “font size” in the search bar.

Modify the “editor” setting your preference.

1. **Customize keybindings**

* Look for keybindings in the research bar.
* Click “edit keybindings” to customize keybindings.

1. **Workspace setting**

* They are specific to the projects you are working on.
* To access them, open a workspace folder and create a **vs code** folder if it does not exist.
* Inside that folder, create a **setting.j son**  file to define a workspace, specific setting.

**A vs code provides extensive customization options hence becomes easy to use.**

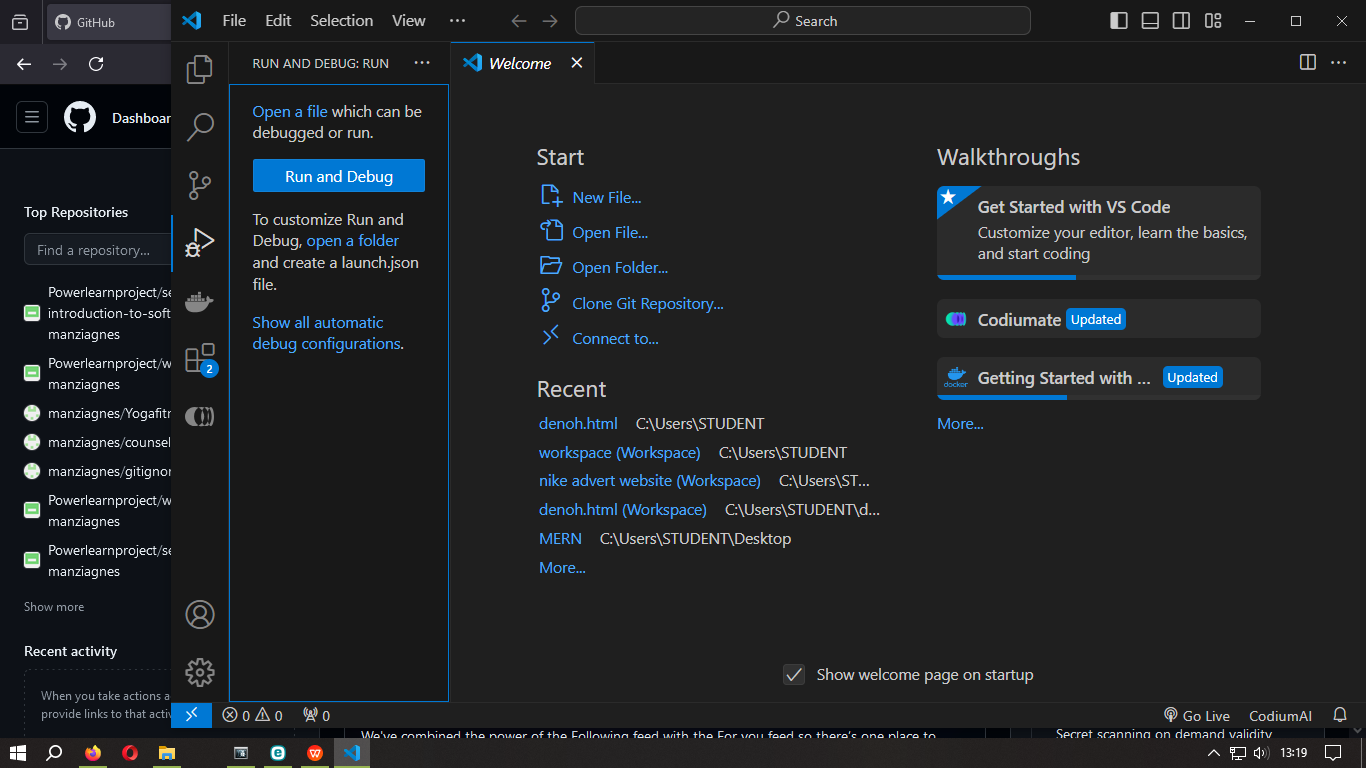
**QUESTION 9**

**DEBUGGING IN VS-CODE**

1. **Run and debug view.**

* Open the run and debug view by clicking the debugging icon in the activity bar on the side of vs-code.







* Keyboard shortcut **ctrl+shift+d**

1. **Launch configuration**

* Create a launch .j son.file
* Vs-code keeps launch configuration information in a vs-code folder with your workspace.
* **To create a launch j son. File**

1. Click Run and Debug in the debug start view.
2. Vs-code will try to detect your debug environment automatically.
3. If not choose it manually.
4. **Break point**

* Set break points by clicking on the editor margin.
* Use breakpoints to pause execution at specific lines for inspection.

1. **Data inspection**

* Inspect variable, watch expressions and call stack in the debug sidebar.
* Evaluate expression using debug console

1. **Debug actions**

* Use the debug tool bar to start step by step through the code.
* Explore features like step over, step into and many more/

**Vs-code runs debugging for every language.**

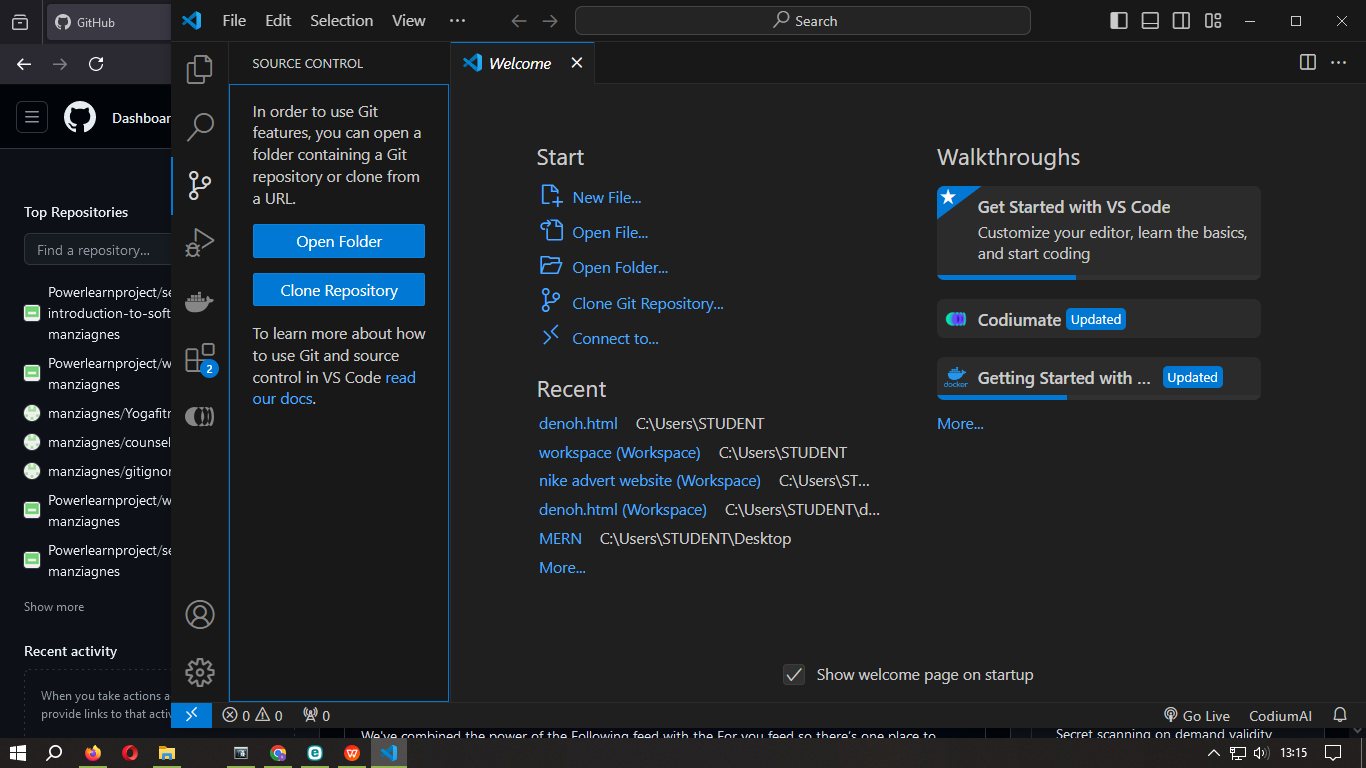
**QUESTION 10**

**USING SOURCE CONTROL**

1. **Install Git**

Ensure you have git installed in your computer.

1. **Clone repository**
2. **To clone an existing repository use the following steps:**
3. Use the Git clone command from the command palate. You can use the keyboard shortcut **ctrl+shift+p**
4. Authenticate if cloning from GitHub.
5. Select a repository to clone.





**B.To initialize a new local Git repository use the following steps:**

1. Open a folder in Vs-code.
2. In the source control view, click initialize repo button.
3. This creates a new git repo in the current folder.
4. **Make code changes in your file.**

* Open the source control view( **ctrl+shift+G)**
* Stage your changes by clicking the “+” icon next to each file.
* Enter a commit message and click the check mark icon to commit.(**ctrl+enter)**

1. **Push to GitHub**
2. Create a git hub repo if you have not.
3. In Vs-code, click the publish to GitHub button in the source control view.
4. Choose a name, description, privacy setting of your repo.
5. Vs-code pusehes your local code to GitHub.