



Module 7: Overview of Python



~ ABHIJIT ZENDE

1. In Local Windows Machine Install and setup Environment Variable for Python  
2. Install Visual Studio Code and Install Python and Terraform Extensions in VS Code 
3. Create Python Console Application to randomly generate OTP kind of secure code 
4. Create Python Console Application to read the contents of .json file and print in the VS Code python console output 
5. Create Python Web Application to using Flask Web Application Framework  

Detailed notes & Guide on my GitHub



<https://github.com/Abhiz2411/Python-101-for-Devops-Enthusiasts>

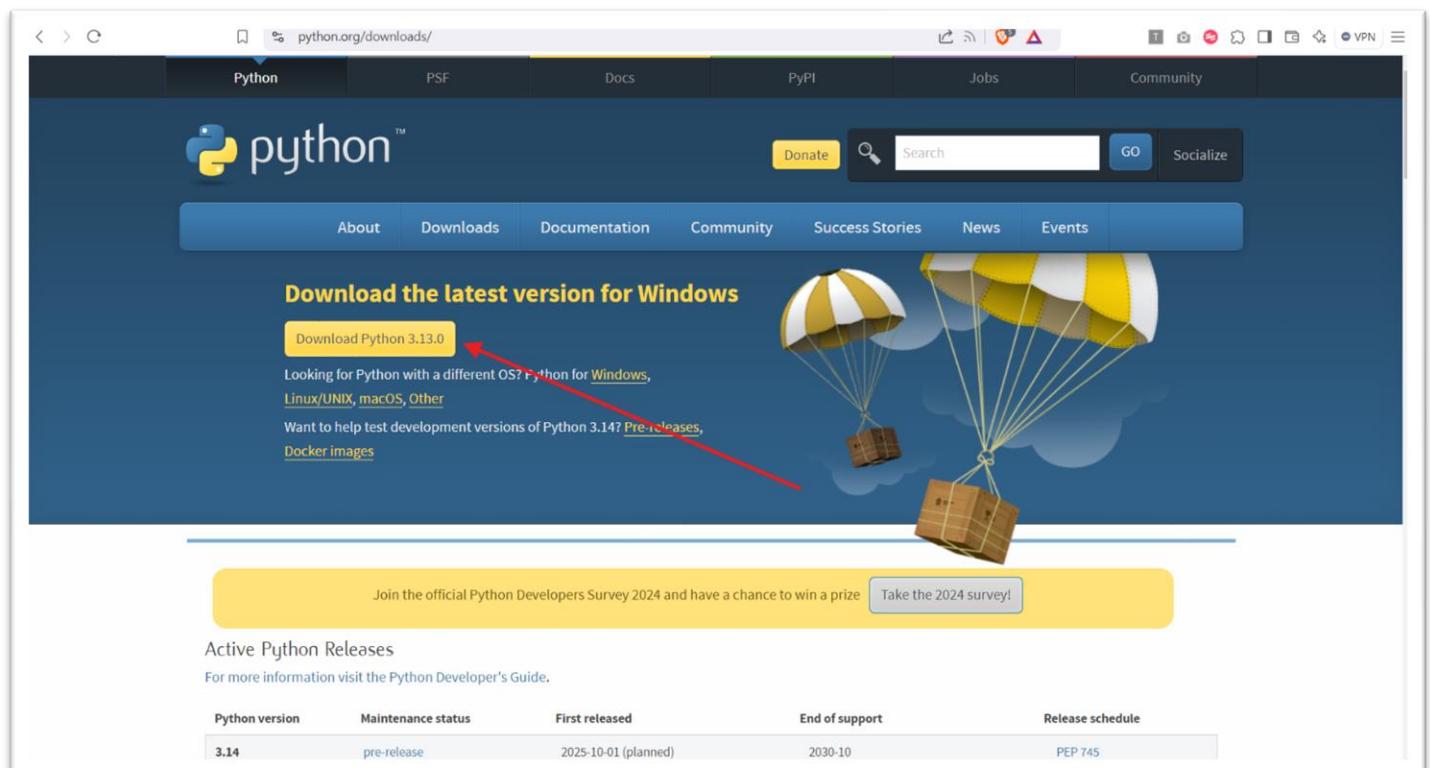


1. L1 – In Local Windows Machine Install and setup Environment Variable for Python

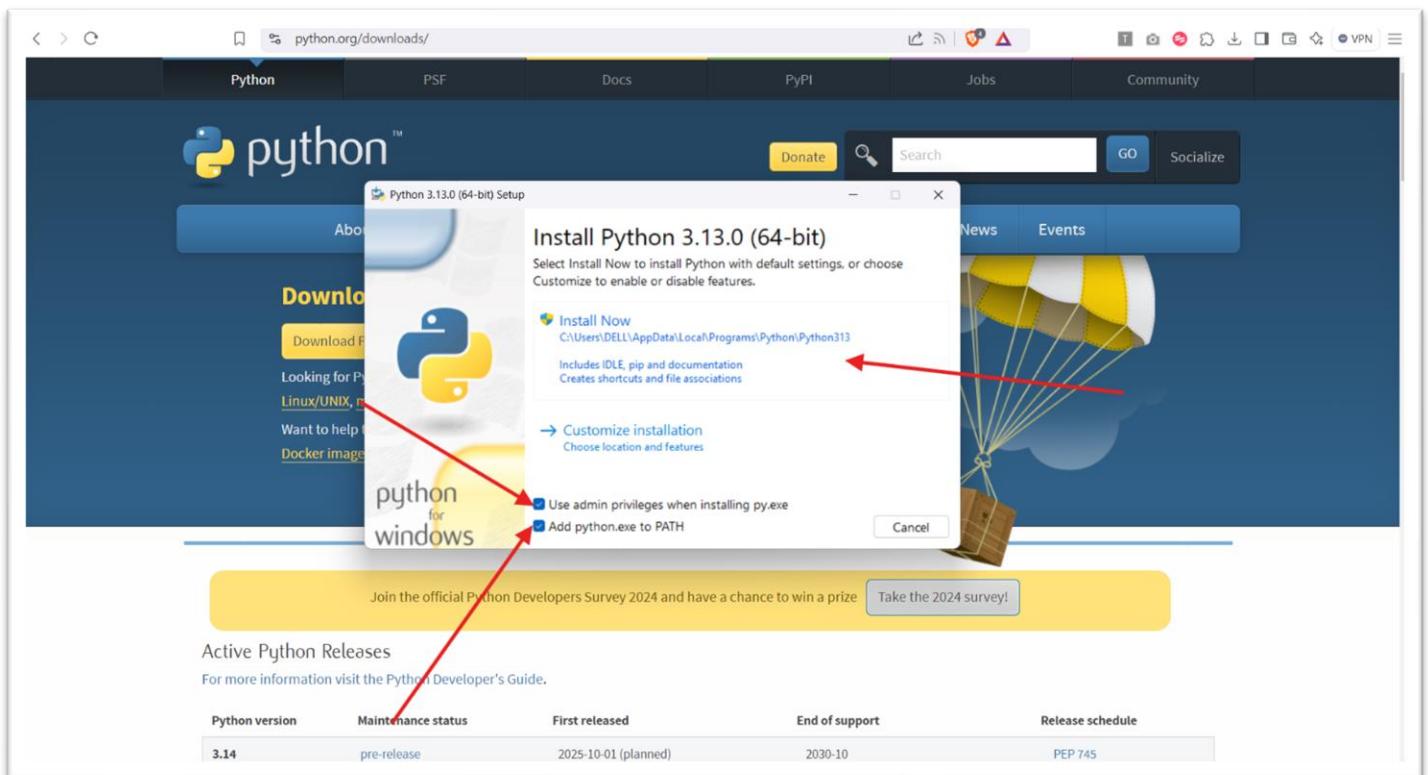
Ans.

(*** Note: Screen shots attached to end of each question ***)

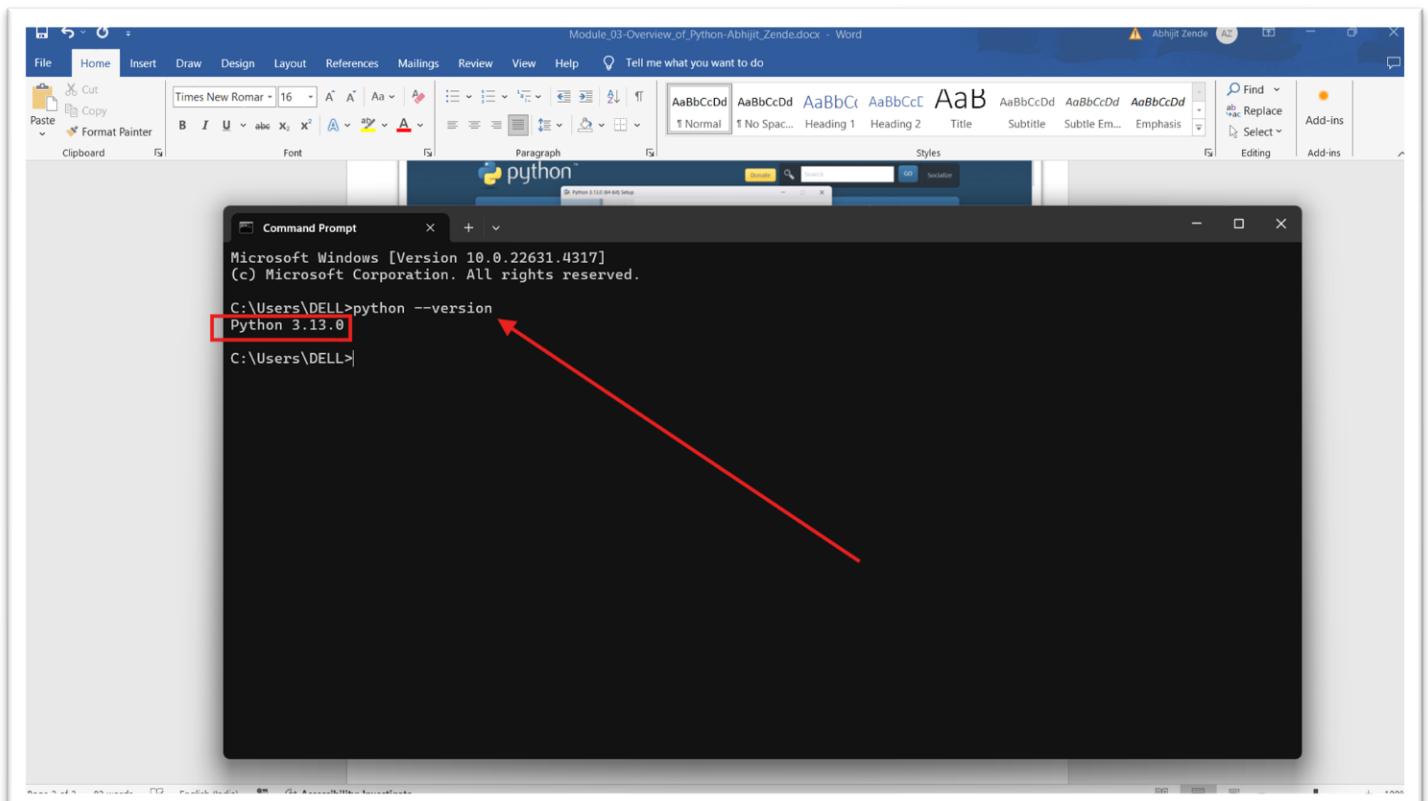
Step 1: Download the latest Python installer from <https://www.python.org/downloads/>



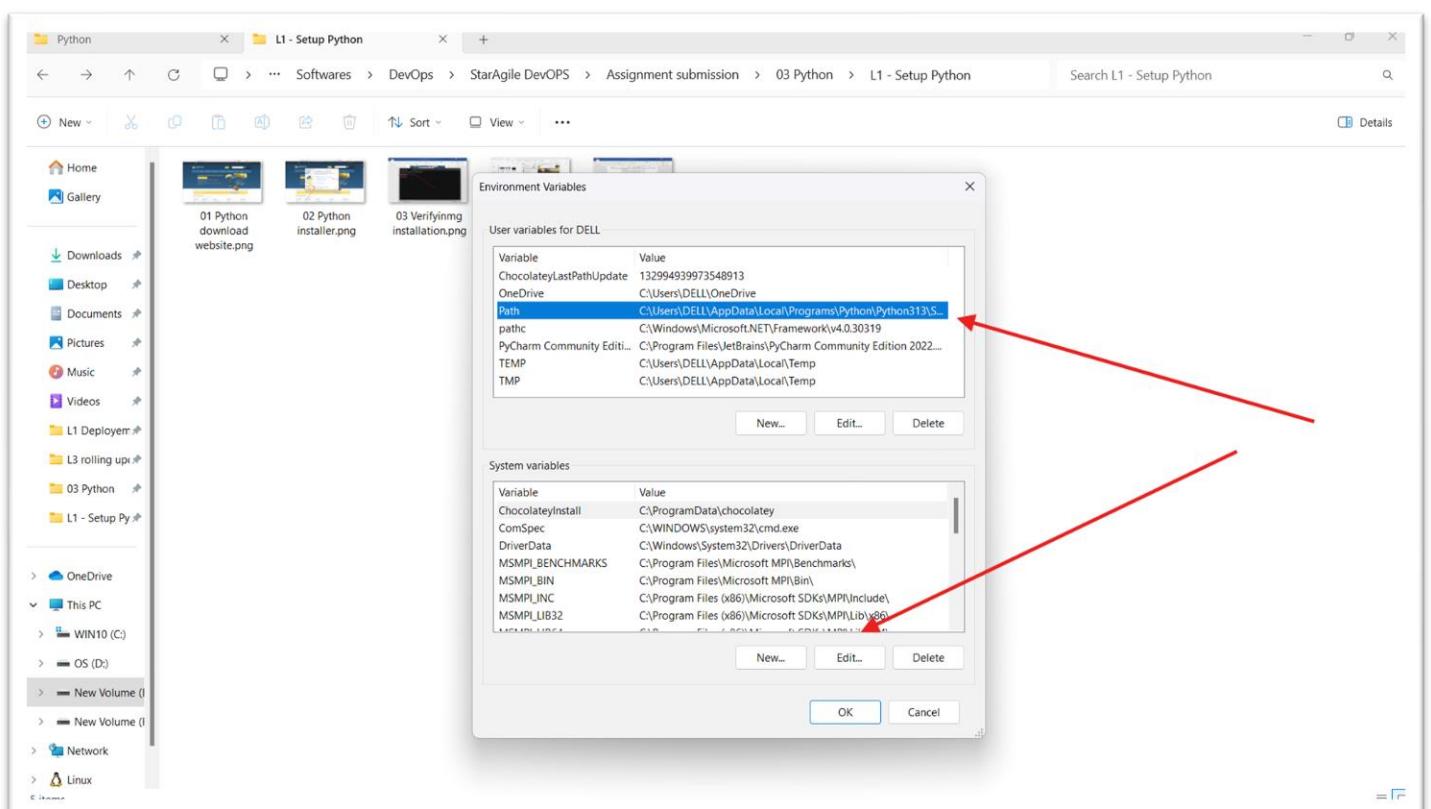
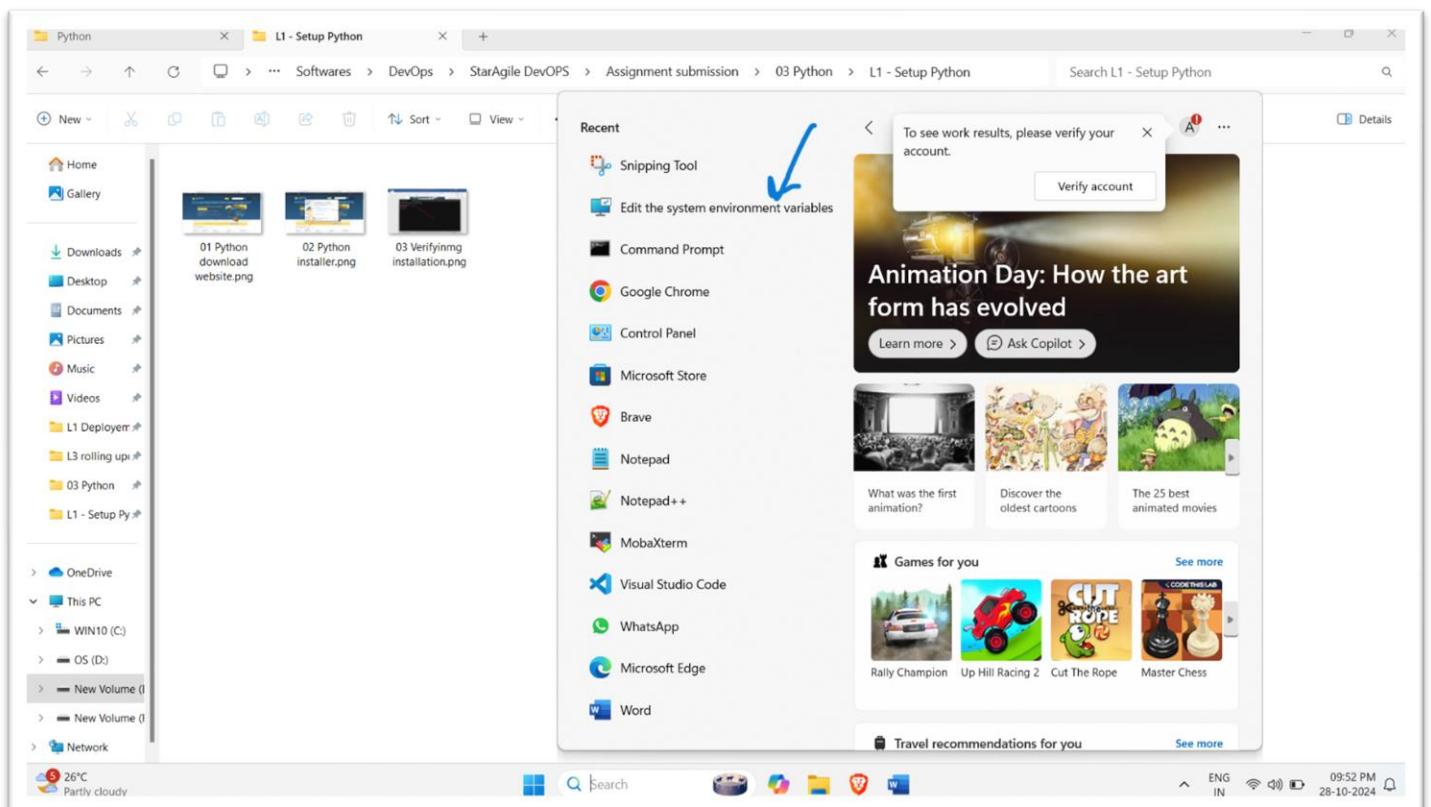
Step 2: Run the installer and check "Add Python to PATH" during installation.

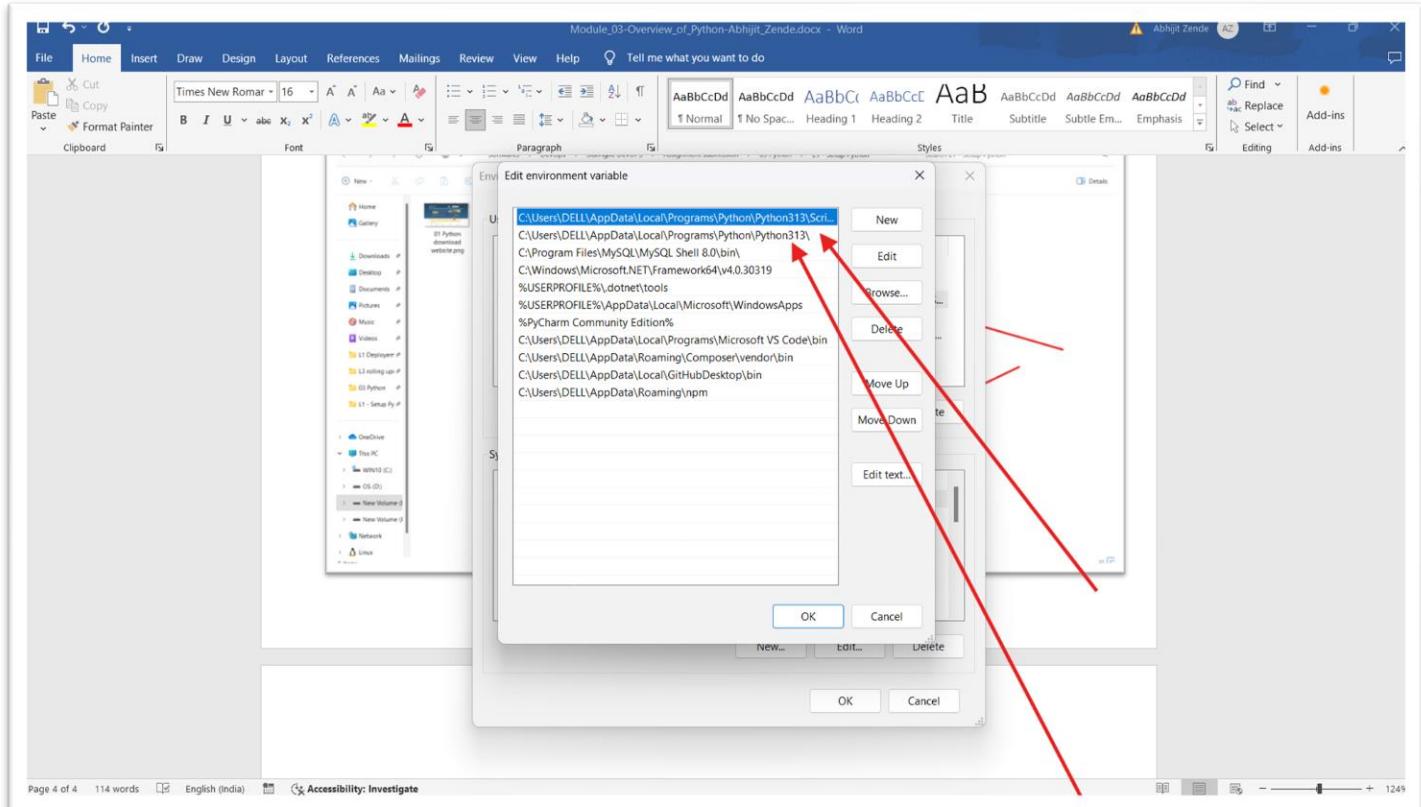


Step 3: Open Command Prompt and verify installation with `python --version`



Step 4: Verify if Python has been added to the System environment variables by searching ‘**Edit Environment variables**’ in windows search bar and click on ‘**Path**’ and ‘**Edit**’ option for path.

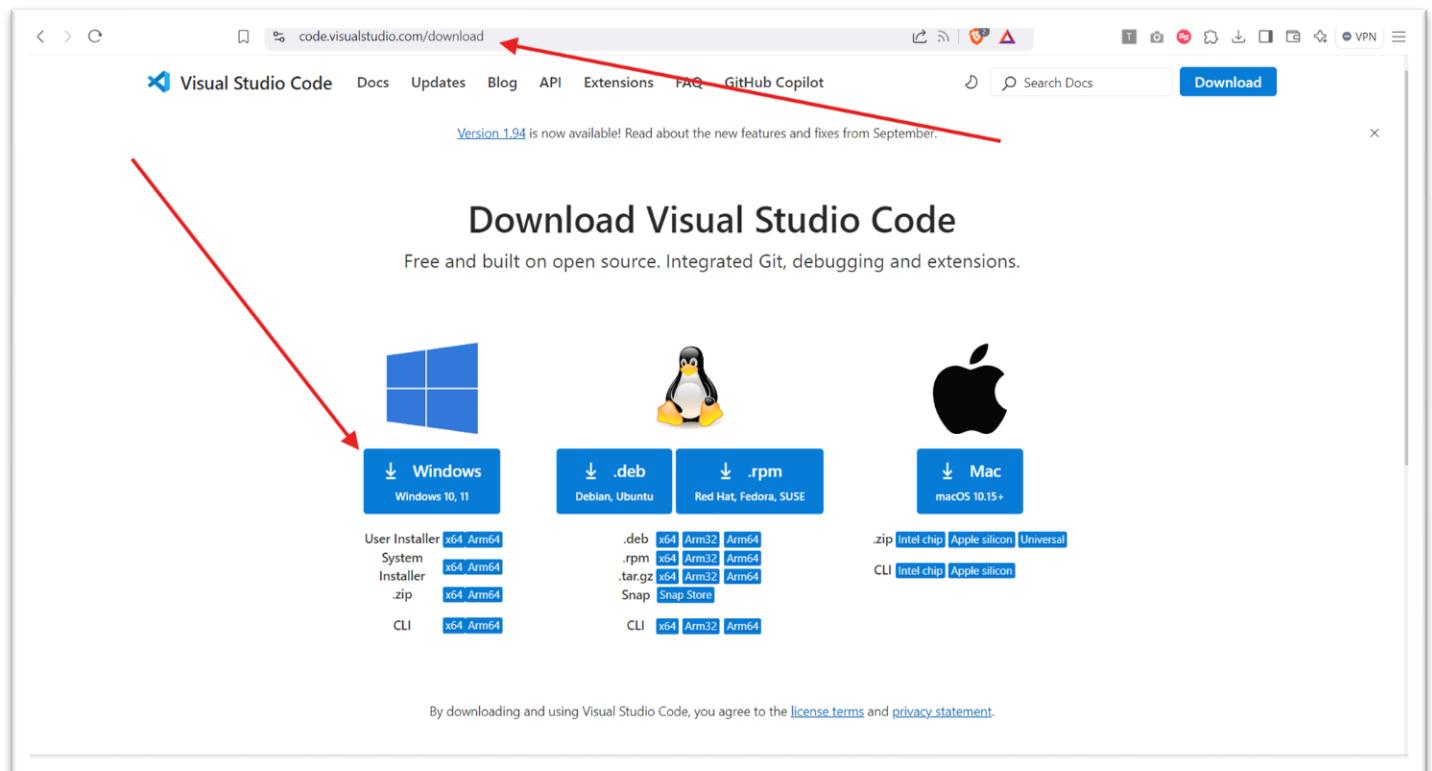




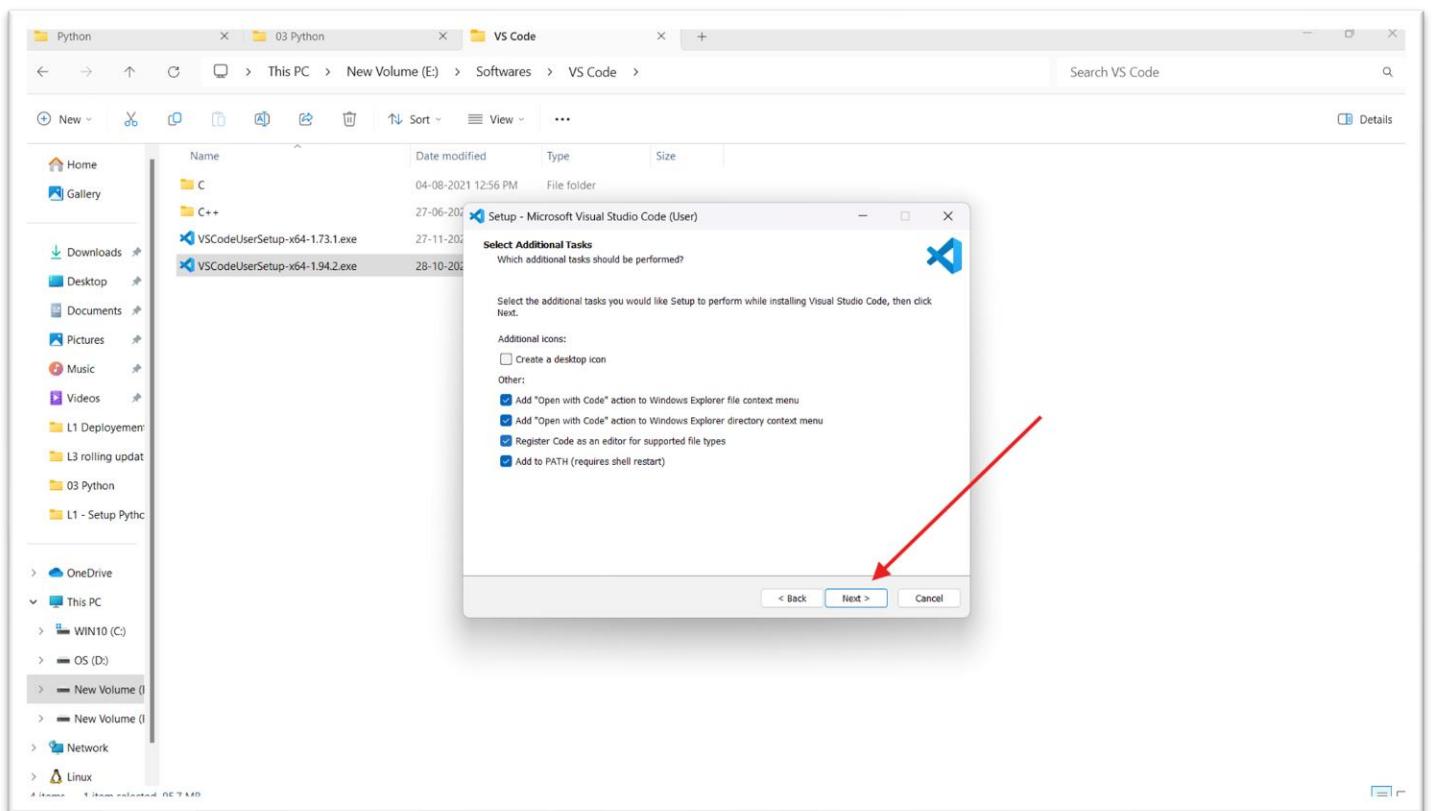
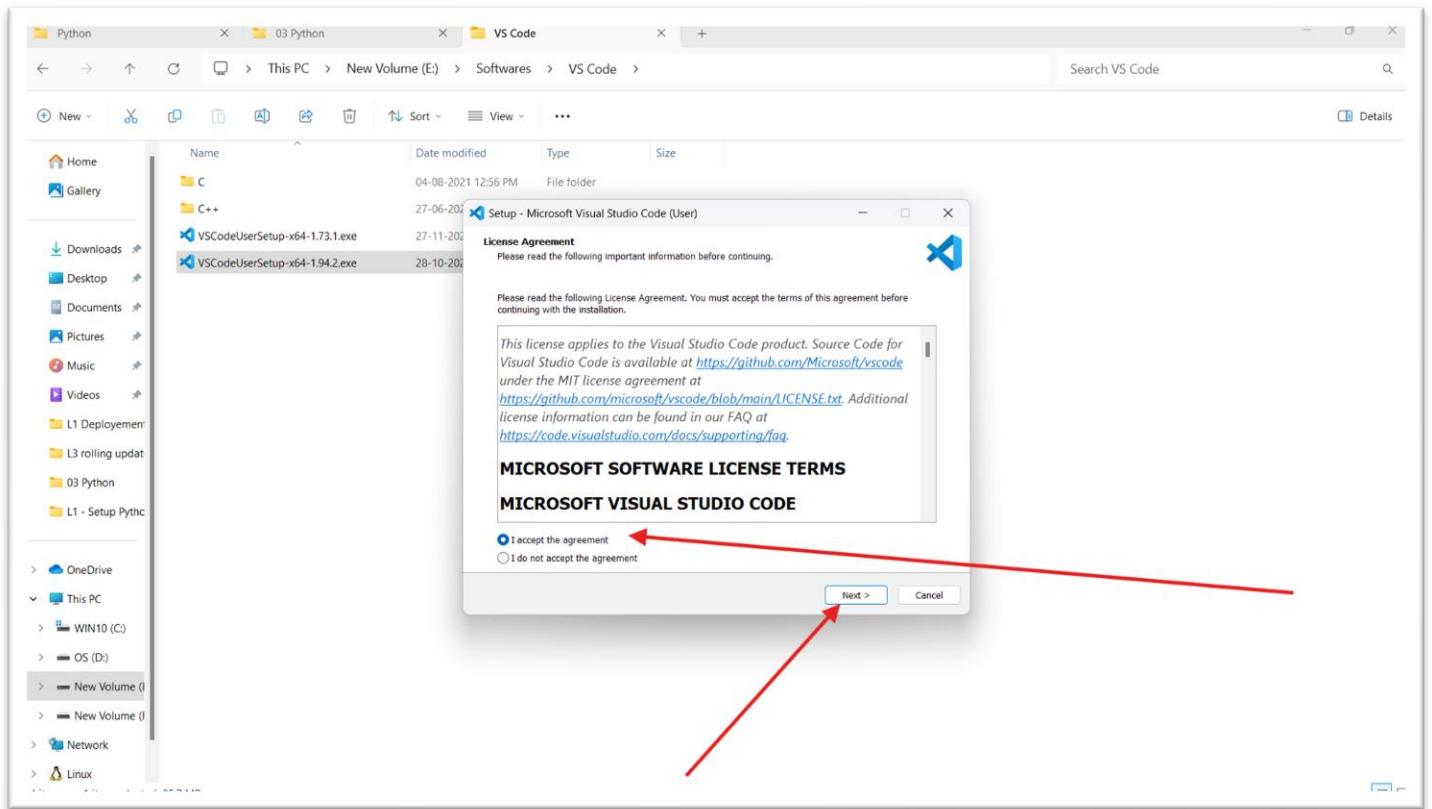
2. L2 - Install Visual Studio Code and Install Python and Terraform Extensions in VS Code

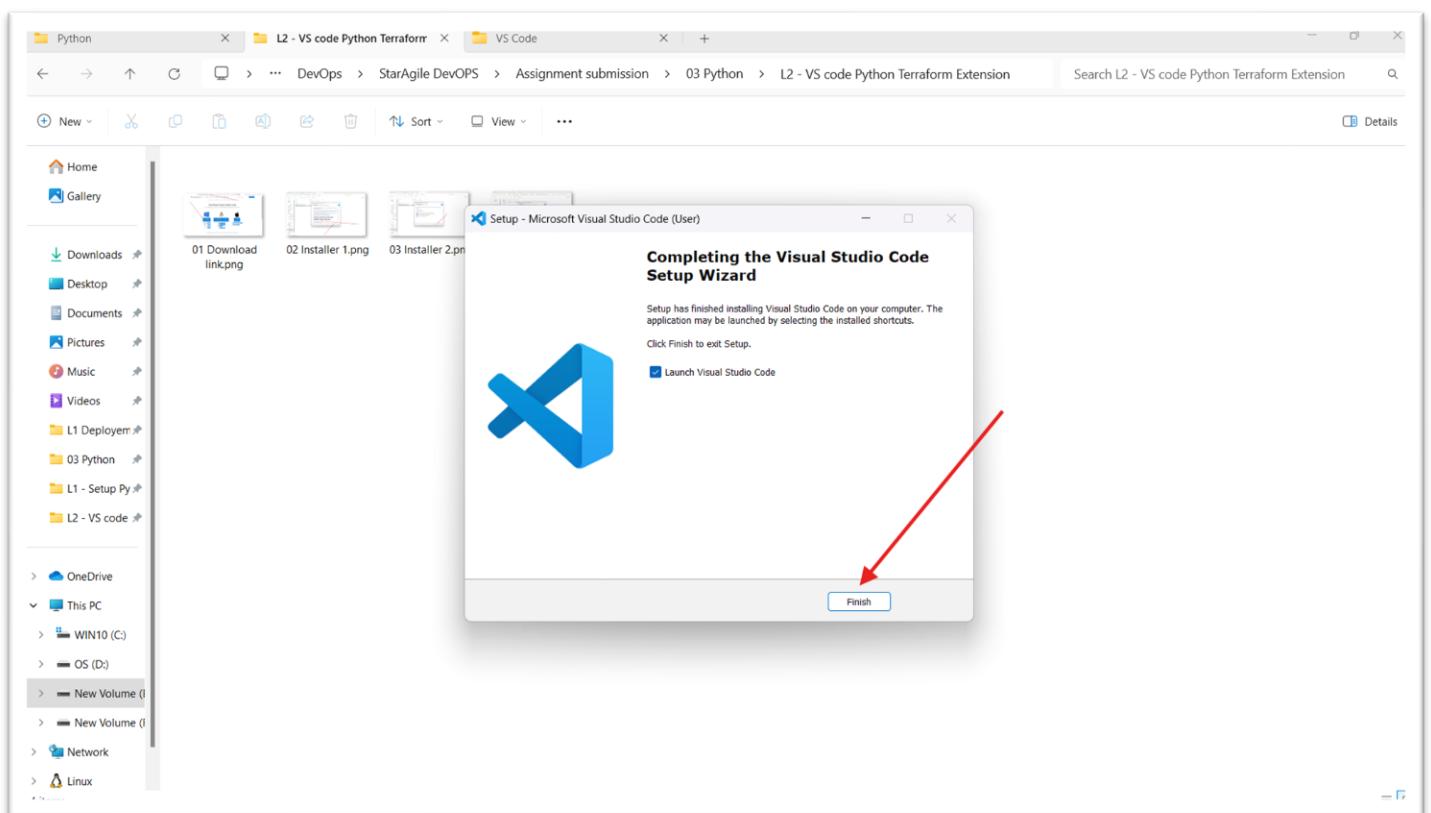
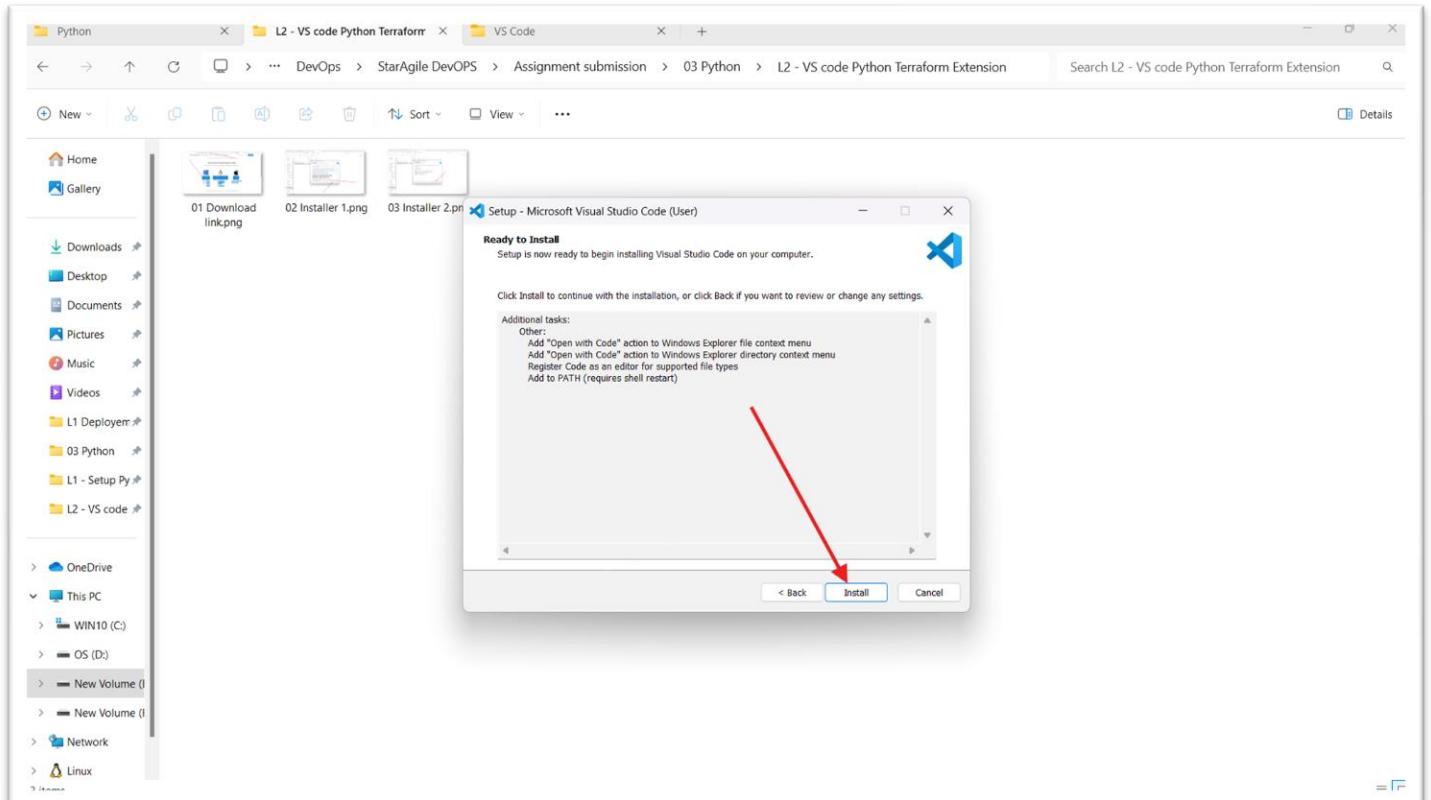
Ans.

Step 1: Download Visual Studio Code from [Visual studio code download link](https://code.visualstudio.com/download).

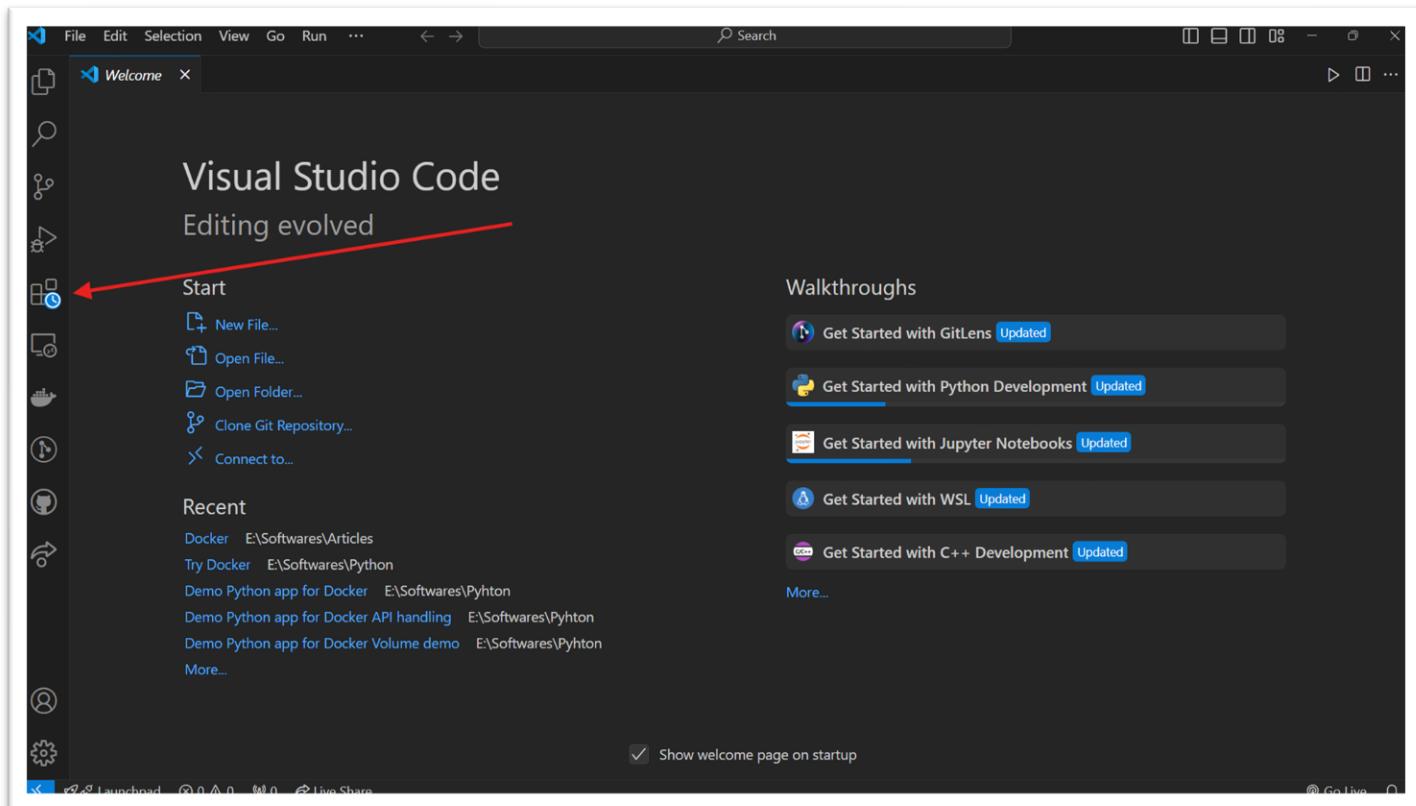


Step 2: Run the installer and complete the installation with default settings.

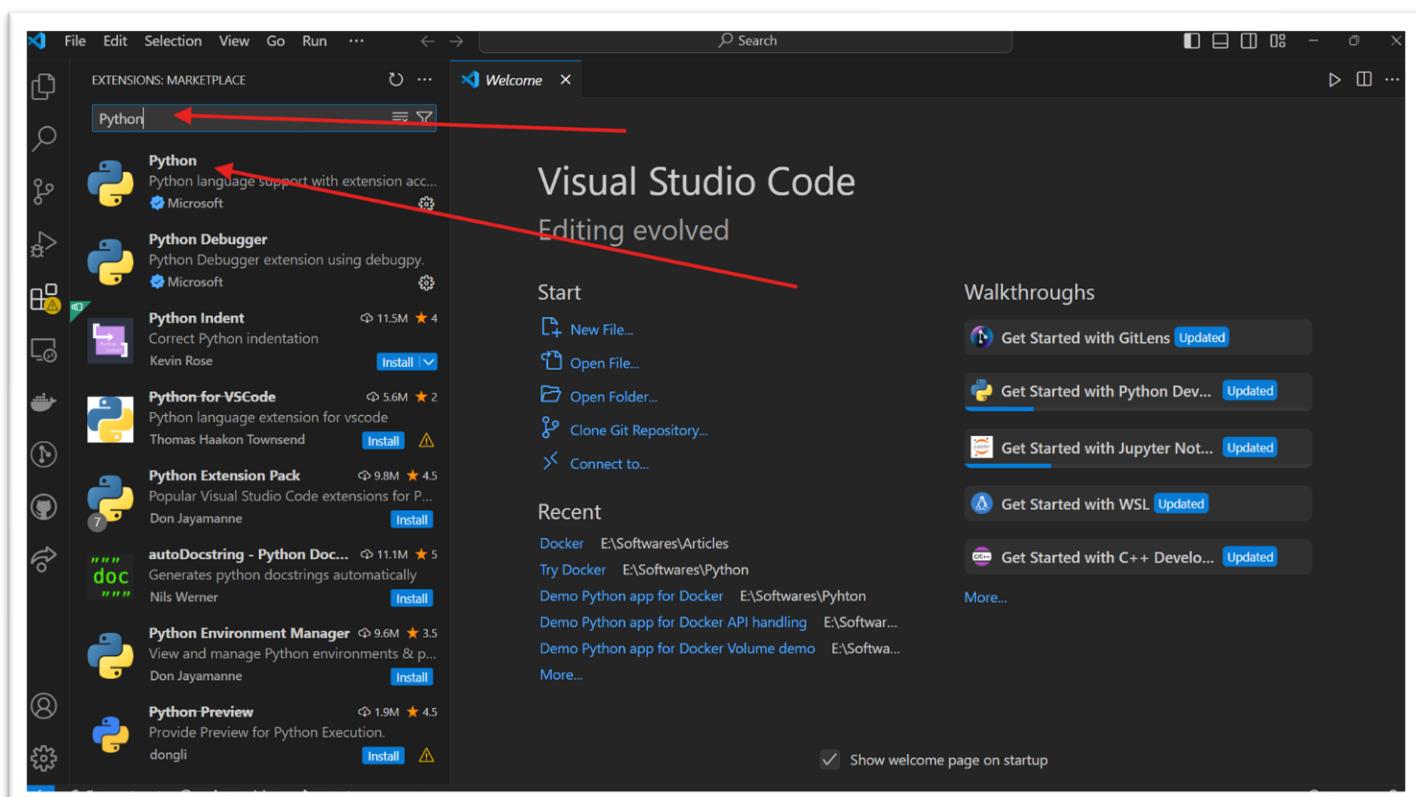


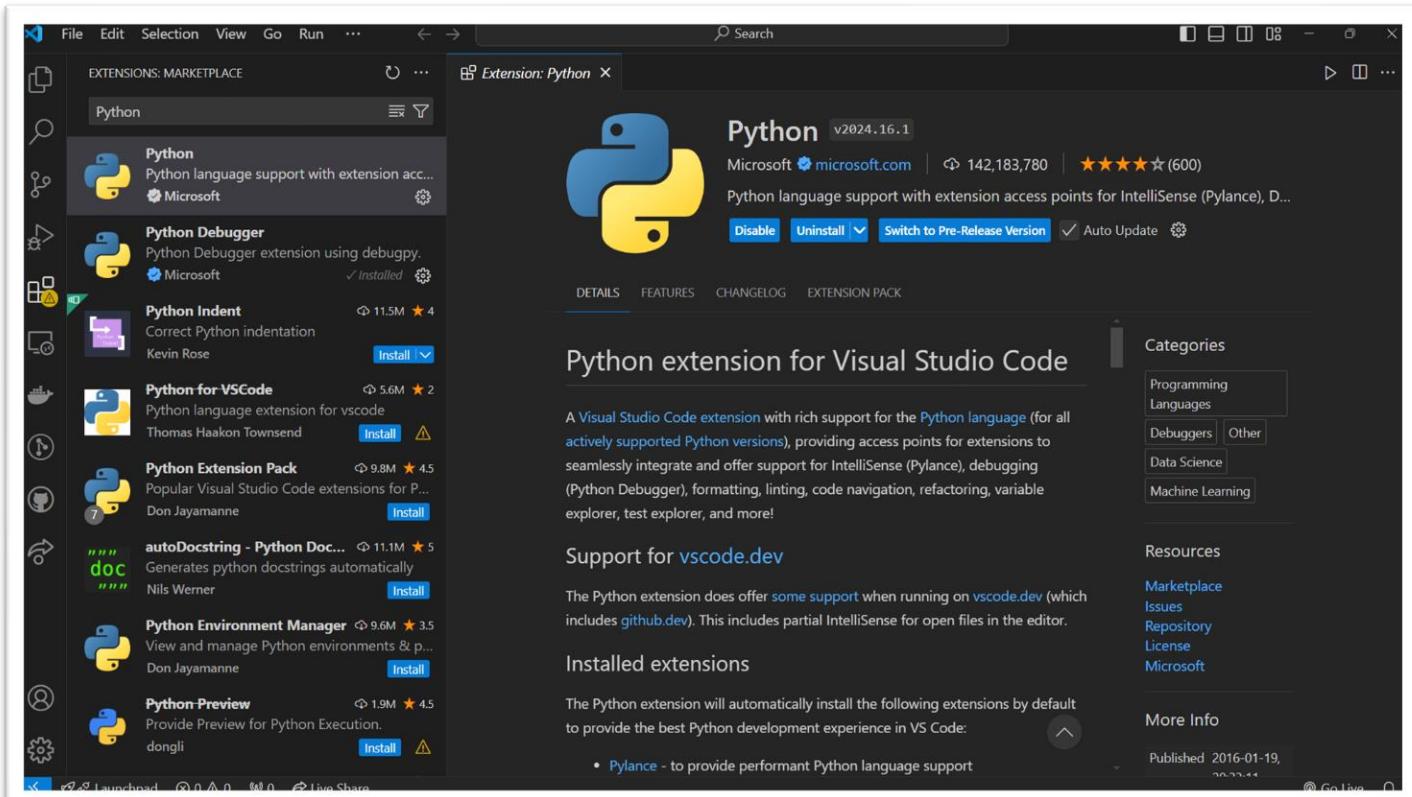
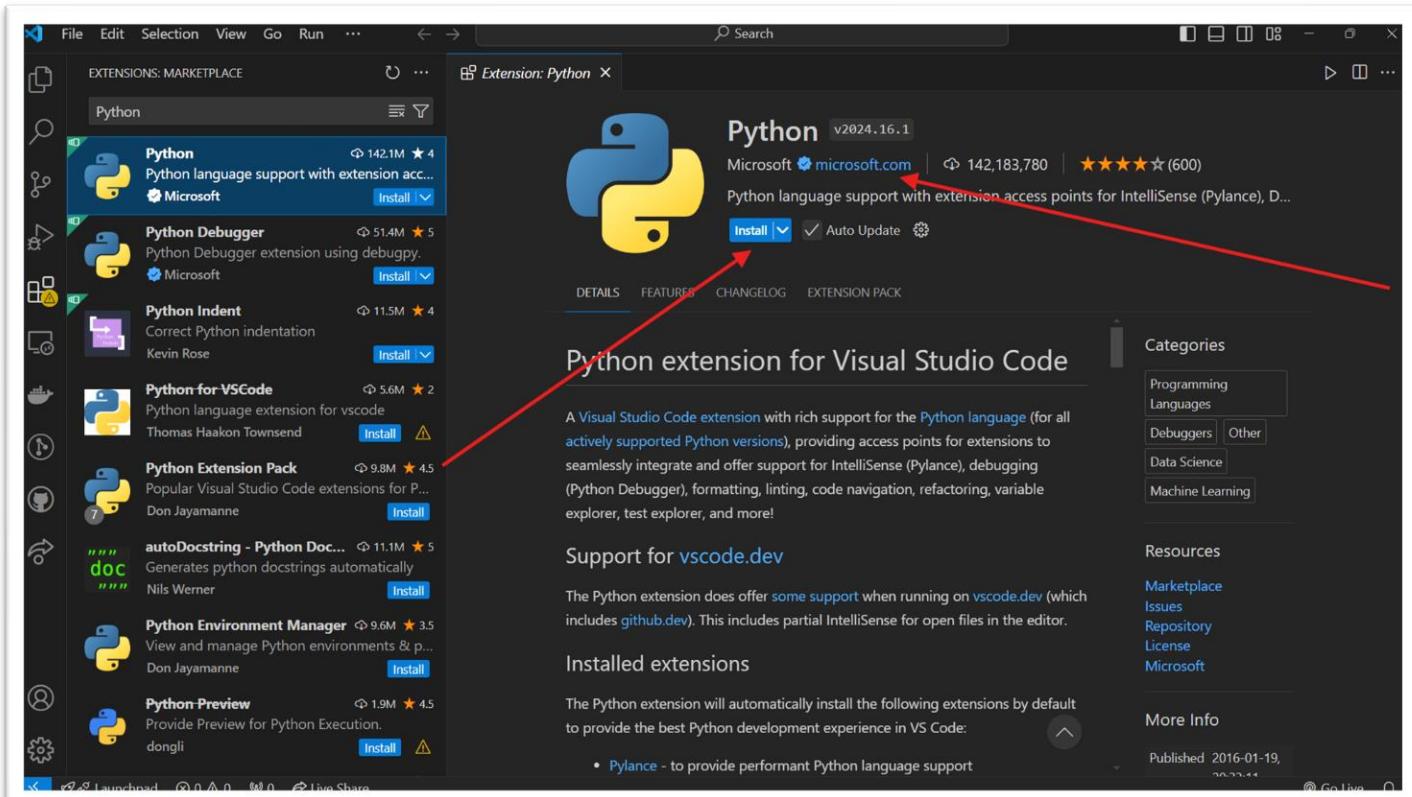


Step 3: Open VS Code and go to the Extensions tab (**Ctrl+Shift+X**).

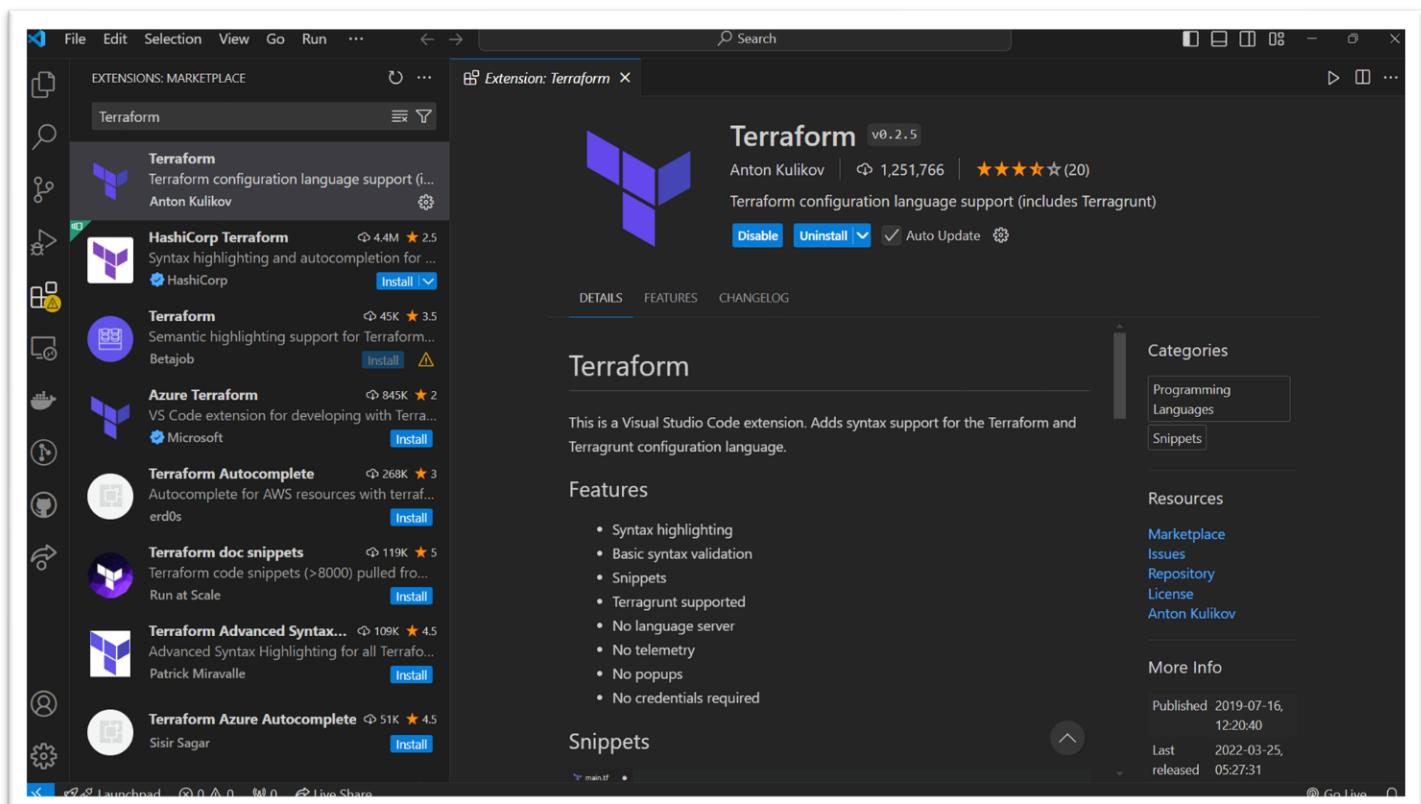
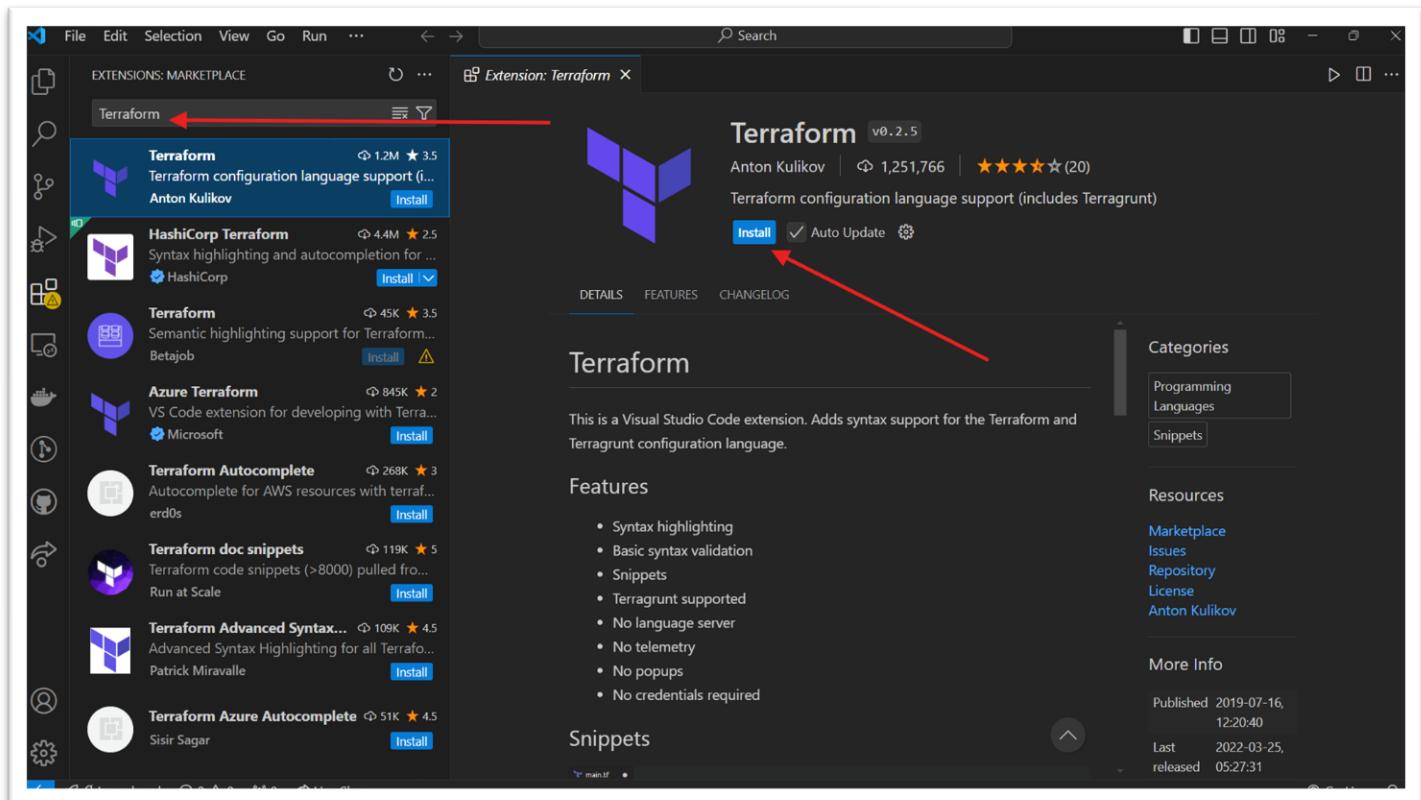


Step 4: Search for **Python** extension and click **Install**.





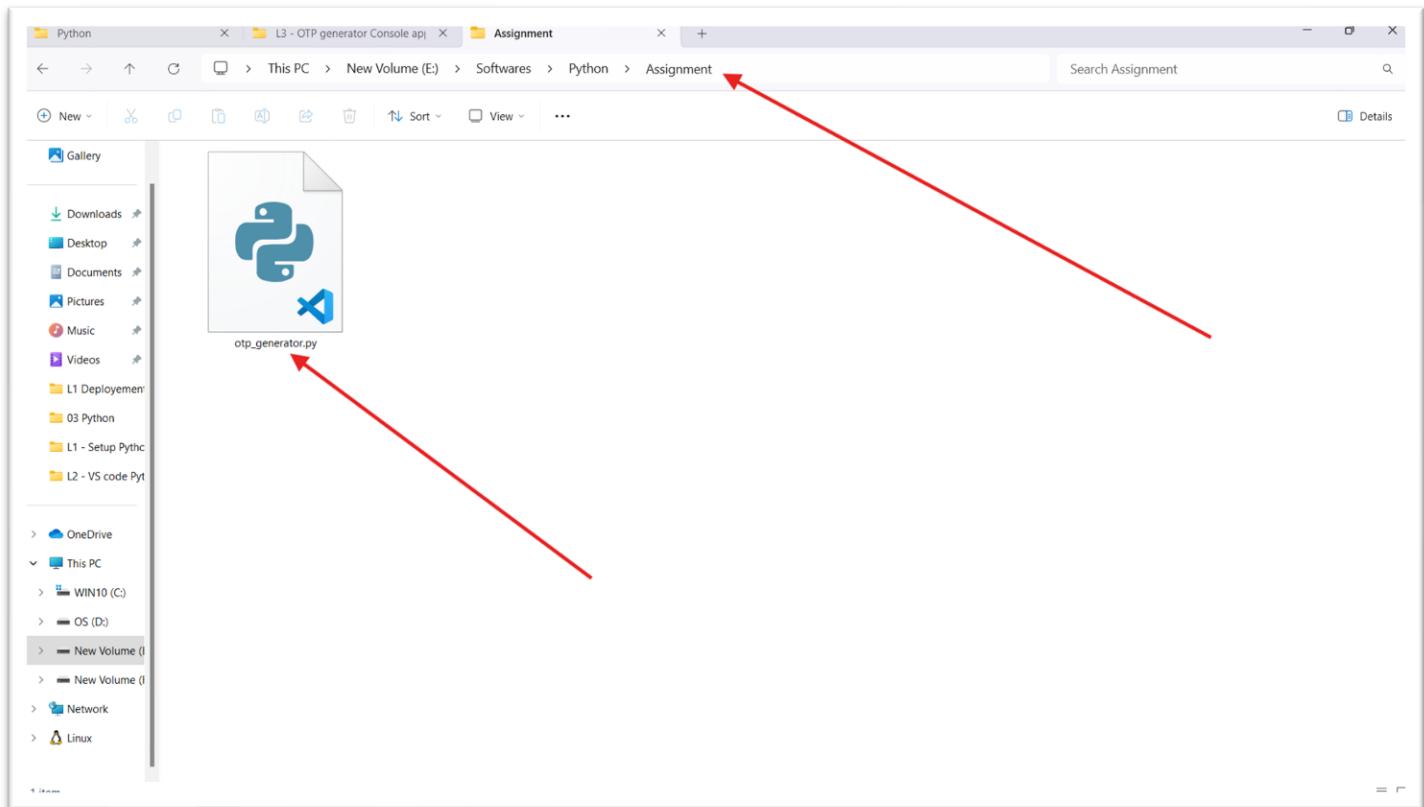
Step 5: Search for Terraform extension and click Install.



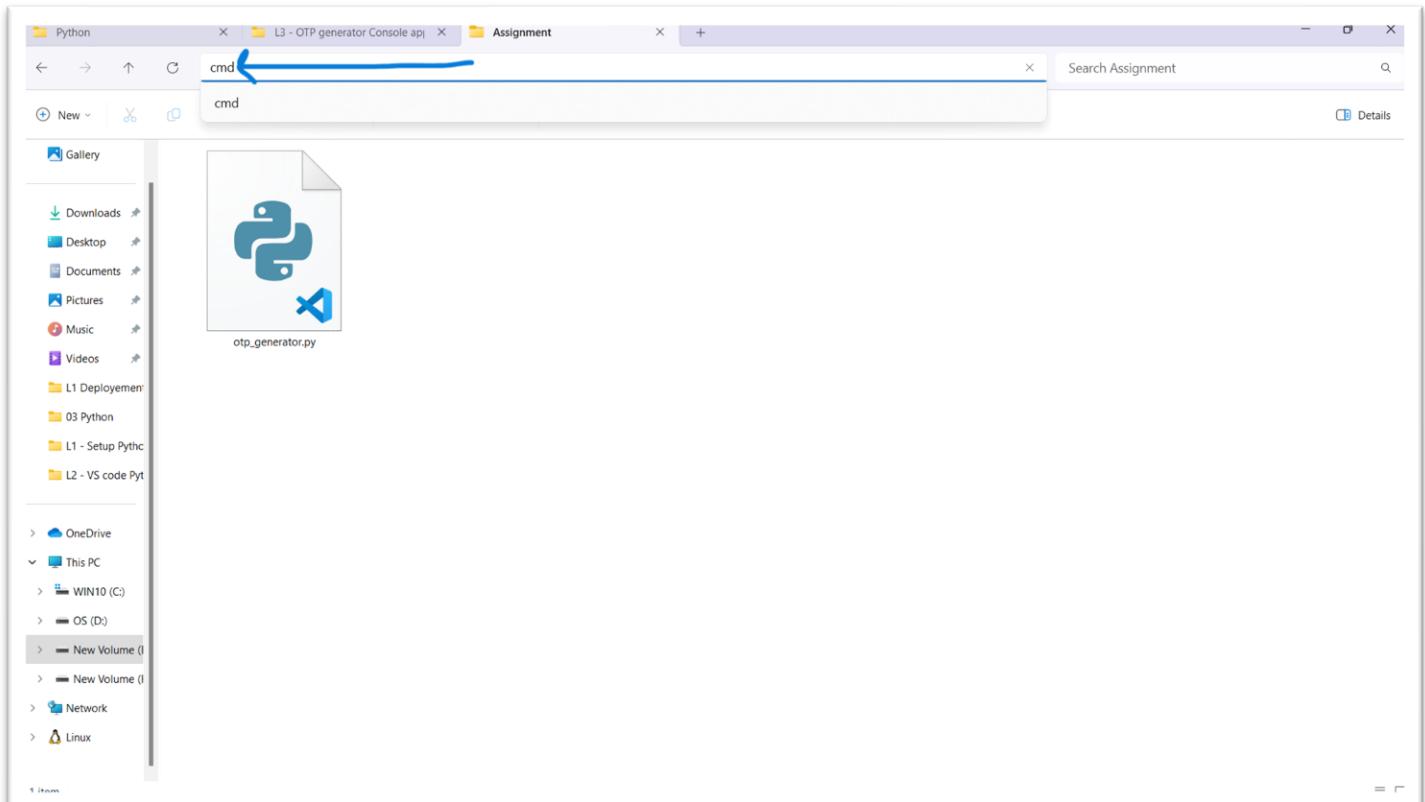
3. L3 - Create Python Console Application to randomly generate OTP kind of secure code

Ans.

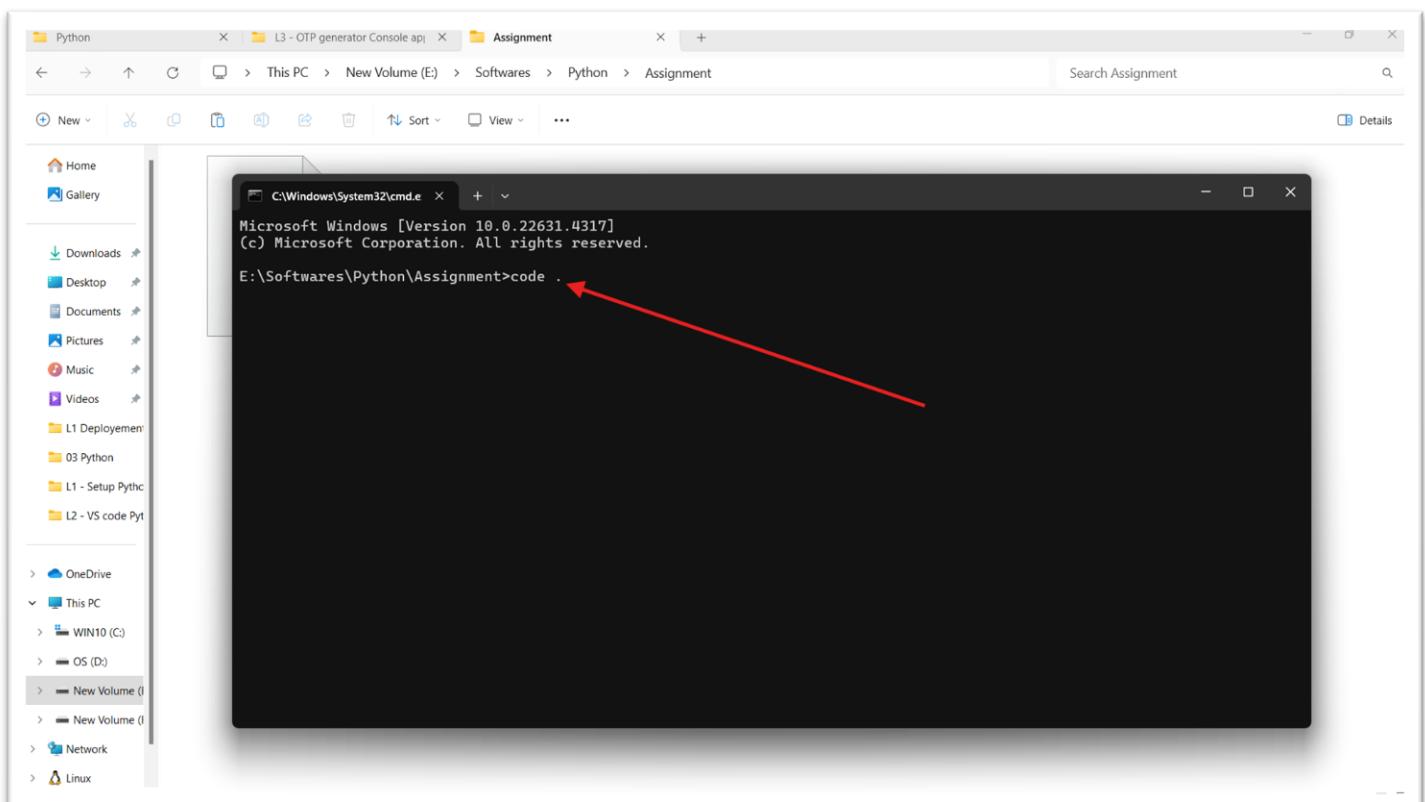
Step 1: Create a **Folder** and python file with it .py extension for our code

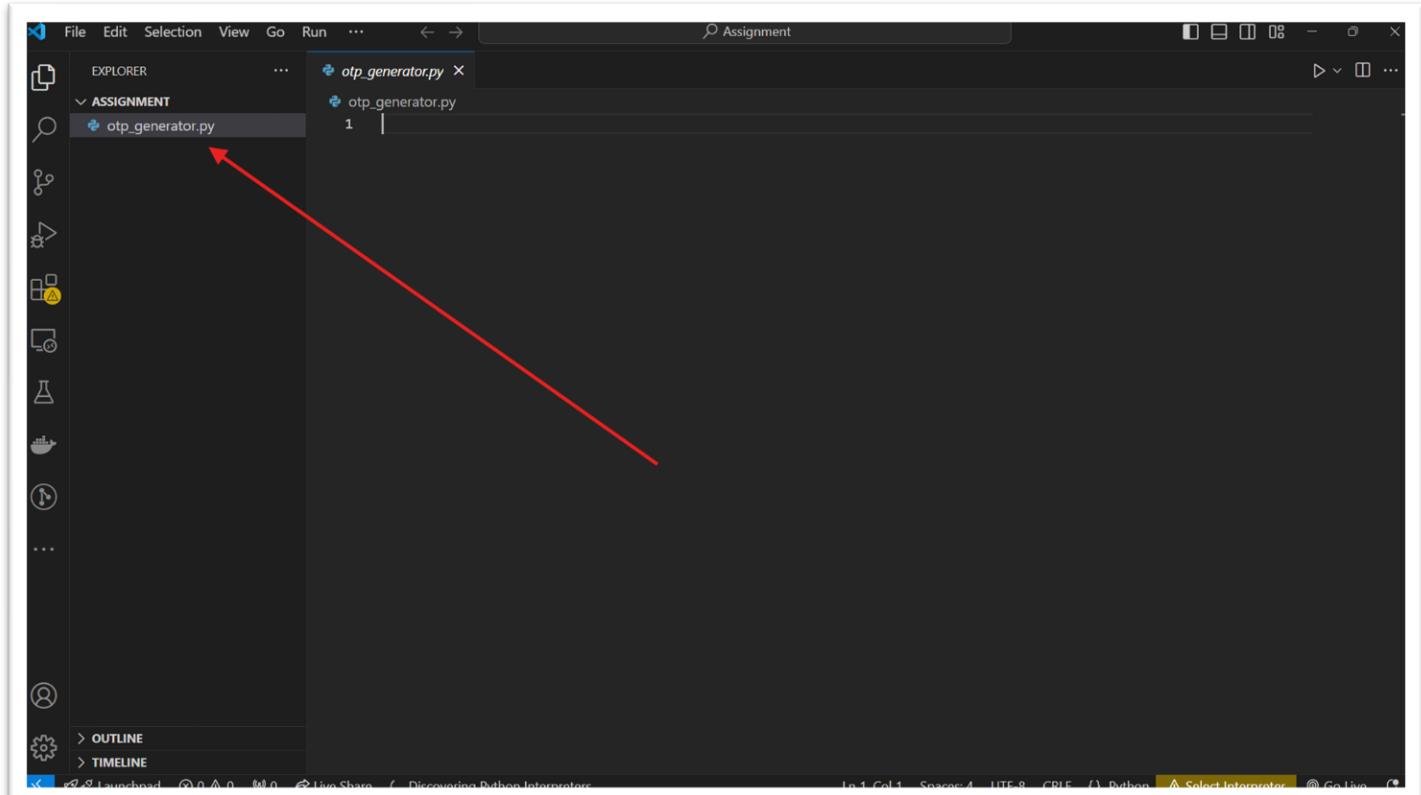


Step 2: Click on address bar of Windows file explorer and enter ‘cmd’ to open command prompt in current location



Step 3: Enter ‘**code .**’ in the command prompt to open VS code in current directory





Step 4: Enter the code in the attached screen shot to make a otp generator python console app. Do ‘**Ctrl + S**’ to save the file once the code is done

A screenshot of the Visual Studio Code interface. The 'EXPLORER' sidebar shows the 'ASSIGNMENT' folder with 'otp_generator.py' selected. The main editor area contains the following Python code:

```
1 import random, string
2 otp = ''.join(random.choices(string.ascii_letters + string.digits, k=6))
3 print("Your OTP is:", otp)
```

The lines of code from 'import' to 'print' are highlighted with a red rectangular box. The status bar at the bottom shows 'In 3, Col 27'. Other status icons are present at the bottom.

Step 5: Do ‘Ctrl + ~’ to open VS code terminal to run the program enter command `python <otp_generator_py_app_name.py>` or Click the Run button of ‘Code Runner extension’ to directly run the program

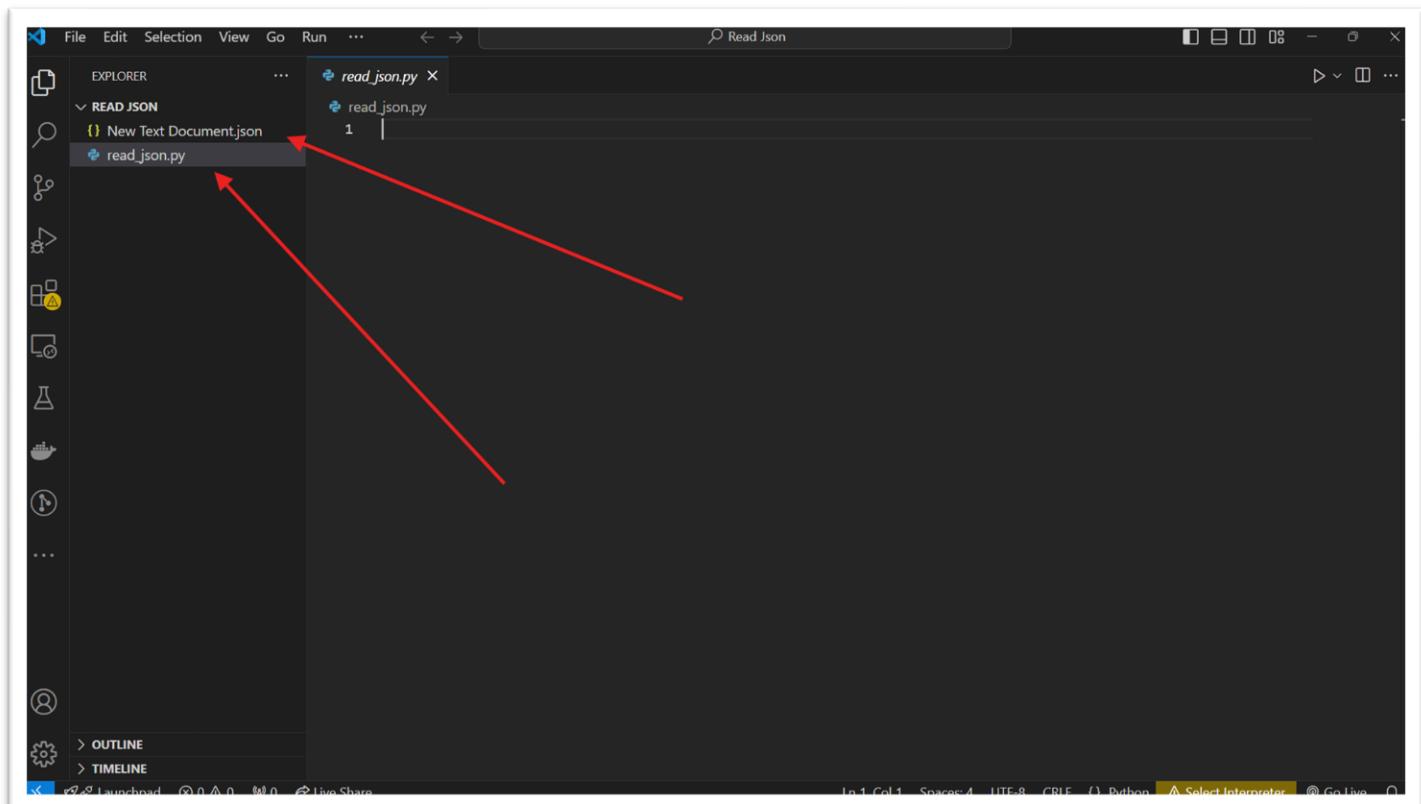
The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a file named "otp_generator.py" under the "Assignment" folder.
- Terminal View:** Displays the command "python .\otp_generator.py" being run three times, each producing a different six-digit OTP.
 - First run: PS E:\Softwares\Python\Assignment> python .\otp_generator.py
Your OTP is: 74SAAD
 - Second run: PS E:\Softwares\Python\Assignment> python .\otp_generator.py
Your OTP is: Jogiou
 - Third run: PS E:\Softwares\Python\Assignment> python .\otp_generator.py
Your OTP is: c07w92
- Status Bar:** Shows the current file is "Assignment", line 3, column 27, spaces 4, and the selected interpreter is "Python".

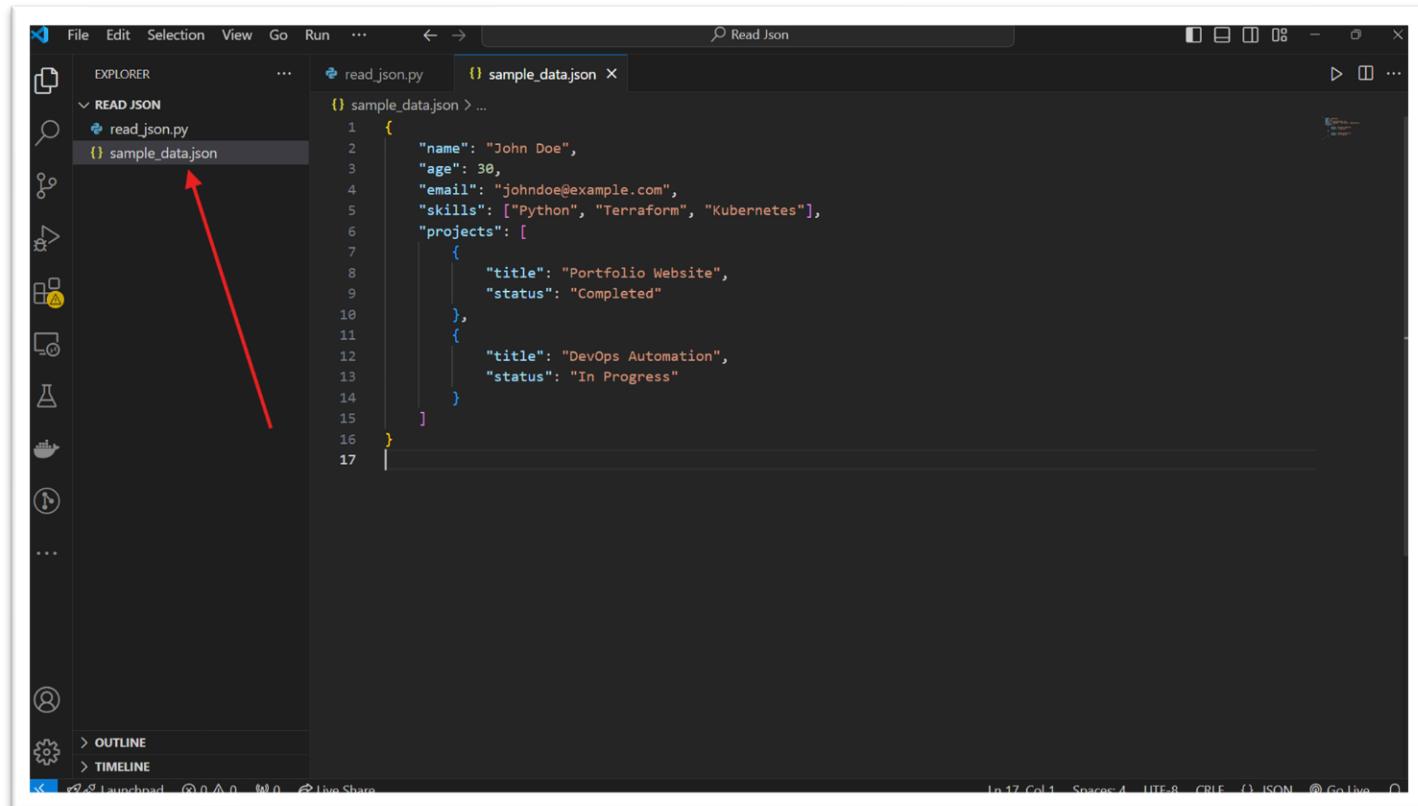
4. L4 - Create Python Console Application to read the contents of .json file and print in the VS Code python console output

Ans.

Step 1: Create a new Folder and python file with .py extension and open it in VS code



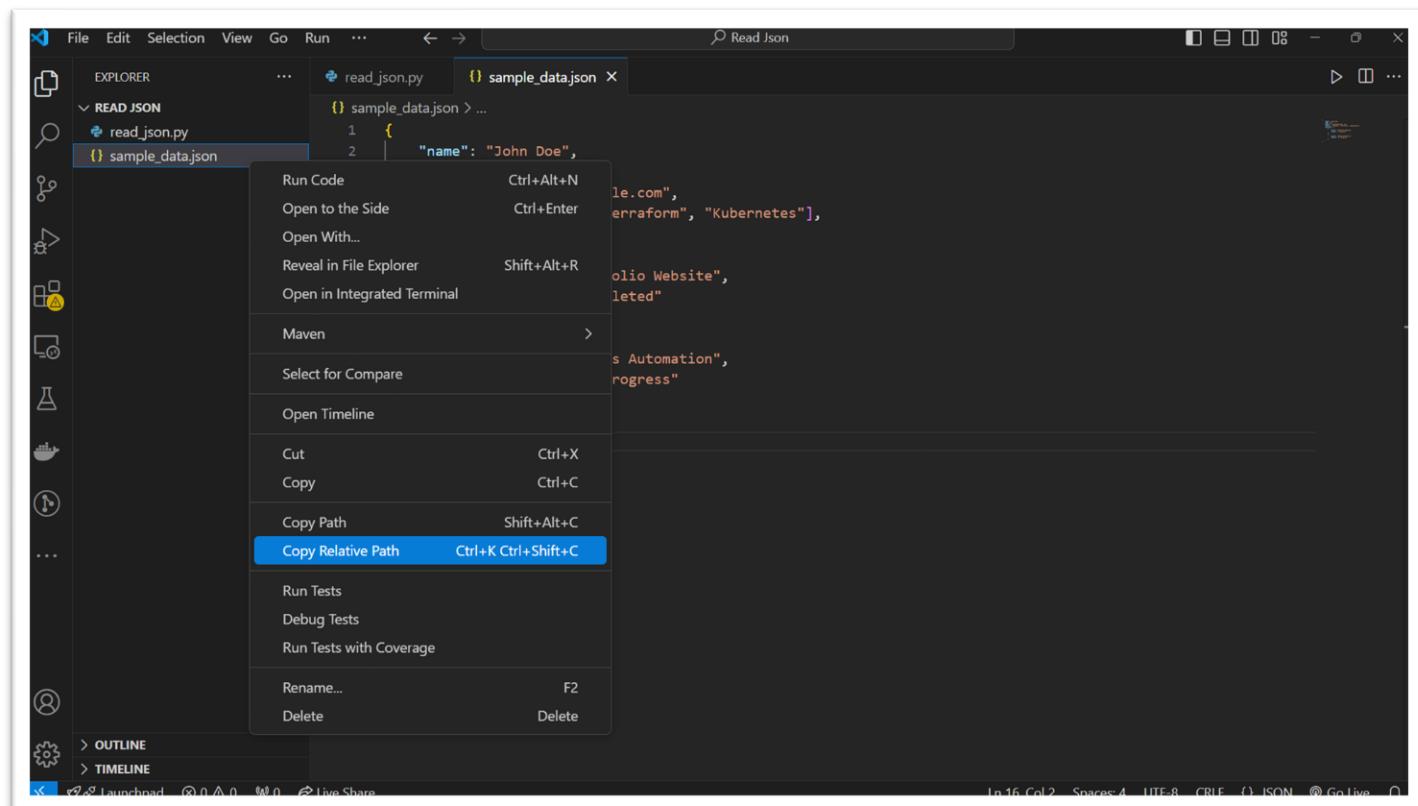
Step 2: Copy paste the code in the attached screenshot to run the app



A screenshot of the Visual Studio Code interface. The left sidebar shows the 'EXPLORER' view with a tree structure. Under 'READ JSON', there are two items: 'read_json.py' and 'sample_data.json'. A red arrow points from the text above to the 'sample_data.json' item. The main editor area displays the contents of 'sample_data.json' as a JSON object:

```
1 {  
2     "name": "John Doe",  
3     "age": 30,  
4     "email": "johndoe@example.com",  
5     "skills": ["Python", "Terraform", "Kubernetes"],  
6     "projects": [  
7         {  
8             "title": "Portfolio Website",  
9             "status": "Completed"  
10        },  
11        {  
12            "title": "DevOps Automation",  
13            "status": "In Progress"  
14        }  
15    ]  
16}  
17
```

The status bar at the bottom indicates 'Line 17 Col 1 Spaces: 4 LITE-8 CRLE {} JSON @ Go Live'.



```
File Edit Selection View Go Run ... ← → ⌂ Read Json
```

EXPLORER
READ JSON
read_json.py sample_data.json

```
❶ import json
❷ with open('sample_data.json', 'r') as file:
❸     data = json.load(file)
❹     print(json.dumps(data, indent=4))
```

Step 3: Run the app by using VS code terminal(**Ctrl + ~**) and entering command '**python <app_name.py>**' or just click the Run button of the '**Code Runner**' extension

```
File Edit Selection View Go Run ... ← → ⌂ Read Json
```

EXPLORER
READ JSON
read_json.py sample_data.json

```
❶ import json
❷ with open('sample_data.json', 'r') as file:
❸     data = json.load(file)
❹     print(json.dumps(data, indent=4))
```

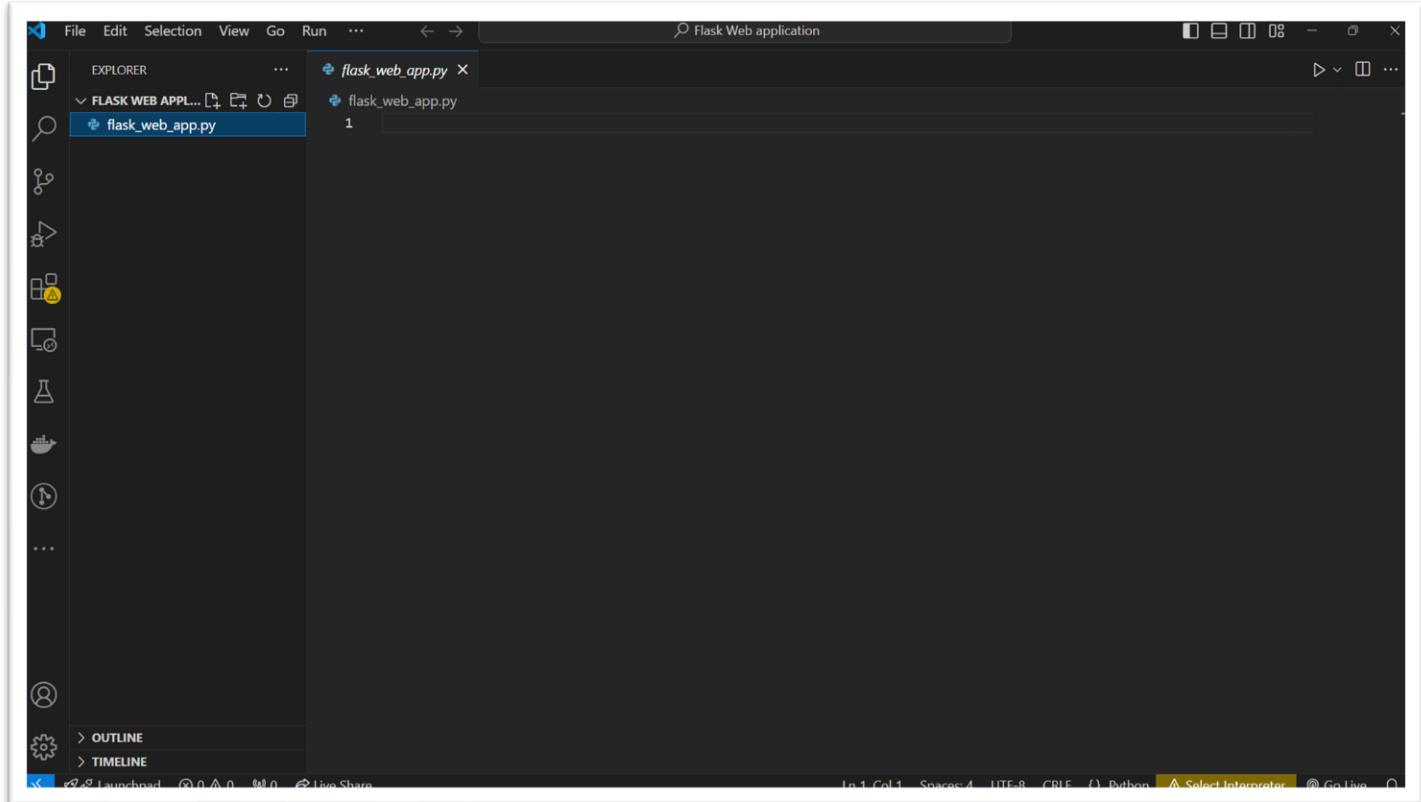
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
PS E:\Softwares\Python\Assignment\Read Json> python .\read_json.py
{
    "name": "John Doe",
    "age": 30,
    "email": "johndoe@example.com",
    "skills": [
        "Python",
        "Terraform",
        "Kubernetes"
    ],
    "projects": [
        {
            "title": "Portfolio Website",
            "status": "Completed"
        },
        {
            "title": "DevOps Automation",
            "status": "In Progress"
        }
    ]
}
```

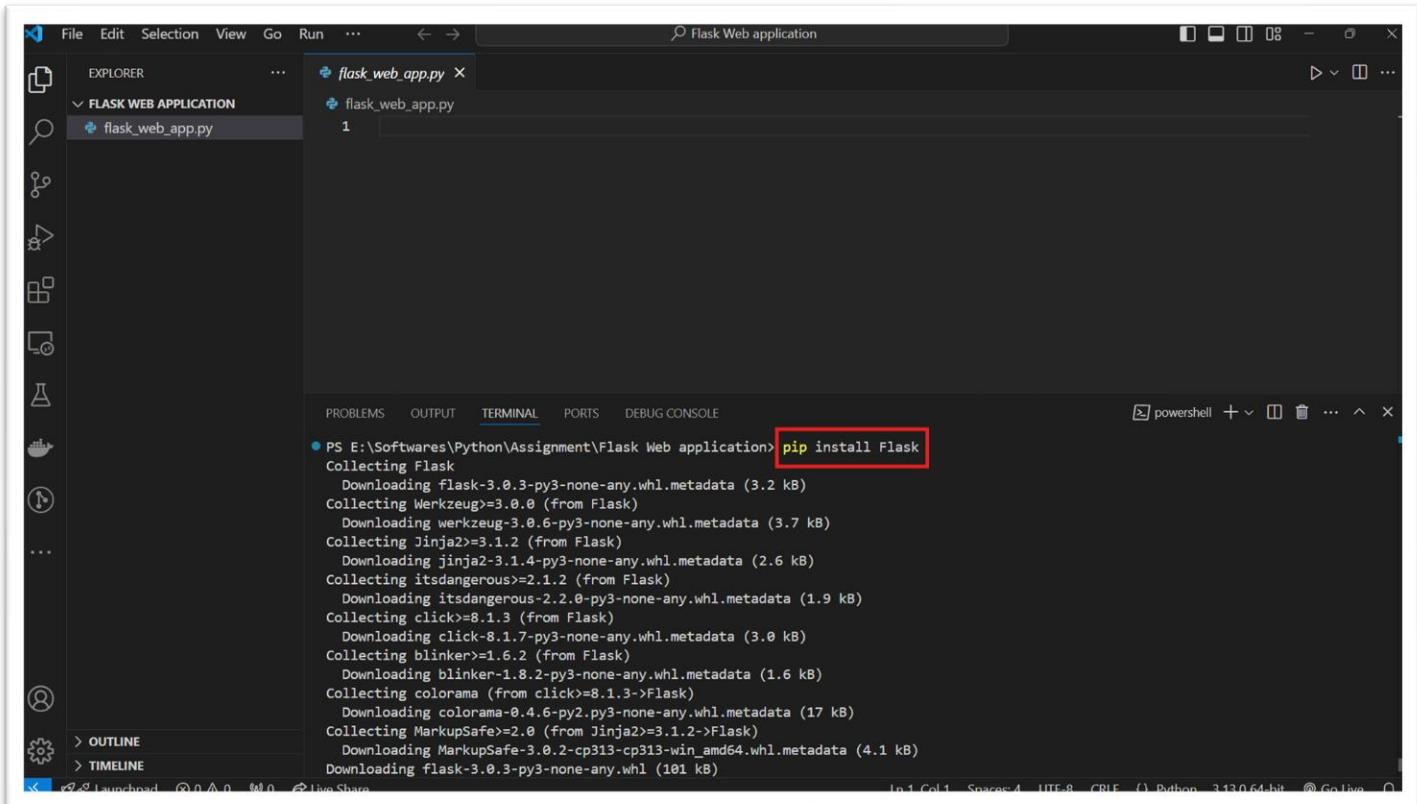
5. L5 - Create Python Web Application to using Flask Web Application Framework

Ans.

Step 1: Open your project folder in Visual Studio Code.



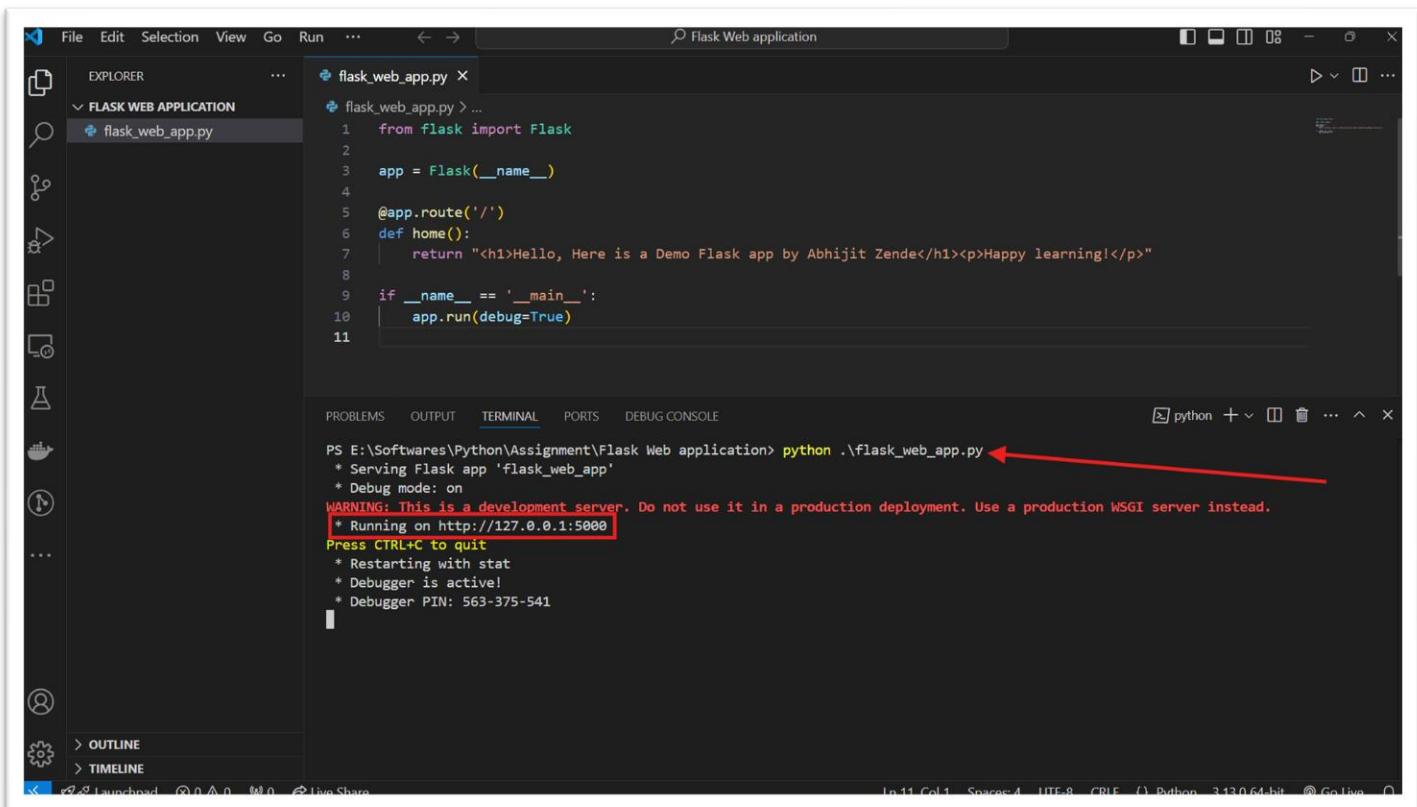
Step 2: Install Flask by running '`pip install Flask`' in the terminal.



A screenshot of the Visual Studio Code interface. The left sidebar shows the 'EXPLORER' view with a folder named 'FLASK WEB APPLICATION' containing a file 'flask_web_app.py'. The main area is the 'TERMINAL' tab, which displays the command 'PS E:\Softwares\Python\Assignment\Flask Web application> pip install Flask' followed by the output of the pip installation process. The output shows Flask and its dependencies being downloaded from PyPI. The terminal window has a red box highlighting the pip command.

```
PS E:\Softwares\Python\Assignment\Flask Web application> pip install Flask
Collecting Flask
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=3.0.0 (from Flask)
  Downloading werkzeug-3.0.6-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from Flask)
  Downloading jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from Flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from Flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from Flask)
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting colorama (from click>=8.1.3->Flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->Flask)
  Downloading MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl.metadata (4.1 kB)
  Downloading flask-3.0.3-py3-none-any.whl (101 kB)
```

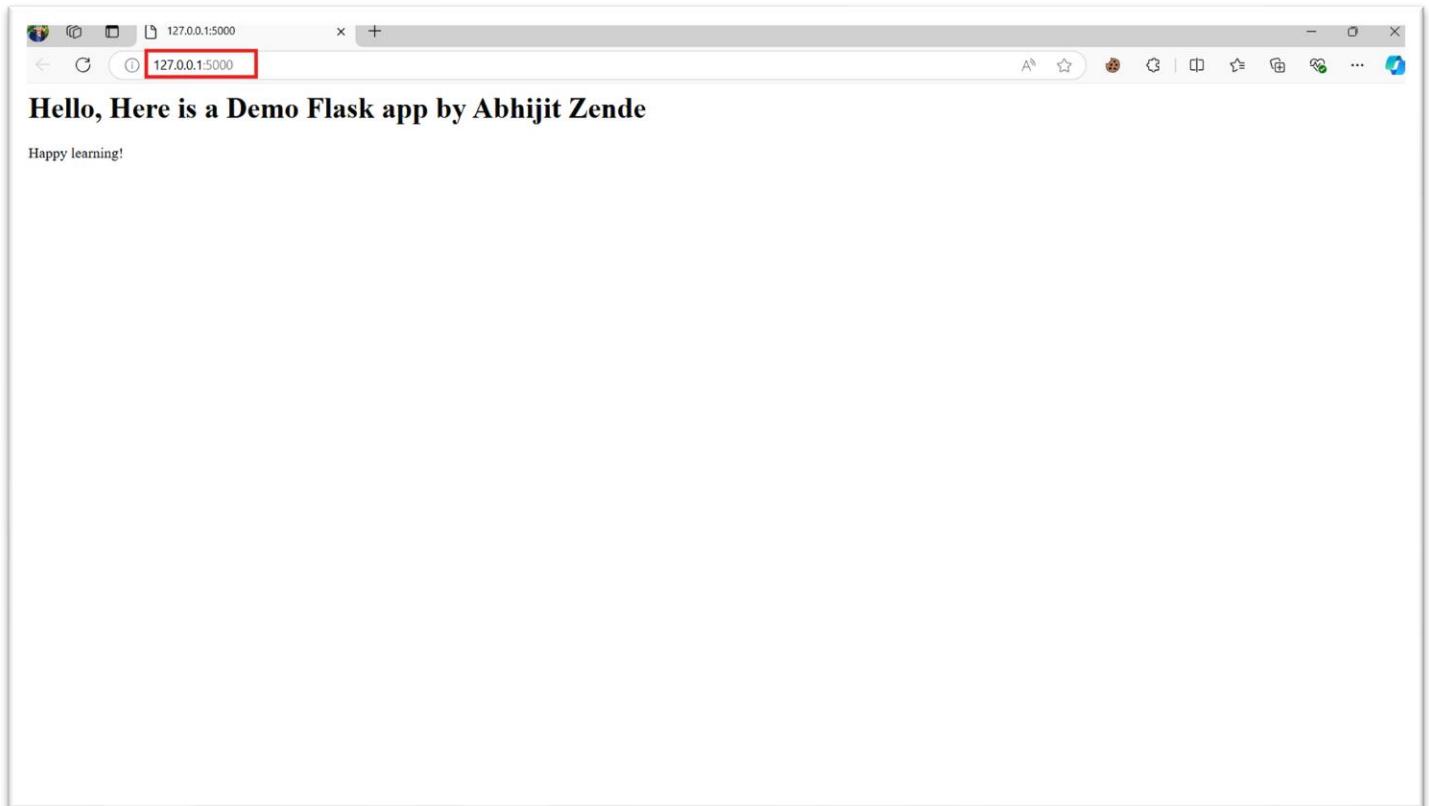
Step 3: Write the code from below screenshot, save, and run the python program



A screenshot of the Visual Studio Code interface. The left sidebar shows the 'EXPLORER' view with a folder named 'FLASK WEB APPLICATION' containing a file 'flask_web_app.py'. The main area is the 'TERMINAL' tab, which displays the command 'PS E:\Softwares\Python\Assignment\Flask Web application> python .\flask_web_app.py' followed by the output of the Python application. The output shows the Flask app running on port 5000. A red arrow points to the command in the terminal. The terminal window has a red box highlighting the command.

```
PS E:\Softwares\Python\Assignment\Flask Web application> python .\flask_web_app.py
 * Serving Flask app 'flask_web_app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 563-375-541
```

Step 4: Open a browser and go to ‘<http://127.0.0.1:5000>’ to see the web app running.



Step 5: Stop the server by pressing **Ctrl+C** in the terminal when done.

