**Project Manual**

1. **How to download Python:**

* Download Python: Go to the official Python website at https://www.python.org/ and navigate to the Downloads section. Choose the version of Python you want to install.
* Select Operating System: Select the appropriate installer for your operating system. Python is available for Windows, macOS, and Linux.
* Download Installer: Click on the download link for the installer. The website should automatically detect your operating system and suggest the correct version.
* Run Installer: Once the installer is downloaded, run it. Follow the installation instructions provided by the installer. Make sure to check the box that says "Add Python to PATH" during the installation process, as it makes it easier to run Python from the command line.
* Verify Installation: After installation, open a command prompt (on Windows) or terminal (on macOS or Linux) and type `python --version` or `python3 --version` to verify that Python has been installed correctly. You should see the version number of Python displayed.

1. **Set the Environment Variable:**

* Find Python Installation Directory: First, locate the directory where Python is installed on your system. By default, on most systems, it is installed in a directory like `C:\Python39` on Windows or `/usr/local/bin/python3` on Unix-based systems. Note down this directory.
* Set Environment Variable on Windows:
  + Right-click on "This PC" (or "My Computer") and select "Properties."
  + In the System Properties window, click on "Advanced system settings."
  + In the System Properties window, click on the "Environment Variables..." button.
  + In the Environment Variables window, under the "System variables" section, find the "Path" variable and select it.
  + Click the "Edit..." button.
  + In the Edit Environment Variable window, click the "New" button and add the path to the Python installation directory you noted down earlier.
  + Click "OK" on all windows to save the changes.
* Set Environment Variable on macOS/Linux:
  + Open a terminal window.
  + Edit the shell profile file, such as `.bash\_profile`, `.bashrc`, `.zshrc`, or `.profile`, depending on your shell.
  + Add the following line at the end of the file, replacing `/path/to/python/bin` with the actual path to your Python installation directory:
* export PATH="/path/to/python/bin:$PATH"
  + Save the file and exit the text editor.
  + To apply the changes immediately, run `source ~/.bash\_profile` (or the corresponding file for your shell) in the terminal.

After setting the environment variable, open a new terminal or command prompt window and type `python --version` (or `python3 --version` on Unix-based systems) to verify that Python is accessible from any directory.

**3.Install Code Editor**

Download: Go to the Visual Studio Code website: [https://code.visualstudio.com/](<https://code.visualstudio.com/>)

* Select Your Platform: VS Code is available for Windows, macOS, and Linux. Click on the download button corresponding to your operating system.
* Install: Once the download is complete, run the installer file. Follow the installation instructions provided by the installer. On Windows and macOS, this usually involves double-clicking the downloaded file and following the prompts in the installation wizard.
* Open Visual Studio Code: After installation, you can find Visual Studio Code in your applications menu or launch it directly from the installation directory.
* Extensions: Visual Studio Code has a rich ecosystem of extensions that enhance its functionality. You can browse and install extensions from the Extensions view (accessible via the Extensions icon in the sidebar) within VS Code. Some popular extensions include Python, for Python language support, and GitLens, for Git integration.
* Update Settings (Optional): You can customize Visual Studio Code to your liking by modifying settings. To access settings, click on the gear icon in the bottom left corner of the window and select "Settings". You can also open the settings by pressing `Ctrl+,` (Cmd+, on macOS) or by selecting "Preferences > Settings" from the menu bar.

1. **Install Necessary Python Libraries**

To install Python libraries using either pip or conda, you first need to have Python and either pip or conda installed on your system. Assuming you have Python installed, here's how you can install libraries using both tools:

* Using pip:
* Open Command Prompt / Terminal: Open a command prompt (Windows) or terminal (macOS/Linux).
* Install a Library: Use the following command to install a library. Replace `library\_name` with the name of the library you want to install.
* pip install library\_name

For example, to install NumPy:

* pip install numpy
* Install Specific Version: If you want to install a specific version of a library, you can specify it using the `==` operator. For example:
* pip install numpy==1.21.0
* Install from Requirements File: If you have a requirements.txt file containing a list of libraries to install, you can use the following command:
* pip install -r requirements.txt
* Using conda:
* Open Command Prompt / Terminal: Open a command prompt (Windows) or terminal (macOS/Linux).
* Install a Library: Use the following command to install a library. Replace `library\_name` with the name of the library you want to install.
* conda install library\_name

For example, to install NumPy:

* conda install numpy
* Install Specific Version: You can specify a specific version of a library using the following command:
* conda install numpy=1.21.0
* Install from Environment File: If you have an environment.yml file containing a list of libraries to install, you can use the following command:
* conda env create -f environment.yml

Remember to replace `library\_name` with the actual name of the library you want to install, and adjust version numbers or file names as needed. Additionally, using conda is recommended when you're managing environments and dealing with non-Python dependencies.

1. **Run the Project:**

To run Python code in Visual Studio Code:

* Open Visual Studio Code: Launch Visual Studio Code by clicking on its icon or searching for it in your applications menu.
* Open/Create a Python File: Open an existing Python file or create a new one. To create a new file, click on "File" in the menu bar, then select "New File" (or press `Ctrl+N` or `Cmd+N`).
* Write Your Python Code: Write or paste your Python code into the editor.
* Save Your Python File: Save your Python file with a `.py` extension. You can do this by clicking on "File" in the menu bar, then selecting "Save As..." (or pressing `Ctrl+S` or `Cmd+S`). Choose a location and name for your file, and make sure it ends with `.py`.
* Run Your Python Code:

- To run your Python code, you can use the built-in terminal within Visual Studio Code.

- Open the terminal by clicking on "Terminal" in the menu bar, then selecting "New Terminal" (or pressing `Ctrl+`` or `Cmd+``).

- Once the terminal is open, navigate to the directory where your Python file is located using the `cd` command.

- Then, type `python filename.py` (replace `filename.py` with the name of your Python file) and press Enter to execute your Python code.

Alternatively, you can use the built-in "Run Python File in Terminal" feature in Visual Studio Code:

- Right-click anywhere within the Python file editor.

- Select "Run Python File in Terminal" from the context menu.

- Your Python code will be executed in the terminal.

**6.Analyze the Results**

To analyze the results of your Python code execution, you would typically follow these steps:

* Review Output: Examine the output generated by your Python code in the terminal or the integrated terminal within Visual Studio Code. Look for any errors, warnings, or messages that indicate the status of your program's execution.
* Check for Expected Behavior: Compare the output of your Python code against your expectations. Ensure that the program is behaving as intended and producing the desired results. If the output is not as expected, review your code for logical errors, syntax mistakes, or incorrect assumptions.
* Debugging: If your Python code encountered errors or produced unexpected results, use debugging techniques to identify and resolve issues. Visual Studio Code provides built-in debugging tools that allow you to set breakpoints, inspect variables, and step through your code to understand its behavior better.
* Data Analysis (If Applicable): If your Python code involves data analysis or processing, you may need to further analyze the output or results generated by your program. This could involve statistical analysis, visualization of data using libraries like Matplotlib or Seaborn, or performing calculations to derive insights from the data.
* Interpret Results: Once you've reviewed the output and performed any necessary analysis, interpret the results of your Python code execution. Determine whether the program achieved its objectives, whether any modifications are required, and what insights or conclusions can be drawn from the results.
* Iterate and Refine (If Needed): Based on your analysis of the results, iterate on your Python code if necessary. Make adjustments, improvements, or optimizations to the code to address any issues or shortcomings identified during the analysis process. Repeat the execution and analysis steps as needed until you are satisfied with the results.

By following these steps, you can effectively analyze the results of your Python code execution and make informed decisions based on the output generated by your programs.