

## ASSIGNMENT

### 1.Windows 11

#### Steps:

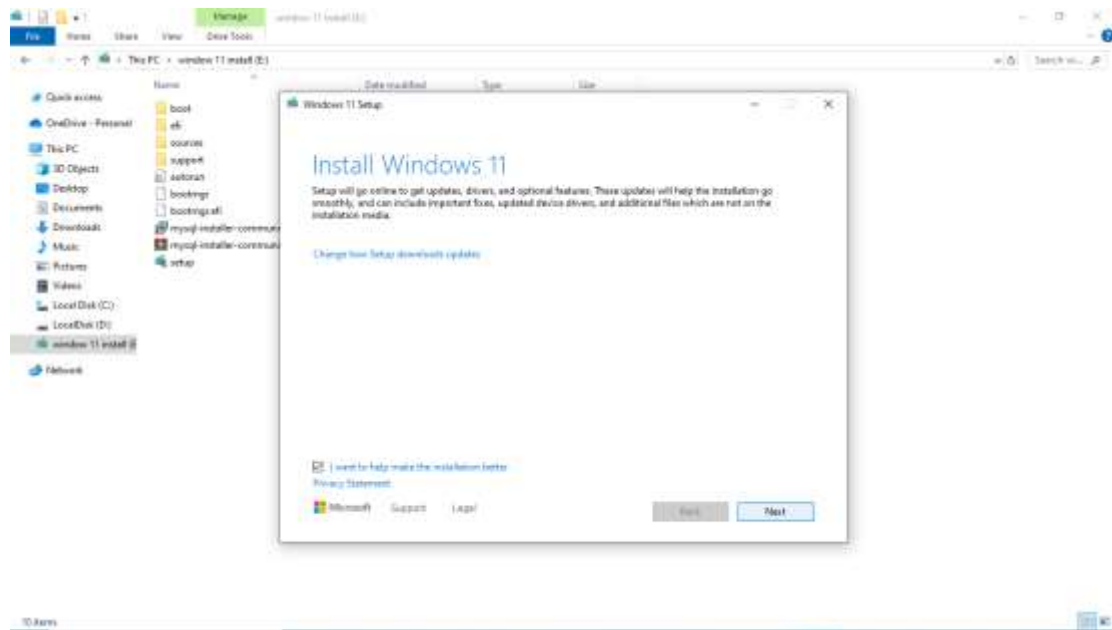
##### Step 1: Check System Requirements

Before downloading Windows 11, ensure your PC meets the following minimum system requirements:

- **Processor:** 1 gigahertz (GHz) or faster with at least 2 cores on a compatible 64-bit processor or System on a Chip (SoC).
- **RAM:** 4 GB or more.
- **Storage:** 64 GB or larger storage device.
- **System firmware:** UEFI, Secure Boot capable.
- **TPM:** Trusted Platform Module (TPM) version 2.0.
- **Graphics card:** DirectX 12 compatible graphics / WDDM 2.x.
- **Display:** >9" with HD Resolution (720p).
- **Internet connection:** Internet connectivity is necessary to perform updates and to download and take advantage of some features.
- **Step 2: Backup Your Data**
- Before upgrading, it's crucial to back up your important data to avoid any loss. You can use an external hard drive, cloud storage, or a backup software.
- **Step 3: Download the PC Health Check App**
- To confirm your device is compatible, download and run the PC Health Check app from Microsoft's official website. This tool will let you know if your PC meets the Windows 11 requirements.
- **Step 4: Obtain Windows 11**
- You can download Windows 11 through various methods:

### Windows Update

1. **Open Settings:** Click on the Start menu and select the settings icon.
2. **Navigate to Update & Security:** Click on 'Update & Security'.
3. **Check for Updates:** Click on 'Check for updates'. If Windows 11 is available for your device, you'll see an option to download and install.



## Step 5: Install Windows 11

Follow the on-screen prompts to complete the installation. Ensure you have a stable internet connection and sufficient battery life (for laptops) or are connected to a power source.

## Step 6: Post-Installation Setup

After the installation is complete:

1. **Update Drivers:** Check for any driver updates to ensure your hardware is fully compatible.
2. **Restore Data:** Restore your backed-up data.
3. **Install Applications:** Reinstall any necessary applications.

## 2. Visual studio code

### Step 1: Visit the Visual Studio Code Website

Open your web browser and go to the Visual Studio Code download page.

### Step 2: Download Visual Studio Code

Click on the download button for your operating system (Windows, macOS, or Linux). For this guide, I'll use Windows as an example.

### Step 3: Run the Installer

Locate the downloaded file (usually in your Downloads folder) and double-click on it to run the installer.

#### Step 4: Accept the License Agreement

Read the license agreement, check the box to accept the terms, and click "Next".

#### Step 5: Select Destination Location

Choose the installation location or leave the default location, and click "Next".

#### Step 6: Select Additional Tasks

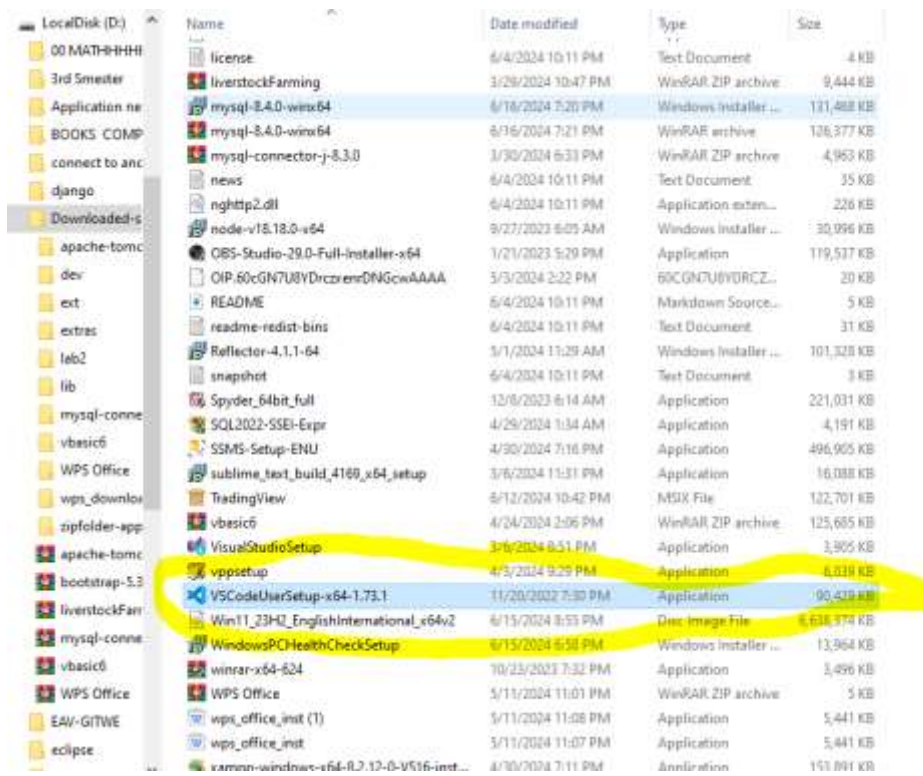
Choose additional tasks such as creating a desktop icon or adding to PATH, and click "Next".

#### Step 7: Install Visual Studio Code

Click "Install" to begin the installation process.

#### Step 8: Complete the Installation

Click "Finish" after the installation is complete. You can choose to launch Visual Studio Code immediately.

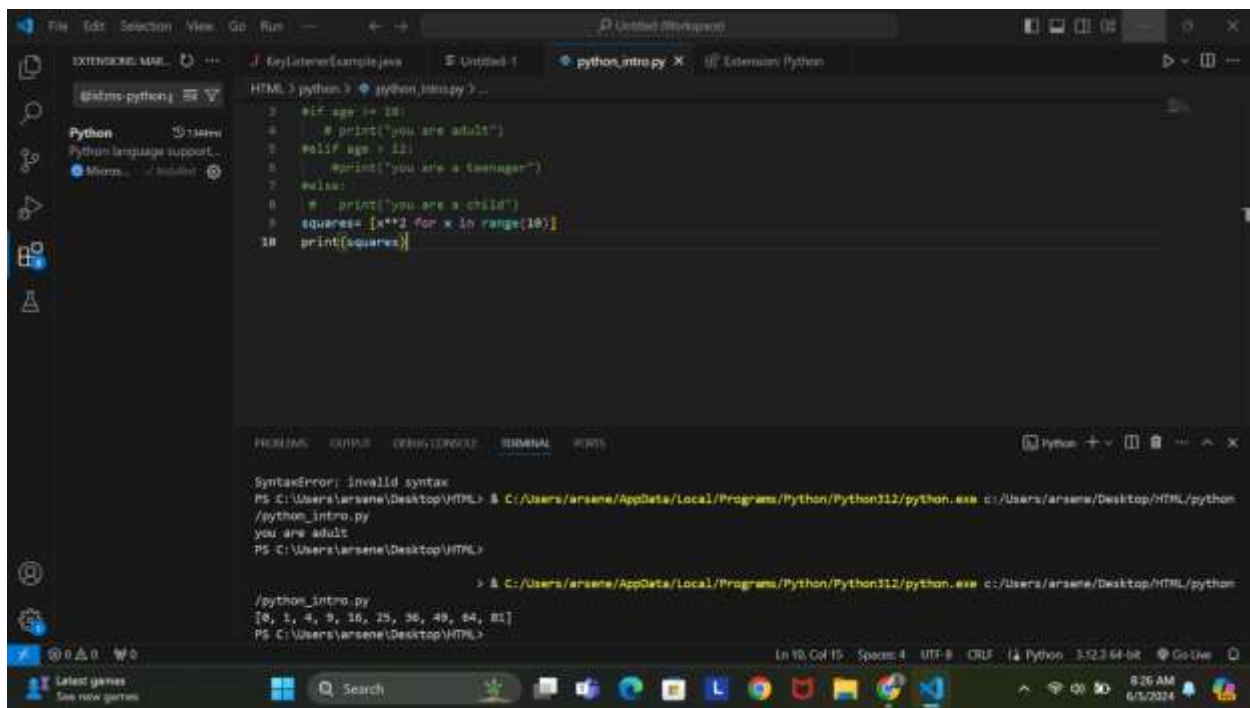
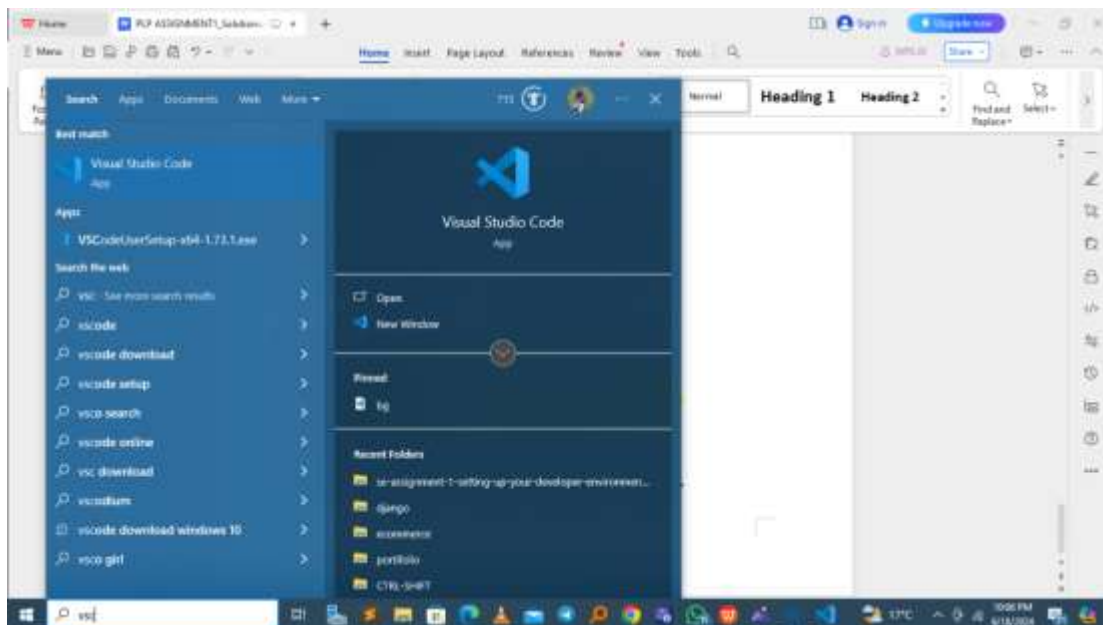


#### Step 9: Launch Visual Studio Code

Open Visual Studio Code if it's not already open. You can find it in the Start menu or on your desktop if you created an icon.

## Step 10: Customize and Start Coding

Customize your setup by installing extensions, setting up themes, and configuring settings as needed. You can access these options from the sidebar and the Command Palette (Ctrl+Shift+P).



### **3.GitHub account**

#### **Step 1: Create a GitHub Account**

Visit the GitHub website: Go to [GitHub](#).

#### **1.Sign up for GitHub:**

Click on the "Sign up" button.

Enter your email address and click "Continue".

Create a password and click "Continue".

Choose a username and click "Continue".

Verify your account by solving the puzzle and click "Create account".

Optionally, choose your plan (free or paid) and complete the signup process.

#### **Step 2: Initialize a Git Repository**

1.Install Git: If you don't have Git installed, download and install it from the official Git website.

2.Open Terminal or Command Prompt:

On Windows: Open Command Prompt or Git Bash.

On macOS or Linux: Open Terminal.

3.Navigate to your project directory: Use the cd command to go to the directory where your project is located. For example:

**cd path/to/your/project**

4.Initialize a Git repository: Run the following command:

**git init**

5.Add your project files to the repository: Use the git add command to add all files. For example:

**git add .**

6.Make your first commit: Commit the added files with a commit message:

**git commit -m "Initial commit"**

### **Step 3: Push to GitHub**

1.Create a new repository on GitHub:

Go to your GitHub homepage.

Click on the "New" button to create a new repository.

Enter the repository name and description (optional).

Choose the repository type (public or private).

Click on "Create repository".

Link your local repository to GitHub:

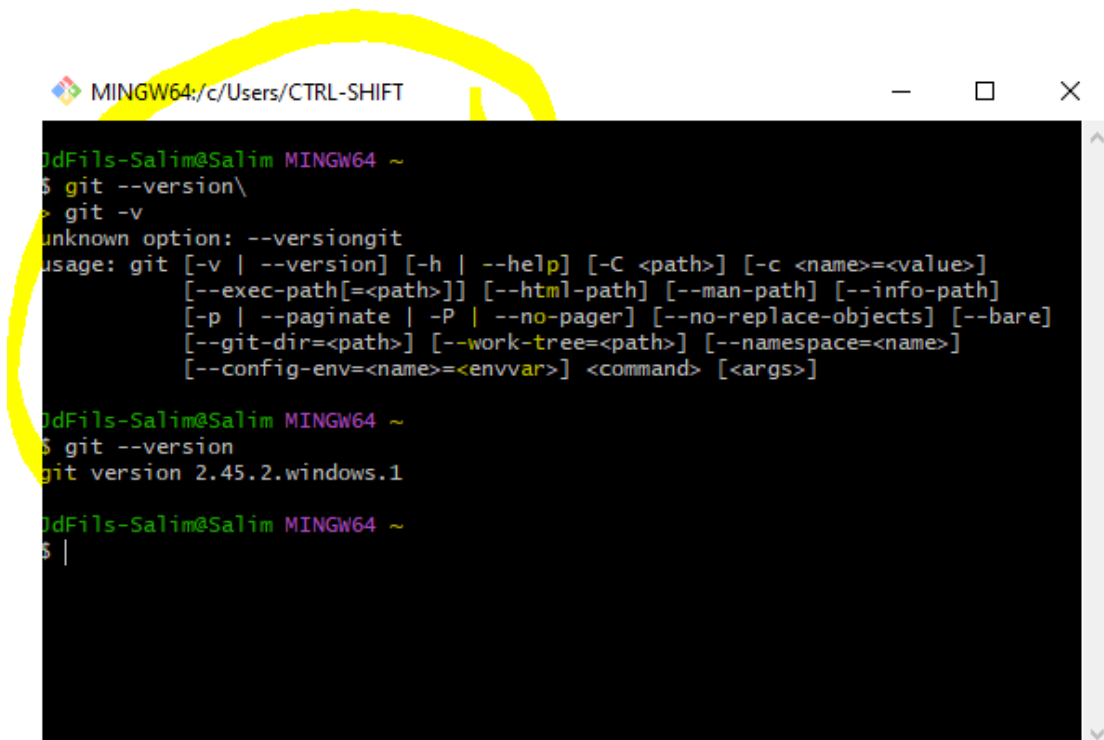
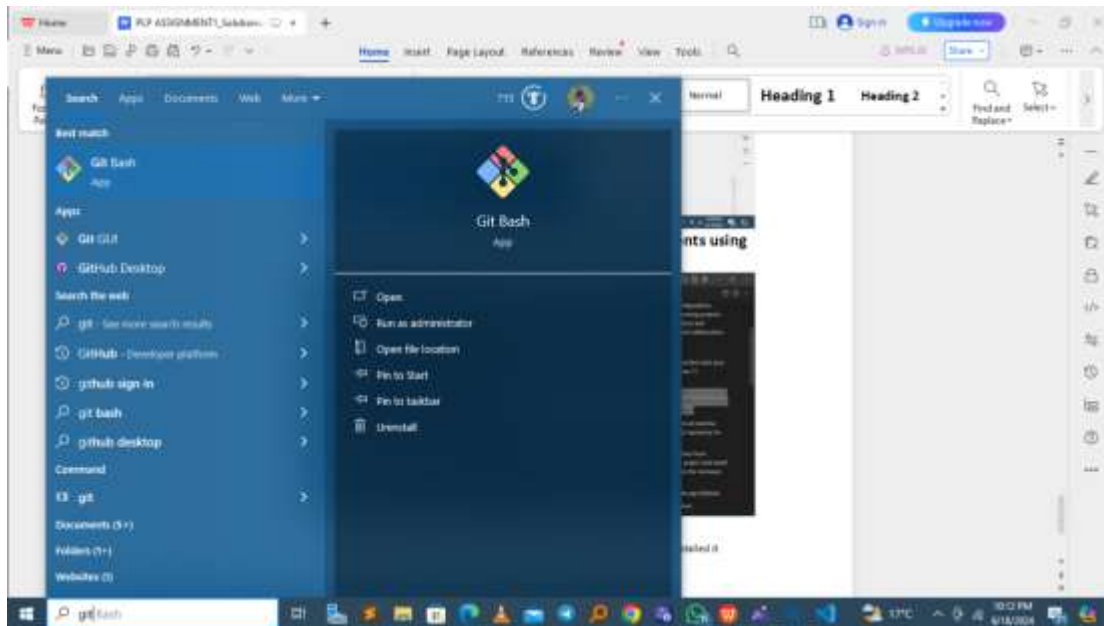
2.Copy the URL of your new GitHub repository.

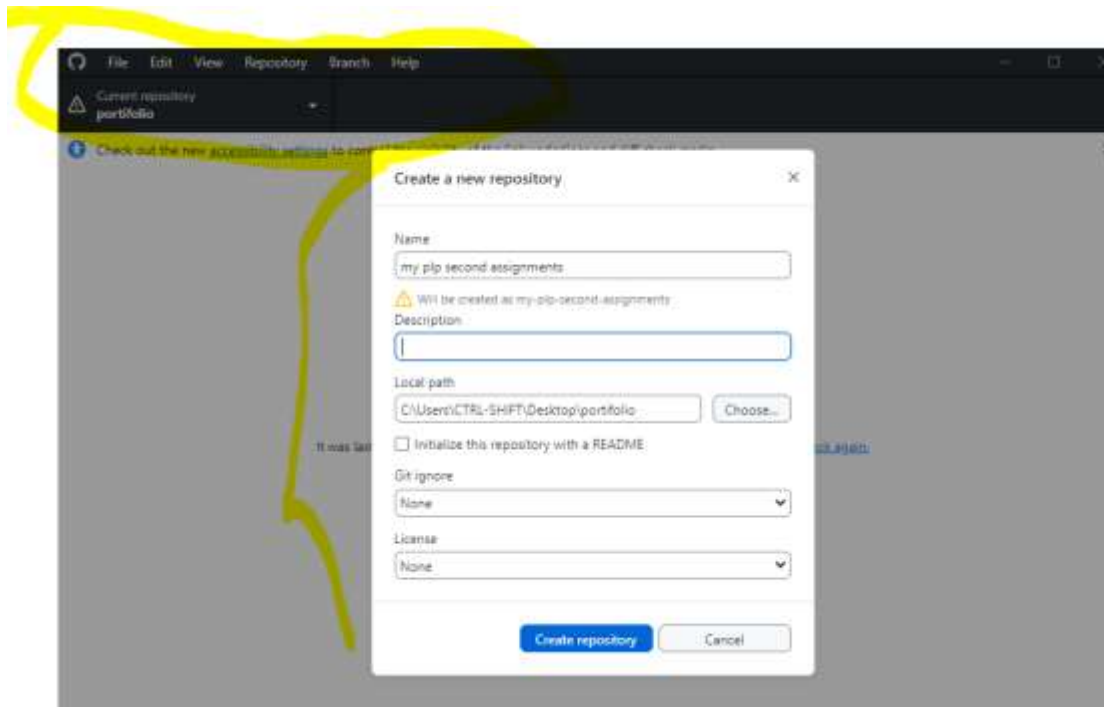
In your terminal, link your local repository to the remote GitHub repository:

**git remote add origin https://github.com/your-username/your-repository.git**

3.Push your code to GitHub: Push the committed code to the master/main branch:

**git push -u origin master**





## 4.PYTHON

### Step 1: Visit the Python Website

Open your web browser and go to the official Python website.

### Step 2: Download Python

Navigate to the Downloads section: Click on the "Downloads" tab.

Select the appropriate version for your operating system:

For Windows, click on the "Download Python 3.x.x" button (the latest version).

For macOS, click on the "Download Python 3.x.x" button.

For Linux, follow the instructions provided on the page or use your package manager.

### Step 3: Install Python

Windows



Run the installer: Locate the downloaded file (usually in your Downloads folder) and double-click on it to run the installer.

Add Python to PATH: Check the box that says **"Add Python 3.x to PATH"**.

Install Python: Click on "Install Now" and follow the prompts to complete the installation.

#### **Step 4: Verify Python Installation**

1. Open a terminal or command prompt:

Check Python version: Type the following command and press Enter:

**python --version**

For Python 3, you might need to use python3 --version.

#### **Step 5: Install Necessary Tools**

Install pip (Python package installer):

Pip is included with Python 3.4+ by default. Verify by typing:

**pip --version**

If pip is not installed, you can install it using:

**python -m ensurepip --upgrade**

Install virtualenv (optional but recommended for project isolation):

**pip install virtualenv**

#### **Step 6: Set Up a Virtual Environment**

1. Create a virtual environment:

virtualenv venv

For Python 3:

**python3 -m venv venv**

Activate the virtual environment:

On Windows:

**venv\Scripts\activate**

On macOS/Linux:

**source venv/bin/activate**

## Step 7: Install Project Dependencies

Create a requirements.txt file in your project directory listing all necessary packages.

Install dependencies:

**pip install -r requirements.txt**

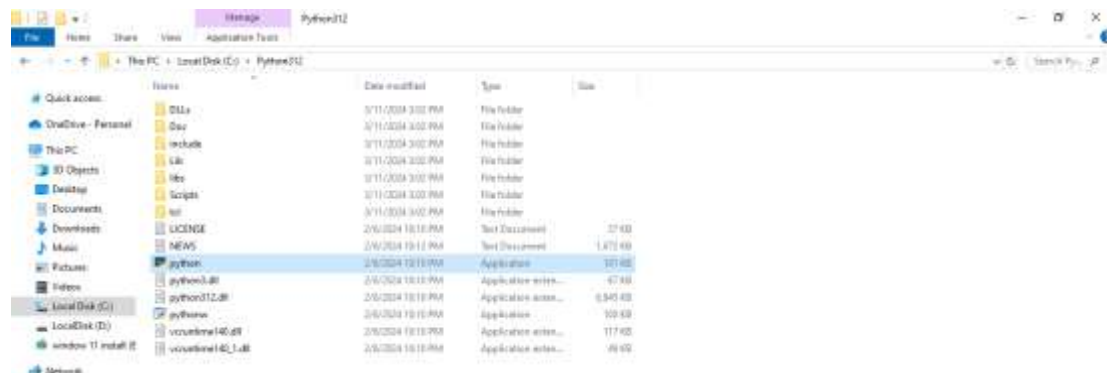
## Step 8: Run Your Python Code

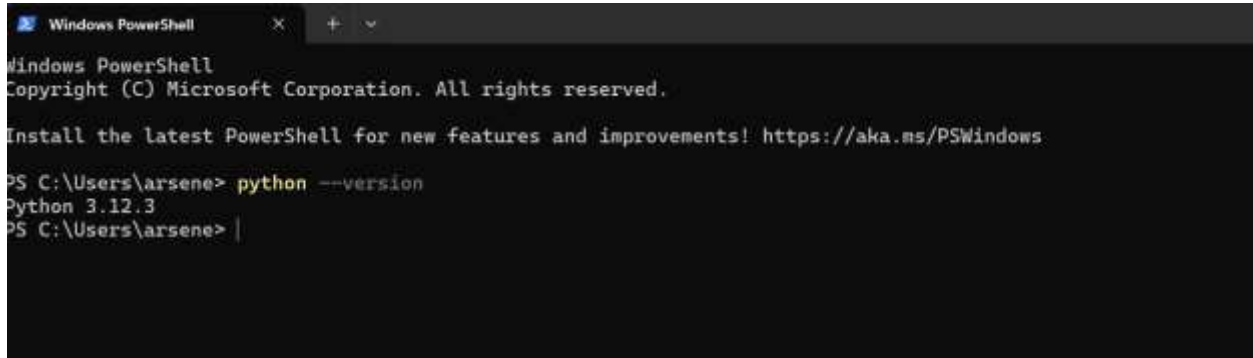
Execute your Python script: Make sure you are in your project directory and run:

**python your\_script.py**

For Python 3:

**python3 your\_script.py**



A screenshot of a Windows PowerShell terminal window. The title bar shows 'Windows PowerShell' with standard window controls. The terminal text includes: 'Windows PowerShell', 'Copyright (C) Microsoft Corporation. All rights reserved.', 'Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows', and a command prompt session where 'python --version' is entered, resulting in the output 'Python 3.12.3'.

## 5.PIP

### Step 1: Ensure Python

is Installed

Check if Python is installed:

Open your terminal or command prompt.

Type the following command and press Enter:

**python --version**

For Python 3, you might need to use **python3 --version**.

If Python is not installed, follow the steps from my previous guide to install Python.

### Step 2: Verify if pip is Installed

Check if pip is installed:

In your terminal or command prompt, type the following command and press Enter:

**pip --version**

For Python 3, you might need to use **pip3 --version**.

If pip is not installed, proceed to the next step to install it.

### Step 3: Install pip (if not already installed)

Windows

1.Download get-pip.py:

Open your web browser and go to the official pip installation page: [get-pip.py](https://bootstrap.pypa.io/get-pip.py).

Right-click on the page and select "Save As" to download the get-pip.py file.

2.Run the get-pip.py script:

Open your terminal or command prompt.

Navigate to the directory where you saved get-pip.py using the cd command. For example:

**cd path\to\directory**

Run the script by typing:

**python get-pip.py**

For Python 3:

**python3 get-pip.py**

**Step 4: Verify pip Installation**

1.Check if pip is installed correctly:

In your terminal or command prompt, type the following command and press Enter:

**pip --version**

For Python 3, you might need to use **pip3 --version**.

You should see the pip version information, confirming that pip is installed.

**Step 5: Install Virtualenv (Optional but Recommended)**

1.Install virtualenv using pip:

In your terminal or command prompt, type the following command and press Enter:

## **pip install virtualenv**

For Python 3:

## **pip3 install virtualenv**

### **Step 6: Set Up a Virtual Environment**

1.Create a virtual environment:

Navigate to your project directory in the terminal.

Type the following command to create a virtual environment:

## **virtualenv venv**

For Python 3:

## **python3 -m venv venv**

2.Activate the virtual environment:

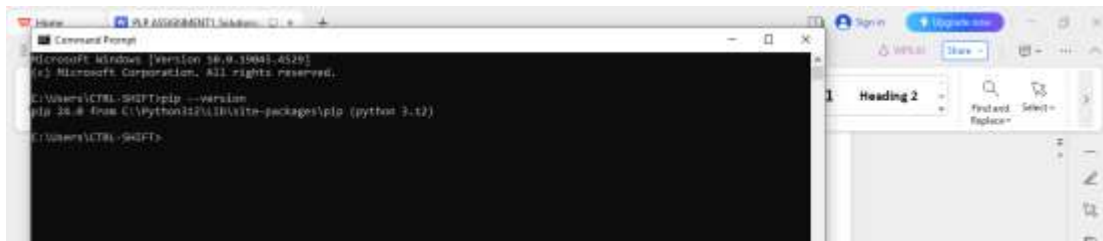
On Windows:

## **venv\Scripts\activate**

3.Deactivate the virtual environment:

To deactivate the virtual environment, simply type:

## **Deactivate**



## **6.MY-SQL**

### **Step 1: Download MySQL**

1.Visit the MySQL website:

Open your web browser and go to the official MySQL download page.

2.Choose the appropriate installer:

Select your operating system (Windows, macOS, Linux) and click on the corresponding download link.

3.Download the installer:

Choose the version you need and click on the "Download" button.

You may be prompted to log in or sign up for an Oracle account. You can skip this by clicking "No thanks, just start my download".

## **Step 2: Install MySQL**

Windows

1.Run the installer:

Locate the downloaded file (usually in your Downloads folder) and double-click on it to run the installer.

2.Choose setup type:

Choose a setup type (Typical, Custom, or Complete). The "Developer Default" option is recommended for most users.

3.Install MySQL:

Click on "Execute" to download and install the required MySQL products.

4.Complete the installation process:

Follow the on-screen instructions to complete the installation process.

## **Step 3: Secure MySQL Installation**

1.Run the MySQL secure installation script:

Open your terminal or command prompt.

Run the following command to secure your MySQL installation:

**sudo mysql\_secure\_installation**

2.Follow the prompts to set up security options:

Set a strong root password.

Remove anonymous users.

Disallow root login remotely.

Remove the test database.

Reload privilege tables.

#### **Step 4: Verify MySQL Installation**

1.Log in to MySQL:

Open your terminal or command prompt.

Log in to the MySQL server using the root user:

**mysql -u root -p**

Enter the root password you set during the secure installation.

2.Check the MySQL server status:

After logging in, check the MySQL server status:

**SHOW VARIABLES LIKE "%version%";**

#### **Step 5: Install MySQL Workbench (Optional)**

1.Download MySQL Workbench:

Go to the MySQL Workbench download page.

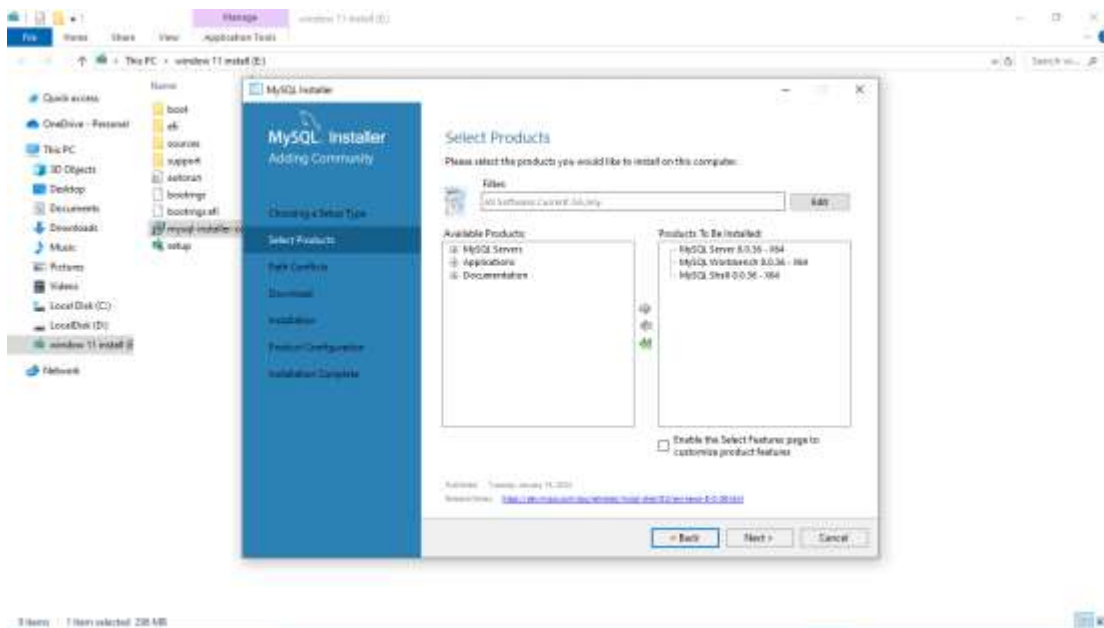
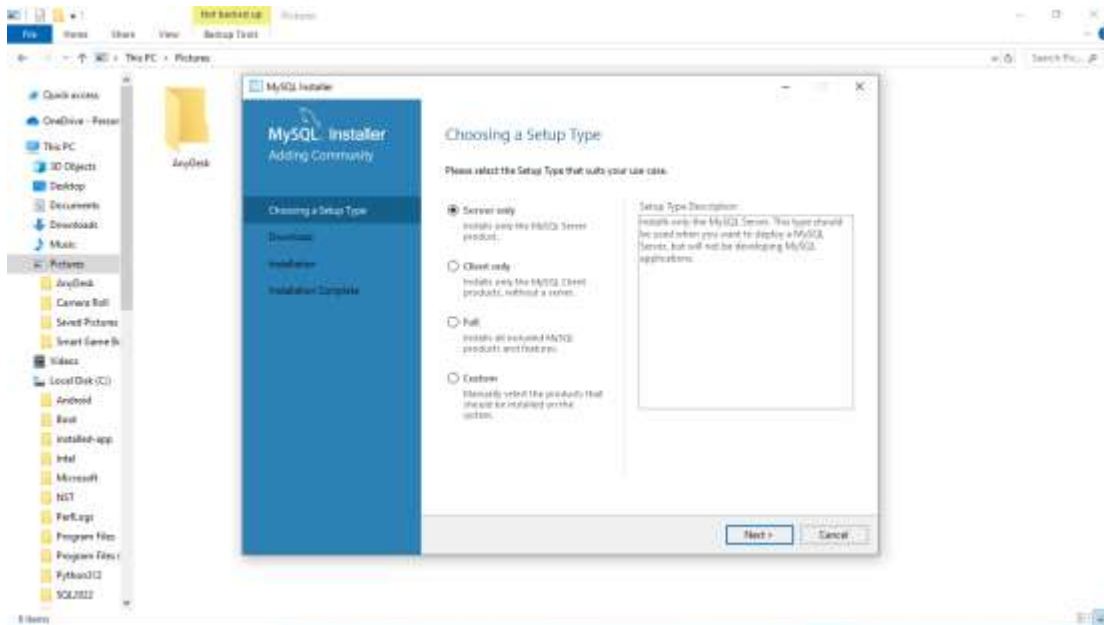
Select your operating system and download the installer.

## 2.Install MySQL Workbench:

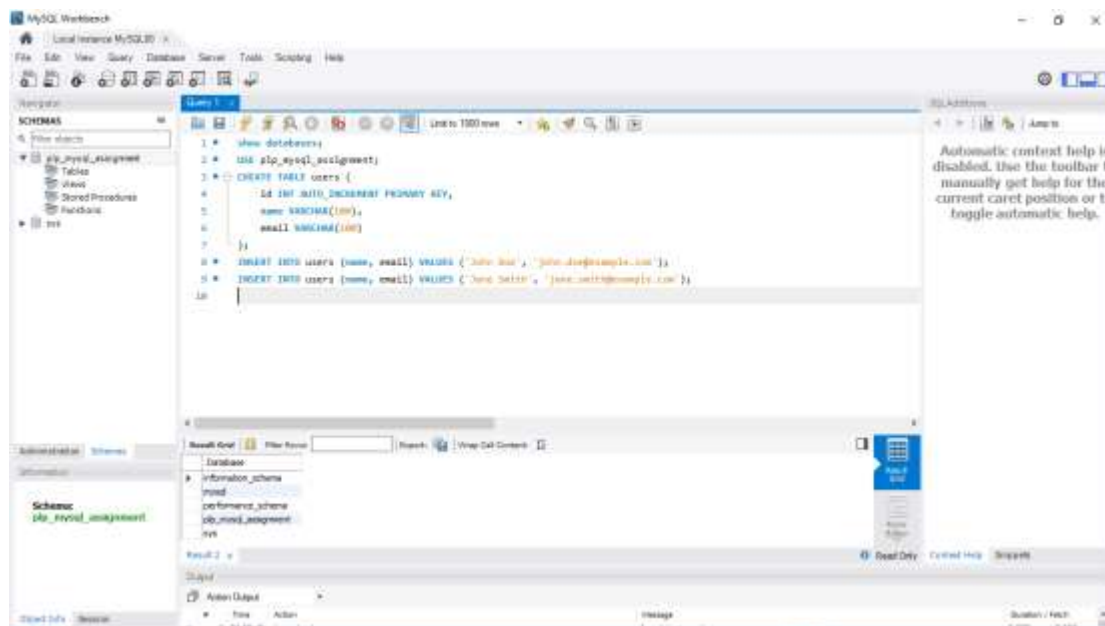
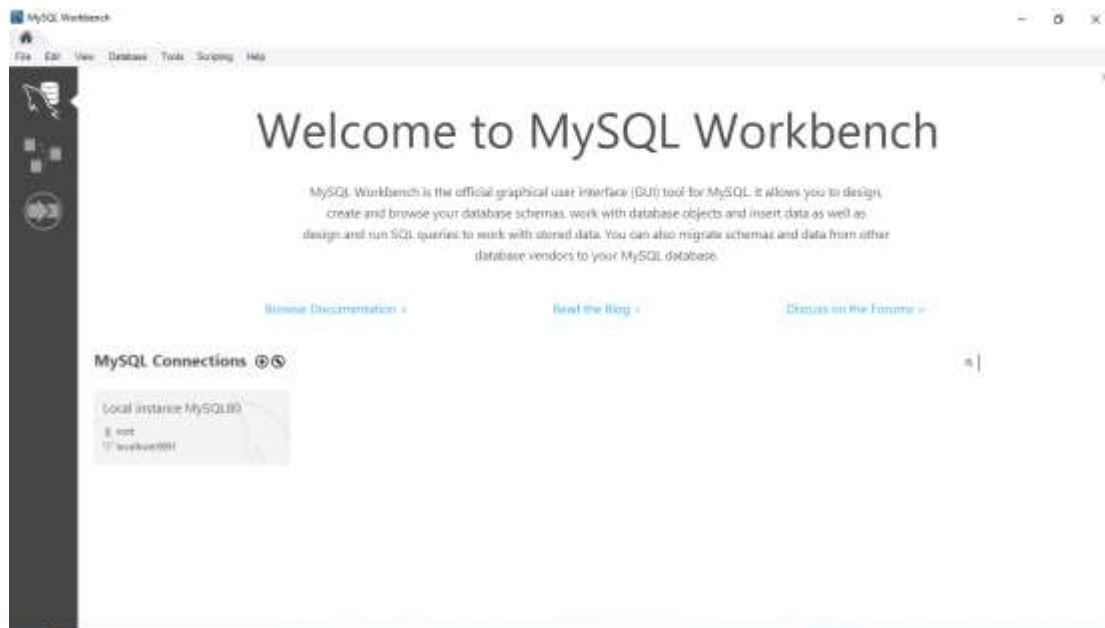
Run the installer and follow the on-screen instructions to complete the installation.

## 3.Open MySQL Workbench:

Launch MySQL Workbench and create a new connection to your MySQL server.







## 7. Development Environments and Virtualization

### Using Docker

#### Step 1: Install Docker

##### 1. Windows and macOS

Download Docker Desktop:

Visit the Docker Desktop download page and download the appropriate installer for your operating system.

##### 2. Run the installer:

Double-click the downloaded file and follow the on-screen instructions to install Docker Desktop.

3. Start Docker Desktop:

After installation, launch Docker Desktop from the Start menu (Windows) or Applications folder (macOS).

## Step 2: Verify Docker Installation

1. Open a terminal or command prompt:
2. Run the hello-world container:

```
docker run hello-world
```

This command downloads a test image and runs it in a container. If Docker is installed correctly, you will see a "Hello from Docker!" message.

Step 3: Set Up Your Project with Docker  
Create a Dockerfile:

In your project directory, create a file named Dockerfile with the following content:

```
# Use the official Python image from the Docker Hub
FROM python:3.9
```

```
# Set the working directory in the container
WORKDIR /app
```

```
# Copy the current directory contents into the container at /app
COPY . /app
```

```
# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt
```

```
# Make port 80 available to the world outside this container
EXPOSE 80
```

```
# Define environment variable
ENV NAME World
```

```
# Run app.py when the container launches
CMD ["python", "app.py"]
```

2. Create a requirements.txt file:

List your project dependencies in a file named requirements.txt in the same directory.  
Copy code

flask  
requests

### 3. Build the Docker image:

Open your terminal or command prompt in the directory containing the Dockerfile and run:

**docker build -t my-python-app .**

### 4. Run the Docker container:

Run your container with the following command:

**docker run -p 4000:80 my-python-app**

This command maps port 80 in the container to port 4000 on your host machine.

### 5. Test your application:

Open your web browser and go to <http://localhost:4000>. Your application should be running inside the Docker container.

## Using Virtual Machines

### **Step 1:** Install a Virtual Machine Manager

#### 1. Download and install VirtualBox:

Visit the VirtualBox download page and download the appropriate installer for your operating system.

#### 2. Run the installer:

Follow the on-screen instructions to complete the installation.

### **Step 2:** Set Up a Virtual Machine

#### 1. Create a new VM:

Open VirtualBox and click on "New".

Follow the prompts to create a new virtual machine. Choose the appropriate settings for your operating system.

#### 2. Install the operating system:

Download an ISO file of the operating system you want to install (e.g., Ubuntu).

Start the virtual machine and follow the prompts to install the operating system using the ISO file.

### **Step 3:** Install Project Dependencies

#### 1. Set up your development environment:

Once the operating system is installed, open a terminal in your VM.

#### 2. Install project dependencies:

Copy your project files to the VM.  
Navigate to your project directory and install dependencies:  
**pip install -r requirements.txt**

#### **Step 4: Run Your Application**

1.Run your Python application:  
Navigate to your project directory and run:  
**python app.py**

### **8. Explore Extensions and Plugins:**

#### **Visual Studio Code (VS Code)**

.

#### **Step 1: Install Extensions in VS Code**

1.Open VS Code:

Launch Visual Studio Code from your applications menu.

**2.**Access the Extensions view:

Click on the Extensions icon in the Activity Bar on the side of the window or press Ctrl+Shift+X.

**3.**Search for Extensions:

In the Extensions view, type the name of the extension you want to install in the search box.

**4.**Install an Extension:

Click on the Install button next to the extension you want to install.

#### **Step 2: Recommended Extensions for VS Code**

**1.**Python Extension:

Provides rich support for the Python language, including features such as IntelliSense, linting, debugging, code formatting, and more.

**2.**Pylint:

Linters for Python to enforce coding standards and detect errors.

**3.**Black Formatter:

Code formatter for Python, enforcing a consistent style.

**4.**Prettier - Code Formatter:

An opinionated code formatter for various languages, ensuring consistent code style.

**5.**GitLens:

Supercharges the built-in Git capabilities, providing advanced features like blame annotations, code lens, and repository insights.

#### **6.Docker:**

Adds syntax highlighting, commands, and snippets for working with Docker.

### **PyCharm**

PyCharm is a popular IDE for Python development, known for its powerful features and integrated tools.

#### **Step 1: Install Plugins in PyCharm**

##### 1.Open PyCharm:

Launch PyCharm from your applications menu.

##### 2.Access the Plugins view:

Go to File > Settings > Plugins (on Windows/Linux) or PyCharm > Preferences > Plugins (on macOS).

##### 3.Search for Plugins:

In the Plugins view, type the name of the plugin you want to install in the search box.

##### 4.Install a Plugin:

Click on the Install button next to the plugin you want to install.

#### **Step 2: Recommended Plugins for PyCharm**

##### 1.Pylint:

Linters for Python to enforce coding standards and detect errors.

##### 2.Black Formatter:

Code formatter for Python, enforcing a consistent style.

##### 3.Docker Integration:

Adds support for Docker, enabling you to manage Docker containers and images directly from PyCharm.

##### 4..env Files Support:

Provides support for environment variables defined in .env files.

##### 5.Git ToolBox:

Enhances Git integration with additional features like inline blame, commit messages templates, and more.

### **Sublime Text**

Sublime Text is a lightweight and fast text editor with a powerful plugin ecosystem.

### **Step 1: Install Package Control**

#### **1. Open Sublime Text:**

Launch Sublime Text from your applications menu.

#### **2. Install Package Control:**

Open the console by pressing Ctrl+` or View > Show Console`.

Paste the following code into the console and press Enter:

```
import urllib.request,os,hashlib;
h = 'ebf4f2f9d13717391a379d6c054d43280b0b761a5f6d72e13f2d0a0ad3a3e7eb';
pf = 'Package Control.sublime-package';
ipp = sublime.installed_packages_path();
urllib.request.install_opener(urllib.request.build_opener(urllib.request.ProxyHandler()
));
by = urllib.request.urlopen('http://packagecontrol.io/' + pf.replace(' ', '%20')).read();
dh = hashlib.sha256(by).hexdigest();
if dh != h: raise Exception('Error validating download (got %s instead of %s)' % (dh, h));
with open(os.path.join(ipp, pf), 'wb') as f: f.write(by);
print('Package Control: Successfully Installed')
```

### **Step 2: Install Plugins in Sublime Text**

#### **1. Open the Command Palette:**

Press Ctrl+Shift+P or Cmd+Shift+P on macOS to open the Command Palette.

#### **2. Install a Package:**

Type Package Control: Install Package and press Enter.

Search for and select the plugin you want to install.

### **Step 3: Recommended Plugins for Sublime Text**

#### **1. Anaconda:**

Adds powerful Python IDE features to Sublime Text, including code linting, autocompletion, and more.

#### **2. SublimeLinter:**

A linter framework for Sublime Text that supports various linters, including Pylint for Python.

#### **3. Dockerfile Syntax Highlighting:**

Adds syntax highlighting for Dockerfiles.

#### **4. GitGutter:**

Shows information about files in a Git repository directly in the editor's gutter.

### **5.Black Formatter:**

Adds support for the Black code formatter for Python.

### **6.Terminus:**

A terminal emulator for Sublime Text

### **9.DOCUMENT YOUR SETUP**