Names: Jado Fils SEZIKEYE
Email: jasezikeye50@gmail.com

PLP ASSIGNMENT 1: Setting Up Your Developer Environment

Q1)Steps for installing window 11

Solutions

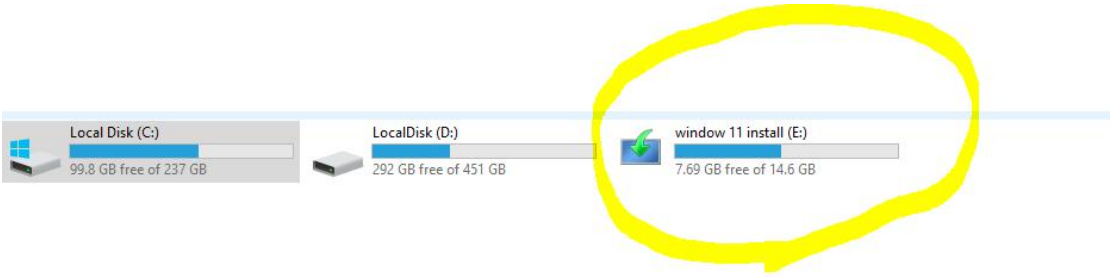
Method: Using Windows 11 Installation Assistant

Step1:

Choose an operating system that best suits your preferences and project requirements. Download and Install Windows 11.

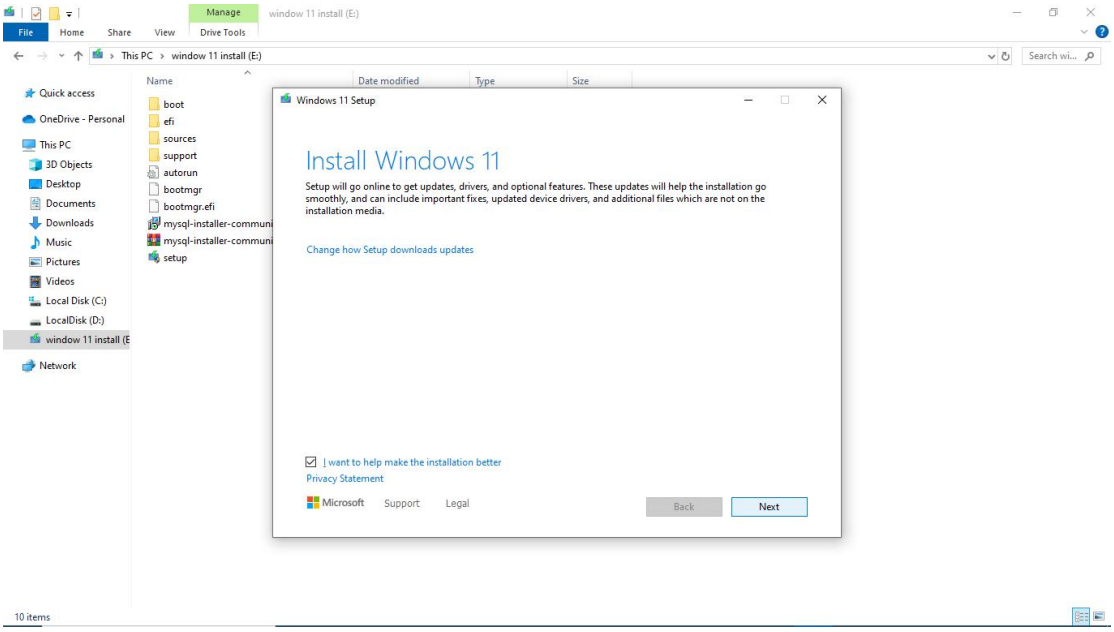
https://www.microsoft.com/software-download/windows11

Step2: Make your storage partitioning if you do not have usb, where you can install your window 11 as shown on screenshot.



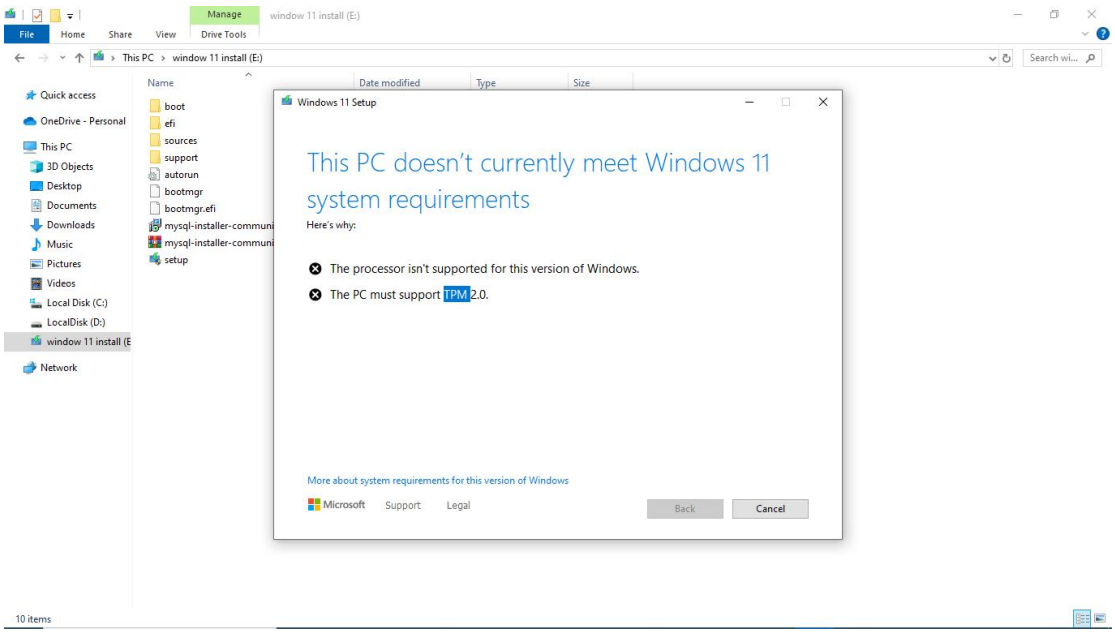
Step 2:

Mount your iso window 11 downloaded from the link given , hence navigate the sources folder inside, hence double click on setup, after some time will look like this.



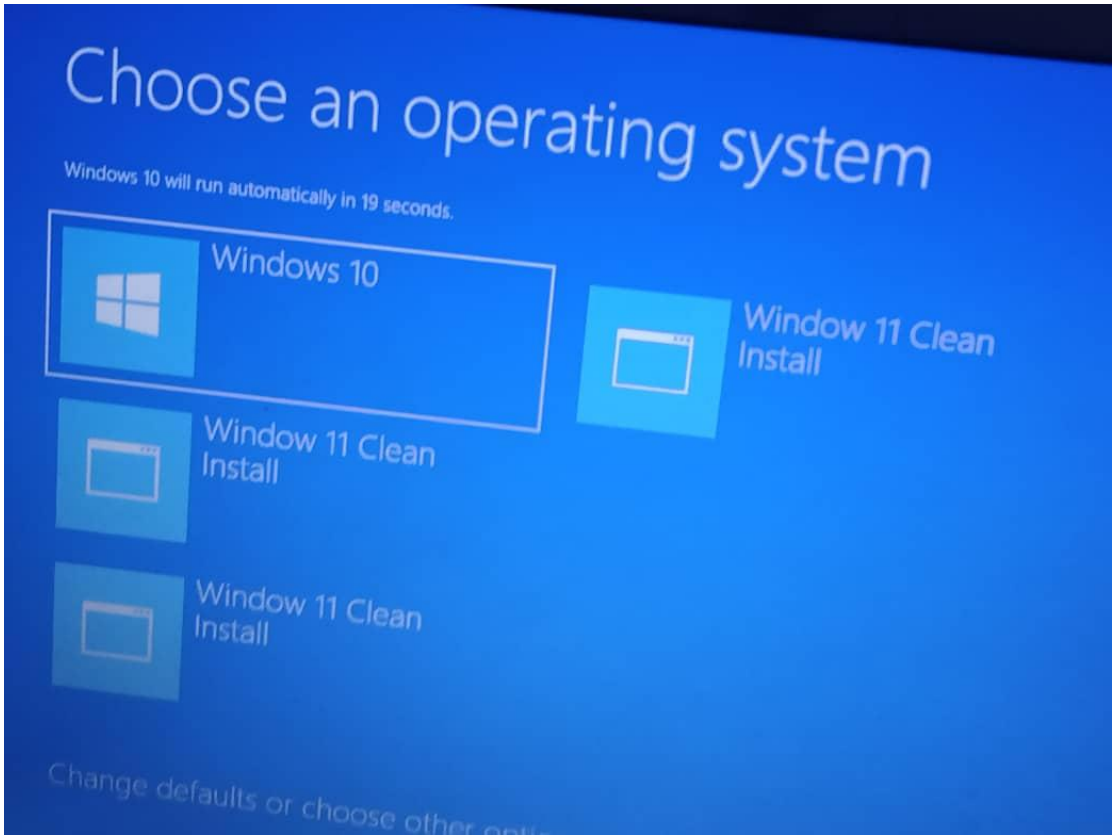
Step 3:

When being installed will check the updates and installing the window cores. If the window not compatible we will see the following output.



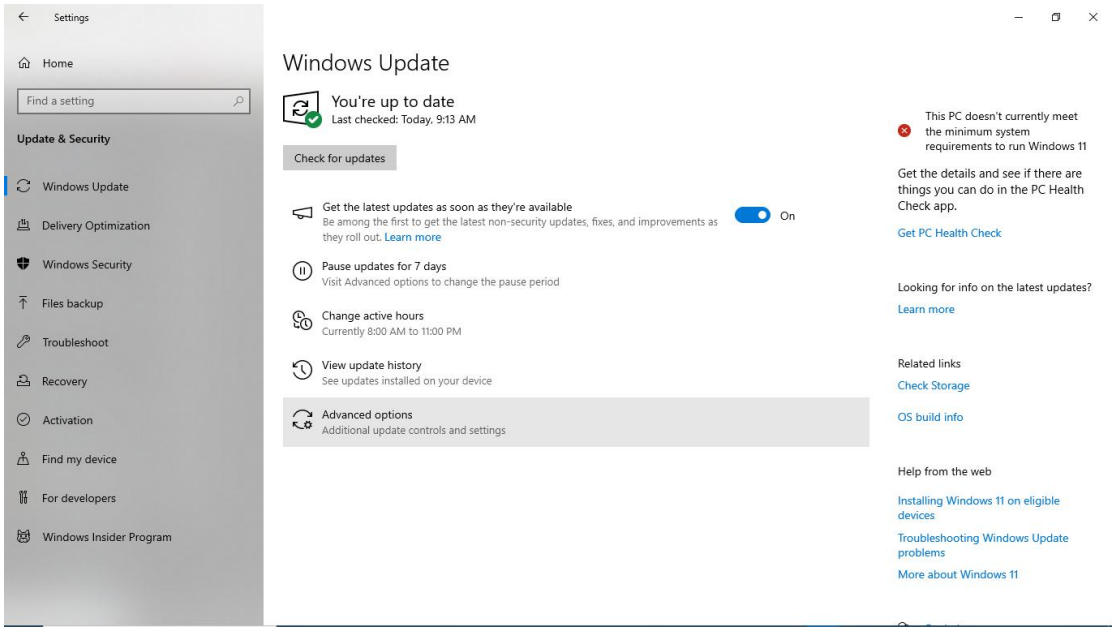
Otherwise will download the window cores.

Step 4: shutdown your pc and turn on again, hence will show the window 10 and window 11 to choose from them as like this.



But becouse the window 11 not compatible with my pc, if I choose window 11 will not work, because even if is looking here is from the one party I have partitioned where I have already downloaded and mounted my window 11.

Step 5: from the pc update you can see the alert message showing that your pc is not compatible like this.



Therefore, my pc is not compatible with windows 11, its why my installation looks like that.

2) Install a Text Editor or Integrated Development Environment (IDE) Select and install a text editor or IDE suitable for your programming languages and workflow. Download and Install Visual Studio Code. https://code.visualstudio.com/Download
To me I have already installed the visual studio code, and I have followed the following steps:

For Windows:

1.

Download the Installer:

2.

1. Visit the official VS Code website.
2. Click on the download button for Windows.
3.

Run the Installer:

4.

1. Once the download is complete, run the installer (`VSCodeUserSetup-{version}.exe`).

5.

Setup Wizard:

6.

1. Follow the prompts in the setup wizard.
2. Accept the license agreement.
3. Choose the destination folder.
4. Select additional tasks (like creating a desktop icon, adding to PATH, and registering code as an editor for supported file types).

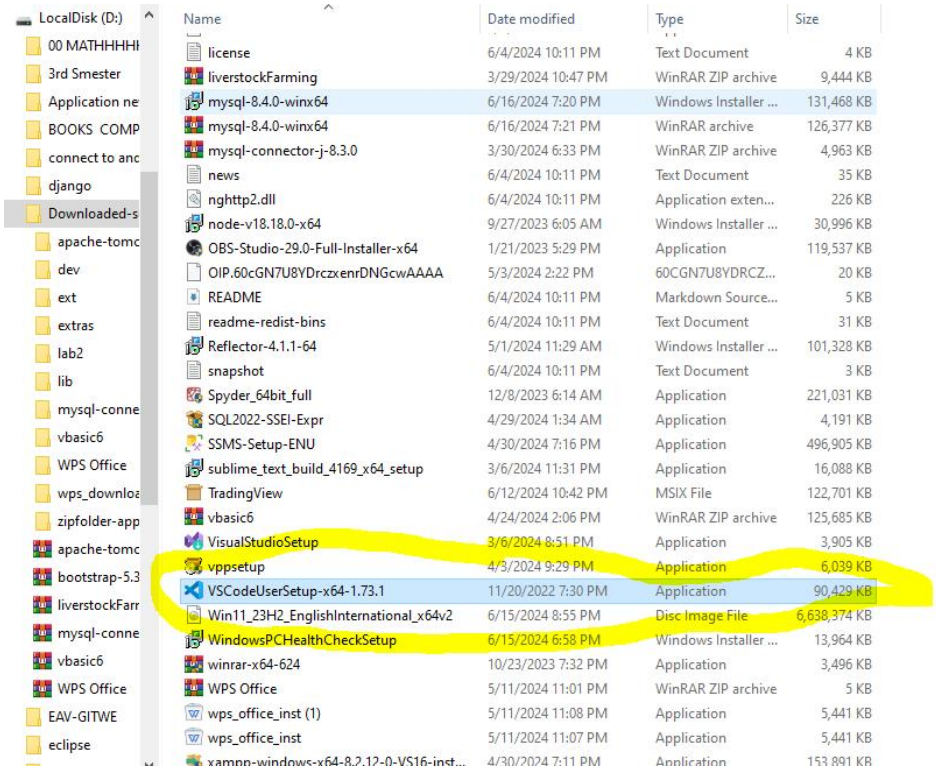
7.

Install:

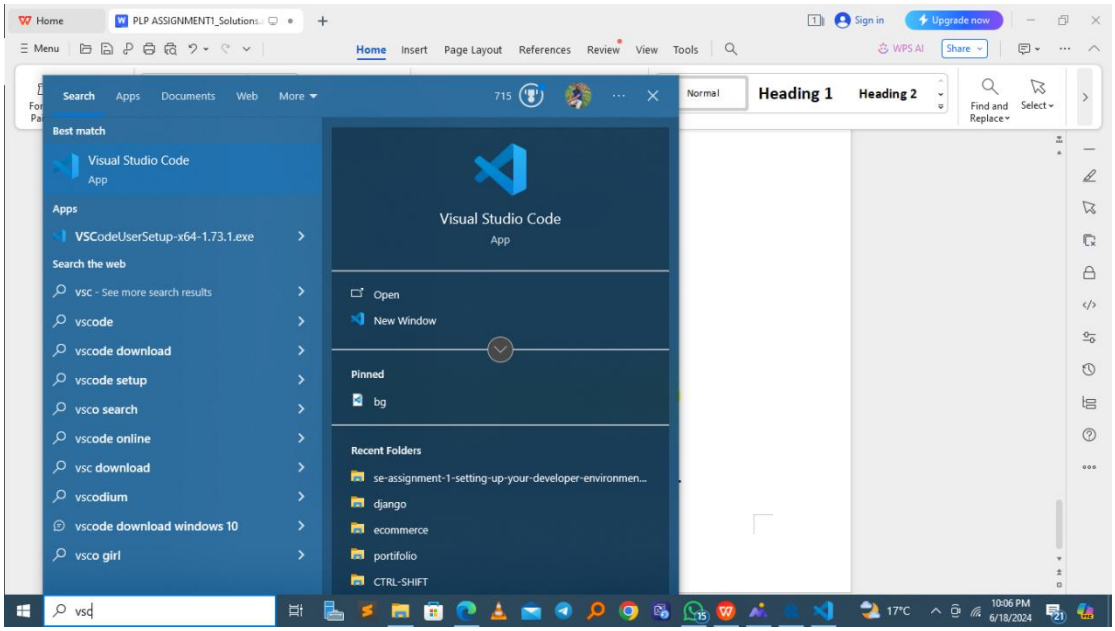
8.

1. Click on the `Install` button.
2. Once the installation is complete, you can launch VS Code.

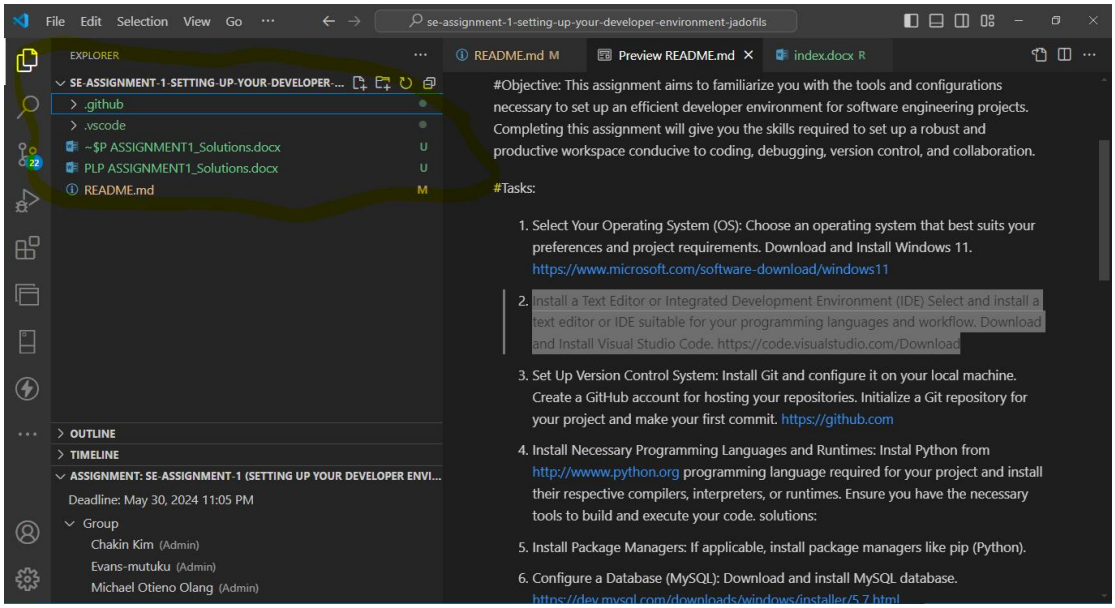
Therefore from my pc,
-Downloaded vscode is:



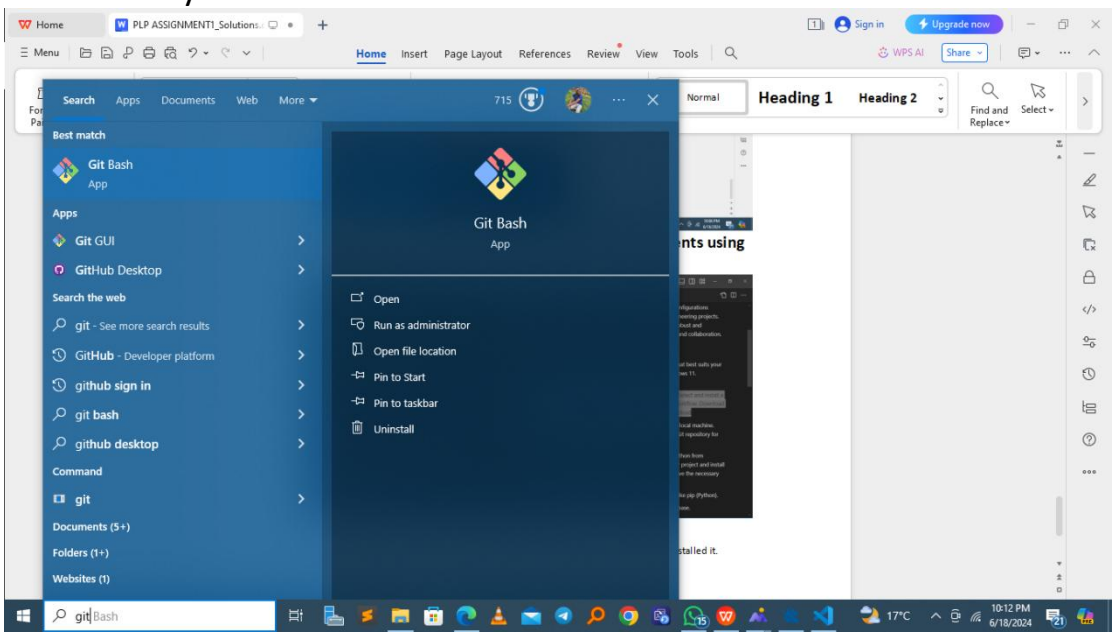
After installation I can search it from the search bar.

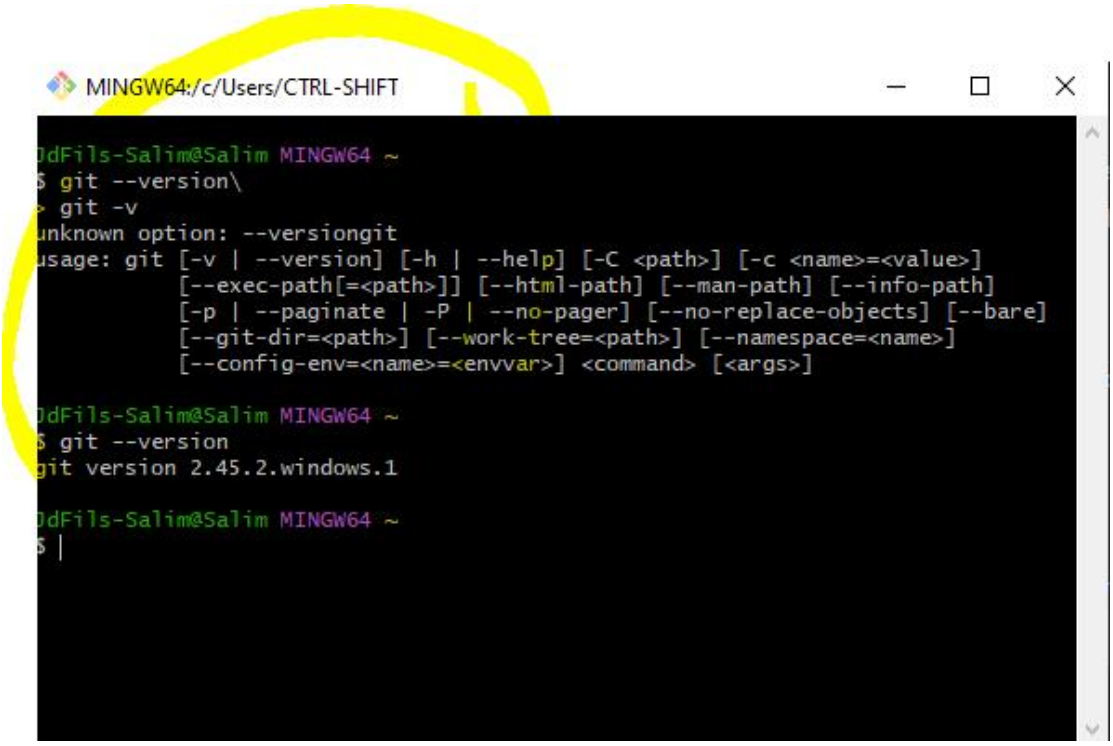


The I opened it and I am answering the plp assignments using the vscode.

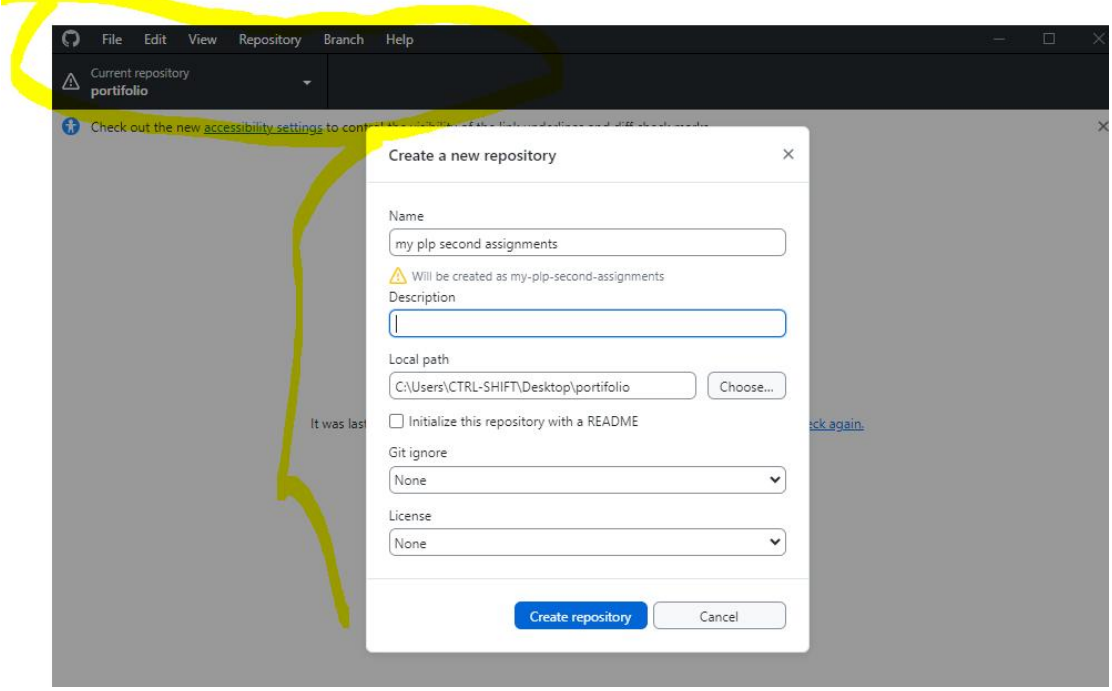


q3) Git hub installation, as you are going to see the screenshot here, I have already installed it.





And I have also downoaded and installing git desktop as shown here



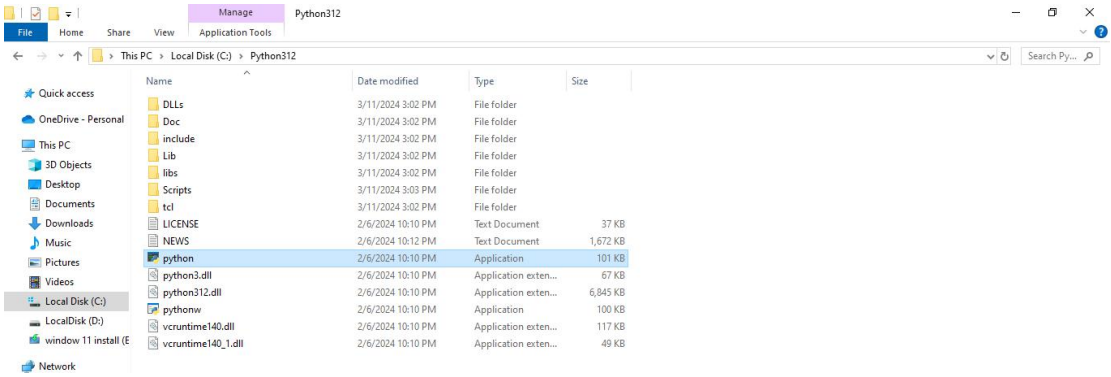
Sor far after this I have pushed different projects used different technologies, such as java, php,react and tailwind, and js.

You can visit my git hub account to check if it is true via :

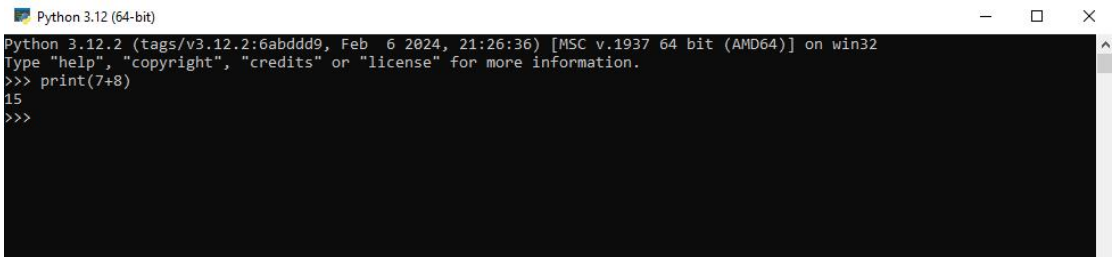
[jadofils(@Salim_Jd_Fils) GitHub]

3) I have also installed the python environments as shown here

First I have downloaded througth the given link
This is the folder where the python is installed.

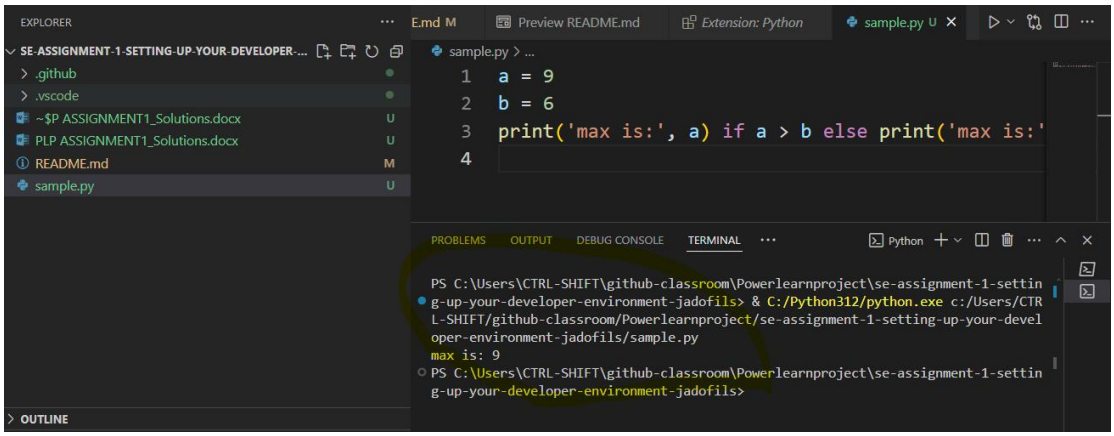


The python command line runtime



I also downloaded the python extension in vs code as hown below

So far I can run the python here sample example via my vscode

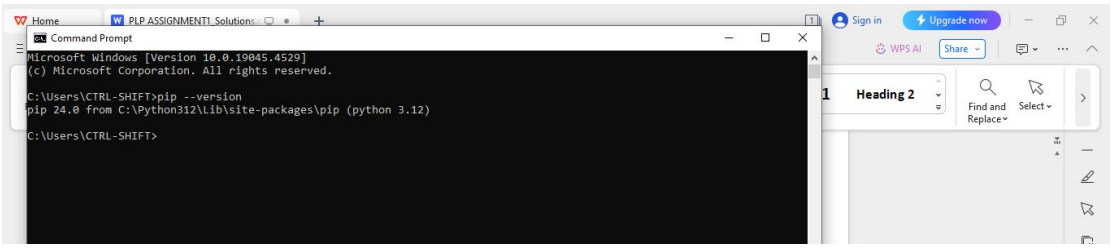


Q4) As python a high level programming language we need to install other dependencies and libraries for interacting with difference modules like pip and django if you want to deal with web application and backend side.

As you see the following is the process of installing pip latest version via the cmd

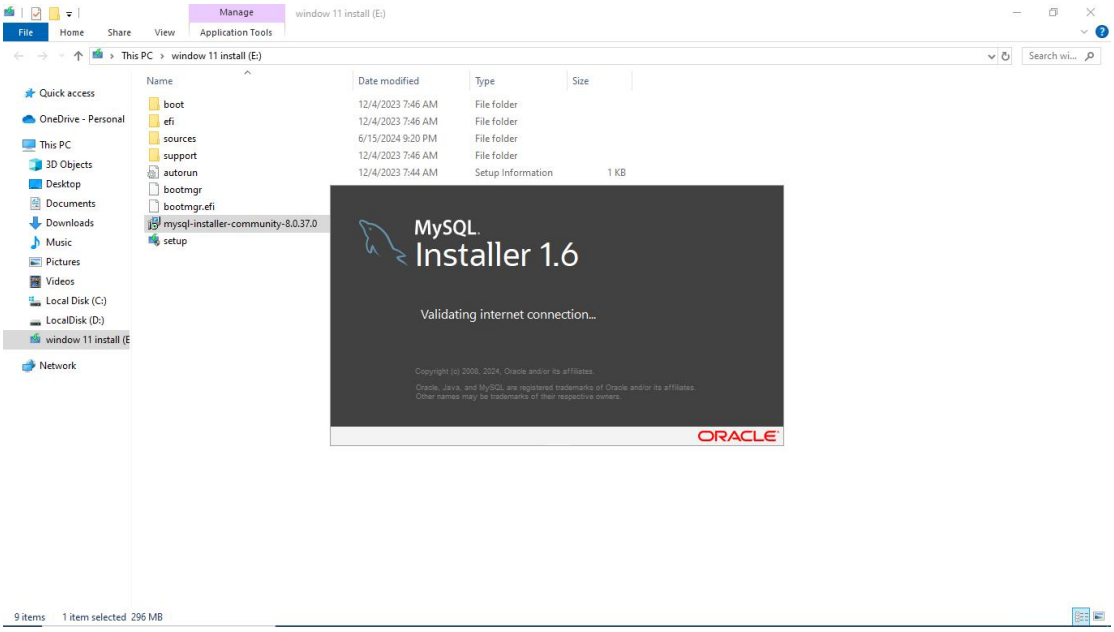
python get-pip.py

On my case I have already installed it

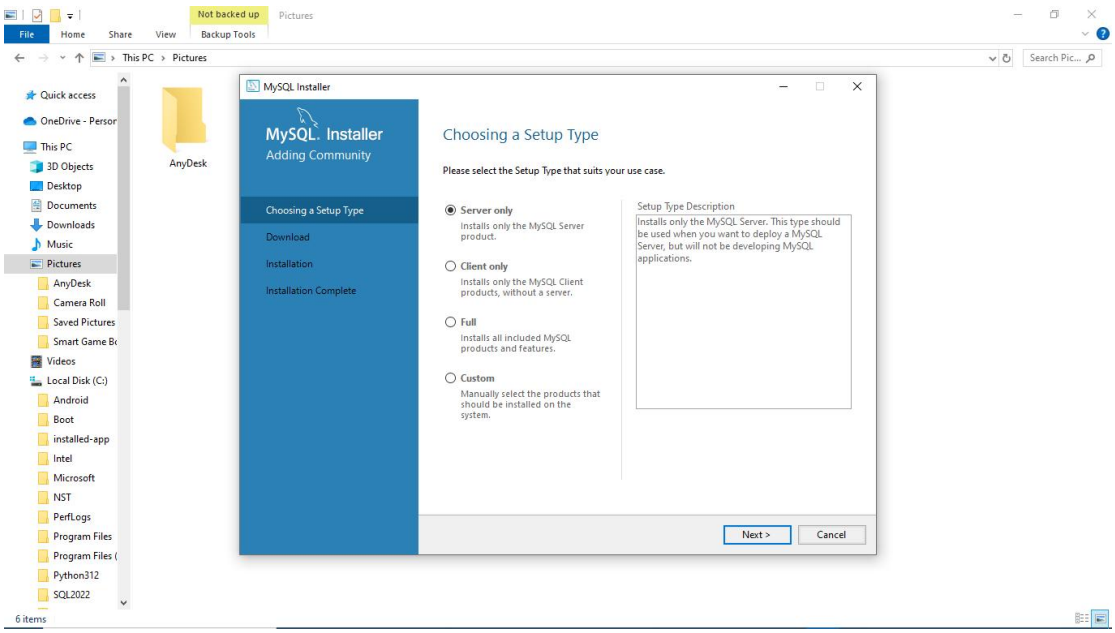


Q5)The following question is installing mysql

Step 1: through the link given I have downloaded the mysql and the following is the steps I have pass through.

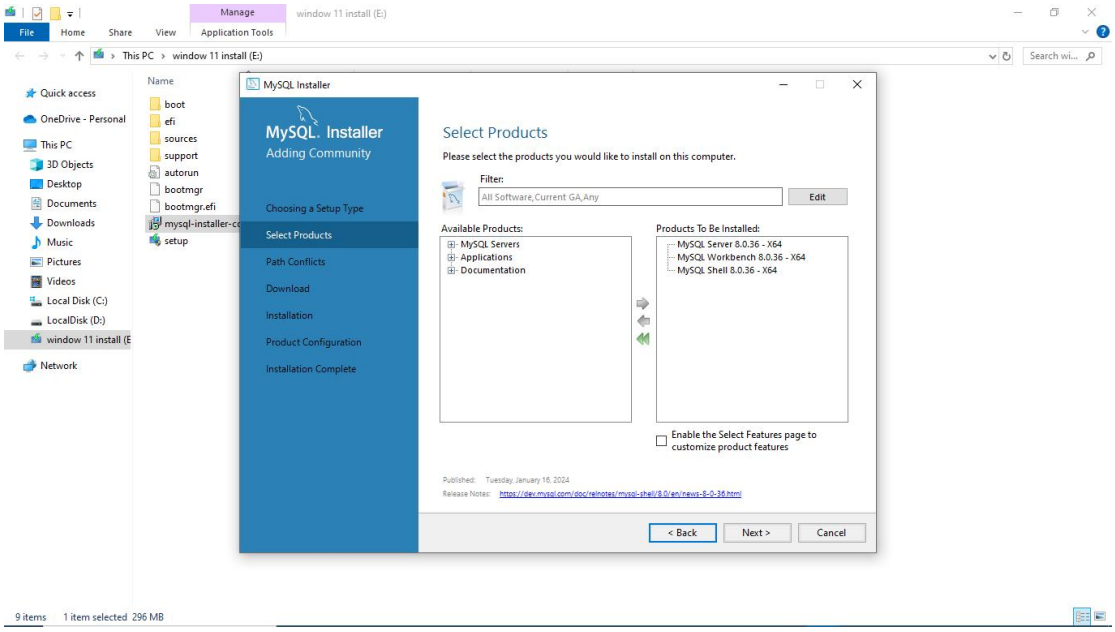


Step 2:



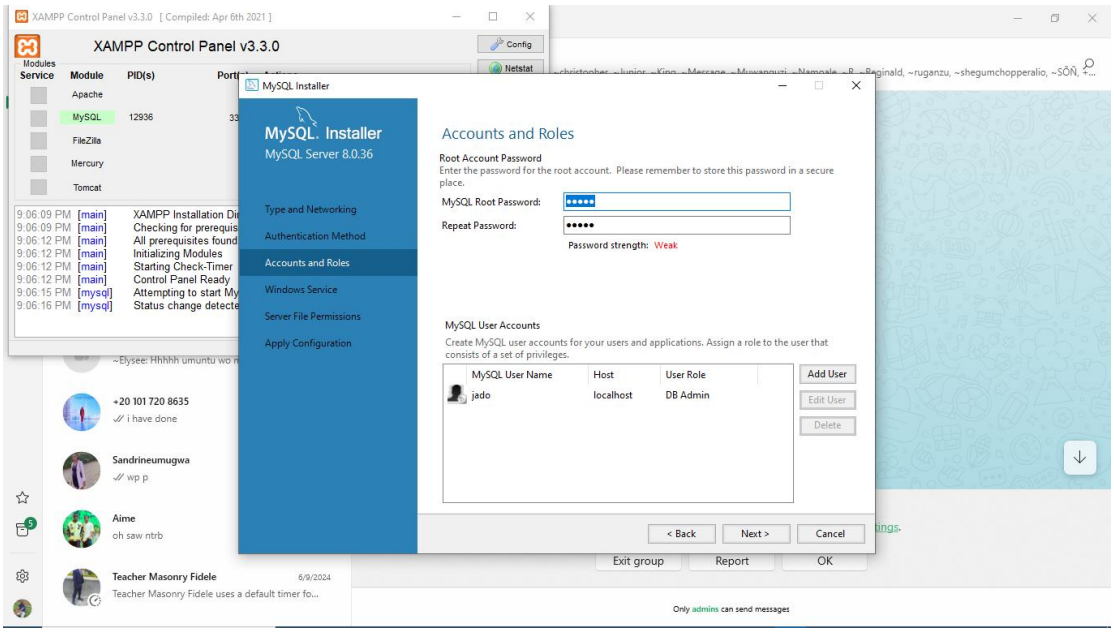
Step3

On my case I choose to install mysql customarily, here I have chosen 3 products to install



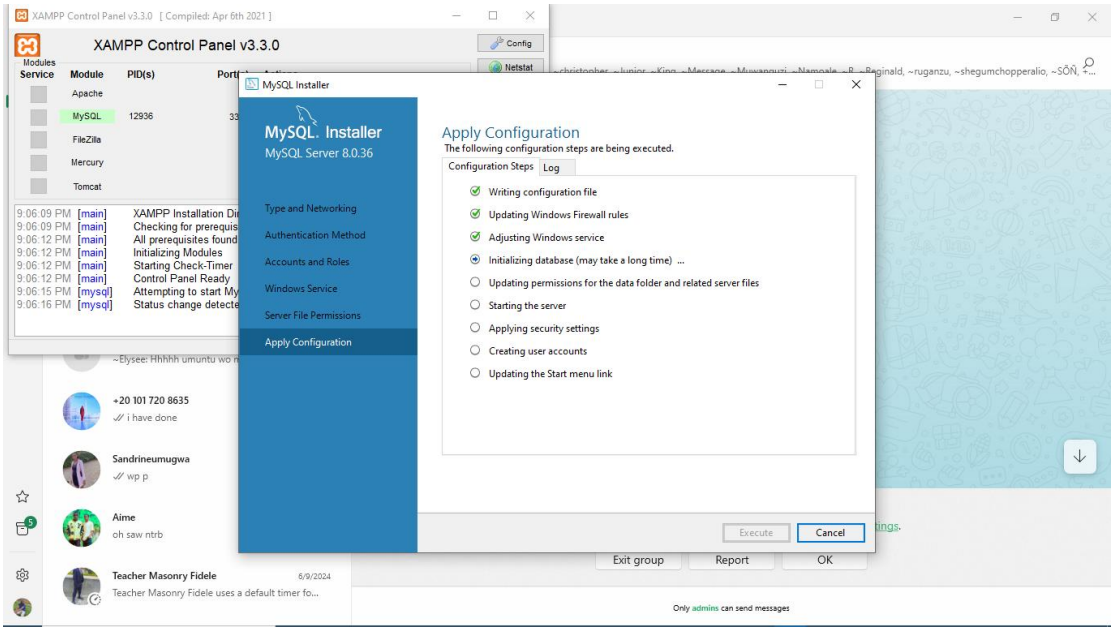
Step 4

After being installed set the default user (root) password and add new users if you want, in my case I add new user Jado

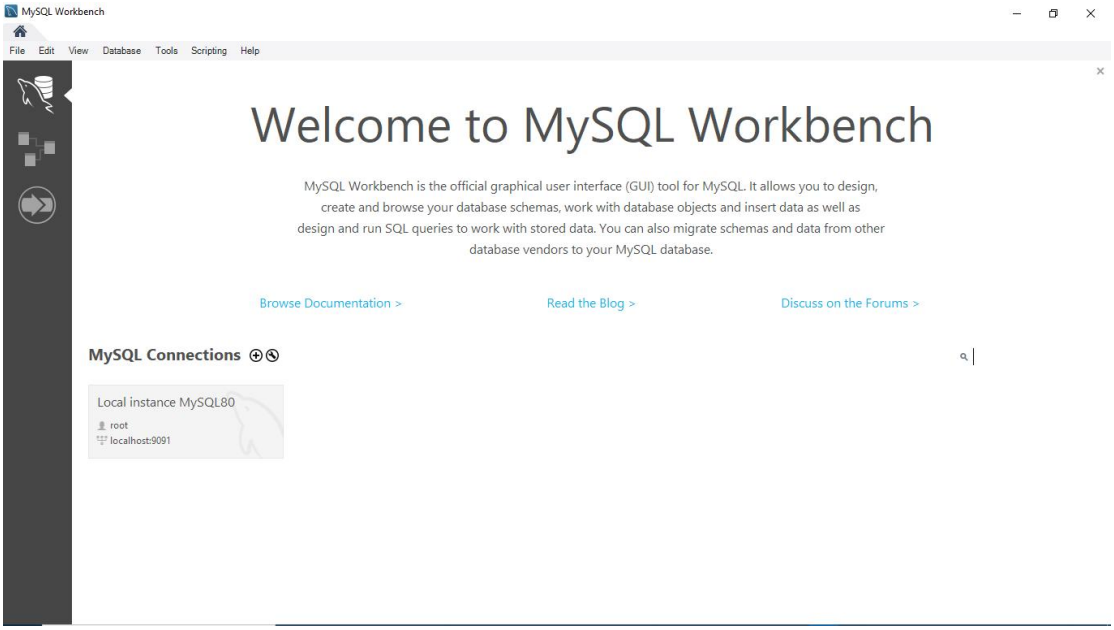


Step 5:

Just here mysql is ready to be installed well,

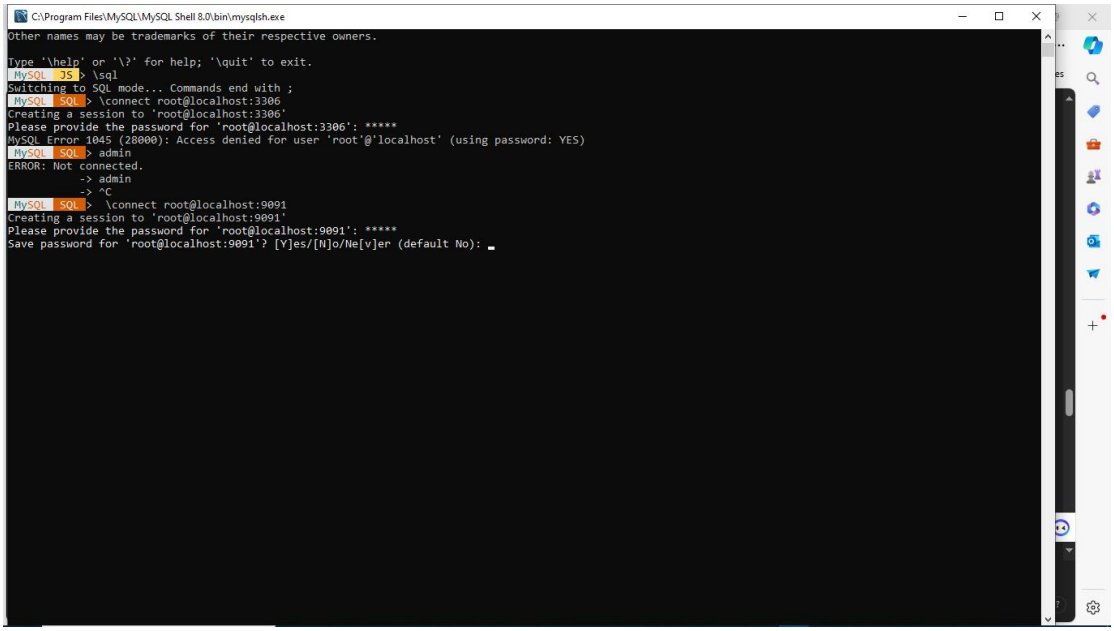


Step 6:then after mysql workbench can be lounched

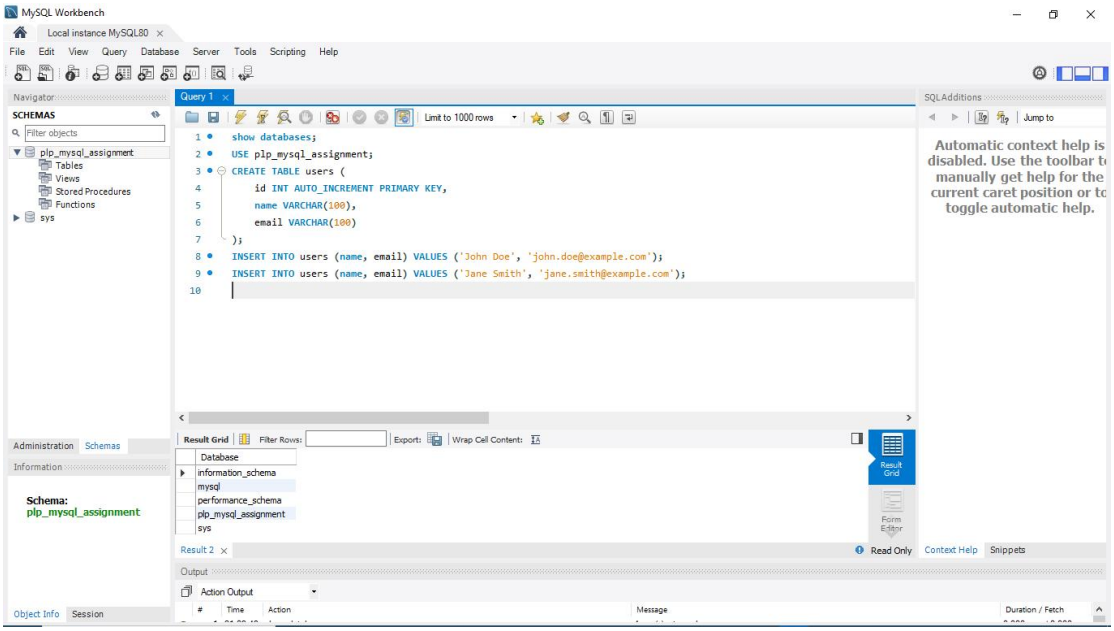


The remaining steps is how we can deal with sql commands througth mysql itself or mysql shell

Step 7: mysql shell

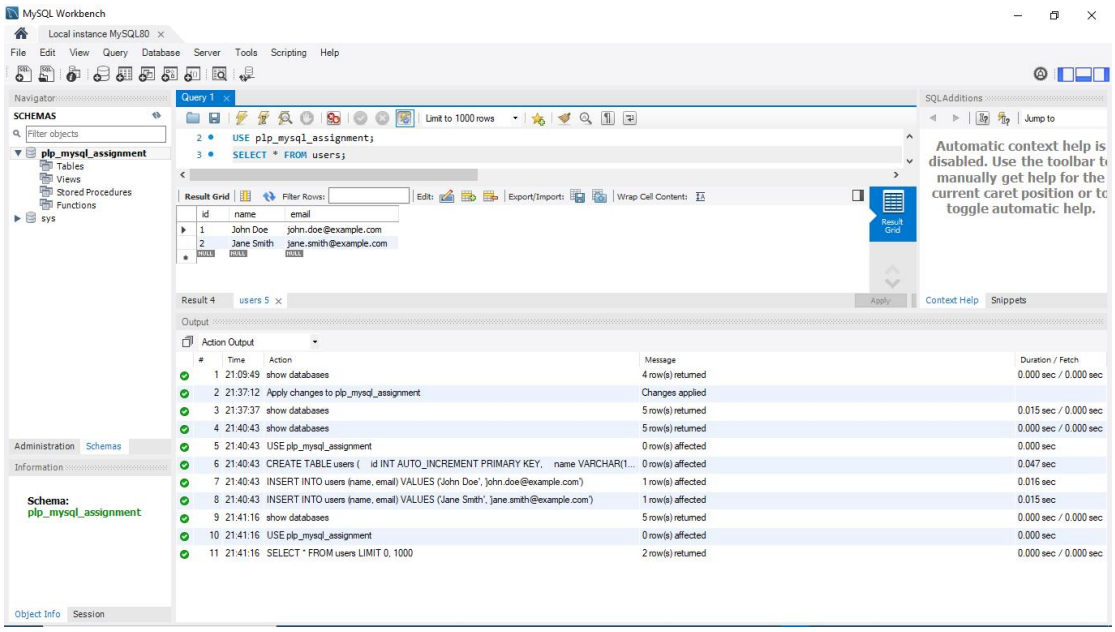


Step 8: mysql workbench running sql



Step 9: mysql workbench running sql

Step 10: after excuting the commands in step 9



Q7) installing docker or other vertual machine or docker

Step 1

Installing

pip install virtualenv

Step 2

 # Using venv
python -m venv env

 # Or using virtualenv
virtualenv env
Step 3
.\env\Scripts\activate
Step4
deactivate

Docker installation
Step one
docker --version

 # Use an official Python runtime as a parent image
FROM python:3.9-slim

 # 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"]

Step 2

docker build -t my-python-app .

Step 3
docker run -p 4000:80 my-python-app
Step 4
docker ps -a
Step 5
docker rm <container_id>
Step 6
docker ps
Step 7
docker ps -a
Thank you

Q9) From all above the following is the summary of what I have done:

Comprehensive Developer Environment Setup

This document outlines the comprehensive steps taken to set up a developer environment on Windows 11, including configurations, customizations, and troubleshooting steps encountered during the process. The setup includes installing Windows 11, Visual Studio Code (VS Code), MySQL, GitHub, Python, and virtual machines using Docker.

Table of Contents

Windows 11 Installation

1.

Download Windows 11:

2.

1. Visit the official Microsoft website to download the Windows 11 installation media.

3.

Create Installation Media:

4.

1. Use the Windows Media Creation Tool to create a bootable USB drive.

5.

Install Windows 11:

6.

1. Boot from the USB drive and follow the on-screen instructions to install Windows 11.

VS Code Installation

1.

Download VS Code:

2.

1. Visit the official VS Code website.

3.

Install VS Code:

4.

1. Run the installer and follow the on-screen instructions.

5.

Setup VS Code:

6.

1. Open VS Code and install essential extensions such as Python, Docker, and GitHub.

MySQL Installation

1.

Download MySQL:

2.

1. Visit the official MySQL website to download the MySQL installer.

3.

Install MySQL:

4.

1. Run the installer and follow the setup wizard to install MySQL Server and MySQL Workbench.

5.

Configure MySQL:

6.

1. Set up the root password and configure MySQL Server as per your requirements.

GitHub Setup

1.

Install Git:

2.

1. Visit the official Git website to download and install Git.

Set Up Git:

```
config --global user.name "Your Name"

        git config --global user.email
        "youremail@example.com"
```

3.

Clone a Repository:

4.

```
git clone https://github.com/your-username/sample-
python-docker-project.git
```

Python Installation

1.

Download Python:

2.

1. Visit the official Python website to download the latest version of Python.

3.

Install Python:

4.

1. Run the installer and make sure to check the box that says "Add Python to PATH."

5.

Verify Installation:

6.

Python Virtual

Environment Setup

1.

Install `virtualenv`:

2.

Although Python 3.3+ comes with `venv`, you can install `virtualenv` for more

```
pip install virtualenv
```

3.

Create a Virtual Environment:

4.

bash

Copy code

```
python -m venv env# Or using virtualenv
virtualenv env
```

5.

Activate the Virtual Environment:

Docker Installation

1.

Download Docker:

2.

1. Visit the official Docker website and download Docker Desktop for Windows

```
# Use an official Python runtime as a
parent image
FROM python:3.9-slim

# 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"]
```

Sample Project on GitHub

I have created a sample project and initialized it with Git, including necessary configuration files. You can find the repository here.

Repository Contents:

- `Dockerfile`
- `.dockerignore`
- `requirements.txt`
- `.gitignore`
- `app.py`
- `README.md`

Sample `.gitignore`

```
gitignore
Copy code
```

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg

# Installer logs
pip-log.txt
pip-delete-this-directory.txt

# Unit test / coverage reports
htmlcov/
.tox/
.nox/
.coverage
.coverage.*
```

```
.cache
nosetests.xml
coverage.xml
*.cover
.hypothesis/
.pytest_cache/
cover/

# Translations
*.mo
*.pot

# Django stuff:
*.log
local_settings.py
db.sqlite3

# Flask stuff:
instance/
.webassets-cache

# Sphinx documentation
docs/_build/
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/
__pypackages__/
```

Sample **requirements.txt**

```
txt
Copy code
flask
requests
```

Sample `app.py`

```
python
Copy code
from flask import Flask

app = Flask(__name__)
@app.route('/')def hello():
    return "Hello, World!"
if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

Reflection on Challenges and Solutions

Challenges Faced

1.

Issue with Python Installation:

2.

- **Challenge:** Ensuring the correct version of Python is installed and added to the PATH.
- **Solution:** During the Python installation, make sure to check the box that says "Add Python to PATH."

3.

Virtual Environment Activation:

4.

- **Challenge:** Activation script not found in the specified path.
- **Solution:** Ensure that the virtual environment is created successfully and navigate to the correct directory before activation.

5.

Docker Daemon Not Running:

6.

- **Challenge:** Docker commands were failing because the Docker daemon was not running.
- **Solution:** Start the Docker daemon manually using Docker Desktop on Windows/macOS or using `systemctl` on Linux.

7.

Port Conflicts with Docker:

8.

- **Challenge:** The specified port was already in use.
- **Solution:** Change the host port mapping in the `docker run` command.