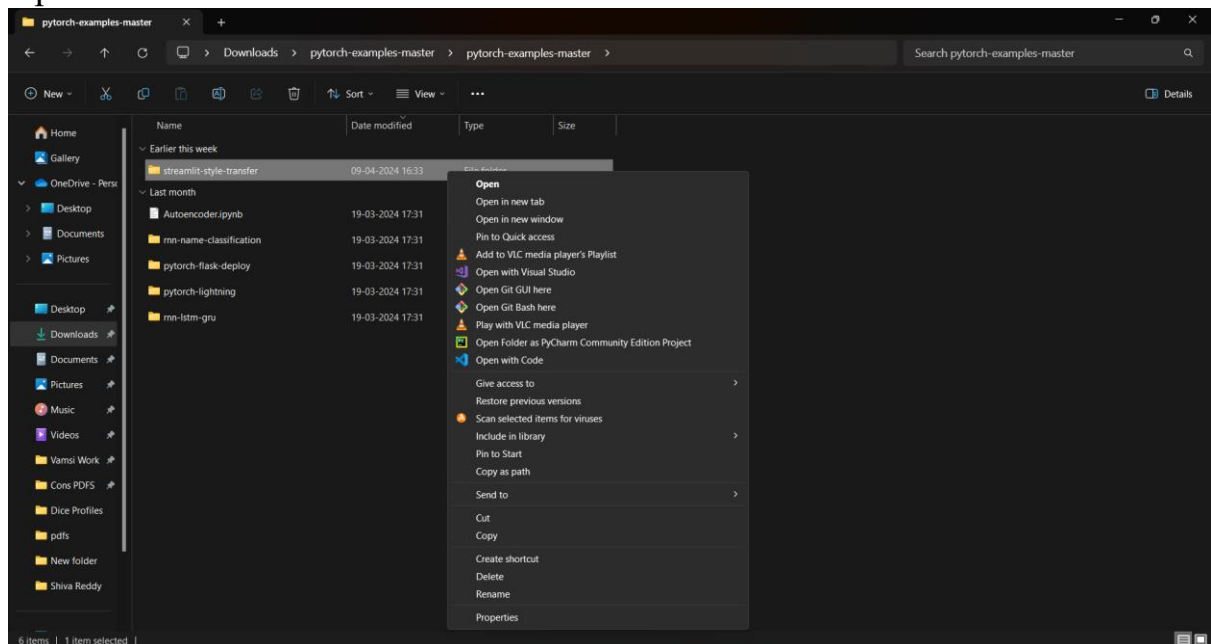
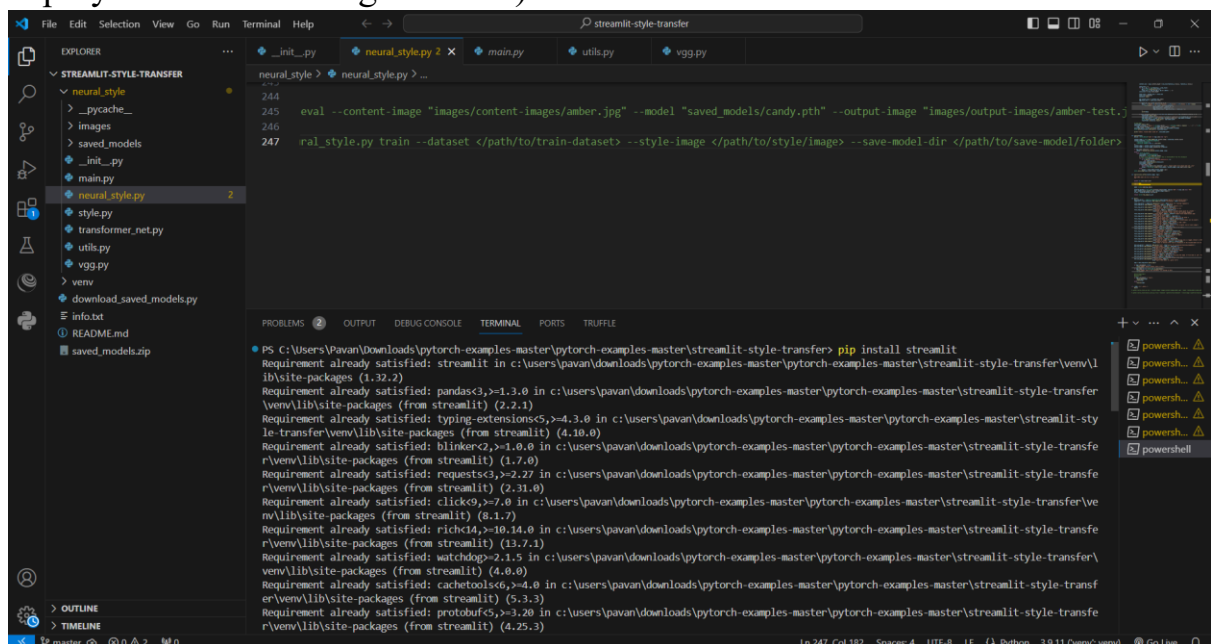


STEP BY STEP IMPLEMENTATION

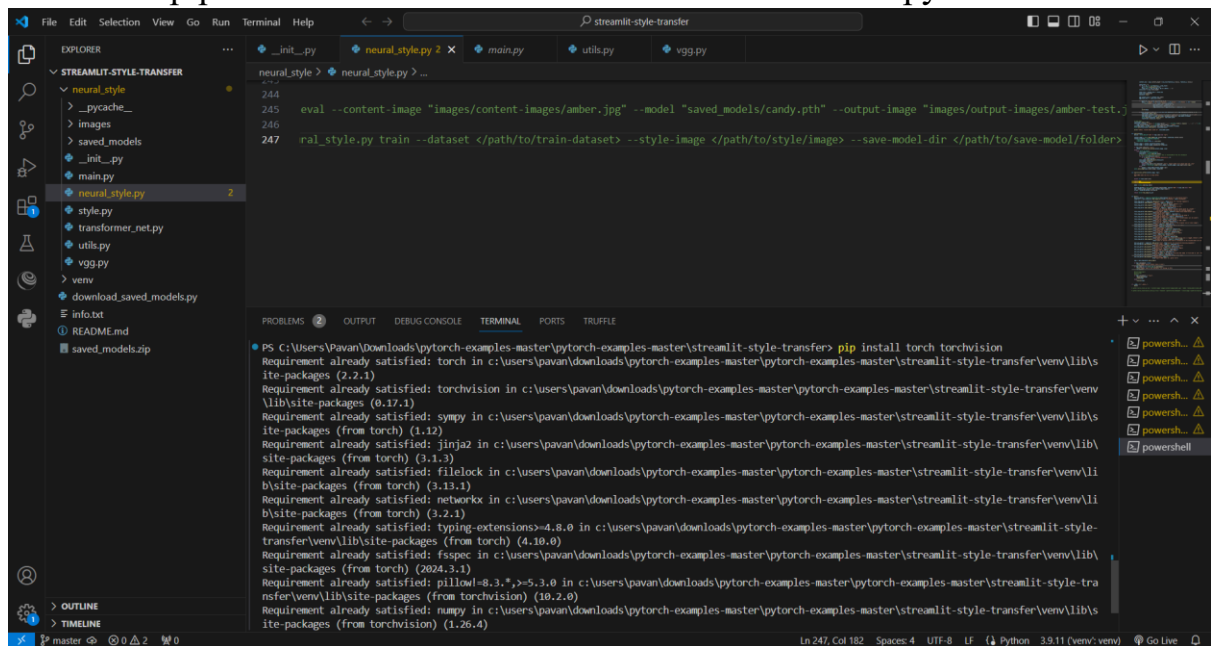
1. Clone this repository : : https://github.com/PavanM07/Neural_Style_Transfer.git using git bash.
2. Open the cloned folder with Visual Studio Code.



3. Open new terminal and run “pip install streamlit” to get streamlit(used to deploy machine learning websites) resources.



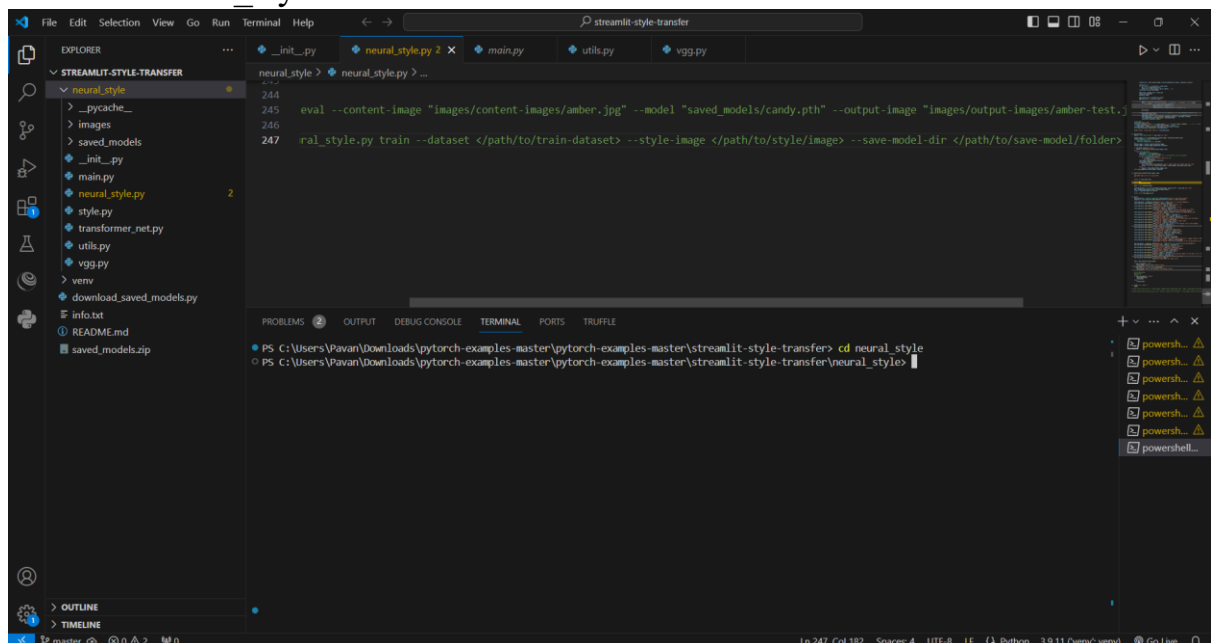
4. Then run “pip install torch torchvision” command to install pyTorch.



The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the project structure for 'STREAMLIT-STYLE-TRANSFER'. The main editor shows the 'neural_style.py' file with lines 244 to 247. The output panel at the bottom displays the results of running 'pip install torch torchvision' in a virtual environment. The output shows that the requirements are already satisfied for torch (2.2.1), torchvision (0.17.1), sympy (1.12), Jinja2 (3.1.3), filelock (3.13.1), networkx (3.2.1), typing-extensions (4.8.0), fsspec (4.10.0), pillow (8.3.0), and numpy (1.26.4).

```
PS C:\Users\Pavan\Downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer> pip install torch torchvision
Requirement already satisfied: torch in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (2.2.1)
Requirement already satisfied: torchvision in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (0.17.1)
Requirement already satisfied: sympy in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torch) (1.12)
Requirement already satisfied: Jinja2 in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torch) (3.1.3)
Requirement already satisfied: filelock in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torch) (3.13.1)
Requirement already satisfied: networkx in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torch) (3.2.1)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torch) (4.10.0)
Requirement already satisfied: fsspec in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torch) (4.10.0)
Requirement already satisfied: pillow<=8.3.*,>=5.3.0 in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torchvision) (10.2.0)
Requirement already satisfied: numpy in c:\users\pavan\downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\venv\lib\site-packages (from torchvision) (1.26.4)
```

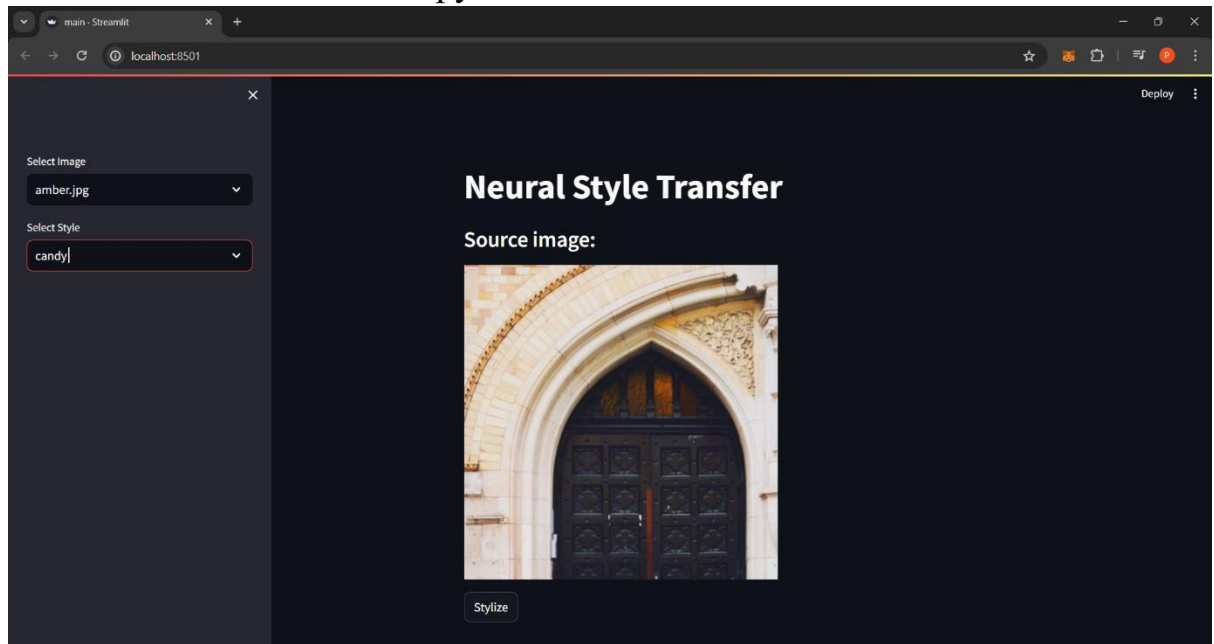
5. Run “cd neural_style” command.



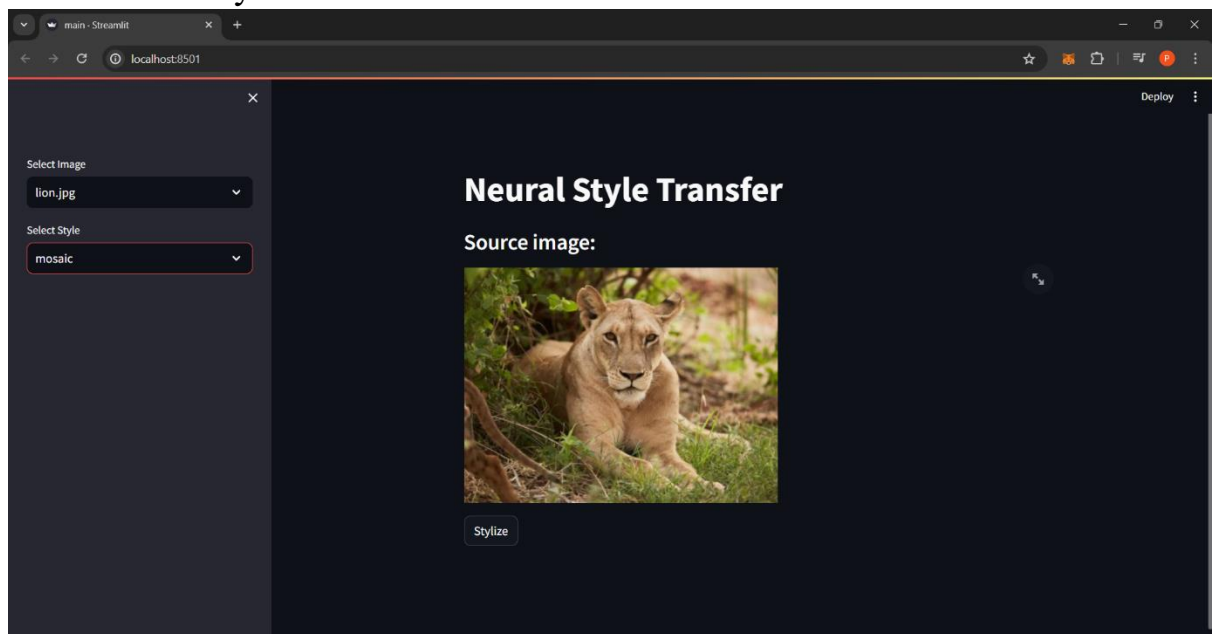
The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the project structure for 'STREAMLIT-STYLE-TRANSFER'. The main editor shows the 'neural_style.py' file with lines 244 to 247. The output panel at the bottom displays the results of running 'cd neural_style' in the terminal. The output shows the current directory being changed to 'C:\Users\Pavan\Downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\neural_style'.

```
PS C:\Users\Pavan\Downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer> cd neural_style
PS C:\Users\Pavan\Downloads\pytorch-examples-master\pytorch-examples-master\streamlit-style-transfer\neural_style>
```

6. Then run “streamlit run main.py” to launch the website in streamlit.



7. Now select the content image from ‘Select Image’ field and style model from ‘Select Style’ field.



8. Then click on ‘Stylize’ button to get the stylized image as Output image.

