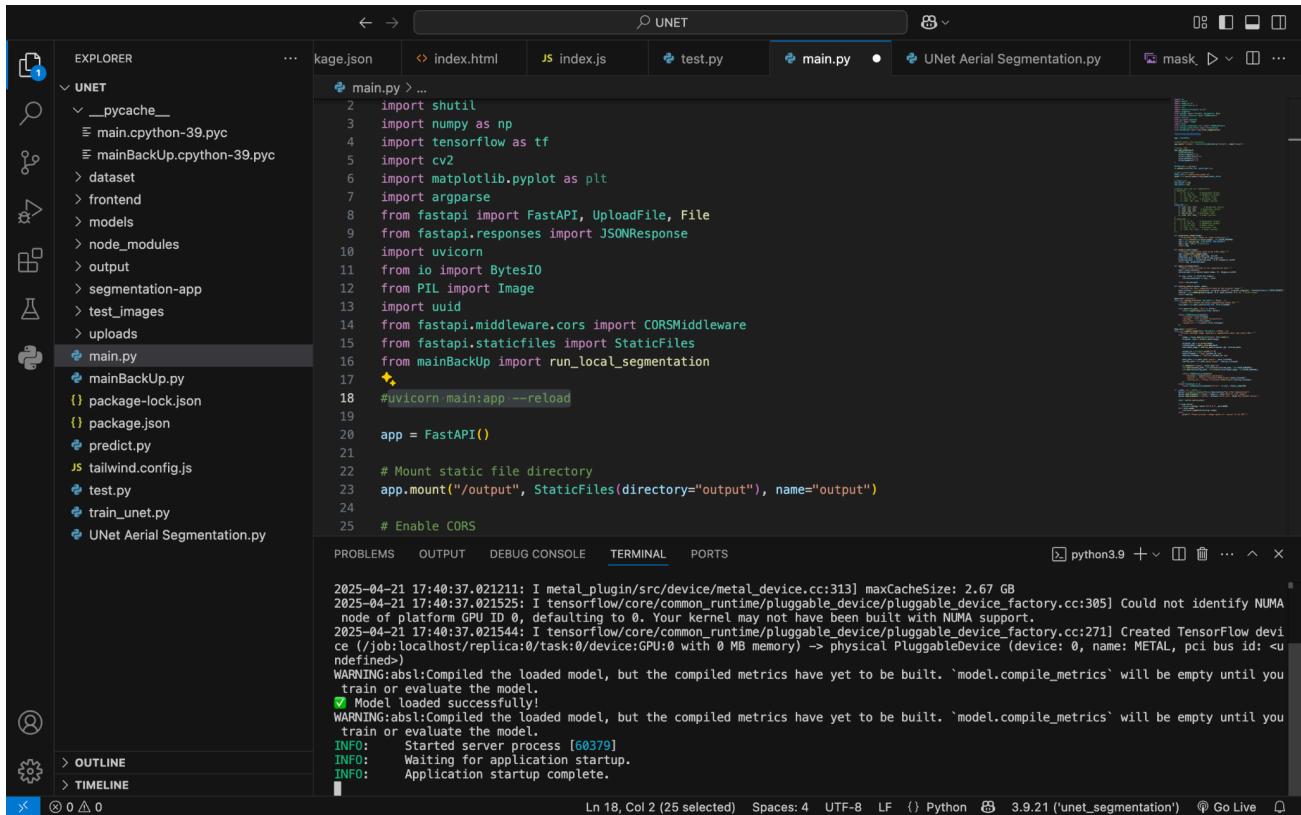


PROJECT OUTPUT :

1.BACKEND TERMINAL RUN



```
2  import shutil
3  import numpy as np
4  import tensorflow as tf
5  import cv2
6  import matplotlib.pyplot as plt
7  import argparse
8  from fastapi import FastAPI, UploadFile, File
9  from fastapi.responses import JSONResponse
10 import uvicorn
11 from io import BytesIO
12 from PIL import Image
13 import uuid
14 from fastapi.middleware.cors import CORSMiddleware
15 from fastapi.staticfiles import StaticFiles
16 from mainBackUp import run_local_segmentation
17 +
18 #uvicorn main:app --reload
19
20 app = FastAPI()
21
22 # Mount static file directory
23 app.mount("/output", StaticFiles(directory="output"), name="output")
24
25 # Enable CORS
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
2025-04-21 17:40:37.021211: I metal_plugin/src/device/metal_device.cc:313] maxCacheSize: 2.67 GB
2025-04-21 17:40:37.021525: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:305] Could not identify NUMA node of platform GPU ID 0, defaulting to 0. Your kernel may not have been built with NUMA support.
2025-04-21 17:40:37.021544: I tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:271] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 0 MB memory) -> physical PluggableDevice (device: 0, name: METAL, pci bus id: <undefined>)
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
✓ Model loaded successfully!
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
INFO:     Started server process [60379]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

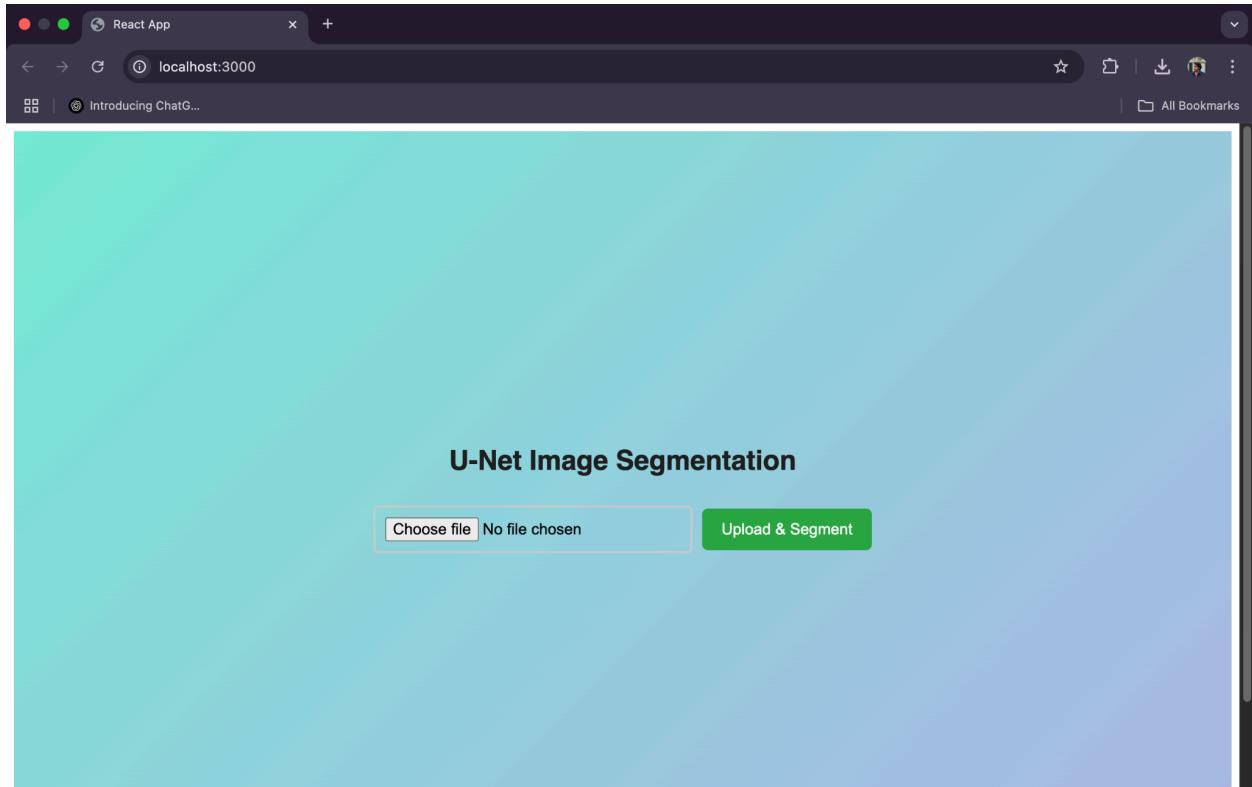
Ln 18, Col 2 (25 selected) Spaces: 4 UTF-8 LF {} Python ⚡ 3.9.21 ('unet_segmentation') ⓘ Go Live ⌂

2.FRONT END RUN

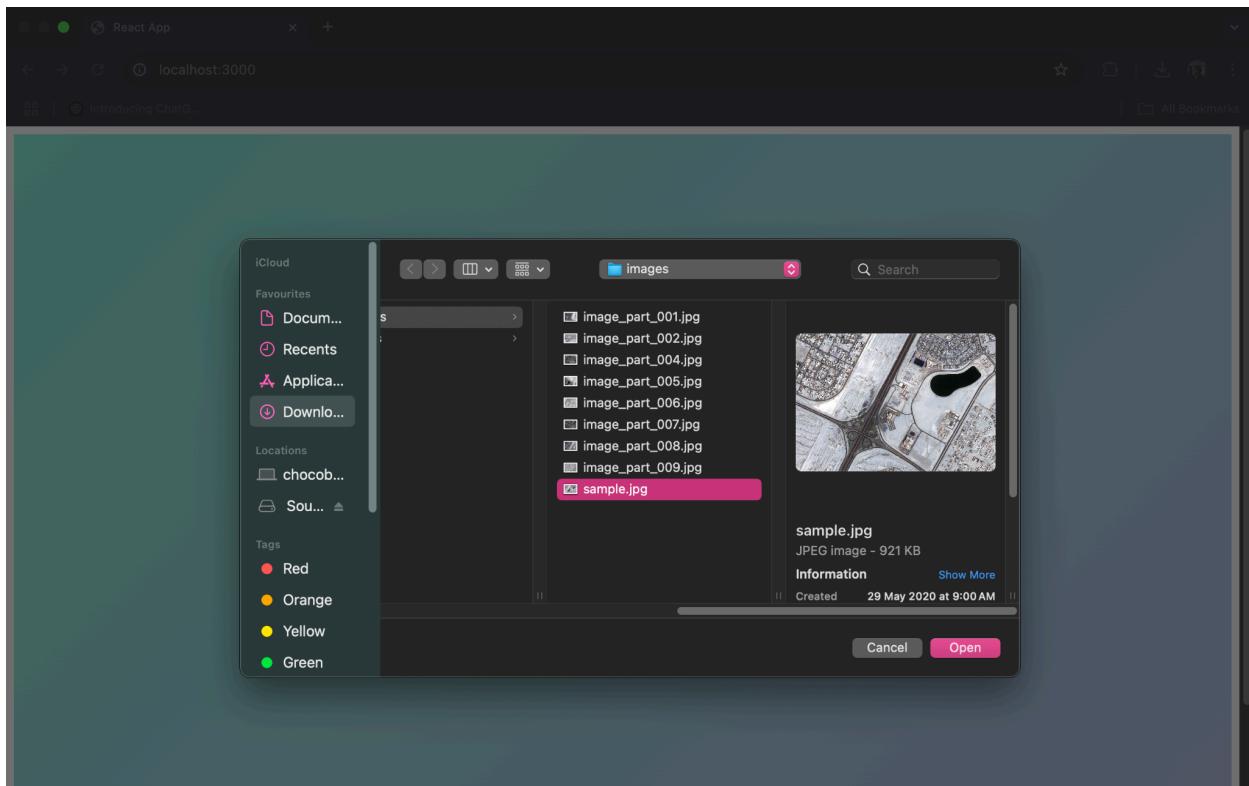
The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure under the `UNET` folder, including `dataset`, `frontend`, `models`, `node_modules`, `output`, `segmentation-app`, `test_images`, `uploads`, `main.py`, `mainBackUp.py`, `package-lock.json`, `package.json`, `predict.py`, `tailwind.config.js`, `test.py`, `train_unet.py`, and `UNet Aerial Segmentation.py`.
- EDITOR:** The `main.py` file is open, showing Python code for a FastAPI application. The code includes imports for `shutil`, `numpy`, `tensorflow`, `cv2`, `matplotlib.pyplot`, `argparse`, `uvicorn`, `BytesIO`, `Image`, `uuid`, `CORSMiddleware`, `StaticFiles`, and `run_local_segmentation`. It uses `#uvicorn main:app --reload` to run the app and mounts the `output` directory as static files.
- TERMINAL:** The terminal shows the output of the command `uvicorn main:app --reload`. It includes messages about compilation, CORS, and a successful webpack build.
- OUTPUT:** Shows logs for `python3.9` and `node front...`.
- STATUS BAR:** Displays the current line (Ln 18), column (Col 2), number of selected lines (25 selected), and other system information.

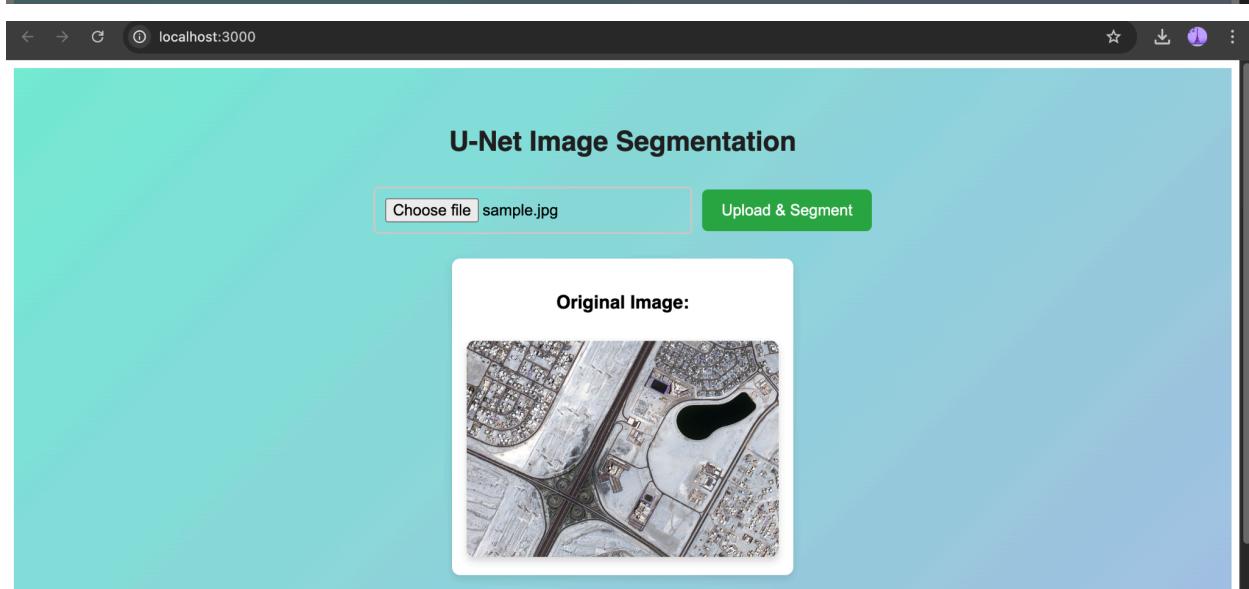
3.USER PAGE



4.INPUT FROM LOCAL

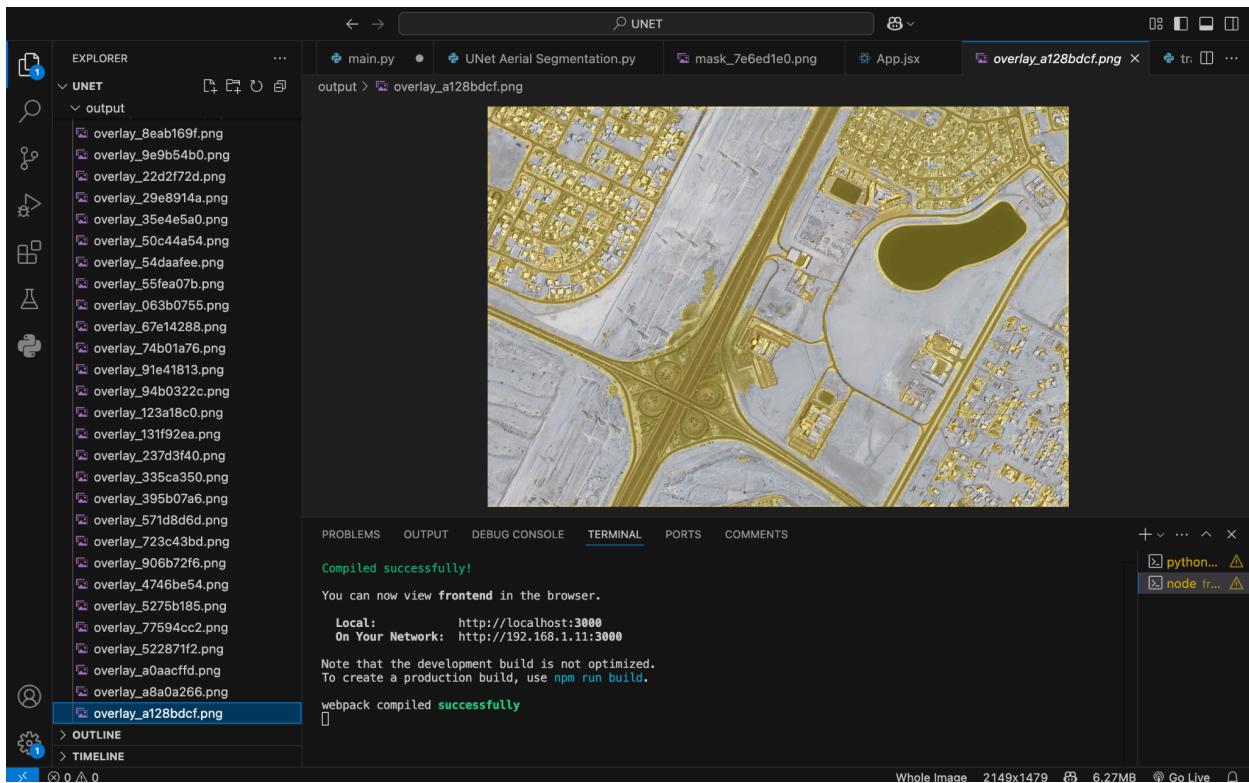


A screenshot of a file selection dialog from a web browser. The dialog shows a file tree under the 'images' folder, with files named 'image_part_001.jpg' through 'image_part_009.jpg' and 'sample.jpg'. The 'sample.jpg' file is highlighted with a pink rectangle. To the right of the file list is a preview image of a satellite map showing a road intersection and some water bodies. Below the preview is a 'sample.jpg' file info card. At the bottom of the dialog are 'Cancel' and 'Open' buttons.



A screenshot of a web application titled 'U-Net Image Segmentation'. The interface includes a file input field labeled 'Choose file sample.jpg' with a red border, an 'Upload & Segment' button, and a preview area titled 'Original Image:' containing a satellite map image. The background of the app has a teal gradient.

5.OUTPUT SEGMENTED IMAGE STORED IN LOCAL DATABASE



5.OUTPUT SEGMENTED IMAGE IN WEB PAGE

