\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Project structure:

.env/

.gitignore #root

requirements.txt #root

crm/

├── crm/

├── static

├── css

├── js

├── settings.py

├── urls.py

├── wsgi.py

├── node\_modules/

├── bin/

├── naniod/

├── picocolors/

├── postcss/

├── source-map.js/

├── tailwindcss/

├── theme/

├── static/css/dist

├── styles.css

├── static\_src/

├── node\_modules/

├── src/

├── styles.css

├── tailwind.config.js

├── package-lock.json

├── package.json

├── postcss.config.js

├── templates/

├── base.html

├── webapp/

├── migrations/

├── templates/webapp

├── base.html

├── index.html

├── node\_modules/

├── manage.py

├── package-lock.json

├── package.json

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Github repo to clone:

|  |
| --- |
| https://github.com/Sicelo-Mzingisi-Mduna/Template\_DjangoProject\_Tailwind\_configured.git |

Run the following commands using bash in the folder you want to clone the repo to

Step 1: **Clone the original template repo locally**:

|  |
| --- |
| git clone https://github.com/Sicelo-Mzingisi-Mduna/Template\_DjangoProject\_Tailwind\_configured.git |

Step2: **Copy it to a new directory**: (this automatically creates a new folder)

|  |
| --- |
| cp -r Template\_DjangoProject\_Tailwind\_configured CODE |

Template\_DjangoProject\_Tailwind\_configured = clones repo

CODE = new folder that you want to create that will have your repo code

|  |
| --- |
| rm -rf Template\_DjangoProject\_Tailwind\_configured |

|  |
| --- |
| cd CODE |

Step 3: **Remove the Git history** (this breaks the link with the template project):

|  |
| --- |
| rm -rf .git |

Step 4: **Initialize a new Git repo**:

|  |
| --- |
| git init |

|  |
| --- |
| git add . |

|  |
| --- |
| git commit -m "Initial commit based on django\_template" |

Step 5: Register this new github repo on Github desktop

------------------------------------------------------------------------------------------------

Step 6: create a virtual env and activate it using the CMD

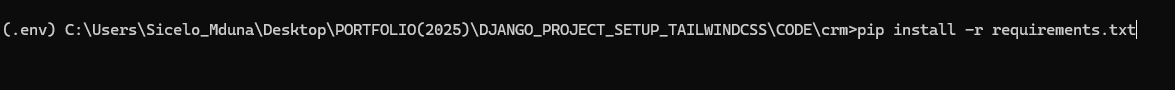
1. Create a virtual env + activate it on the CMD and visual studio

|  |
| --- |
| python -m venv .env |

|  |
| --- |
| .env\Scripts\activate |

Step 7: use the same external terminal to pip install the python requirements.txt file and the json file in the theme folder:

python Django dependencies



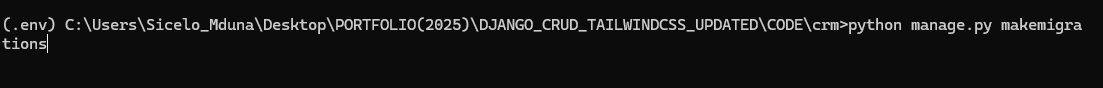
|  |
| --- |
| pip install -r requirements.txt |

Json tailwind dependencies

A black and white text

AI-generated content may be incorrect.

Step 8: make your migrations(in the root crm folder)



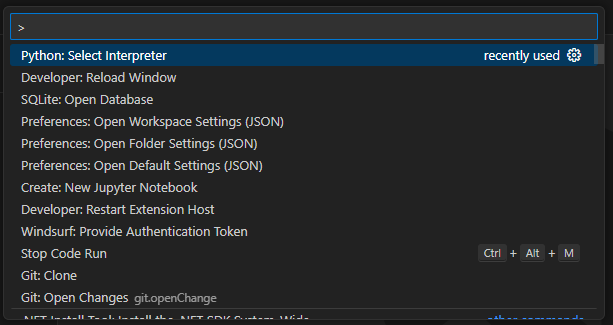
|  |
| --- |
| python manage.py makemigrations |

|  |
| --- |
| python manage.py migrate |

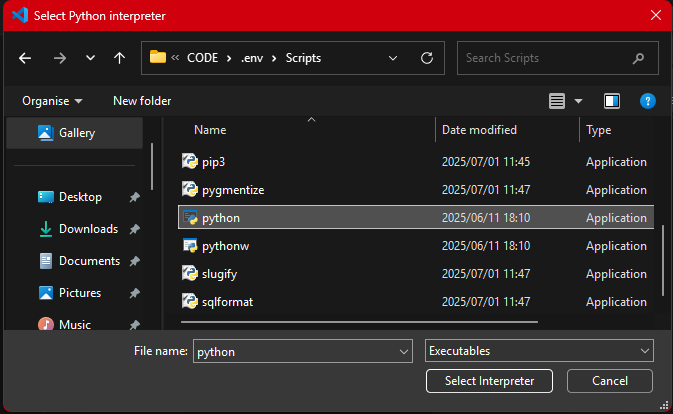
**Step 8.2**

Register the Venv in visual studio, set up the required CMD Terminals at the root and the gitbash

1. Press Cntrl + Shift + P -> Python: Select Intepreter



1. Click on “Enter Intepreter path”
2. Click on Find
3. Select the folder “.env” -> Select the “Scripts” folder ->
4. Select the Python interpreter in that folder



1. Launch the 2 CMD and 1 gitbash terminals

A screenshot of a computer program

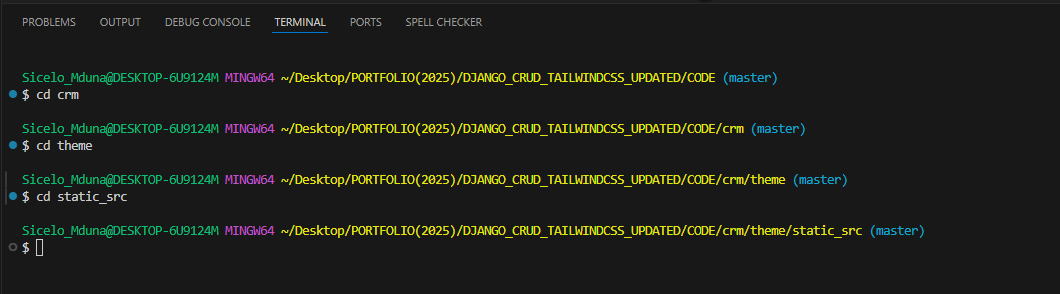
AI-generated content may be incorrect.

Result:  
A black screen with white text

AI-generated content may be incorrect.

**Step 9 (Visual studio terminals)**

Run this command in bash(cd to the “static\_src” folder)

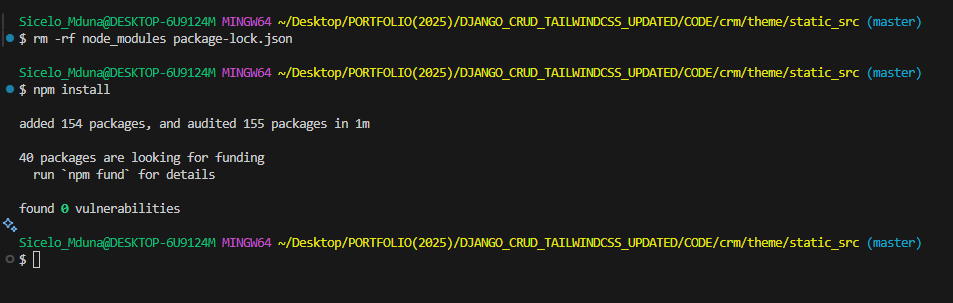


Then run these bash commands

|  |
| --- |
| rm -rf node\_modules package-lock.json |

|  |
| --- |
| npm install |

Result after running the bash commands:



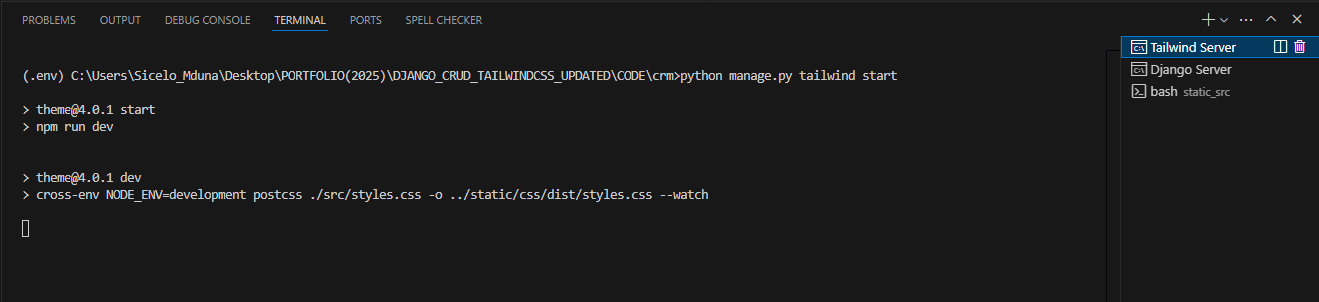
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Run this command this in CMD in the crm directory(each on a separate CMD – that’s why you created 2) – (Rename the CMD terminals)

CMD 1 = Tailwind Server (run the tailwind server)

|  |
| --- |
| python manage.py tailwind start |

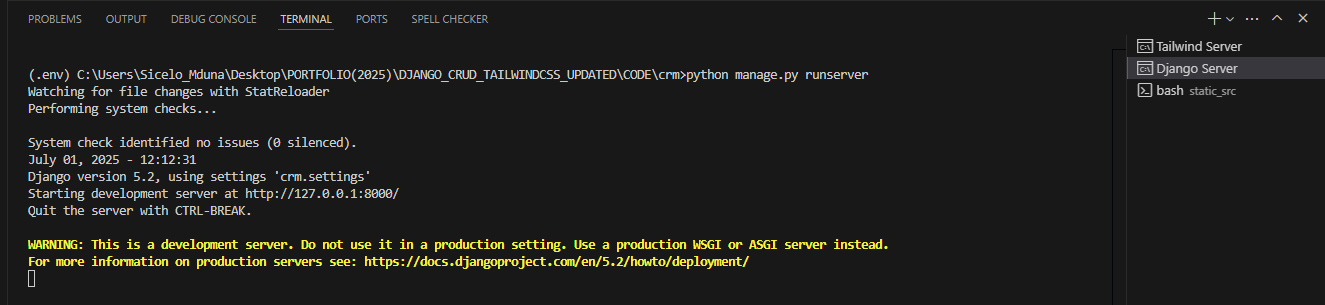
Result after this command has successfully ran:



CMD 2 =Django Server (run the Django server)

|  |
| --- |
| python manage.py runserver |

Result after running the Django Server



Final Step : copy the Django development server URL and paste into a web browser. Copy the: http://127.0.0.1:8000/ and paste this into a web browser

FINAL RESULTS (THE PROJECT RUNS – CONGRATULATIONS)

