**SETTING UP**

Create folder (mkdir backend frontend)

Cd backend

add python environment (python -m venv env - env\Scripts\Activate)

install sqlalchemy and pydantic

create main.py-database.py-crud.py-schemas.py-main.py-models.py

run in  **cmd “**uvicorn main:app –reload”

**FOR GITHUB COMMIT:**

main folder> git init

git remote add origin <url>.git

git add <folder>

git commit -m ""

git branch( to check default branch)

git branch -M main(to switch into main branch)

git push origin main( main is the default branch)

**backend deployment setup:**

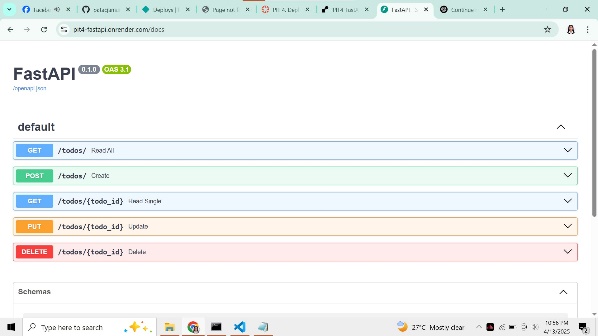
STARTCOMMAND( RENDER):

gunicorn main:app -k uvicorn.workers.UvicornWorker --bind 0.0.0.0:10000

INSTALLATIONS ADDED:

pip install sqlalchemy

pip freeze > requirements.txt – inside backend folder



**frontend deployment setup:**

**Base directory:** frontend

**Build command:** npm run build

ADD environment variables in site configurations( KEY: **VITE\_API\_URL** value: **backend site/docs**)

**Publish directory:** frontend/build

npm install react react-dom

npm install react-scripts

npm install axios

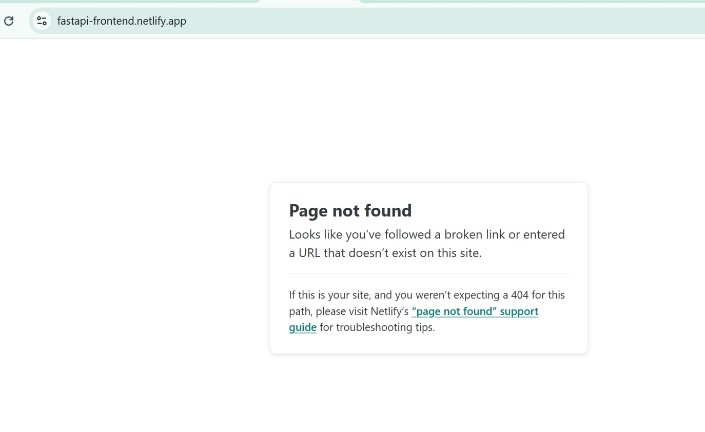
npm install react-router-dom

**DIFFERENCE BETWEEN DJANGORESTFRAMEWORK AND FASTAPI:**

Django REST Framework (DRF) and FastAPI are both robust API-building tools in Python, but they are used for different purposes. DRF is a Django-based framework that provides a full-stack, batteries-included experience with integrated authentication, admin panel, and close integration with Django's ORM—suitable for big, monolithic applications. But it's synchronous, which can cause slower performance, and its class-based views can have a steeper learning curve. Conversely, FastAPI is a high-performance, modern framework built for creating fast APIs using Python's async /await syntax and Pydantic data validation. It provides auto-generated Swagger and ReDoc documentation and is suited for asynchronous workloads as well as microservices. FastAPI provides more flexibility but does not include built-in features such as an admin panel or a well-established ecosystem like Django's. In short, DRF stands out for feature-rich web applications with high-backend functionality, while FastAPI is the best in lean, efficient, and scalable API creation.

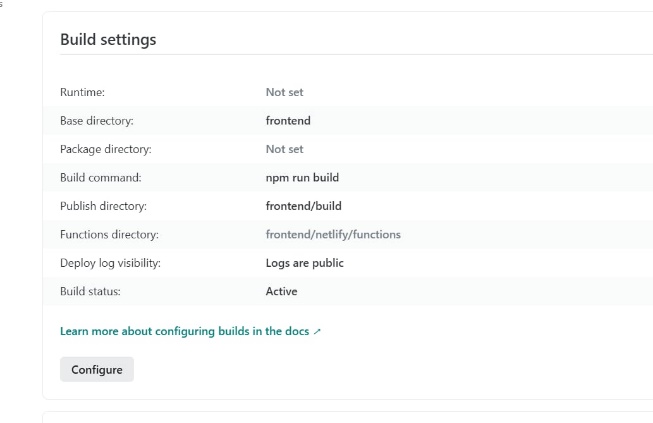
Overall, I prefer using FastAPI from the word itself fast. It takes only smaller time than Django ,especially the Django token takes a lot of time for better security and needs to have a lot of dependencies to install.

**CHALLENGES/PROBLEMS ENCOUNTERED:**

****

-I move “\_\_redirect” file from build file to inside the public of my frontend

-still get error in frontend: go to netlify - add environment variables and fixed the deployment build



I have also encountered endpoints duplication in my frontend and backend by testing it first locally.