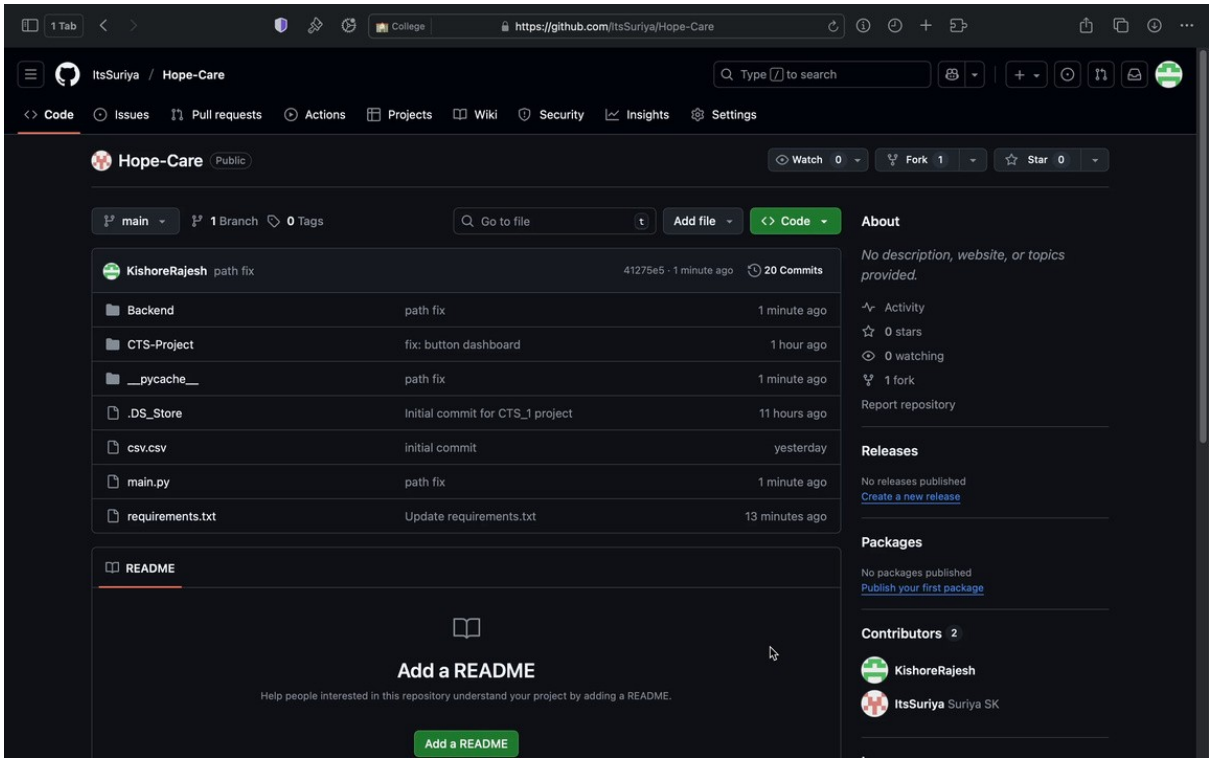
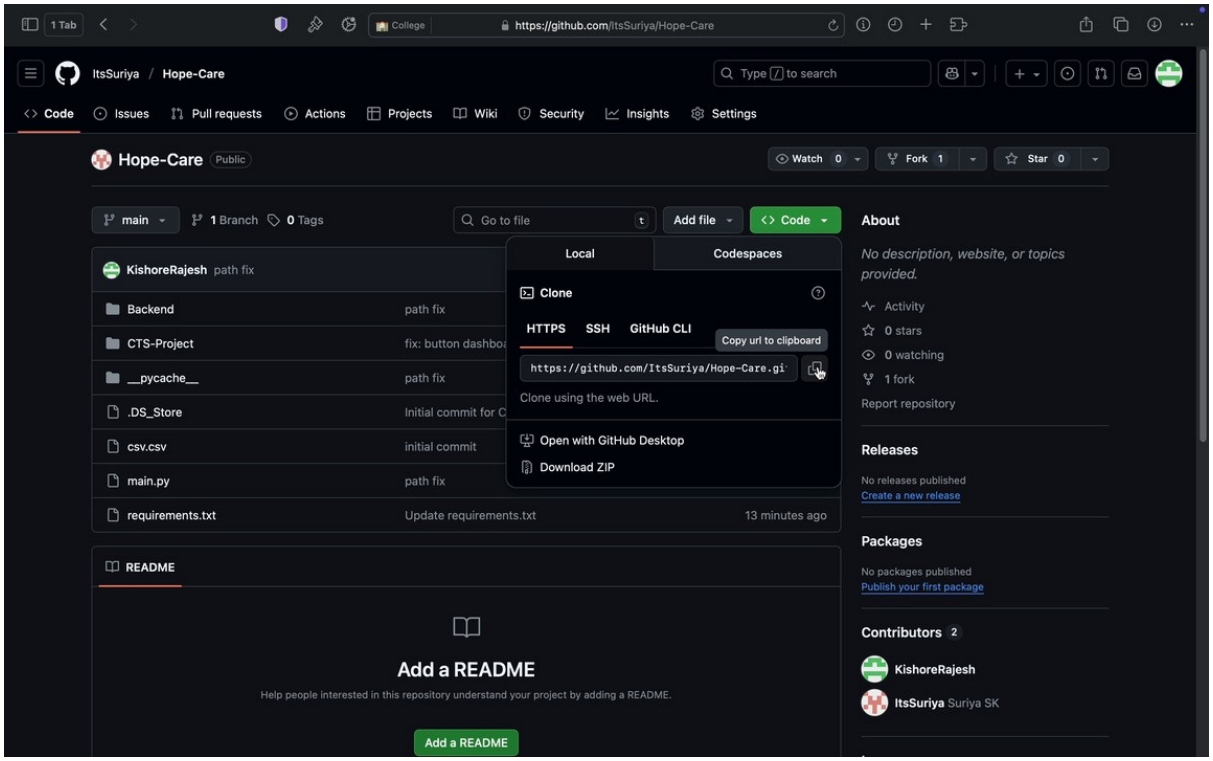
How To Run The WebApplication From Github To Local

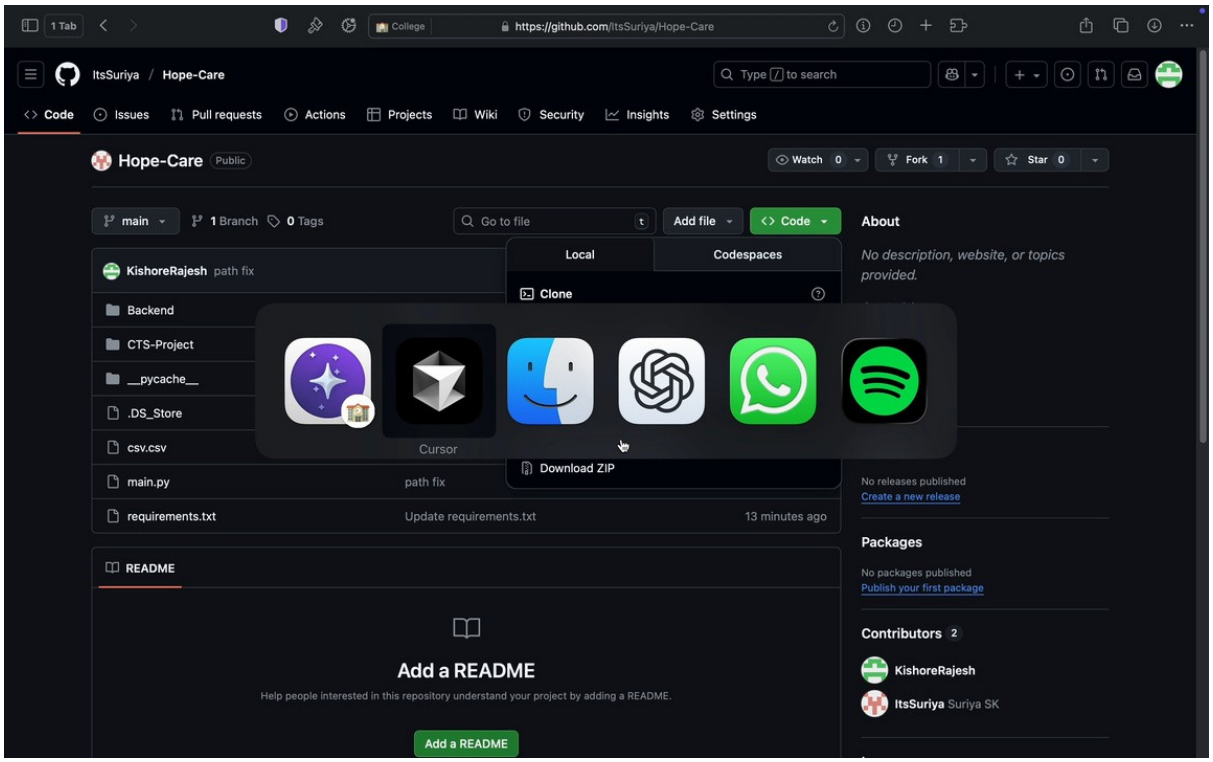
Step1: Go to provided Github Link - https://github.com/ItsSuriya/Hope-Care



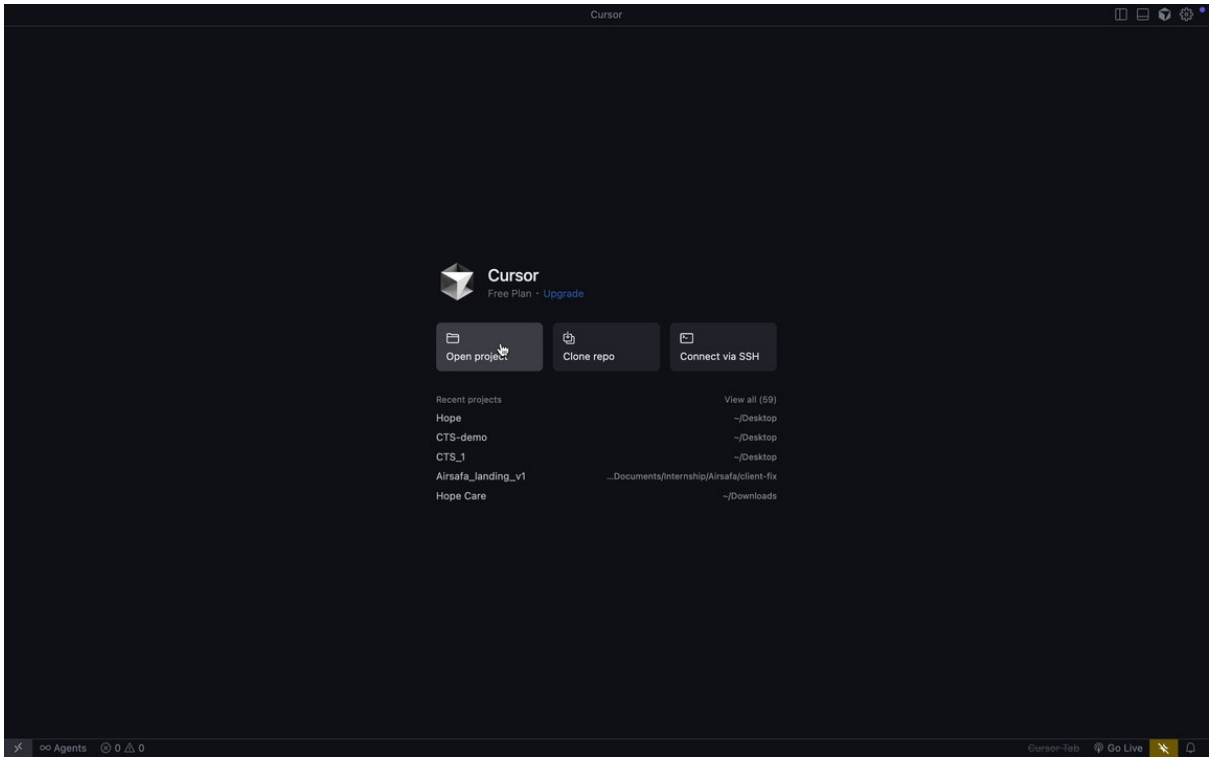
Step 2: Go to Code → copy HTTPS Link



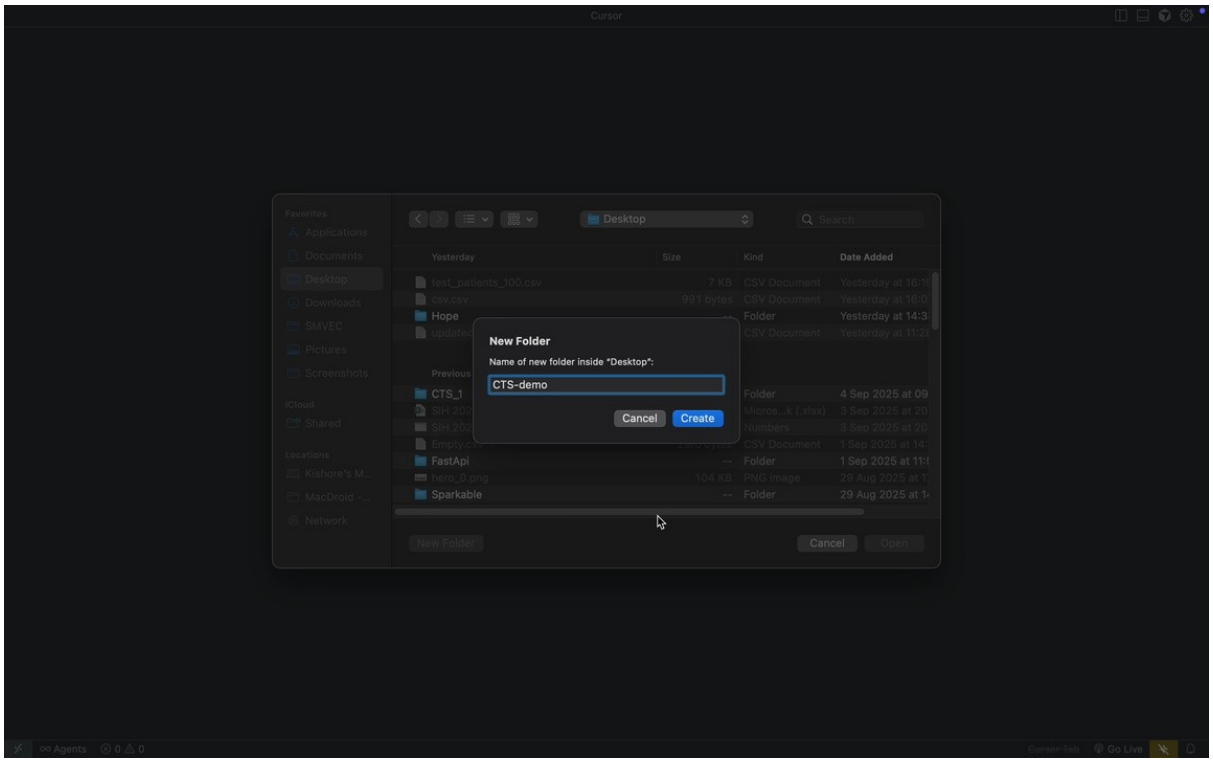
Step 3:Navigate to Cursor or VS Code as you prefer



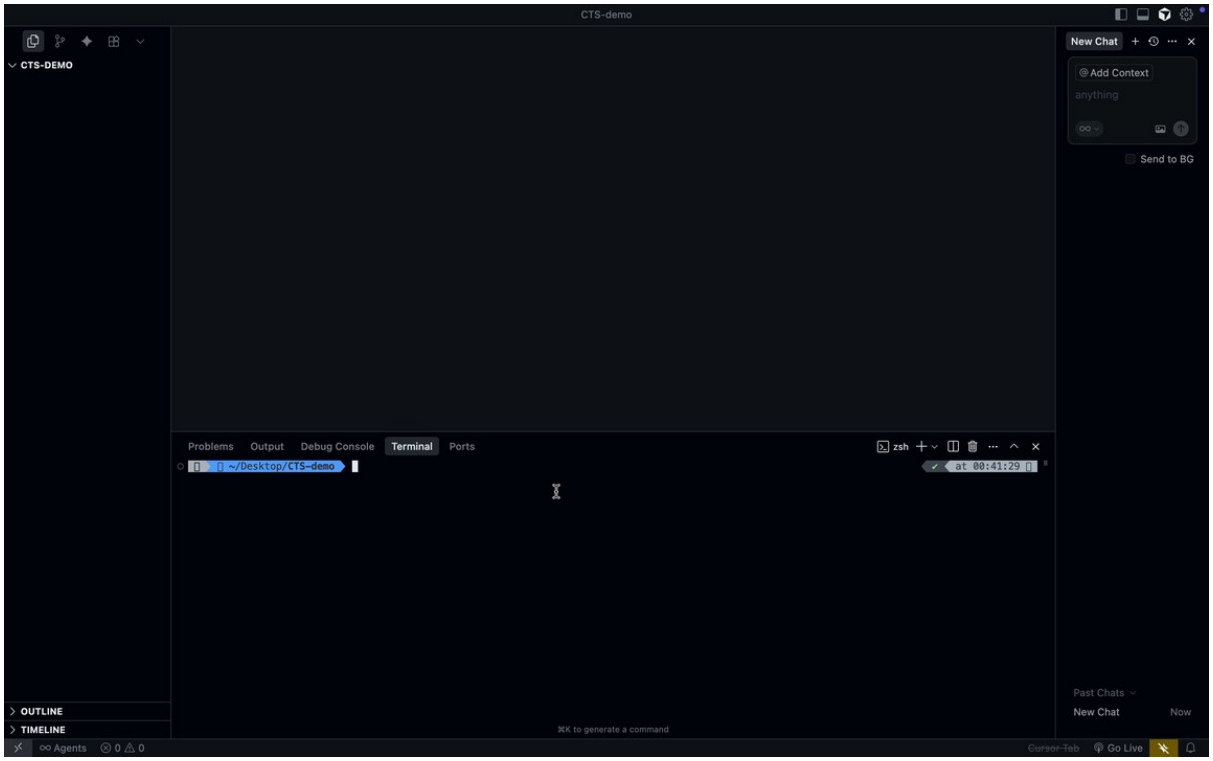
Step 4:Create a New Project



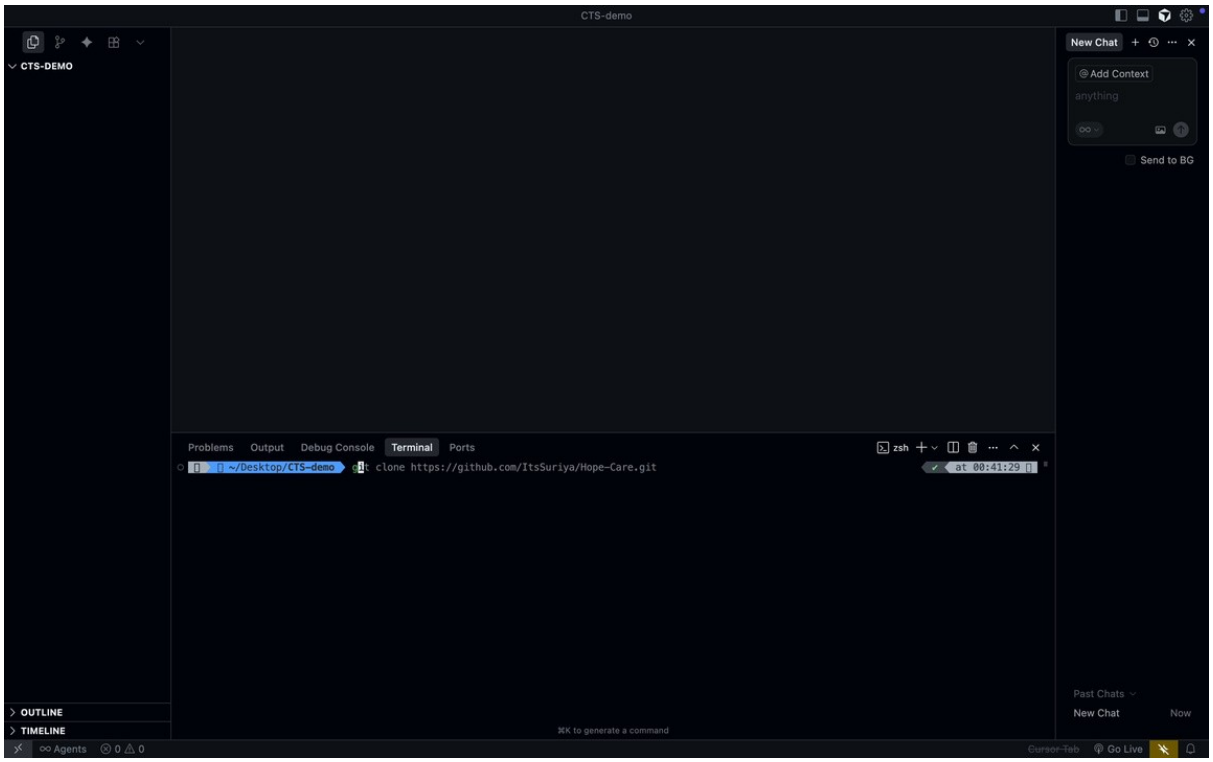
Name The Project



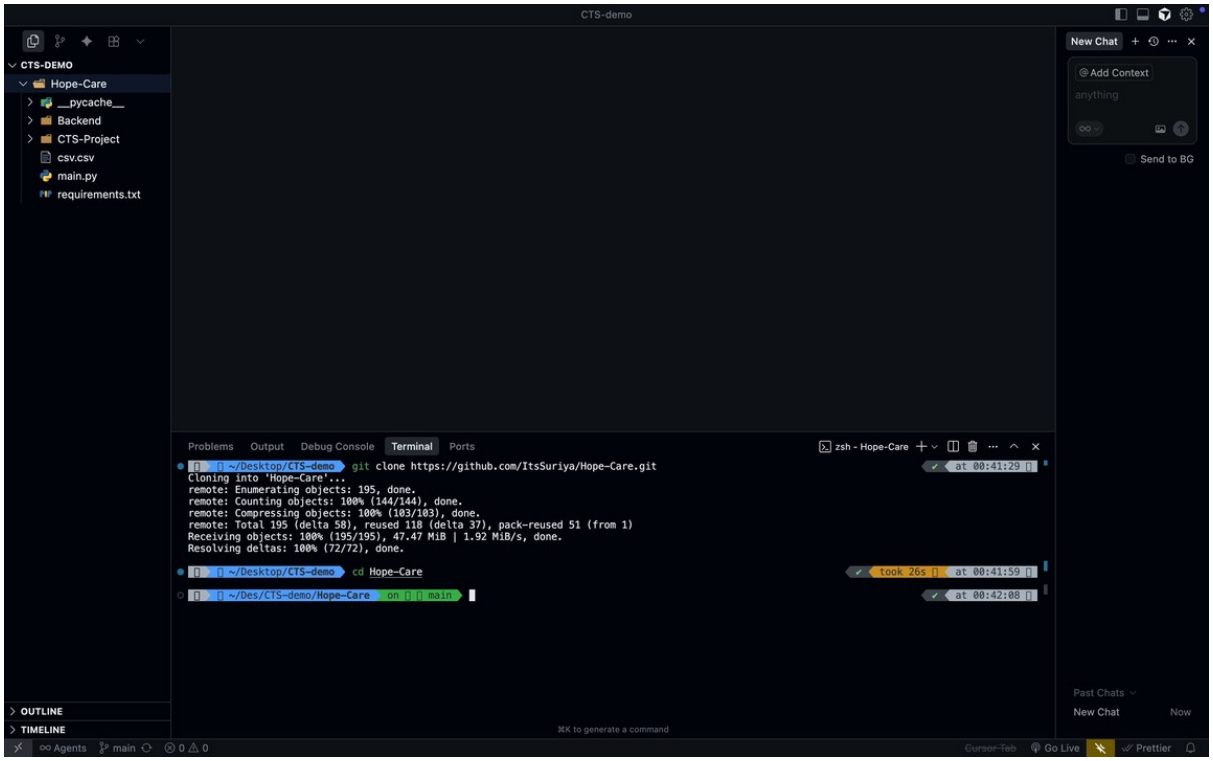
Step 5:Open The Terminal In the Cursor or VS Code



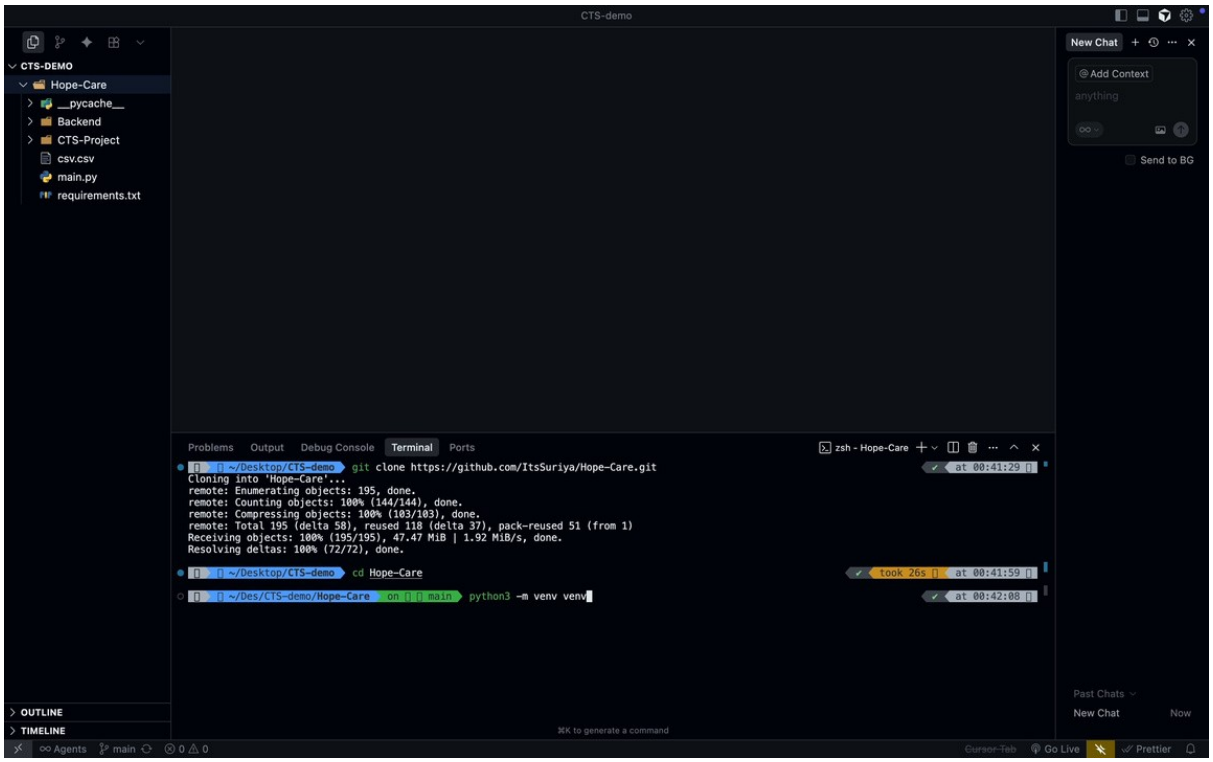
Clone The git



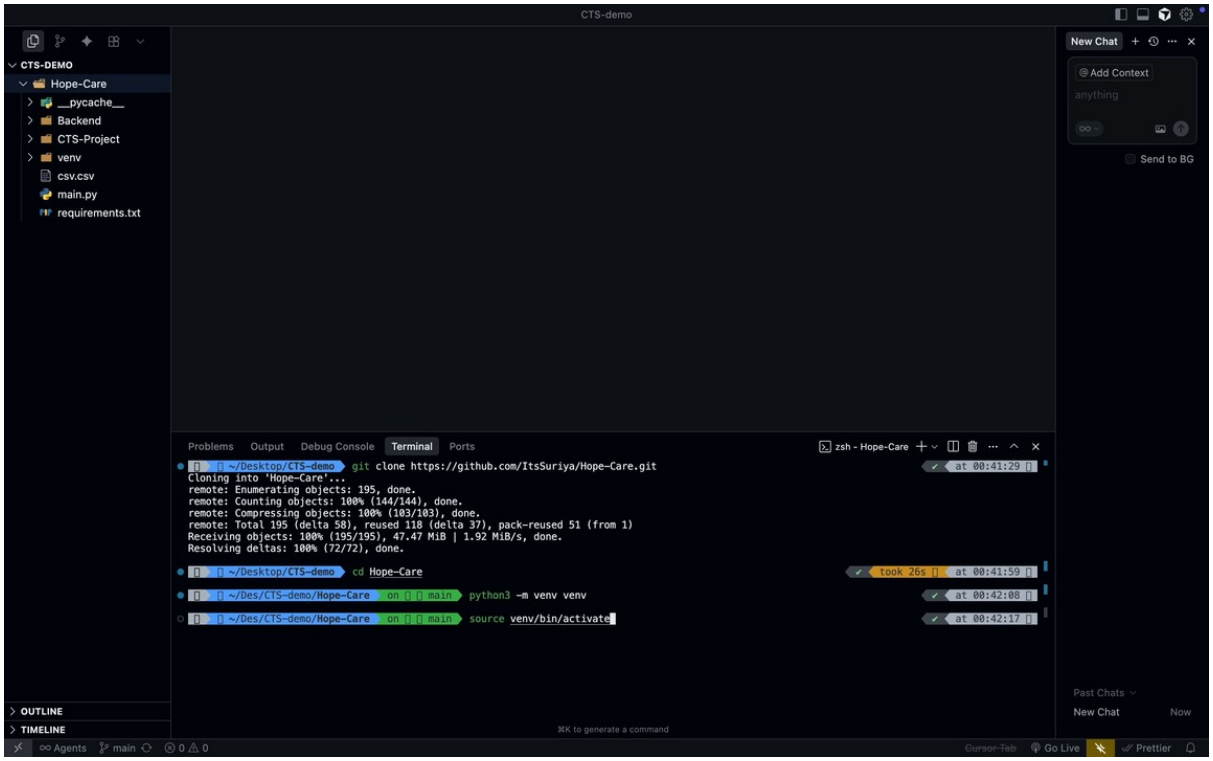
Change the Directory



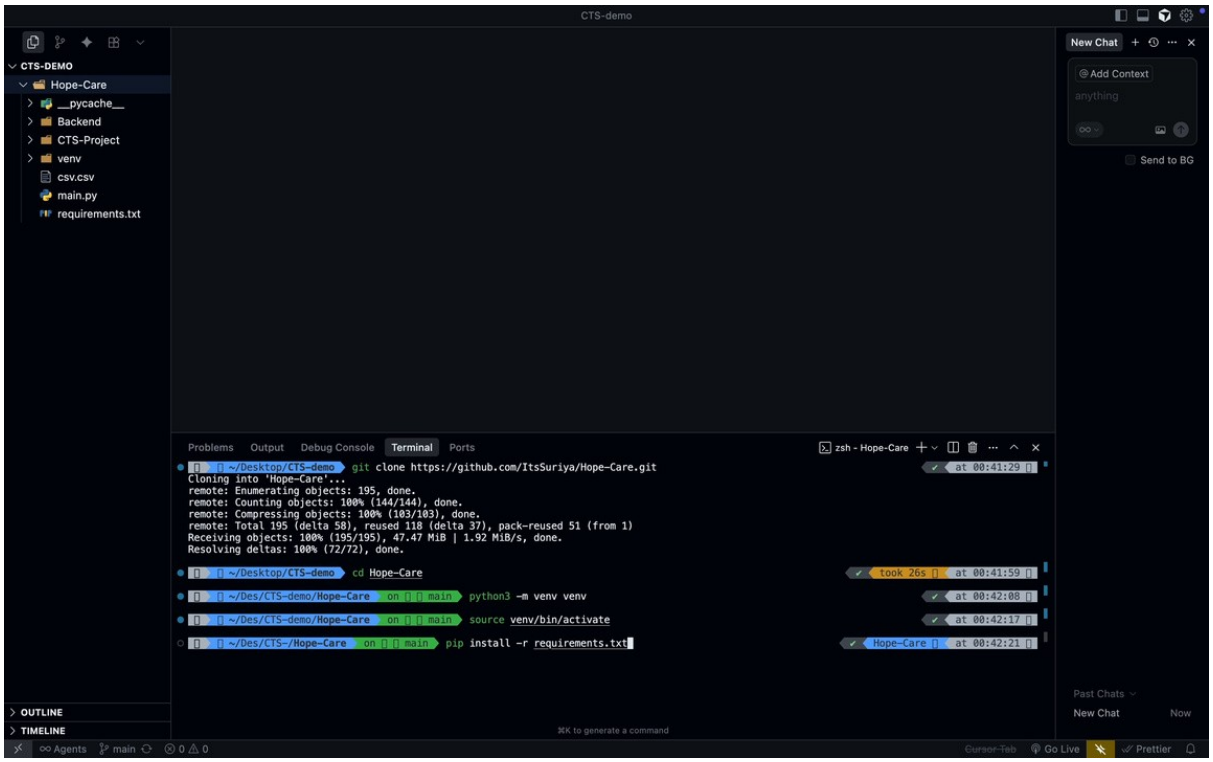
Create the Virtual enviroment



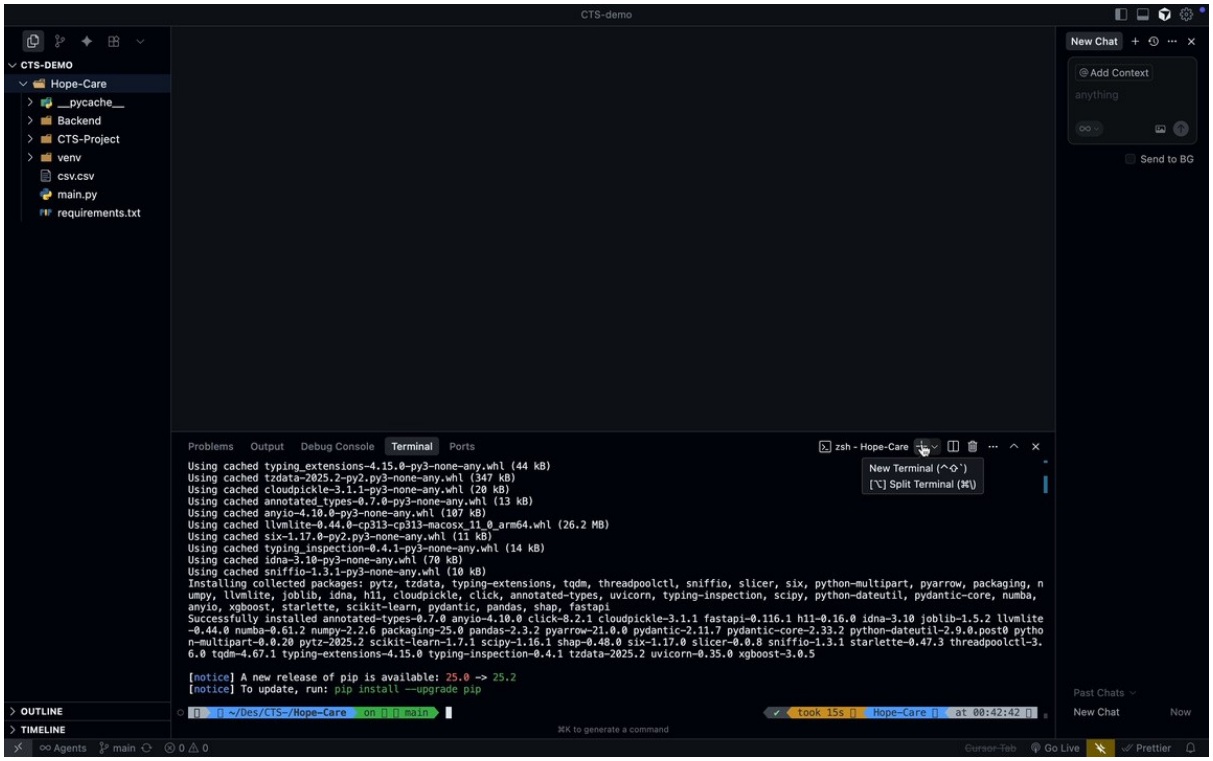
Activate the Virtual Enviroment



Install All the Dependencies From the requirement.txt



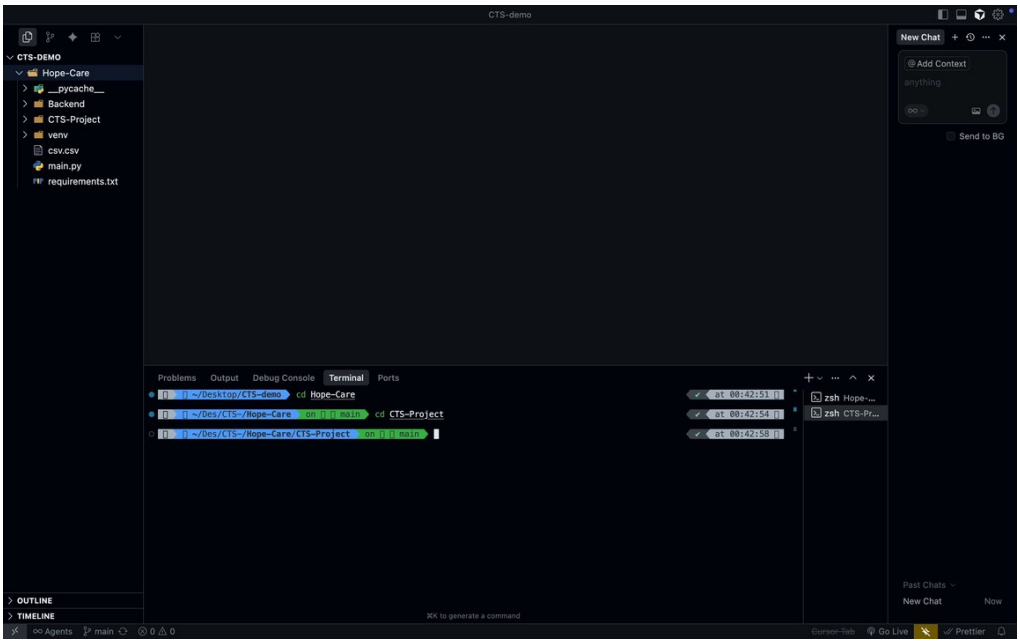
Open another Terminal



Change the directory to

cd Hope-Care

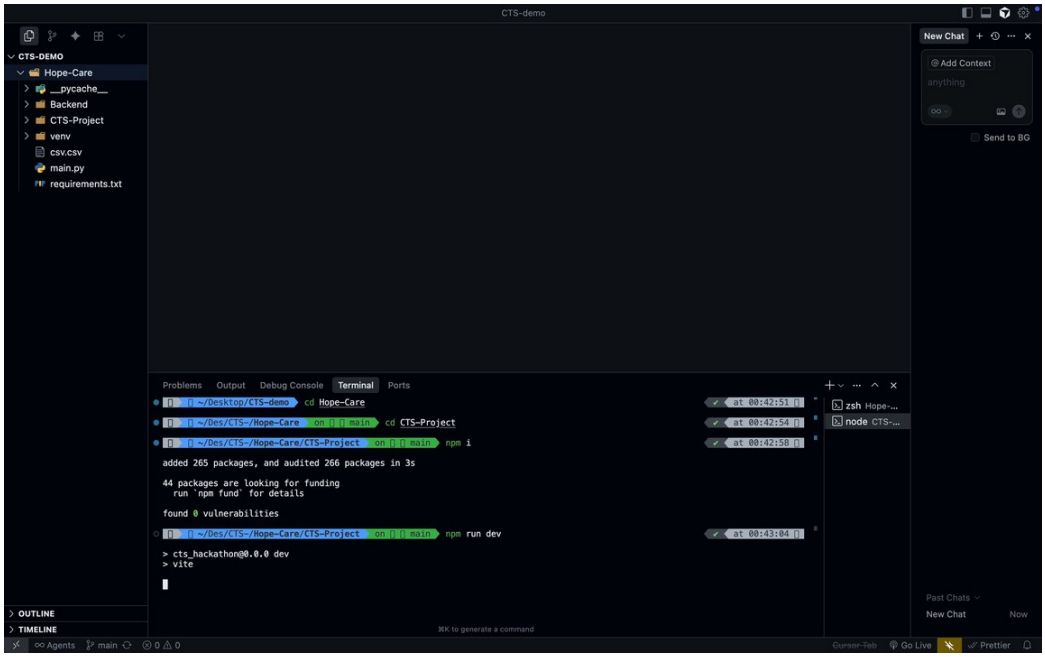
cd CTS-Project



Run these npm commands in the 2nd terminal

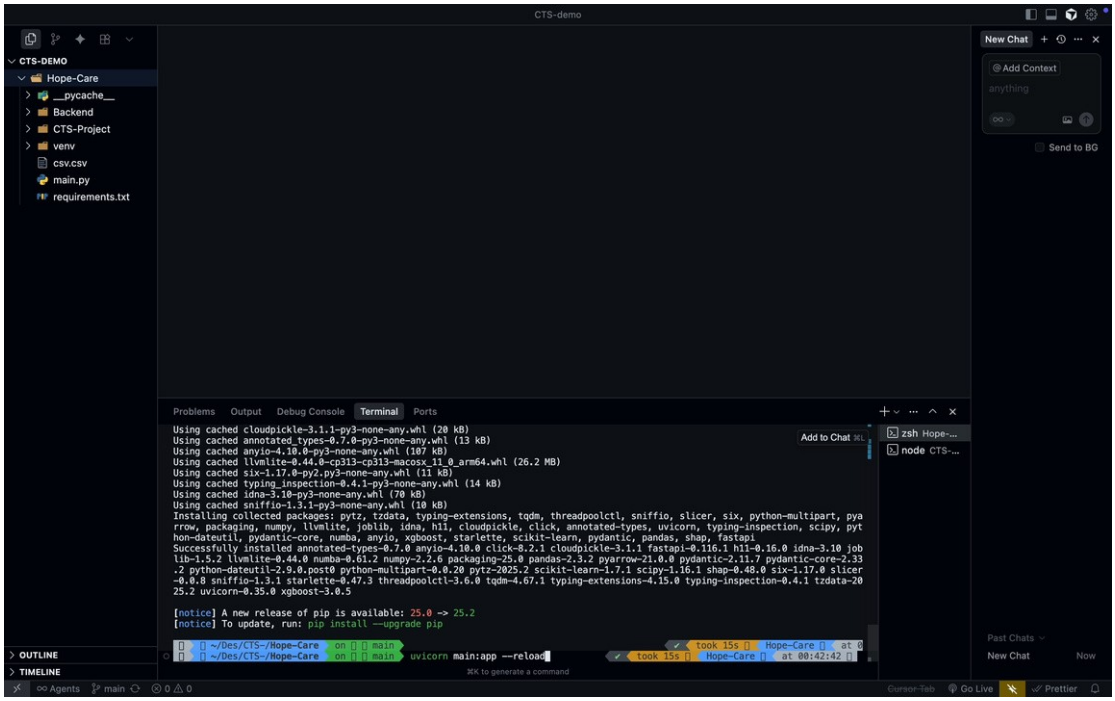
npm i

npm run dev

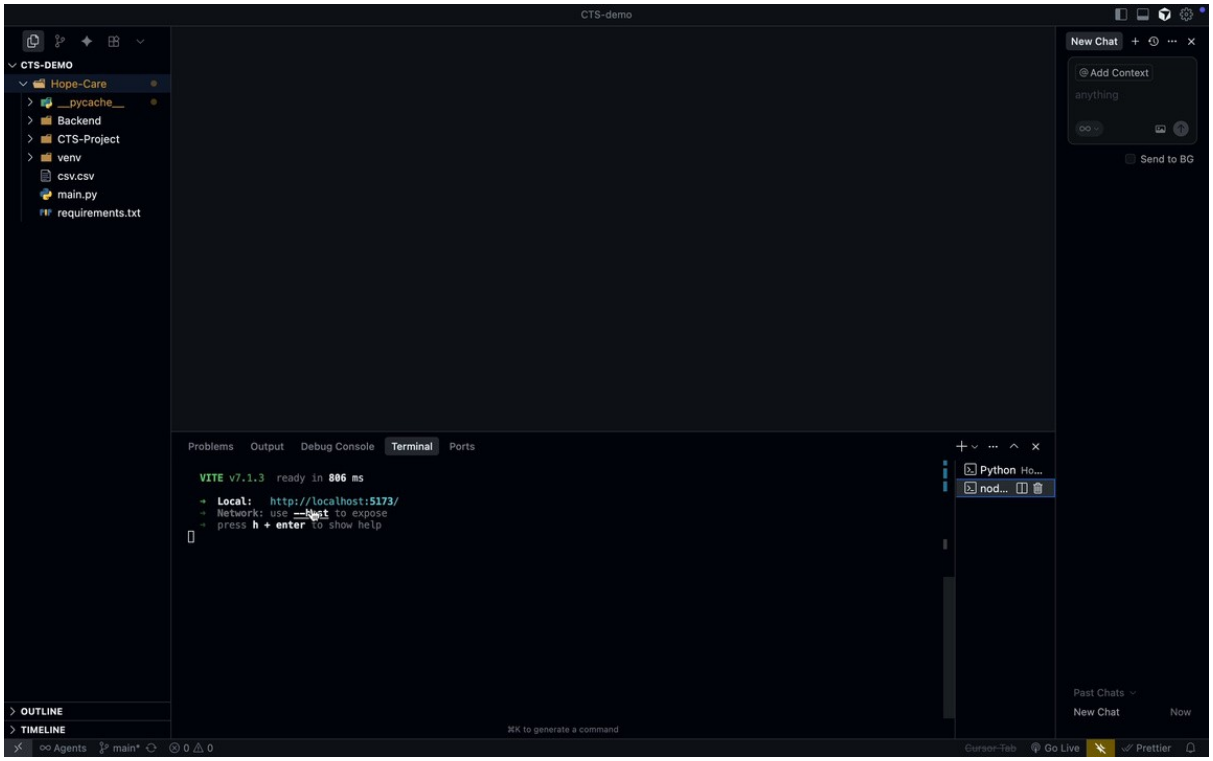


Run this in the First Terminal to run the backend

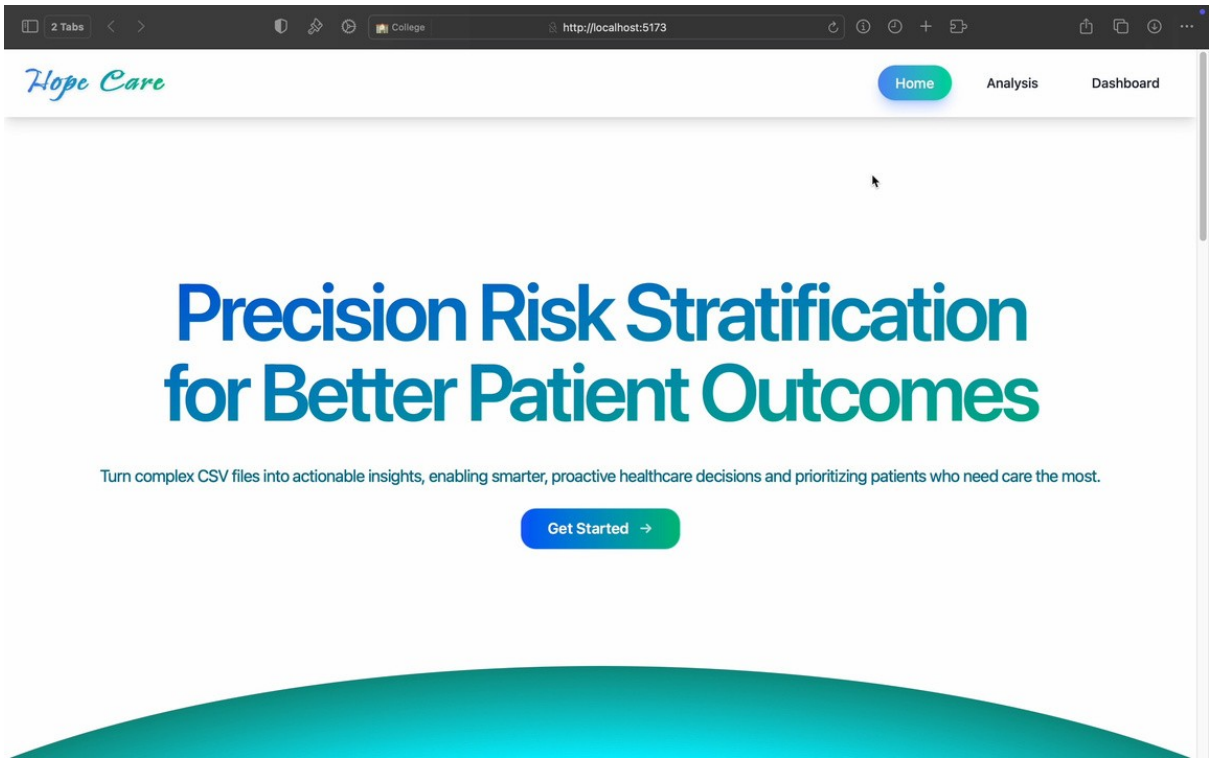
uvicorn main:app --reload



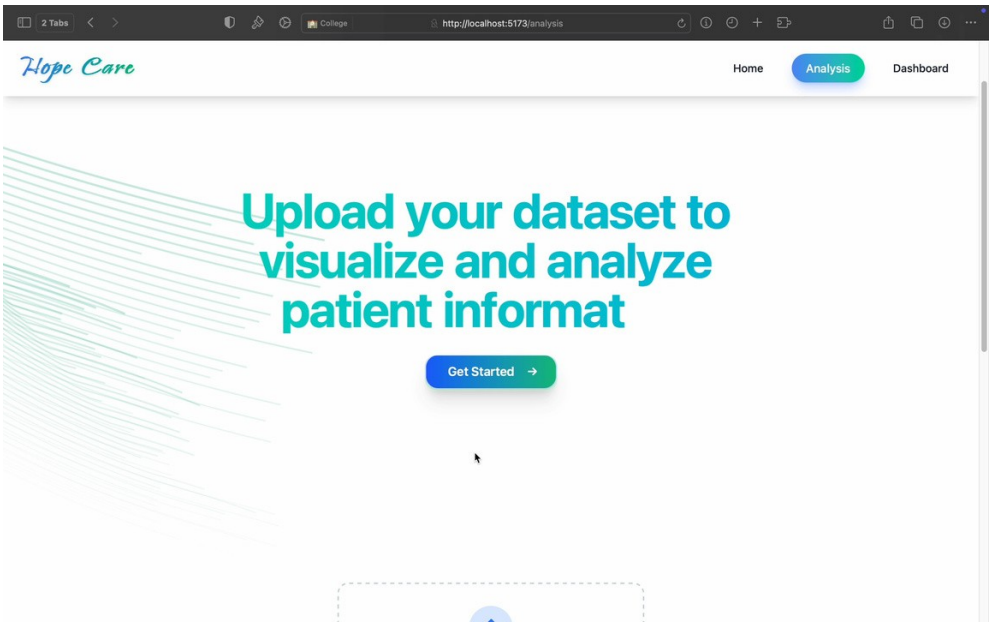
Click on the local host ip in the 2nd terminal



This opens the WebApplication



Go to Analysis Tab



Before that go back to cursor or VS Code

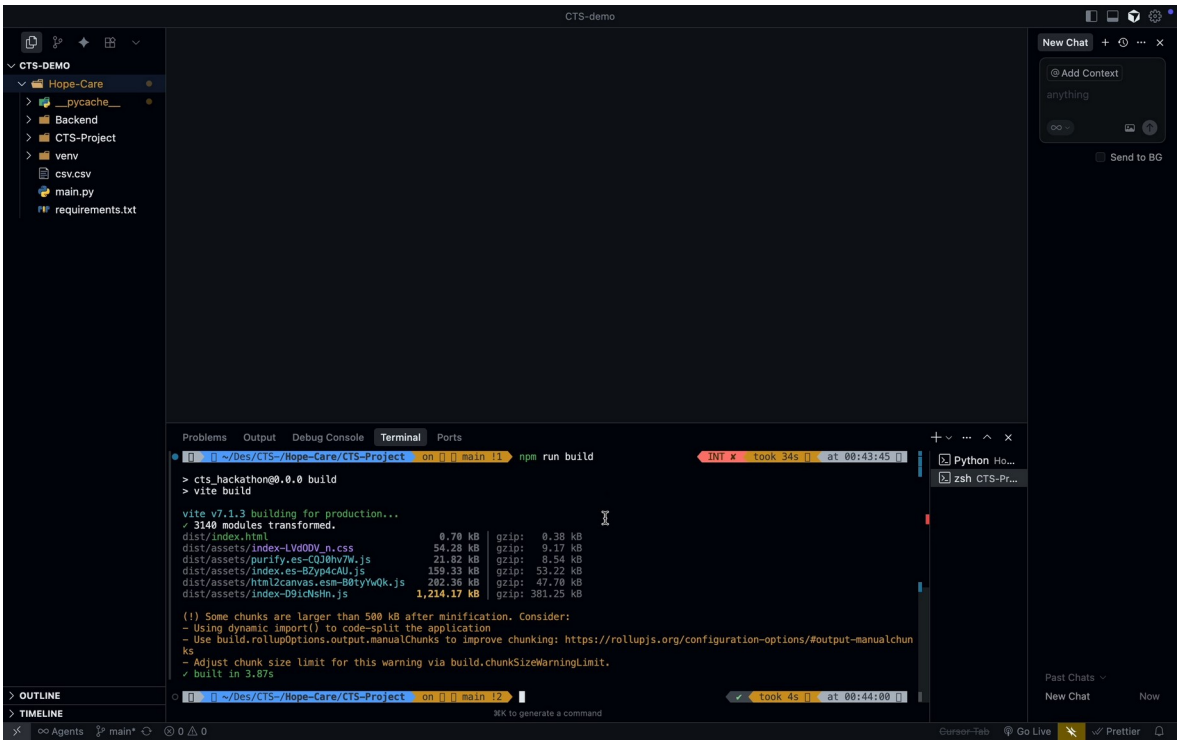
Close the uvicorn and npm dev run and build again

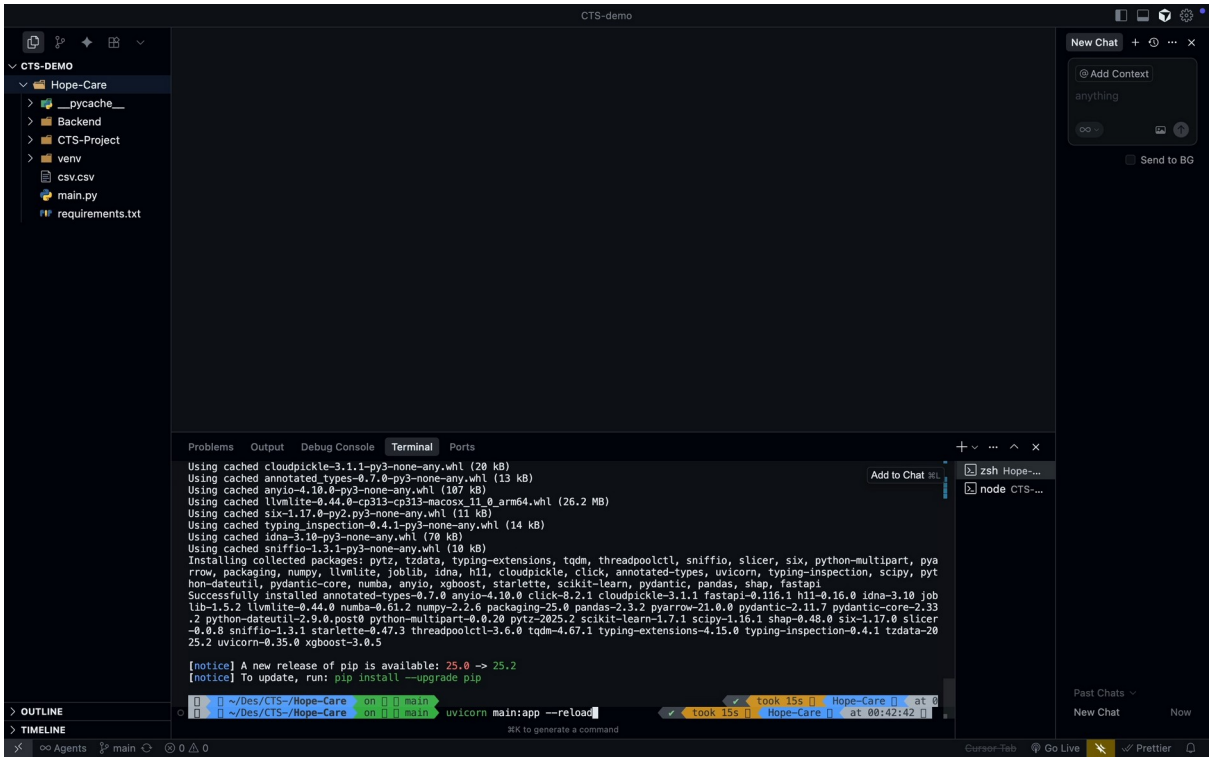
Run→npm run build

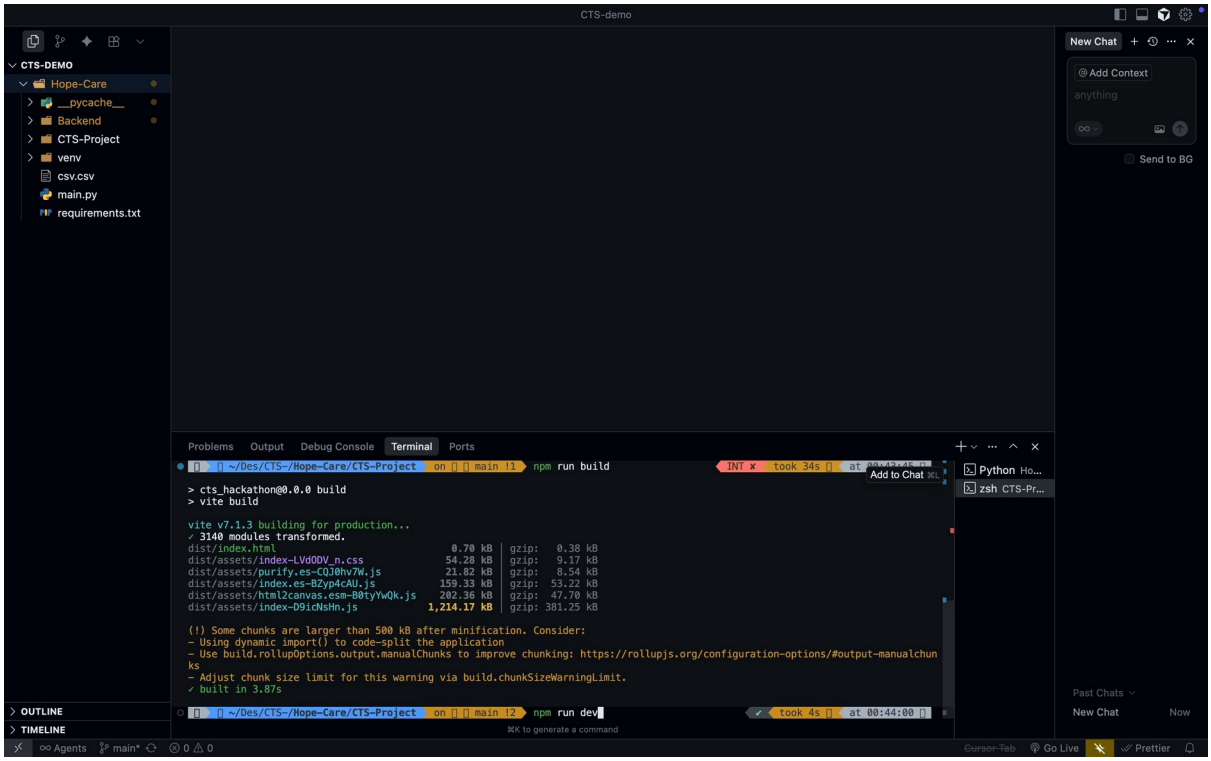
Run→ npm run dev

Run→ uvicorn main:app --reload

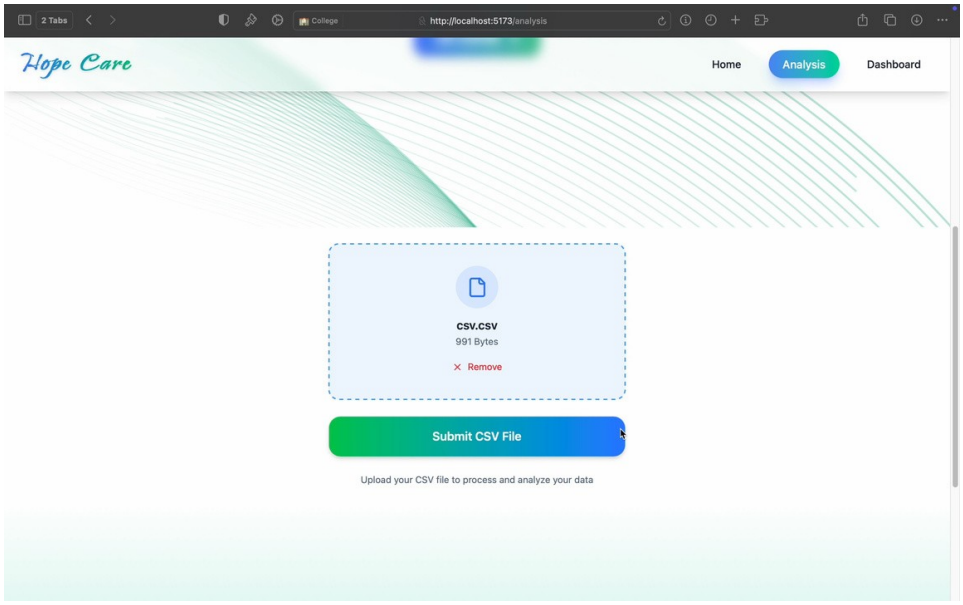
while the page is open in background


CTS-demo
CTS-DEMO
Hope-Care
__pycache__
Backend
CTS-Project
venv
csv.csv
main.py
requirements.txt
New Chat
Add Context
anything
Send to BG
Problems
Output
Debug Console
Terminal
Ports
~/Des/CTS-/Hope-Care/CTS-Project on main !1 npm run build
INT x took 34s at 00:43:45
> cts_hackathon@0.0.0 build
> vite build
vite v7.1.3 building for production...
✓ 3140 modules transformed.
dist/index.html 0.70 kB | gzip: 0.38 kB
dist/assets/index-LVdODV_n.css 54.28 kB | gzip: 9.17 kB
dist/assets/purify.es-CQJ0hv7W.js 21.82 kB | gzip: 8.54 kB
dist/assets/index.es-BZyp4cAU.js 159.33 kB | gzip: 53.22 kB
dist/assets/html2canvas.esm-B0tyYwQk.js 202.36 kB | gzip: 47.70 kB
dist/assets/index-D9icNsHn.js 1,214.17 kB | gzip: 381.25 kB
(!) Some chunks are larger than 500 kB after minification. Consider:
- Using dynamic import() to code-split the application
- Use build.rollupOptions.output.manualChunks to improve chunking: https://rollupjs.org/configuration-options/#output-manualchunks
- Adjust chunk size limit for this warning via build.chunkSizeWarningLimit.
✓ built in 3.87s
~/Des/CTS-/Hope-Care/CTS-Project on main !2
took 4s at 00:44:00
⌘K to generate a command
Python Ho...
zsh CTS-Pr...
Past Chats
New Chat Now
OUTLINE
TIMELINE
Agents main* 0 0
Go Live
Prettier

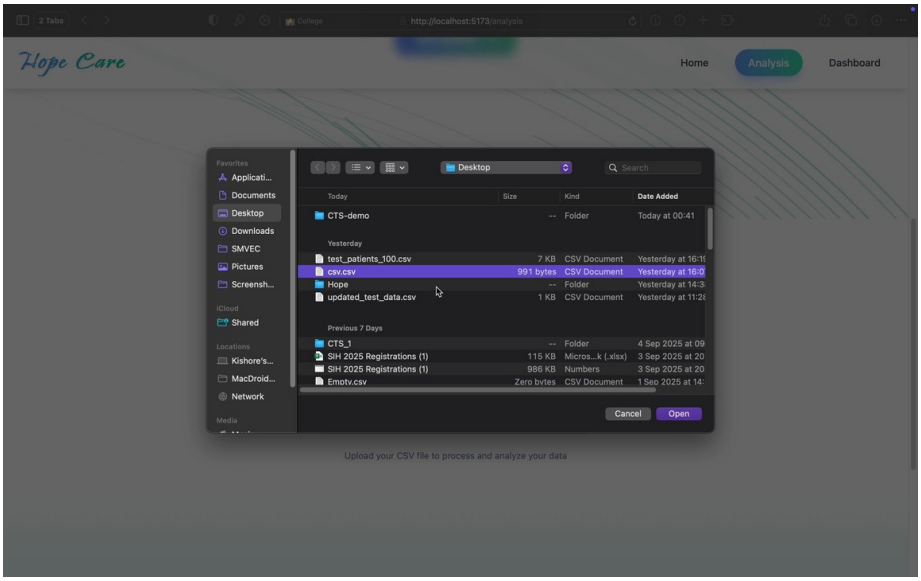

CTS-demo
CTS-DEMO
Hope-Care
__pycache__
Backend
CTS-Project
venv
csv.csv
main.py
requirements.txt
New Chat
Add Context
anything
Send to BG
Problems
Output
Debug Console
Terminal
Ports
Using cached cloudpickle-3.1.1-py3-none-any.whl (20 kB)
Using cached annotated_types-0.7.0-py3-none-any.whl (13 kB)
Using cached anyio-4.10.0-py3-none-any.whl (107 kB)
Using cached llvmlite-0.44.0-cp313-cp313-macosx_11_0_arm64.whl (26.2 MB)
Using cached six-1.17.0-py2.py3-none-any.whl (11 kB)
Using cached typing_inspection-0.4.1-py3-none-any.whl (14 kB)
Using cached idna-3.10-py3-none-any.whl (70 kB)
Using cached sniffio-1.3.1-py3-none-any.whl (10 kB)
Installing collected packages: pytz, tzdata, typing-extensions, tqdm, threadpoolctl, sniffio, slicer, six, python-multipart, pyarrow, packaging, numpy, llvmlite, joblib, idna, h11, cloudpickle, click, annotated-types, uvicorn, typing-inspection, scipy, python-dateutil, pydantic-core, numba, anyio, xgboost, starlette, scikit-learn, pydantic, pandas, shap, fastapi
Successfully installed annotated-types-0.7.0 anyio-4.10.0 click-8.2.1 cloudpickle-3.1.1 fastapi-0.116.1 h11-0.16.0 idna-3.10 joblib-1.5.2 llvmlite-0.44.0 numba-0.61.2 numpy-2.2.6 packaging-25.0 pandas-2.3.2 pyarrow-21.0.0 pydantic-2.11.7 pydantic-core-2.33.2 python-dateutil-2.9.0.post0 python-multipart-0.0.20 pytz-2025.2 scikit-learn-1.7.1 scipy-1.16.1 shap-0.48.0 six-1.17.0 slicer-0.0.8 sniffio-1.3.1 starlette-0.47.3 threadpoolctl-3.6.0 tqdm-4.67.1 typing-extensions-4.15.0 typing-inspection-0.4.1 tzdata-2025.2 uvicorn-0.35.0 xgboost-3.0.5
[notice] A new release of pip is available: 25.0 -> 25.2
[notice] To update, run: pip install --upgrade pip
~/Des/CTS-/Hope-Care on main
took 15s Hope-Care at 0
~/Des/CTS-/Hope-Care on main uvicorn main:app --reload
took 15s Hope-Care at 00:42:42
Add to Chat ⌘L
zsh Hope-...
node CTS-...
⌘K to generate a command
Past Chats
New Chat Now
OUTLINE
TIMELINE
Agents main 0 0
Go Live
Prettier


CTS-demo
CTS-DEMO
Hope-Care
__pycache__
Backend
CTS-Project
venv
csv.csv
main.py
requirements.txt
New Chat
@ Add Context
anything
Send to BG
Problems
Output
Debug Console
Terminal
Ports
~/Des/CTS-/Hope-Care/CTS-Project on main !1 npm run build
INT x took 34s
Add to Chat
Python Ho...
zsh CTS-Pr...
> cts_hackathon@0.0.0 build
> vite build
vite v7.1.3 building for production...
✓ 3140 modules transformed.
dist/index.html 0.70 kB | gzip: 0.38 kB
dist/assets/index-LVdODV_n.css 54.28 kB | gzip: 9.17 kB
dist/assets/purify.es-CQJ0hv7W.js 21.82 kB | gzip: 8.54 kB
dist/assets/index.es-BZyp4cAU.js 159.33 kB | gzip: 53.22 kB
dist/assets/html2canvas.esm-B0tyYwQk.js 202.36 kB | gzip: 47.70 kB
dist/assets/index-D9icNsHn.js 1,214.17 kB | gzip: 381.25 kB
(!) Some chunks are larger than 500 kB after minification. Consider:
- Using dynamic import() to code-split the application
- Use build.rollupOptions.output.manualChunks to improve chunking: https://rollupjs.org/configuration-options/#output-manualchunks
- Adjust chunk size limit for this warning via build.chunkSizeWarningLimit.
✓ built in 3.87s
~/Des/CTS-/Hope-Care/CTS-Project on main !2 npm run dev
took 4s at 00:44:00
⌘K to generate a command
OUTLINE
TIMELINE
Past Chats
New Chat Now
Agents main* 0 0
Cursor Tab
Go Live
Prettier

`Comeback to the Webapplication and go to Analysis Tab Scroll down you will find the Upload CSV file table



Upload the sample CSV files provided to test



That’s it you will get the output and you can go through the processed data and even download the report ,also you can access the Dashboard tab to

views the Analytics Visualized