Customer Support System

Using ChatGPT, openAI API and node.js.

Prepared by: Kruti Dhyani

Table of content

Introduction

- To satisfy need of the customer using Leveraging Artificial Intelligence for enhanced support the Application is developed
- Used openAI API_key and Org_ID to use openAI API. For that account is created and set the key and org_id then credit the balance to use its API
- **If your API_key is compromised then you can create a new key and use it in .env file. You will get email from open AI that your key is compromised and we disabled it.**
- The Objective is develop web based application for website related queries. If Information is mentioned in that website It prints the ans otherwise prints Don’t know.
- **Key Technology:** ChatGPT by OpenAI, Node.js.

Project Phases

Three phases

1. Phase 1: Data Collation-Gathering information from various sources

   Data can be in any form: Webpages, Local Files, Database, Videos, Drives

   For this first project Web Pages are used

2. Phase 2: Training-preparing ChatGPT project for effective response

   Training can be done using files, API and Fine-tuning

   For this first project ChatGPT API are used

3. Phase 3: User Interface - output method.

   Text message or Speech. For this project **Text message is used**

Design

To achieve fully we have to do the following in sequence. This steps are used for creating NLP model that collect data and model training for large dataset or LLM

1. **Crawler [Data collection]:** In this step, First, the given website or page link is passed in as domain and then it crawl the website using this steps. Generate .csv file which has all the website crawling information and stored with name “Scraped.csv”
2. **Embedding [Model Training]:**Once you have gathered a substantial dataset, you can proceed to the embedding phase, which involves training an NLP model like GPT (Generative Pre-trained Transformer) on this data. The model learns to understand the patterns, relationships, and semantics present in the text data through a process called "embedding." Here “embeddings.csv” file is generated at this step.
3. **Testing:** output gives by combining above 2 methods using Question - Answers model, either by command line argument or GUI based method.

Prerequisite Implementation in node.js

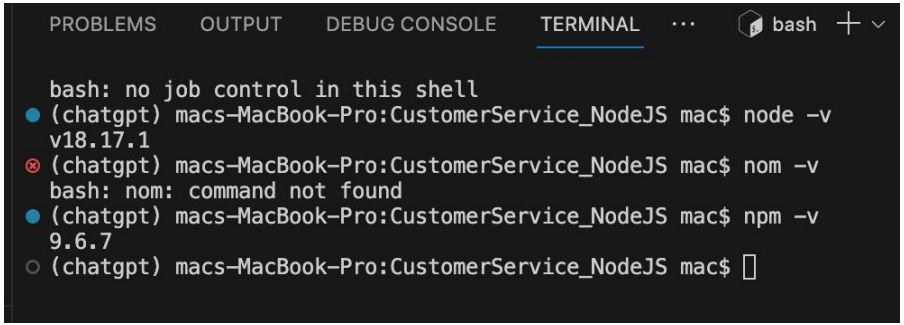
<u>NOTE:</u> work on VScode[editor] terminal only otherwise sometimes it gives error

1. Do not require to create a new virtual environment every time. If you created venv successfully in homework-1 then Directly write command

workon <your venv name > For Example, workon chatgpt

2. Check your node and npm is installed or not using **node -v and npm -v** respectively. If not then install it using **brew install node** command.

<u>Output screenshot=></u>



Trial of given sample code: clone repository steps

1. Make a folder using command: **mkdir quickstart_nodejs**
2. Go to that folder using command: **cd quickstart_nodejs/**
3. Now clone the repository with command: **git clone https://github.com/openai/openai-quickstart-node.git**

OUTPUT Screenshot:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE

bash: no job control in this shell
(chatgpt) macs-MacBook-Pro:CustomerService_NodeJS mac$ mkdir quickstart_nodejs
(chatgpt) macs-MacBook-Pro:CustomerService_NodeJS mac$ cd quickstart_nodejs/
(chatgpt) macs-MacBook-Pro:quickstart_nodejs mac$ git clone https://github.com/openai/openai-quickstart-node.git
Cloning into 'openai-quickstart-node'...
remote: Enumerating objects: 120, done.
remote: Counting objects: 100% (120/120), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 120 (delta 52), reused 92 (delta 39), pack-reused 0
Receiving objects: 100% (120/120), 93.61 KiB | 1.46 MiB/s, done.
Resolving deltas: 100% (52/52), done.
(chatgpt) macs-MacBook-Pro:quickstart_nodejs mac$
```

Reference:website:https://github.com/openai/openai-quickstart-node.git

Trial of given sample code:set .env & check flies

1. Go to the folder: **cd openai-quickstart-node**
2. Check all the files using command: ls -al
3. Create .env file of your own in that respective project folder. Stored your openAI API key using one line:OPENAI_API_KEY = ‘YOUR_API_KEY’

OUTPUT screenshot:

```
Resolving deltas: 100% (52/52), done.
(chatgpt) macs-MacBook-Pro:quickstart_nodejs mac$ cd openai-quickstart-node
(chatgpt) macs-MacBook-Pro:openai-quickstart-node mac$ ls -al
total 104
drwxr-xr-x  12 mac  staff    384 Oct  2 14:05 .
drwxr-xr-x   3 mac  staff     96 Oct  2 14:05 ..
-rw-r--r--   1 mac  staff     91 Oct  2 14:05 .env.example
drwxr-xr-x  12 mac  staff    384 Oct  2 14:05 .git
drwxr-xr-x   3 mac  staff     96 Oct  2 14:05 .github
-rw-r--r--   1 mac  staff    377 Oct  2 14:05 .gitignore
-rw-r--r--   1 mac  staff   1063 Oct  2 14:05 LICENSE
-rw-r--r--   1 mac  staff   1374 Oct  2 14:05 README.md
-rw-r--r--   1 mac  staff  29234 Oct  2 14:05 package-lock.json
-rw-r--r--   1 mac  staff    338 Oct  2 14:05 package.json
drwxr-xr-x   5 mac  staff    160 Oct  2 14:05 pages
drwxr-xr-x   3 mac  staff     96 Oct  2 14:05 public
(chatgpt) macs-MacBook-Pro:openai-quickstart-node mac$
```

Trial of given sample code:execute file

1. Install dependencies with npm. Which helps to run node.js files: **npm install**

   Output screenshot:

```
added 26 packages, and audited 27 packages in 4s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

2. Now execute your project using: **npm run dev**

   Here in package. Json file has one key “dev” in which has value “next dev”

   Output screenshot:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POSTMAN CONSOLE        bash - openai-quickstart-node

 'next-swc.darwin-x64.node' (load command 0x80000034 is unknown)
event - compiled client and server successfully in 1530 ms (150 modules)
Attention: Next.js now collects completely anonymous telemetry regarding usage.
Follow link (cmd + click)  d to shape Next.js' roadmap and prioritize features.
luding how to opt-out if you'd not like to participate in this anonymous program, by visiting the following URL:
https://nextjs.org/telemetry
```

Trial of given sample code: error solution

1. While running npm run dev first time, It throws error:

```
(chatgpt) macs-MacBook-Pro:CustomerService_NodeJS mac$ npm run dev

> simpletrynode@0.1.0 dev
> next dev

sh: next: command not found.  Provide solution
```

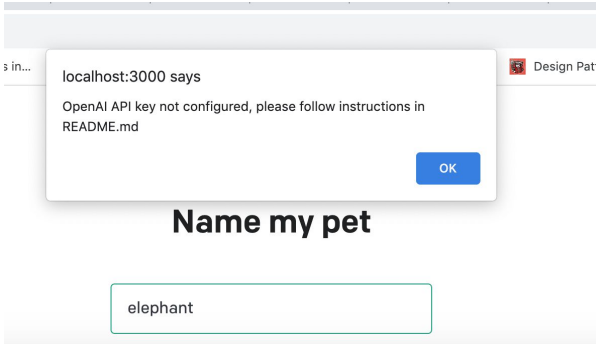
Solution to remove error install next package using following command:

```
npm install next
```

Trial of given sample code: error solution

2. While successfully run using command **npm run dev. Port is listening to 3000 so, to run the webpage type** http://localhost:3000

Throws error=>

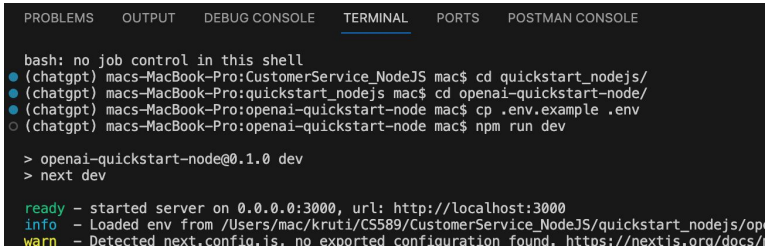


Solution=>

.env file is missing/ may be generated in wrong path

Solution copy the file in the proper directory using

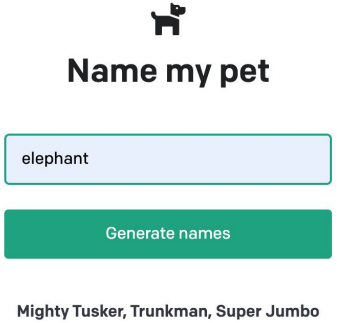
Command: cp .env.example .env

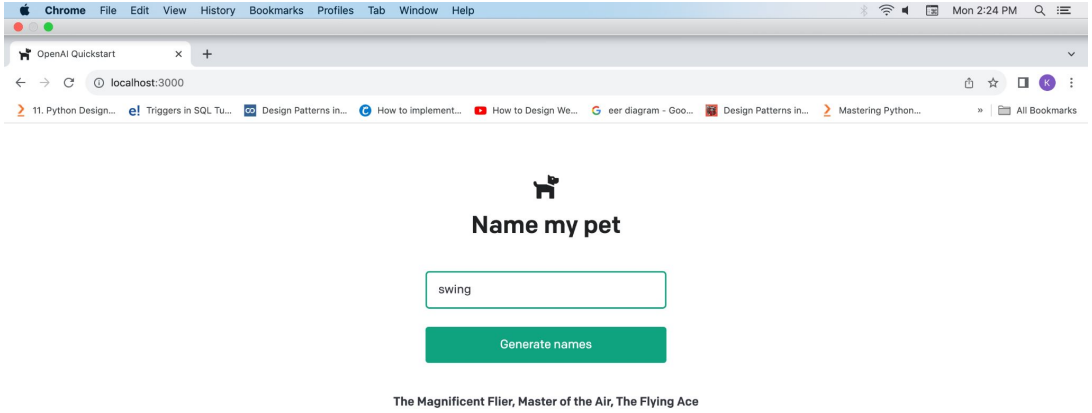


Trial of given sample code: error solution

Run command **npm run dev, Port is listening to 3000 so, to run the webpage type** http://localhost:3000

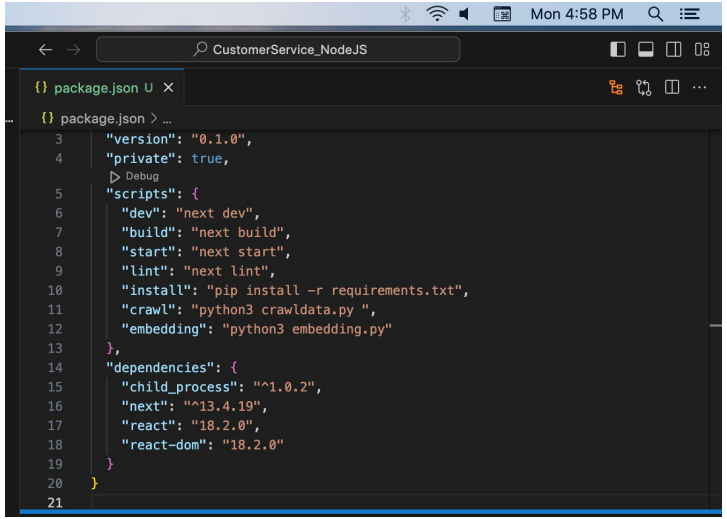
Input any name and to get output click on generate output button and you get output as name description

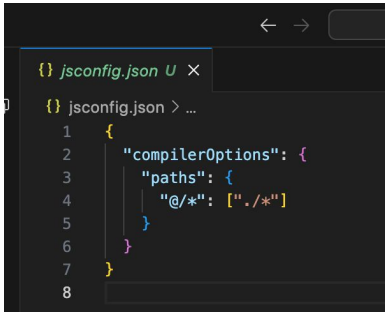


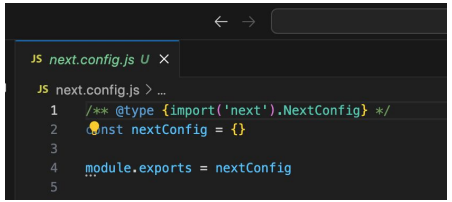


Implementation of Customer support system

1. Use code of index.js, getResult.js and index.module.css worked for above pet name project
2. Do some Changes in your package.json file. Screenshot -1 from leftis the following
3. Make sure your jsconfig.json and next.config.js file is like below and its generated.



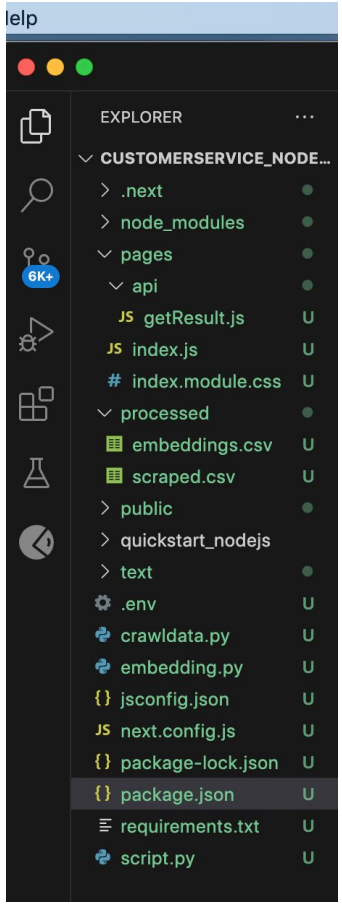




FILE HIERARCHY

- File hierarchy is same as home work1 or python flask only node js related supported files are added for view and supporting files.

- The code for crawl, embedding and script is same as we did for last time



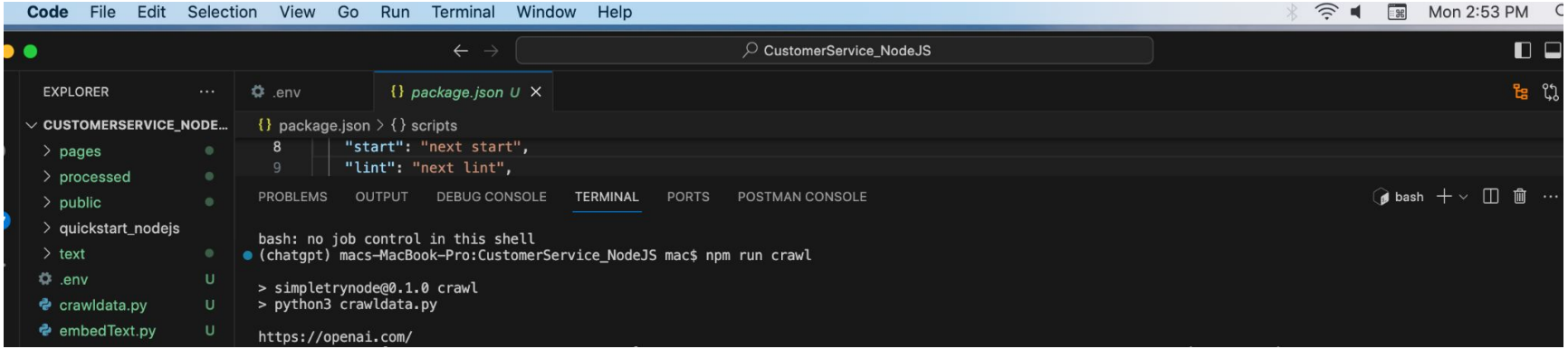
Executing Project

1. **Step:1** crawling the website so for that run the command **npm run crawl**

   In package.json file key:crawl and value: “python3 crawldata.py”. So, automatically run crawldata.py file.

   Output file: scraped.csv file is generated

   Output screenshot:



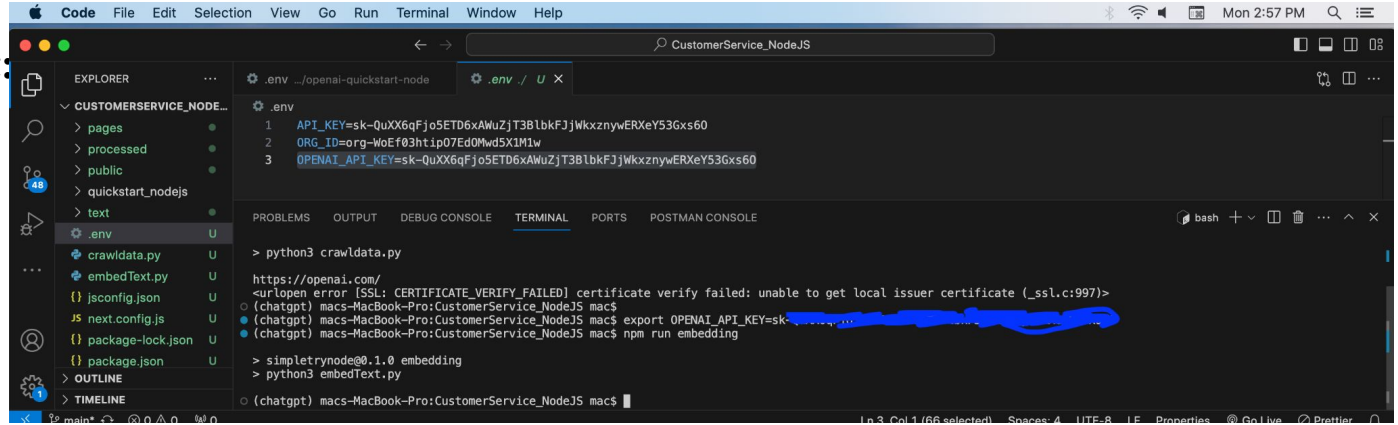
Executing Project cont..

2. **Step:2** embedding, Before embedding export the openai_api_key to avoid any error then run the code, so for that run the command **npm run embedding**

In package.json file key:embedding and value: “python3 embedding.py”. So, automatically run crawldata.py file.

Output file: embeddings.csv file is generated

Output screenshot:

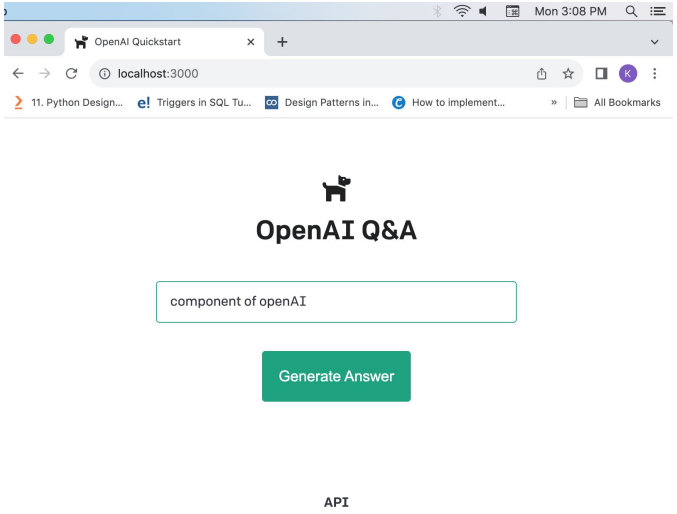


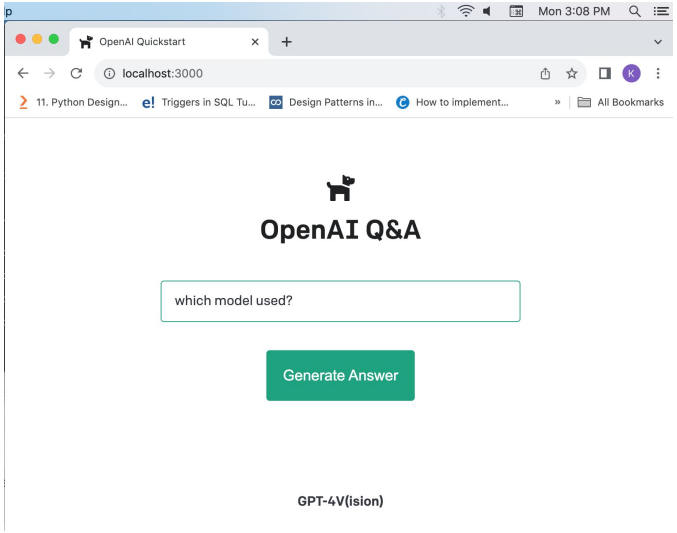
Test the Application

- web based Output: using node.js

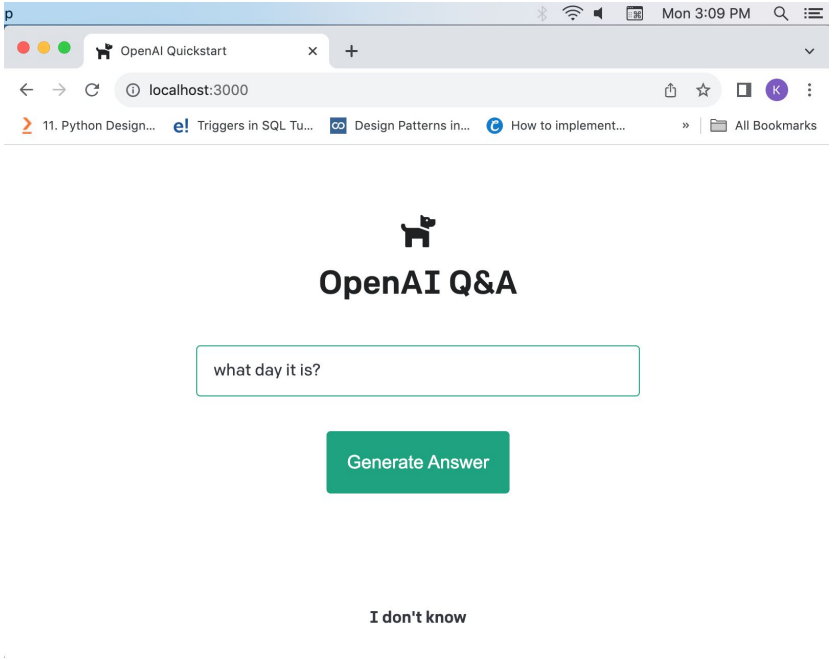
3. **Step:3 Execute the program using npm run dev. Same its running on http://localhost:3000**

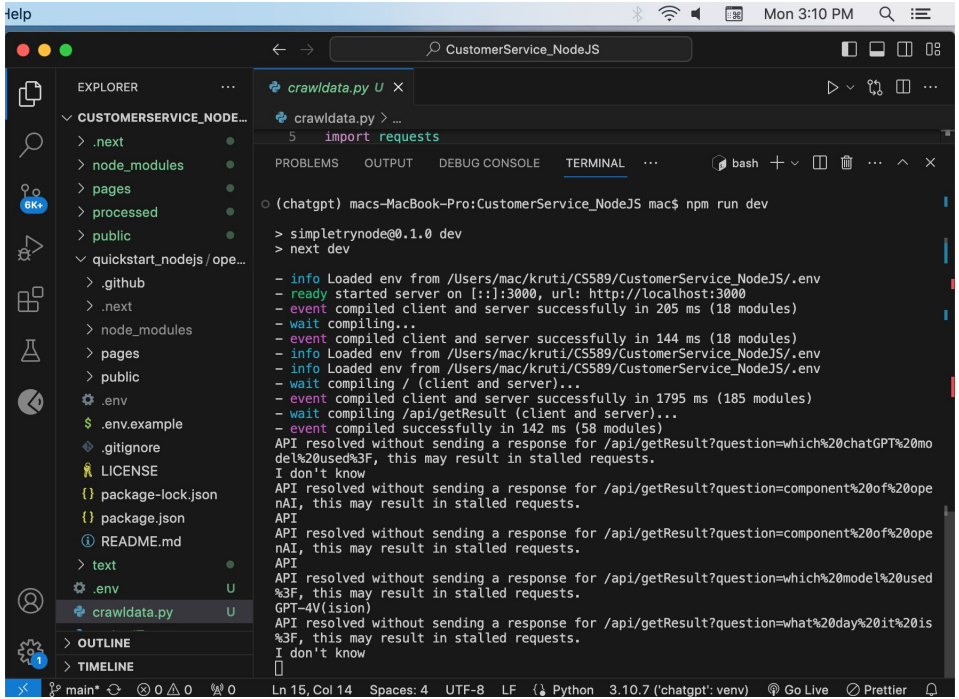
output:





Test cont.


Mon 3:09 PM
OpenAI Quickstart
localhost:3000
11. Python Design...
Triggers in SQL Tu...
Design Patterns in...
How to implement...
All Bookmarks
OpenAI Q&A
what day it is?
Generate Answer
I don't know


Mon 3:10 PM
CustomerService_NodeJS
EXPLORER
CUSTOMERSERVICE_NODE...
.next
node_modules
pages
processed
public
quickstart_nodejs / ope...
.github
.next
node_modules
pages
public
.env
.env.example
.gitignore
LICENSE
package-lock.json
package.json
README.md
text
.env
crawldata.py
OUTLINE
TIMELINE
crawldata.py U
import requests
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
bash
(chatgpt) macs-MacBook-Pro:CustomerService_NodeJS mac$ npm run dev
> simpletrynode@0.1.0 dev
> next dev
- info Loaded env from /Users/mac/kruti/CS589/CustomerService_NodeJS/.env
- ready started server on [::]:3000, url: http://localhost:3000
- event compiled client and server successfully in 205 ms (18 modules)
- wait compiling...
- event compiled client and server successfully in 144 ms (18 modules)
- info Loaded env from /Users/mac/kruti/CS589/CustomerService_NodeJS/.env
- info Loaded env from /Users/mac/kruti/CS589/CustomerService_NodeJS/.env
- wait compiling / (client and server)...
- event compiled client and server successfully in 1795 ms (185 modules)
- wait compiling /api/getResult (client and server)...
- event compiled successfully in 142 ms (58 modules)
API resolved without sending a response for /api/getResult?question=which%20chatGPT%20model%20used%3F, this may result in stalled requests.
I don't know
API resolved without sending a response for /api/getResult?question=component%20of%20openAI, this may result in stalled requests.
API
API resolved without sending a response for /api/getResult?question=component%20of%20openAI, this may result in stalled requests.
API
API resolved without sending a response for /api/getResult?question=which%20model%20used%3F, this may result in stalled requests.
GPT-4V(ision)
API resolved without sending a response for /api/getResult?question=what%20day%20it%20is%3F, this may result in stalled requests.
I don't know
main*
Ln 15, Col 14
Spaces: 4
UTF-8
LF
Python
3.10.7 ('chatgpt': venv)
Go Live
Prettier

Conclusion

AI contributed to generate NLP projects using API based on large dataset model very quick compare to traditional approach. It require some basic knowledge and how to connect the phases. Used 3 steps crawling, embedding and testing in sequence to implement. Perform both Command based and web based solution. For Node.js implementation add supported files and packages mentioned.

Github link

Github project repository link:

https://github.com/DKruti/CustomerService_NodeJS

References

- https://github.com/openai/openai-quickstart-node.git
- https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine_learning/chatgpt/slide/exercise_chatgpt.html
- https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine_learning/chatgpt/slide/quickstart.html#Quickstart%20-%20Node.js