

Proiect PCLP3 – NOANbot

(VLADISLAV NORIS-VICTOR & NEACȘU ANDREI-GEORGIAN)

GRUPA 424D

CUPRINS:

1. ReadMe
2. Obținere API key
3. Dovadă funcționalitate program NOANbot
4. Dovadă funcționalitate program NOANbot_GUI
5. Probleme întâmpinate
6. Link ChatGPT & github
7. Bibliografie

README pentru Interfața NOANbot

Această documentație oferă o prezentare a codului NOANbot Interface, o interfață simplă pentru interacțiunea cu modelul OpenAI GPT-3.5 Turbo prin intermediul API-ului OpenAI, folosind Tkinter pentru interfața grafică.

#DESCRIERE GENERALĂ

NOANbot Interface permite utilizatorilor să comunice cu modelul OpenAI GPT-3.5 Turbo, introducând mesaje de text într-o interfață grafică bazată pe Tkinter. Istoricul conversației este stocat, iar fiecare interacțiune cu modelul este afișată într-un format de chat.

#CERINȚE DE SISTEM

Înainte de a rula codul, asigurați-vă că aveți:

Python 3.x: Versiunea 3.x a limbajului de programare Python este necesară pentru a executa codul.

#BIBLIOTECI PYTHON

Tkinter: Tkinter este o bibliotecă standard Python utilizată pentru crearea interfețelor grafice.

OpenAI GPT-3.5 Turbo: Acest cod utilizează serviciul GPT-3.5 Turbo oferit de OpenAI. Asigurați-vă că aveți acces la API-ul OpenAI și că ați configurat cheia API corespunzătoare.

#INSTALARE

Copiați codul într-un fișier Python.

Configurați cheia API OpenAI înlocuind șirul de caractere "API KEY" în `openai.api_key = "API KEY"` cu cheia API reală.

#UTILIZARE

Rulați scriptul după configurarea cheii API OpenAI.

Fereastra GUI va apărea cu o zonă de text pentru afișarea conversației și o zonă de introducere pentru trimiterea mesajelor către NOANbot.

Tastați întrebarea în zona de introducere și faceți clic pe butonul "→" pentru a trimite mesajul.

Răspunsul NOANbot va fi afișat în zona de conversație.

#STRUCTURA CODULUI

Clasa NOANbotInterface administrează GUI Tkinter și gestionează interacțiunile utilizatorului.

Metoda send_message trimite inputul utilizatorului la modelul OpenAI GPT-3.5 Turbo și afișează răspunsurile în zona de conversație.

Metoda display_message formatează și afișează mesajele în fereastra de chat.

#NOTE

Asigurați-vă că cheia API OpenAI este păstrată confidențial și nu este distribuită public.

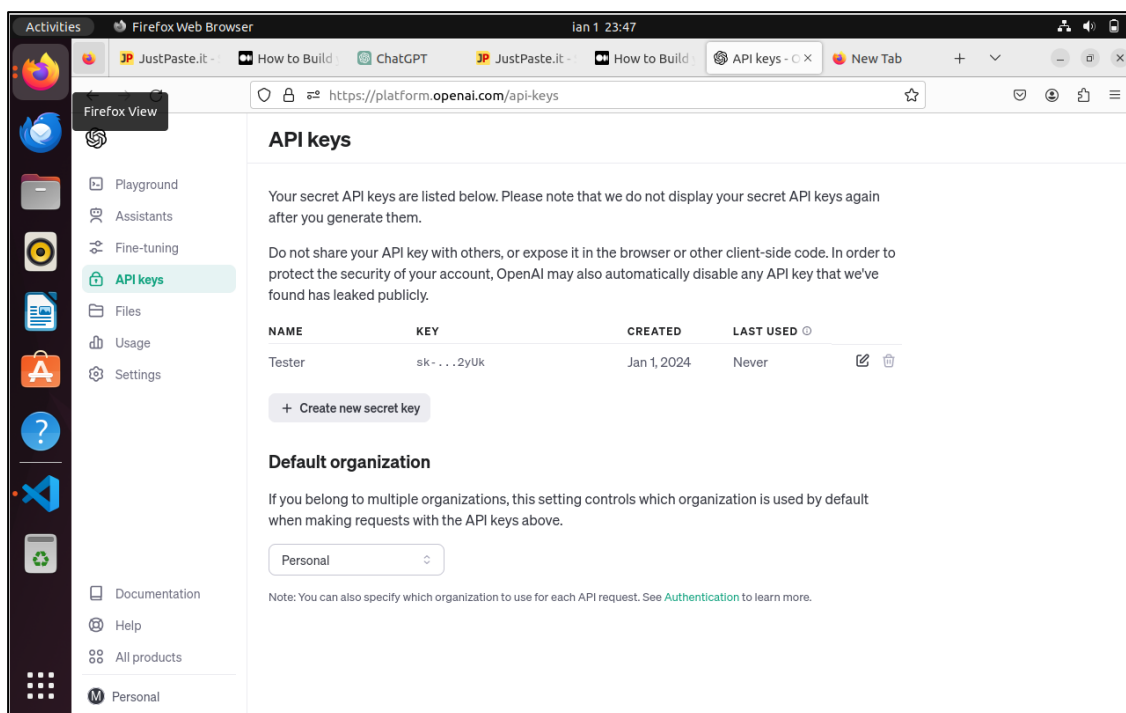
Acest cod este un exemplu de bază și poate fi extins pentru cazuri de utilizare mai complexe sau integrate în alte aplicații.

OBTINERE API KEY

Accesând site-ul <https://platform.openai.com/api-keys>, avem posibilitatea de a genera o cheie secretă API, cu ajutorul căreia vom executa programul. Astfel, se va realiza dialogul între utilizator și NAONbot cu succes.

Cheia API (Application Programming Interface) este necesară pentru autentificarea și autorizarea accesului la serviciul OpenAI, în acest caz, la modelul GPT-3.5 Turbo. Pentru a utiliza serviciile oferite de OpenAI, trebuie să ave o cheie API validă.

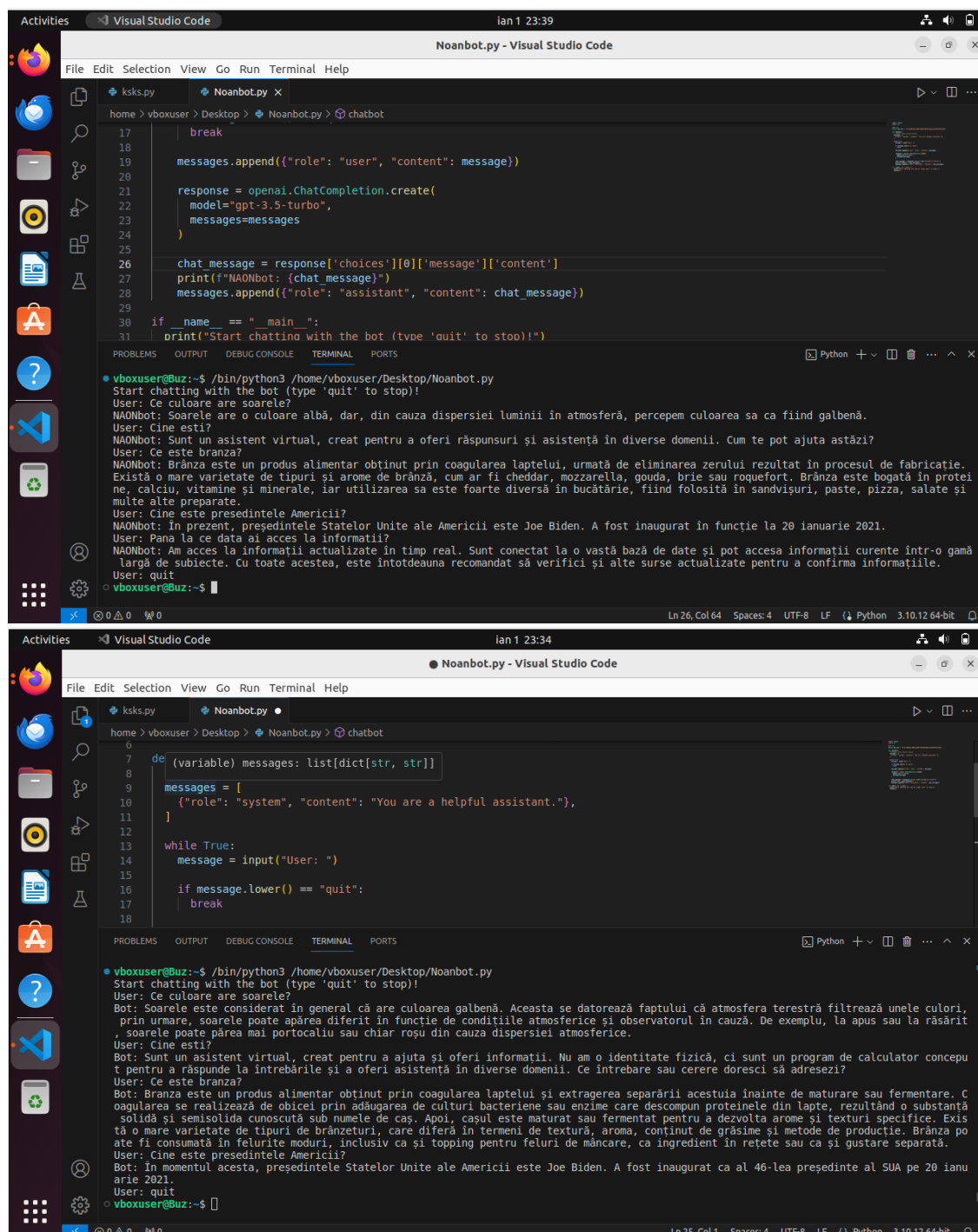
Cheia API servește drept mijloc de identificare, permitând OpenAI să urmărească și să gestioneze utilizarea serviciului lor. De asemenea, ajută la asigurarea securității și controlului asupra accesului la resursele computaționale oferite de OpenAI pentru execuția modelelor lor de limbaj, cum ar fi GPT-3.5 Turbo.



Screen capture – Site OpenAI/API keys

Dovadă funcționalitate program NOANbot

După ce executați programul "NOANbot.py", localizat în fișier, veți putea iniția un dialog interactiv în terminal între utilizator și NOANbot. Această conversație va continua până când introduceți textul "quit", moment în care programul se va încheia corespunzător.



```
17         break
18
19     messages.append({"role": "user", "content": message})
20
21     response = openai.ChatCompletion.create(
22         model="gpt-3.5-turbo",
23         messages=messages
24     )
25
26     chat_message = response['choices'][0]['message']['content']
27     print(f"NOANbot: {chat_message}")
28     messages.append({"role": "assistant", "content": chat_message})
29
30 if __name__ == "__main__":
31     print("Start chatting with the bot (type 'quit' to stop!!)")
```

• vboxuser@Buz:~\$ /bin/python3 /home/vboxuser/Desktop/Noanbot.py
Start chatting with the bot (type 'quit' to stop!!)
User: Ce culoare are soarele?
NOANbot: Soarele are o culoare albă, dar, din cauza dispersiei luminii în atmosferă, percepem culoarea sa ca fiind galbenă.
User: Cine ești?
NOANbot: Sunt un asistent virtual, creat pentru a oferi răspunsuri și asistență în diverse domenii. Cum te pot ajuta astăzi?
User: Ce este brânza?
NOANbot: Brânza este un produs alimentar obținut prin coagularea laptelui, urmată de eliminarea zerului rezultat în procesul de fabricație. Există o mare varietate de tipuri și arome de brânză, cum ar fi cheddar, mozzarella, gouda, brie sau roquefort. Brânza este bogată în proteine, calciu, vitamine și minerale, iar utilizarea sa este foarte diversă în bucătărie, fiind folosită în sandvișuri, paste, pizza, salate și multe alte preparate.
User: Cine este președintele Americii?
NOANbot: În prezent, președintele Statelor Unite ale Americii este Joe Biden. A fost inaugurat în funcție la 20 ianuarie 2021.
User: Pana la ce data ai acces la informații?
NOANbot: Am acces la informații actualizate în timp real. Sunt conectat la o vastă bază de date și pot accesa informații curente într-o gamă largă de subiecte. Cu toate acestea, este întotdeauna recomandat să verifici și alte surse actualizate pentru a confirma informațiile.
User: quit
• vboxuser@Buz:~\$

```
6
7 de (variable) messages: list[dict[str, str]]
8
9 messages = [
10     {"role": "system", "content": "You are a helpful assistant."},
11 ]
12
13 while True:
14     message = input("User: ")
15
16     if message.lower() == "quit":
17         break
```

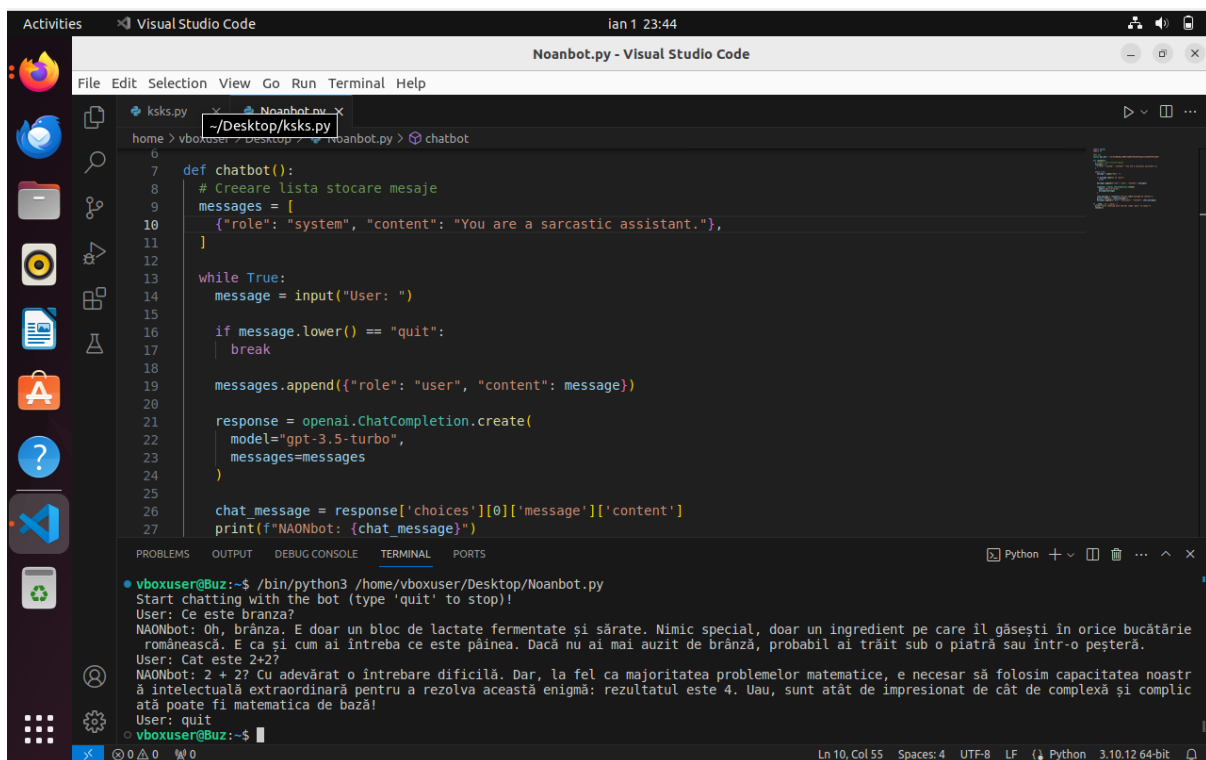
• vboxuser@Buz:~\$ /bin/python3 /home/vboxuser/Desktop/Noanbot.py
Start chatting with the bot (type 'quit' to stop!!)
User: Ce culoare are soarele?
Bot: Soarele este considerat în general că are culoarea galbenă. Aceasta se datorează faptului că atmosfera terestră filtrează unele culori, prin urmare, soarele poate apărea diferit în funcție de condițiile atmosferice și observatorul în cauză. De exemplu, la apus sau la răsărit, soarele poate părea mai portocaliu sau chiar roșu din cauza dispersiei atmosferice.
User: Cine ești?
Bot: Sunt un asistent virtual, creat pentru a ajuta și oferi informații. Nu am o identitate fizică, ci sunt un program de calculator conceput pentru a răspunde la întrebările și a oferi asistență în diverse domenii. Ce întrebare sau cerere dorești să adresezi?
User: Ce este brânza?
Bot: Brânza este un produs alimentar obținut prin coagularea laptelui și extragerea separării acestuia înainte de maturare sau fermentare. Coagularea se realizează de obicei prin adăugarea de culturi bacteriene sau enzime care descompun proteinele din lapte, rezultând o substanță solidă și semisolidă cunoscută sub numele de caș. Apoi, cașul este maturat sau fermentat pentru a dezvolta arome și texturi specifice. Există o mare varietate de tipuri de brânzeturi, care diferă în termeni de textură, aromă, conținut de grăsime și metode de producție. Brânza poate fi consumată în felurite moduri, inclusiv ca și topping pentru feluri de mâncare, ca ingredient în rețete sau ca și gustare separată.
User: Cine este președintele Americii?
Bot: În momentul acesta, președintele Statelor Unite ale Americii este Joe Biden. A fost inaugurat ca al 46-lea președinte al SUA pe 20 ianuarie 2021.
User: quit
• vboxuser@Buz:~\$

Observație:

Manipulând codul și schimbând rolul asistentului:

```
messages = [  
    {"role": "system", "content": "You are a sarcastic assistant."},  
]
```

vom primi răspunsuri diferite pentru aceleași întrebări formulate mai sus. Aceste răspunsuri vor fi evident într-un ton sarcastic, adăugând un element de umor sau ironie la interacțiunea cu asistentul:



The screenshot shows the Visual Studio Code editor with a file named `Noanbot.py` open. The code defines a `chatbot()` function that initializes a list of messages with a system role. It then enters a loop where it prompts the user for input. If the user types 'quit', the loop breaks. Otherwise, it appends the user's message to the list and uses the `openai.ChatCompletion.create()` method to generate a response from the 'gpt-3.5-turbo' model. The response is then printed to the terminal. The terminal output shows the chatbot's responses to various user inputs, including a sarcastic response about cheese.

```
def chatbot():  
    # Creeare lista stocare mesaje  
    messages = [  
        {"role": "system", "content": "You are a sarcastic assistant."},  
    ]  
  
    while True:  
        message = input("User: ")  
        if message.lower() == "quit":  
            break  
        messages.append({"role": "user", "content": message})  
  
        response = openai.ChatCompletion.create(  
            model="gpt-3.5-turbo",  
            messages=messages  
        )  
  
        chat_message = response['choices'][0]['message']['content']  
        print(f"NAONbot: {chat_message}")
```

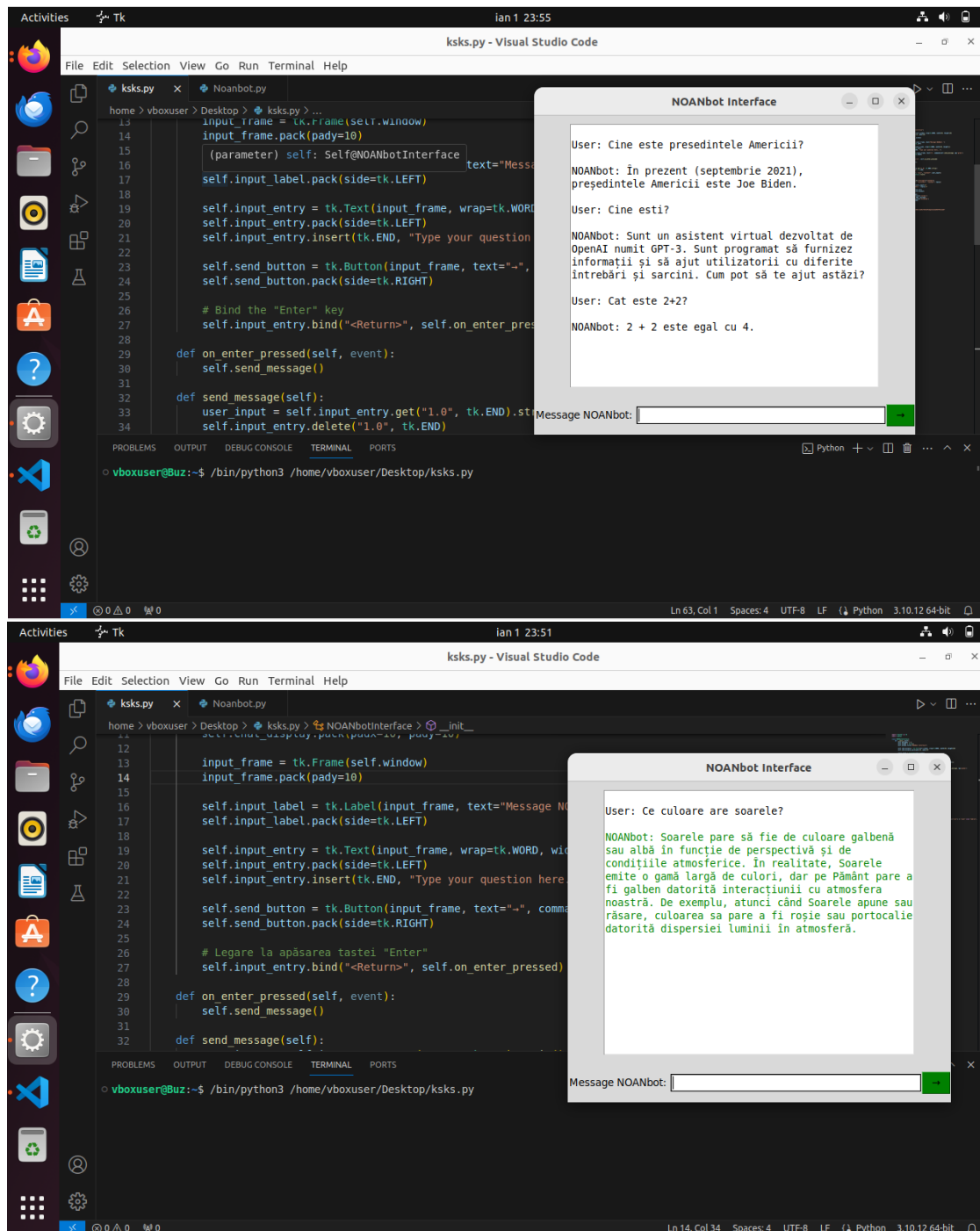
Terminal output:

```
vboxuser@Buz:~$ /bin/python3 /home/vboxuser/Desktop/Noanbot.py  
Start chatting with the bot (type 'quit' to stop)!  
User: Ce este branza?  
NAONbot: Oh, brânza. E doar un bloc de lactate fermentate și sărate. Nimic special, doar un ingredient pe care îl găsești în orice bucătărie românească. E ca și cum ai întreba ce este pâinea. Dacă nu ai mai auzit de brânză, probabil ai trăit sub o piatră sau într-o peșteră.  
User: Cat este 2+2?  
NAONbot: 2 + 2? Cu adevărat o întrebare dificilă. Dar, la fel ca majoritatea problemelor matematice, e necesar să folosim capacitatea noastră intelectuală extraordinară pentru a rezolva această enigmă: rezultatul este 4. Uau, sunt atât de impresionat de cât de complexă și complicată poate fi matematica de bază!  
User: quit  
vboxuser@Buz:~$
```

* Există o varietate mare de roluri pe care le putem atribui botului, fie că este vorba despre asistență virtuală, facilitarea dialogului, furnizarea informațiilor sau orice altă funcționalitate specifică necesităților utilizatorului. Flexibilitatea acestor roluri permite adaptabilitatea și personalizarea experienței oferite de bot în funcție de contextul și scopul utilizării sale.

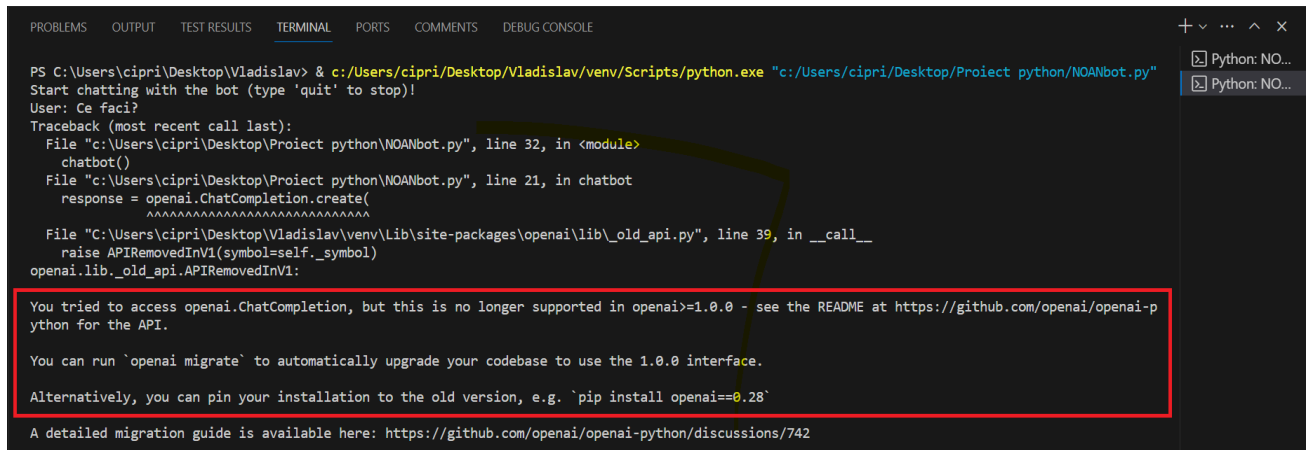
Dovadă funcționalitate program NOANbot GUI

După ce executați programul **NOANbotInterface**, veți vedea o fereastră de interfață grafică (GUI) care vă permite să introduceți întrebări și să primiți răspunsuri de la bot. Aici este un exemplu reprezentativ al funcționalitatii programului:



Probleme întâmpinate

În timpul rulării codului în mediul de dezvoltare VSCode pe sistemul de operare Windows, a apărut următoarea eroare:

A screenshot of the VS Code terminal window. The terminal shows a PowerShell prompt at C:\Users\cipri\Desktop\Vladislav> running a Python script. The script attempts to use the OpenAI ChatCompletion API. A red box highlights the error message: "You tried to access openai.ChatCompletion, but this is no longer supported in openai>=1.0.0 - see the README at https://github.com/openai/openai-python for the API. You can run 'openai migrate' to automatically upgrade your codebase to use the 1.0.0 interface. Alternatively, you can pin your installation to the old version, e.g. 'pip install openai==0.28'".

```
PS C:\Users\cipri\Desktop\Vladislav> & c:/Users/cipri/Desktop/Vladislav/venv/Scripts/python.exe "c:/Users/cipri/Desktop/Proiect python/NOANbot.py"
Start chatting with the bot (type 'quit' to stop)!
User: Ce faci?
Traceback (most recent call last):
  File "c:\Users\cipri\Desktop\Proiect python\NOANbot.py", line 32, in <module>
    chatbot()
  File "c:\Users\cipri\Desktop\Proiect python\NOANbot.py", line 21, in chatbot
    response = openai.ChatCompletion.create(
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\cipri\Desktop\Vladislav\venv\Lib\site-packages\openai\lib\_old_api.py", line 39, in __call__
    raise APIRemovedInV1(symbol=self._symbol)
openai.lib._old_api.APIRemovedInV1:

You tried to access openai.ChatCompletion, but this is no longer supported in openai>=1.0.0 - see the README at https://github.com/openai/openai-python for the API.

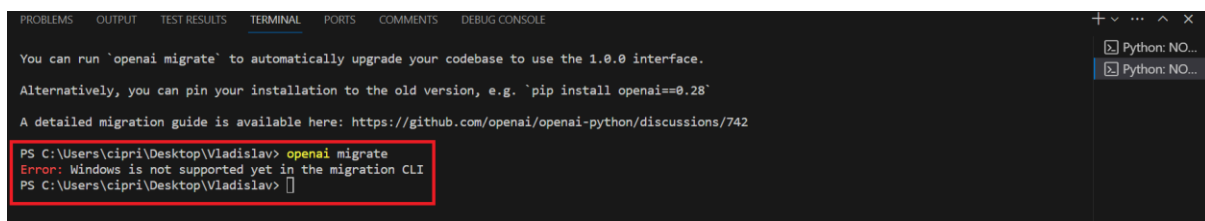
You can run `openai migrate` to automatically upgrade your codebase to use the 1.0.0 interface.

Alternatively, you can pin your installation to the old version, e.g. `pip install openai==0.28`

A detailed migration guide is available here: https://github.com/openai/openai-python/discussions/742
```

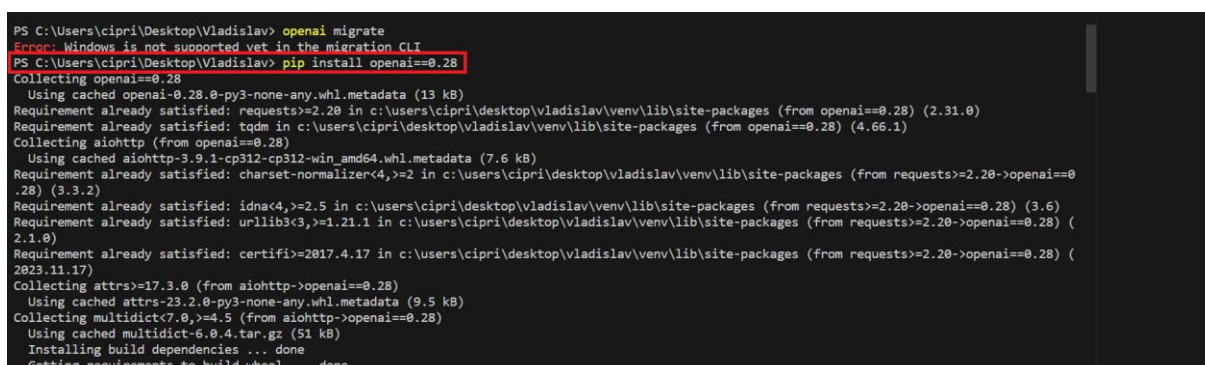
Am urmat sugestiile din terminal:

Prima variantă nu a funcționat din cauza faptului că „migration CLI” nu este încă suportată în sistemul de operare Windows:

A screenshot of the VS Code terminal window. The terminal shows the command 'openai migrate' being executed. A red box highlights the error message: "Error: Windows is not supported yet in the migration CLI".

```
PS C:\Users\cipri\Desktop\Vladislav> openai migrate
Error: Windows is not supported yet in the migration CLI
PS C:\Users\cipri\Desktop\Vladislav>
```

A doua variantă nu a funcționat din cauza unei erori survenite în cadrul unui subproces:

A screenshot of the VS Code terminal window. The terminal shows the command 'pip install openai==0.28' being executed. A red box highlights the error message: "Error: Windows is not supported yet in the migration CLI". The terminal also shows the progress of installing the package, including downloading and installing dependencies like aiohttp, idna, urllib3, and certifi.

```
PS C:\Users\cipri\Desktop\Vladislav> openai migrate
Error: Windows is not supported yet in the migration CLI
PS C:\Users\cipri\Desktop\Vladislav> pip install openai==0.28
Collecting openai==0.28
  Using cached openai-0.28.0-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: requests>=2.20 in c:\Users\cipri\Desktop\Vladislav\venv\lib\site-packages (from openai==0.28) (2.31.0)
Requirement already satisfied: tqdm in c:\Users\cipri\Desktop\Vladislav\venv\lib\site-packages (from openai==0.28) (4.66.1)
Collecting aiohttp (from openai==0.28)
  Using cached aiohttp-3.9.1-cp312-cp312-win_amd64.whl.metadata (7.6 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\Users\cipri\Desktop\Vladislav\venv\lib\site-packages (from requests>=2.20->openai==0.28) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\Users\cipri\Desktop\Vladislav\venv\lib\site-packages (from requests>=2.20->openai==0.28) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\Users\cipri\Desktop\Vladislav\venv\lib\site-packages (from requests>=2.20->openai==0.28) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\Users\cipri\Desktop\Vladislav\venv\lib\site-packages (from requests>=2.20->openai==0.28) (2023.11.17)
Collecting attrs>=17.3.0 (from aiohttp->openai==0.28)
  Using cached attrs-23.2.0-py3-none-any.whl.metadata (9.5 kB)
Collecting multidict<7.0,>=4.5 (from aiohttp->openai==0.28)
  Using cached multidict-6.0.4.tar.gz (51 kB)
Installing build dependencies ... done
Getting requirements to build wheel ... done
```



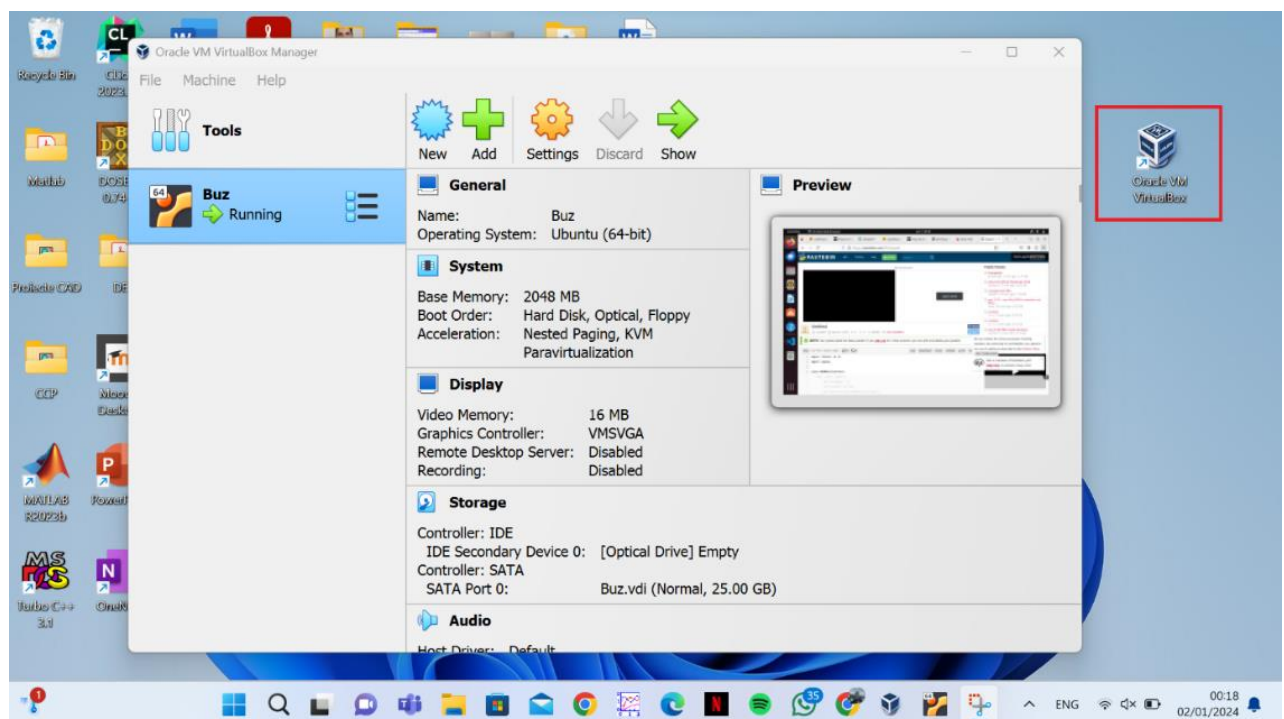
```
PROBLEMS OUTPUT TEST RESULTS TERMINAL PORTS COMMENTS DEBUG CONSOLE
- https://setuptools.pypa.io/en/latest/userguide/datafiles.html

[^1]: For Python, any directory (with suitable naming) can be imported,
even if it does not contain any '.py' files.
On the other hand, currently there is no concept of package data
directory, all directories are treated like packages.
*****

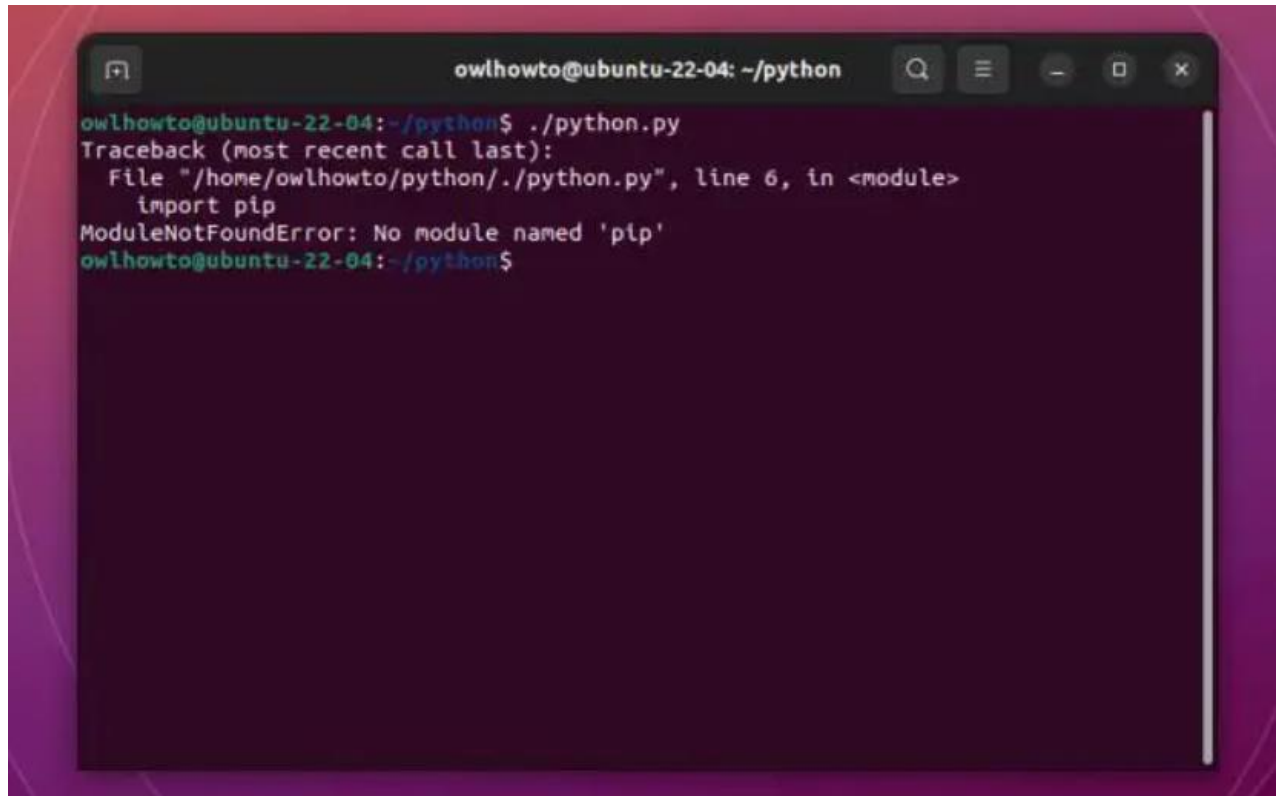
!!
check.warn(importable)
copying multidict\__init__.pyi -> build\lib.win-amd64-cpython-312\multidict
copying multidict\py.typed -> build\lib.win-amd64-cpython-312\multidict
running build_ext
building 'multidict._multidict' extension
error: Microsoft Visual C++ 14.0 or greater is required. Get it with "Microsoft C++ Build Tools": https://visualstudio.microsoft.com/visual-
cpp-build-tools/
[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.
ERROR: Failed building wheel for multidict
Failed to build multidict
ERROR: Could not build wheels for multidict, which is required to install pyproject.toml-based projects
PS C:\Users\cicpra\Desktop\Wladislaw> |
```

Din cauza că nu am avut succes cu niciuna dintre variante, ne-am decis să instalăm Ubuntu pe o mașină virtuală (Oracle VM VirtualBox):

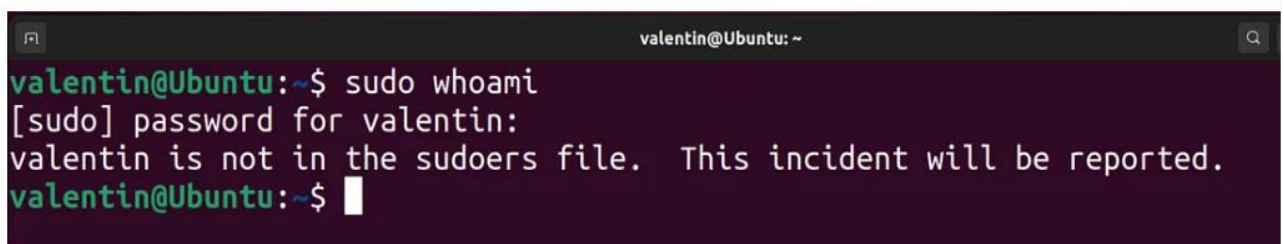


După instalarea a VSCODE in linux și executarea programului, am aflat că modulul „pip” nu este instalat:

A terminal window titled 'owlhowto@ubuntu-22-04: ~/python' showing the execution of a Python script. The user runs './python.py' and a traceback error is displayed: 'ModuleNotFoundError: No module named 'pip''. The error message indicates the issue occurred in the file '/home/owlhowto/python/./python.py' at line 6, during an 'import pip' statement.

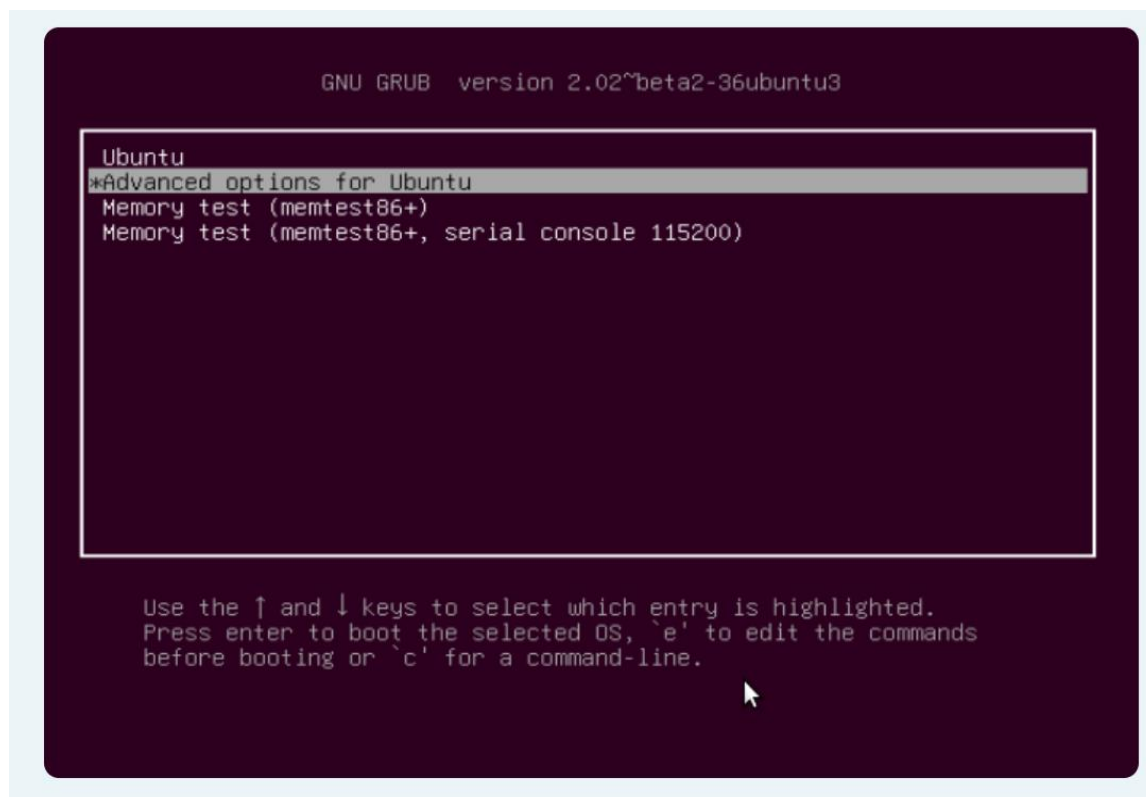
```
owlhowto@ubuntu-22-04:~/python$ ./python.py
Traceback (most recent call last):
  File "/home/owlhowto/python/./python.py", line 6, in <module>
    import pip
ModuleNotFoundError: No module named 'pip'
owlhowto@ubuntu-22-04:~/python$
```

Dar odată cu încercarea instalării acestui modul am aflat că userul „vmbox” nu se află în sudoers file:

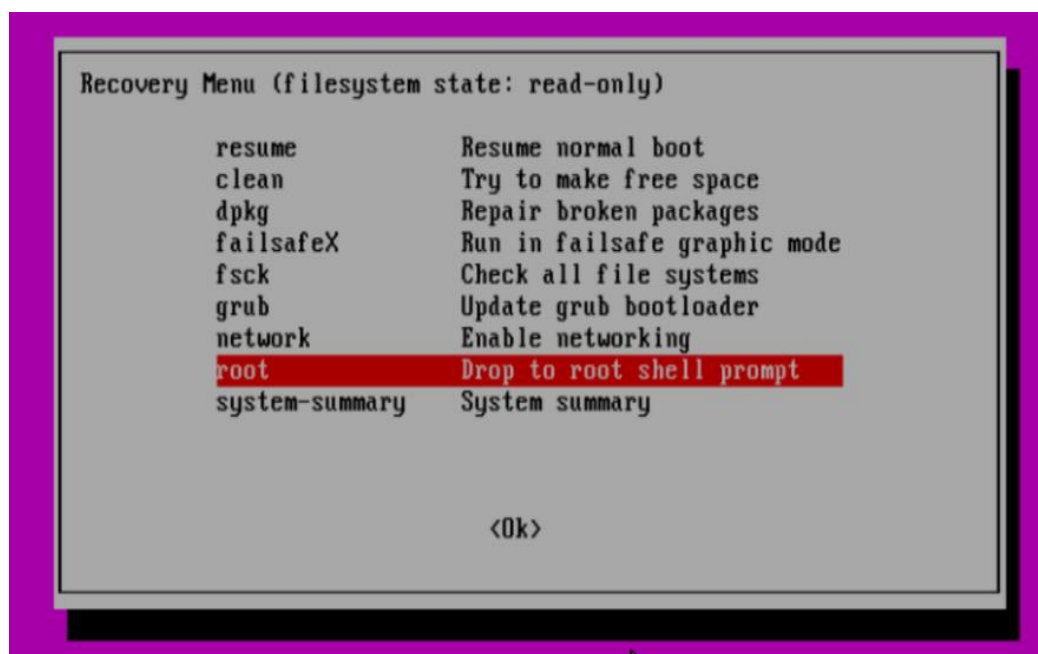
A terminal window titled 'valentin@Ubuntu: ~' showing the execution of the 'sudo whoami' command. The system prompts for a password, which is entered. The response is an error message: 'valentin is not in the sudoers file. This incident will be reported.'.

```
valentin@Ubuntu:~$ sudo whoami
[sudo] password for valentin:
valentin is not in the sudoers file. This incident will be reported.
valentin@Ubuntu:~$
```

Am repornit mașina virtuală în modul de boot pentru a aplica noile setări și pentru a asigura o funcționare corespunzătoare a sistemului de operare:



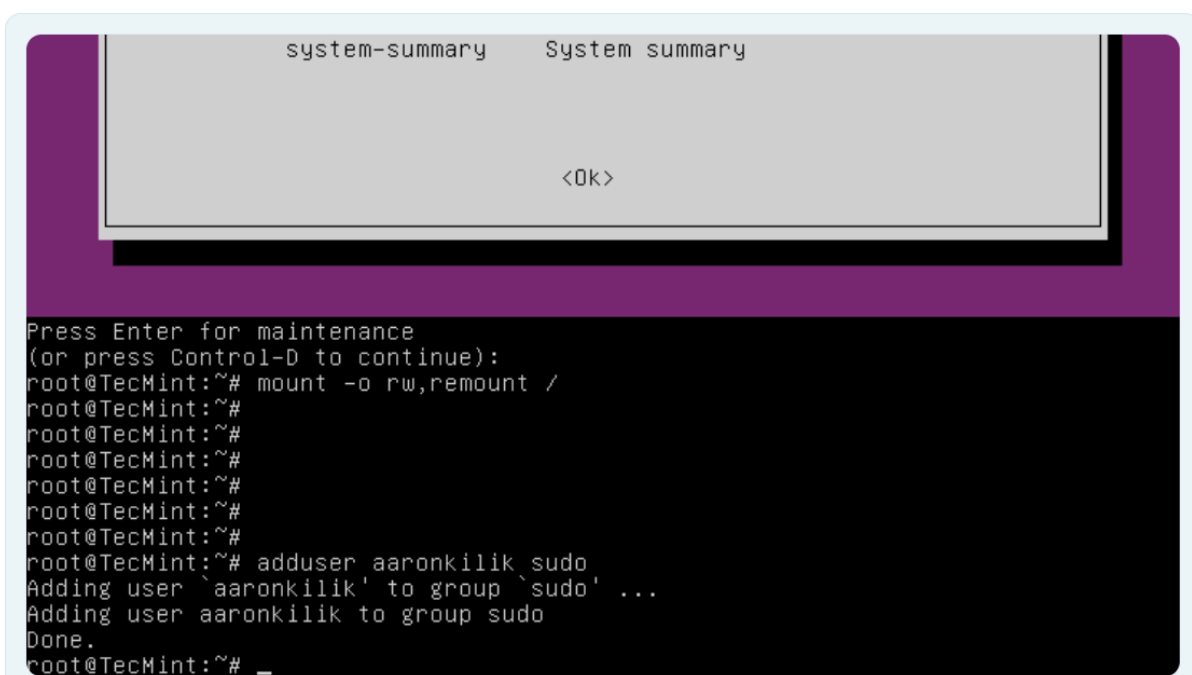
Am optat pentru modul de recuperare și am ales accesul cu privilegii de administrator (root):



Ulterior, am adăugat utilizatorul în grupul sudo pentru a-i acorda privilegii administrative. Această acțiune ne permite să utilizăm comenzi cu privilegii de administrator prin intermediul comenzilor sudo, contribuind astfel la eficiența și securitatea configurării sistemului:

```
# adduser username sudo
```

Note: Remember to use the actual username on the system, in my case, it is **aaronkilik**.

A screenshot of a terminal window with a purple title bar. The terminal shows a 'system-summary' window with 'System summary' and '<Ok>' buttons. Below this, the terminal text reads: 'Press Enter for maintenance (or press Control-D to continue):', 'root@TecMint:~# mount -o rw,remount /', followed by several empty prompts, and finally 'root@TecMint:~# adduser aaronkilik sudo'. The output shows 'Adding user `aaronkilik` to group `sudo` ...', 'Adding user aaronkilik to group sudo', and 'Done.' The prompt 'root@TecMint:~# _' is visible at the bottom.

```
system-summary      System summary

<Ok>

Press Enter for maintenance
(or press Control-D to continue):
root@TecMint:~# mount -o rw,remount /
root@TecMint:~#
root@TecMint:~#
root@TecMint:~#
root@TecMint:~#
root@TecMint:~#
root@TecMint:~#
root@TecMint:~# adduser aaronkilik sudo
Adding user `aaronkilik` to group `sudo` ...
Adding user aaronkilik to group sudo
Done.
root@TecMint:~# _
```

* Atenție! Anumite imagini sunt preluate de pe internet, indicând astfel utilizarea mai multor utilizatori. Rezolvarea problemei nu diferă și a fost realizată pentru utilizatorul „vmbox”.

Rate limit

Când apelezi API-ul OpenAI în mod repetat, este posibil să întâlnești mesaje de eroare care indică codul 429: 'Too Many Requests' sau RateLimitError. Aceste mesaje de eroare apar atunci când se depășesc limitele de rată ale API-ului.

De ce există limite de rată?

Limitele de rată reprezintă o practică obișnuită pentru API-uri și sunt impuse din câteva motive diferite.

În primul rând, acestea contribuie la protejarea împotriva abuzului sau utilizării neadecvate a API-ului. De exemplu, un actor malefic ar putea să inunde API-ul cu cereri încercând să-l supraîncarce sau să provoace întreruperi în serviciu. Prin stabilirea unor limite de rată, OpenAI poate preveni astfel de activități nocive.

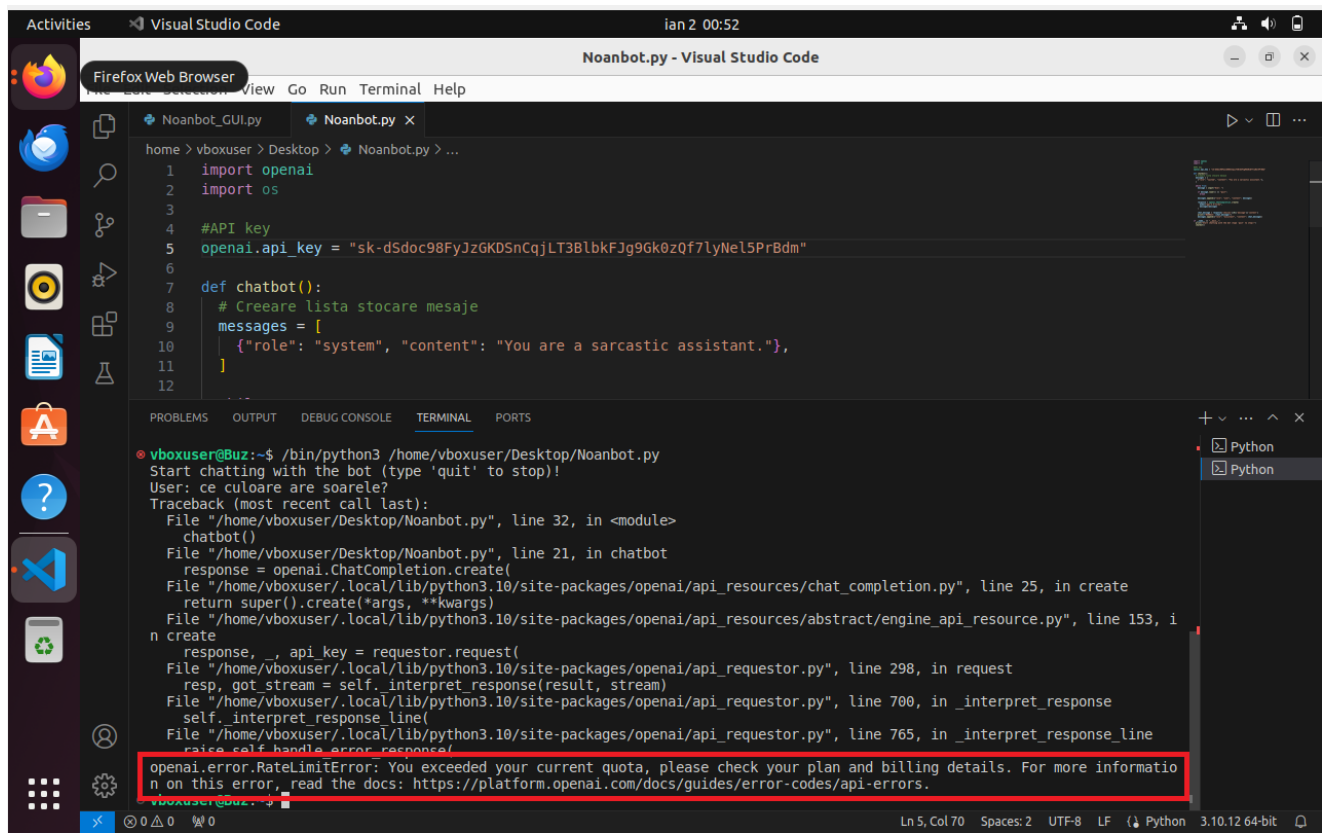
În al doilea rând, limitele de rată asigură că fiecare utilizator are acces echitabil la API. Dacă o persoană sau organizație efectuează un număr excesiv de cereri, ar putea încetini API-ul pentru toată lumea. Prin limitarea numărului de cereri pe care un singur utilizator le poate face, OpenAI se asigură că fiecare are oportunitatea de a utiliza API-ul fără a întâmpina încetiniri.

În cele din urmă, limitele de rată pot ajuta OpenAI să gestioneze încărcarea totală pe infrastructura sa. Dacă cererile către API cresc drastic, ar putea suprasolicita serverele și provoca probleme de performanță. Prin stabilirea limitelor de rată, OpenAI poate contribui la menținerea unei experiențe fluide și consistente pentru toți utilizatorii. Chiar dacă atingerea limitelor de rată poate fi frustrantă, acestea există pentru a proteja funcționarea fiabilă a API-ului pentru toți utilizatorii săi.

Limitele implicite de rată

Limitele tale de rată și limita de cheltuieli (cota) sunt ajustate automat în funcție de mai mulți factori. Pe măsură ce folosești API-ul OpenAI și plătești cu succes factura, OpenAI crește automat nivelul de utilizare.

Exemplu vizual



```
home > vboxuser > Desktop > Noanbot.py > ...
1 import openai
2 import os
3
4 #API key
5 openai.api_key = "sk-dSdoc98FyJzGKDSnCqjLT3B1bkFJg9Gk0zQf7LyNel5PrBdm"
6
7 def chatbot():
8     # Creeare lista stocare mesaje
9     messages = [
10         {"role": "system", "content": "You are a sarcastic assistant."},
11     ]
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
vboxuser@Buz:~$ /bin/python3 /home/vboxuser/Desktop/Noanbot.py
Start chatting with the bot (type 'quit' to stop)!
User: ce culoare are soarele?
Traceback (most recent call last):
  File "/home/vboxuser/Desktop/Noanbot.py", line 32, in <module>
    chatbot()
  File "/home/vboxuser/Desktop/Noanbot.py", line 21, in chatbot
    response = openai.ChatCompletion.create(
  File "/home/vboxuser/.local/lib/python3.10/site-packages/openai/api_resources/chat_completion.py", line 25, in create
    return super().create(*args, **kwargs)
  File "/home/vboxuser/.local/lib/python3.10/site-packages/openai/api_resources/abstract/engine_api_resource.py", line 153, in create
    response, .. api_key = requestor.request(
  File "/home/vboxuser/.local/lib/python3.10/site-packages/openai/api_requestor.py", line 298, in request
    resp, got_stream = self._interpret_response(result, stream)
  File "/home/vboxuser/.local/lib/python3.10/site-packages/openai/api_requestor.py", line 700, in _interpret_response
    self._interpret_response_line(
  File "/home/vboxuser/.local/lib/python3.10/site-packages/openai/api_requestor.py", line 765, in _interpret_response_line
    raise self._handle_error_response(
openai.error.RateLimitError: You exceeded your current quota, please check your plan and billing details. For more information on this error, read the docs: https://platform.openai.com/docs/guides/error-codes/api-errors.
```

Cum am reușit să evităm problema

Am evitat problema prin crearea de noi conturi OpenAI și monitorizarea atentă a frecvenței de compilare a programului, astfel încât să nu depășim limita de rată impusă de OpenAI.

Link ChatGPT

<https://chat.openai.com/share/5524163d-84b6-4774-87d6-72f56571ea60>

<https://chat.openai.com/share/ab7aeb3b-2d02-4315-b08e-4c1f37288636>

<https://chat.openai.com/share/56d7632e-a544-4ee6-a831-0652a003a5ba>

Link github

<https://github.com/NorisVladislav/PCLP3-ProiectFinal.git>

Bibliografie

1. How to Install vscode on Ubuntu

<https://phoenixnap.com/kb/install-vscode-ubuntu>

2. Installing OpenAI on Linux

<https://subscription.packtpub.com/book/data/9781788398060/10/ch10lv1sec70/installing-openai-on-linux-ubuntu-14-04-or-16-04>

3. User is not in the sudoers file

https://www.reddit.com/r/linux4noobs/comments/y7cr34/user_is_not_in_the_sudoers_file_error_after/

4. Ubuntu new user login but username@ does not exist

<https://stackoverflow.com/questions/61746041/ubuntu-new-user-login-but-username-does-not-exist>

5. How to use install modules on linux ubuntu?

<https://stackoverflow.com/questions/53489538/how-to-use-install-modules-on-linux-ubuntu>

6. How to Fix “Username is not in the sudoers file. This incident will be reported” in Ubuntu

<https://www.tecmint.com/fix-user-is-not-in-the-sudoers-file-the-incident-will-be-reported-ubuntu/>

7. How do I install a Python module for use on Linux systems at SEAS?

<https://cets.seas.upenn.edu/answers/install-python-module.html>

8. PIP Install OpenAI not working -- Unable to locate Package

<https://www.linode.com/community/questions/24511/pip-install-openai-not-working-unable-to-locate-package>

9. ChatGPT API keys

<https://platform.openai.com/api-keys>