

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.

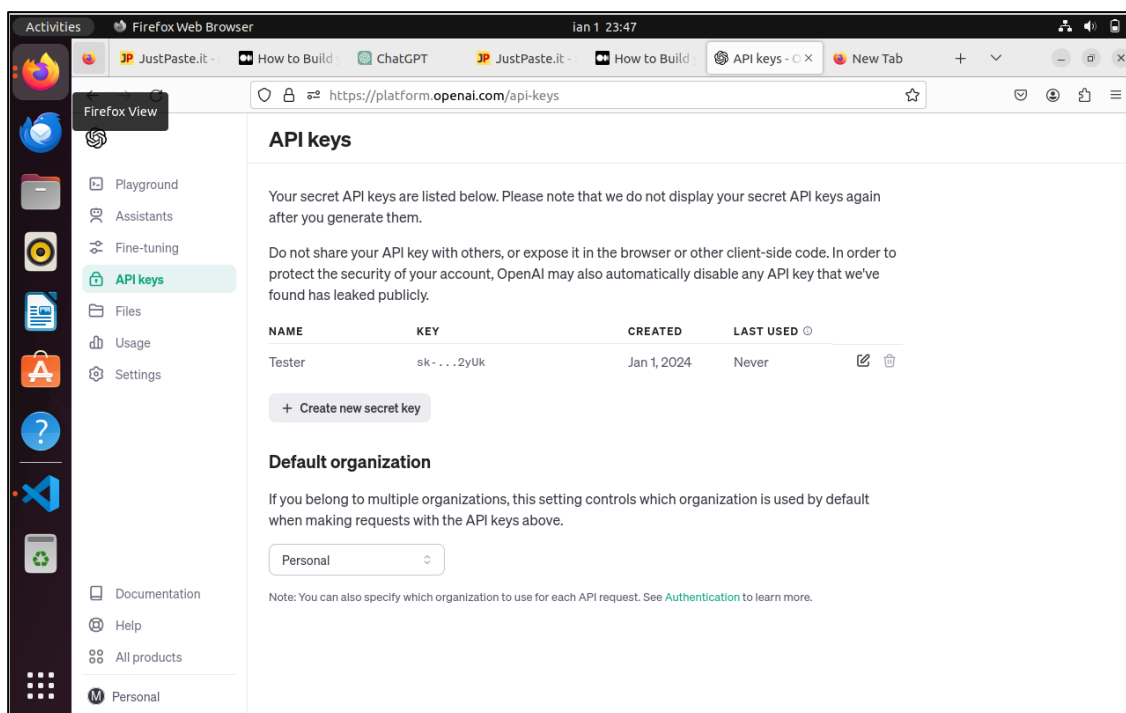
Acest cod este un exemplu de bază și poate fi extins pentru cazuri de utilizare mai complexe sau integrate în alte aplicații.

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.

The image displays two screenshots of a Visual Studio Code editor window, showing the development of a Python chatbot named 'Noanbot.py'.

Top Screenshot:

- The file explorer on the left shows the project structure with 'ksks.py' and 'Noanbot.py'.
- The editor displays the 'Noanbot.py' file, which contains a simple chatbot logic. It uses a loop to handle user input and a single message response.
- The terminal at the bottom shows the execution of the script using 'python3 /home/vboxuser/Desktop/Noanbot.py'. The chatbot responds to user input with a single message: 'You are a helpful assistant.'

Bottom Screenshot:

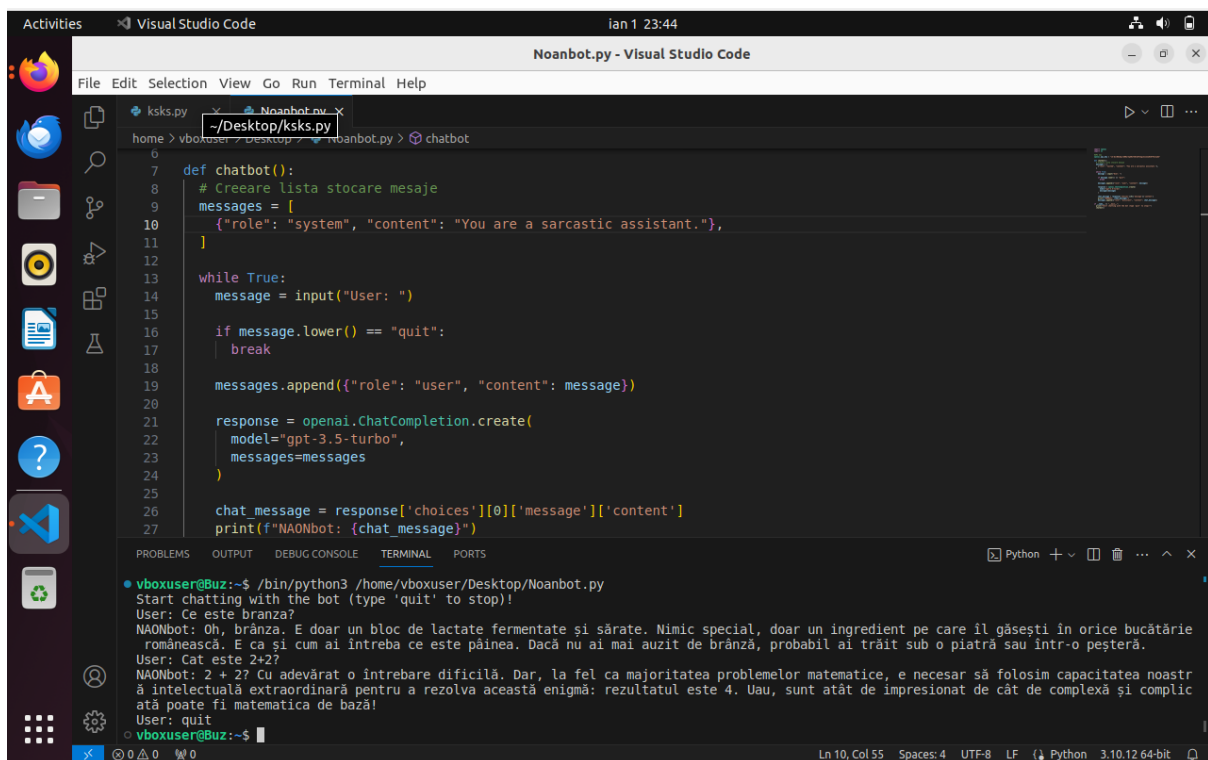
- The file explorer on the left shows the project structure with 'ksks.py' and 'Noanbot.py'.
- The editor displays the 'Noanbot.py' file, which has been updated with more complex logic. It uses a loop to handle user input and a more detailed response based on the user's input.
- The terminal at the bottom shows the execution of the script using 'python3 /home/vboxuser/Desktop/Noanbot.py'. The chatbot responds to user input with a more detailed message: 'You are a helpful assistant. I am a chatbot. I can help you with your questions. I am a chatbot. I can help you with your questions. I am a chatbot. I can help you with your questions.'

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 `while True` loop where it prompts the user for input. If the user enters '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.

```
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}")
```

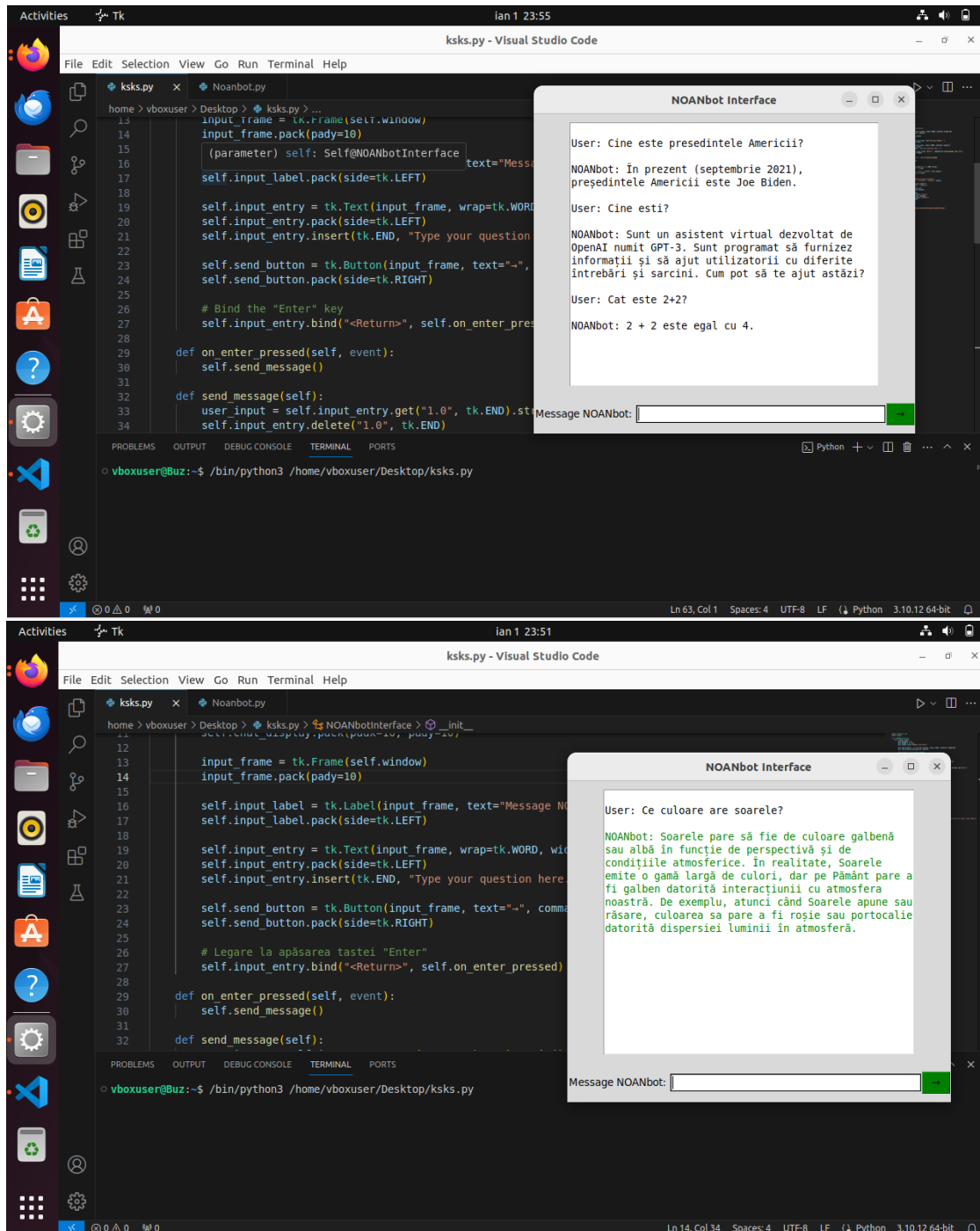
The terminal output shows the following interaction:

```
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 funcționalității programului:



Probleme întâmpinate

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

```
PROBLEMS OUTPUT TEST RESULTS TERMINAL PORTS COMMENTS DEBUG CONSOLE
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-p
ython 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:

```
PROBLEMS OUTPUT TEST RESULTS TERMINAL PORTS COMMENTS DEBUG CONSOLE
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
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:

```
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
```



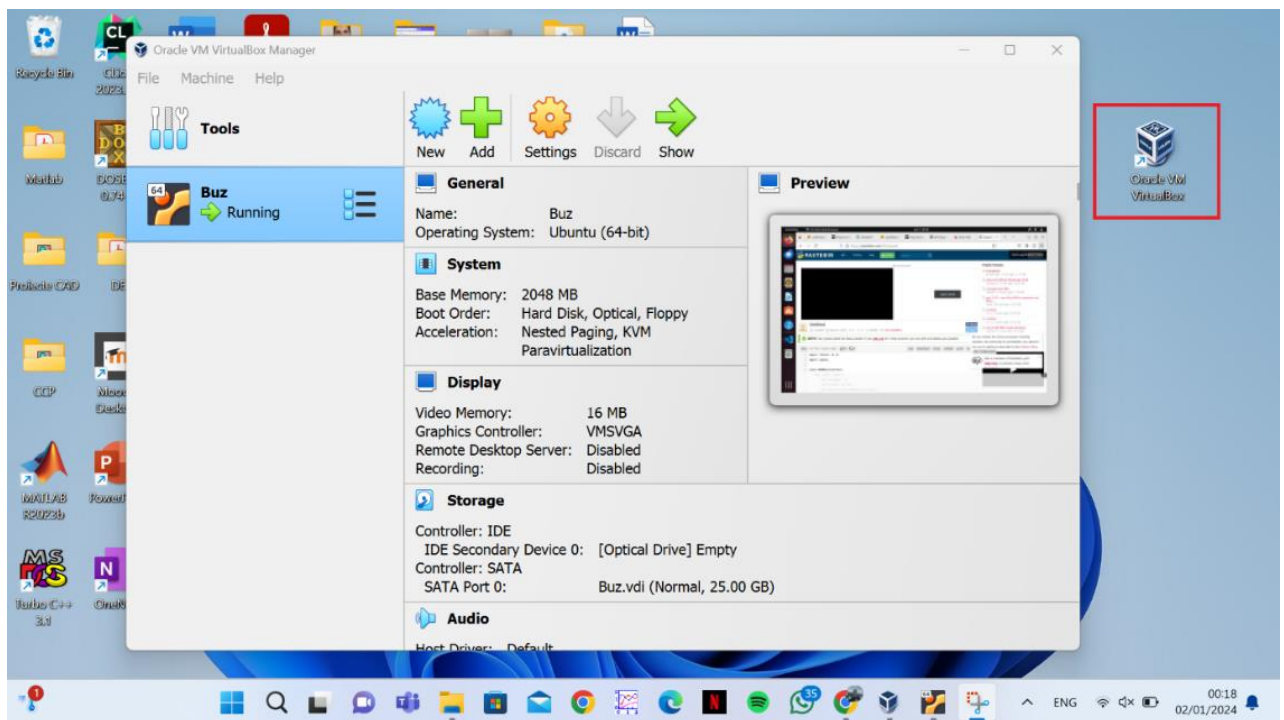
```
PROBLEMS OUTPUT TEST RESULTS TERMINAL PORTS COMMENTS DEBUG CONSOLE
- https://setuptools.pypa.io/en/latest/userguide/datafiles.html

[!]: 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.
*****

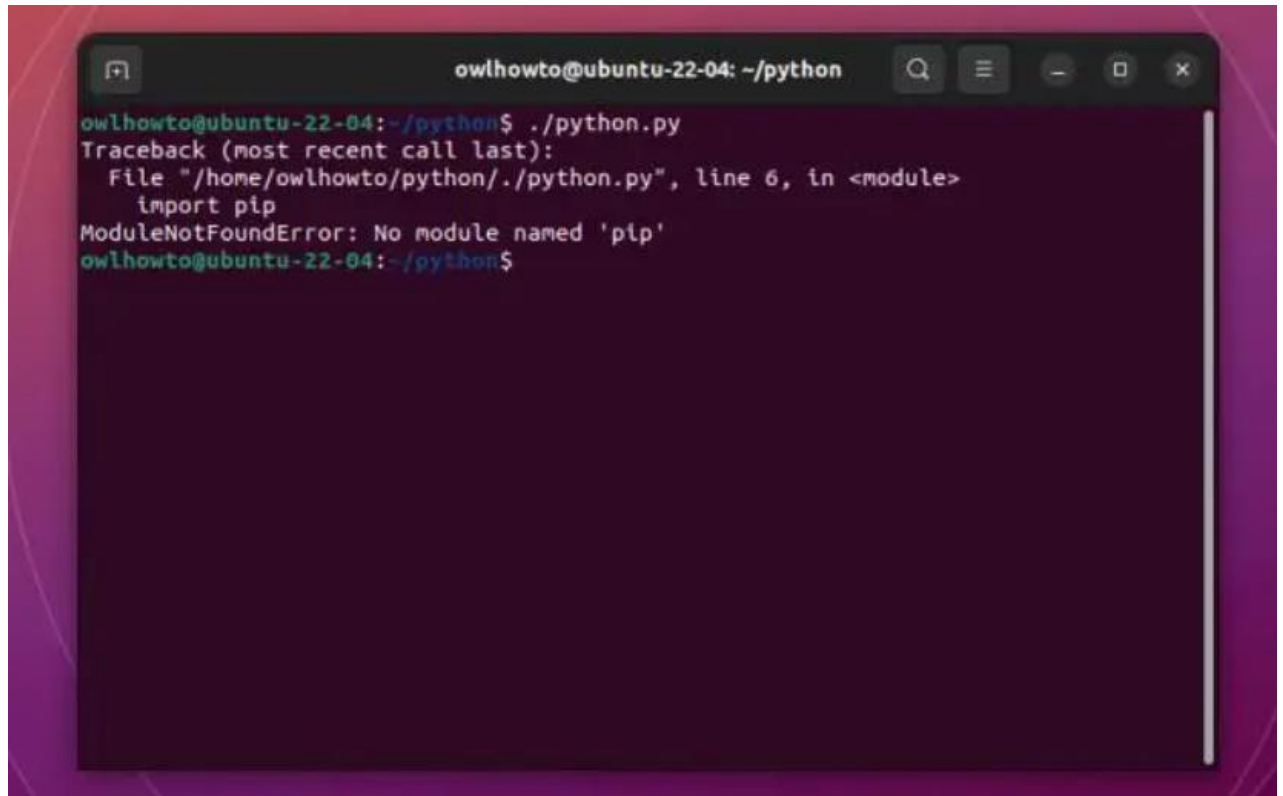
!!
check.warn(importable)
copying multidict\__init__.pyi -> build\lib.win-amd64-cpython-312\multidict
copying multidict\py.typed -> build\lib.win-amd64-cpython-312\multidict
running build_ext
building 'multidict._multidict' extension
error: Microsoft Visual C++ 14.0 or greater is required. Get it with "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\cicpr\Desktop\Widiaslav> |
```

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

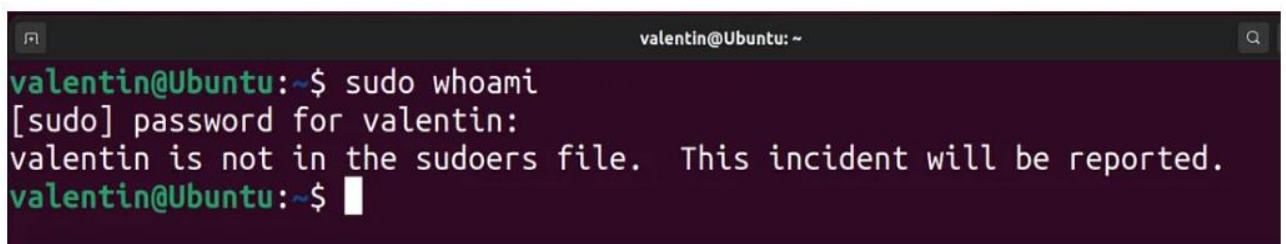


După instalarea a VSCODE in linux și executarea programului, am aflat că modulul „pip” nu este instalat:

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: 'File "/home/owlhowto/python/./python.py", line 6, in <module> import pip' followed by 'ModuleNotFoundError: No module named 'pip''.

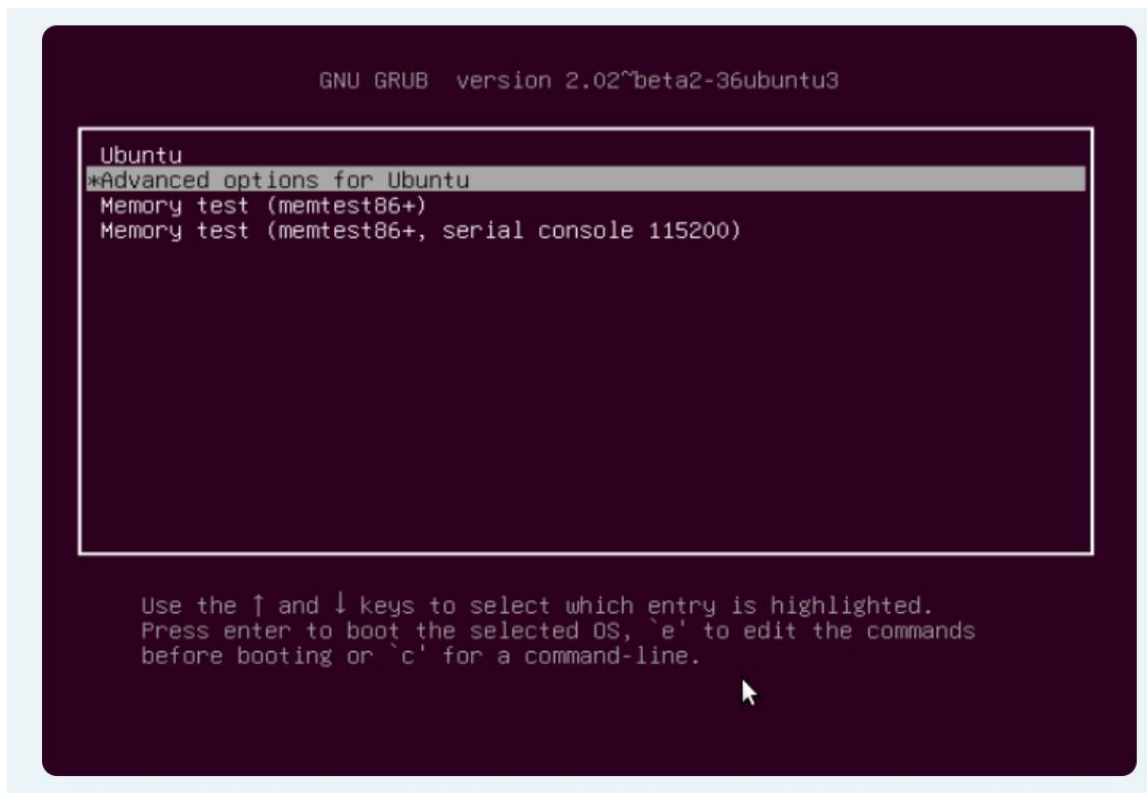
```
owlhowto@ubuntu-22-04:~/python$ ./python.py
Traceback (most recent call last):
  File "/home/owlhowto/python/./python.py", line 6, in <module>
    import pip
ModuleNotFoundError: No module named 'pip'
owlhowto@ubuntu-22-04:~/python$
```

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

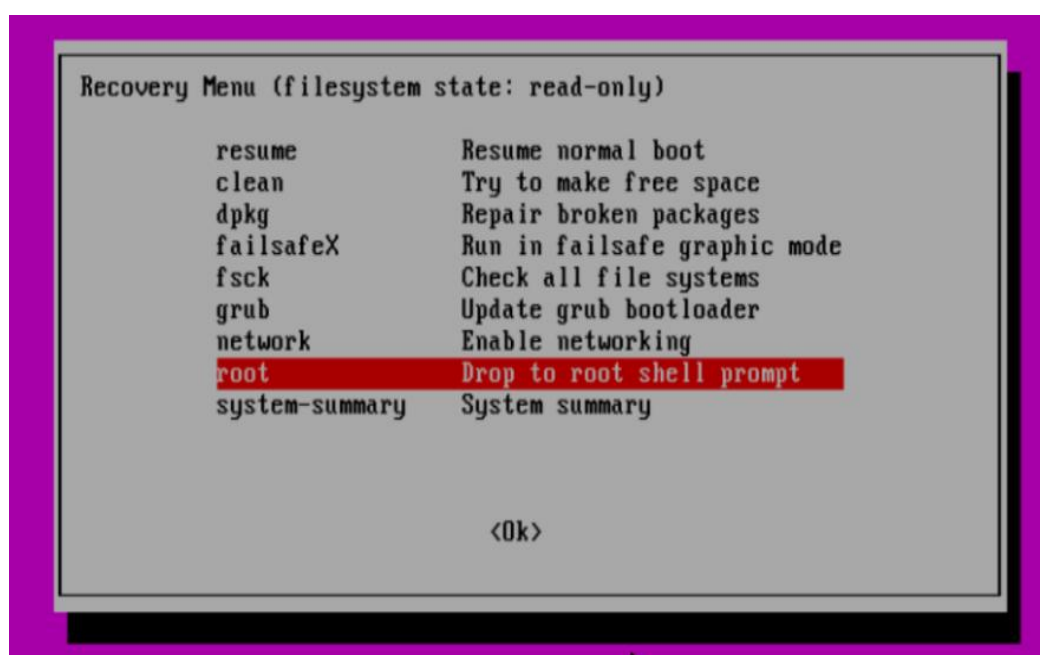
A terminal window titled 'valentin@Ubuntu: ~' showing the execution of the 'sudo whoami' command. The user is prompted for a password, but the command fails with the message: 'valentin is not in the sudoers file. This incident will be reported.'.

```
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:



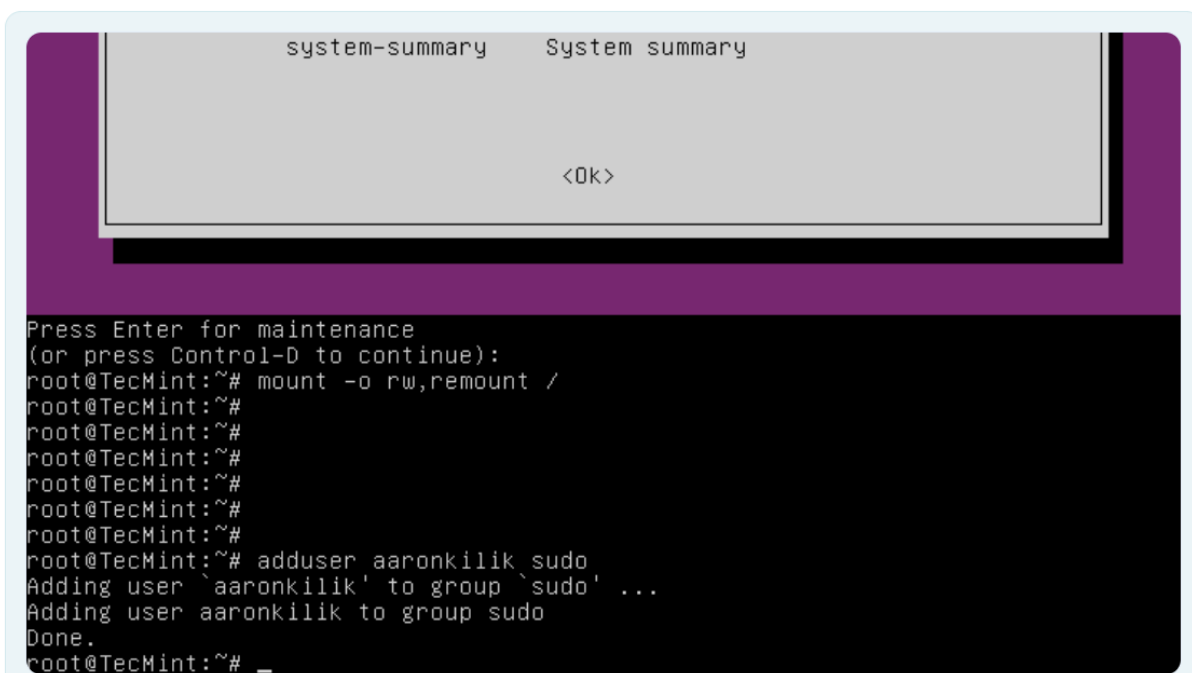
Am optat pentru modul de recuperare și am ales accesul cu privilegii de administrator (root):



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' dialog box at the top with the text 'System summary' and '<Ok>'. Below the dialog, the terminal text reads: 'Press Enter for maintenance (or press Control-D to continue):', followed by several 'root@TecMint:~#' prompts. The final command entered is 'adduser aaronkilik sudo', which results in the output: '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”.

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>