DEPARTMENT OF “BACHELOR OF COMPUTER APPLICATION”
NESS WADIA COLLEGE OF COMMERCE, PUNE
ACADEMIC YEAR 2024-2025
SEMESTER – VI

PROJECT REPORT ON
“Virtual Mouse, Keyboard, and Virtual Assistant Project”

SUBMITTED TO
SAVITRIBAI PHULE PUNE UNIVERSITY

SUBMITTED BY
Rajpal Singh Rathore
Shweta Verma

UNDER THE GUIDANCE OF
Prof. Seema Purandare
Prof. Ashwini Waghmare

ACKNOWLEDGEMENT

I would wish to thank all the people who have helped to make this project possible. A project for a student is an experience, in the course of which he/she realizes the real-world problems that one has to undergo during the development of any project. Hence without the help and guidance of our teachers, this project wouldn't have been successful. We are grateful to them for supporting us throughout the Design, Implementation and Evaluation phase of the project.

We extend our heartfelt gratitude to our guides, YouTube tutorials, and ChatGPT for their invaluable assistance throughout the development of this project. Their insightful guidance and support have been instrumental in navigating the various challenges encountered along the way.

We extend our heartfelt gratitude to Professor Seema Purandare & Ashwini Waghmare for her invaluable guidance, unwavering support, and insightful ideas throughout the duration of this project. Professor Purandare and Waghmare expertise, encouragement, and constructive feedback have been instrumental in shaping our understanding and approach towards developing our project.

Yours Sincerely,

Rajpal Singh & Shweta Verma

INDEX

Project: Virtual Mouse, Virtual Keyboard, and Virtual Assistant 'Buddy'

Project Overview

The project integrates three main components: a gesture-based Virtual Mouse, a Virtual Keyboard, and a voice-based Virtual Assistant named 'Buddy.' The project aims to provide a hands-free, AI-powered system capable of performing a wide variety of tasks using voice commands and gestures.

Problem Statement

In today’s fast-paced digital environment, users increasingly seek efficient, hands-free methods to interact with technology. Traditional input devices like the mouse and keyboard, although effective, often limit the user’s flexibility, accessibility, and productivity, especially when multitasking or performing repetitive tasks. Additionally, the complexity of day-to-day activities such as web browsing, opening applications, sending emails, and managing digital content can be time-consuming when done manually.

Moreover, the current generation of virtual assistants, while useful, lacks advanced integration and versatility in combining gesture-based control with voice-activated commands. Existing solutions, such as Siri, Alexa, and Google Assistant, are primarily voice-based and do not offer the customization or add-on functionalities required for more specialized tasks like Python code generation or AI-based image creation.

Need for the New System

As technology continues to evolve, users increasingly demand seamless, efficient, and hands-free interaction with digital devices. Traditional systems relying heavily on manual input through keyboards and mice no longer meet the needs of a dynamic, fast-paced, and multitasking digital environment. The need for the new system arises from the following challenges and limitations in current technologies:

1. Automation of Repetitive Tasks

- Routine tasks such as opening applications, sending messages, searching for content online, and managing emails are time-consuming when performed manually. Automating these tasks through a voice-controlled assistant simplifies the process, increasing productivity and reducing user effort.

2. Limitations of Current Virtual Assistants

- Existing virtual assistants (e.g., Siri, Alexa, Google Assistant) lack advanced capabilities such as gesture integration and specialized tools like code generation or AI-based image creation. Users need a more versatile assistant that can handle not only basic tasks but also cater to the specific needs of developers and creatives.

3. Enhanced Multitasking

- Users who need to manage multiple tasks simultaneously struggle with switching between different interfaces and tools. The new system provides a unified platform that allows gesture control for easy navigation, along with voice commands to manage complex tasks like sending emails or finding directions.

4. Customization and Scalability

- The current systems are often rigid and lack customization options. The new system allows for expansion and integration of features like Buddy CodeMaster (for code generation) and Buddy ImageMaster (for AI image generation), offering greater flexibility and the ability to scale with user needs.

5. Increased Productivity for Developers and Creatives

- For professionals such as developers and creatives, tools like Buddy CodeMaster (for generating Python code) and Buddy ImageMaster (for AI-based image creation) save time and effort. The new system integrates these functionalities into the virtual assistant, significantly enhancing workflow and productivity.

6. High Cost and Device Limitations of Existing Systems

- Assistants like Siri are restricted to Apple devices, which are costly and limit their accessibility to users who do not own Apple products. Similarly, systems like Alexa are often part of paid services and subscriptions, making them less appealing for users who need an affordable and open-source solution. The new system, 'Buddy,' is designed to be platform-independent, cost-effective, and free from restrictive device ecosystems.

Scope of the Project

Before diving into the scope of the project, it is important to understand the motivation behind its development. The integration of gesture-based controls and a versatile virtual assistant aims to enhance user interaction with digital devices by offering hands-free operation and task automation.

Objectives:

- Develop a hands-free, gesture-based input system (mouse/keyboard).
- Create a voice-activated assistant for multitasking and automation.
- Integrate AI for code/image generation and conversational support.

In-Scope Features:

- Gesture recognition using computer vision (OpenCV, MediaPipe).
- Voice recognition (Speech-to-Text libraries like speech_recognition, pyttsx3).
- Third-party API integrations (Google, Spotify, WhatsApp, AI Services).
- AI model integration (DeepSeek R-1 for chatbot, stable-defusion x1 for images).

Potential Use Cases:

- Automating routine tasks like opening applications and sending emails.
- Hands-free control of a computer interface using gestures.
- Efficiently searching and consuming content on platforms like YouTube, Google, and Spotify.

Features of the Project

The Virtual Mouse, Virtual Keyboard, and Virtual Assistant "Buddy" is an **all-in-one AI-driven solution** designed to revolutionize human-computer interaction. By merging **gesture recognition**, **voice commands**, and **advanced automation**, the system eliminates reliance on physical hardware while enhancing accessibility and productivity.

A. Virtual Mouse

1. Gesture-Based Control:

- Hand/finger tracking for cursor movement.
- Gestures for left/right clicks, drag-and-drop, scrolling.
- Palm detection for idle state.

B. Virtual Keyboard

1. Gesture-Based Typing:

- Finger/air-tap recognition for key presses.

C. Virtual Assistant "Buddy"

1. Application Automation:

- Open/close local applications (e.g., Notepad, Calculator).
- Launch web applications (e.g., Gmail, Google Docs).

2. Web Automation:

- Search content on Google.
- Search products on specific websites (e.g., Amazon, eBay).

3. YouTube Integration:

- Play specific videos.
- Search and queue content.

4. WhatsApp Automation:

- Send messages, initiate calls/video calls (desktop version).

5. Productivity Tools:

- Set reminders (calendar integration).
- Take notes (text-to-speech transcription).

6. Navigation Support:

- Find shortest routes via Google Maps integration.

7. Communication Tools:

- Make phone calls and send SMS (via GSM module or third-party API).

8. AI Chatbot:

- Natural language processing (NLP) for answering queries.

9. AI Image Generation:

- Generate images from text prompts using models like DALL-E/Stable Diffusion.

10. Email Automation:

- Send emails via Gmail/Outlook integration.

11. Coding Support:

- Write code snippets in Python (via CodeMaster add-on).

12. Spotify Integration:

- Play music, search artists/songs, create playlists.

13. Hotword Detection:

- Wake word activation (e.g., "Hey Buddy").

14. Chat History:

- Log and retrieve past interactions.

Add-Ons:

Buddy CodeMaster:

- Generate code in Python (and other languages via extensions).
- Debugging support.

Feasibility Study

The feasibility study for this project evaluates the technical, operational, and economic aspects of implementing a Virtual Mouse, Virtual Keyboard, and Virtual Assistant system. From a technical perspective, the project relies on readily available technologies like OpenCV, speech recognition, and automation libraries, making it feasible to develop. Operationally, the system can be integrated into everyday tasks like application control, web searches, and automation, enhancing productivity. Economically, the project's development cost is low, as it primarily uses open-source tools and APIs, making it viable for personal or small-scale business applications. Additionally, hardware requirements, such as a decent camera and microphone, are accessible to most users.

Technical Feasibility

- **Technology Stack:**
  - **Programming Languages:** Python, OpenCV for gesture recognition.
  - **Libraries and APIs:** PyAutoGUI (for mouse and keyboard control), SpeechRecognition (for voice commands), eel (for web interface), Spotify API, WhatsApp Web automation, AI-based image generation APIs.
  - **Platforms:** Compatible with Microsoft Windows.

- **Hardware Requirements:**
  - Webcam or built-in camera for gesture detection.
  - Microphone for voice input.

- **Software Requirements:**
  - Python environment.
  - Internet access for web automation and AI-based tasks (e.g., searching on Google, Spotify).

Operational Feasibility

- **User Interface:**
  - Gesture-based control for accessibility.
  - Voice commands for ease of use.
  - Web-based UI for configuration and interaction.

- **User Training:**
  - Minimal user training is required for basic operations, such as giving voice commands and making gestures.

- **Maintenance:**
  - Regular updates for new features.
  - Cloud-based or local storage for chat history and user preferences.

Hardware & Software requirement

This section outlines the necessary hardware and software components required to build and run the Virtual Mouse, Virtual Keyboard, and Virtual Assistant system efficiently. Both hardware and software need to be compatible with each other to ensure optimal performance, smooth gesture recognition, and voice-based functionalities. Below are the key requirements:

Hardware Requirements:

1. Camera: A high-definition camera (720p or higher) for accurate hand gesture detection.
2. Microphone: A quality microphone to capture voice commands clearly.
3. Processor: Minimum Intel i5 or equivalent for processing gesture recognition and voice input simultaneously.
4. RAM: 8 GB or higher for smooth multitasking.
5. GPU: A dedicated graphics card is recommended for faster image processing and AI features.
6. Storage: Minimum of 512 GB SSD for fast data access and program storage.

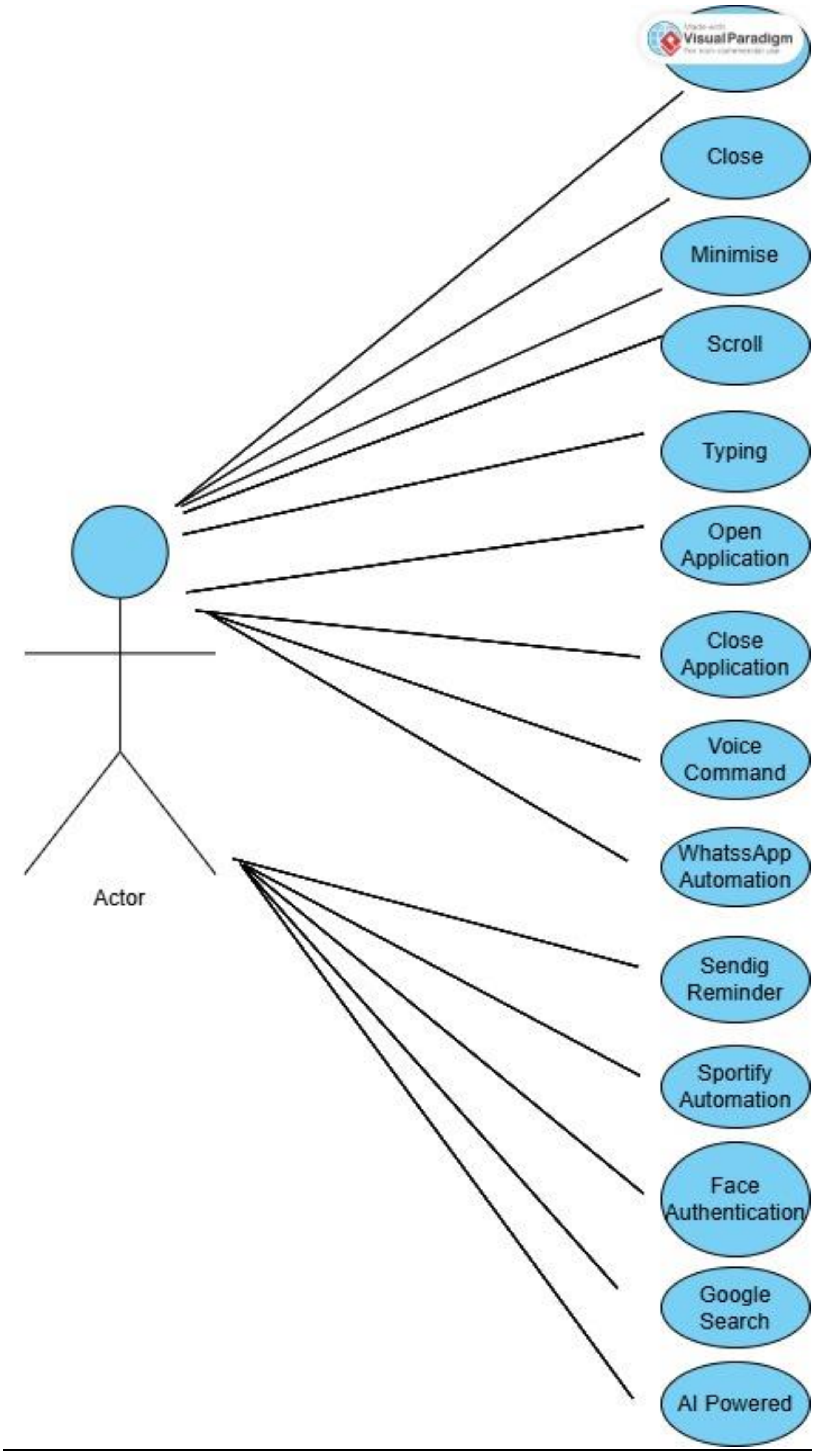
Software Requirements:

1. Operating System: Windows 10/11.
2. Python: Version 3.7 or higher for running the main scripts.
3. Libraries: OpenCV, Numpy, PyAutoGUI, SpeechRecognition, HugChat API, and other required packages for gesture and voice recognition.
4. IDE: Any Python-friendly IDE like PyCharm, VS Code, or Jupyter Notebook.
5. Browser: Microsoft Edge or Google Chrome for web-based functionalities and automation.
6. APIs: HugChat API for AI-powered conversations and interactions.

Diagrams

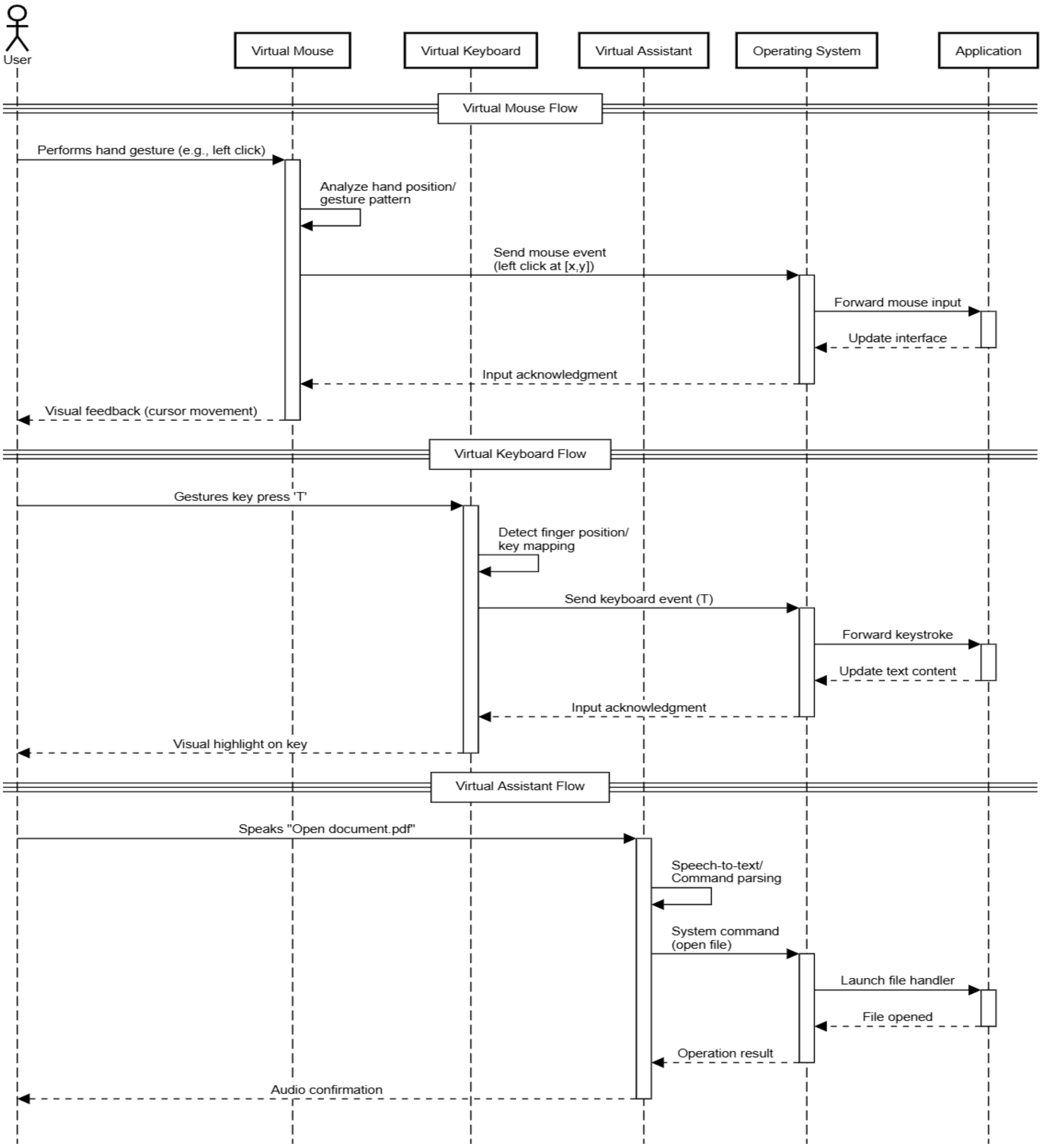
- ## Use Case Diagram

The use case diagram for the Virtual Mouse, Virtual Keyboard, and Virtual Assistant project illustrates the interactions between the user and the system's core features. It visually represents the system's functionalities, showcasing how users can perform tasks like gesture-based control, voice-activated automation, and AI-driven actions. Key use cases include application automation, web search, media control, and productivity tools like note-taking and reminders. The diagram highlights the system's ability to respond to voice commands and gestures, enabling hands-free interaction. This overview provides a clear understanding of how the various components of the project work together to enhance user experience.



- ## Sequence Diagram

This sequence diagram illustrates the interaction between the user and three main system components: the Virtual Mouse, Virtual Keyboard, and Virtual Assistant. The **Virtual Mouse** flow begins with the user performing a hand gesture, such as a left-click. The system analyzes the gesture, sends the corresponding mouse event, and updates the interface with visual feedback, like cursor movement. In the **Virtual Keyboard** flow, the user gestures to press a key (e.g., 'T'), and the system detects the finger position, sends the key press event, and updates the text content, providing visual feedback by highlighting the key. The **Virtual Assistant** flow involves the user speaking a command, such as "Open document.pdf." The system processes the speech, interprets the command, and executes the operation, providing audio confirmation once the task is completed.



Project Limitations

Before diving into the limitations, it's essential to recognize that the project demonstrates significant potential by integrating gesture control, voice commands, and automation features into a unified virtual assistant. Despite its capabilities, certain limitations may affect its overall performance and user experience.

Limitations:

1. Gesture recognition may struggle with accuracy due to factors like lighting or camera quality.
2. Latency issues can arise, especially for internet-based features like web search or automation.
3. Hardware requirements, such as a high-quality camera and microphone, are essential for smooth performance.
4. Privacy concerns may emerge as the system processes sensitive data, such as voice commands and personal reminders.
5. Some applications may not fully support automation due to security restrictions or API limitations.

Future Enhancements

As the project progresses, there are numerous opportunities for enhancement that can improve its functionality, usability, and overall user experience. These future improvements aim to increase the system's adaptability, make it more user-friendly, and enhance its offline capabilities. Below are some key areas for potential enhancements:

Future Enhancements:

1. **Enhanced Gesture Recognition**
   • Implement AI-based gesture learning to adapt to different hand sizes and movements.
   • Add multi-finger gestures for more precise controls.
2. **Voice Assistant Improvements**
   • Enable custom voice commands so users can define their own shortcuts.
   • Add multi-language support for better accessibility.
3. **Improved Virtual Keyboard**
   • Add predictive text and auto-correction for smoother typing.
   • Support emoji and special character input.
4. **Offline Functionality**
   • Enable basic commands and operations to work without an internet connection.
   • Develop a local AI model to process voice commands without sending data online.

Bibliography

The following resources were utilized during the development of this project, providing valuable insights into gesture recognition, voice processing, automation, and AI integration. These references include official documentation, research papers, and external APIs that contributed to the functionality and implementation of the Virtual Mouse, Virtual Keyboard, and Virtual Assistant features. Each source was integral in ensuring accurate performance and reliable outputs for the project.

□ **OpenCV Documentation**
OpenCV Development Team. "OpenCV: Open Source Computer Vision Library." Available at: https://opencv.org/. Accessed March 2025.

□ **Python Official Documentation**
Python Software Foundation. "Python Documentation." Available at: https://docs.python.org/3/. Accessed March 2025.

□ **Eel Documentation**
Eel Development Team. "Eel: For Simple Electron-Like Offline/Online Apps." Available at: https://github.com/samuelhwilliams/Eel. Accessed March 2025.

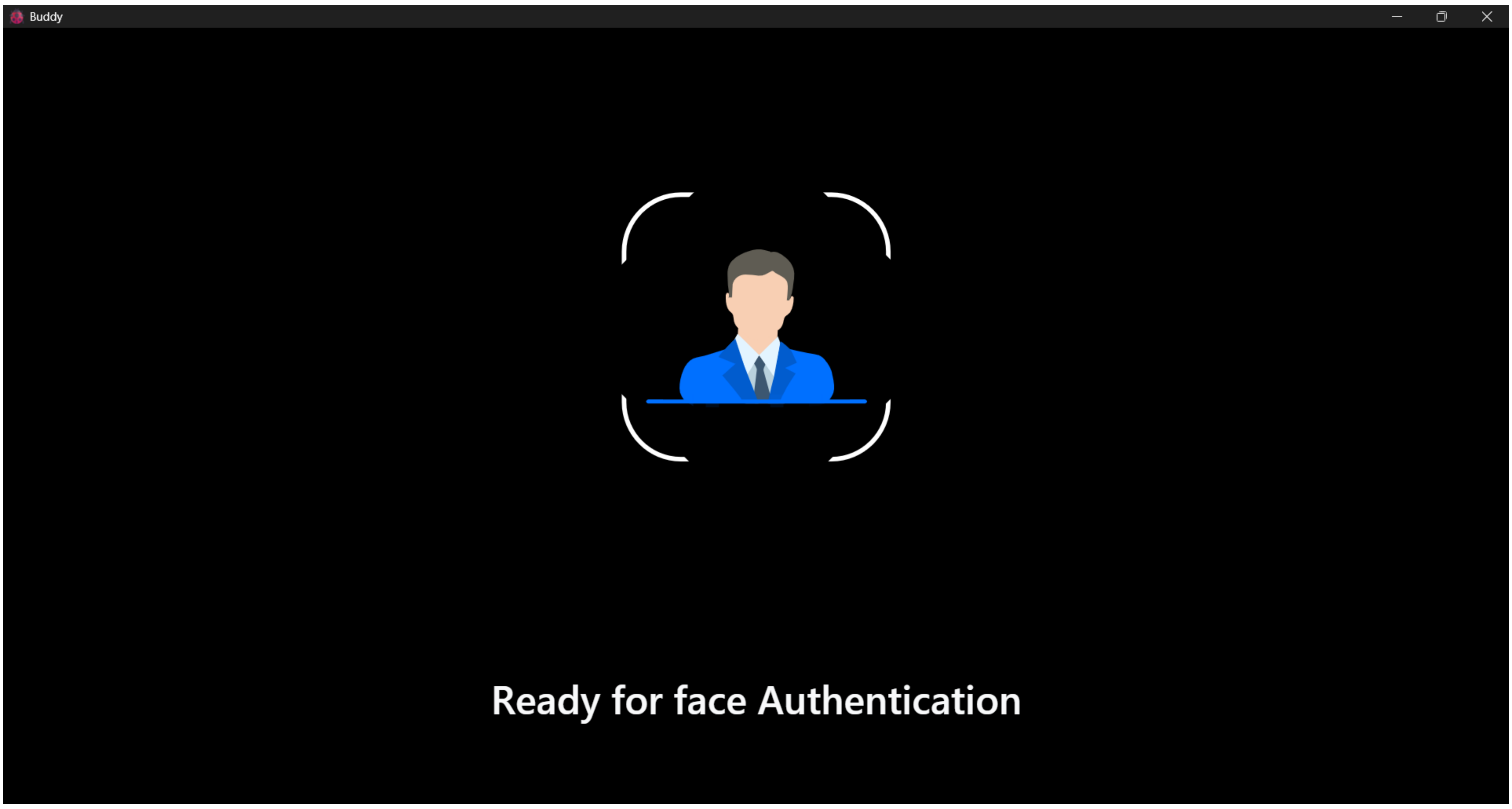
□ **Spotify Web API Documentation**
Spotify Developers. "Spotify for Developers: Web API." Available at: https://developer.spotify.com/documentation/web-api/. Accessed March 2025.

□ **PyAutoGUI Documentation**
Al Sweigart. "PyAutoGUI: Python GUI Automation." Available at: https://pyautogui.readthedocs.io/en/latest/. Accessed March 2025.

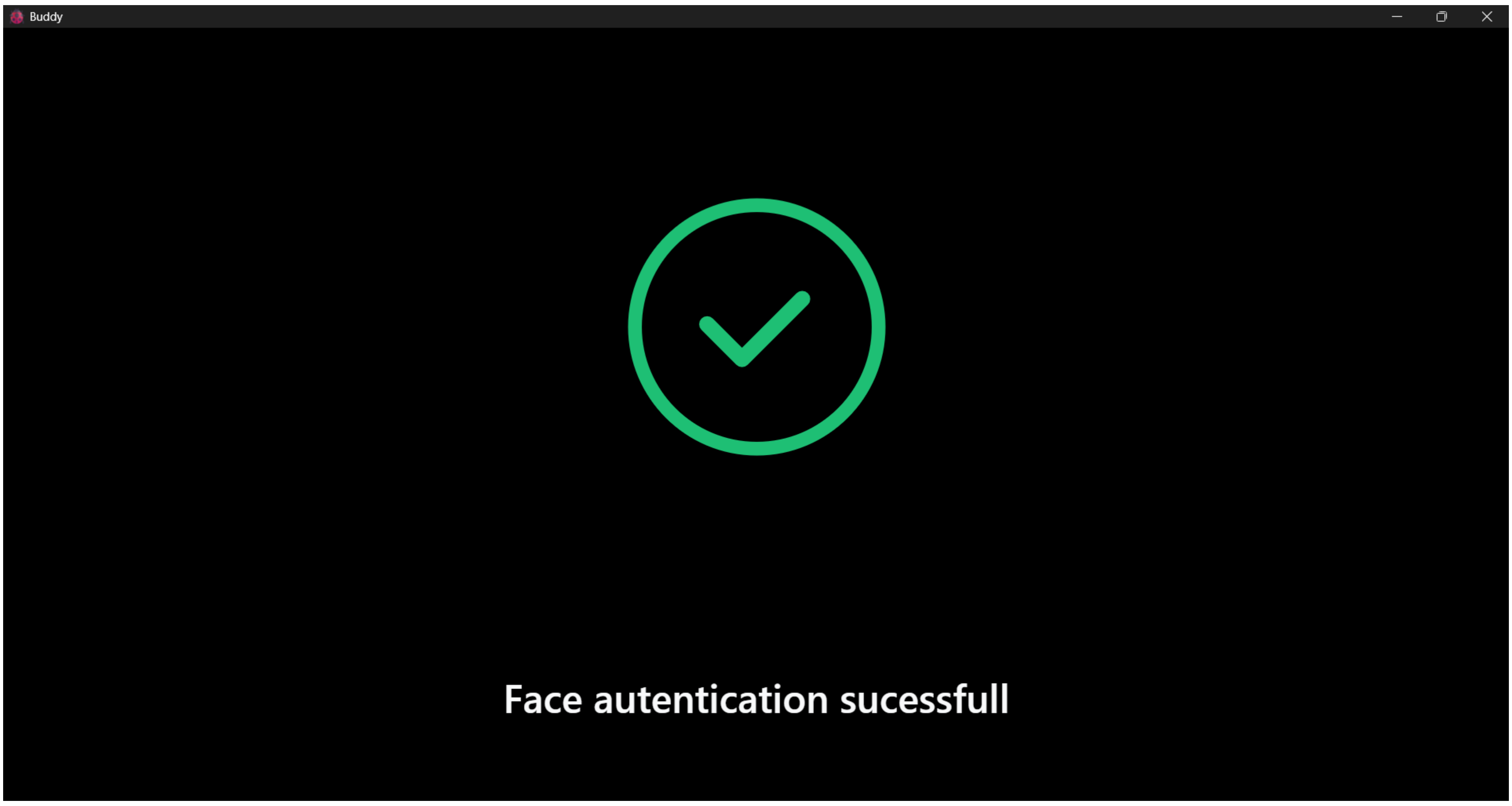
□ **SpeechRecognition Library Documentation**
Uberi. "SpeechRecognition: Speech Recognition in Python." Available at: https://pypi.org/project/SpeechRecognition/. Accessed March 2025.

□ **Flask Documentation**
Pallets Projects. "Flask Documentation." Available at: https://flask.palletsprojects.com/en/2.0.x/. Accessed March 202

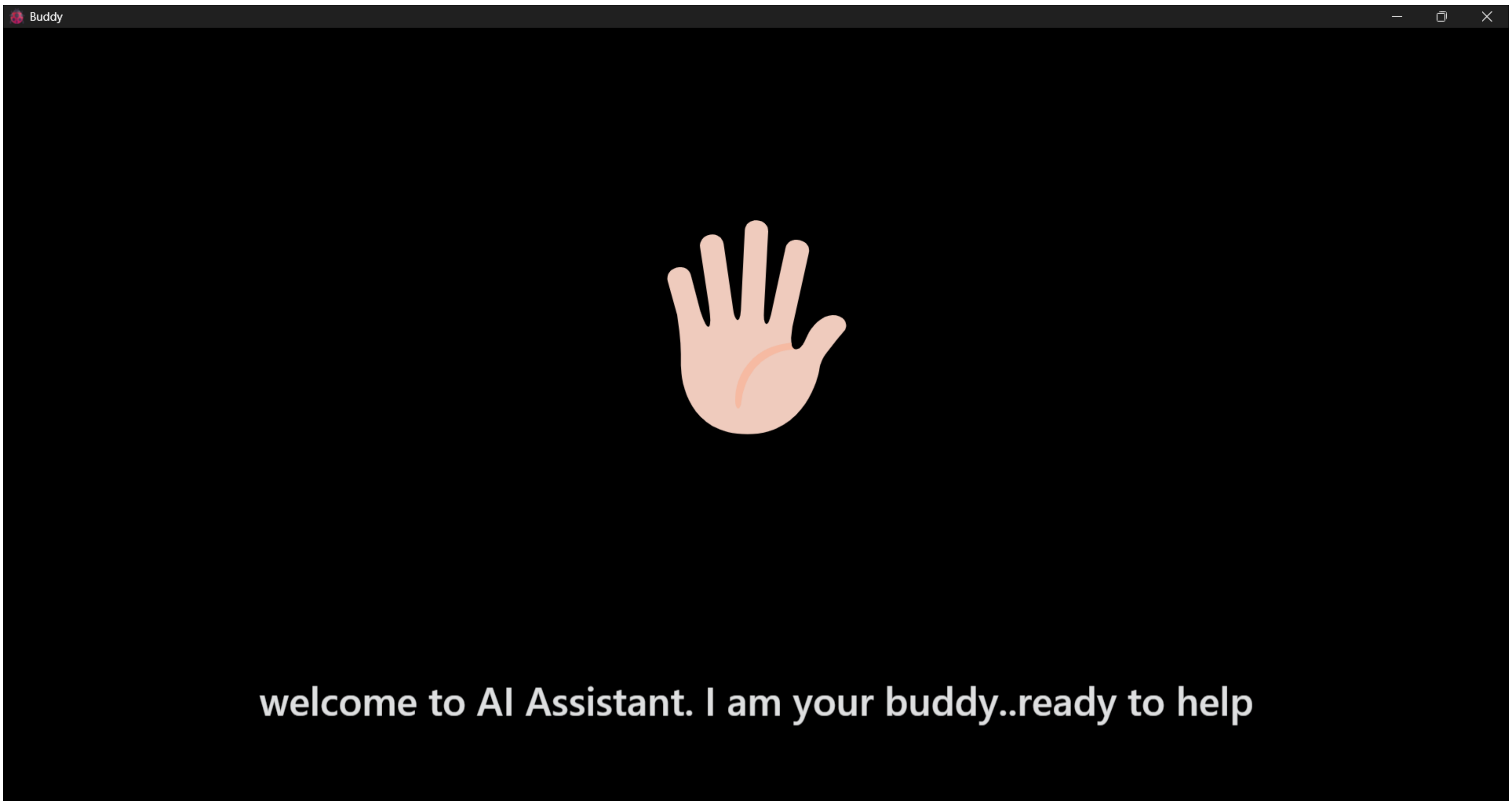
Screenshots



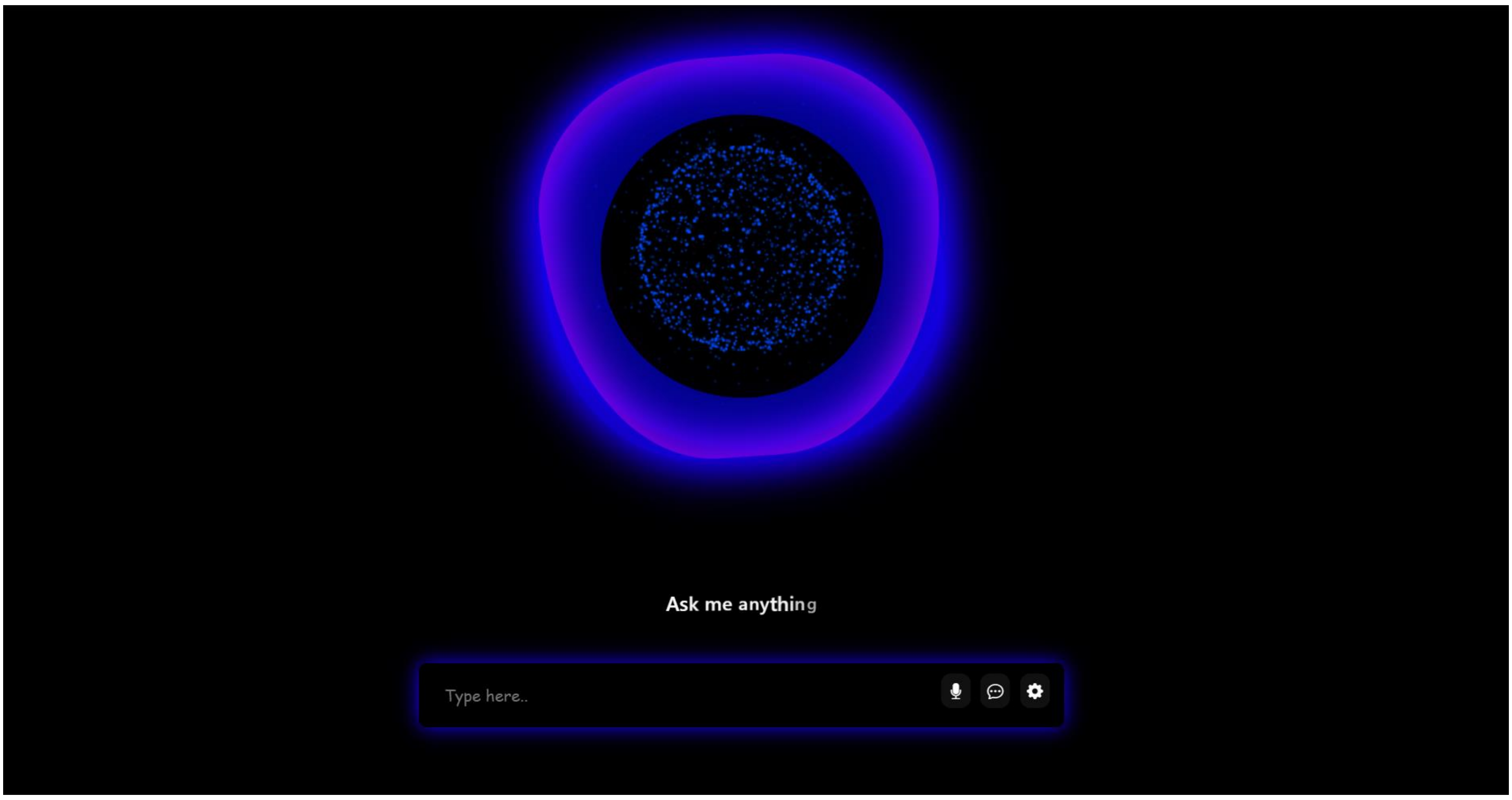
Ready for face Authentication



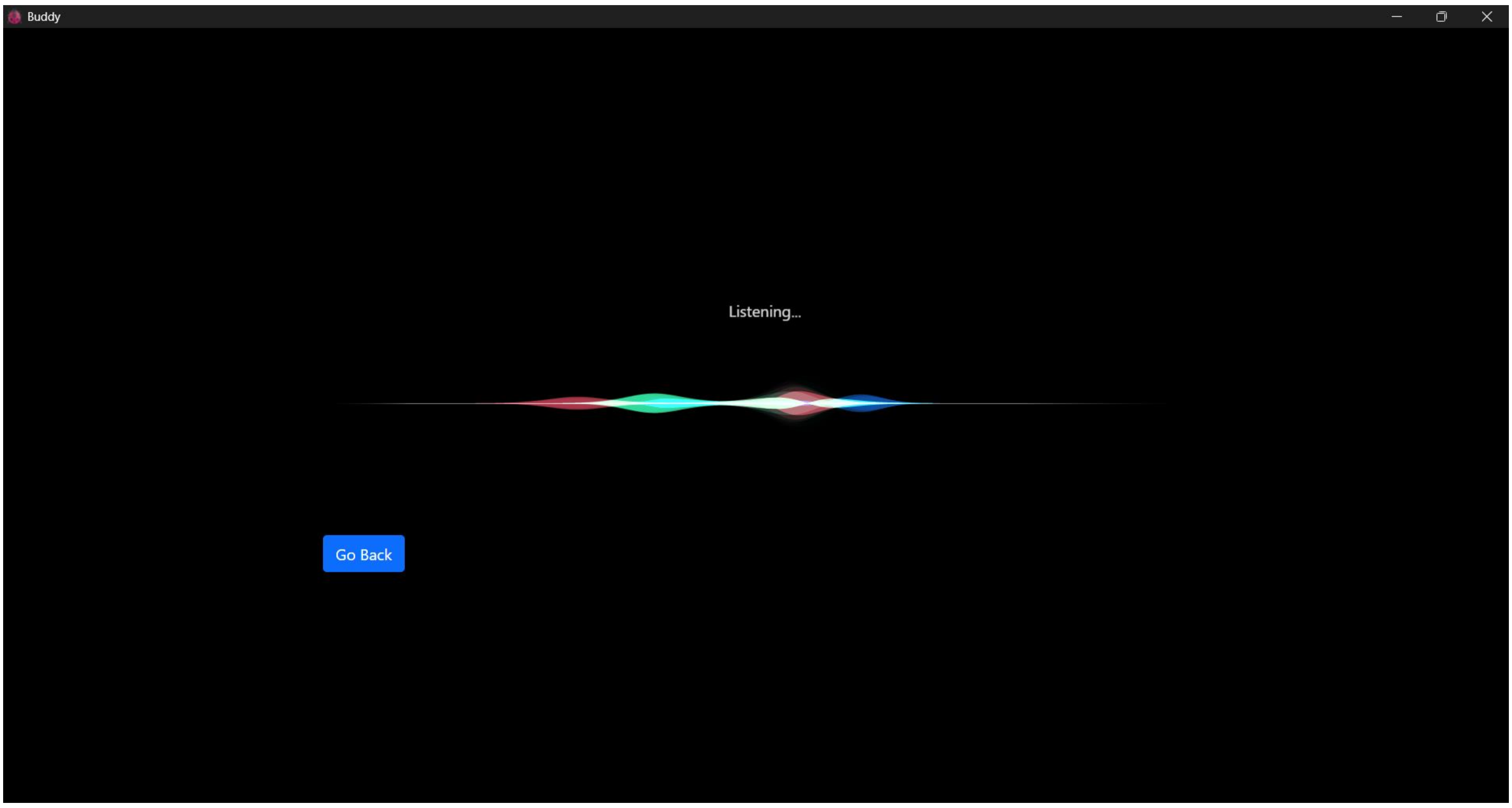
Face autentication sucessfull



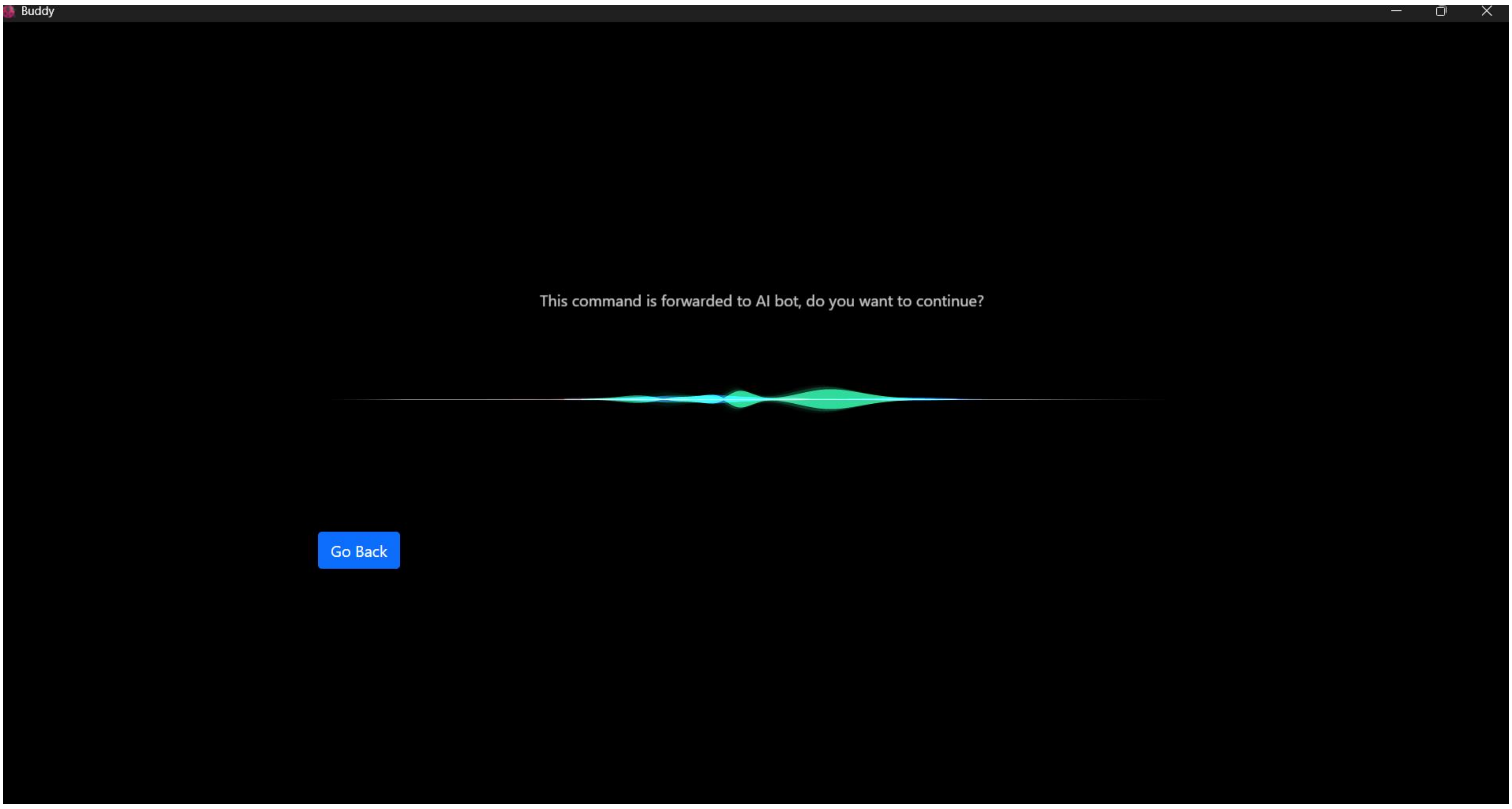
welcome to AI Assistant. I am your buddy..ready to help



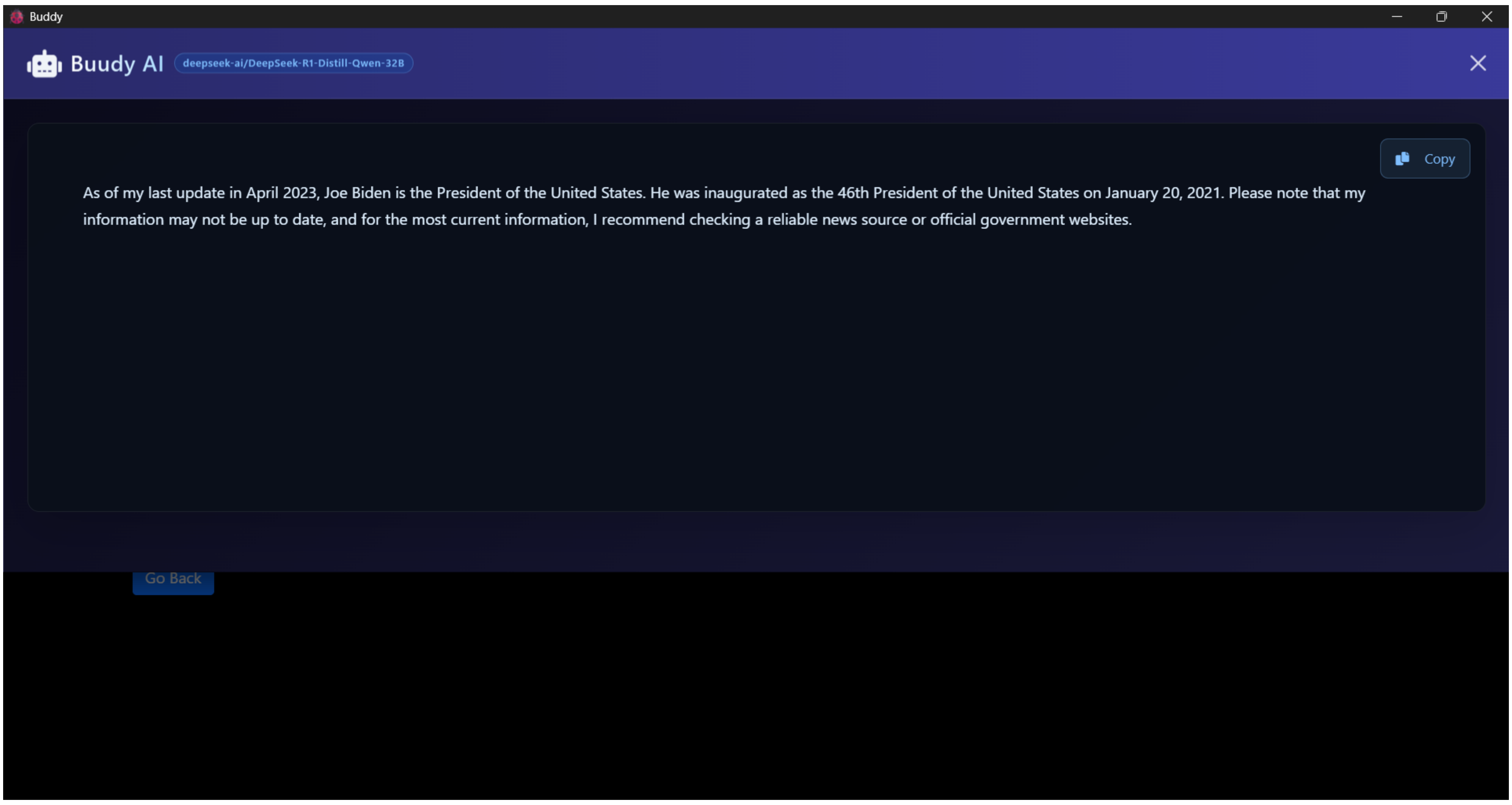
Ask me anything



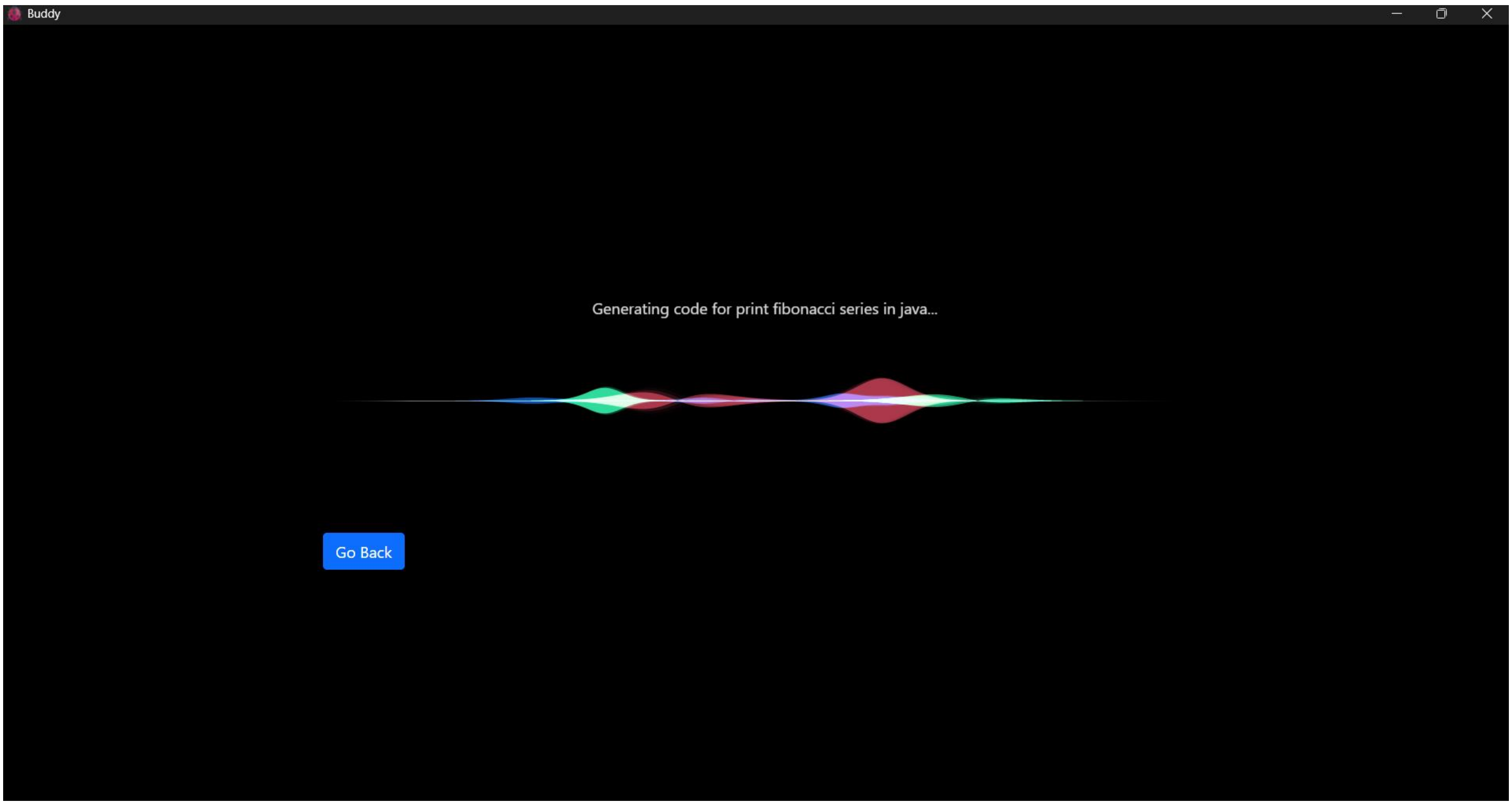
Listening...



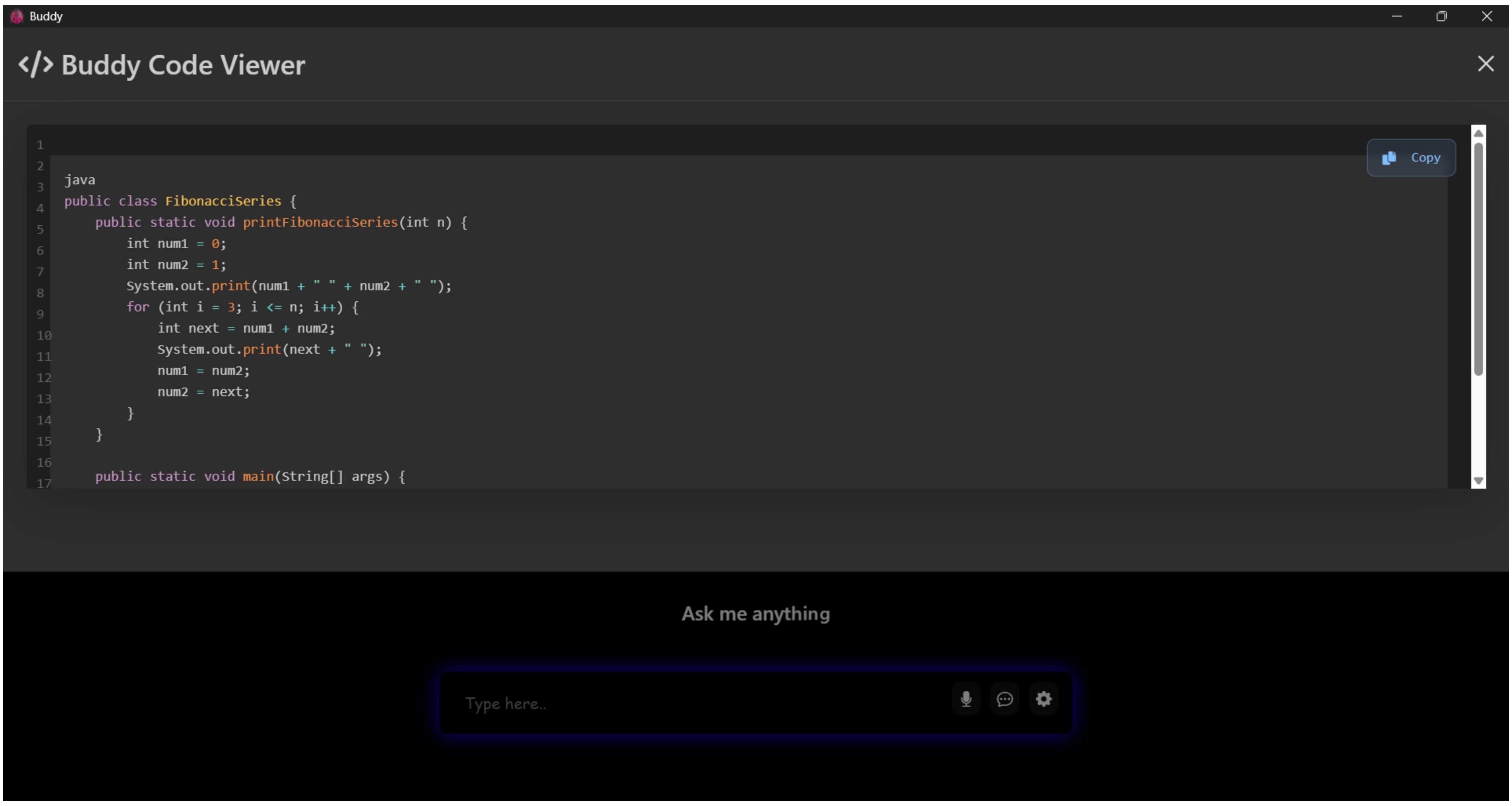
This command is forwarded to AI bot, do you want to continue?



As of my last update in April 2023, Joe Biden is the President of the United States. He was inaugurated as the 46th President of the United States on January 20, 2021. Please note that my information may not be up to date, and for the most current information, I recommend checking a reliable news source or official government websites.

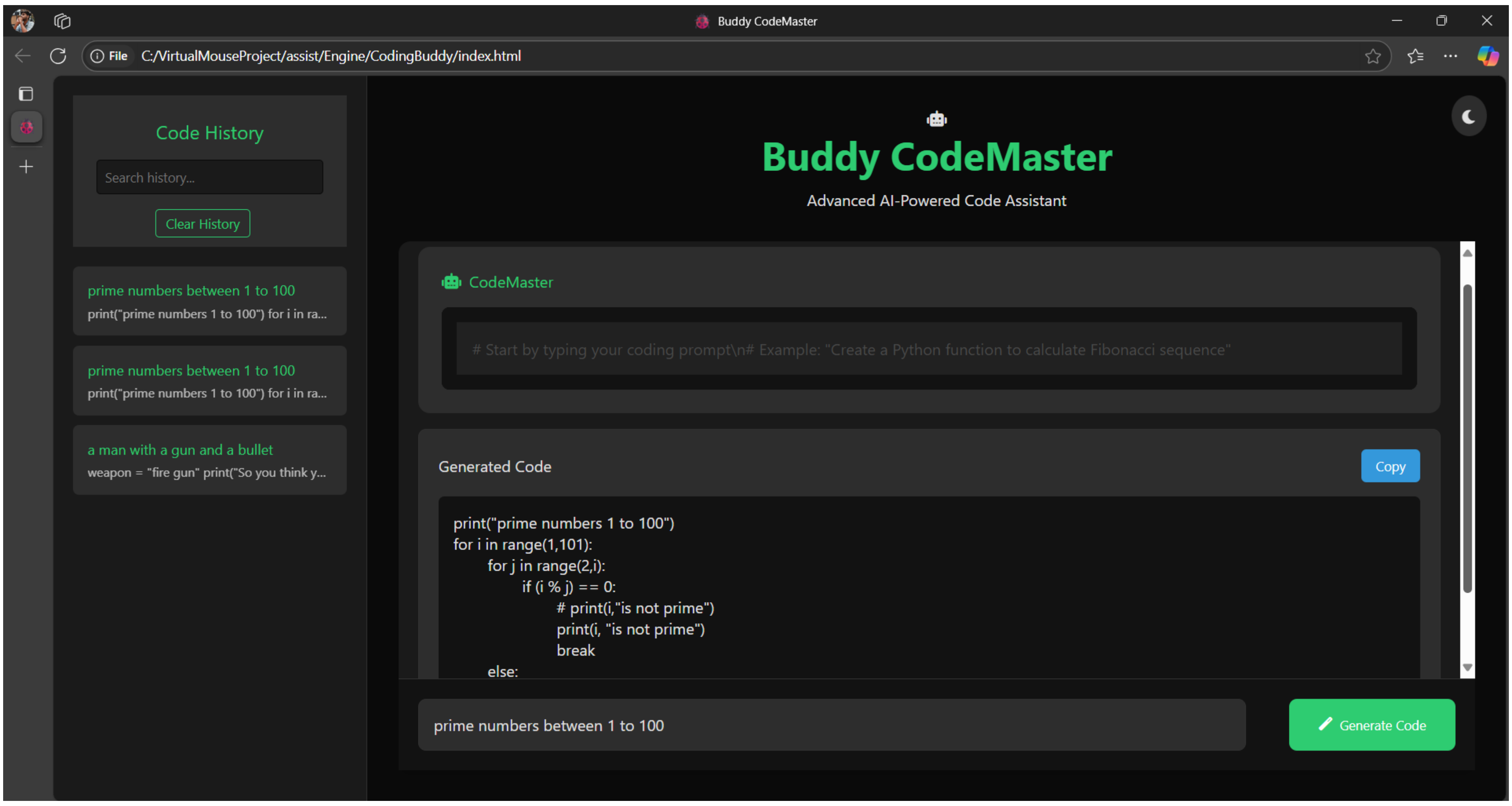


Generating code for print fibonacci series in java...



```java
java
public class FibonacciSeries {
    public static void printFibonacciSeries(int n) {
        int num1 = 0;
        int num2 = 1;
        System.out.print(num1 + " " + num2 + " ");
        for (int i = 3; i <= n; i++) {
            int next = num1 + num2;
            System.out.print(next + " ");
            num1 = num2;
            num2 = next;
        }
    }

    public static void main(String[] args) {
```



prime numbers between 1 to 100
print("prime numbers 1 to 100") for i in ra...

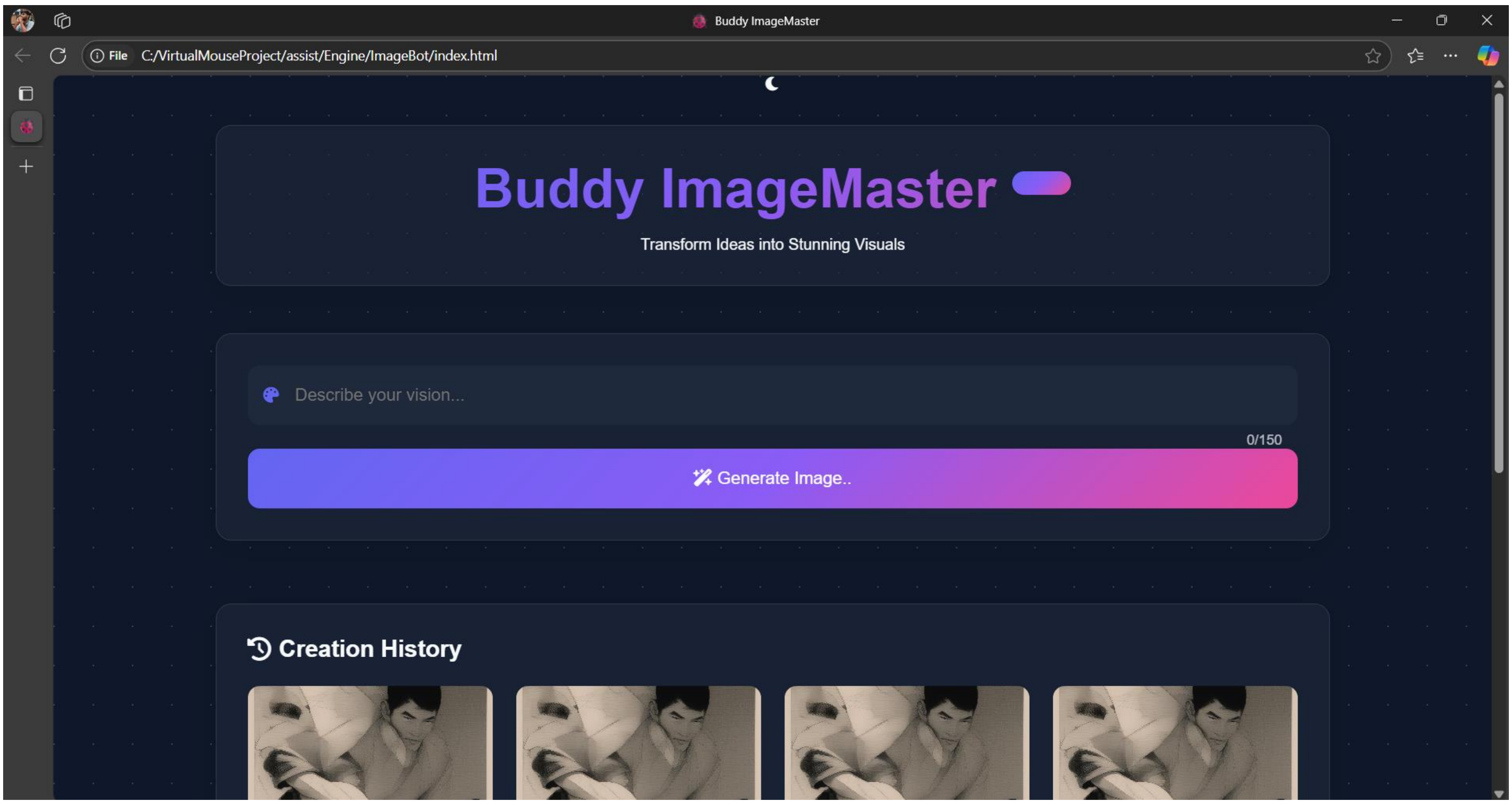
prime numbers between 1 to 100
print("prime numbers 1 to 100") for i in ra...

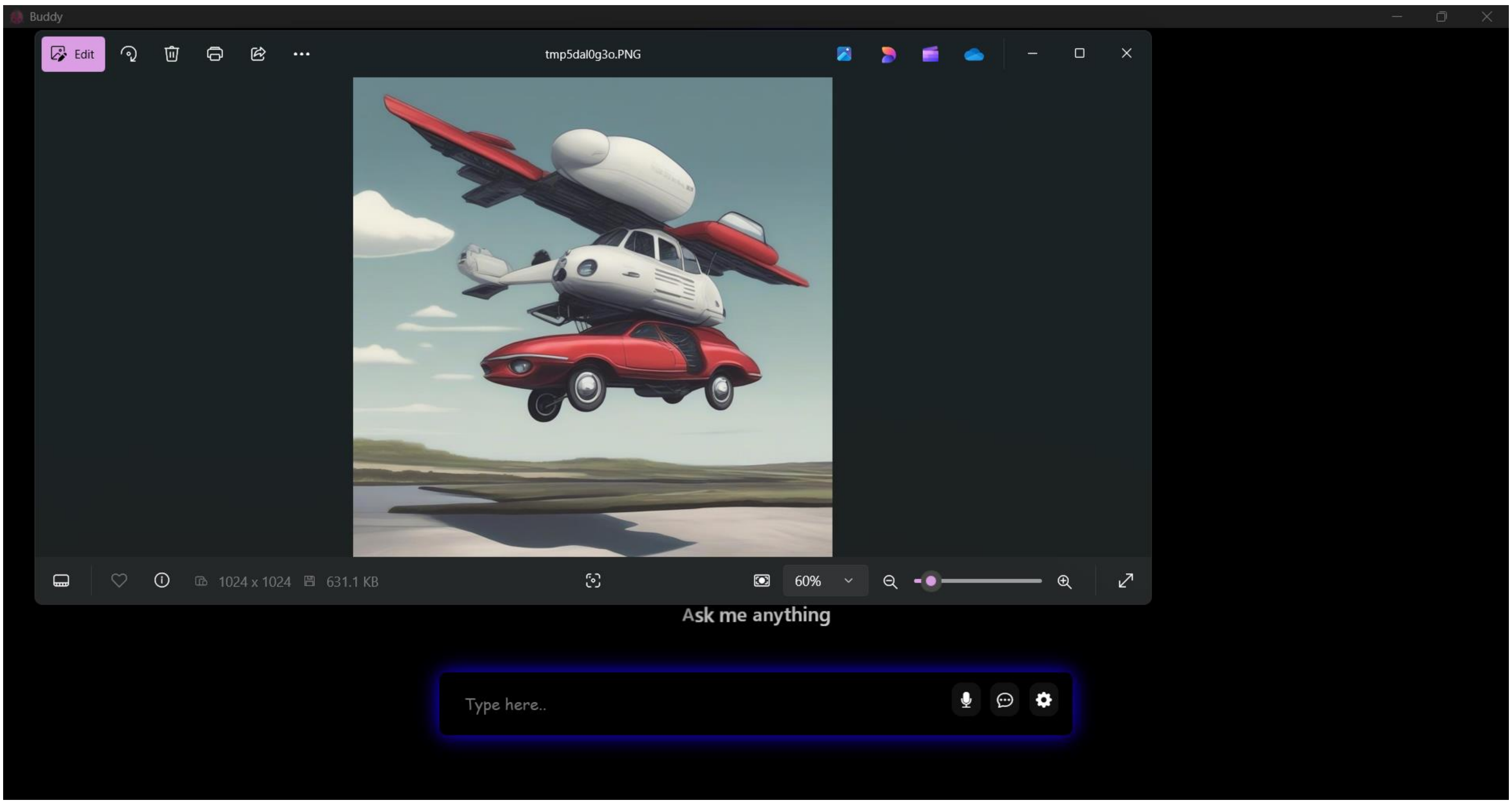
a man with a gun and a bullet
weapon = "fire gun" print("So you think y...

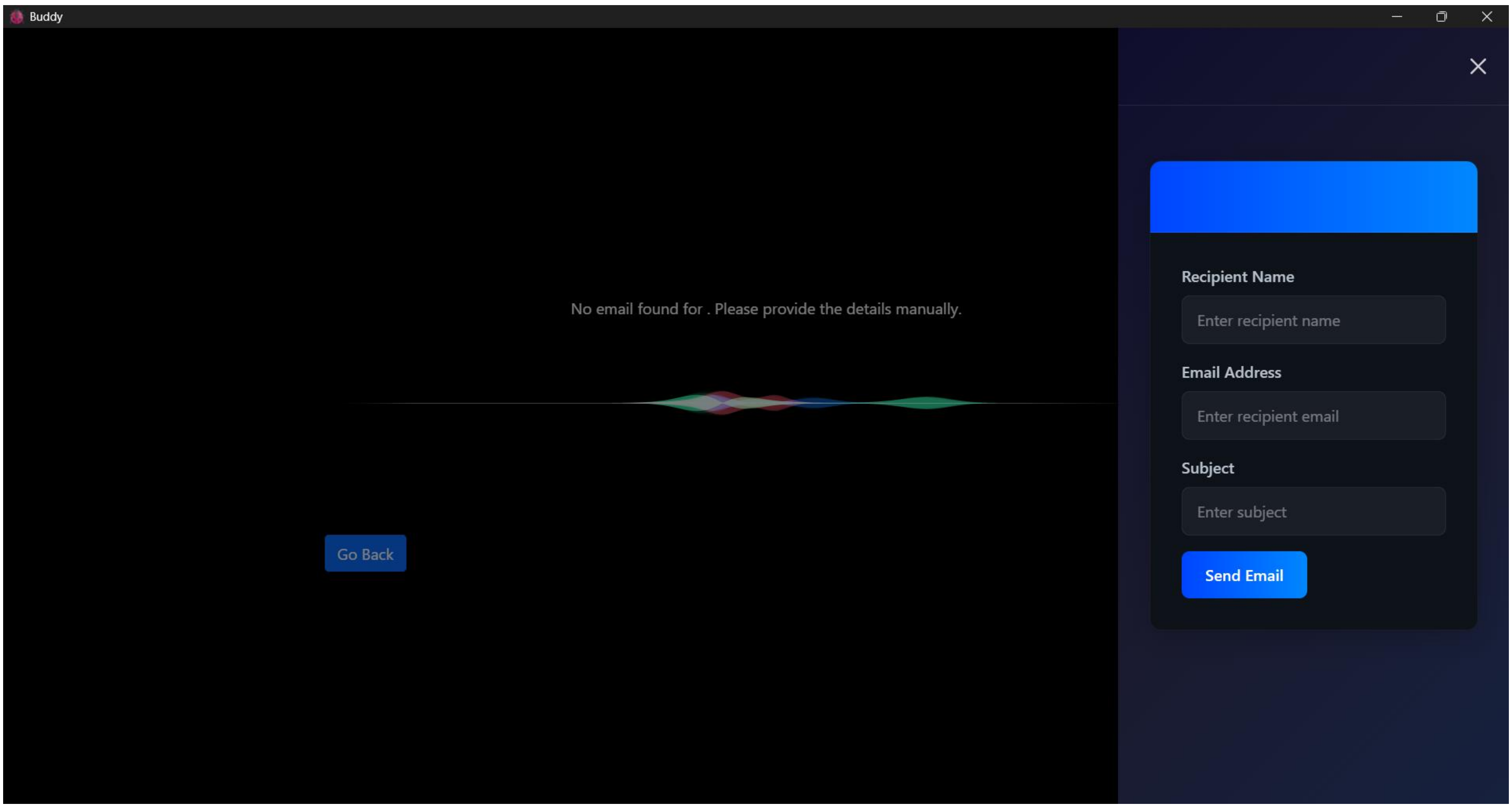
Start by typing your coding prompt\n# Example: "Create a Python function to calculate Fibonacci sequence"

```python
print("prime numbers 1 to 100")
for i in range(1,101):
        for j in range(2,i):
                if (i % j) == 0:
                        # print(i,"is not prime")
                        print(i, "is not prime")
                        break
        else:
```

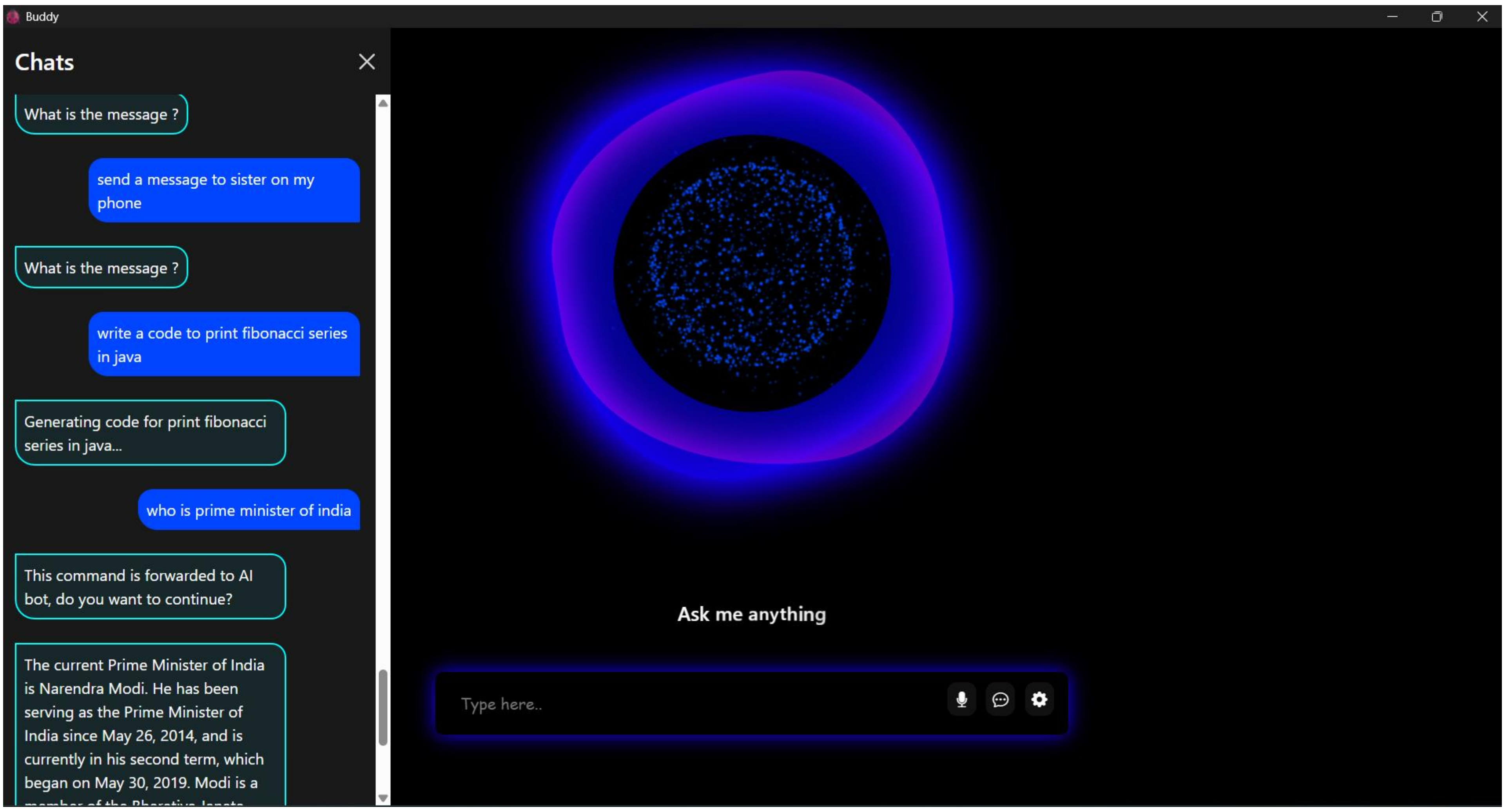
prime numbers between 1 to 100


Buddy ImageMaster
File C:/VirtualMouseProject/assist/Engine/ImageBot/index.html
Buddy ImageMaster
Transform Ideas into Stunning Visuals
Describe your vision...
0/150
Generate Image..
Creation History


Buddy
Edit
tmp5dal0g3o.PNG
1024 x 1024
631.1 KB
60%
Ask me anything
Type here..



No email found for . Please provide the details manually.



What is the message ?

send a message to sister on my phone

What is the message ?

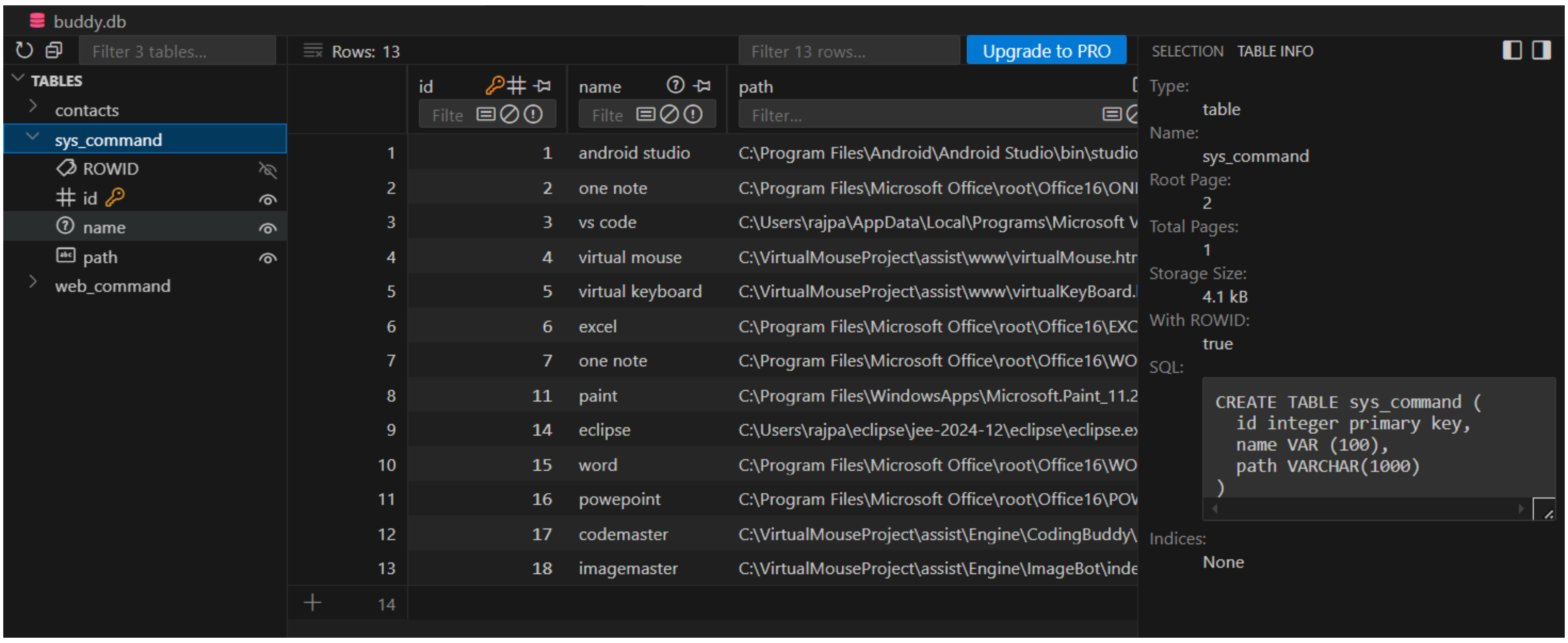
write a code to print fibonacci series in java

Generating code for print fibonacci series in java...

who is prime minister of india

This command is forwarded to AI bot, do you want to continue?

The current Prime Minister of India is Narendra Modi. He has been serving as the Prime Minister of India since May 26, 2014, and is currently in his second term, which began on May 30, 2019. Modi is a

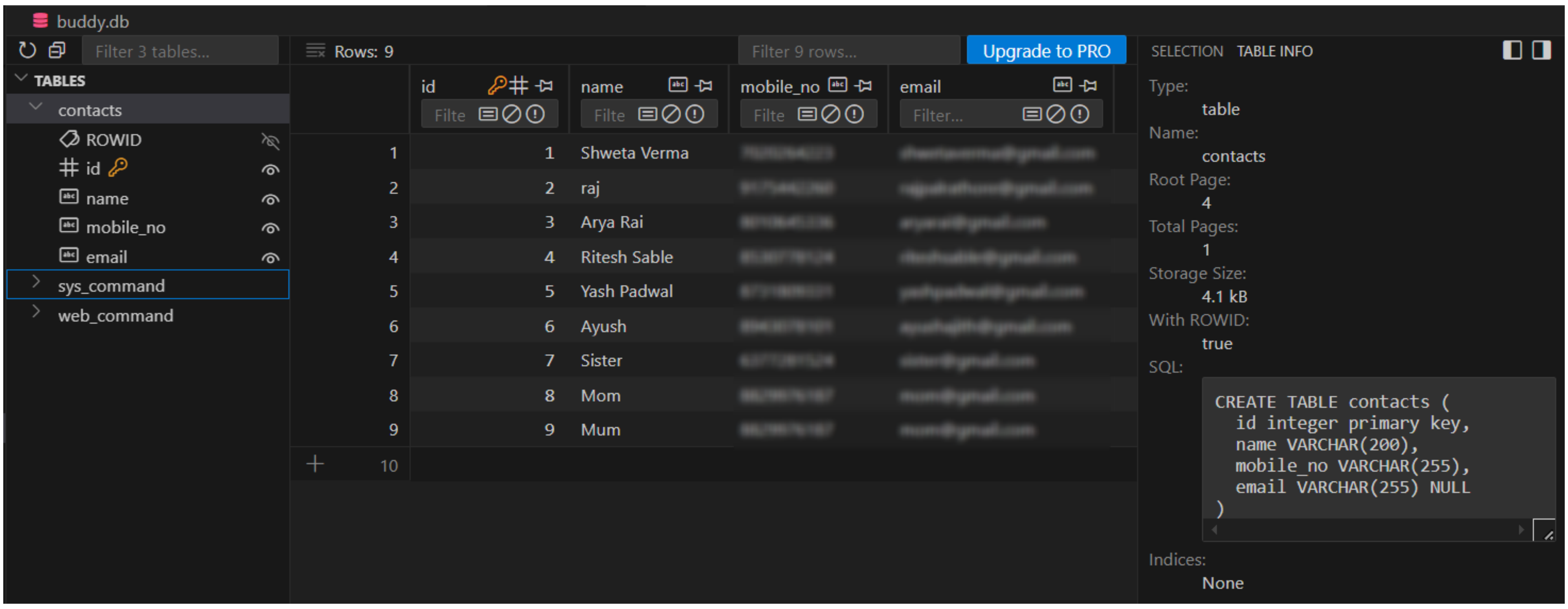


	id	name	path
1	1	android studio	C:\Program Files\Android\Android Studio\bin\studio
2	2	one note	C:\Program Files\Microsoft Office\root\Office16\ONI
3	3	vs code	C:\Users\rajpa\AppData\Local\Programs\Microsoft V
4	4	virtual mouse	C:\VirtualMouseProject\assist\www\virtualMouse.htr
5	5	virtual keyboard	C:\VirtualMouseProject\assist\www\virtualKeyBoard.l
6	6	excel	C:\Program Files\Microsoft Office\root\Office16\EXC
7	7	one note	C:\Program Files\Microsoft Office\root\Office16\WO
8	11	paint	C:\Program Files\WindowsApps\Microsoft.Paint_11.2
9	14	eclipse	C:\Users\rajpa\eclipse\jee-2024-12\eclipse\eclipse.ex
10	15	word	C:\Program Files\Microsoft Office\root\Office16\WO
11	16	powepoint	C:\Program Files\Microsoft Office\root\Office16\POV
12	17	codemaster	C:\VirtualMouseProject\assist\Engine\CodingBuddy\
13	18	imagemaster	C:\VirtualMouseProject\assist\Engine\ImageBot\inde

Type:
table
Name:
sys_command
Root Page:
2
Total Pages:
1
Storage Size:
4.1 kB
With ROWID:
true
SQL:

```
CREATE TABLE sys_command (
  id integer primary key,
  name VAR (100),
  path VARCHAR(1000)
)
```

Indices:
None

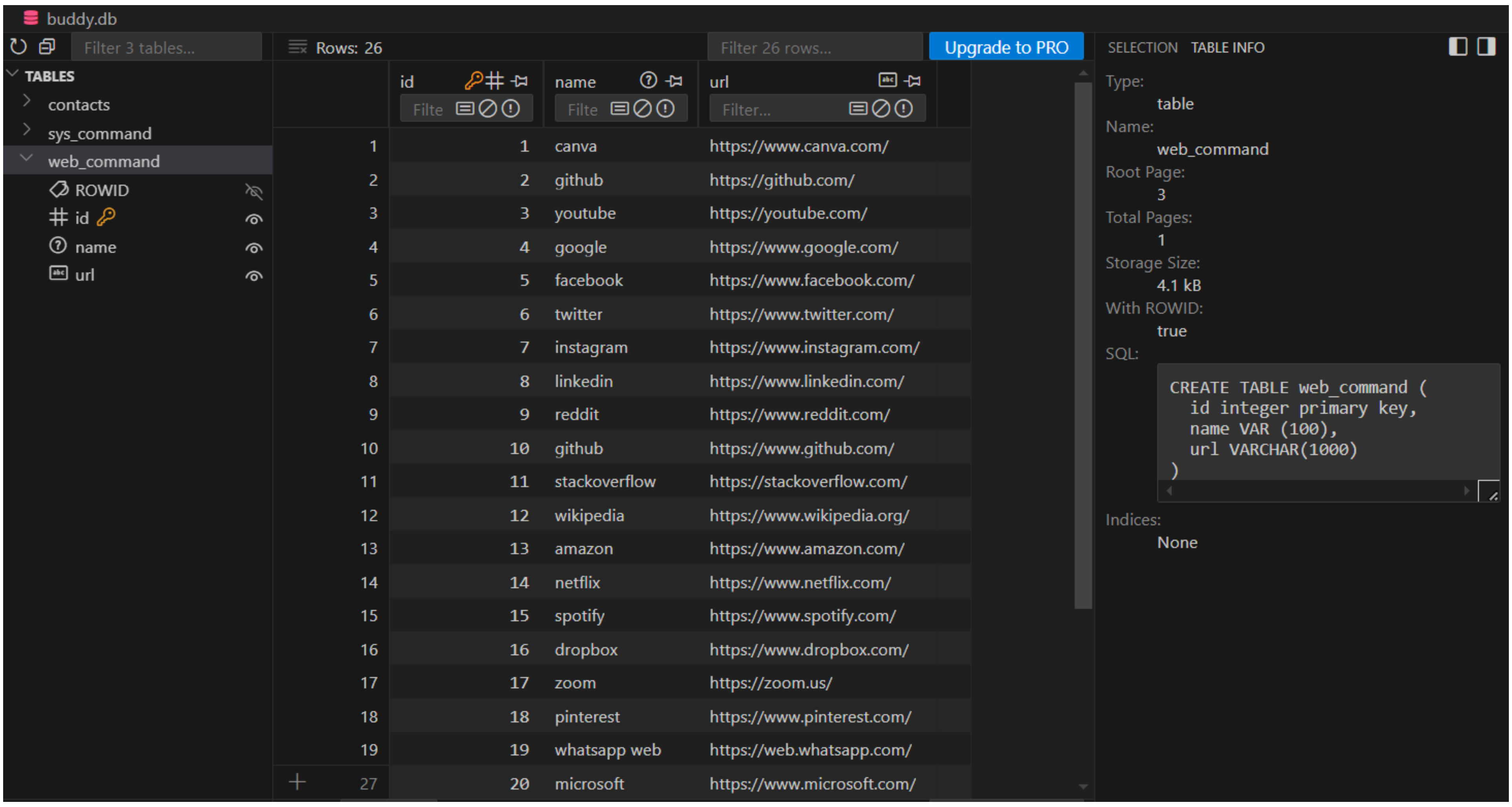


	id	name	mobile_no	email
1	1	Shweta Verma	[redacted]	[redacted]
2	2	raj	[redacted]	[redacted]
3	3	Arya Rai	[redacted]	[redacted]
4	4	Ritesh Sable	[redacted]	[redacted]
5	5	Yash Padwal	[redacted]	[redacted]
6	6	Ayush	[redacted]	[redacted]
7	7	Sister	[redacted]	[redacted]
8	8	Mom	[redacted]	[redacted]
9	9	Mum	[redacted]	[redacted]

Type:
table
Name:
contacts
Root Page:
4
Total Pages:
1
Storage Size:
4.1 kB
With ROWID:
true
SQL:

```
CREATE TABLE contacts (
  id integer primary key,
  name VARCHAR(200),
  mobile_no VARCHAR(255),
  email VARCHAR(255) NULL
)
```

Indices:
None



TABLES

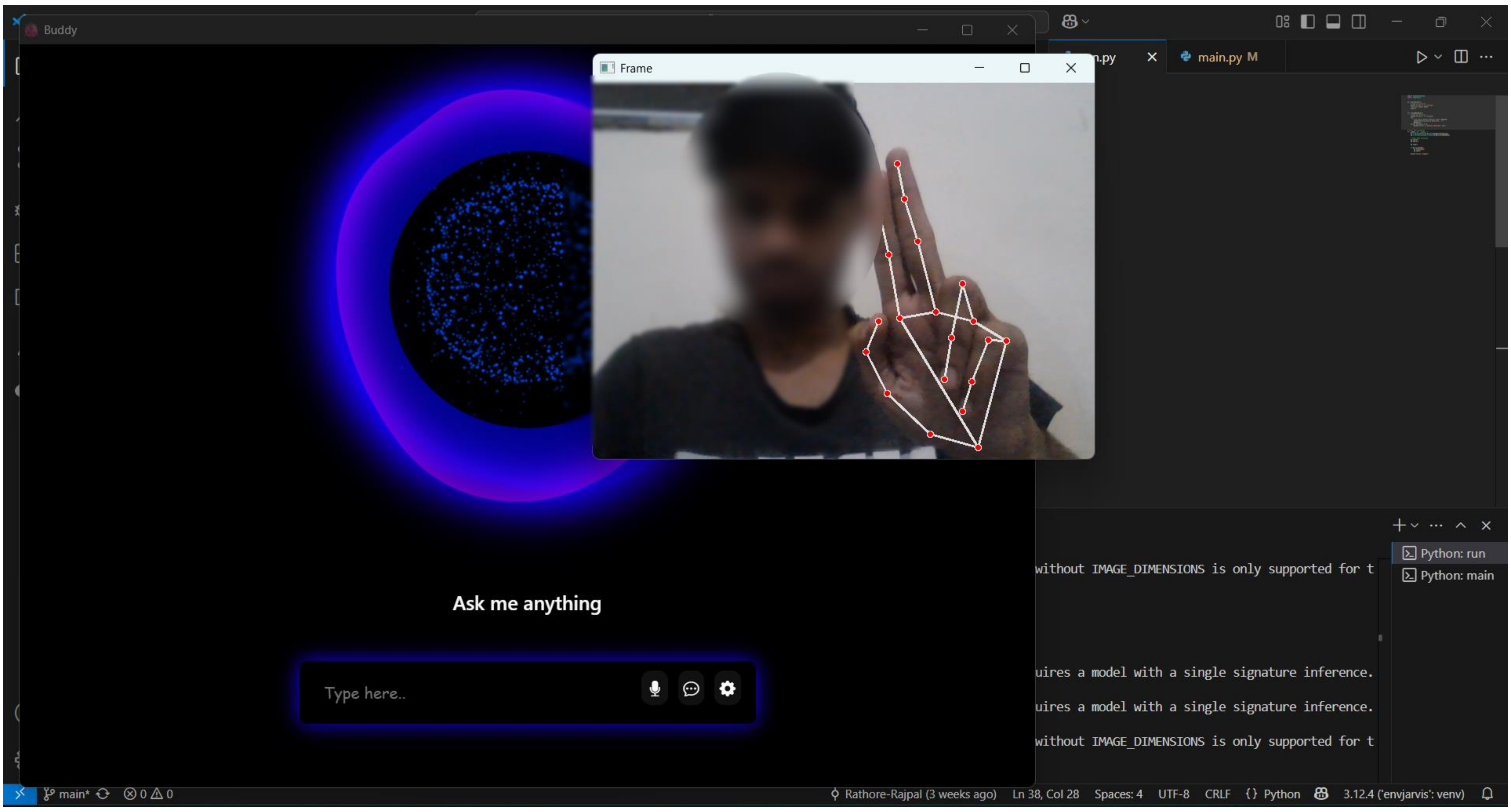
- contacts
- sys_command
- web_command
  - ROWID
  - id
  - name
  - url

	id	name	url
1	1	canva	https://www.canva.com/
2	2	github	https://github.com/
3	3	youtube	https://youtube.com/
4	4	google	https://www.google.com/
5	5	facebook	https://www.facebook.com/
6	6	twitter	https://www.twitter.com/
7	7	instagram	https://www.instagram.com/
8	8	linkedin	https://www.linkedin.com/
9	9	reddit	https://www.reddit.com/
10	10	github	https://www.github.com/
11	11	stackoverflow	https://stackoverflow.com/
12	12	wikipedia	https://www.wikipedia.org/
13	13	amazon	https://www.amazon.com/
14	14	netflix	https://www.netflix.com/
15	15	spotify	https://www.spotify.com/
16	16	dropbox	https://www.dropbox.com/
17	17	zoom	https://zoom.us/
18	18	pinterest	https://www.pinterest.com/
19	19	whatsapp web	https://web.whatsapp.com/
27	20	microsoft	https://www.microsoft.com/

Type:
table
Name:
web_command
Root Page:
3
Total Pages:
1
Storage Size:
4.1 kB
With ROWID:
true
SQL:

```
CREATE TABLE web_command (
  id integer primary key,
  name VAR (100),
  url VARCHAR(1000)
)
```

Indices:
None



Ask me anything

Type here..

```
without IMAGE_DIMENSIONS is only supported for t

uires a model with a single signature inference.
uires a model with a single signature inference.
without IMAGE_DIMENSIONS is only supported for t
```

main* ⊗ 0 ⚠ 0   Rathore-Rajpal (3 weeks ago) Ln 38, Col 28 Spaces: 4 UTF-8 CRLF {} Python 3.12.4 ('envjarvis': venv)