



Auto Reply Chatbot AI

Mega Project 2: Auto Reply Chatbot (WhatsApp Auto Responder)

? Problem Statement

Build a Python bot that:

- Selects chat text from the screen
- Copies it
- Sends it to Gemini AI with a specific role/personality
- Types and sends the AI-generated response back in the chat app

Requirements & Setup

Install Required Libraries

```
#Shell Script
pip install pyautogui
pip install pyperclip
pip install google-generativeai
```

File Structure

```
CopyEdit
AutoReplyBot/
├── program.py
├── get_cursor.py
```

Concepts Used

Feature	Concept Used
Cursor tracking	<code>pyautogui.position()</code>
Screen automation	<code>pyautogui.moveTo()</code> , <code>click()</code>
Clipboard copy/paste	<code>pyperclip.paste()</code>
Delay/sleep	<code>time.sleep()</code>
Google Gemini AI Integration	<code>google.generativeai</code>
Simulating human typing	<code>pyautogui.write()</code>

API Key Warning

API Key has been removed for privacy.

```
genai.configure(api_key="YOUR_API_KEY")
```

`get_cursor.py` (For Getting Coordinates)

```
import pyautogui

# Continuously prints the current mouse cursor position
while True:
    a = pyautogui.position()
    print(a)
```

Used to identify screen positions for drag, click, paste, and send actions.



program.py

– Full Code with Comments

```
import pyautogui
import time
import pyperclip
from google import generativeai as genai

print("Program has initiated")

# Configure Gemini API – REPLACE API KEY
genai.configure(api_key="YOUR_API_KEY")
model = genai.GenerativeModel("gemini-2.0-flash")

# Delay to switch to chat window
time.sleep(2)

# Step 1: Click on the app icon (Edge or WhatsApp Web)
pyautogui.click(644, 18)
time.sleep(1)

# Step 2: Drag to select the chat messages
pyautogui.moveTo(755, 520)
pyautogui.mouseDown()
```

```
pyautogui.moveTo(962, 673, duration=0.5)
pyautogui.mouseUp()
time.sleep(0.5)
```

```
# Step 3: Copy the selected text
pyautogui.hotkey('ctrl', 'c')
time.sleep(0.5)
```

```
# Step 4: Get text from clipboard
chat_history = pyperclip.paste()
print("Copied Text:", chat_history)
```

```
# Step 5: Generate reply using Gemini AI
response = model.generate_content(
    contents=[
        {
            "role": "user",
            "parts": ["You are Prathamesh, a 17yr old boy from India who is passionate about programming. Your role is to reply to messages (short and like real human WhatsApp replies). Don't sound like AI, reply to all messages in one go. No time or fancy structure."]
        },
        {
            "role": "user",
            "parts": [chat_history]
        }
    ]
)
```

```
# Step 6: Save generated text
generated_text = response.text
print(generated_text)
```

```
# Step 7: Paste and send reply
time.sleep(5)
pyautogui.click(834, 729) # Click on input field
```

```
time.sleep(0.5)
pyautogui.write(generated_text, interval=0.01) # Simulate typing

# Step 8: Click send button
time.sleep(0.5)
pyautogui.click(1285, 733)
```

Custom Cursor Coordinates

Task	Coordinates (x, y)
Open Edge browser	(644, 18)
Drag chat (start)	(755, 520)
Drag chat (end)	(962, 673)
Paste in chat input	(834, 729)
Click send	(1285, 733)
VS Code or debugging coords	(571, 11), (737, 538)

Improvements & Extensions

- Add keyboard shortcut to trigger the bot
- Integrate with OpenAI or other LLMs
- Detect new incoming messages automatically
- Add GUI to set drag/select positions
- Add translation or multi-language support
- Encrypt API key in a `.env` file