Anonymous Chatbot Project Using FastAPI

This document provides a concise guide to creating, running, and accessing a simple anonymous messaging web application using FastAPI. The chatbot allows users to send messages anonymously via a link, and you can read the messages on any device.

1. **Requirements**

- Python 3.x
- FastAPI
- Uvicorn

Install packages in VS Code Terminal:

bash

pip install fastapi uvicorn

2. **Project Code**

```python
from fastapi import FastAPI, Form
from fastapi.responses import HTMLResponse

app = FastAPI()
messages = []

@app.get("/", response_class=HTMLResponse)
def form():
    return """
    <form method="post" action="/send">
      <textarea name="msg" placeholder="Type an anonymous message..."></textarea>
      <br>
      <button type="submit">Send Message</button>
    </form>
    """

@app.post("/send")
def send(msg: str = Form(...)):
    messages.append(msg)
    return {"status": "Message sent anonymously!"}

@app.get("/inbox")
def inbox():
    return {"messages": messages}
```

3. **Running the Server**

To accept access from your mobile or other devices:

bash

uvicorn main:app --host 0.0.0.0 --port 8000

or to run within same IP address

uvicorn main:app --reload

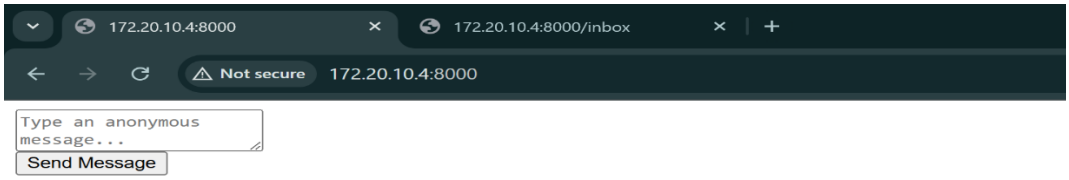
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

                                                    uvicorn main:app --host 0.0.0.0 --port 8000
>> C:\Users\veere\Desktop\manish resume\Ananymous Chat bot>
INFO:     Started server process [13948]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     172.20.10.4:50273 - "GET /inbox HTTP/1.1" 200 OK
INFO:     172.20.10.4:50273 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO:     172.20.10.4:50275 - "GET / HTTP/1.1" 200 OK
INFO:     172.20.10.4:50276 - "POST /send HTTP/1.1" 200 OK
INFO:     172.20.10.4:50277 - "GET /inbox HTTP/1.1" 200 OK
INFO:     172.20.10.4:50278 - "GET / HTTP/1.1" 200 OK
INFO:     172.20.10.4:50341 - "GET / HTTP/1.1" 200 OK
INFO:     172.20.10.4:50348 - "POST /send HTTP/1.1" 200 OK
INFO:     172.20.10.4:50349 - "GET /inbox HTTP/1.1" 200 OK
INFO:     172.20.10.4:50360 - "GET / HTTP/1.1" 200 OK
```

4. **Accessing the App**

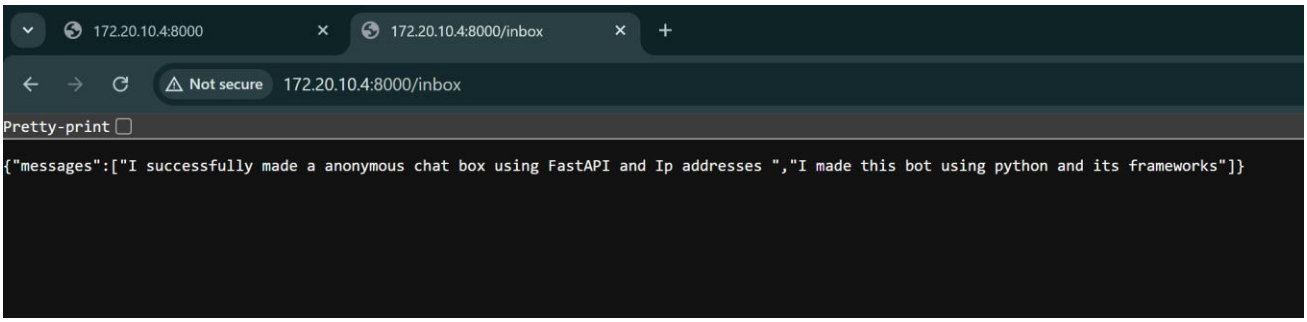
On Your Local Network

1. **Get your PC's IP:**
    - On Windows, run ipconfig and note the IPv4 address.
    - On Mac, run ifconfig or check Network settings.
2. **On mobile or any device:**
    - To send a message: Go to
      http://<your_pc_ip>:8000/
      Example: http://192.168.1.5:8000/




4:12
4G
80
...0b-fb43-d731.a.free.pinggy.link
Type an anonymous message...
Send Message
17

- To read messages:
  http://<your_pc_ip>:8000/inbox



Both devices must be on the same Wi-Fi for this to work.

Across Different Networks

To receive messages from anywhere (outside your local Wi-Fi):

- Use a tunneling service like **Pinggy**:

text

ssh -p 443 -R0:localhost:8000 qr@a.pinggy.io

Use the generated link to access from any network.

5. **Reading and Sending Messages**

- **Sending:** Open the root URL on any device in a browser, type your message, and submit.
- **Reading:** Open /inbox in any browser to see all anonymous messages.

6. **Notes & Considerations**

- **Persistence:** The provided example stores messages in memory (they reset if you restart your app). For production, use a real database.
- **Security:** /inbox is publicly accessible to anyone with the link. Add password or authentication for privacy.
- **Deployment:** For global access, deploy on Heroku/Render/AWS or use a persistent tunnel.
- **Firewall:** Make sure your PC's firewall allows inbound traffic on port 8000 for LAN access.

7. **Project Extensions**

- Add notification on message receipt (email, Telegram, etc.).
- Store messages in SQLite/PostgreSQL.
- Deploy to a cloud platform for permanent availability.

This project is a foundation for an anonymous message board and can be enhanced for security, persistence, and notification as needed.