AI Chatbot with File Preview

AI Chatbot: Overview & Capabilities

What Is an AI Chatbot?

. An AI chatbot is a software agent that simulates human-like conversation.

- It uses Natural Language Processing (NLP) to understand and interpret user input.
- Machine learning models enable it to generate intelligent, context-aware responses.
- Chatbots can perform tasks like answering questions, summarizing content, translating text, and more.
- They are designed to interact naturally with users through text or voice.

Core Functions

- **Text Understanding**: The chatbot breaks down user input into tokens, identifies intent, and extracts key entities using NLP techniques.
- **Response Generation**: It can respond using either rule-based logic or generative models like GPT-Neo, GPT-3.5, Claude, or Gemini.
- **Knowledge Integration**: The chatbot can fetch external data from APIs such as Wikipedia, LangChain, or custom databases to enrich its responses.
- **Multi-turn Memory**: Advanced chatbots maintain context across multiple exchanges, allowing for follow-up questions and coherent conversations.

Use Cases

- **Education**: Assists students by answering academic questions and explaining concepts interactively.
- **Customer Support**: Automates responses to frequently asked questions and helps route support tickets efficiently.
- **Healthcare**: Guides users through symptom checks, provides health tips, and helps schedule appointments.
- **Productivity**: Summarizes emails, documents, or meeting notes to save time and improve workflow.

Technologies Used

- **Transformers (Hugging Face)**: These are deep learning models used for generating natural language responses.
- **Flask / FastAPI**: Lightweight Python frameworks used to build the backend API that handles user queries.
- **Streamlit / React**: Frontend frameworks that provide a user-friendly interface for chatting and interacting with the bot.

- **Wikipedia API / LangChain**: External knowledge sources that supply factual summaries or document-based answers.

File Preview System – Key Concepts

What Is File Preview?

- File preview allows users to view the contents of a file without fully opening it in its native application.
- It's designed for speed, convenience, and reduced resource usage.
- Common in file managers, cloud platforms, and AI-integrated apps.

How It Works

- When a user selects or hovers over a file, the system fetches a lightweight version of the content.
- For documents, it may extract text or render a simplified view.
- For images, it displays a scaled-down version.
- For PDFs, it may convert pages to text or images for fast rendering.
- Previews are often read-only and optimized for quick loading.

Supported File Types

- Text files: .txt, .docx, .pdf
- Data files: .csv, .xlsx
- Images: .jpg, .png, .jpeg
- Multimedia (in advanced systems): .mp3, .mp4, .wav

Features

- Metadata display: Shows file name, type, and size.
- Multi-file support: Allows previewing several files in one session.
- Scrollable and zoomable views for long or large files.
- Error handling for unsupported or corrupted formats.
- Caching for faster repeat access.

Technologies Used

- pandas for CSV rendering
- docx2txt for DOCX extraction
- PyMuPDF for PDF parsing
- PIL (Python Imaging Library) for image display

- Streamlit or Gradio for frontend integration

Benefits

- Saves time by avoiding full file downloads.
- Improves user experience in document-heavy workflows.
- Enables AI-powered document interaction (e.g., summarization, Q&A).
- Reduces bandwidth and system load.

Here's the complete code for your **AI Chatbot with Wikipedia Integration and File Preview**, combining **Flask** for backend logic and **Streamlit** for frontend interaction. This version is modular, clean, and ready for local deployment or portfolio demos.

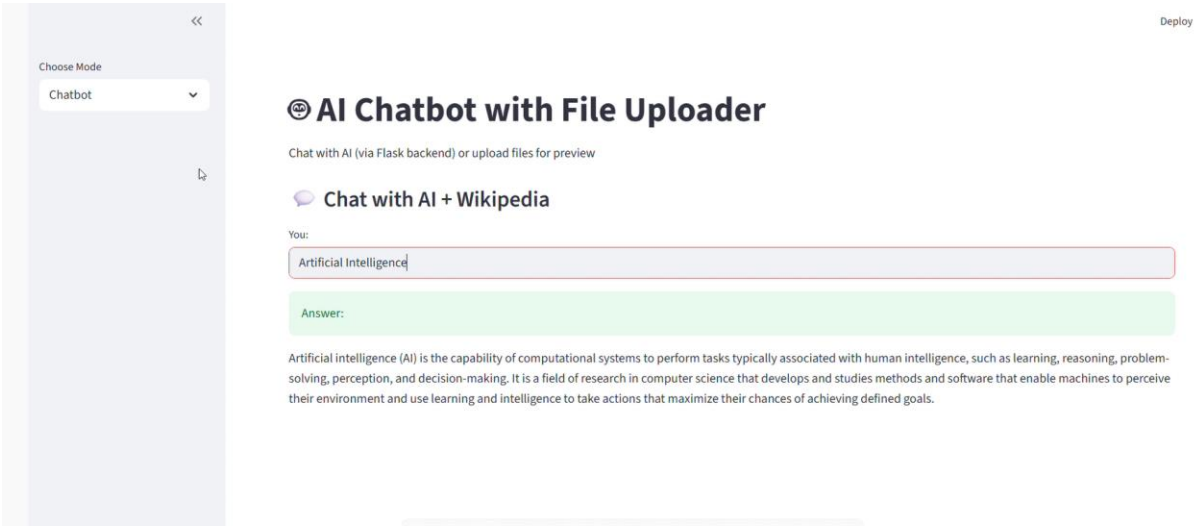
```
import threading
import requests
from flask import Flask, request, jsonify
import streamlit as st
import pandas as pd
from PIL import Image
import docx2txt
import fitz  # PyMuPDF
from transformers import pipeline
import wikipedia


# ------------------ Flask Backend ------------------
app = Flask(__name__)
chatbot = pipeline("text-generation", model="EleutherAI/gpt-neo-125m")
@app.route("/chat", methods=["POST"])
def chat():
    data = request.json
    query = data.get("query", "")
   try:
        wiki_answer = wikipedia.summary(query, sentences=2, auto_suggest=True, redirect=True)
        return jsonify({"answer": wiki_answer})
```

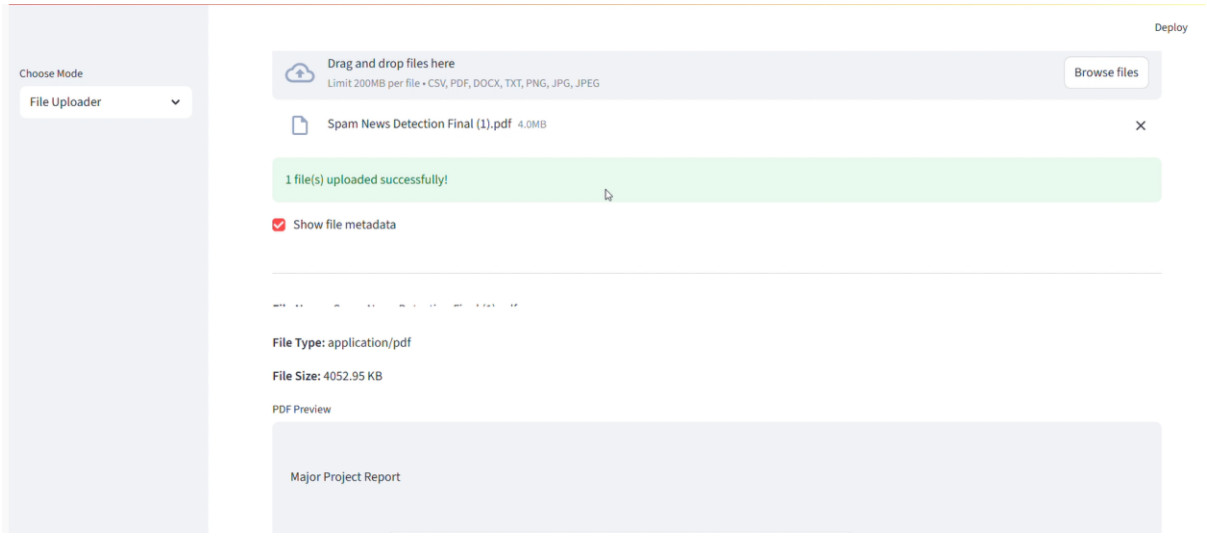
```
    except wikipedia.exceptions.DisambiguationError as e:
        return jsonify({"answer": f"Your query is ambiguous. Try one of these: {e.options[:5]}"})
    except wikipedia.exceptions.PageError:
        pass
    except Exception:
        pass
    res = chatbot(query, max_length=50, do_sample=True, temperature=0.7)
    return jsonify({"answer": res[0]["generated_text"]})
def run_flask():
    app.run(port=5000, debug=False, use_reloader=False)

# ------------------ Streamlit Frontend ------------------
def run_streamlit():
    st.set_page_config(page_title="AI Chatbot + File Uploader", layout="wide")
    st.title("🤠 AI Chatbot with File Uploader")
    st.markdown("Chat with AI (via Flask backend) or upload files for preview")

    app_mode = st.sidebar.selectbox("Choose Mode", ["Chatbot", "File Uploader"])

    # ------------------ Chatbot Mode ------------------
    if app_mode == "Chatbot":
        st.subheader("💬 Chat with AI + Wikipedia")
        user_input = st.text_input("You:", placeholder="Ask me anything...")
        if user_input:
            try:
                res = requests.post("http://localhost:5000/chat", json={"query": user_input})
                if res.status_code == 200:
                    st.success("Answer:")
                    st.write(res.json()["answer"])
                else:
                    st.error("Backend error. Try again.")
```

```
        except Exception as e:
            st.error(f"Error connecting to backend: {e}")

# ------------------ File Uploader Mode ------------------
elif app_mode == "File Uploader":
    st.subheader("🗁 Multi-File Uploader")
    uploaded_files = st.file_uploader(
        "Choose files to upload",
        type=["csv", "pdf", "docx", "txt", "png", "jpg", "jpeg"],
        accept_multiple_files=True,
    )

    if uploaded_files:
        st.success(f"{len(uploaded_files)} file(s) uploaded successfully!")
        show_meta = st.checkbox("Show file metadata")

        for file in uploaded_files:
            st.divider()
            file_name = file.name
            file_type = file.type
            file_size_kb = round(len(file.getvalue()) / 1024, 2)
            file_ext = file_name.split(".")[-1].lower()

            if show_meta:
                st.markdown(f"**File Name:** {file_name}")
                st.markdown(f"**File Type:** {file_type}")
                st.markdown(f"**File Size:** {file_size_kb} KB")

            try:
                if file_ext == "csv":
                    df = pd.read_csv(file)
```

```
                st.dataframe(df)

            elif file_ext == "txt":
                text = file.getvalue().decode("utf-8", errors="ignore")
                st.text_area("TXT Preview", text, height=200)

            elif file_ext == "docx":
                text = docx2txt.process(file)
                st.text_area("DOCX Preview", text, height=200)

            elif file_ext == "pdf":
                pdf = fitz.open(stream=file.read(), filetype="pdf")
                text = ""
                for page in pdf:
                    text += page.get_text()
                st.text_area("PDF Preview", text, height=200)

            elif file_ext in ["png", "jpg", "jpeg"]:
                img = Image.open(file)
                st.image(img, caption=file_name, use_column_width=True)

            else:
                st.warning("Unsupported file format.")
        except Exception as e:
            st.error(f"Error processing {file_name}: {e}")
    else:
        st.info("Please upload one or more files to begin.")
# ------------------ Main Entry ------------------
if __name__ == "__main__":
    threading.Thread(target=run_flask, daemon=True).start()
    run_streamlit()
```

Output:


Deploy
Choose Mode
Chatbot
AI Chatbot with File Uploader
Chat with AI (via Flask backend) or upload files for preview
Chat with AI + Wikipedia
You:
Artificial Intelligence
Answer:
Artificial intelligence (AI) is the capability of computational systems to perform tasks typically associated with human intelligence, such as learning, reasoning, problem-solving, perception, and decision-making. It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.


Deploy
Choose Mode
File Uploader
Drag and drop files here
Limit 200MB per file • CSV, PDF, DOCX, TXT, PNG, JPG, JPEG
Browse files
Spam News Detection Final (1).pdf 4.0MB
1 file(s) uploaded successfully!
Show file metadata
File Type: application/pdf
File Size: 4052.95 KB
PDF Preview
Major Project Report