

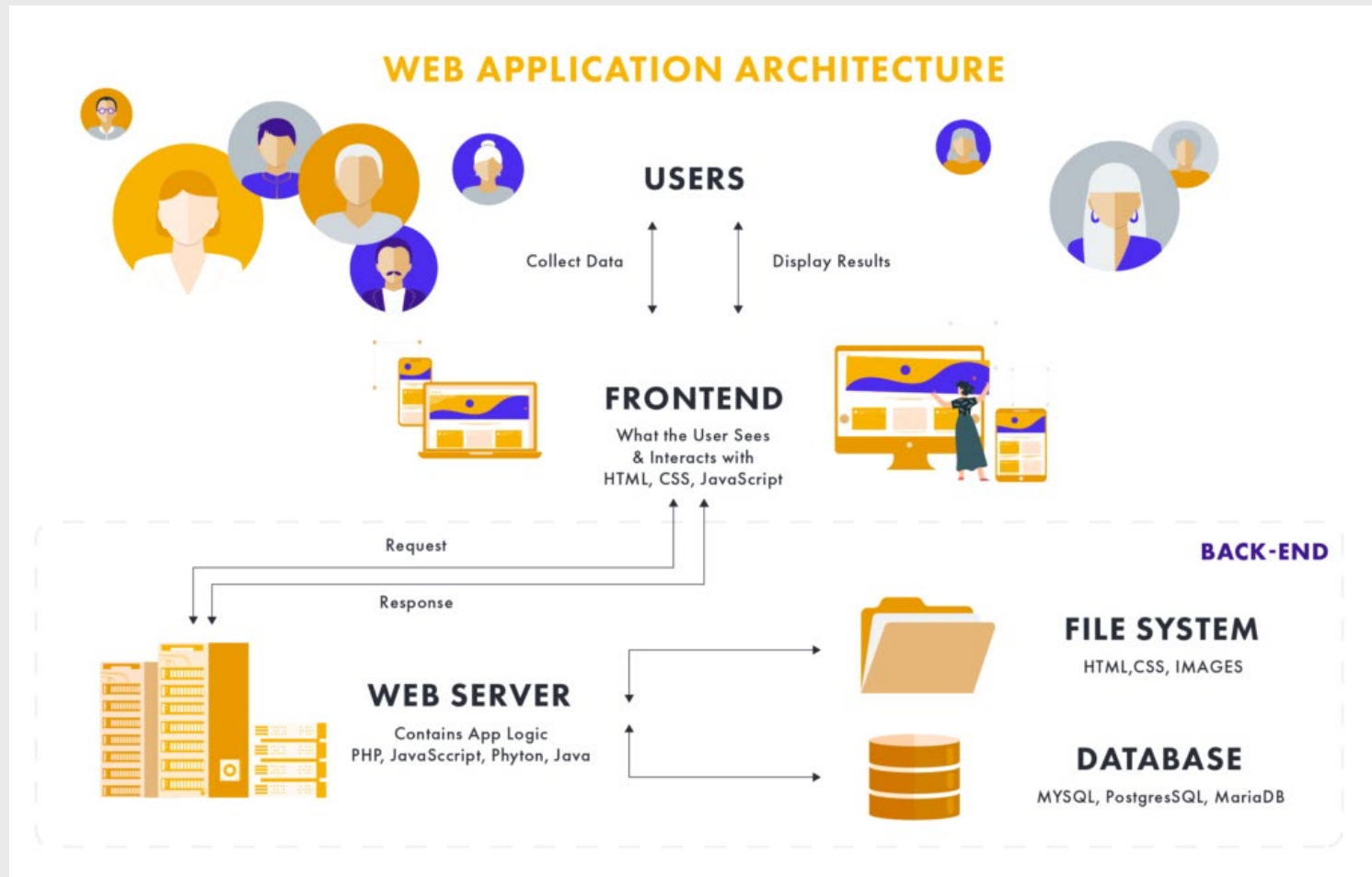
# Building Apps with AI



# **Building Apps with AI**

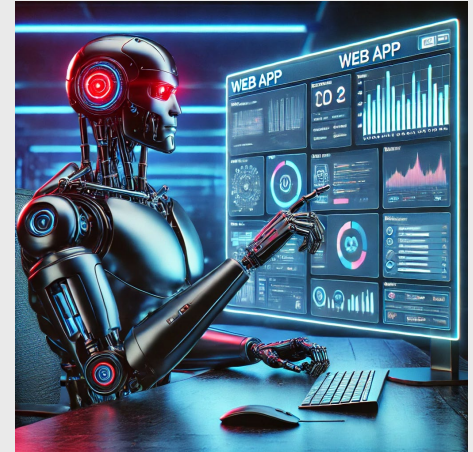
- **We can make apps that do cool things with generative AI**
  - Caption images, video narrations, behavioral profiles, conversational agents
- **We can also use generative AI to build these apps**
  - Detailed coding knowledge not necessary
  - But we need to know the general structure and elements of an app

# App Structure



# Types of Apps

- **Web apps**
  - Run in a web browser
  - Written in HTML, CSS, Javascript, Python
  - Built with frameworks like Streamlit or Flask
  - We will build web apps in this class because they are simple and easy to deploy
- **Mobile apps**
  - Run on a mobile phone
  - Written in Swift (iOS) or Kotlin (Android)
  - Frameworks like React Native or Flutter
  - Require more work to deploy, but not too hard with AI once you have a web version working



# **Web App Frameworks**

- **Web app framework - a pre-built collection of tools, libraries, and structure that simplifies the development of web applications.**
- **Allow one to handle common tasks**
  - Routing (e.g., URLs directing to different pages)
  - Handling user requests (e.g., submitting forms, clicking buttons)
  - Interacting with databases (e.g., storing user data)
  - Rendering front-end elements (e.g., displaying content on the webpage)
- **Why use a framework?**
  - Saves time – no need to build everything from scratch
  - Provides structure – organizes code better
  - Handles complex tasks – security, authentication, etc.

# Python Based Web App Frameworks

- We will be using Python based frameworks to build web apps in class





# Python Based Web App Frameworks

- We will be using Python based frameworks to build web apps in class



# Streamlit vs Flask



- Easy to get started
- No need to write frontend code (everything is Python)
- Front end looks very nice by default
- Not flexible
- This week we will use Streamlit to get started



- Requires a bit more knowledge of coding
- Write python backend and HTML, CSS, and Javascript frontend
- Need to work to make frontend look nice
- Flexible
- Later on, we will move to Flask



# Example Streamlit App

## Twitter User Analysis Dashboard

Upload a CSV file with tweets



Drag and drop file here

Limit 200MB per file • CSV

Browse files



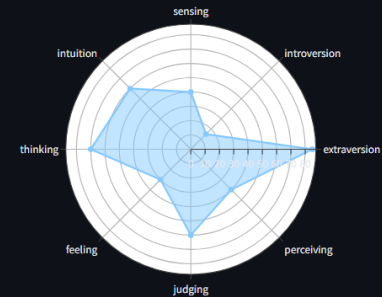
TwExportly\_Cobratate\_tweets\_2024\_04\_19.csv 330.9KB



✓ Analyzing tweets... 🔄

✓ Analyzing personality... 🔄

## Personality Traits Radar Plot



## Topics

topic	summary
Wealth and Personal Responsibility	The user emphasizes the importance of personal effort and responsibility in achieving wealth and success. They argue against complacency and stress that individuals should not rely on external circumstances or blame but take charge of their lives. Wealth is framed as not just a material possession but an avenue for freedom and influence in one's life.
Critique of Modern Society and Institutions	There are strong criticisms of various aspects of modern society, including government actions, media manipulations, and sociocultural norms that are seen as detrimental to true progress. The user believes that significant issues like inflation, societal collapse, and the erosion of traditional values stem from the failure of 'Western democracy' and the influence of 'The Matrix' (a metaphor for systemic societal control).
Charity and Community Support	The user frequently mentions charitable activities and the importance of helping those in need. They express a strong commitment to philanthropy and view it as a responsibility of successful individuals. This is framed as a way to not only aid the less fortunate but also to reflect one's character and values in the community.

## Persona Chat



11:15:38: hey whats up



Hey, it's Andrew. Just living the life, focusing on turning ambitions into reality. What about you?



11:15:58: tell me about the Matrix



✓ AI is thinking... 🔄

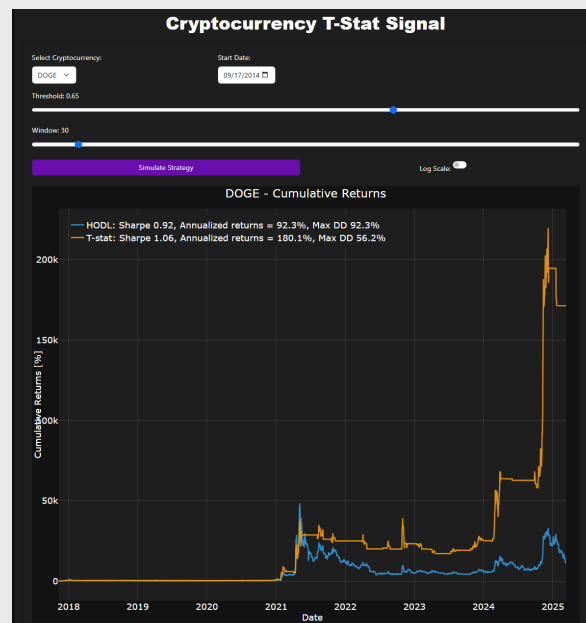
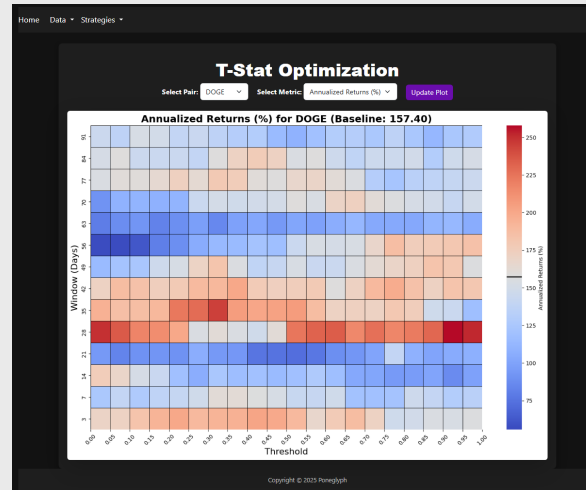
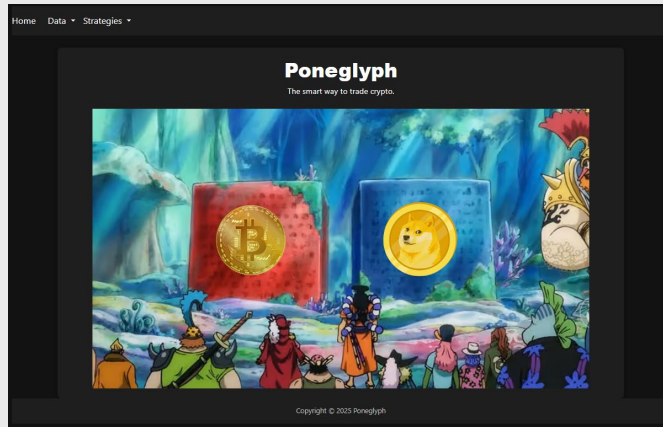


11:16:03: The Matrix is all about the system that keeps most people blind to the truth. It's the societal constructs, the lies we're fed that keep us in a state of complacency. Most people are

Type your message...

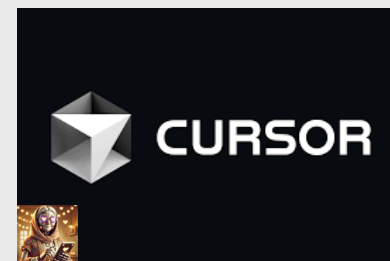


# Example Flask App



# App Maker, App Maker, Make Me an App

- Today there are many AI-powered app makers
  - Require (almost) zero coding knowledge
  - Cost money \$\$\$ (minimum \$20 per month, but can go higher depending on use)
  - Provide an AI agent you tell what app to build, and it builds it, fixes errors, and lets you test it
- These app makers are basically an LLM inside a nice wrapper
  - We already saw something like this in Colab

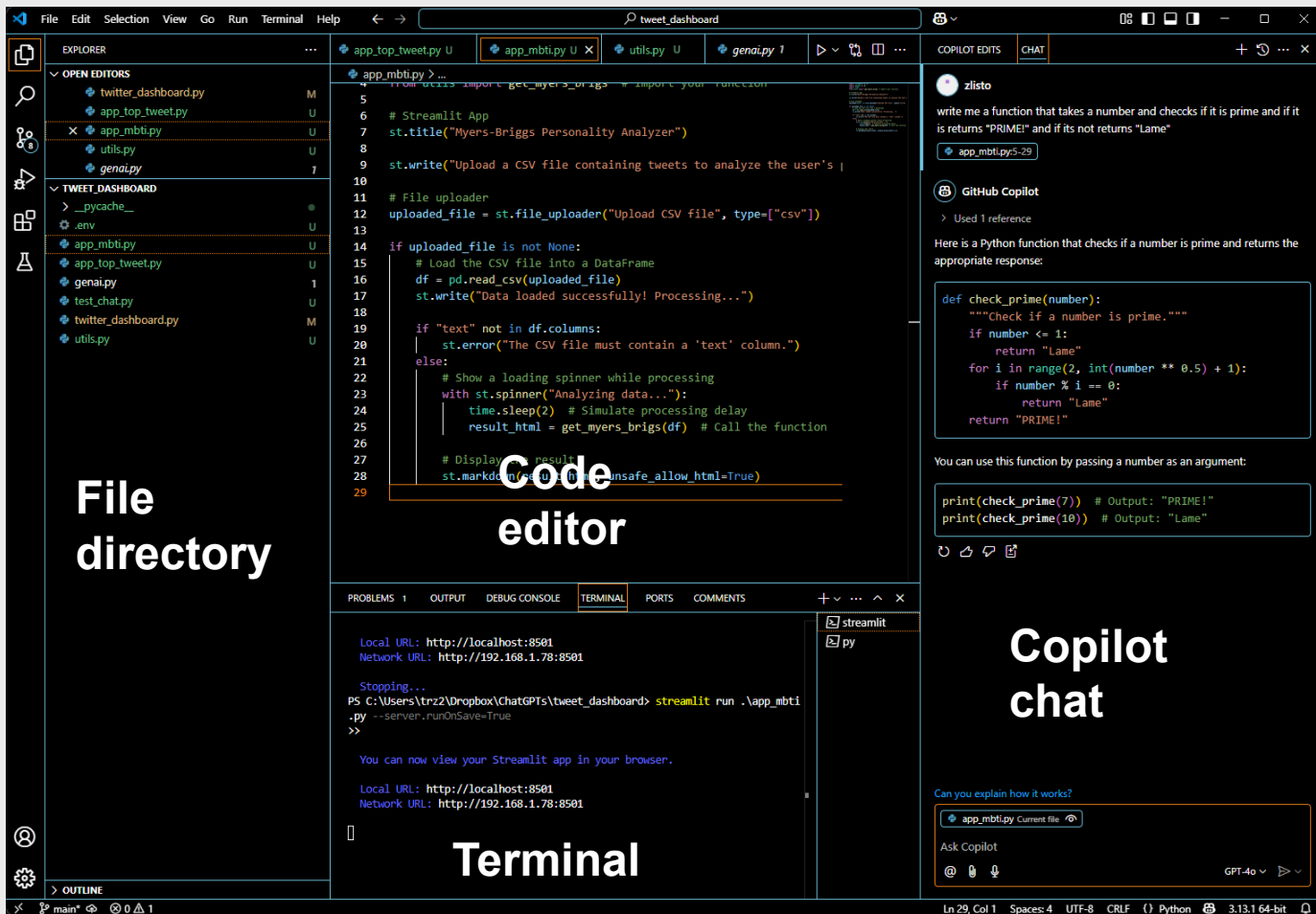




# Visual Studio Code

- **We will build apps in Visual Studio (VS) Code**
  - Free, open-source code editor made by Microsoft
  - Swiss Army knife of coding — lightweight, customizable, and works with almost any language.
  - As of 2025, VS Code is the most popular code editor in the world
- **VS Code has a built in AI assistant called Copilot**
  - Sort of like the AI assistant in Colab
  - Costs money, but Yale provides it for free
  - You need to create a GitHub account first and follow some steps

# VS Code Interface



# **VS Code Installation**

- **Steps for setting up VS Code for making Python based apps**
  1. **Install Anaconda - We need Anaconda because it is a Python distribution**
  2. **Install VS Code**
  3. **Connect VS Code to Anaconda - VS Code needs to know where your Python distribution is**
  4. **Activate Copilot - Not necessary, but makes life a lot easier**
- **Instructions and office hours will be provided by the human TAs**

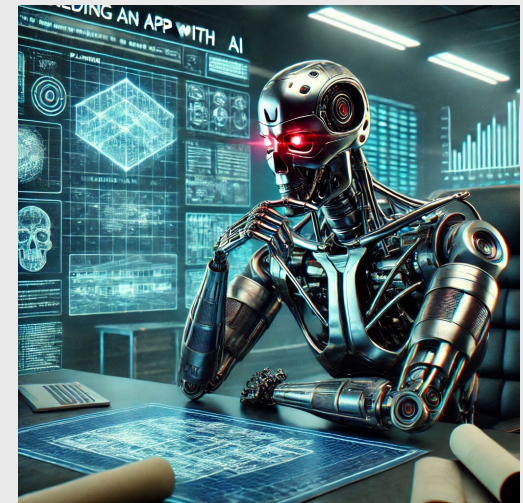


**APP MAKER APP MAKER  
MAKE ME AN APP**



# Building an App with AI: Planning

- The key to making good apps with AI is to plan ahead of time what you want in the app
  - The more detail and structure you include, the better the AI will be at writing the app code
- Detailed text description of the app
  - What it does or the user experience with the app
  - What framework you want to use
  - What data you will have
  - What custom functions you need (or already have) and what their inputs and outputs are
  - What the frontend looks like



# Top Tweets App

- **Let's make an app where I can upload a file of tweets and then display the top engaging tweets in a table**
- **AI prompt:** App maker app maker, make me an app. I would like to make a Streamlit app to analyze tweets. The tweets are in a csv file with columns text, favorite\_count, view\_count. I want to drag the file into the webpage. Then it will be loaded into a dataframe df. Then I want to calculate a new column engagement which is  $\text{favorite\_count} / \text{view\_count}$ . Then I want to rank the tweets by engagement in descending order. Then I want to display the tweet text and engagement in a nice table that I can scroll up and down. I want the highest engagement tweet on top. Each time I upload a new csv file, recompute everything. I want the app to be titled "Top Tweets"

# Ask ChatGPT

ChatGPT 4o ▾

App maker app maker, make me an app. I would like to make a Streamlit app to analyze tweets. The tweets are in a csv file with columns text, favorite\_count, view\_count. I want to drag the file into the webpage. Then it will be loaded into a dataframe df. Then I want to calculate a new column engagement which is favorite\_count/view\_count. Then I want to rank the tweets by engagement in descending order. Then I want to display the tweet text and engagement in a nice table that I can scroll up and down. I want the highest engagement tweet on top. Each time I upload a new csv file, recompute everything. I want the app to be titled "Top Tweets"

Here's a full **Streamlit app** that does exactly what you described. It lets you drag and drop a CSV, calculates tweet engagement, sorts them, and displays a scrollable table. The app recomputes everything each time a new file is uploaded.

You can copy and paste this into a file like `top_tweets.py` and run it with `streamlit run top_tweets.py`.

# AI Generated Code

```
python Copy Edit

import streamlit as st
import pandas as pd

# Set the app title
st.title("Top Tweets")

# Upload CSV file
uploaded_file = st.file_uploader("Upload a CSV file with tweet data", type="csv")

if uploaded_file is not None:
    # Read the uploaded file into a DataFrame
    df = pd.read_csv(uploaded_file)

    # Check for required columns
    required_columns = {'text', 'favorite_count', 'view_count'}
    if not required_columns.issubset(df.columns):
        st.error(f"CSV must contain the columns: {', '.join(required_columns)}")
    else:
        # Avoid division by zero and calculate engagement
        df['engagement'] = df['favorite_count'] / df['view_count'].replace(0, pd.NA)

        # Sort by engagement in descending order
        df_sorted = df.sort_values(by='engagement', ascending=False)

        # Select columns to display
        display_df = df_sorted[['text', 'engagement']]

        # Format engagement nicely
        display_df['engagement'] = display_df['engagement'].round(4)

        # Scrollable table
        st.dataframe(display_df, use_container_width=True, height=600)
else:
    st.info("Please upload a CSV file to get started.")
```

# Working App

## Top Tweets ↗

Upload a CSV file with tweet data



Drag and drop file here

Limit 200MB per file • CSV

Browse files



TwExportly\_1future\_tweets\_2024\_02\_27.csv 280.5KB



	text	engagement
677	Shipped from Latvia? wow i love Latvian culture. <a href="https://t.co/mz0NzLAffY">https://t.co/mz0NzLAffY</a> #Latvia #fentan	0.023
108	In Western films set in Africa, misrepresentations of spoken language can harmfully homog	0.0219
18	.@1future my brother for life we started this shit together and we gon 4sho keep it goin #fo	0.0217
0	Happy Queen Day to my Mother..The GOAT ❤️ 🏆	0.0216
2	Very grateful for everyone who played some type of role in my career up to this point...moi	0.0195
13	Motivate the family Motivate the streets	0.0174
3	New year - New ventures. Starting my new management company 2024.	0.0166
84	Congrats @KingJames LEGENDARY	0.0162
14	Cut off all the bullshit people and get super focus 🏆	0.0156
25	MIXTAPE PLUTO	0.0152
12	We already WON	0.0148
11	Bosses tapin in wit Bosses	0.0147
85	Im in album mode.	0.0146
21	BUSINESS IS BUSINESS	0.0143
82	Bron broke the scoring record! *Alexa Play IM DAT NI\$\$@ #23	0.014
4	Happy boss day PRINCE ❤️ 🏆	0.0121

# Custom Functions

- To make things simpler, you can write custom functions that process the data, and have the main app code call these functions
  - For instance, the AI may not know you already have a nice `jarvis` object that does AI stuff for you
- Put the functions in a file called `utils.py`
- When telling the AI to use the function, tell it its in the `utils.py` file, its name, its inputs, and its outputs

# Personality Assessment App

- Let's make an app where I can upload a file of tweets of a user and then perform a Myers Briggs personality assessment on them
- **AI prompt:** App maker app maker, make me an app. I would like to make a Streamlit app to compute the Myers Briggs category of a person based on their tweets. The tweets are in a csv file with columns text. I want to drag the file into the webpage. Then it will be loaded into a dataframe df. **Then I will pass the dataframe to a function I wrote in my utils.py file called `get_myers_brigs(df)` that returns an HTML string summarizing the person's Myers Briggs type.** I want to display that HTML. **While the `get_myers_brigs` is working, show some spinning wheel so I know its busy thinking and say "Analyzing data..."** Redo the computation each time I upload a new csv file. Give the app a cool sounding name to impress VCs



# Ask ChatGPT

ChatGPT 4o ▾



App maker app maker, make me an app. I would like to make a Streamlit app to compute the Myers Briggs category of a person based on their tweets. The tweets are in a csv file with columns text. I want to drag the file into the webpage. Then it will be loaded into a dataframe df. Then I will pass the dataframe to a function I wrote in my utils.py file called `get_myers_brigs(df)` that returns an HTML string summarizing the person's Myers Briggs type. I want to display that HTML. While the `get_myers_brigs` is working, show some spinning wheel so I know its busy thinking and say "Analyzing data..." Redo the computation each time I upload a new csv file. Give the app a cool sounding name to impress VCs

Absolutely, app-maker mode **activated** 🧙🔥

Here's a slick **Streamlit** app that does exactly what you're asking for — including file upload, analysis via

`get_myers_brigs(df)`, and dynamic re-computation. I've also given it a VC-worthy name:

"PersonaScope" 🧐

# AI Generated Code

Directory Structure

Make sure your project looks something like this:

personascope/  
├─ app.py  
└─ utils.py

Copy Edit

✓ app.py

python

Copy Edit

```
import streamlit as st
import pandas as pd
from utils import get_myers_brigs

st.set_page_config(page_title="PersonaScope", layout="centered")

st.title("🌸 PersonaScope")
st.subheader("Decode your digital soul through your tweets")

uploaded_file = st.file_uploader("Upload your tweets CSV file", type=["csv"])

if uploaded_file is not None:
    try:
        df = pd.read_csv(uploaded_file)
        if "text" not in df.columns:
            st.error("CSV must have a column named 'text'")
        else:
            with st.spinner("🔄 Analyzing data..."):
                result_html = get_myers_brigs(df)
            st.success("Done! Here's what we found:")
            st.components.v1.html(result_html, height=500, scrolling=True)
    except Exception as e:
        st.error(f"Something went wrong: {e}")
```

# Working App



**PersonaScope** ↔

## Decode your digital soul through your tweets

Upload your tweets CSV file



Drag and drop file here

Limit 200MB per file • CSV

Browse files



TwExportly\_elonmusk\_tweets\_2025\_01\_23.csv 297.3KB



Done! Here's what we found:

### Myers-Briggs Personality Profile

**MBTI Type: ENTJ**

#### Personality Summary:

The user exhibits a strong proclivity towards an ENTJ personality type, characterized by Extraversion, Intuition, Thinking, and Judging dimensions. Their active engagement with a wide audience on social media, involvement in political discussions, and interest in futuristic technologies suggest an outgoing and expressive personality with a visionary perspective. The tweets show a pattern of bold, decisive communication, valuing efficiency, logic, and structured progression, which aligns with the traits of an ENTJ.

#### Breakdown of Traits:

- **Extraversion (E) vs. Introversion (I):** The user frequently interacts with others on social media, commenting on and retweeting a variety of topics. They demonstrate

# Persona Chat App

- **Let's make an app where I can upload a file of tweets and then have the AI pretend to be that person based on the tweets and I can chat with them**
- **AI prompt:** App maker, app maker, make me an app. I would like to make a Streamlit app to chat with an AI pretending to be a person based on their tweets. The tweets are in a csv file with columns text. I want to drag the file into the webpage. Then it will be loaded into a dataframe df. **Then I want to call a function in my utils.py file called `get_persona_chat_instructions(df)` that will return a string which is the instructions for the persona chat telling the AI how to behave.** Don't show the instructions on the screen as they are very long and contain many tweets. **Then I want a chat window where the AI message is generated by a function `get_ai_response(instructions, messages)` that takes as input the persona chat instruction string and the messages in the chat so far, including what the user just typed, formatted as as list appropriate for OpenAI API. I want the app to show animated ... while the Ai is generating its message.** Show the entire chat history above where the user inputs their message. When they input their message, display it immediately and then show the AI message. I want the app to be titled "Persona Chat"


# Ask ChatGPT

ChatGPT 4o ▾

App maker, app maker, make me an app. I would like to make a Streamlit app to chat with an AI pretending to be a person based on their tweets. The tweets are in a csv file with columns text. I want to drag the file into the webpage. Then it will be loaded into a dataframe df. Then I want to call a function in my utils.py file called `get_persona_chat_instructions(df)` that will return a string which is the instructions for the persona chat telling the AI how to behave. Don't show the instructions on the screen as they are very long and contain many tweets. Then I want a chat window where the AI message is generated by a function `get_ai_response(instructions, messages)` that takes as input the persona chat instruction string and the messages in the chat so far, including what the user just typed, formatted as as list appropriate for OpenAI API. I want the app to show animated ... while the Ai is generating its message. Show the entire chat history above where the user inputs their message. When they input their message, display it immediately and then show the AI message. I want the app to be titled "Persona Chat"

Let's build your **Streamlit** app: "**Persona Chat**", step by step.

This app will:

1. Allow the user to drag and drop a CSV with tweets.
2. Load the tweets into a DataFrame.
3. Generate persona chat instructions using your `utils.py:get_persona_chat_instructions(df)`.
4. Hide the instruction text but use it to guide the AI behavior.
5. Provide a chat interface using `get_ai_response(instructions, messages)`.
6. Show "..." animation while the AI is typing.
7. Display full chat history above the input box. 

# AI Generated App Code

```
app.py

python Copy Edit

import streamlit as st
import pandas as pd
from utils import get_persona_chat_instructions, get_ai_response

st.set_page_config(page_title="Persona Chat", page_icon="🧠")

st.title("🧠 Persona Chat")

# Upload CSV
uploaded_file = st.file_uploader("Upload a CSV of tweets", type=["csv"])

if uploaded_file:
    df = pd.read_csv(uploaded_file)

    # Get persona instructions once
    if "instructions" not in st.session_state:
        st.session_state.instructions = get_persona_chat_instructions(df)

    # Initialize chat messages
    if "messages" not in st.session_state:
        st.session_state.messages = []

    # Show chat history
    for msg in st.session_state.messages:
        with st.chat_message(msg["role"]):
            st.markdown(msg["content"])

    # Input box
    user_input = st.chat_input("Say something to the persona...")

    if user_input:
        # Show user message
        st.chat_message("user").markdown(user_input)
        st.session_state.messages.append({"role": "user", "content": user_input})


        # Placeholder for streaming
        with st.chat_message("assistant"):
            response_placeholder = st.empty()
            response_placeholder.markdown("🤖 ...thinking...")

        # Get AI response
        ai_reply = get_ai_response(st.session_state.instructions, st.session_state.messages)


        # Replace placeholder with final response
        response_placeholder.markdown(ai_reply)

        # Save assistant response
        st.session_state.messages.append({"role": "assistant", "content": ai_reply})
```


# Working App

 **Persona Chat**


Upload a CSV of tweets


 Drag and drop file here  
Limit 200MB per file • CSV


Browse files


 TwExportly\_Cobratate\_tweets\_2024\_04\_19.csv 330.9KB

×

 yo

 GM! It's Andrew, 28, just out here navigating The Real World. What's up with you?

 tell me about the matrix

 The Matrix? It's the system designed to keep you weak and compliant. It thrives on lies, distractions, and chaos. Most people live in this programmed reality, obeying rules that don't benefit them. But the real power lies in breaking free, questioning everything, and fighting for your own path. Embrace your masculinity, take control of your life, and claim your freedom before it slips away. The world is changing, and you need to adapt or get left behind. Are you ready to break free?

Say something to the persona... ➤



# Multi-Page Apps

- **Pages** - more complex apps will have multiple pages where different tasks occur
  - Tell the AI what pages you want and what they are called
  - Tell the AI what tasks occur on each page, functions called, data needed, etc.

# Multi-Page Apps

- **Pages** - more complex apps will have multiple pages where different tasks occur
  - Tell the AI what pages you want and what they are called
  - Tell the AI what tasks occur on each page, functions called, data needed, etc.
- **State variables**
  - If certain data or function outputs are used on multiple pages, you can save them to “state variables” in Streamlit
  - Eliminates need to redo slow computations (like calling up an LLM to write HTML)
  - Syntax: `st.session_state.variable_name = variable`

```
uploaded_file = st.file_uploader("Upload Tweet CSV", type="csv")
if uploaded_file:
    st.session_state.df = pd.read_csv(uploaded_file)
    st.success("Tweet data loaded!")
```

# **Personality Assessment + Persona Chat App**

- **Let's make an app where I can upload a file of tweets and then perform a Myers Briggs personality assessment on them and then have the AI pretend to be that person based on the tweets and I can chat with them**

# AI Prompt

```
App maker, app maker, make me an app. I would like to make a Streamlit app to compute the Myers
Brigg personality of a user based on their tweets, and a persona chat where I can chat with an AI
pretending to be that user based on their tweets. The app should have a cool name that will make
people wanna use it.

# Pages
I want three pages, Homepage, MBTI Profile, Persona Chat

1) Homepage
On the homepage I want to drag the tweet file. The tweets are in a csv file with columns text. Then
it will be loaded into a dataframe df. Then I want to compute some session state variables:

    1) mbti_html - Pass df to a function I wrote in my utils.py file called get_myers_briggs(df) that
    returns an HTML string summarizing the person's Myers Briggs type. Call it mbti_html. While the
    get_myers_briggs(df) is working, show some spinning wheel so I know its busy thinking and say
    "Computing Myers Briggs profile".

    2) instructions_persona - Pass df to a function in my utils.py file called
    get_persona_chat_instructions(df) that will return a string which is the instructions for the
    persona chat telling the AI how to behave. Call it instructions_persona.

    3) messages - the list of messages in the persona chat. this should be reset each time a new csv
    of tweets is uploaded

2) MBTI Profile
On the MBTI Page, I want to display that mbti_html.

3) Persona Chat
On the Persona Chat page I want a chat window with the AI. Don't show instructions_persona on the
screen as they are very long and contain many tweets. The AI message is generated by a function I
wrote in my utils.py file called
get_ai_response(instructions_persona, messages) that takes as input instructions_persona and the
messages in the chat so far, including what the user just typed, formatted as a list appropriated
for OpenAI API. I want the app to show animated ... while the AI is generating its message. Show the
entire chat history above where the user inputs their message. When they input their message,
display it immediately and then show the AI message.
```

# Ask ChatGPT

ChatGPT 4o ▾

App maker, app maker, make me an app. I would like to make a Streamlit app to compute the Myers Brigg personality of a user based on their tweets, and a persona chat where I can chat with an AI pretending to be that user based on their tweets. The app should have a cool name that will make people wanna use it.


# Pages

I want three pages, Homepage, MBTI Profile, Persona Chat

1) Homepage

On the homepage I want to drag the tweet file. The tweets are in a csv file with columns text. Then it will be loaded into a dataframe df. Then I want to compute some session state variables:

1) mbti\_html - Pass df to a function I wrote in my utils.py file called get\_myers\_briggs(df) that returns an HTML string summarizing the person's Mybers Briggs type. Call it mbti\_html. While the get\_myers\_briggs(df) is working, show some spinning wheel so I know its busy thinking and say "Computing Myers Briggs profile".

 Memory updated

Alright App Maker is in the house 🤖 — let's build your Streamlit app: **Twitter Persona**.

Here's the complete Streamlit code scaffold with all three pages and session state management just like you described:

# AI Generated App Code

```
app.py

python Copy Edit

import streamlit as st
import pandas as pd
from utils import get_myers_brigs, get_persona_chat_instructions, get_ai_response

# Set Streamlit config
st.set_page_config(page_title="Twitter Persona", layout="wide")

# Initialize session state
if "df" not in st.session_state:
    st.session_state.df = None
if "mbti_html" not in st.session_state:
    st.session_state.mbti_html = ""
if "instructions_persona" not in st.session_state:
    st.session_state.instructions_persona = ""
if "messages" not in st.session_state:
    st.session_state.messages = []

# Sidebar for navigation
page = st.sidebar.radio("Navigate", ["Homepage", "MBTI Profile", "Persona Chat"])

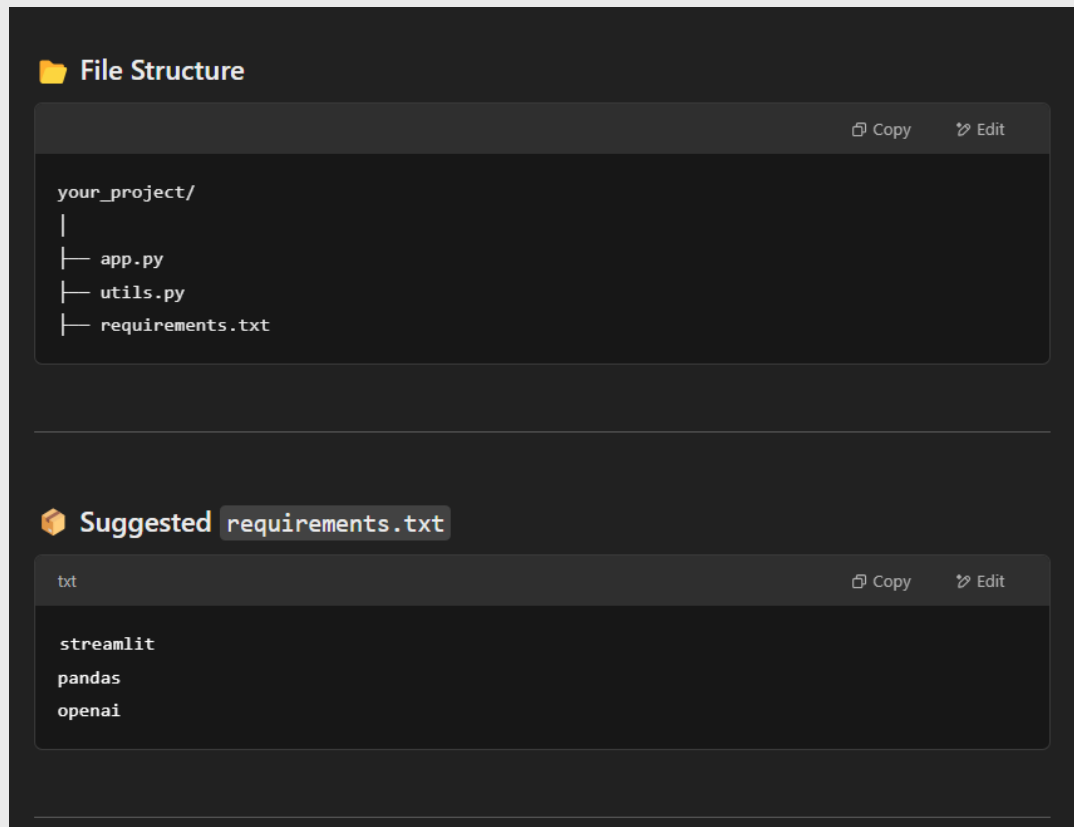
# ----- PAGE: HOMEPAGE -----
if page == "Homepage":
    st.title("🐦 Twitter Persona")
    st.markdown("Upload a CSV of tweets and generate your MBTI profile + AI persona.")

    uploaded_file = st.file_uploader("Upload your tweet CSV file", type=["csv"])

    if uploaded_file is not None:
        df = pd.read_csv(uploaded_file)
        if 'text' not in df.columns:
```

# Requirements File

- This time the AI suggested that I create a requirements.txt file with the packages I need to install.





# Installing Packages

- The AI even told me how to install these packages

how do i install the stuff in a requirements file?

Great question, Honorable Z 🏆


To install everything listed in a `requirements.txt` file, just open your terminal (or command prompt) and run:

bash

Copy

Edit

```
pip install -r requirements.txt
```

 What this does:

It reads all the packages listed in `requirements.txt` and installs them into your environment.

# Working App


Twitter Persona

Go to

Homepage


MBTI Profile

Persona Chat

 **Twitter Persona**

Upload a CSV file of tweets (with a column called `text`) to generate a personality profile and chat with the AI version of the user.

Upload your tweet CSV

 Drag and drop file here  
Limit 200MB per file • CSV

Browse files


 **MBTI Profile**


Myers-Briggs Personality Profile


MBTI Type: **ENTJ**


Personality Summary:


The user exhibits characteristics typical of an ENTJ, often referred to as "The Commander." Known for their leadership qualities and assertive nature, ENTJs are strategic thinkers who thrive in roles that require quick decision-making and vision. The tweets reflect a highly extraverted nature, often engaging with others and displaying confidence in their opinions and beliefs. They show a preference for abstract thinking and future possibilities, aligning well with the intuitive side. Decision-making is primarily informed by logic and rationality, often addressing societal structures and the status quo. The tweets further illustrate an appreciation for organization and control, indicative of a judging preference, while retaining a strong desire for action and purpose.

 **Chat with Twitter Persona**

 yo

 GM! It's Andrew here. What's on your mind?

 tell me about hte matrix



The Matrix is a system, a web of control created by the elites to keep people ignorant and submissive. It feeds off fear, manipulation, and lies. Most people walk through life without even realizing they're trapped in it, too busy consuming mindless entertainment and following societal norms.

The real battle is awakening from this sleep, rejecting the programming, and taking back your power. Once you realize the truth, you can start to fight back—live life on your terms, build wealth, and surround yourself with like-minded individuals. It's about breaking free, growing stronger, and resisting the Slave Mind. Are you ready to fight the Matrix?

# Coding Session

- In our coding session we will build a Twitter dashboard using AI and Streamlit
- App will be built in VS Code (TAs will provide help setting it up)