WEATHER OUTFIT HELPER

By: Rumaysa Abdul Bari

13-08-2025

Table of Content

Introduction

Weather conditions are becoming more and more unpredictable because of climate change, an effect that results primarily from human activities . The changes have resulted in intensified and more frequent weather conditions, complicating the choice of what to wear every day. Today may be sunny and hot, and tomorrow may be too cool or stormy unexpectedly. To assist with this everyday fight, I designed a website known as Outfit Helper. It helps individuals select the ideal outfit depending on temperature and weather, helping it become simpler to adjust to our constantly fluctuating climate

The Outfit Helper is a basic web application that recommends what to wear depending on today's weather in your location or a temperature you specify. It either retrieves the current weather from an API or takes your input and runs a set of rules to advise what to wear on warm, moderate, or chilly days. You may also upload an image of your outfit, preview it, and save it for later use. Developed using Python and Streamlit

Requirements

Physical requirements

- Processor: Intel Core i3 or above / AMD equivalent
- RAM: Minimum 4 GB (8 GB recommended)
- Storage: At least 500 MB free space for Python, libraries, and project files
- Display: Minimum 1280×720 resolution
- Internet Connection: Required for fetching live weather data (optional if using manual input)

Software requirements

- Operating System: Windows 10/11, macOS, or Linux
- Programming Language: Python 3.8 or higher
- Required Python Libraries:
- streamlit – For building the interactive web app
- pandas – For managing and processing data
- requests – For connecting to weather APIs
- Pillow – For handling and displaying images
- Development Tools:
- Code Editor: Visual Studio Code / PyCharm / Notepad++ (VS Code recommended)
- Python package manager: pip (comes with Python)
- Web Browser: Google Chrome, Microsoft Edge, or Mozilla Firefox to view the app

Quick Setup (if physical requirements are met)

If your system meets the above, you can directly install the dependencies using the provided requirements.txt file:

- **pip install -r requirements.txt**

Construction

STEP 1 :Install Python

- Install Python 3.8+ from python.org and make sure Add to PATH is checked. Verify:

python --version
pip --version

STEP 2: Create project folder

- Make a folder named **outfit-helper** and open it in your code editor or File Explorer.

STEP 3: Create project folder

- Add files
- Inside **outfit-helper** create:
- **app.py**
- **requirements.txt**
- (optional) **style.css**
- (optional) **outfits/ folder**

STEP 4: Create & activate virtual environment

- Windows:

cd "path\to\outfit-helper"
python -m venv venv
venv\Scripts\activate

STEP 5: Prepare requirements.txt

- Put this in requirements.txt:

streamlit
requests
pandas
pillow

STEP 6: Install dependencies

pip install -r requirements.txt

STEP 7: Add basic app code

- Open **app.py** and add the Streamlit app code (title, temp input, button, outfit suggestion logic).
- Save the file.

STEP 8:Add features

If you want extras later:

- Add live weather: use an API and requests.
- Add image upload: **st.file_uploader(**) + save to outfits/.
- Add styles: put CSS in style.css and load via st.markdown(..., unsafe_allow_html=True).

STEP 9:Run the app locally

- From the project folder:

  streamlit run app.py

STEP 10:Document & finalize

- Add screenshots and update **README.md** (intro, how to run, requirements, construction, future scope). When ready, initialize git and push to GitHub:

How to run

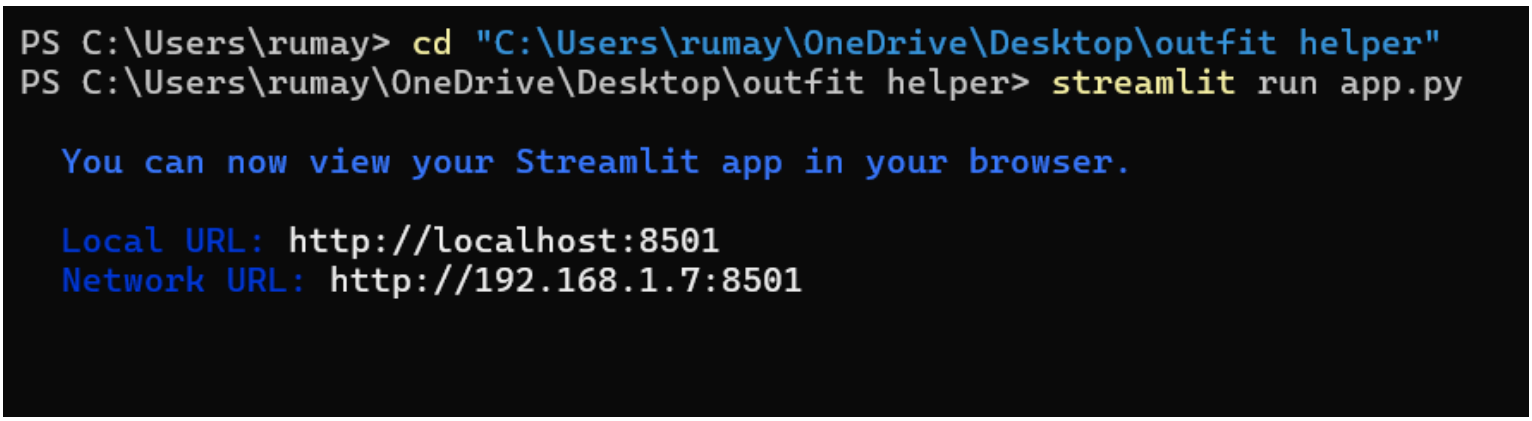
- Install Python (version 3.8 or above) from python.org and make sure it's added to your PATH.
- Download the project folder to your computer.
- Open a terminal/command prompt in the project folder.
- Install the required libraries:

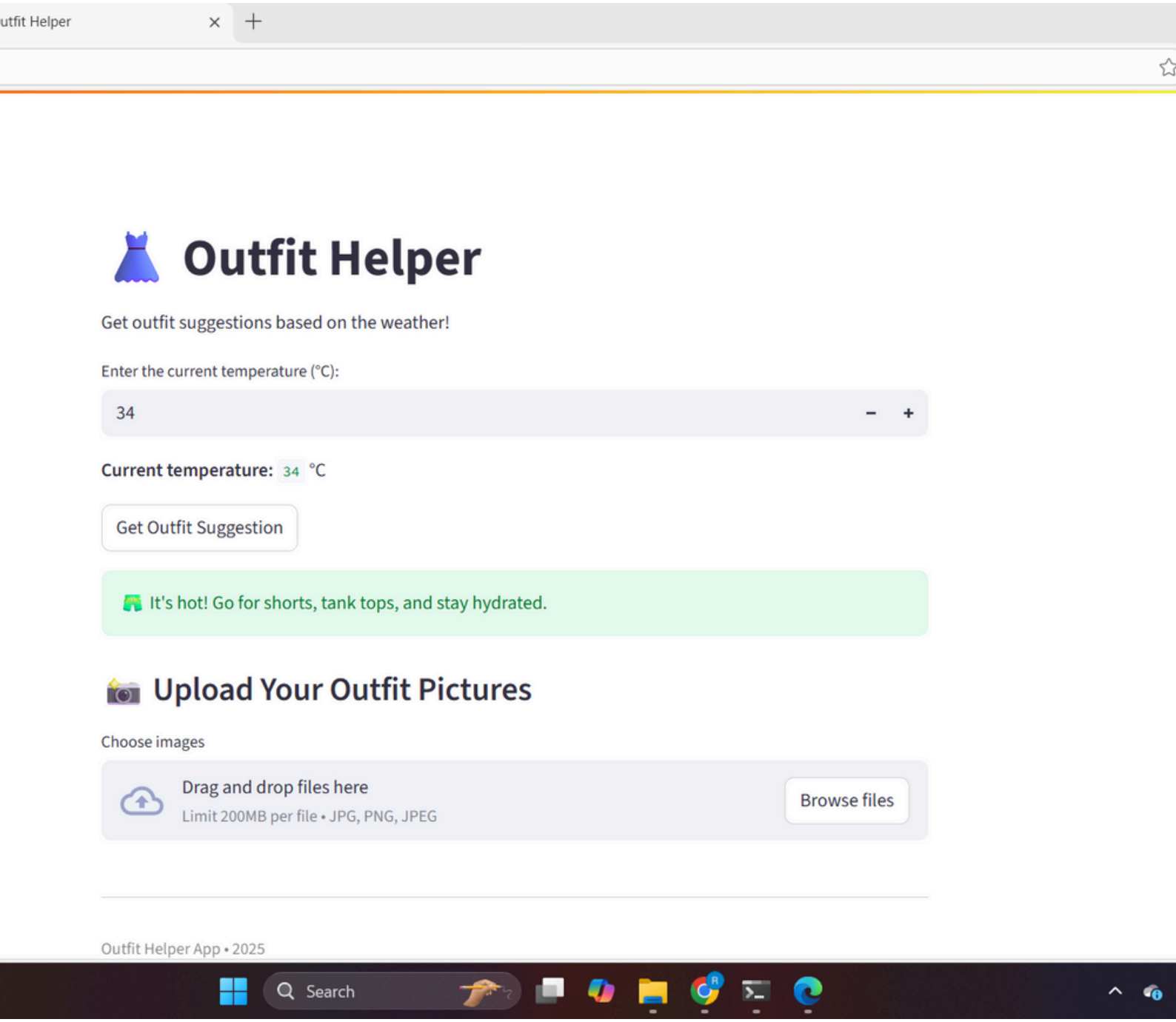
pip install -r requirements.txt

- Start the application:

streamlit run app.py

- The app will open automatically in your default browser





Get Outfit Suggestion

Syntax Code

```
mport streamlit as st
from datetime import datetime
# ---------------------
# APP CONFIG
# ---------------------
st.set_page_config(page_title="Outfit Helper",
page_icon="👗", layout="centered")
st.title("👗 Outfit Helper")
st.write("Get outfit suggestions based on the weather!")
# ---------------------
# WEATHER + OUTFIT LOGIC
# ---------------------
def outfit_tip(temp_c):
 if temp_c < 10:
 return "🧥 It's cold! Wear a coat, scarf, and warm
boots."
 elif 10 <= temp_c < 20:
 return "🧶 Slightly chilly – try a sweater or light
jacket."
 elif 20 <= temp_c < 30:
 return "👕 Warm weather – t-shirt, jeans, or a summer
dress."
 else:
 return "🩳 It's hot! Go for shorts, tank tops, and stay
hydrated."
# ---------------------
# USER INPUT
# ---------------------
temp = st.number_input("Enter the current temperature
(°C):", value=25)
st.write("**Current temperature:**", temp, "°C")
if st.button("Get Outfit Suggestion"):
 st.success(outfit_tip(temp))
```

```
# ---------------------
# OPTIONAL IMAGE UPLOAD
# ---------------------
st.subheader("📸 Upload Your Outfit Pictures")
uploaded_files = st.file_uploader("Choose
images", accept_multiple_files=True, type=
["jpg", "png", "jpeg"])

if uploaded_files:
 for file in uploaded_files:
 st.image(file, caption=file.name,
use_column_width=True)

# ---------------------
# FOOTER
# ---------------------
st.write("---")
st.caption(f"Outfit Helper App •
{datetime.now().year}")
```

Conclusion & Futurescope

The Outfit Helper project showcases how simple technology can be applied to solve everyday problems, such as deciding what to wear based on the current temperature. By combining Python and Streamlit, the application offers an interactive and user-friendly interface that is easy to use for people of all age groups. It not only serves as a beginner-friendly programming project but also provides real-world utility.

Looking ahead, the project has significant scope for enhancement. Future versions could include integration with real-time weather APIs for automatic temperature detection, AI-powered recommendations based on current trends, and a categorized clothing database for different occasions such as formal, casual, or seasonal wear. Additional features like user accounts, style history, and mobile app compatibility can make the tool more personalized and accessible.

In conclusion, the Outfit Helper successfully meets its primary objective of suggesting appropriate clothing according to temperature while providing an enjoyable user experience. With continuous improvements and advanced features, it holds the potential to evolve into a comprehensive digital fashion assistant that blends technology with personal style.

Thank you