Streamlit Exercises • Explanations & Step by Step Guide

Prepared for KCS

This document explains two beginner friendly Streamlit exercises. Each section includes what the app does, prerequisites, step by step instructions to run it, and a brief explanation of how the code works.

Exercise 1: Student Score Dashboard

What it does: Upload a CSV of student marks and view a Maths bar chart and a Science pie chart.

Prerequisites:

- Python 3.8+ installed
- Packages: streamlit, pandas, matplotlib
- A CSV file with at least columns: Name, Maths, Science

Step by Step:

1. Install packages: pip install streamlit pandas matplotlib
2. Create a file student_dashboard.py and paste the code below.
3. Run the app: streamlit run student_dashboard.py
4. Upload your CSV using the uploader and explore the charts.

Sample CSV structure (save as marks.csv):

```
Name,Maths,Science
Asha,88,76
Bala,72,91
Chitra,95,84
Deepak,63,70
```

Code:

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

st.title("Student Score Dashboard")
uploaded_file = st.file_uploader("Upload your CSV file", type=["csv"])

if uploaded_file is not None:
    try:
        df = pd.read_csv(uploaded_file)
        # Basic validation
        needed = {"Name", "Maths", "Science"}
        if not needed.issubset(df.columns):
            st.error(f"CSV must include columns: {needed}")
        else:
            st.write("### Student Marks Data")
            st.dataframe(df)

            st.write("### Maths Scores")
            st.bar_chart(df.set_index("Name")["Maths"])

            st.write("### Science Score Distribution")
            fig, ax = plt.subplots()
            ax.pie(df["Science"], labels=df["Name"], autopct="%1.1f%%")
            ax.axis("equal")
            st.pyplot(fig)
    except Exception as e:
        st.error(f"Could not read file: {e}")
```

How it works:

- st.file_uploader lets a user pick a CSV.
- pandas.read_csv loads the file into a DataFrame.
- st.bar_chart quickly draws the Maths bar chart from the DataFrame.
- The Science pie chart is drawn with matplotlib and shown using st.pyplot.

Note: Pie charts of raw marks may be less informative than bar charts or histograms; choose what suits your data best.

Exercise 2: Mood based Music & GIF Recommender

What it does: Pick a mood and get a fun GIF and a YouTube song recommendation.

Prerequisites:

- Python 3.8+ installed
- Package: streamlit
- Internet connection to load GIFs and YouTube links

Step by Step:

1. Install Streamlit: pip install streamlit
2. Create a file mood_recommender.py and paste the code below.
3. Run the app: streamlit run mood_recommender.py
4. Choose a mood and click **Get Recommendation**.

Code:

```
import streamlit as st
import random

st.set_page_config(page_title="Mood-based Music & GIF Recommender", layout="centered")
st.title("Mood-based Music & GIF Recommender.")
st.markdown("Select your current mood and get a fun GIF or song recommendation!")

# Mood selection
mood = st.selectbox(
    "How are you feeling today?",
    ["Happy", "Sad", "Excited", "Relaxed", "Sleepy"]
)

# Recommendations dictionary
recommendations = {
    "Happy": {
        "GIFs": [
            "https://media.giphy.com/media/3o7TKtnuHOHHUjR38Y/giphy.gif",
            "https://media.giphy.com/media/111ebonMs90YLu/giphy.gif"
        ],
        "Songs": [
            "https://www.youtube.com/watch?v=y6Sxv-sUYtM",
            "https://www.youtube.com/watch?v=ZbZSe6N_BXs"
        ]
    },
    "Sad": {
        "GIFs": [
            "https://media.giphy.com/media/ROF8OQvDmxytW/giphy.gif",
            "https://media.giphy.com/media/3og0IPxMM0erATueVW/giphy.gif"
        ],
        "Songs": [
            "https://www.youtube.com/watch?v=9tT1t3C9Gd8",
            "https://www.youtube.com/watch?v=RB-RcX5DS5A"
        ]
    },
    "Excited": {
        "GIFs": [
            "https://media.giphy.com/media/l0MYt5jPR6QX5pnqM/giphy.gif",
            "https://media.giphy.com/media/xT0xeJpnrWC4XWblEk/giphy.gif"
        ],
        "Songs": [
            "https://www.youtube.com/watch?v=JGwWNGJdvx8",
            "https://www.youtube.com/watch?v=OPf0YbXqDm0"
        ]
    },
    "Relaxed": {
        "GIFs": [
            "https://media.giphy.com/media/l0HlOvJ7yaacpuSas/giphy.gif",
            "https://media.giphy.com/media/3o6Zt481isNVuQI1l6/giphy.gif"
        ],
        "Songs": [
            "https://www.youtube.com/watch?v=2Vv-BfVoq4g",
            "https://www.youtube.com/watch?v=RgKAFK5djSk"
        ]
```

```
        },
        "Sleepy": {
            "GIFs": [
                "https://media.giphy.com/media/l0MYt5jPR6QX5pnqM/giphy.gif",
                "https://media.giphy.com/media/3o7abKhOpu0NwenH3O/giphy.gif"
            ],
            "Songs": [
                "https://www.youtube.com/watch?v=1ZYbU82GVz4",
                "https://www.youtube.com/watch?v=5qap5aO4i9A"


        }
    }

    if st.button("Get Recommendation"):
        gif_choice = random.choice(recommendations[mood]["GIFs"])
        song_choice = random.choice(recommendations[mood]["Songs"])

        st.subheader(f"Recommended GIF for {mood} mood:")
        st.image(gif_choice, width=400)

        st.subheader(f"Recommended Song for {mood} mood:")
        st.markdown(f"[Click here to play the song]({song_choice}) ", unsafe_allow_html=True)
```

How it works:

- st.selectbox captures the mood choice.
- A Python dictionary stores GIF and song URLs for each mood.
- When the button is pressed, random.choice selects one GIF and one song.
- st.image displays the GIF; the song opens via a clickable link.

Optional enhancement:

```
    # Optional: Embed YouTube player instead of a link
    # if "youtube.com/watch" in song_choice:
    #     st.video(song_choice)
    # else:
    #     st.markdown(f"[Click here to play the song]({song_choice})")
```

Tips & Troubleshooting

- If images or videos don’t load, check your internet connection and that the URLs are valid.
- To stop the app, return to the terminal and press Ctrl+C.
- For deployment, explore streamlit share or cloud options such as Streamlit Community Cloud.