Streamlit to Build & Deploy Apps like a Data Scientist

@sidgupta234

Siddharth Gupta (**@sidgupta234** on internet)

Researcher at Potsdam University, Germany

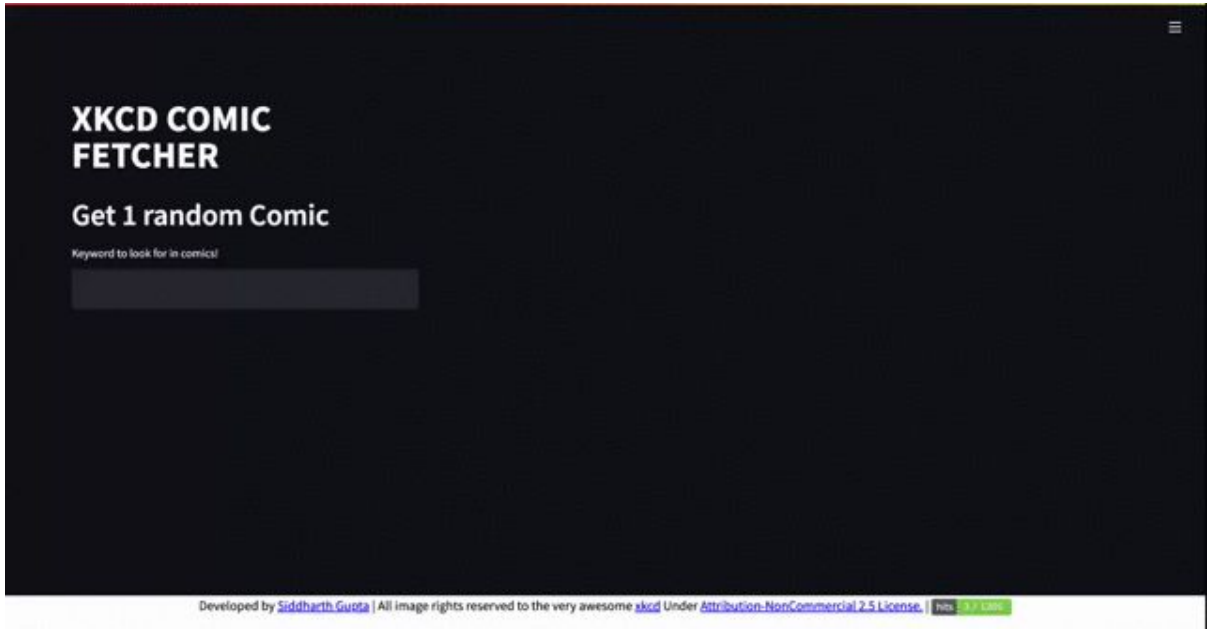
Hobbies include ♫, ♟, ⚽, 🎤 and 😴!

Contents!

- What is Streamlit?
- Why Streamlit?
- Hands on time!
  - Installation and setting up
  - Understanding different features of Streamlit
  - Comic fetcher
  - Lyric fetcher (exercise)
- The know how of deploy(ayay!)

What is Streamlit?

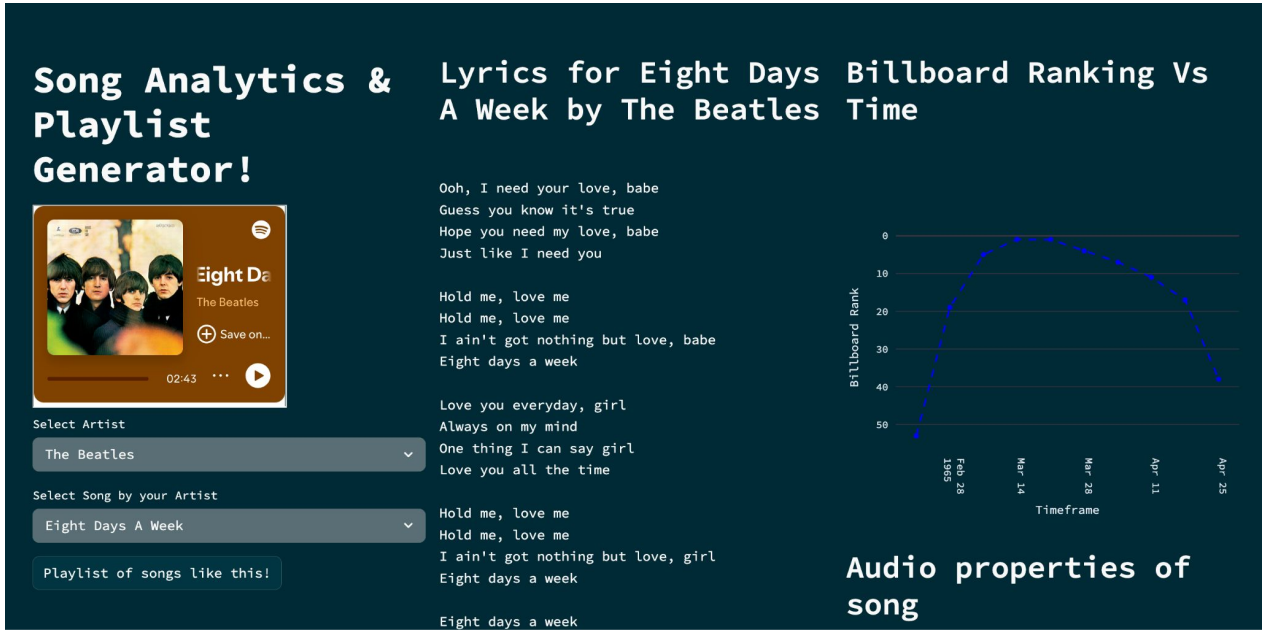
- Open-source Python library
- Makes it easy to make and deploy beautiful web apps for ML and data science
- Build and share data-related apps in no time



Streamlit Eg:https://xkcdcomic.streamlit.app/
code: github.com/sidgupta234/xkcd-comic-fetcher

Streamlit Why?

- **Quick Prototyping:** Build apps fast with just Python
- **Simple API:** Works with Pandas, NumPy, etc.
- **Auto Updates:** Live reload on code changes
- **UI Widgets:** Sliders, uploads, dropdowns—built-in
- **Easy Deployment:** Deploy with Streamlit Cloud



Streamlit Eg:https://playlist-gen.streamlit.app/

Understanding different features of Streamlit

Installation and setting up!

- Go to the repo: tinyurl.com/pystreamlit clone repo
- install streamlit (pip install streamlit), pandas
- run ‘streamlit hello’ to check installation
- check comic_fetcher/**app.py**

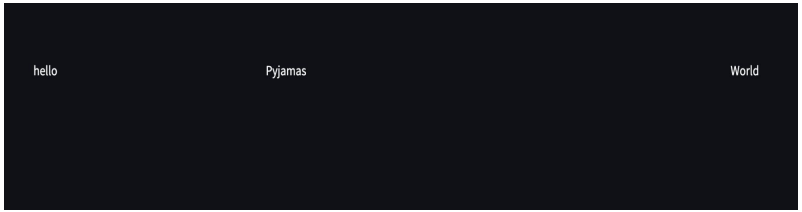
```
import streamlit as st
import pandas as pd
st.set_page_config(page_title='Comic Fetcher',
page_icon = "favicon.png",
layout = 'wide',
initial_sidebar_state = 'auto')
```

Set config page (Code)

Understanding different features of Streamlit

Columns

Page can be divided into columns of different ratios!



Columns (Output)

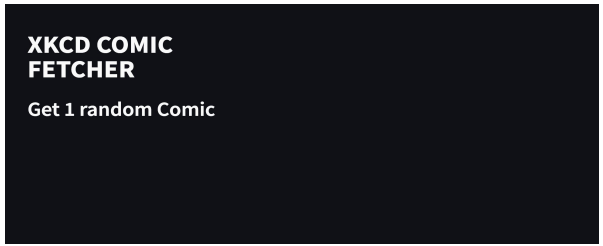
```
c1, c2, c3= st.columns((1, 2, 1))

with c1:
    st.write("hello")

with c2:
    st.write("Austria")

with c3:
    st.write("World")
```

Columns (Code)

Understanding different features of Streamlit

containers

Inserts one invisible container into app, helps with modularizing.



Containers (Output)

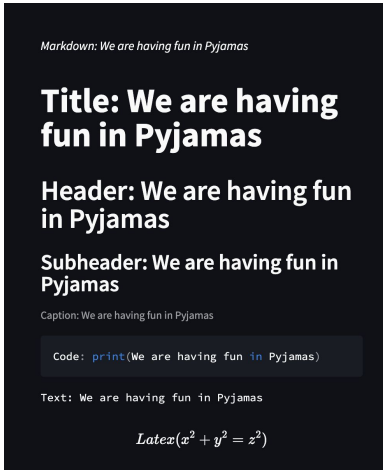
```python
c1, c2= st.columns((1, 2))

header = st.container()

with header and c1:
    st.title("XKCD COMIC FETCHER")
    st.header("Get 1 random Comic")
```

Containers (Code)

Understanding different features of Streamlit

Text display types

Text can be represented in various ways.



Text Display Types (Output)

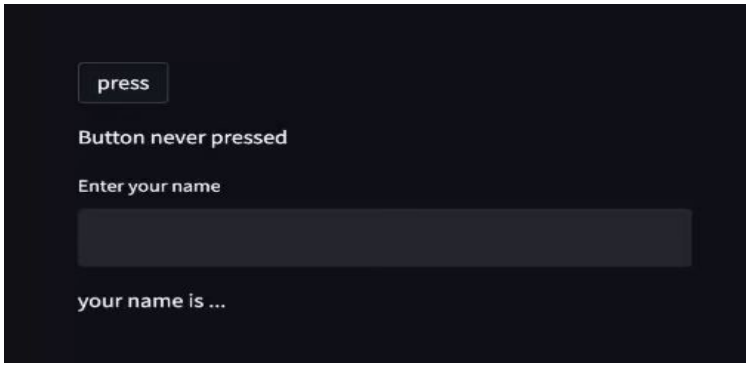
```python
c1, c2 = st.columns((1, 2))

with c1:
    st.markdown("<i>Markdown: We are having fun in Pycon Austria </i>", unsafe_allow_html=True)
    st.title("Title: We are having fun in Pycon Austria")
    st.header("Header: We are having fun in Pycon Austria")
    st.subheader("Subheader: We are having fun in Pycon Austria")
    st.caption("Caption: We are having fun in Pycon Austria")
    st.code("Code: print(We are having fun in Pycon Austria)")
    st.text("Text: We are having fun in Pycon Austria")
    st.latex(r"\text{We}^{\text{are}} + \text{having}^{\text{fun}} = \text{Pycon}^{\text{Austria}}")
```

Text Display Types (Code)

Understanding different features of Streamlit

Input Widgets

Various input widgets for buttons, text input available.



Input Widgets (Output)

```python
c1, c2 = st.columns((1, 2))

with c1:

    if st.button('press'):
        st.write('Button pressed')
    else:
        st.write('Button never pressed')

    text_input = st.text_input("Enter your name")
    st.write("your name is ...", text_input)
```

Input Widget (Code)

Understanding different features of Streamlit

chart elements

Various default chart elements are available. Matplotlib and plotly can be used as well.

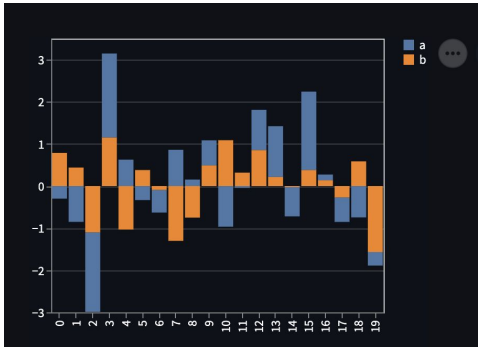


Chart Elements (Output)

```
c1, c2 = st.columns((1, 2))

with c1:

    chart_data = pd.DataFrame(
        np.random.randn(20, 2),
        columns=["a", "b"])

    st.bar_chart(chart_data)
```

Chart Elements (Code)

Understanding different features of Streamlit

Media elements

Media elements such as audio, video and image can be easily shown by streamlit

Media Elements (Output)

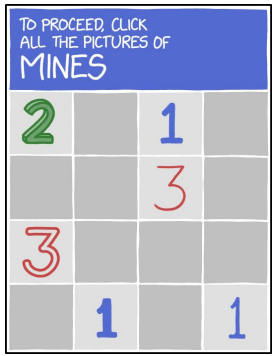
```
c1, c2 = st.columns((1, 2))

with c1:
    st.image("https://i.imgur.com/q5xmsD1.png")
```

Media Elements (Code)

Building XKCD Comic Fetcher

DATASET

Details of 2496 XKCD Comics.



Sample Comic View

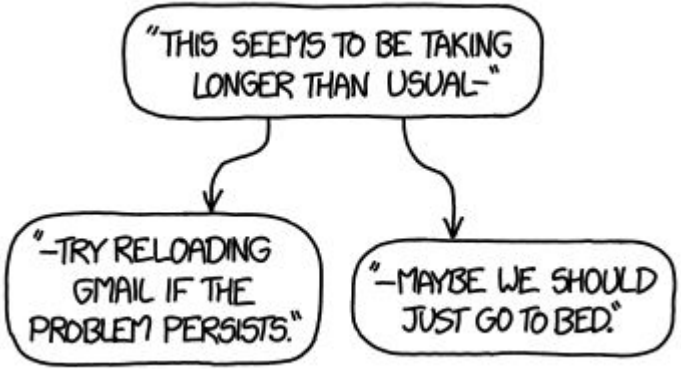

```
"root" :  143 items
  [  100 items
    0 : {  11 items
      "month" : string "7"
      "num" : int 2496
      "link" : string ""
      "year" : string "2021"
      "news" : string ""
      "safe_title" : string "Mine Captcha"
      "transcript" : string ""
      "alt" :
      string "This data is actually going into improving our self-driving car project, so hurry up--it's almost at the minefield."
      "img" : string "https://imgs.xkcd.com/comics/mine_captcha.png"
      "title" : string "Mine Captcha"
      "day" : string "30"
    }
```


Sample Comic Data

Off to Code now ....

Deployment!

- Need to have at 1 file beside **app.py**
  - requirements.txt -> libraries needed to be installed
- Go to streamlit apps and deploy.



Thank you!

Audience!

Pycon Austria <3

Streamlit and their documentation

XKCD Comics