

Day 4: Interactive UI



Learning Objectives

Towards the end of this training, you should be able to:

- Use the Python **streamlit** package to develop an interactive UI.
- Host the **streamlit** page into **streamlit** cloud by using GitHub.

We Need These...



Visual Studio Code



ANACONDA®



Streamlit



Cloud

Gallery

Components

Generative AI

Community Docs

Blog

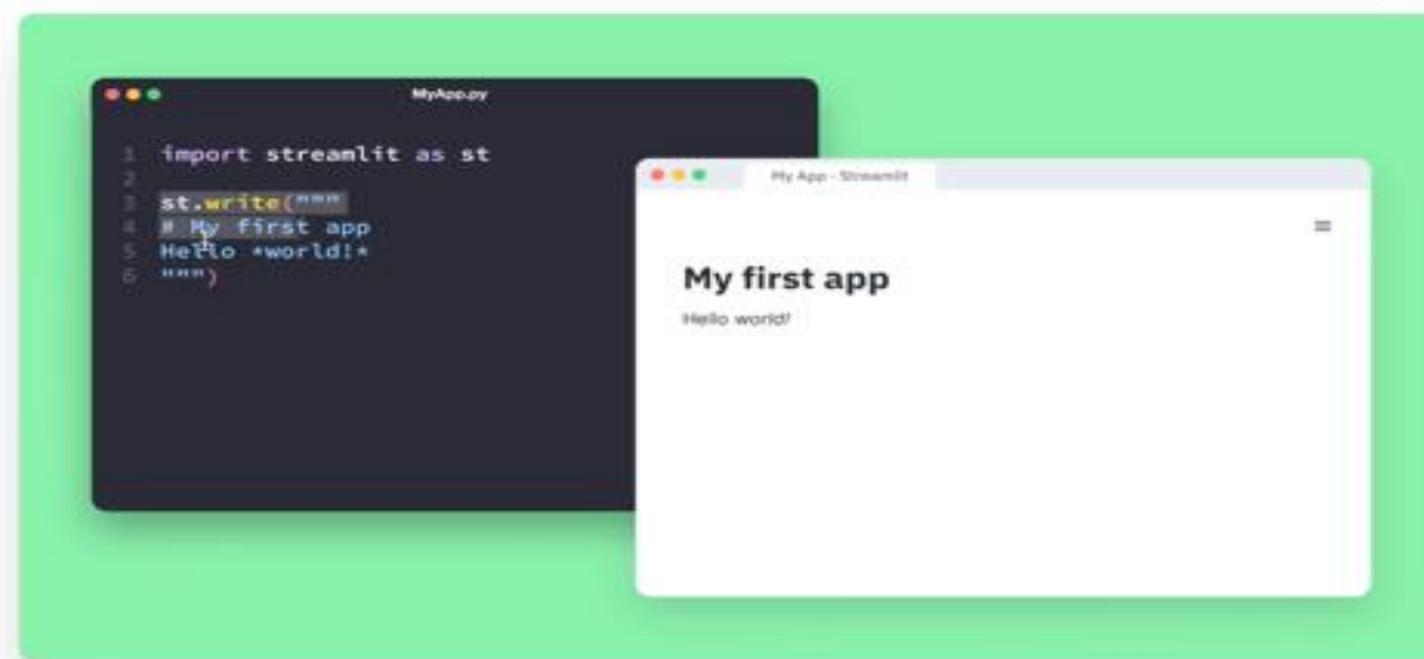
Sign in

Sign up

A faster way to build and share data apps

Streamlit turns data scripts into shareable web apps in minutes.

All in pure Python. No front-end experience required.

[Try Streamlit now](#)[Deploy on Community Cloud \(it's free!\)](#)

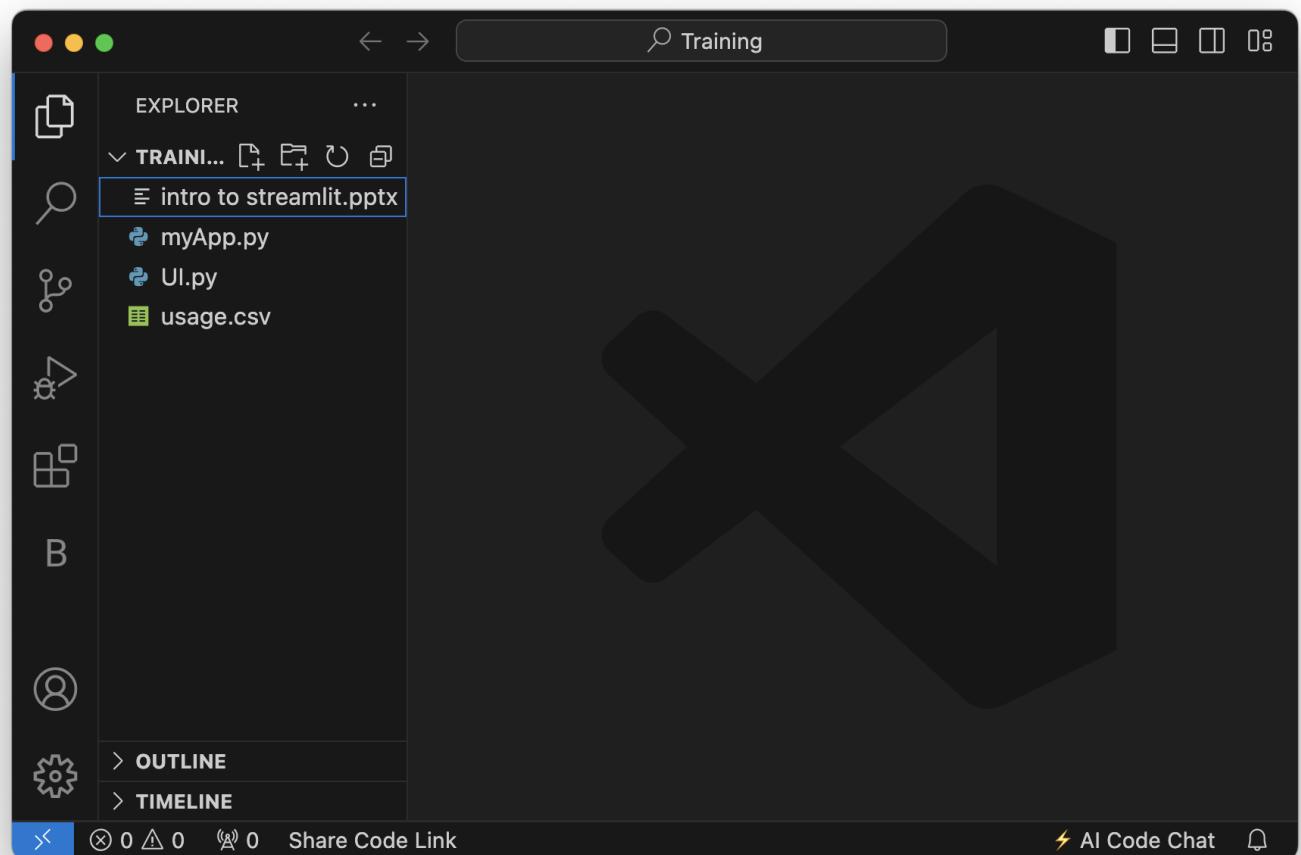
Learn more: Why Generative AI + Streamlit are a perfect match

What is Streamlit?

- Streamlit is a Python package.
- Streamlit is an open source package.
- Streamlit allows creation of shareable web apps fast.
- The web apps developed can be hosted in Heroku.com or **Streamlit** Community Cloud.
- Minimal to No skillset in HTML is required to build a simple web app.

Requirements

- **Streamlit** requires Python or Anaconda
- Python editor such as Visual Studio Code.
- This tutorial is based on:
 - VS Code
 - Anaconda



Install Streamlit

- You can install Streamlit by typing:

> **pip install streamlit**



```
(base) cyting@x86_64-apple-darwin13 ~ % pip install streamlit
```

Creating the First App



Creating a Simple App

- Write the following codes into VS Code and save it as **myApp.py**

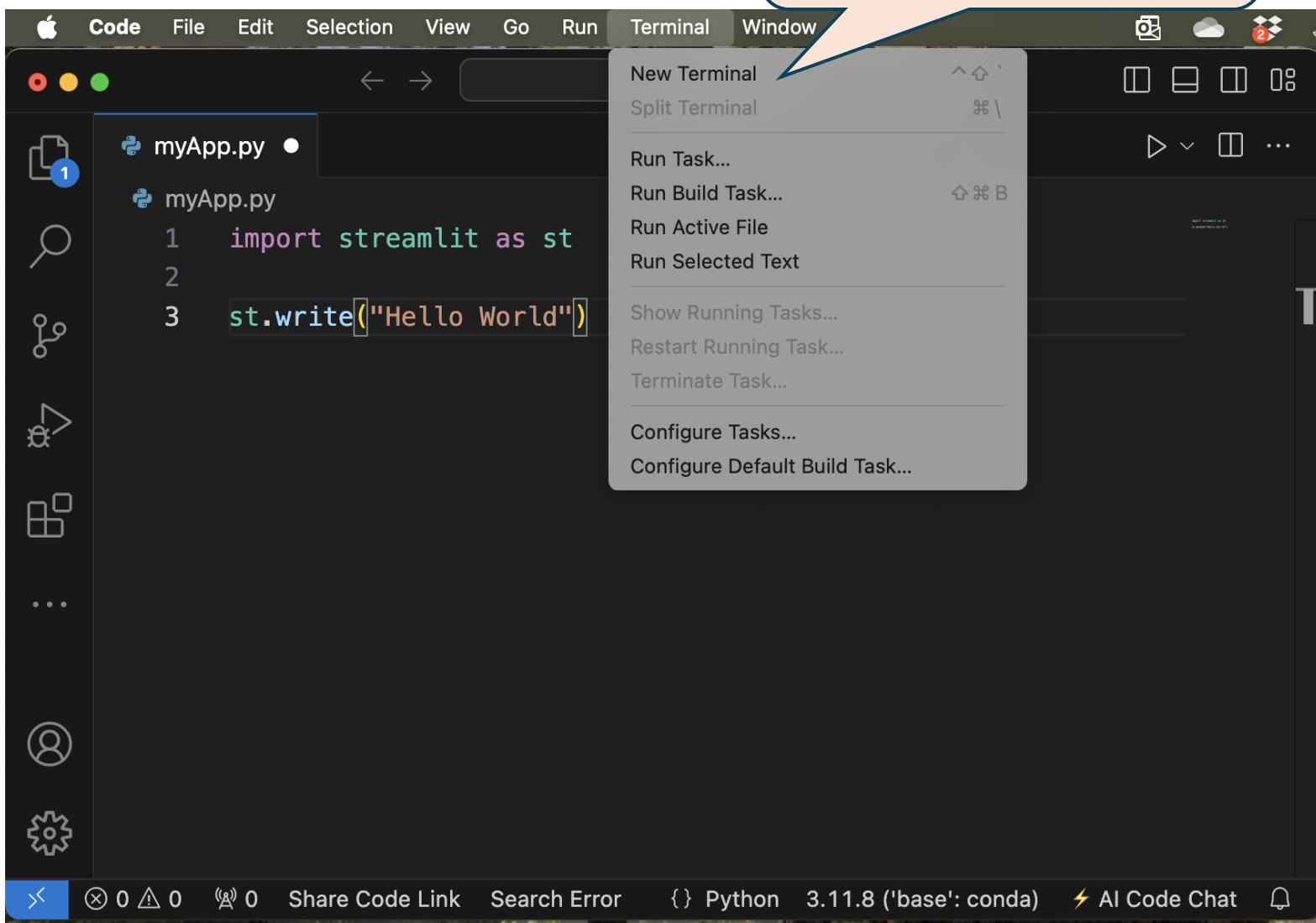


Write a simple **streamlit** code

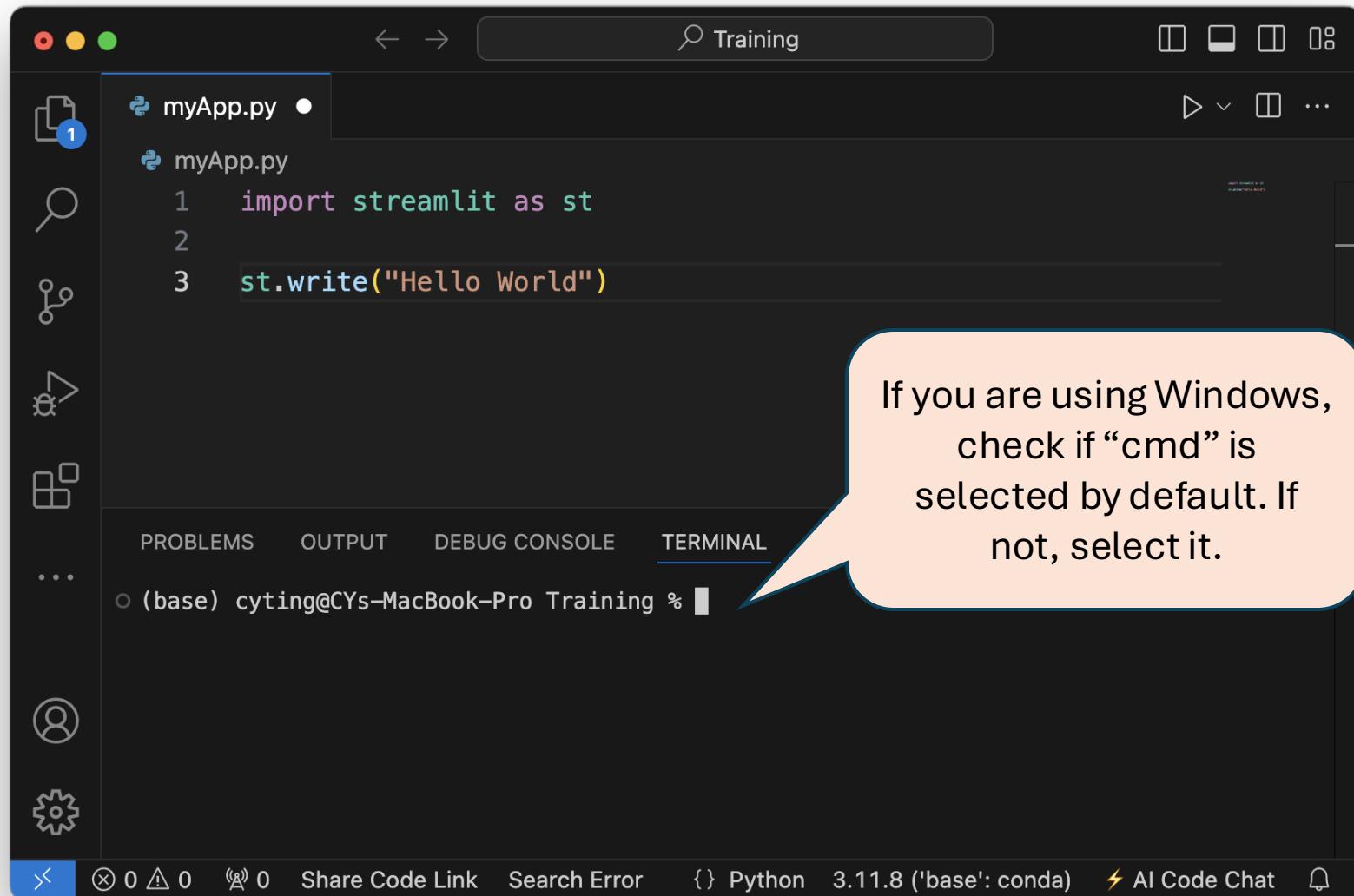
```
import streamlit as st
st.write("Hello World")
```

Let us start with importing the **streamlit** package first.

Creating a Simple App



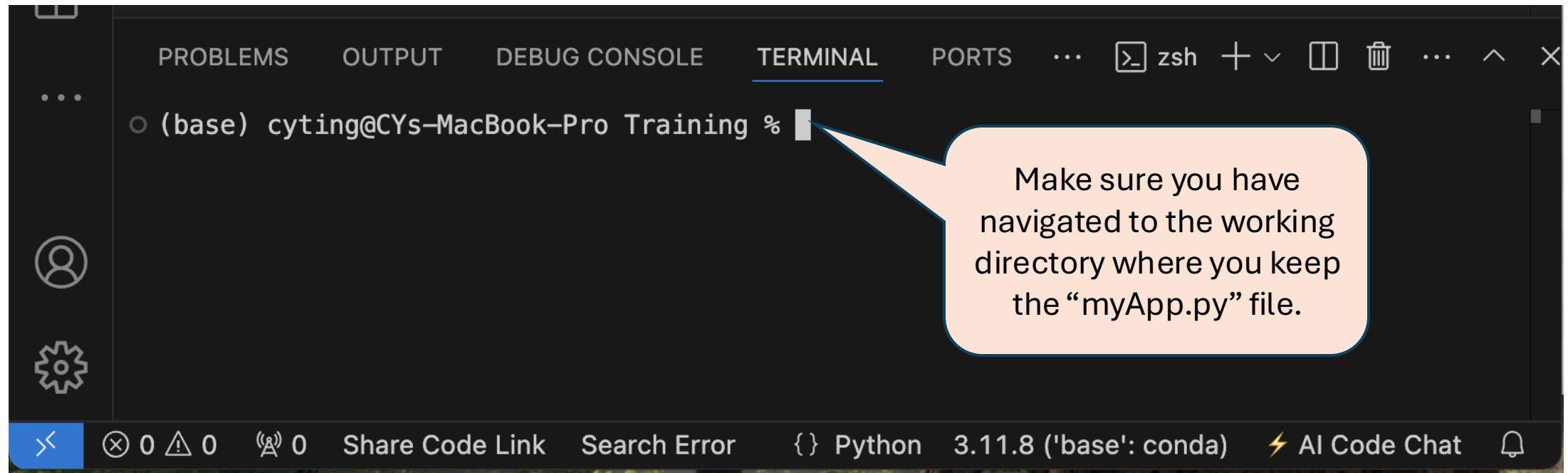
Creating a Simple App



If you are using Windows,
check if “cmd” is
selected by default. If
not, select it.

```
1 import streamlit as st
2
3 st.write("Hello World")
```

Creating a Simple App



Creating a Simple App

Type “**streamlit run myApp.py**” and then press Enter.

A screenshot of the Visual Studio Code (VS Code) interface. On the left is a dark-themed sidebar with various icons for file operations like Open, Save, Find, and Settings. The main workspace shows a single Python file named "myApp.py" with the following code:

```
1 import streamlit as st
2
3 st.write("Hello World")
```

Below the editor is a tab bar with "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The "TERMINAL" tab is active, showing a terminal session with the command:

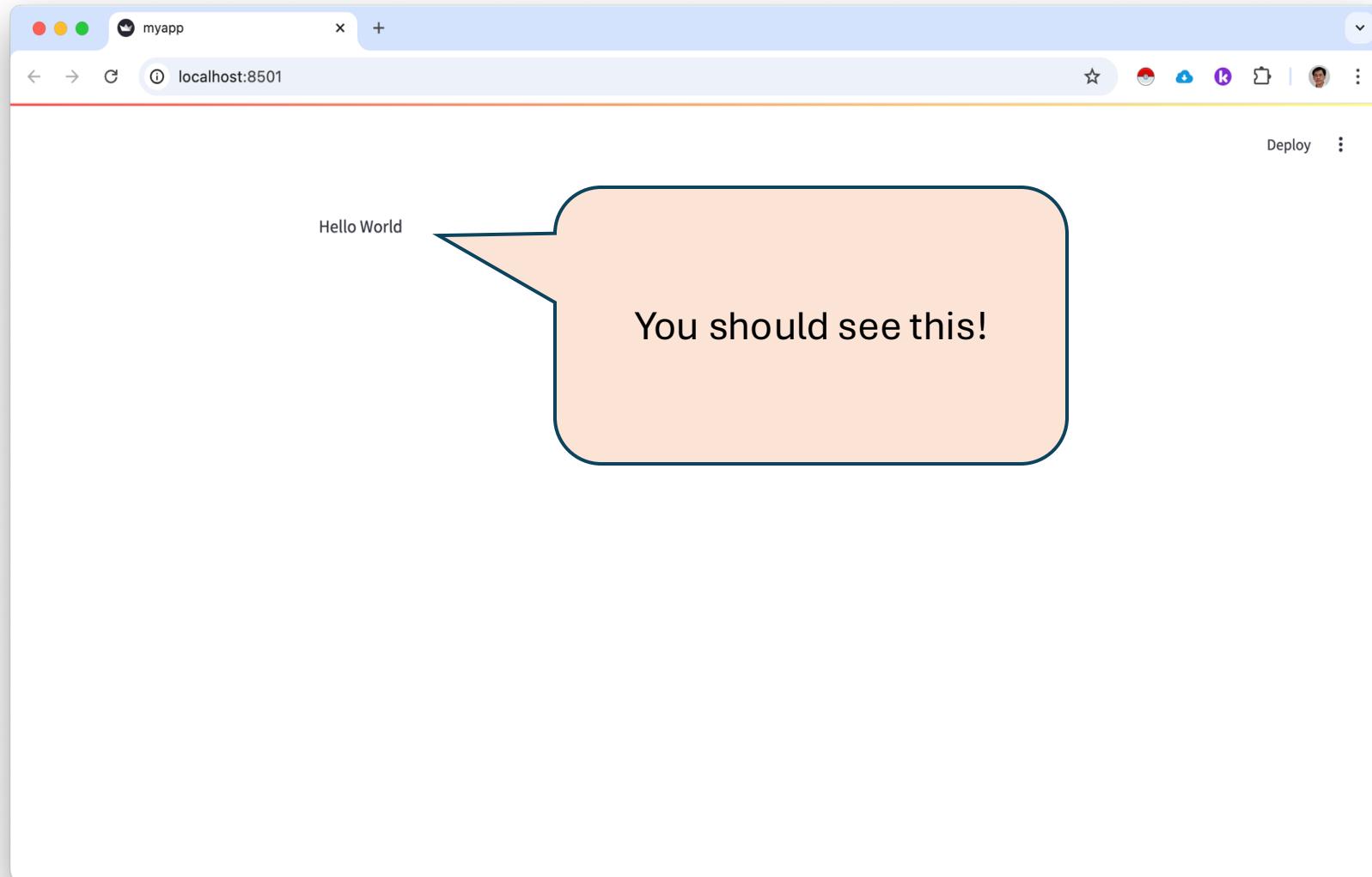
```
(base) cyting@CYs-MacBook-Pro Training % streamlit run myapp.py
```

Following the command, there is a message in blue text: "You can now view your Streamlit app in your browser." Below this, the local and network URLs are displayed:

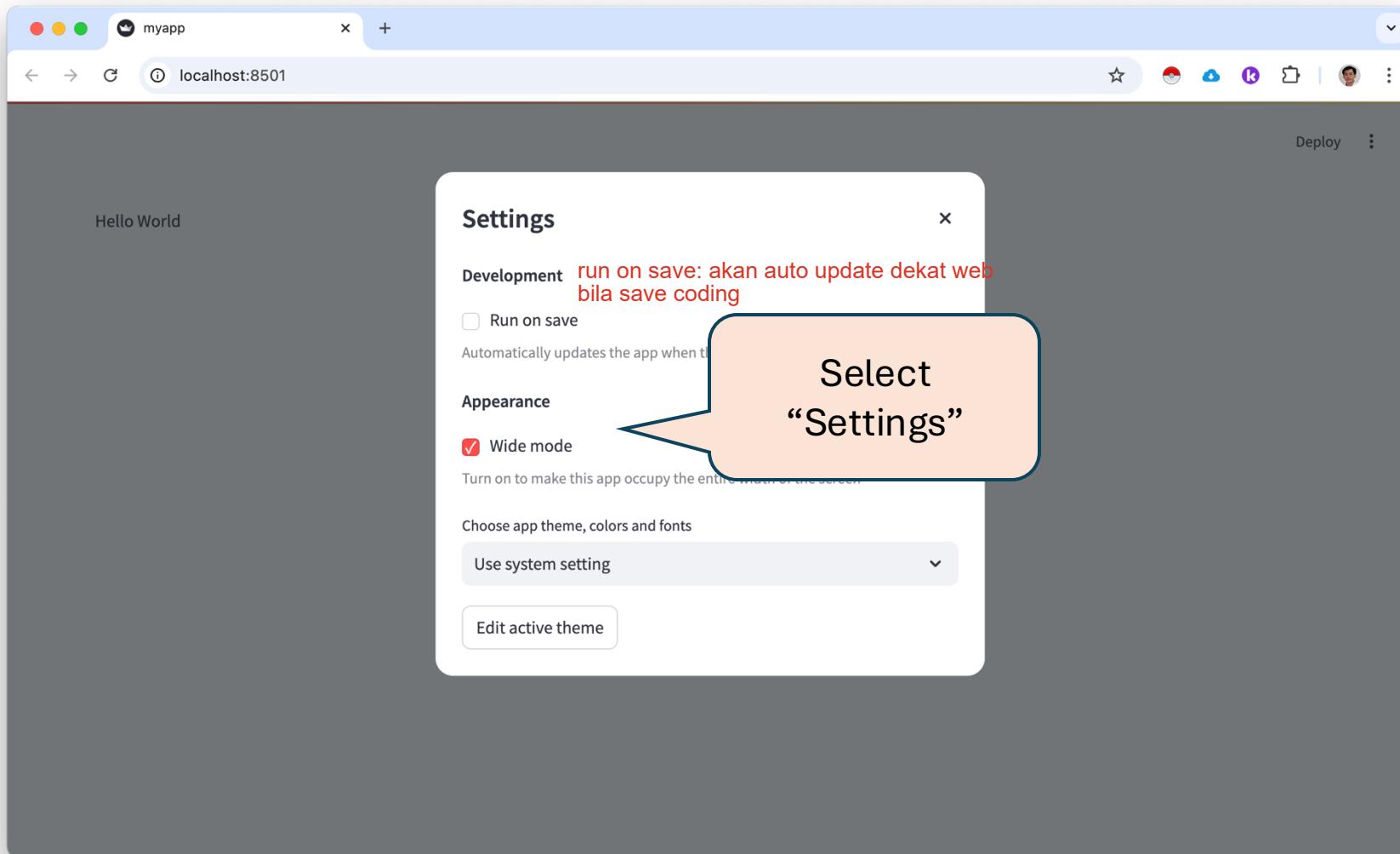
```
Local URL: http://localhost:8501
Network URL: http://192.168.50.178:8501
```

A red rectangular box highlights the terminal command "streamlit run myapp.py". A light orange callout bubble points from this highlighted area to the text "This things show that you are running the app locally at your computer." located in the bottom right corner of the slide.

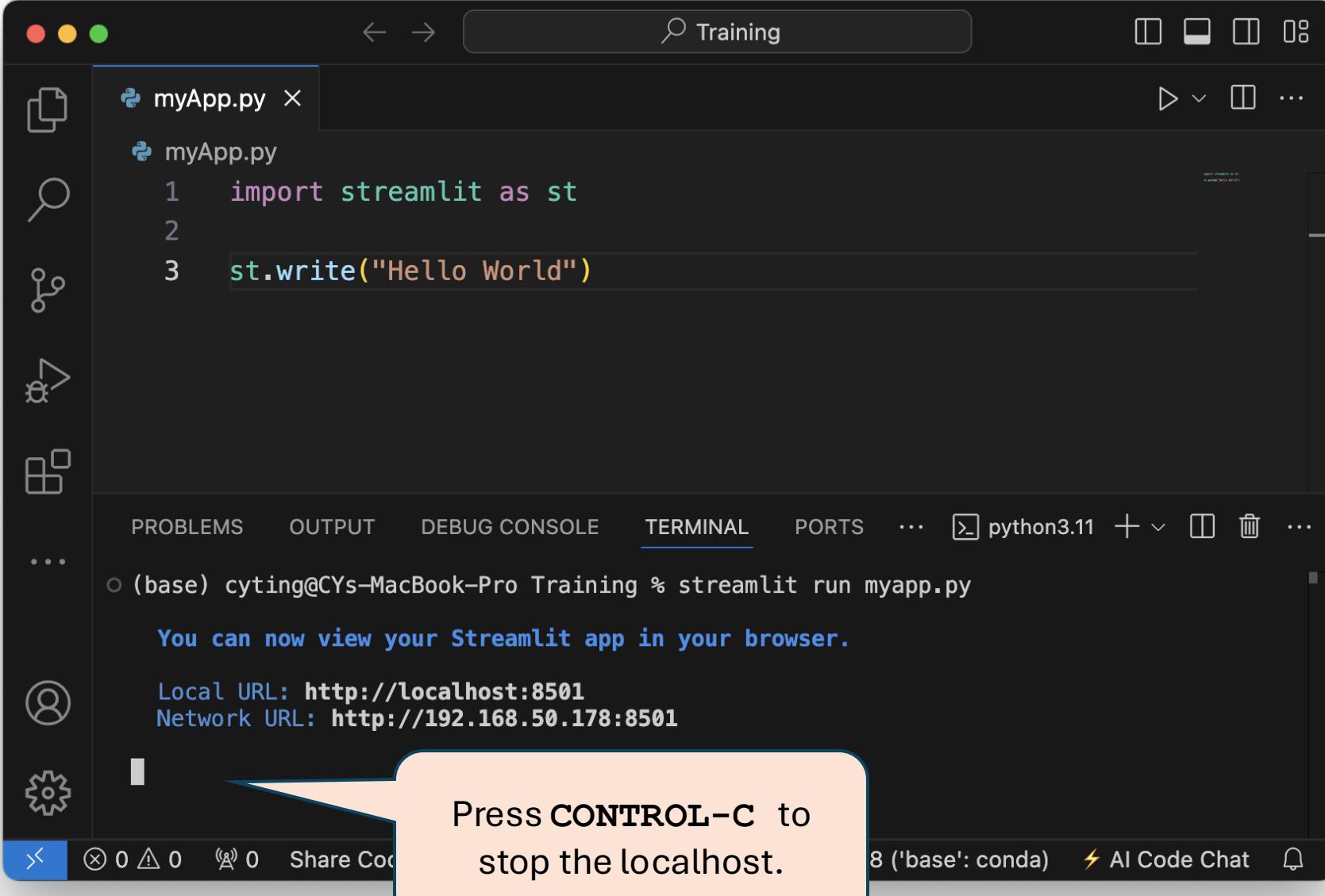
Seeing the Output



Modifying the Output



Stopping the Localhost (CTRL-C)



The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) running on a Mac. The title bar says "Training". The left sidebar has icons for file, search, and other tools. The main editor window contains a Python file named "myApp.py" with the following code:

```
myApp.py
1 import streamlit as st
2
3 st.write("Hello World")
```

Below the editor is a terminal window showing the command:

```
(base) cyting@CYs-MacBook-Pro Training % streamlit run myapp.py
```

Followed by the Streamlit output:

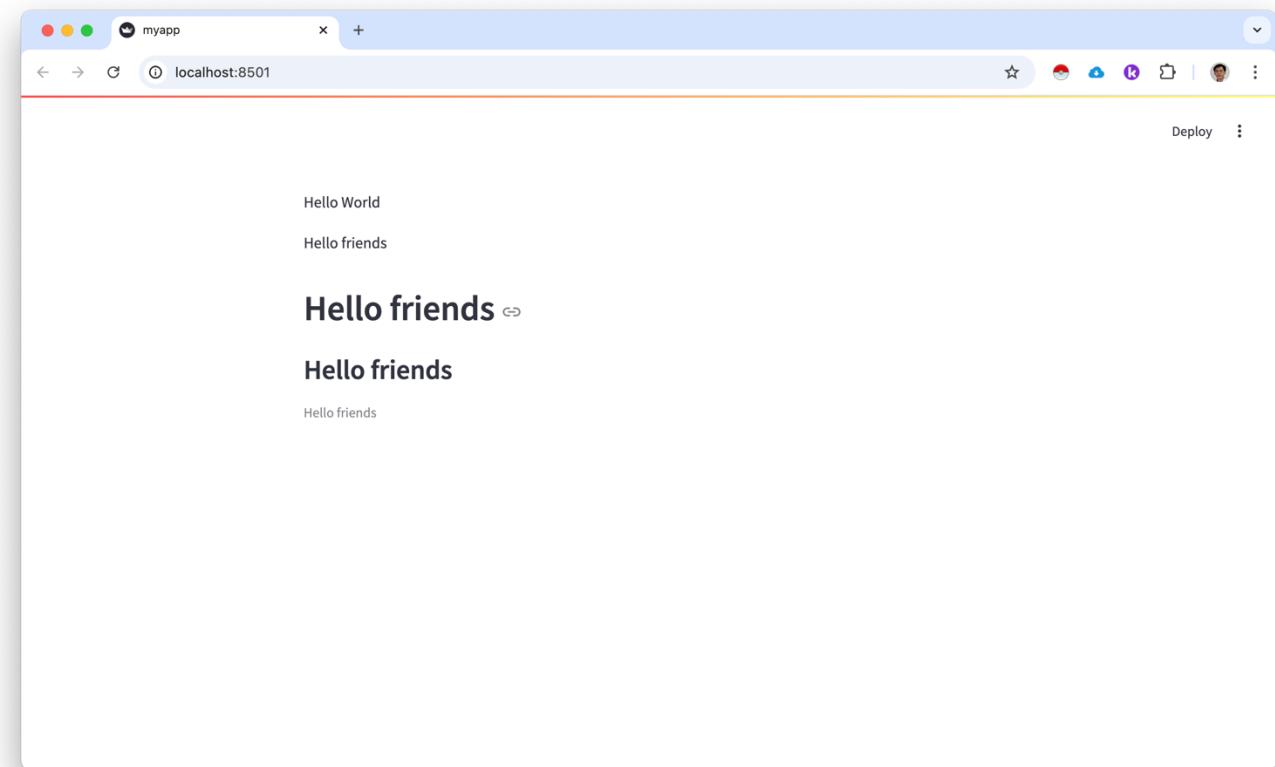
```
You can now view your Streamlit app in your browser.  
Local URL: http://localhost:8501  
Network URL: http://192.168.50.178:8501
```

A callout bubble in the bottom right corner of the terminal area contains the text:

Press **CONTROL-C** to stop the localhost.

Text Formatting

```
st.write("Hello friends")
st.header("Hello friends")
st.subheader("Hello friends")
st.caption("Hello friends")
```



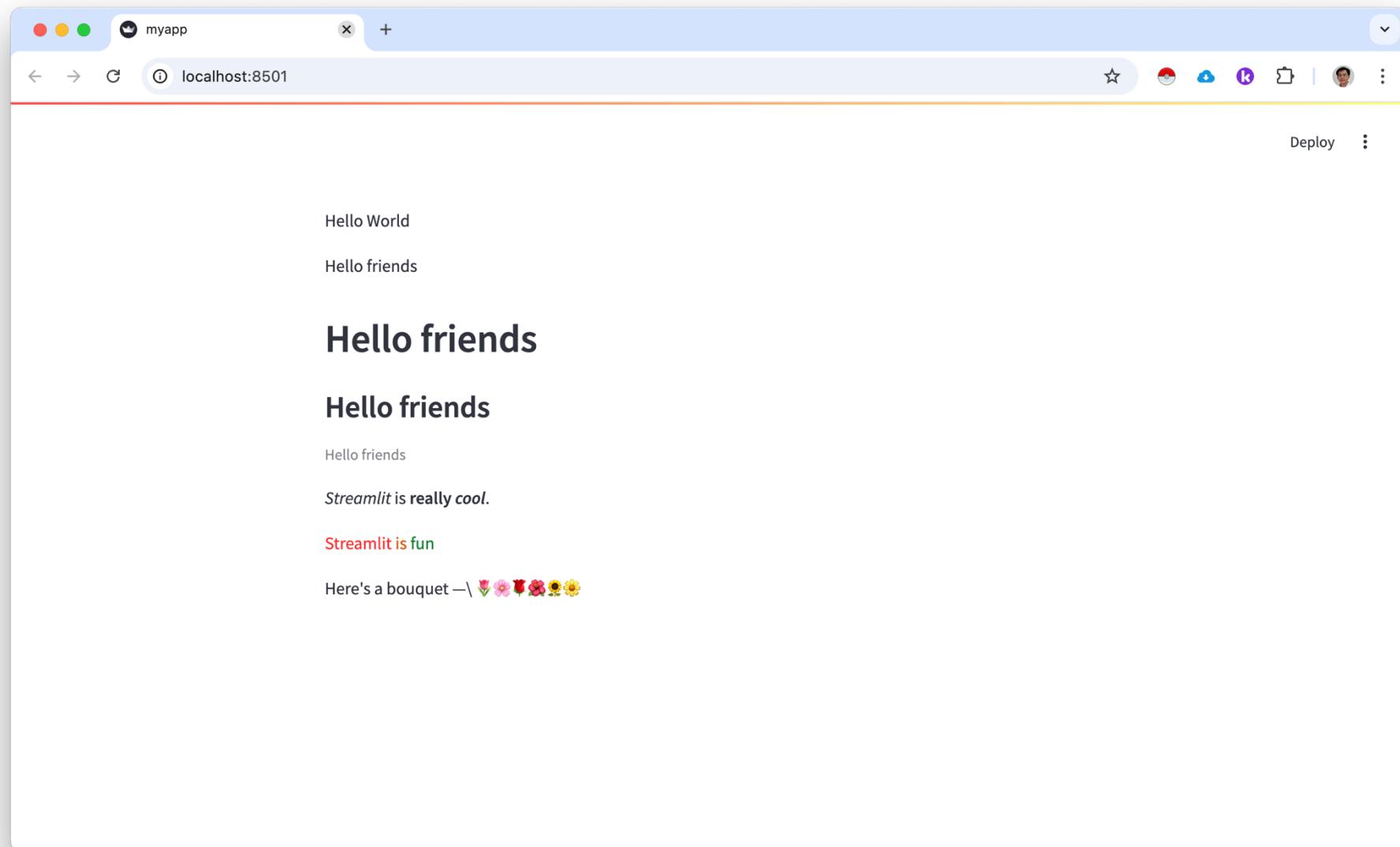
Text Formatting

```
st.markdown("'*Streamlit* is **really** ***cool***.'")
```

```
st.markdown(''':red[Streamlit] :orange[is]  
:green[fun]'''')
```

```
st.markdown("Here's a bouquet —\n:tulip::cherry_blossom::rose::hibiscus::sunflower::b  
lossom:")
```

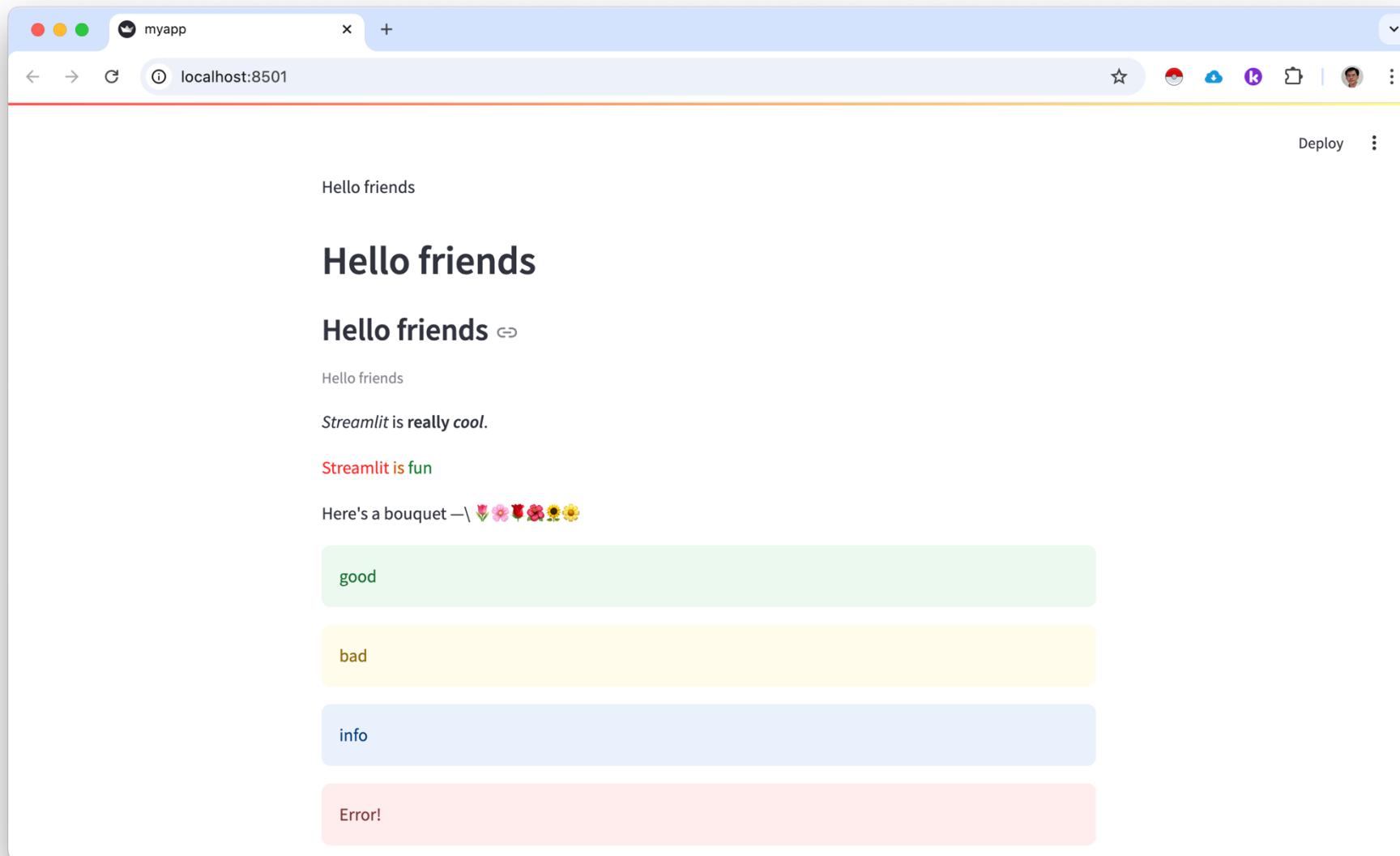
Text Formatting



Text Formatting

```
st.success("good")
st.warning("bad")
st.info("info")
st.error("Error!")
```

Text Formatting

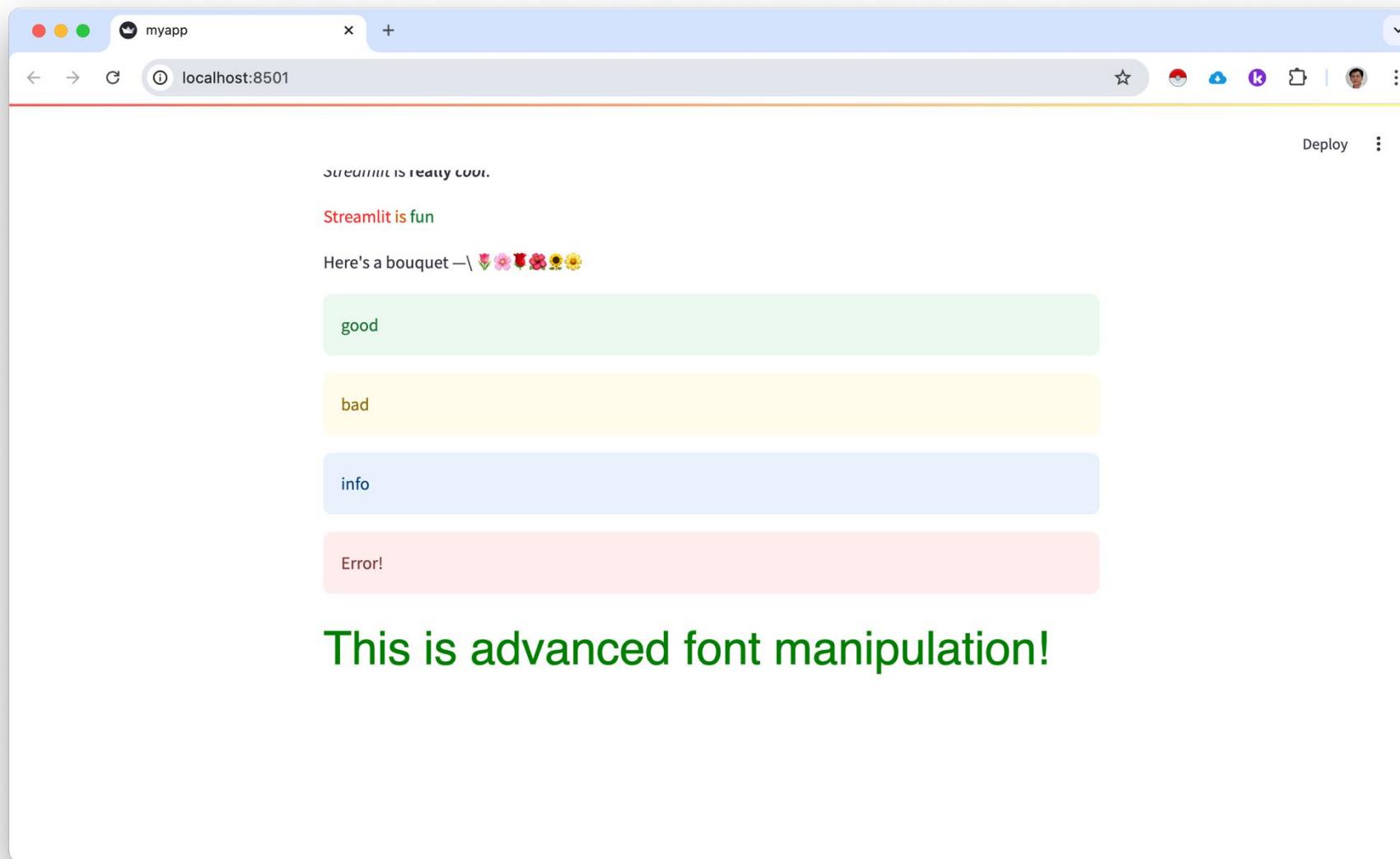


Advanced Font

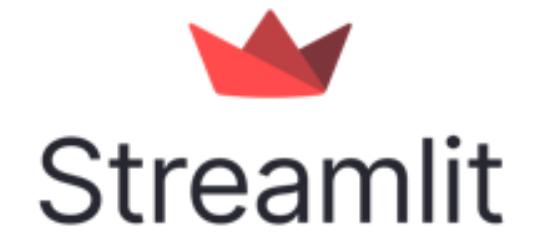
```
new_title = '<p style="font-family:sans-serif;  
color:Green; font-size: 42px;">This is advanced font  
manipulation!</p>'
```

```
st.markdown(new_title, unsafe_allow_html=True)
```

Advanced Font



UI Components



Selection Box

- st.selectbox("Kuala Lumpur is located at",
['Malaysia', 'Thailand', 'UK'])
- st.multiselect('Select 2 states',
['Selangor', 'Johor', 'Kedah'])

Kuala Lumpur is located at

Malaysia ▾

Select 2 states

Choose an option ▾

Buttons

```
st.button("Click Here to Proceed")
```

```
st.slider("Select the length of stay", 1,10, value=3)
```

Click Here to Proceed

Select the length of stay

3

10

Input Box

```
number = st.number_input("Insert a number",  
value=None, placeholder="Type a number...")  
  
st.write("The current number is ", number)
```

Insert Graphics

- from PIL import Image
- im = Image.open('shrdc_logo.png')
- st.image(im, width=300)

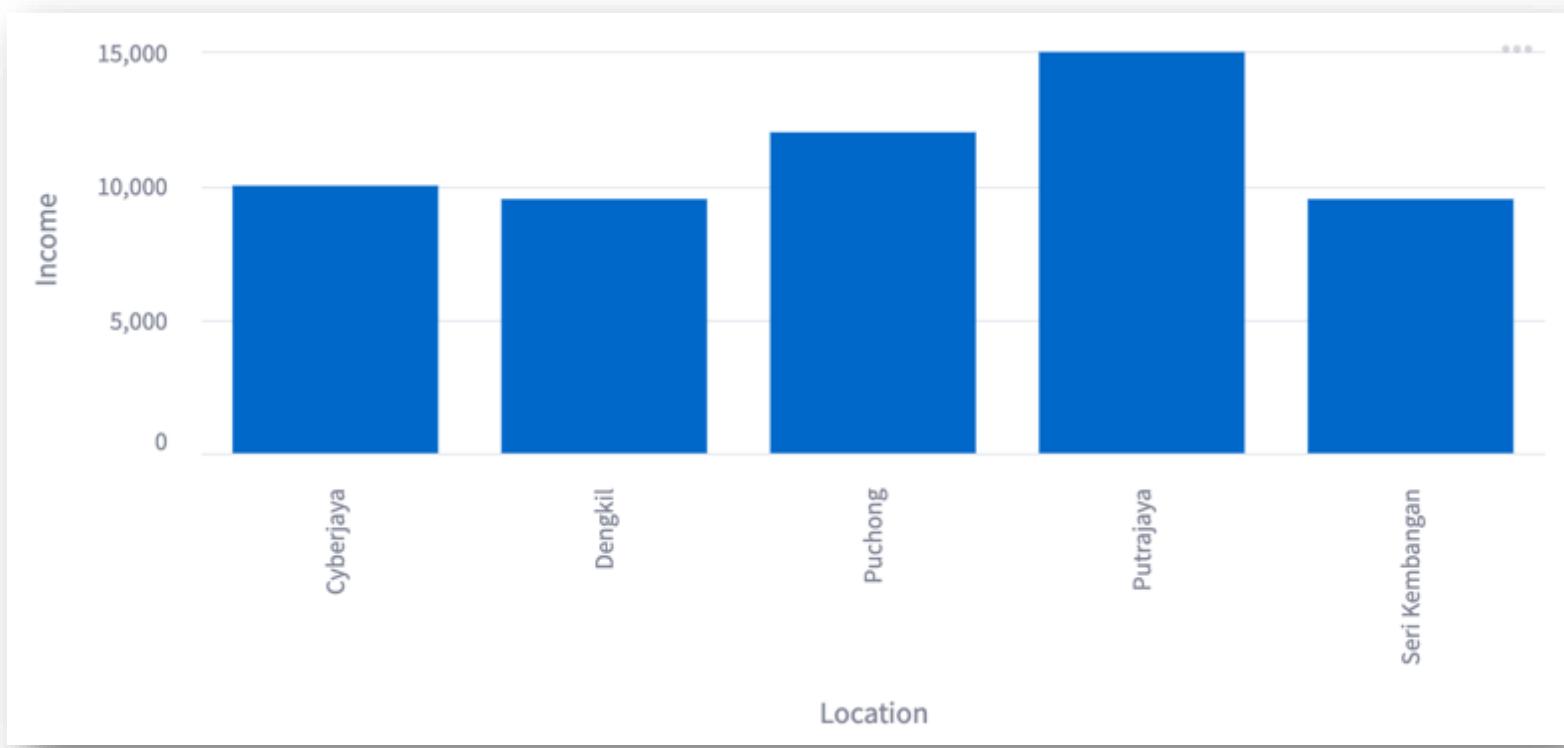
Dataframe

```
import pandas as pd  
df = pd.read_excel('sampledata.xlsx')  
st.dataframe(df)
```

	Location	House Type	Month Fees	Household	Income
0	Cyberjaya	Terrace	500	3	10,000
1	Putrajaya	Condominium	500	4	15,000
2	Puchong	Bungalow	600	2	12,000
3	Seri Kembangan	Semi-D	550	5	9,500
4	Dengkil	Semi-D	450	6	9,000

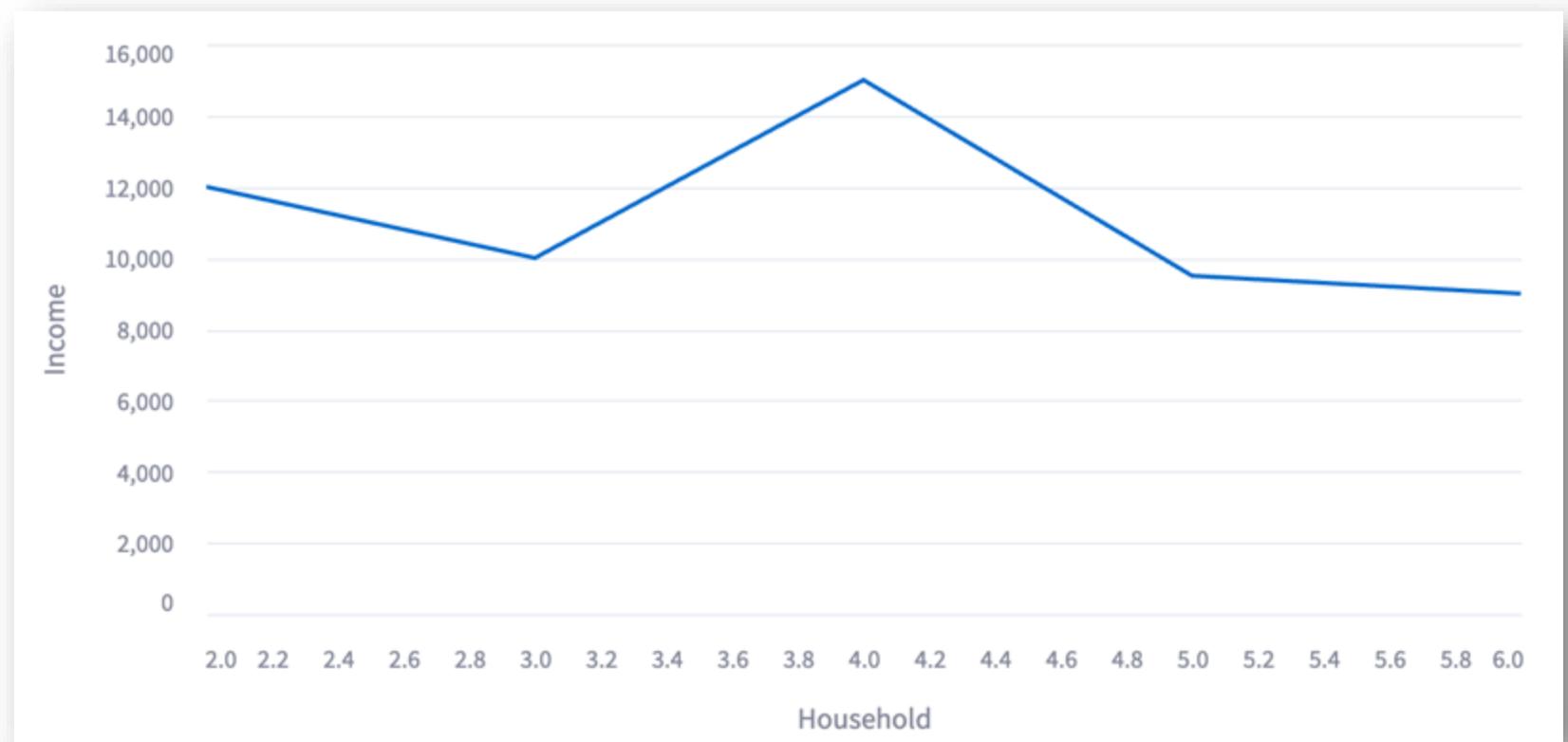
Bar Chart

```
st.bar_chart(df, x="Location", y="Income")
```



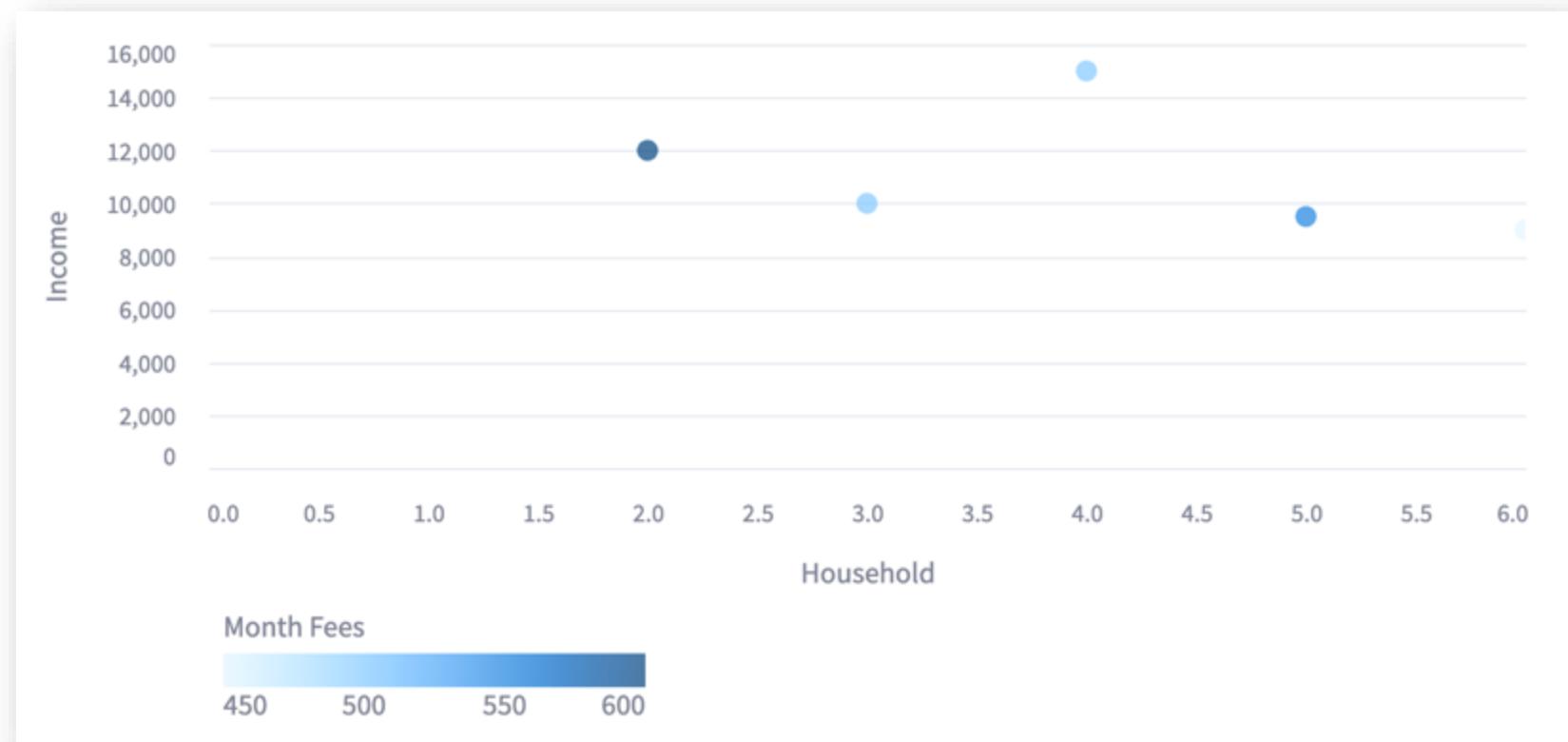
Line Chart

```
st.line_chart(df, x="Household", y="Income")
```



Scatter Chart

```
st.scatter_chart(df, x="Household", y="Income")
```



Creating multi- tabpage

```
tab1, tab2, tab3 = st.tabs(["Cat", "Dog", "Owl"])

with tab1:
    st.header("A cat")
    st.image("https://static.streamlit.io/examples/cat.jpg", width=200)

with tab2:
    st.header("A dog")
    st.image("https://static.streamlit.io/examples/dog.jpg", width=200)

with tab3:
    st.header("An owl")
    st.image("https://static.streamlit.io/examples/owl.jpg", width=200)
```

Creating multi- tabpage

Cat Dog Owl

A cat



Create Multi-columns

```
col1, col2, col3 = st.columns(3)

with col1:
    st.header("A cat")
    st.image("https://static.streamlit.io/examples/cat.jpg")

with col2:
    st.header("A dog")
    st.image("https://static.streamlit.io/examples/dog.jpg")

with col3:
    st.header("An owl")
    st.image("https://static.streamlit.io/examples/owl.jpg")
```

Create Multi-columns

A cat



A dog



An owl



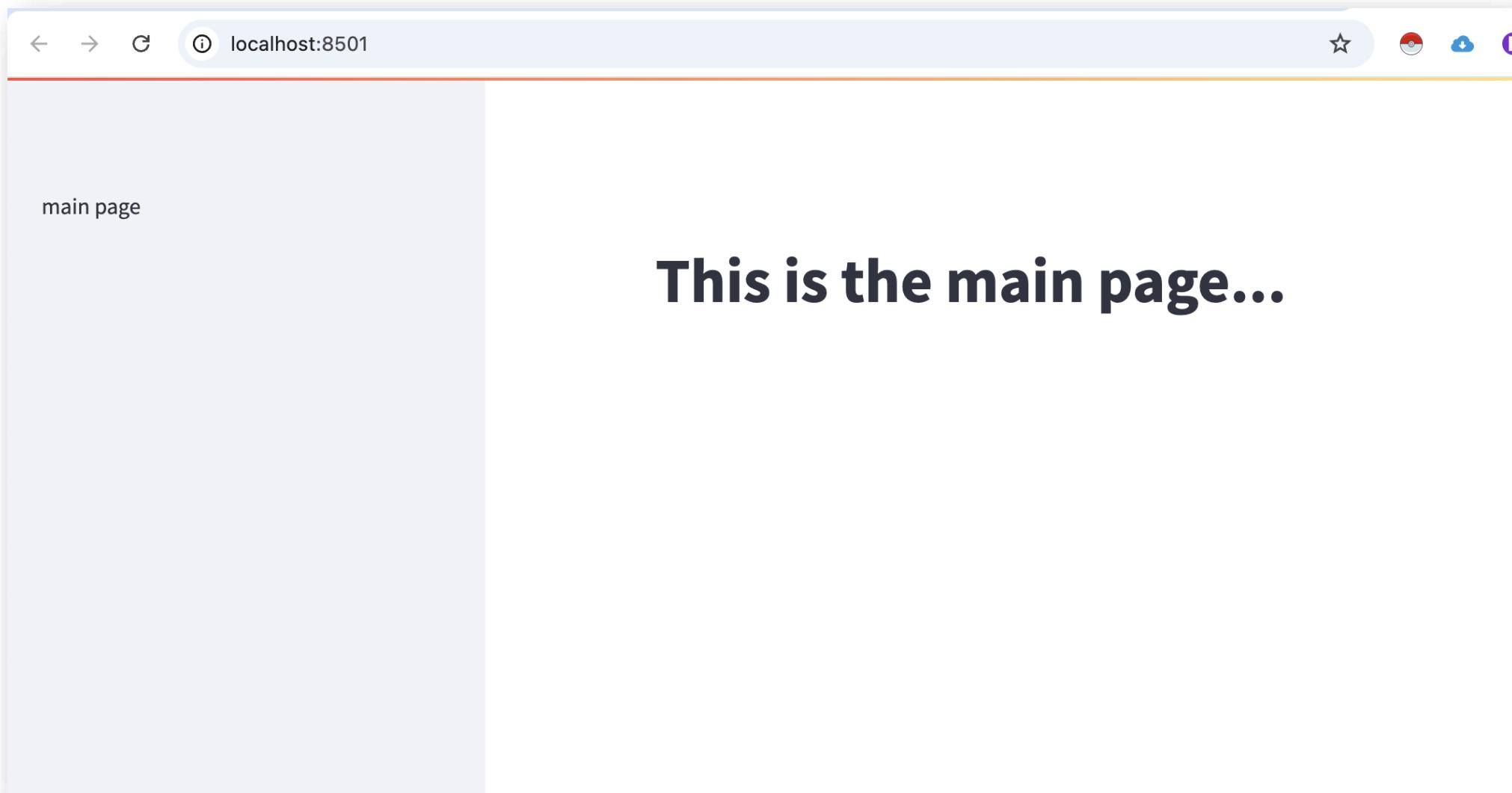
Create Multipages

- Create a python file named “main.py”.

Type your code here!

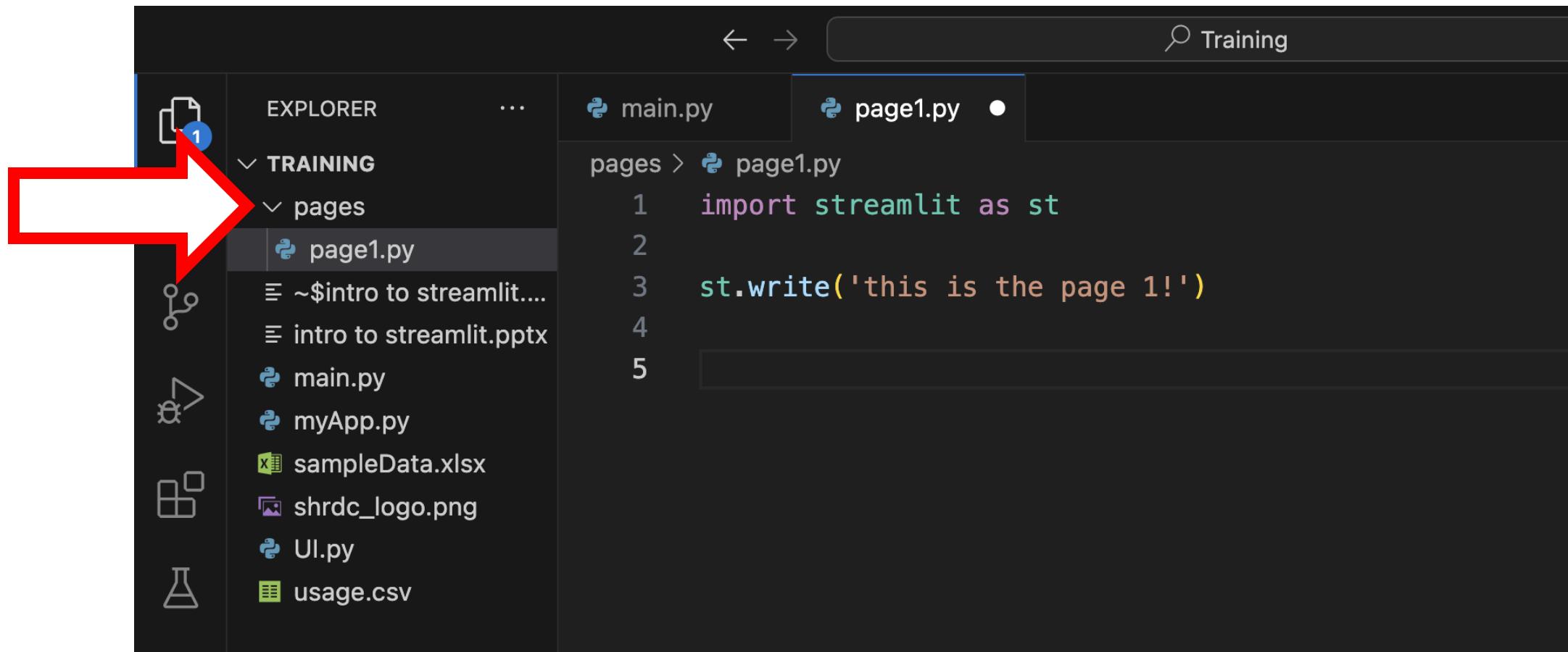
```
1 import streamlit as st
2
3 st.set_page_config(
4     page_title="test"
5 )
6
7 st.markdown('# This is the main page...')
8 st.sidebar.write('main page')
```

Create Multipages

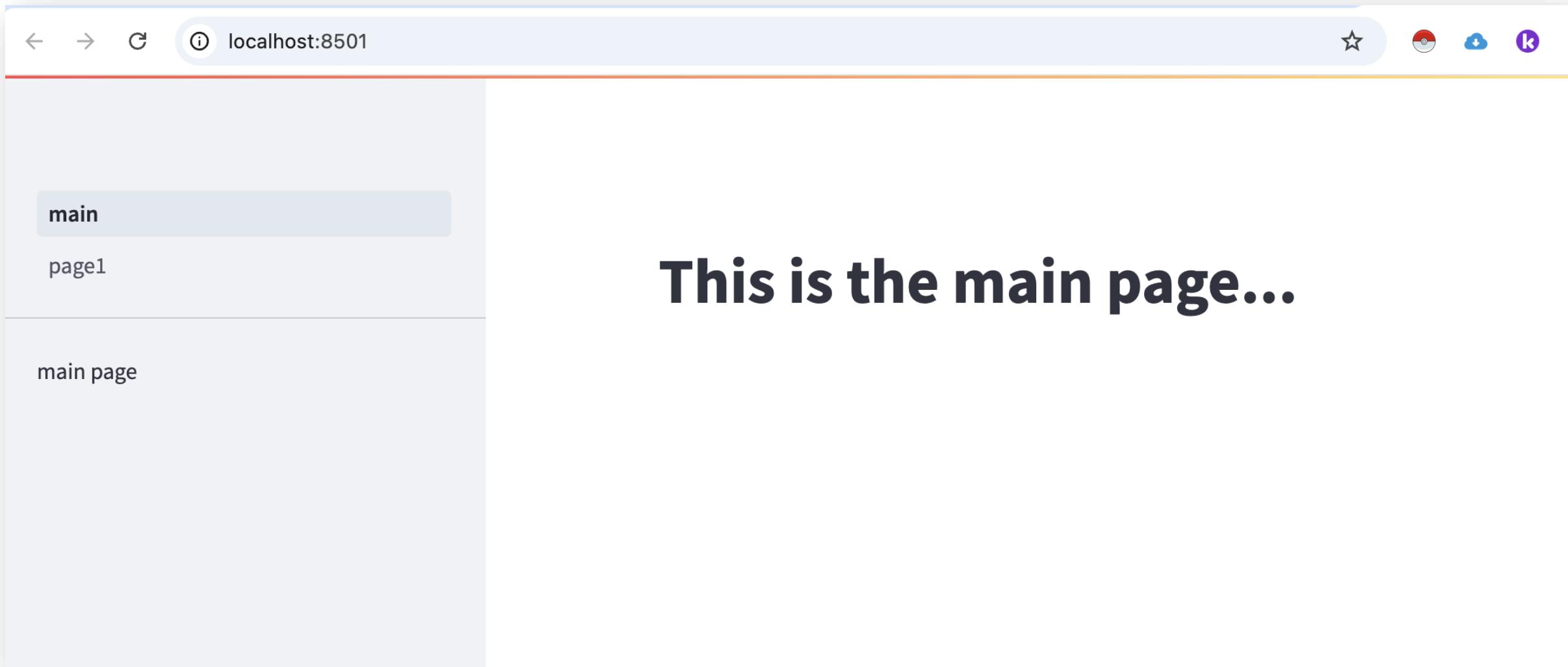


Create Multipages

- Create a folder named “**pages**”. You MUST use the name “**pages**”, nothing else!

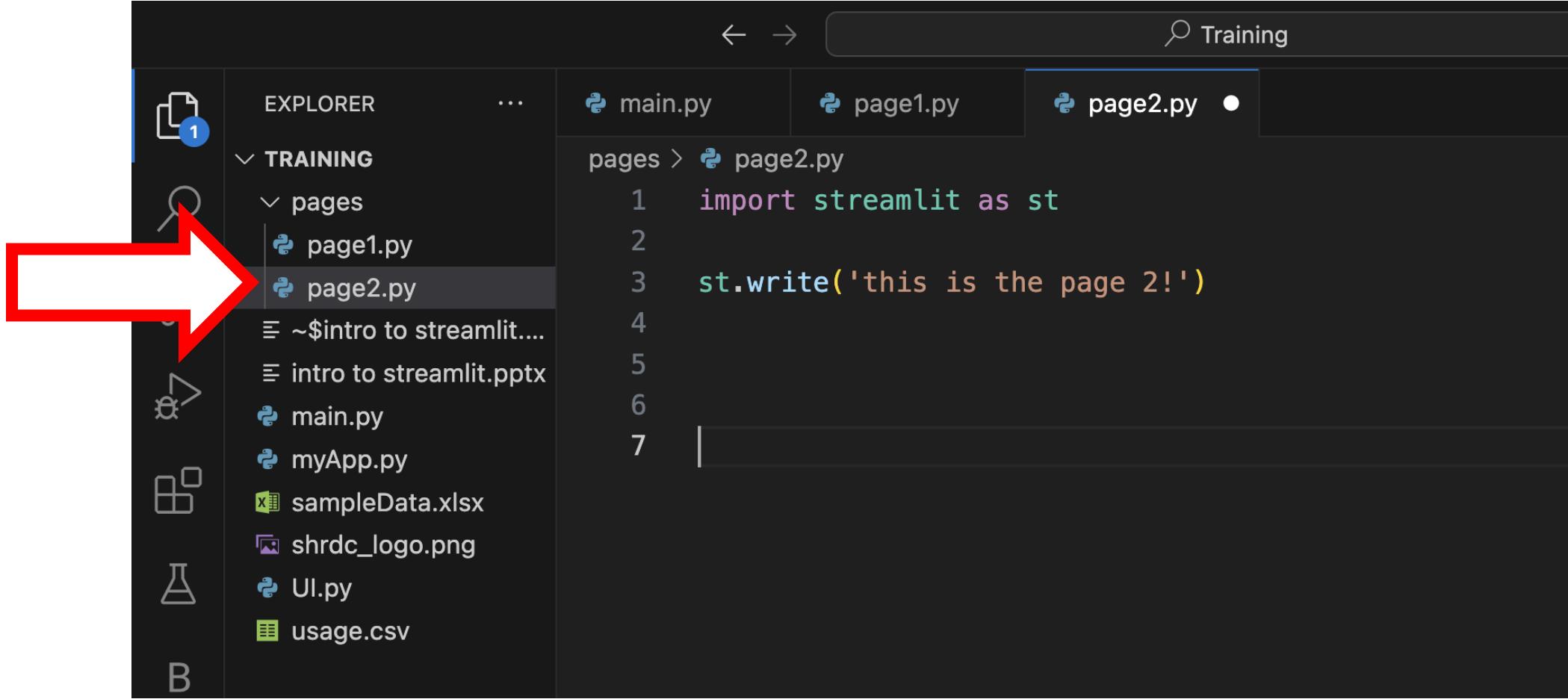


Create Multipages



Create Multipages

- Create a new python file named page2.py inside the “pages” folder.

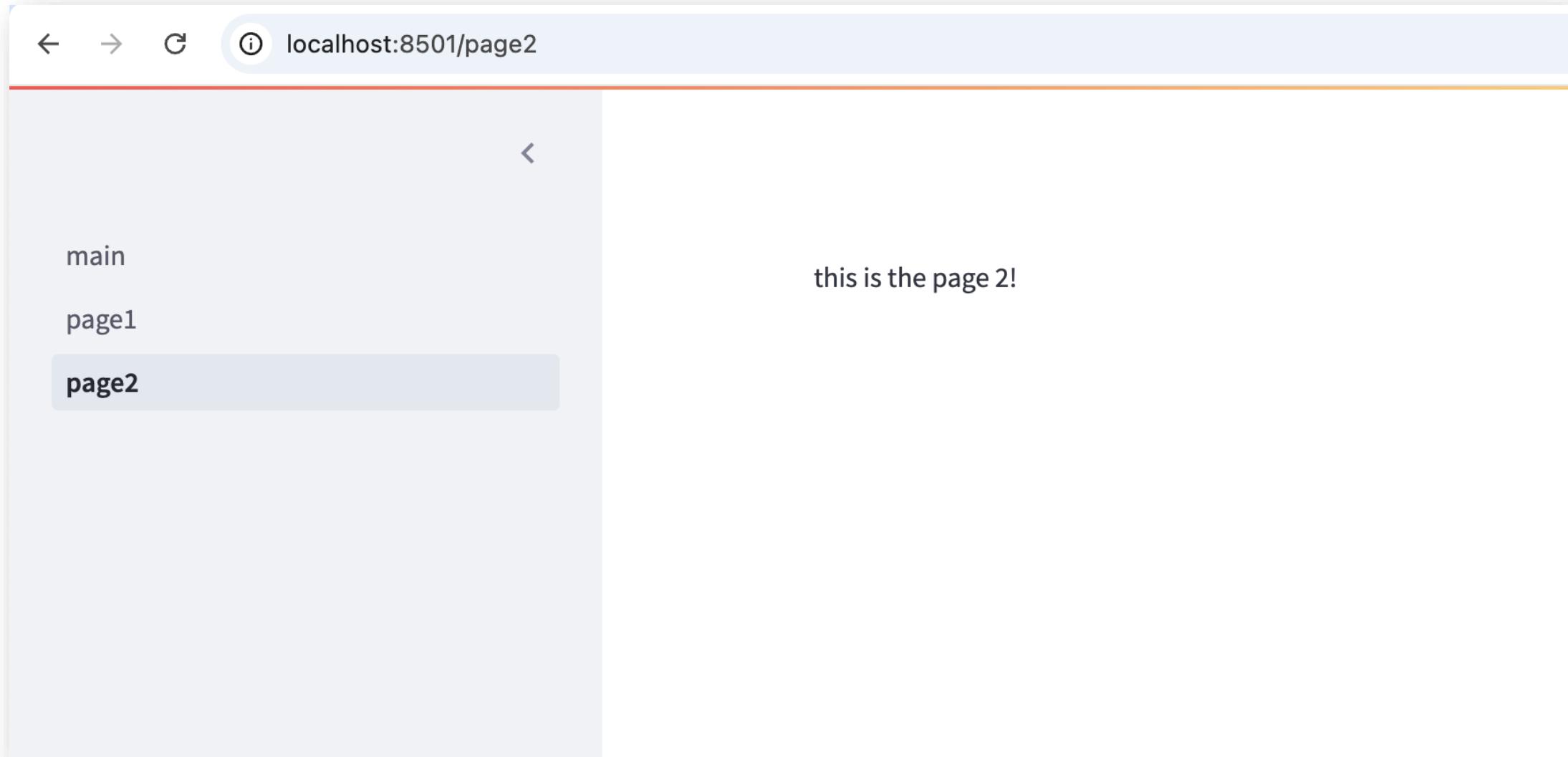


The screenshot shows a code editor interface with the following details:

- EXPLORER View:** On the left, it shows a file tree under the "TRAINING" root. The "pages" folder contains "page1.py" and "page2.py". Other files listed include "main.py", "myApp.py", "sampleData.xlsx", "shrdc_logo.png", "UI.py", and "usage.csv".
- Search Bar:** At the top right, there is a search bar with the placeholder text "Training".
- File Tabs:** Below the search bar, there are tabs for "main.py", "page1.py", and "page2.py". The "page2.py" tab is currently active, indicated by a blue border and a white dot on the right.
- Code Editor:** The main area displays the content of "page2.py". The code is:

```
1 import streamlit as st
2
3 st.write('this is the page 2!')
```

Create Multipages



Deploying Streamlit



Important Steps

Step 1 -> Register GitHub account

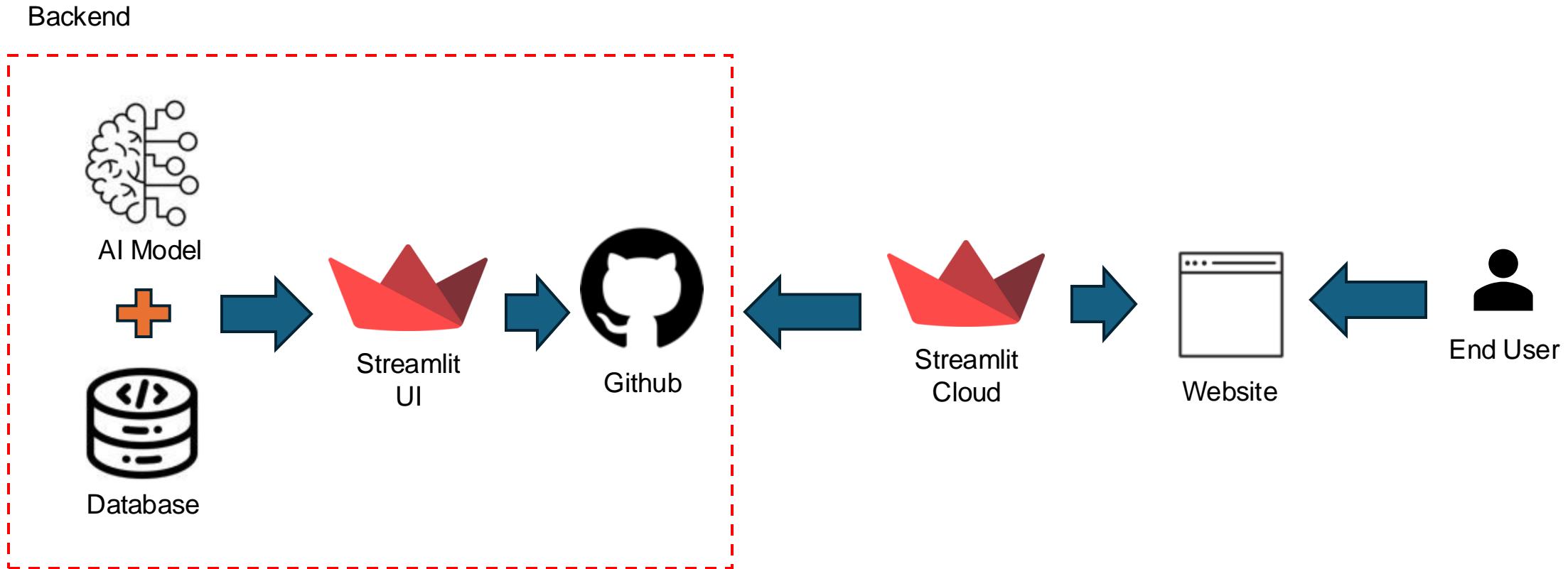
Step 2 -> Set up GitHub repository

Step 3 -> Upload relevant to the repository

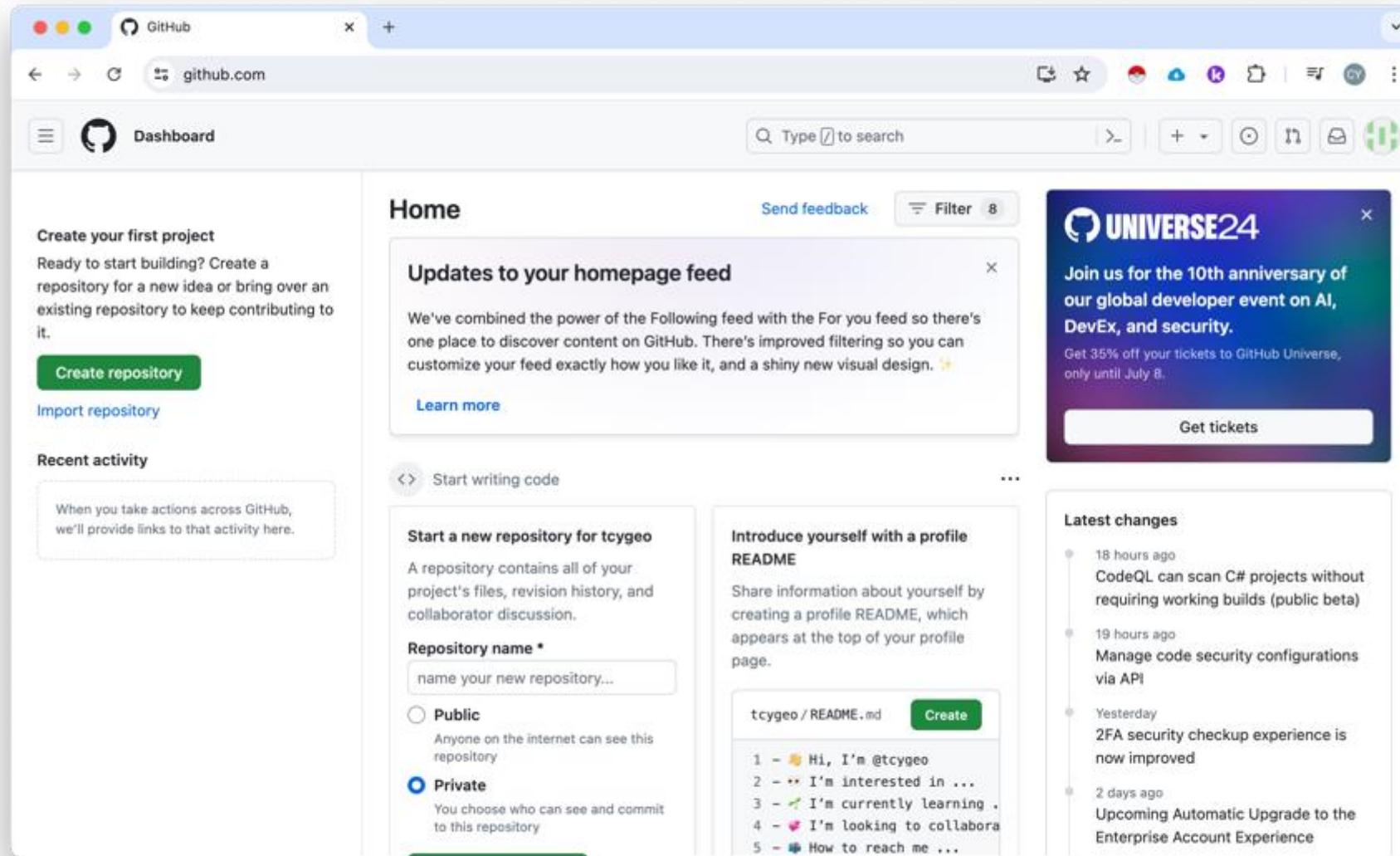
Step 4 -> Sign Up **Streamlit** Cloud account

Step 5 -> Link GitHub repository to **Streamlit** Cloud

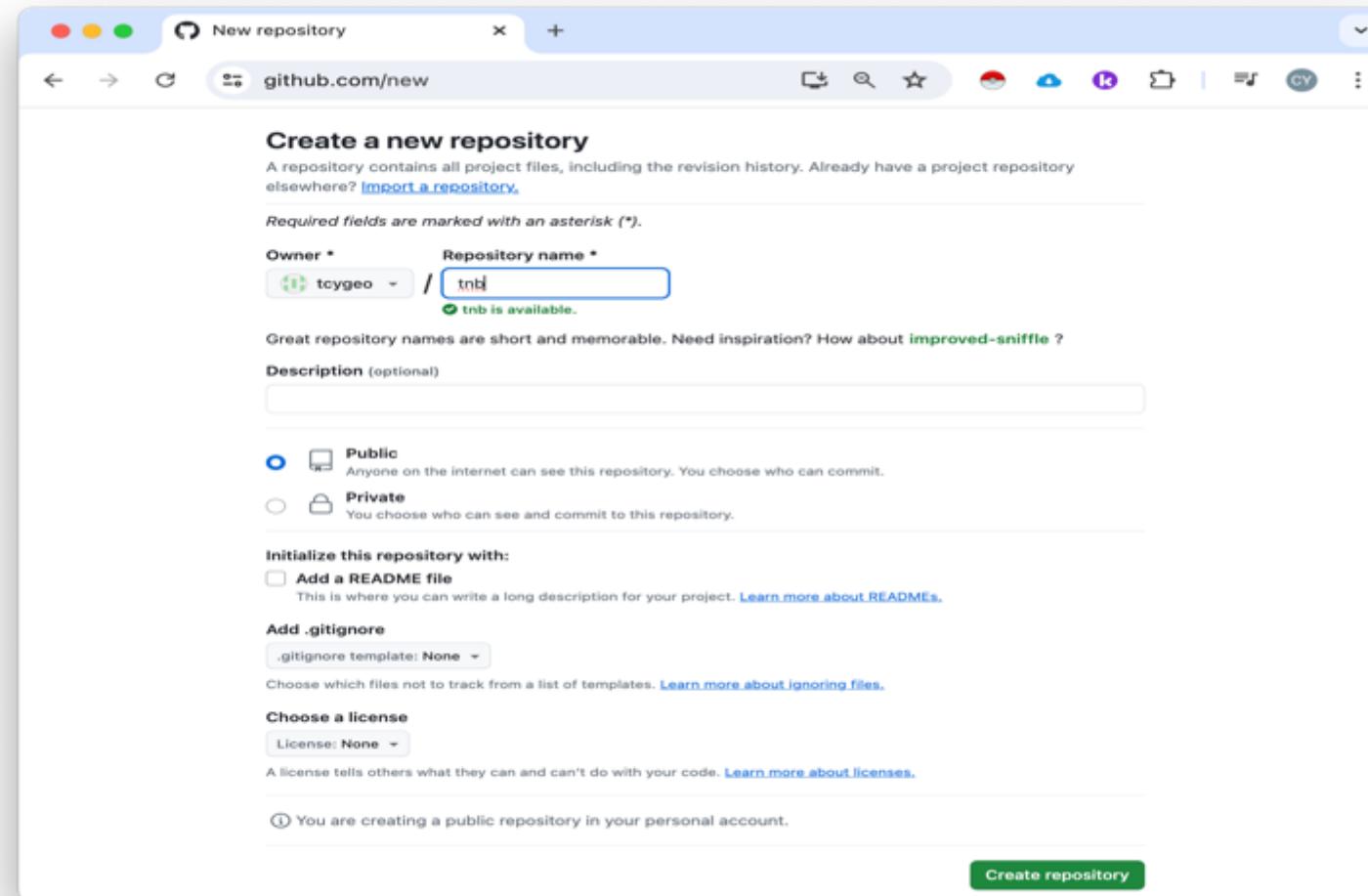
Overall Flow for Deployment



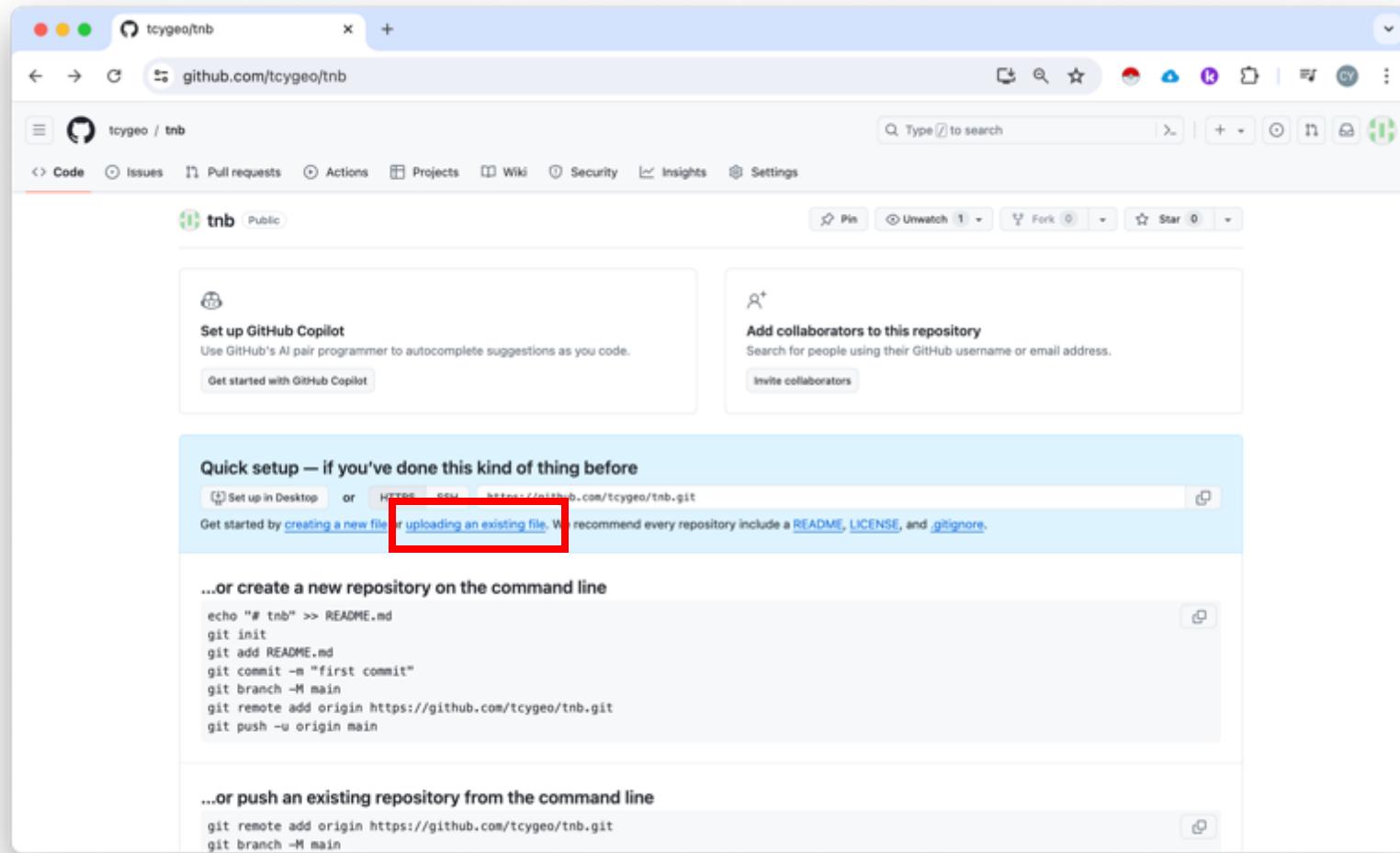
Create Account at GitHub



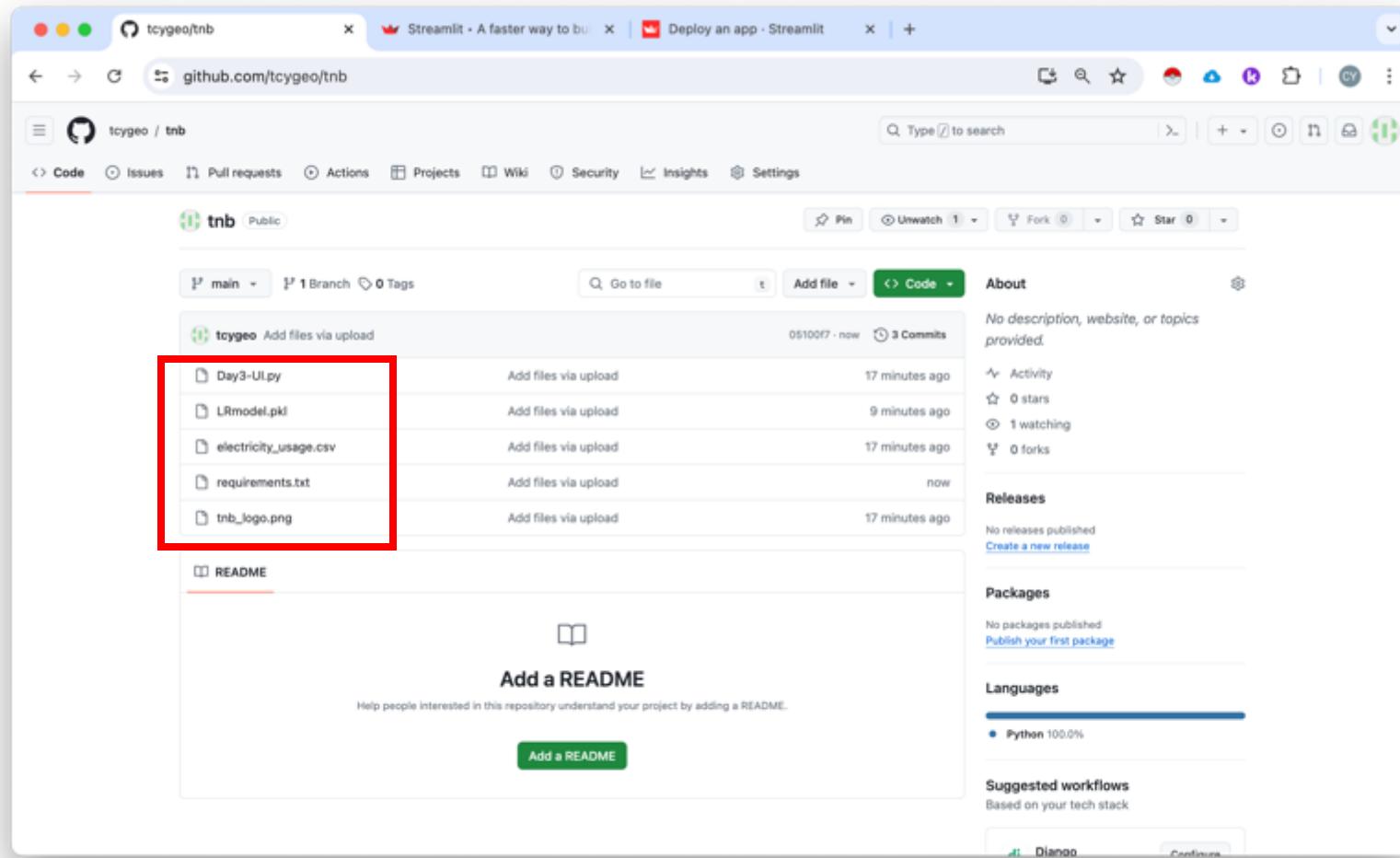
Create a Repository



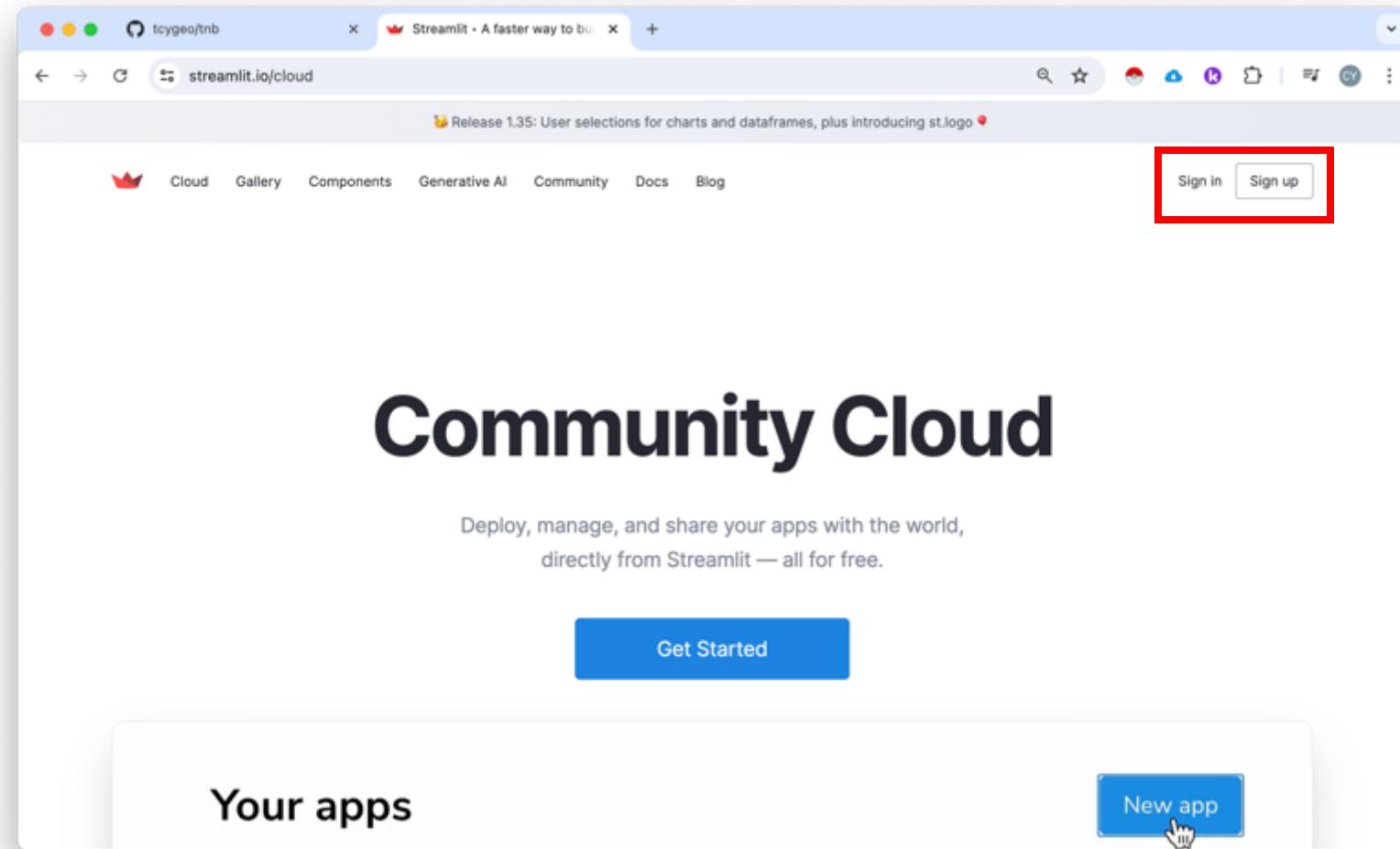
Upload Relevant Files



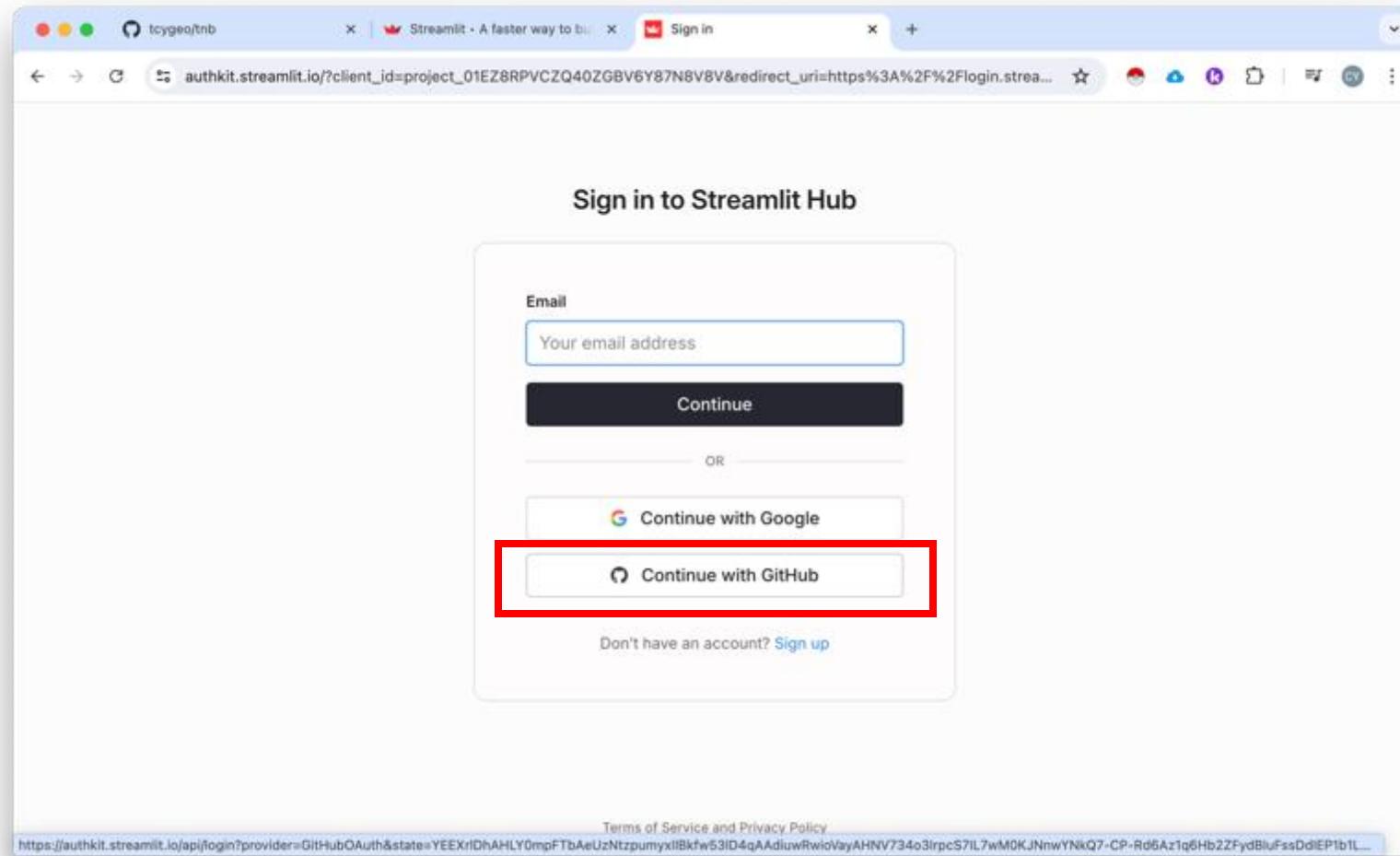
Upload Relevant Files



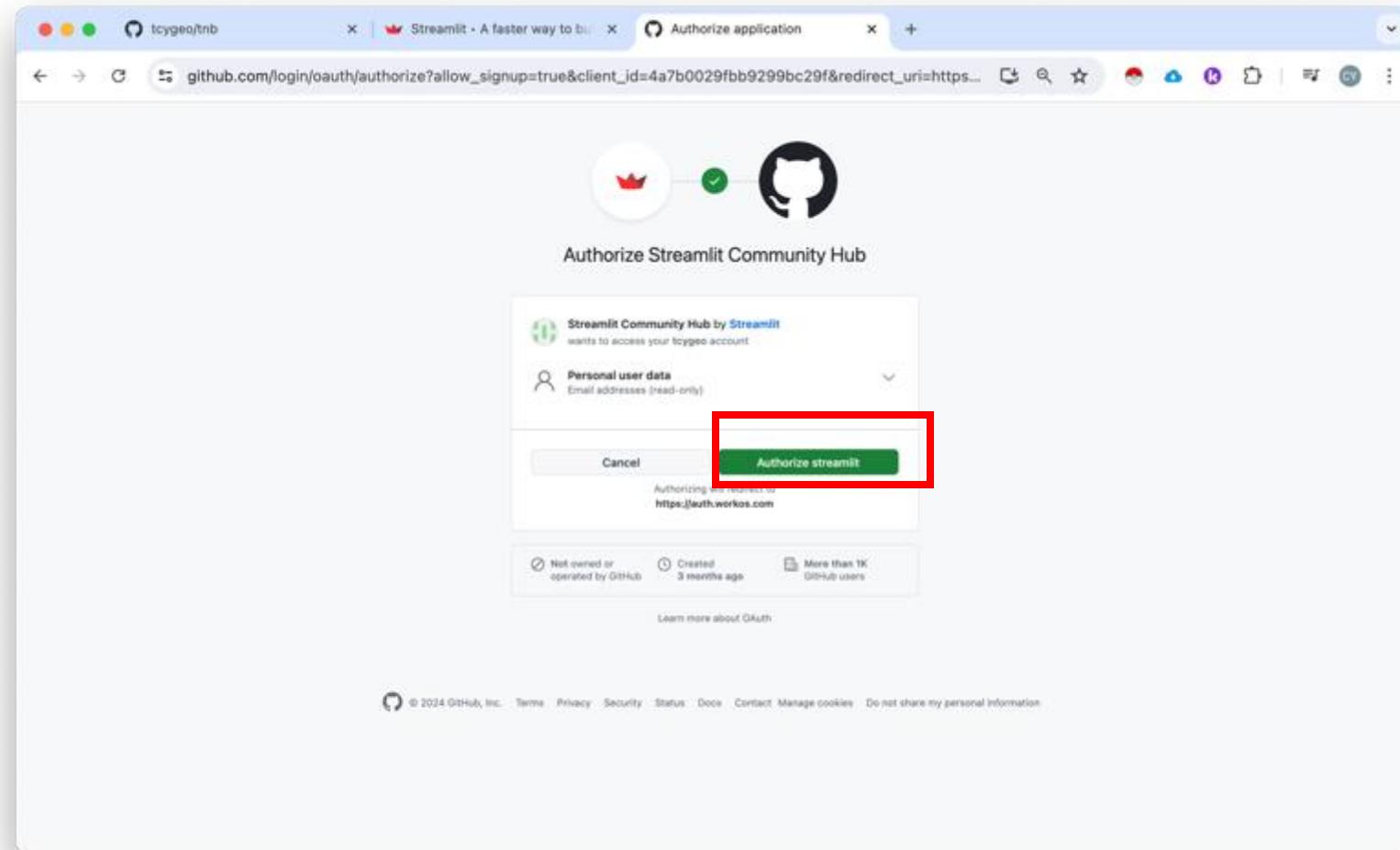
Sign Up Community Cloud



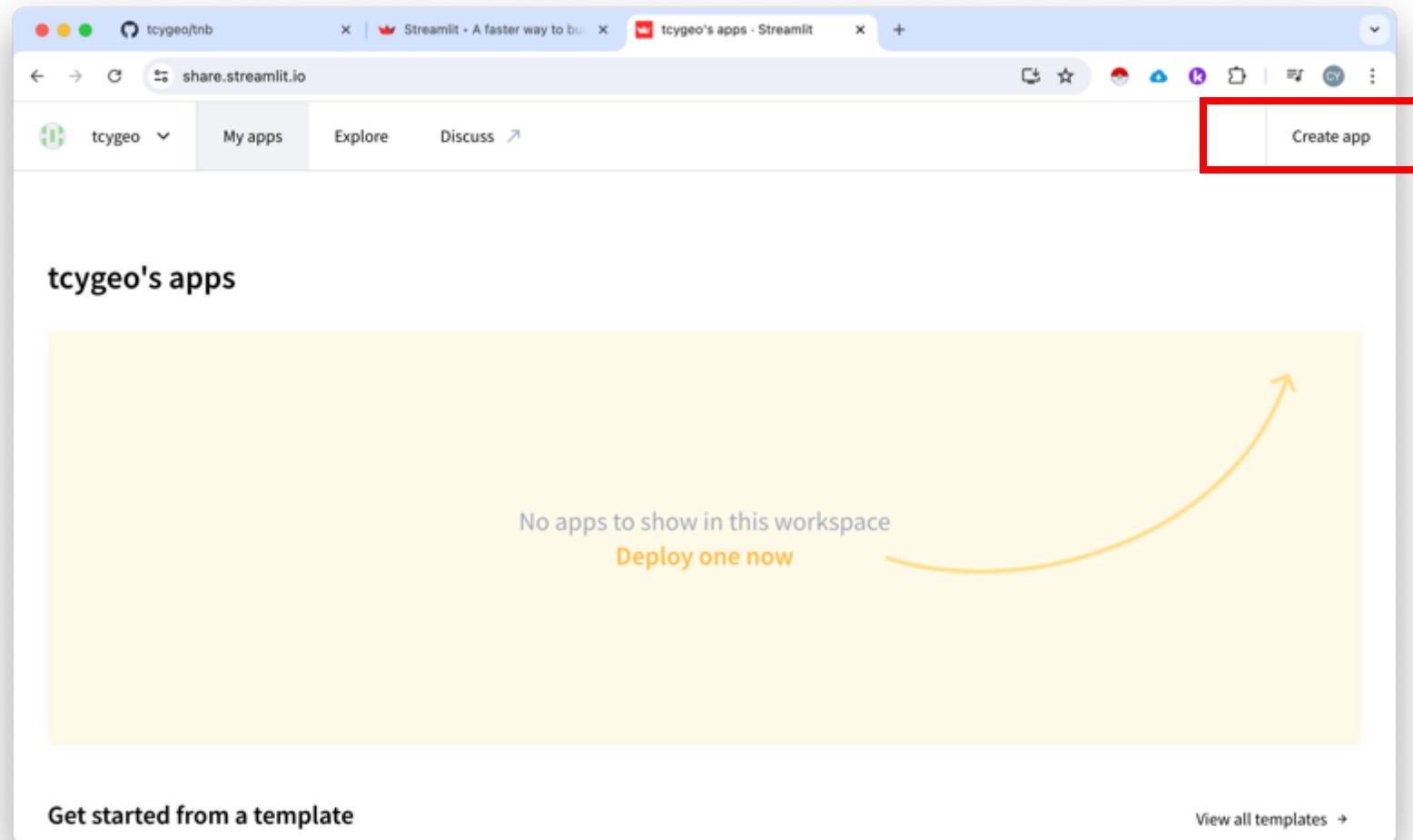
Sign Up using GitHub Account



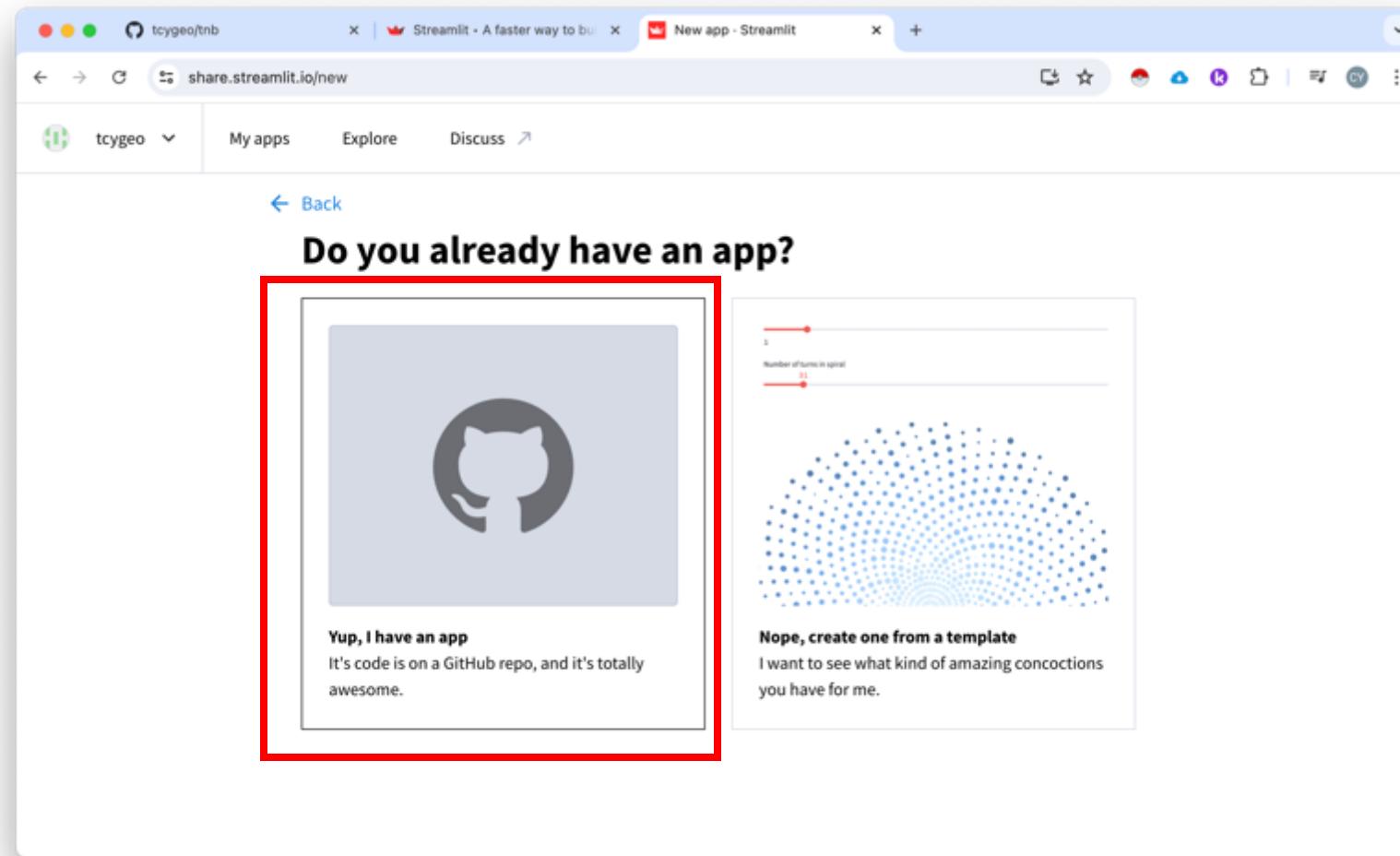
Sign Up using GitHub Account



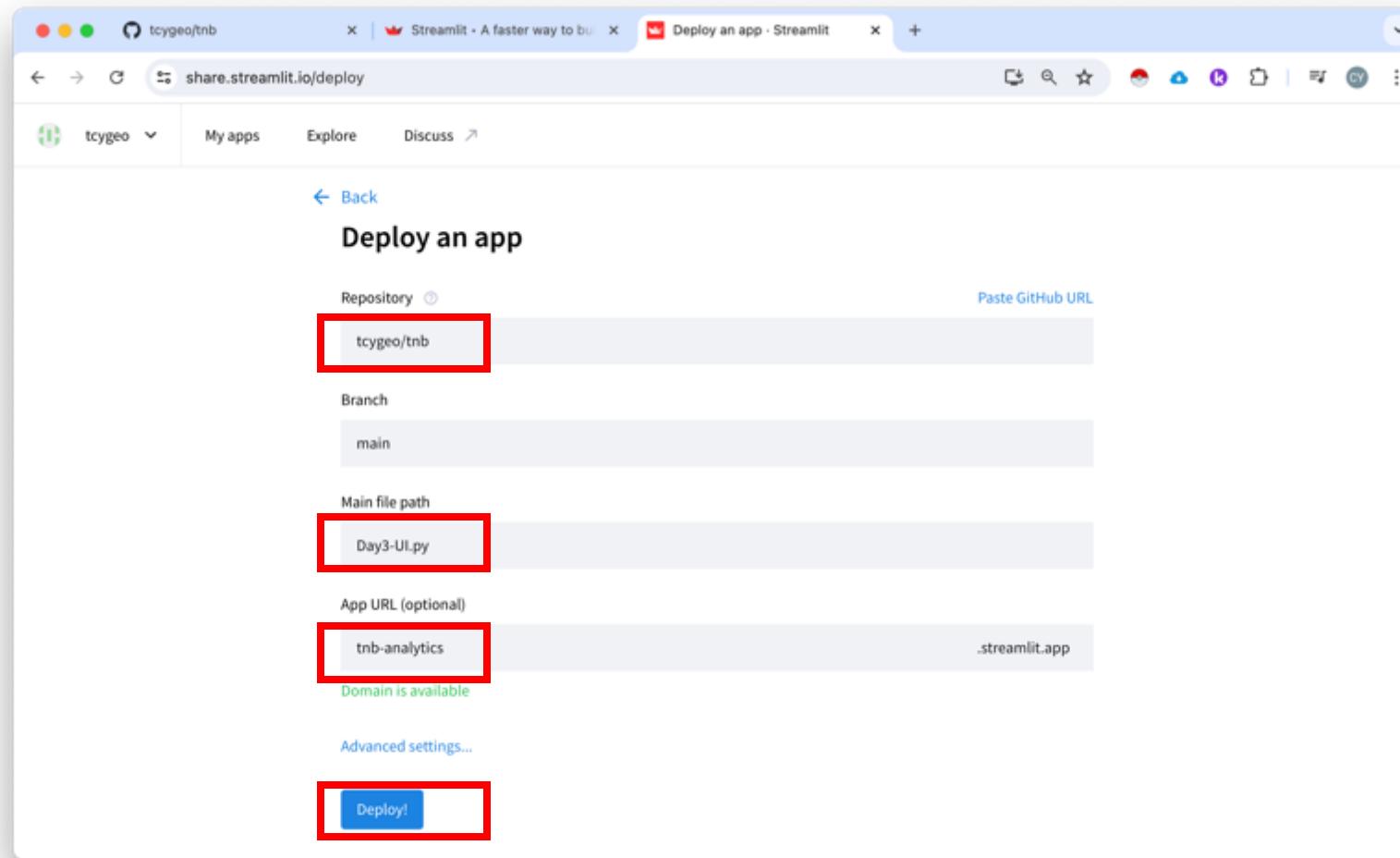
Create App



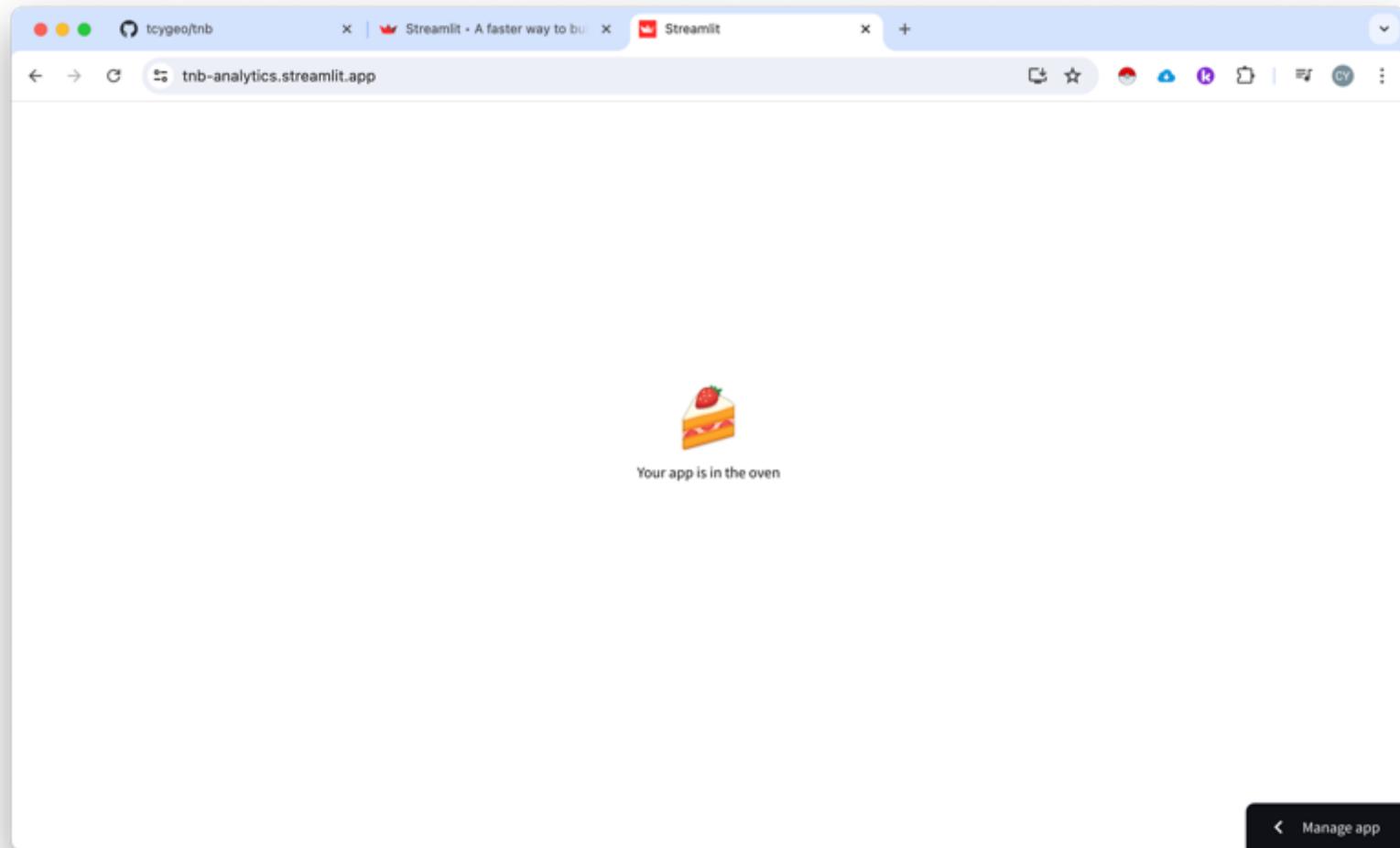
Create App



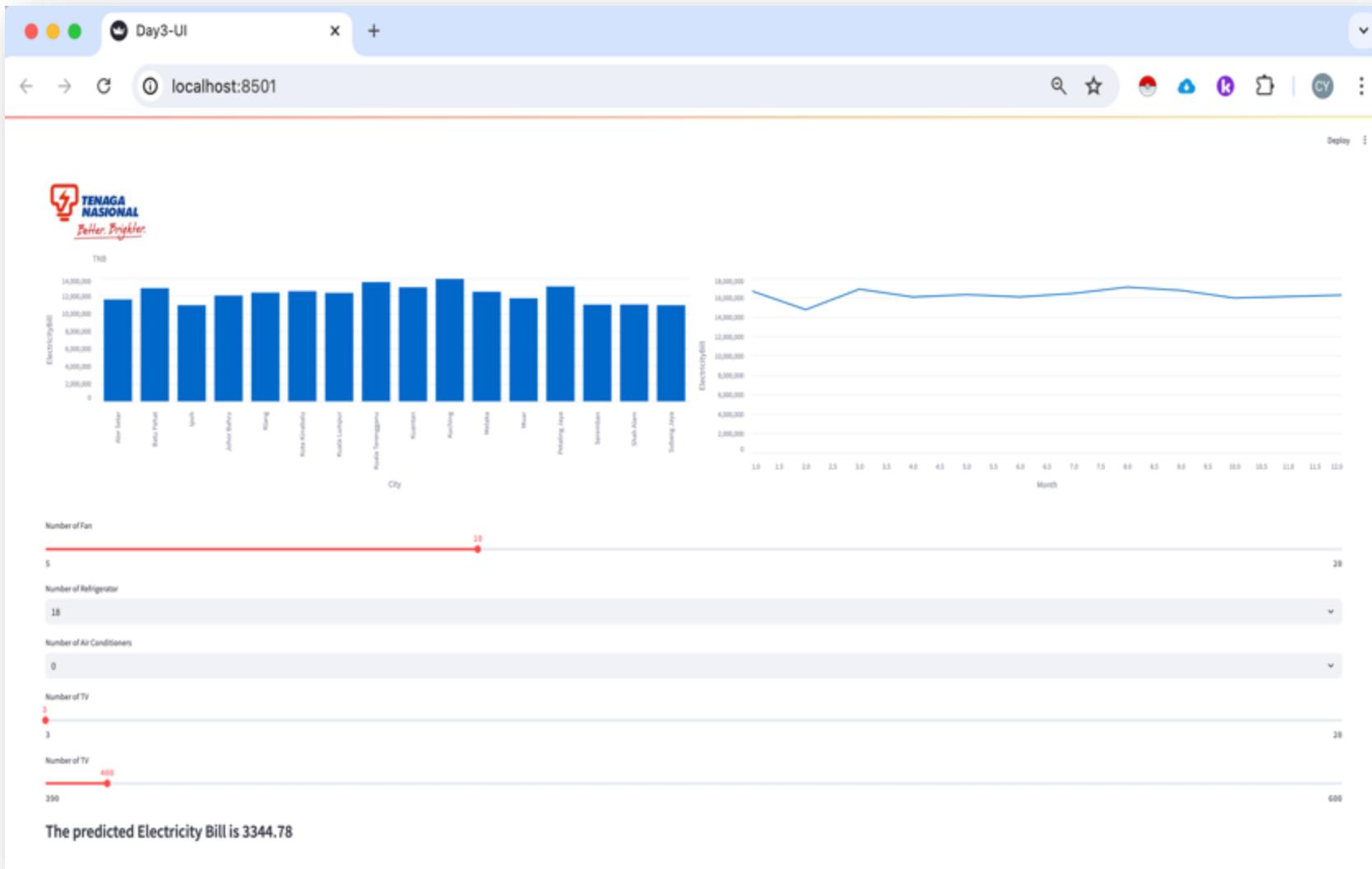
Create App



Create App



Finally Deployed



THANKS