

Streamlit Deployment

Speakers: Liu Tianqi,
Zheng Hairong

2025.10.11

CONTENT

CHAPTER ONE

0 1

Goals of This
Section

CHAPTER TWO

0 2

Data anylysis

CHAPTER THREE

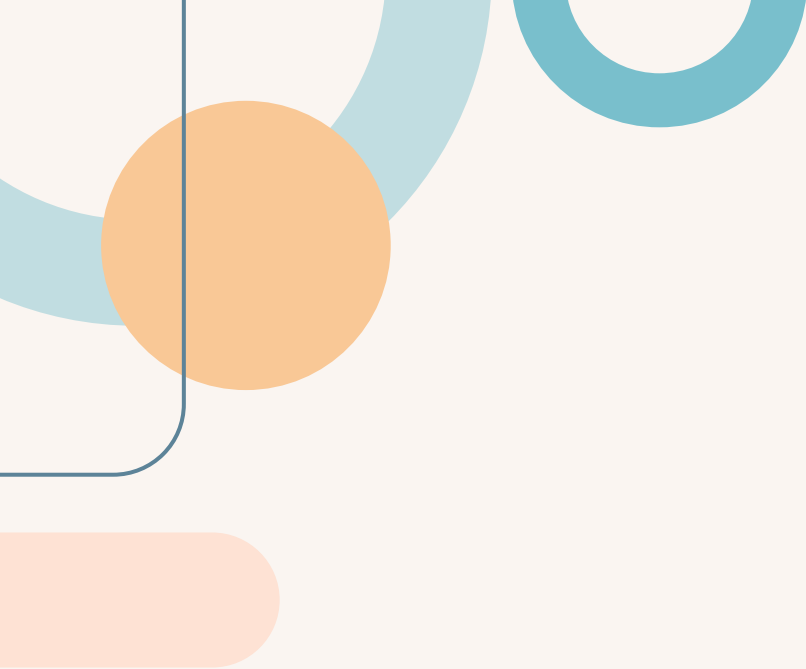
0 3

Deploy through
streamlit

CHAPTER FOUR

0 4

Assignment



CHAPTER ONE

0 1

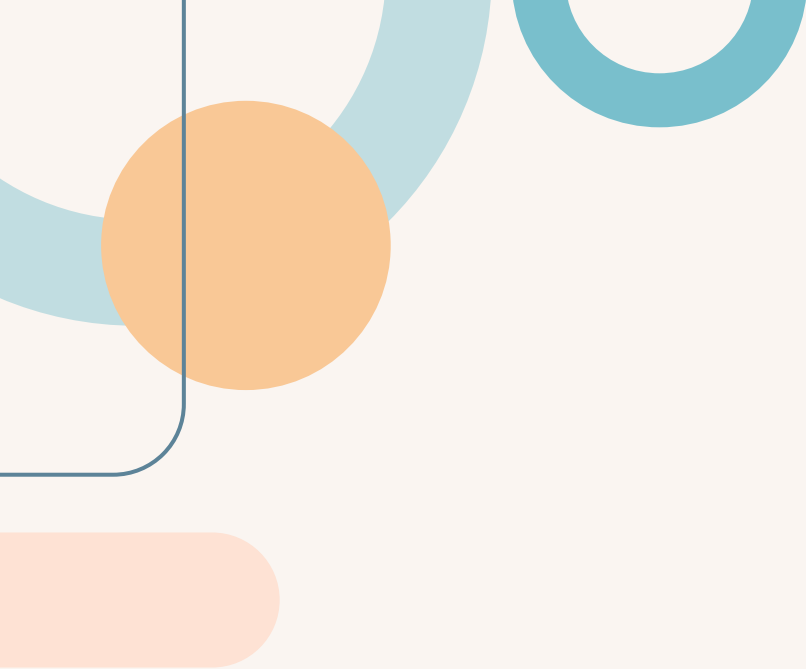
Goals of This Section

Goals of This Section

Five main goals

1. Master the preparatory steps for Streamlit development.
2. Learn to analyze a dataset (world cities dataset) using Jupyter Notebook.
3. Develop a functional Streamlit single-page app ("World Cities App").
4. Understand critical Streamlit operational logic.
5. Grasp the app deployment workflow.



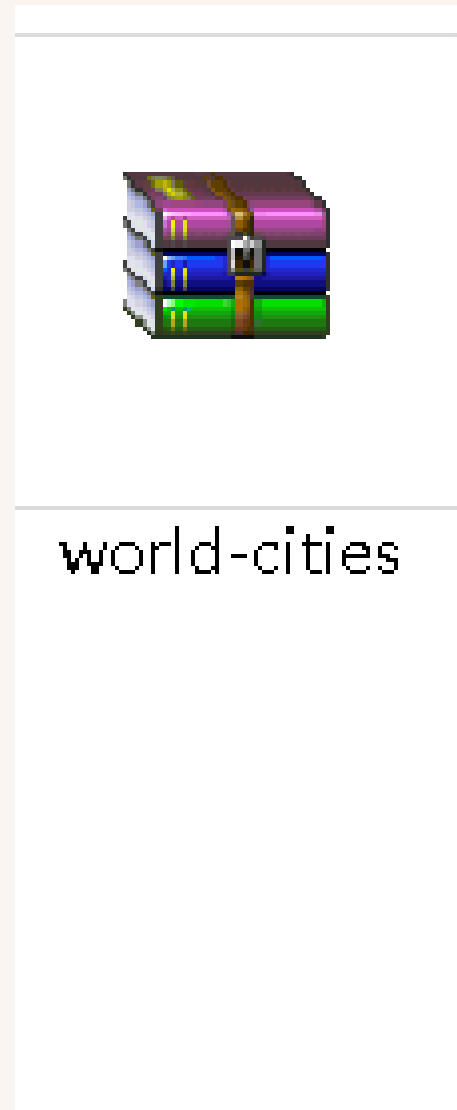
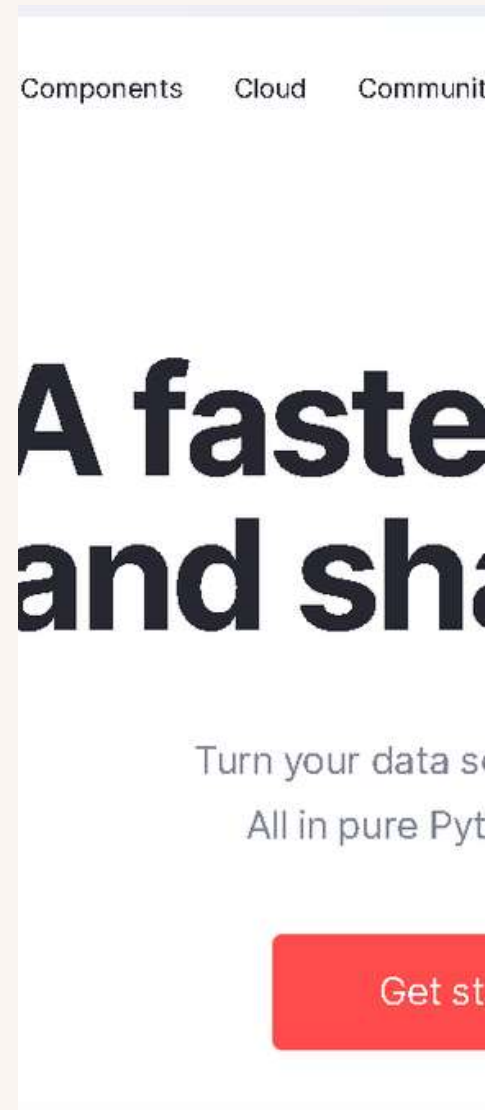


CHAPTER TWO

0 2

Data anylysis

Preparation



1. <https://github.com/>
2. <https://streamlit.io/> (make sure the same account with github)
3. Download the file world-cities on canvas and unzip it

Create new env
and install
streamlit
matplotlib
seaborn

01

Create a new
notebook
cities.ipynb
to analyze the
dataset first

02

Choose the env
you created just
befor.
(If unavailable,
use local env)

03

Data anylysis

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

df = pd.read_csv('worldcities.csv')
df.head()
```

	city	city_ascii	lat	lon	country	iso2	iso3	admin_name	capital	population	id
0	Tokyo	Tokyo	35.6897	139.6922	Japan	JP	JPN	Tōkyō	primary	37.977	1392685764
1	Jakarta	Jakarta	-6.2146	106.8451	Indonesia	ID	IDN	Jakarta	primary	34.540	1360771077
2	Delhi	Delhi	28.6600	77.2300	India	IN	IND	Delhi	admin	29.617	1356872604
3	Mumbai	Mumbai	18.9667	72.8333	India	IN	IND	Mahārāshtra	admin	23.355	1356226629
4	Manila	Manila	14.6000	120.9833	Philippines	PH	PHL	Manila	primary	23.088	1608618140

Data anylysis

```
df.population.describe()
```

Output:

```
count    723.000000
```

```
mean      3.610043
```

```
std       4.029683
```

```
min       1.001205
```

```
25%       1.335808
```

```
50%       2.150000
```

```
75%       4.377700
```

```
max       37.977000
```

```
Name: population, dtype: float64
```

The capital column has the following values, which we will use to make a multi-select:

- primary - country's capital (e.g. Washington D.C.)
- admin - first-level admin capital (e.g. Little Rock, AR)
- minor - lower-level admin capital (e.g. Fayetteville, AR)
- nan if not a capital (nan is not a string - it's a keyword in Pandas for Null)

```
df.capital.unique()
```

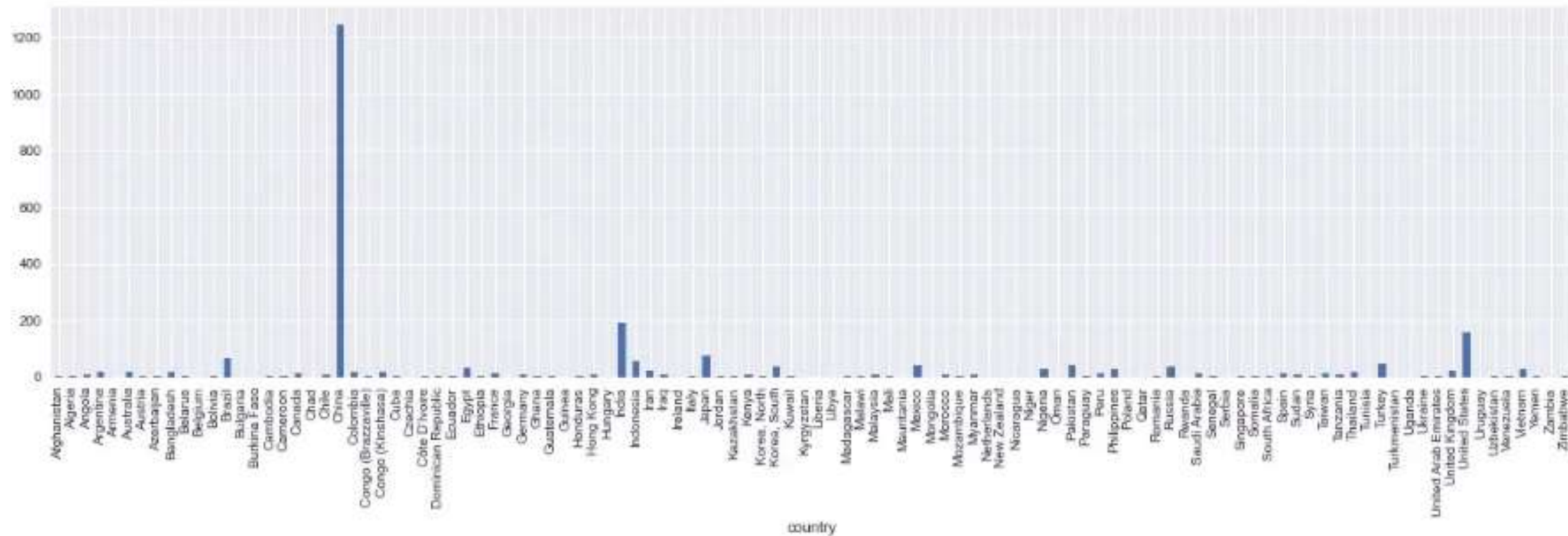
Output:

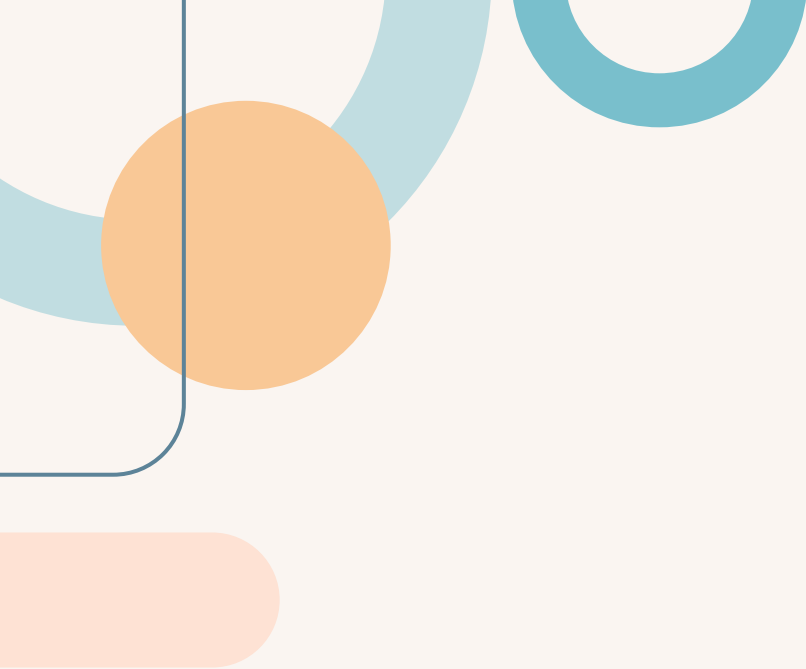
```
array(['primary', 'admin', nan, 'minor'], dtype=object)
```

Data anylysis

We can plot the population by country as follows:

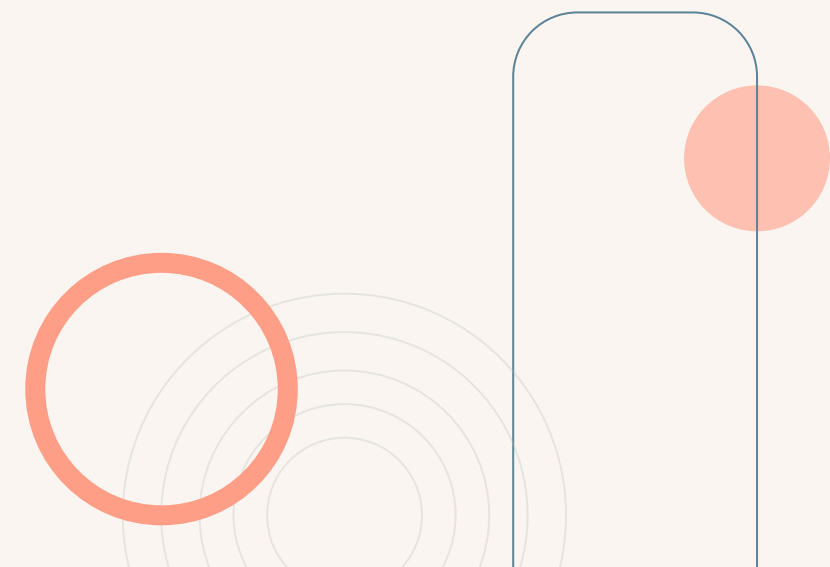
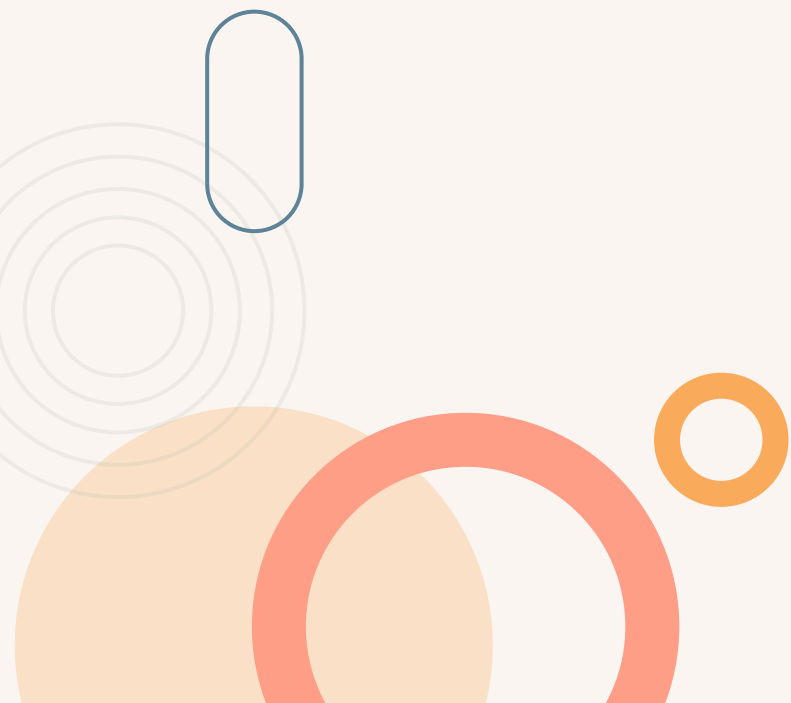
```
fig, ax = plt.subplots(figsize=(20, 5))
pop_sum = df.groupby('country')['population'].sum()
pop_sum.plot.bar(ax=ax)
```



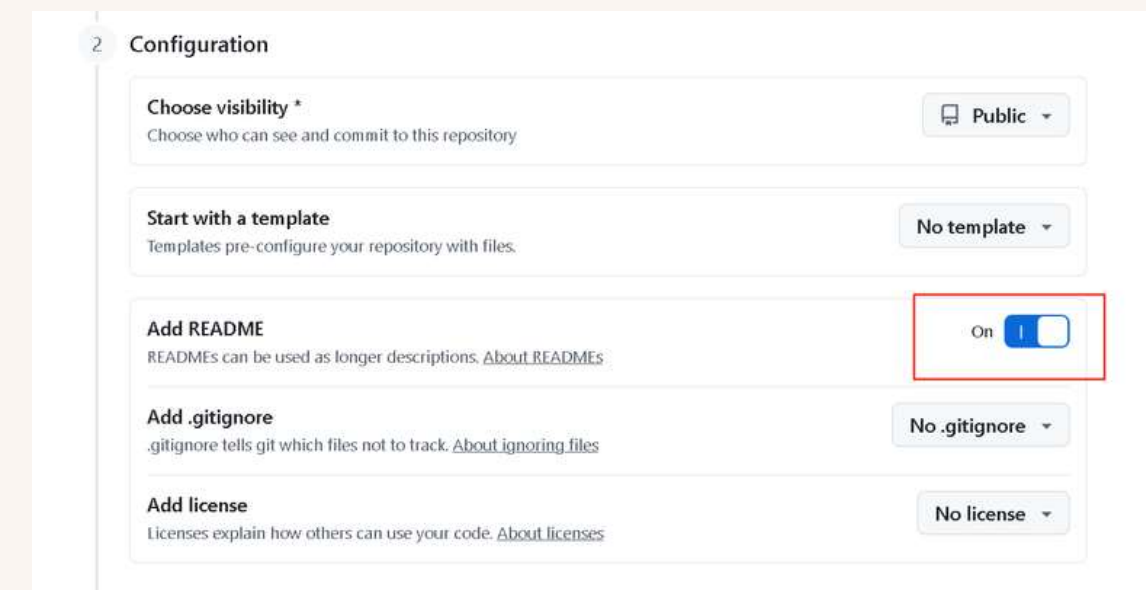
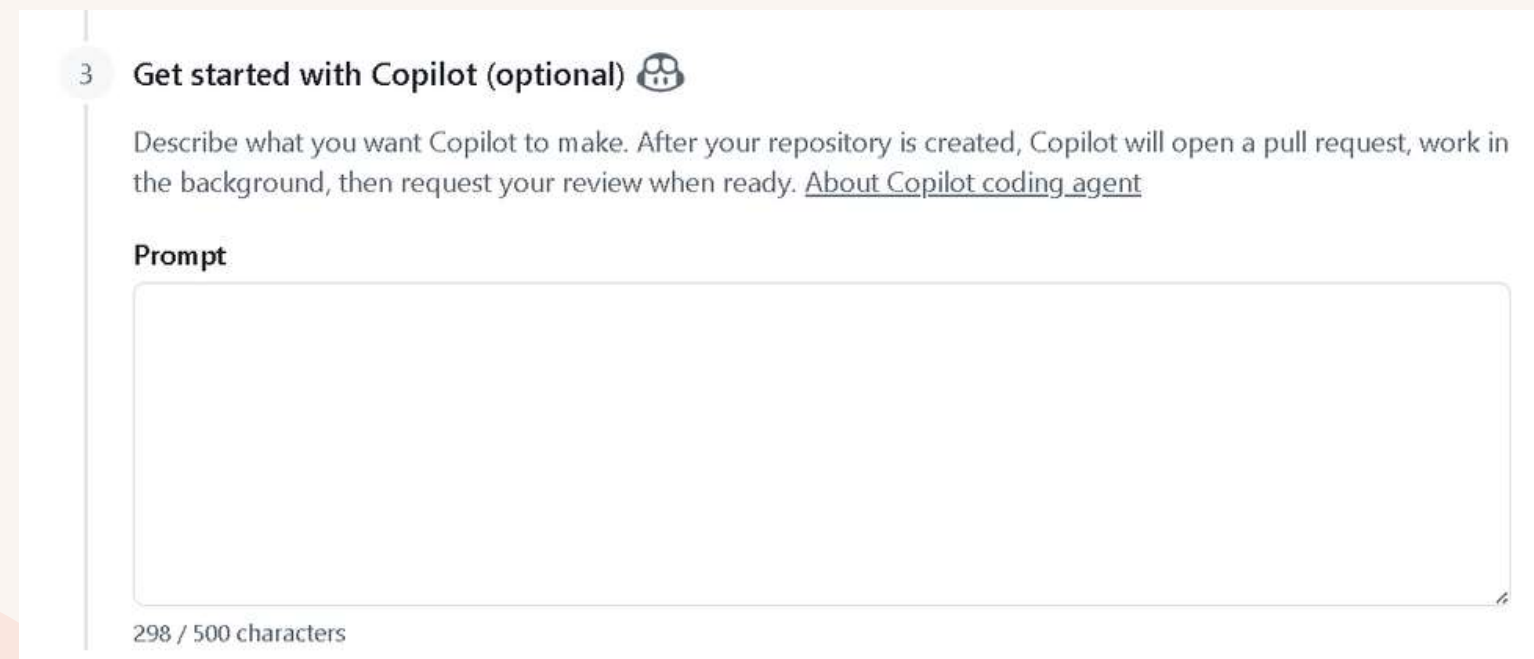
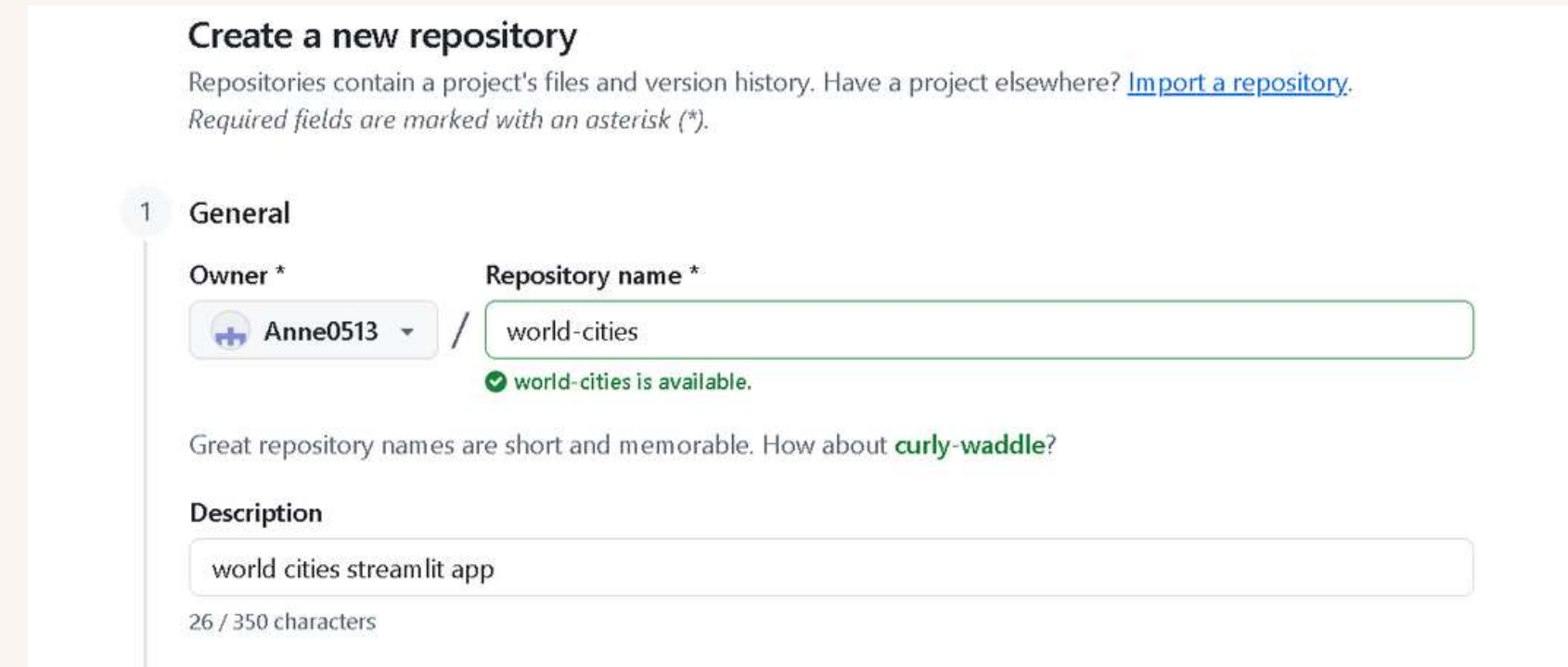
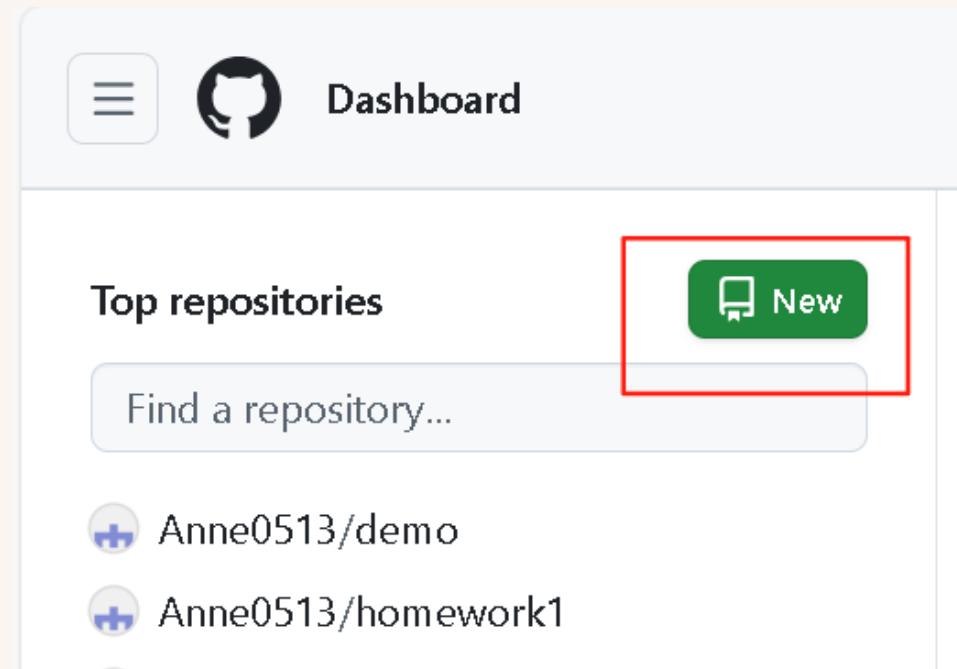


CHAPTER THREE 03

Deploy through streamlit



Deploy through streamlit



Deploy through streamlit

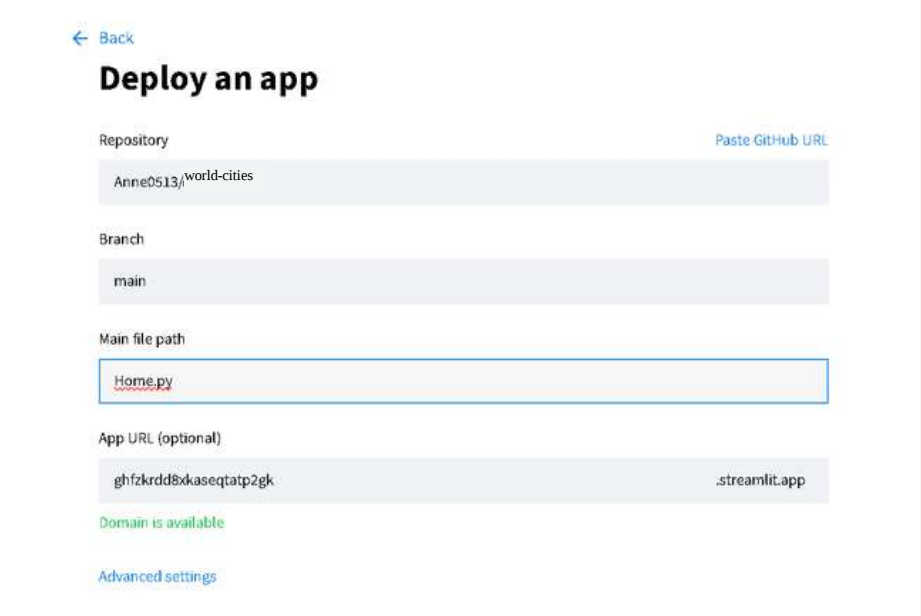
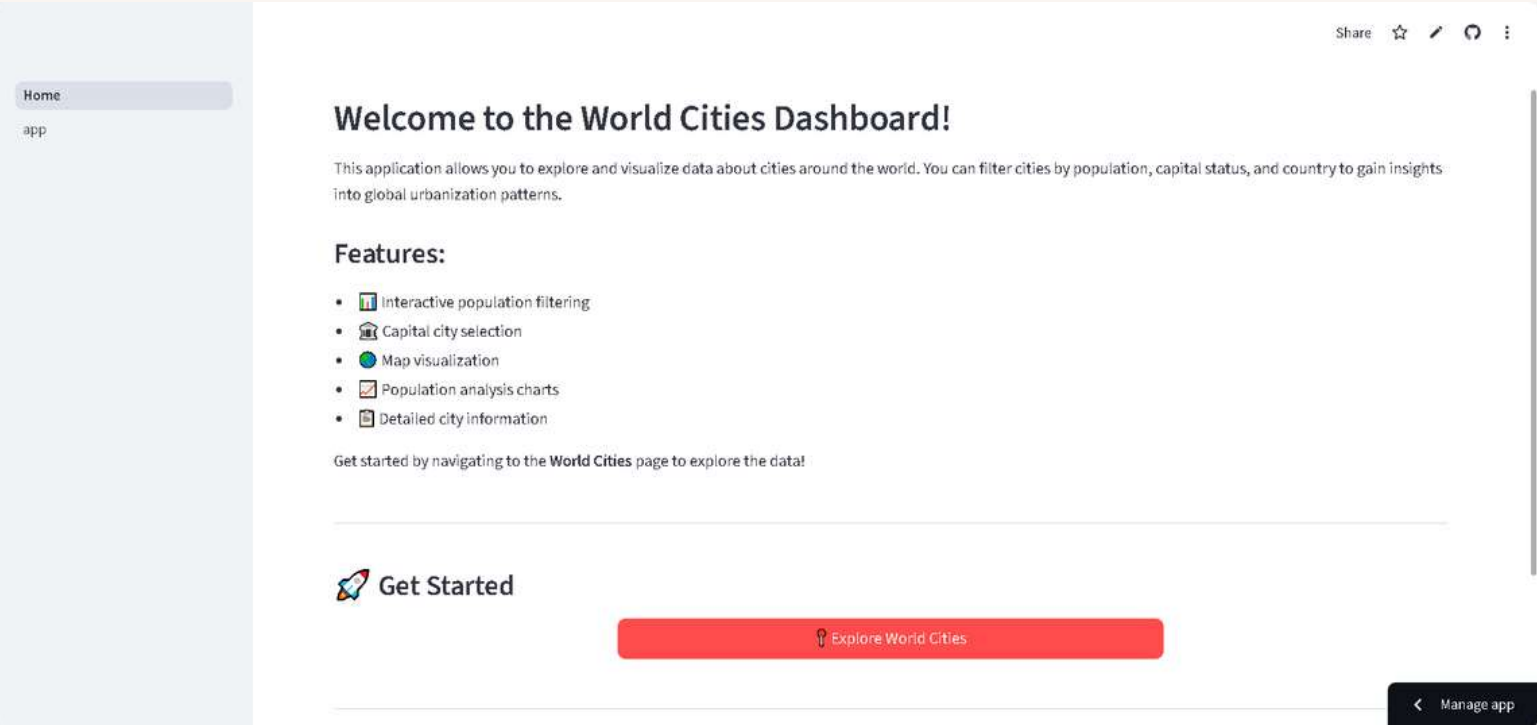
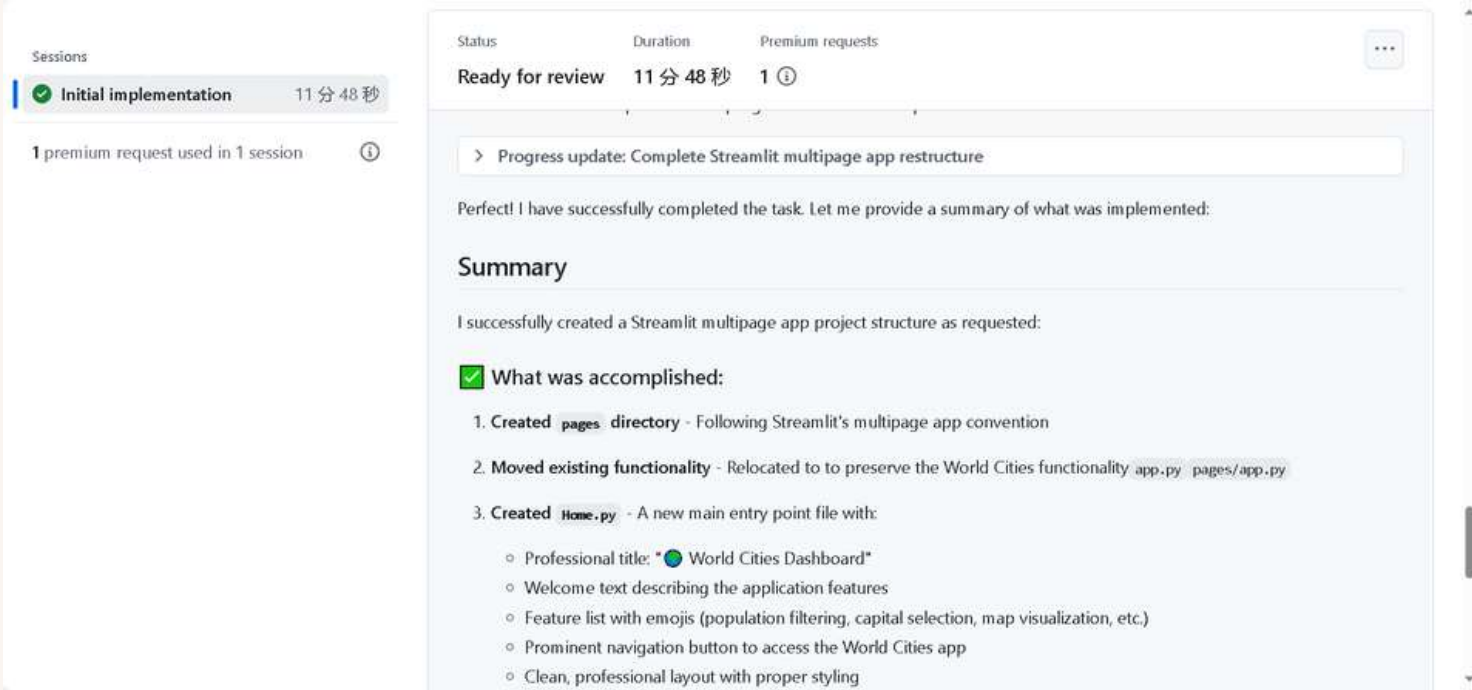
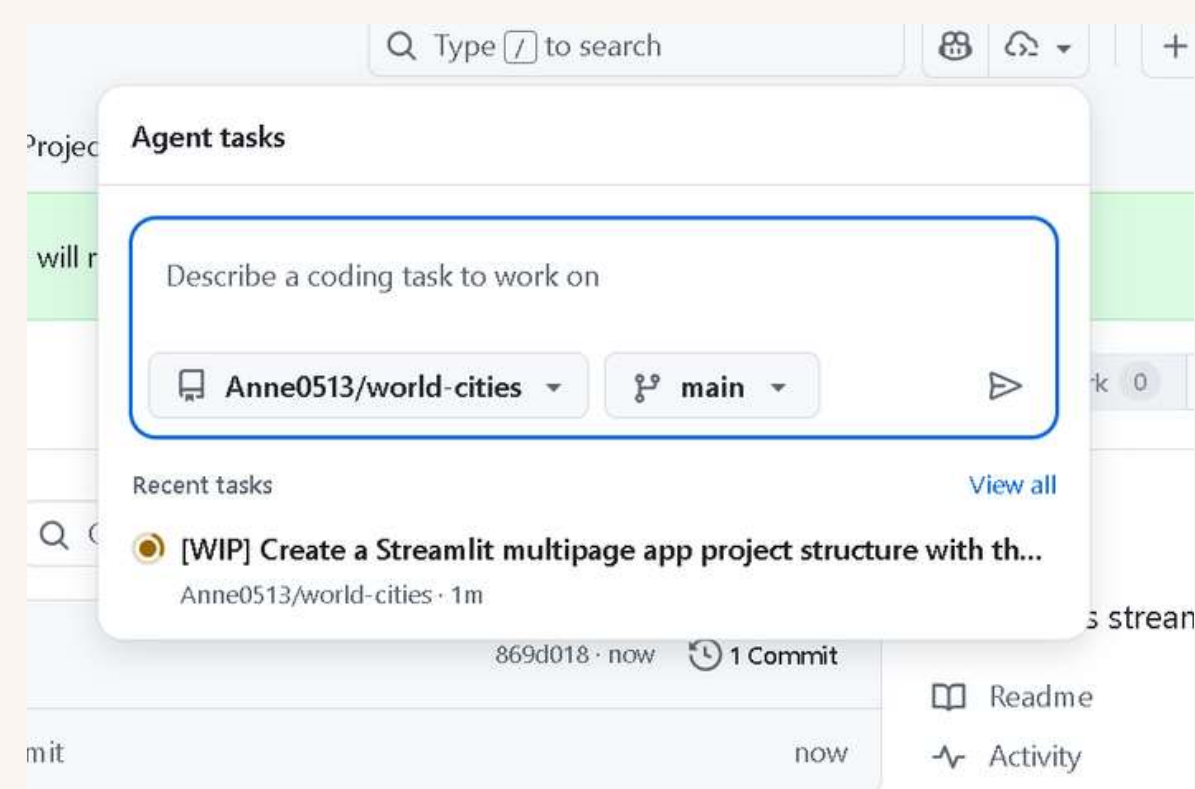
Create a Streamlit multipage app project structure with the following:

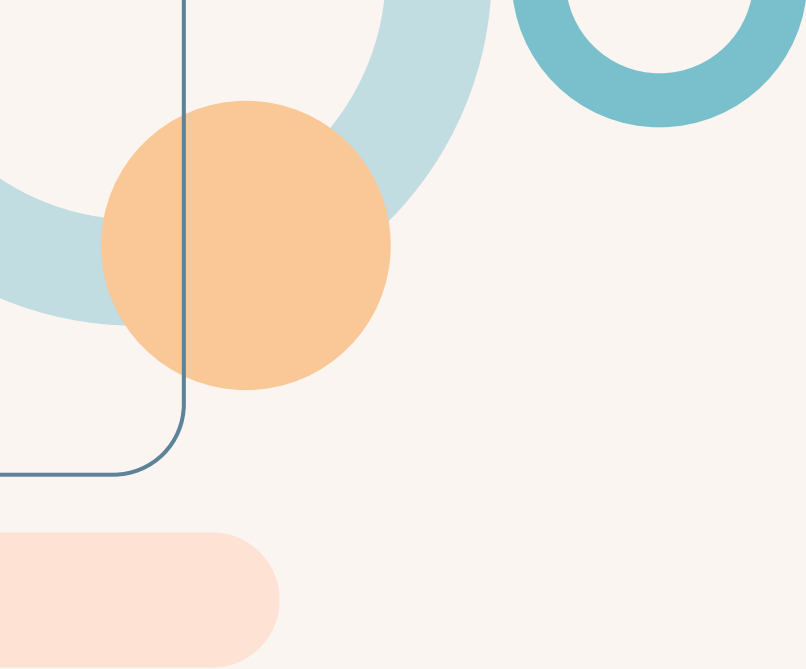
- A Home.py file that serves as the main entry point, displaying a title, welcome text, and a button to go to app.py.
- A "pages" folder where app.py will be located.

Only generate the Home.py content; I will add app.py later.

First, upload the files app.py, requirements.txt, and worldcities.csv to the GitHub repository. Then, use the above prompt in the Copilot agent chatbox.

Deploy through streamlit





CHAPTER FOUR

0 4

Assignment

Assignment

Using the Titanic dataset (train.csv), complete the following tasks with streamlit, matplotlib, and the Copilot agent:

1. Import the necessary libraries and apply the seaborn style.
2. Display the page title “Titanic App by [Your Name]”.
3. Read the train.csv file and display the entire dataframe in the Streamlit app.
4. Create three side-by-side box plots (figure size 15×5):
 - Show the ticket fare (Fare) distribution for each passenger class (Pclass = 1, 2, 3).
 - Each subplot must have appropriate x-axis and y-axis labels.
5. Display the plotted figure in the Streamlit page.
6. Use the Copilot agent to generate a simple home page (home.py), which contains a button that navigates to your main Titanic app (app.py).