TAGS SCRAPING:

```
import requests
from bs4 import BeautifulSoup
import streamlit as st
import pandas as pd


st.title("WEB SCRAPING")
url = st.text_input('Enter the URL', "https://www.w3schools.com/")

st.title("Select the tags to scrape")

tags = ['a', 'p', 'meta', 'body', 'label', 'i', 'br']
selected_tags = st.multiselect('Select tags', tags)

links = paragraphs = bodies = labels = italic_text = line_breaks = []
soup = None
fetch_button = st.button("Scrape")
if fetch_button:
    if not url == '':
        response = requests.get(url)
        soup = BeautifulSoup(response.content, "html.parser")

        data = {tag: [] for tag in selected_tags}

        for tag in selected_tags:
            if tag == 'a':
                links = soup.find_all('a')
                for link in links:
                    href = link.get('href')
                    data['a'].append(href)

            if tag == 'p':
                paragraphs = soup.find_all("p")
                for paragraph in paragraphs:
                    text = paragraph.text
                    data['p'].append(text)

            if tag == 'meta':
                meta_tags = soup.find_all("meta")
                for meta_tag in meta_tags:
                    content = meta_tag.get('content')
                    data['meta'].append(content)

            if tag == 'body':
```

```
                bodies = soup.find_all("body")
                for body in bodies:
                    body_text = body.text
                    data['body'].append(body_text)

            if tag == 'label':
                labels = soup.find_all("label")
                for label in labels:
                    label_text = label.text
                    data['label'].append(label_text)

            if tag == 'i':
                italic_text = soup.find_all("i")
                for italic in italic_text:
                    italicized_text = italic.text
                    data['i'].append(italicized_text)

            if tag == 'br':
                line_breaks = soup.find_all("br")
                for line_break in line_breaks:
                    data['br'].append("<br>")

        # Fill lists with empty strings or None to ensure equal length
        max_length = max(len(lst) for lst in data.values())
        for key, value in data.items():
            if len(value) < max_length:
                value.extend([''] * (max_length - len(value)))

        if data:
            df = pd.DataFrame(data)
            st.write(df)
        else:
            st.write("No tags selected.")
```

WEB SCRAPING: (USING BEAUTIFUL SOUP AND STREAMLIT)

```
import csv
import requests
from bs4 import BeautifulSoup
import streamlit as st

st.title("Book Details:")

# Select the CSV file
csv_file = st.file_uploader("Upload CSV file", type=["csv"])
```

```
if csv_file is not None:
    try:
        # Read data from CSV
        with open(csv_file.name, 'r') as file:
            csv_reader = csv.reader(file)
            data = []
            titles = []
            for index, row in enumerate(csv_reader):
                if index > 0:  # to skip the first row of the CSV file, assuming it
contains headers, and process the subsequent data rows
                    title = row[0]
                    url = row[1]
                    titles.append(title)
                    data.append([title, url])

        # Display select box for title
        selected_title = st.selectbox("Select The Book Title:", options=titles)

        # Fetch novel details based on selected title
        for row in data:
            if row[0] == selected_title:
                title = row[0]
                url = row[1]

                response = requests.get(url)
                soup = BeautifulSoup(response.content, 'html.parser')

                name_author = soup.find('div', class_='insert-details')
                author_label = name_author.find('label', text='Author:')
                author_name = author_label.next_sibling.strip()  # It is essentially the
element that comes immediately after the current element at the same level within the HTML
structure.

                year_r = soup.find('label', text='Year Released:')
                year = year_r.next_sibling.strip()

                st.subheader("Selected Book Details:")
                st.write("Title:", title)
                st.write("URL:", url)
                st.write("Author Name:", author_name)
                st.write("Year Released:", year)

    except UnicodeDecodeError:
        st.error("Unable to decode the CSV file. Please make sure the file is encoded
properly.")
```

STREAMLIT MAP:

```
import streamlit as st
import pandas as pd
df = pd.DataFrame({
    "id":[1,2,3],
    "name":['Sheza','Wardah','Kubrah']

})

st.write("My First streamlit application")

st.write(df)

import streamlit as st
import pandas as pd
import numpy as np

df = pd.DataFrame(
    np.random.randn(1000, 2) / [50, 50] + [37.76, -122.4],
    columns=['lat', 'lon'])

st.map(df)
```