3/multipage.py

```python
import streamlit as st

st.set_page_config(page_title="Multipage Demo")


def intro():
    import streamlit as st

    st.write("# Welcome to Streamlit!  ")
    st.sidebar.success("Select a demo above.")

    st.markdown(
        """
        Streamlit is an open-source app framework built specifically for
        Machine Learning and Data Science projects.

        **  Select a demo from the dropdown on the left** to see some examples
        of what Streamlit can do!

        ### Want to learn more?

        - Check out [streamlit.io](https://streamlit.io)
        - Jump into our [documentation](https://docs.streamlit.io)
        - Ask a question in our [community
          forums](https://discuss.streamlit.io)

        ### See more complex demos

        - Explore a [New York City rideshare dataset](https://github.com/streamlit/demo-uber-nyc-pickups)
    """
    )

def plotting_demo():
    import streamlit as st
    import time
    import numpy as np

    st.markdown(f'# {list(page_names_to_funcs.keys())[1]}')
    st.write(
        """
        This demo illustrates a combination of plotting and animation with
Streamlit. We're generating a bunch of random numbers in a loop for around
5 seconds. Enjoy!
"""
    )

    progress_bar = st.sidebar.progress(0)
    status_text = st.sidebar.empty()
    last_rows = np.random.randn(1, 1)
    chart = st.line_chart(last_rows)

    for i in range(1, 101):
        new_rows = last_rows[-1, :] + np.random.randn(5, 1).cumsum(axis=0)
        status_text.text("%i%% Complete" % i)
        chart.add_rows(new_rows)
        progress_bar.progress(i)
        last_rows = new_rows
        time.sleep(0.05)

    progress_bar.empty()

    # Streamlit widgets automatically run the script from top to bottom. Since
    # this button is not connected to any other logic, it just causes a plain
    # rerun.
    st.button("Re-run")


def data_frame_demo():
    import streamlit as st
    import pandas as pd
    import altair as alt

    from urllib.error import URLError

    st.markdown(f"# {list(page_names_to_funcs.keys())[3]}")
    st.write(
        """
        This demo shows how to use `st.write` to visualize Pandas DataFrames.

(Data courtesy of the [UN Data Explorer](http://data.un.org/Explorer.aspx).)
"""
    )

    @st.cache_data
    def get_UN_data():
        AWS_BUCKET_URL = "http://streamlit-demo-data.s3-us-west-2.amazonaws.com"
        df = pd.read_csv(AWS_BUCKET_URL + "/agri.csv.gz")
        return df.set_index("Region")

    try:
        df = get_UN_data()
        countries = st.multiselect(
            "Choose countries", list(df.index), ["China", "United States of America"]
        )
        if not countries:
            st.error("Please select at least one country.")
        else:
            data = df.loc[countries]
            data /= 1000000.0
            st.write("### Gross Agricultural Production ($B)", data.sort_index())

            data = data.T.reset_index()
            data = pd.melt(data, id_vars=["index"]).rename(
                columns={"index": "year", "value": "Gross Agricultural Product ($B)"}
            )
            chart = (
                alt.Chart(data)
                .mark_area(opacity=0.3)
                .encode(
                    x="year:T",
                    y=alt.Y("Gross Agricultural Product ($B):Q", stack=None),
                    color="Region:N",
                )
            )
            st.altair_chart(chart, use_container_width=True)
    except URLError as e:
        st.error(
            """
            **This demo requires internet access.**

            Connection error: %s
        """
            % e.reason
        )

def mapping_demo():
    import streamlit as st
    import pandas as pd
    import pydeck as pdk

    from urllib.error import URLError

    st.markdown(f"# {list(page_names_to_funcs.keys())[2]}")
    st.write(
        """
        This demo shows how to use
[`st.pydeck_chart`](https://docs.streamlit.io/develop/api-reference/charts/st.pydeck_chart)
to display geospatial data.
"""
    )

    @st.cache_data
    def from_data_file(filename):
        url = (
            "http://raw.githubusercontent.com/streamlit/"
            "example-data/master/hello/v1/%s" % filename
        )
        return pd.read_json(url)

    try:
        ALL_LAYERS = {
            "Bike Rentals": pdk.Layer(
                "HexagonLayer",
                data=from_data_file("bike_rental_stats.json"),
                get_position=["lon", "lat"],
                radius=200,
                elevation_scale=4,
                elevation_range=[0, 1000],
                extruded=True,
            ),
            "Bart Stop Exits": pdk.Layer(
                "ScatterplotLayer",
                data=from_data_file("bart_stop_stats.json"),
                get_position=["lon", "lat"],
                get_color=[200, 30, 0, 160],
                get_radius="[exits]",
                radius_scale=0.05,
            ),
            "Bart Stop Names": pdk.Layer(
                "TextLayer",
                data=from_data_file("bart_stop_stats.json"),
                get_position=["lon", "lat"],
                get_text="name",
                get_color=[0, 0, 0, 200],
                get_size=15,
                get_alignment_baseline="'bottom'",
            ),
            "Outbound Flow": pdk.Layer(
                "ArcLayer",
                data=from_data_file("bart_path_stats.json"),
                get_source_position=["lon", "lat"],
                get_target_position=["lon2", "lat2"],
                get_source_color=[200, 30, 0, 160],
                get_target_color=[200, 30, 0, 160],
                auto_highlight=True,
                width_scale=0.0001,
                get_width="outbound",
                width_min_pixels=3,
                width_max_pixels=30,
            ),
        }
        st.sidebar.markdown("### Map Layers")
        selected_layers = [
            layer
            for layer_name, layer in ALL_LAYERS.items()
            if st.sidebar.checkbox(layer_name, True)
        ]
        if selected_layers:
            st.pydeck_chart(
                pdk.Deck(
                    map_style="mapbox://styles/mapbox/light-v9",
                    initial_view_state={
                        "latitude": 37.76,
                        "longitude": -122.4,
                        "zoom": 11,
                        "pitch": 50,
                    },
                    layers=selected_layers,
                )
            )
        else:
            st.error("Please choose at least one layer above.")
    except URLError as e:
        st.error(
            """
            **This demo requires internet access.**

            Connection error: %s
        """
            % e.reason
        )

page_names_to_funcs = {
    "-": intro,
    "Plotting Demo": plotting_demo,
    "Mapping Demo": mapping_demo,
    "DataFrame Demo": data_frame_demo
}

demo_name = st.sidebar.selectbox("Choose a demo", page_names_to_funcs.keys())
page_names_to_funcs[demo_name]()
```