اونيۏرسيتي مليسيا ڤهڠ السلطان عبدالله
UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

FAKULTI TEKNOLOGI KEJURUTERAAN ELEKTRIK DAN ELEKTRONIK

PROJECT

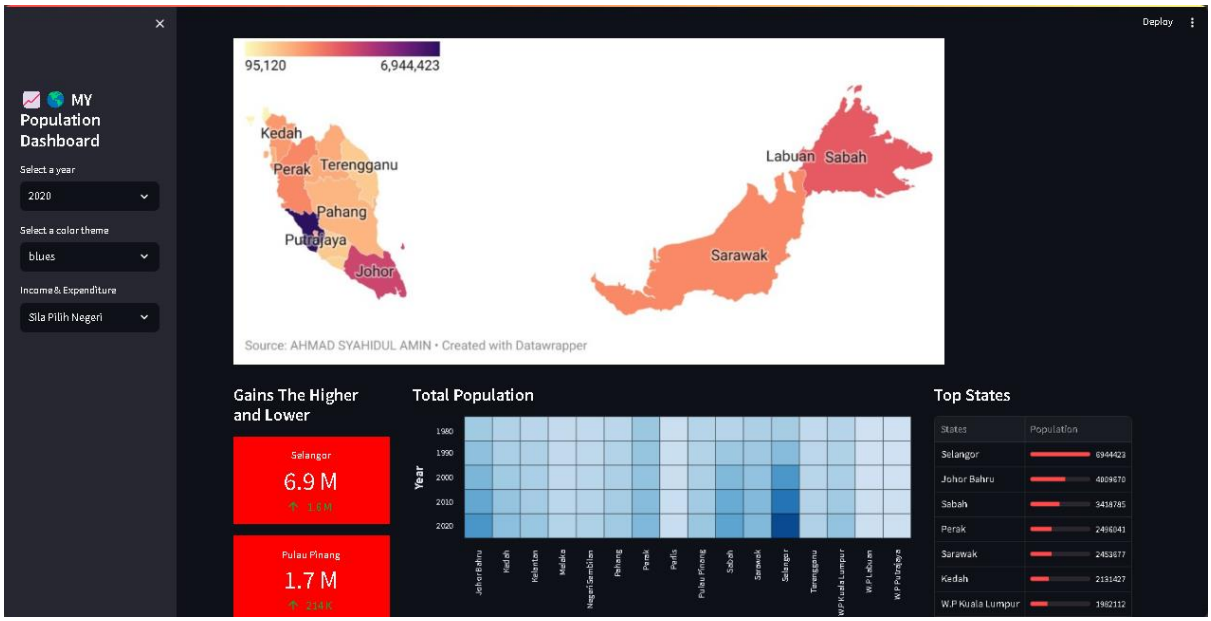
BVI 1234

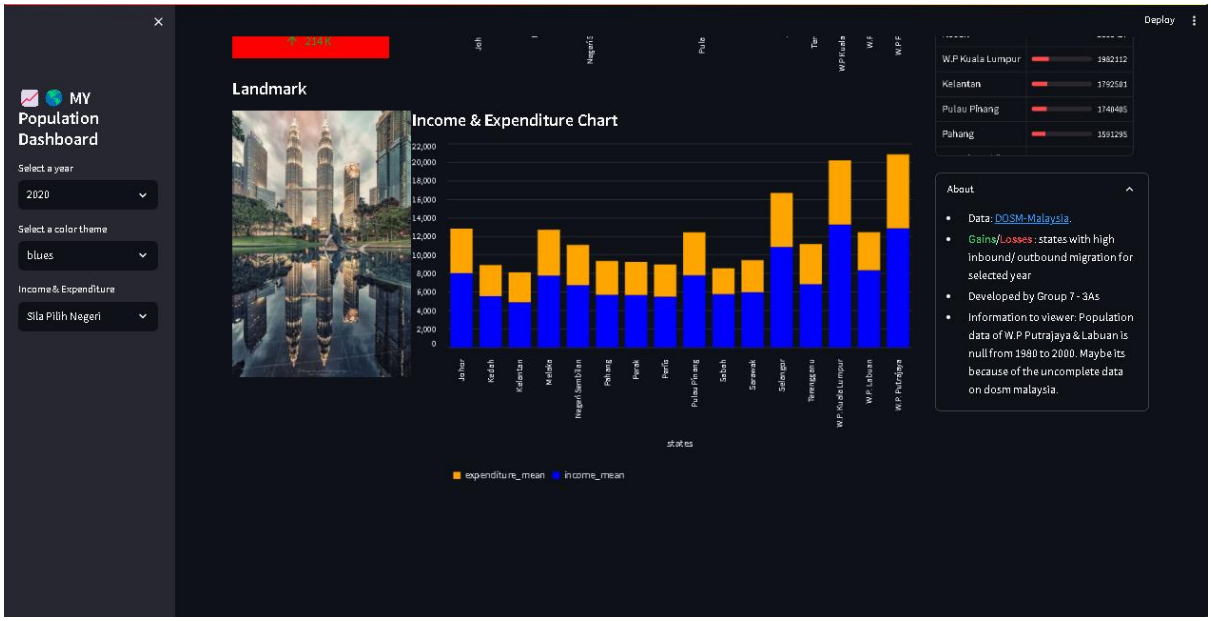
TECHOLOGY SYSTEM PROGRAMMING II

Group Of Members		
Bil	**Student Name**	**Student ID**
1	SYAMSYUL ARIFIN BIN AWI BOWO	VC23047
2	MUHAMMAD AMIRUL ADLY BIN MOHAMAD RASHIDI	VC23053
3	AHMAD SYADIL AMIN BIN AZNAL	VC23054
4	MOHAMAD AQMAR HARITH BIN KAMAROLZAMAN	VC23052

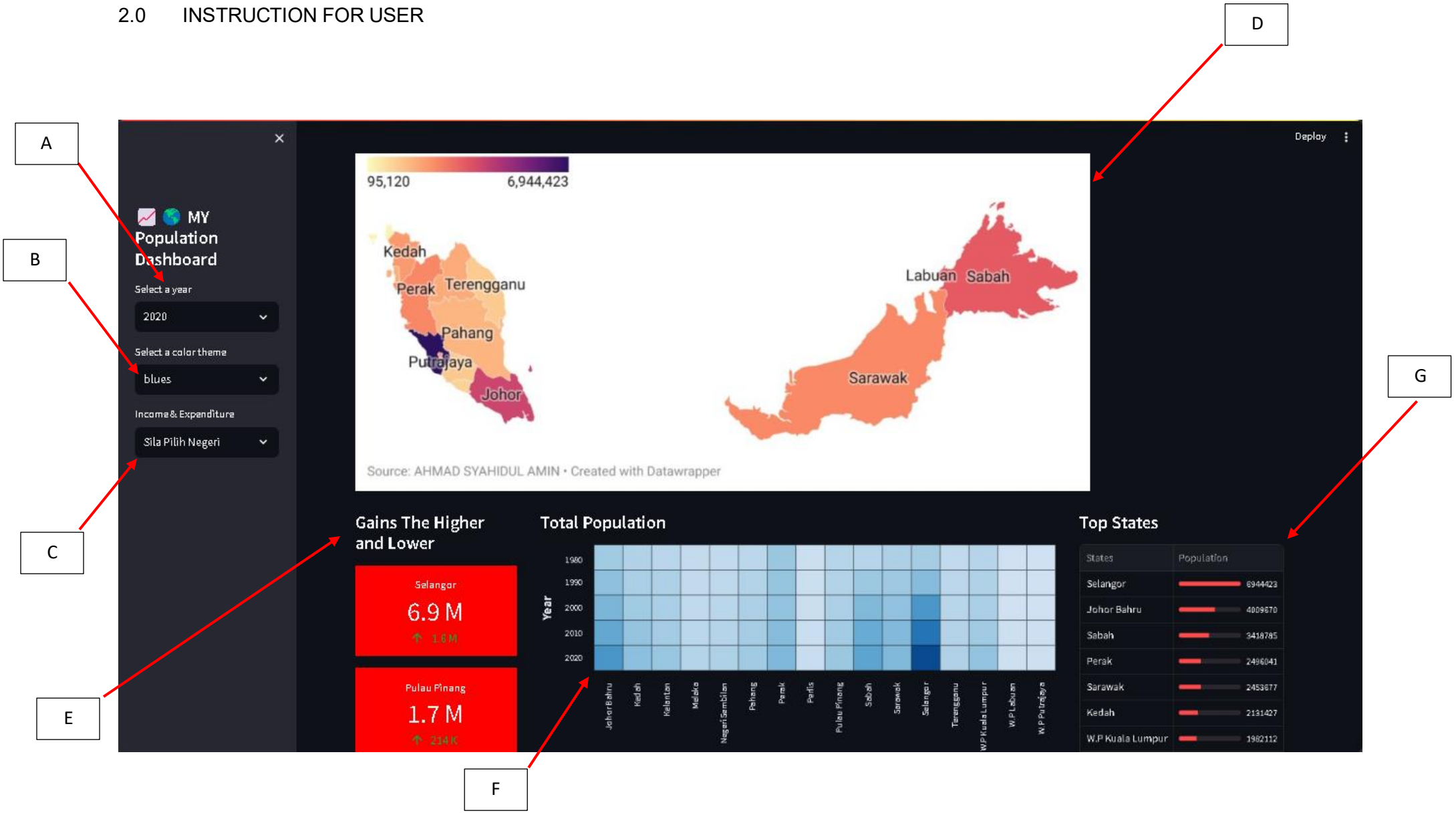
LIST OF TABLES

1.0 OVERVIEW OF THE PROJECT

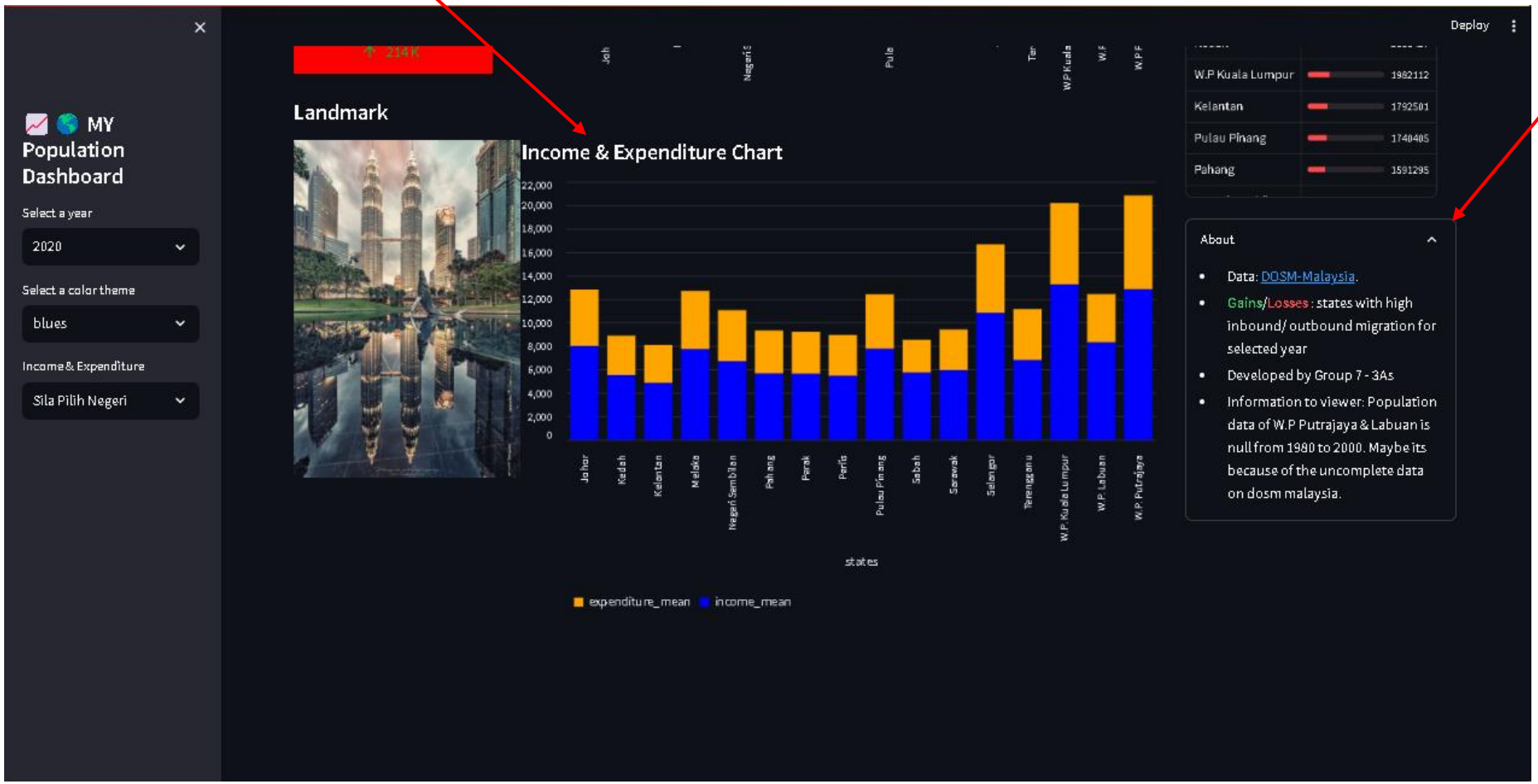




2.0 INSTRUCTION FOR USER

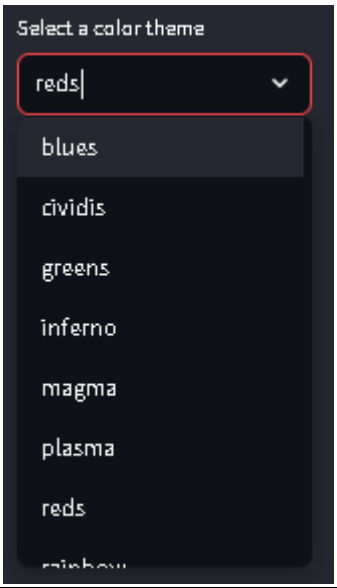


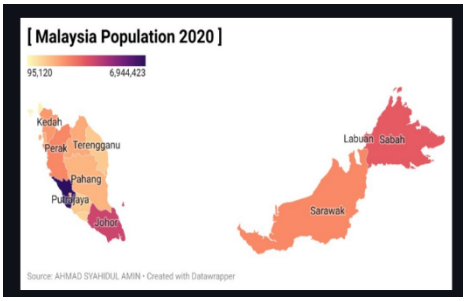
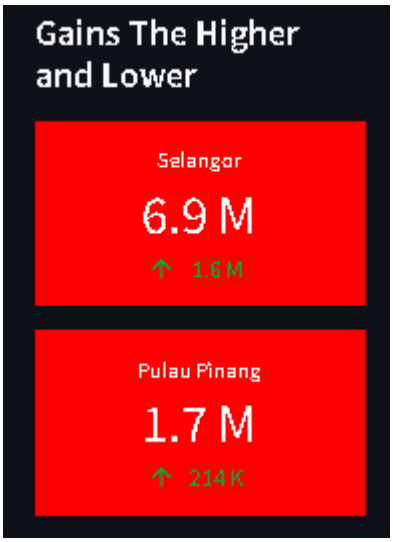
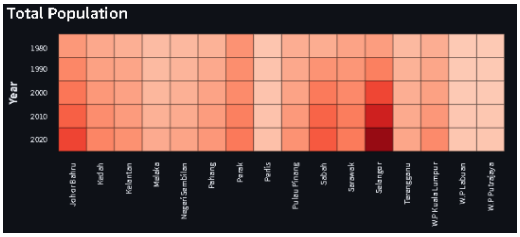
H

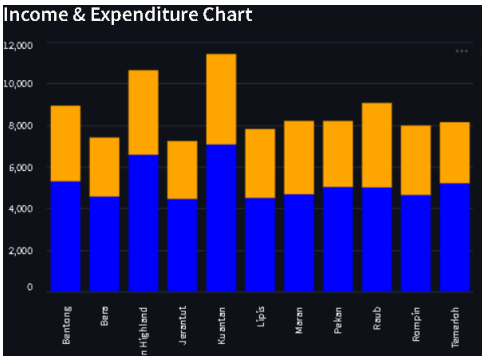
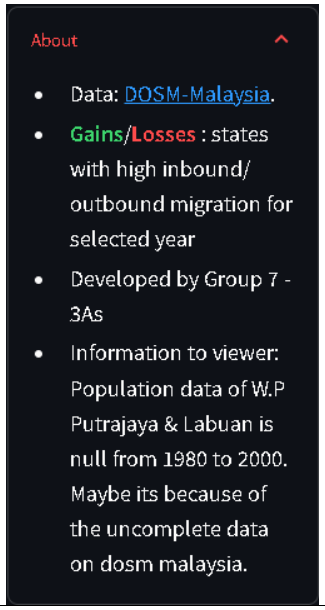

MY Population Dashboard
Select a year
2020
Select a color theme
blues
Income & Expenditure
Sila Pilih Negeri
Deploy
Landmark
Income & Expenditure Chart
W.P Kuala Lumpur 1982112
Kelantan 1792501
Pulau Pinang 1740405
Pahang 1591295
About
Data: DOSM-Malaysia.
Gains/Losses : states with high inbound/ outbound migration for selected year
Developed by Group 7 - 3As
Information to viewer: Population data of W.P Putrajaya & Labuan is null from 1980 to 2000. Maybe its because of the uncomplete data on dosm malaysia.
Johor
Kedah
Kelantan
Melaka
Negeri Sembilan
Pahang
Perak
Perlis
Pulau Pinang
Sabah
Sarawak
Selangor
Terengganu
W.P Kuala Lumpur
W.P Labuan
W.P Putrajaya
states
expenditure_mean
income_mean

I

MANUAL USER

Sequence	Instruction	Picture
A	Select year you want example 2020	
B	Option, select a colour theme you want example red	
C	Choose state you want to see income and expenditure example Pahang	

D	Dashboard visual image for year 2020	
E	Dashboard visual gains the higher and lower population in state of Malaysia	
F	Dashboard visual total population for each state in Malaysia using Heat map	
G	Dashboard visual top states in Malaysia for 2020	

H	Dashboard will visual income & expenditure district of Pahang as selected in income & expenditure	
I	This is information corner and disclaimer for dashboard Malaysia population	

3.0 CODE FOR PROJECT DASHBOARD MALAYSIA POPULATION

```
########################
# Import libraries
import streamlit as st
import pandas as pd
import altair as alt
########################
# Page configuration
st.set_page_config(
    page_title="MY Population Dashboard",
    page_icon="MY",
    layout="wide",
    initial_sidebar_state="expanded")
alt.themes.enable("dark")
########################
# CSS styling for gains the higher and lower matric
st.markdown("""
<style>
[data-testid="block-container"] {
    padding-left: 2rem;
    padding-right: 2rem;
    padding-top: 1rem;
    padding-bottom: 0rem;
    margin-bottom: -7rem;
}
[data-testid="stVerticalBlock"] {
    padding-left: 0rem;
    padding-right: 0rem;
}
[data-testid="stMetric"] {
    background-color: #FF0000;
```

```
    text-align: center;
    padding: 15px 0;
}
[data-testid="stMetricLabel"] {
  display: flex;
  justify-content: center;
  align-items: center;
}
[data-testid="stMetricDeltaIcon-Up"] {
    position: relative;
    left: 38%;
    -webkit-transform: translateX(-50%);
    -ms-transform: translateX(-50%);
    transform: translateX(-50%);
}
[data-testid="stMetricDeltaIcon-Down"] {
    position: relative;
    left: 38%;
    -webkit-transform: translateX(-50%);
    -ms-transform: translateX(-50%);
    transform: translateX(-50%);
}
</style>
""", unsafe_allow_html=True)
########################
# Load data
df_reshaped = pd.read_csv('my-population-1980-2020-reshaped.csv')
########################
# Sidebar
with st.sidebar:
    st.title('📈🌎 MY Population Dashboard')
    year_list = list(df_reshaped.year.unique())[::-1]
```

```
    selected_year = st.selectbox('Select a year', year_list)
    df_selected_year = df_reshaped[df_reshaped.year == selected_year]
    df_selected_year_sorted = df_selected_year.sort_values(by="population",
ascending=False)
    color_theme_list = ['blues', 'cividis', 'greens', 'inferno', 'magma', 'plasma', 'reds', 'rainbow'
'turbo', 'viridis']
    selected_color_theme = st.selectbox('Select a color theme', color_theme_list)
    pilihan_negeri = ["Sila Pilih Negeri","Johor",'Kedah','Kelantan','Melaka',
                      'Negeri Sembilan',"Pahang",'Perak','Pulau Pinang',
                      'Sabah',"Sarawak","Selangor",'Terengganu',]
    negeri_dipilih = st.selectbox("Income & Expenditure", options=pilihan_negeri)
#Image Map Malaysia
information = {'2020':{'MY population 2020.jpg'},
               '2010':{'MY population 2010.jpg'},
               '2000':{'MY population 2000.jpg'},
               '1990':{'MY population 1990.jpg'},
               '1980':{'MY population 1980.jpg'}}
if selected_year == 2020 :
    st.image('MY population 2020.jpg', width=1000)
elif selected_year == 2000 :
    st.image('MY population 2000.jpg', width=1000)
elif selected_year == 2010 :
    st.image('MY population 2010.jpg', width=1000)
elif selected_year == 1990 :
    st.image('MY population 1990.jpg', width=1000)
else :
    st.image('MY population 1980.jpg', width=1000)
# Heatmap function
def make_heatmap(input_df, input_y, input_x, input_color, input_color_theme):
    heatmap = alt.Chart(input_df).mark_rect().encode(
          y=alt.Y(f'{input_y}:O', axis=alt.Axis(title="Year", titleFontSize=18, titlePadding=15,
titleFontWeight=900, labelAngle=0)),
          x=alt.X(f'{input_x}:O', axis=alt.Axis(title="", titleFontSize=18, titlePadding=15,
titleFontWeight=900)),
```

```
            color=alt.Color(f'max({input_color}):Q',
                            legend=None,
                            scale=alt.Scale(scheme=input_color_theme)),
            stroke=alt.value('black'),
            strokeWidth=alt.value(0.25),
        ).properties(width=900
        ).configure_axis(
        labelFontSize=12,
        titleFontSize=12
        ).properties(height=300)
    return heatmap
# Convert population to text
def format_number(num):
    if num > 1000000:
        if not num % 1000000:
            return f'{num // 1000000} M'
        return f'{round(num / 1000000, 1)} M'
    return f'{num // 1000} K'
# Calculation year-over-year population migrations
def calculate_population_difference(input_df, input_year):
  selected_year_data = input_df[input_df['year'] == input_year].reset_index()
  previous_year_data = input_df[input_df['year'] == input_year - 10].reset_index()
  selected_year_data['population_difference'] =
selected_year_data.population.sub(previous_year_data.population, fill_value=0)
  return pd.concat([selected_year_data.states, selected_year_data.id,
selected_year_data.population, selected_year_data.population_difference],
axis=1).sort_values(by="population_difference", ascending=False)
########################
# Dashboard Main Panel
col = st.columns((1.5, 4.5, 2), gap='medium')


with col[0]:
    st.markdown('#### Gains/Losses')
```

```
    df_population_difference_sorted = calculate_population_difference(df_reshaped,
selected_year)

    if selected_year > 1980:

        first_state_name = df_population_difference_sorted.states.iloc[0]

        first_state_population =
format_number(df_population_difference_sorted.population.iloc[0])

        first_state_delta =
format_number(df_population_difference_sorted.population_difference.iloc[0])

    else:

        first_state_name = '-'

        first_state_population = '-'

        first_state_delta = ''

    st.metric(label=first_state_name, value=first_state_population, delta=first_state_delta)

    if selected_year > 1980:

        last_state_name = df_population_difference_sorted.states.iloc[-1]

        last_state_population =
format_number(df_population_difference_sorted.population.iloc[-1])

        last_state_delta =
format_number(df_population_difference_sorted.population_difference.iloc[-1])

    else:

        last_state_name = '-'

        last_state_population = '-'

        last_state_delta = ''

    st.metric(label=last_state_name, value=last_state_population, delta=last_state_delta)

    st.markdown('#### Landmark')

    st.image('klcc.jpeg', width=200)#pic for 100% zoom/default zoom

with col[1]:

    st.markdown('#### Total Population')

    heatmap = make_heatmap(df_reshaped, 'year', 'states', 'population',
selected_color_theme)

    st.altair_chart(heatmap, use_container_width=True)

     #Bar Chart for Income & Expenditure Data

    st.markdown('#### Income & Expenditure Chart')
```

```
    if negeri_dipilih == 'Johor':
        df = pd.read_csv('income_johor_district.csv')
        st.bar_chart(df, x='District',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Kedah':
        df = pd.read_csv('income_kedah_district.csv')
        st.bar_chart(df, x='District',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Kelantan':
        df = pd.read_csv('income_kelantan_district.csv')
        st.bar_chart(df, x='District',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Melaka':
        df = pd.read_csv('income_melaka_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Negeri Sembilan':
        df = pd.read_csv('income_N9_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Pahang':
        df = pd.read_csv('income_pahang_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Perak':
        df = pd.read_csv('income_perak_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Pulau Pinang':
        df = pd.read_csv('income_PP_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Sabah':
        df = pd.read_csv('income_sabah_district.csv')
```

```
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Sarawak':
        df = pd.read_csv('income_sarawak_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Selangor':
        df = pd.read_csv('income_selangor_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    elif negeri_dipilih == 'Terengganu':
        df = pd.read_csv('income_terangganu_district.csv')
        st.bar_chart(df, x='district',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
    else:
        df = pd.read_csv('MY income dan expend 2019.csv')
        st.bar_chart(df, x='states',
y=['income_mean','expenditure_mean'],color=['#FFA500','#0000FF'],width=300, height=500,)
with col[2]:
    st.markdown('#### Top States')
    st.dataframe(df_selected_year_sorted,
                 column_order=("states", "population"),
                 hide_index=True,
                 width=None,
                 column_config={
                   "states": st.column_config.TextColumn(
                       "States",
                   ),
                   "population": st.column_config.ProgressColumn(
                       "Population",
                       format="%f",
                       min_value=0,
                       max_value=max(df_selected_year_sorted.population),
                    )}
```

```
            )

    #information and disclaimer

    with st.expander('About', expanded=True):

        st.write('''

            - Data: [DOSM-Malaysia](https://github.com/dosm-malaysia/data-open/tree/main/datasets/census).

            - :green[**Gains**]/:red[**Losses**] : states with high inbound/ outbound migration for selected year

            - Developed by Group 7 - 3As

            - Information to viewer: Population data of W.P Putrajaya & Labuan is null from 1980 to 2000. Maybe its because of the uncomplete data on DOSM Malaysia.

            ''')
```