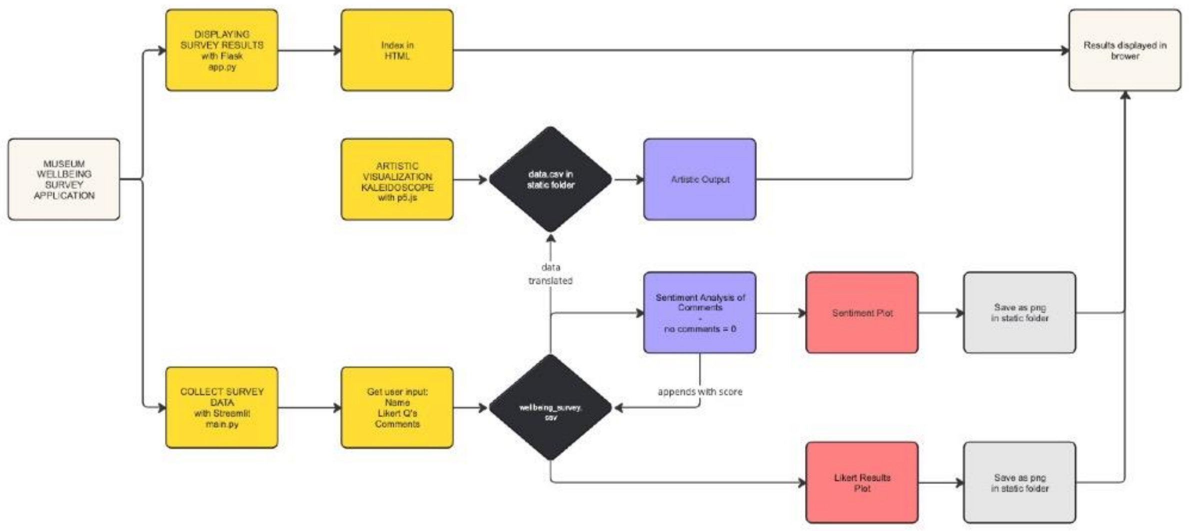
MUSEUM WELLBEING SURVEY DOCUMENTATION

Overview: The Wellbeing Survey Application is designed to collect, analyze, and visualize survey data related to participants' wellbeing experiences in the museum. It leverages sentiment analysis to gauge the tone of user comments and visualizes survey responses using Likert scale plot, sentiment plot, and an artistic visualization in the form of a digital kaleidoscope. This application is useful for museum organizations looking to collect and analyze survey data related to wellbeing, providing both qualitative and quantitative insights into participants' experiences.



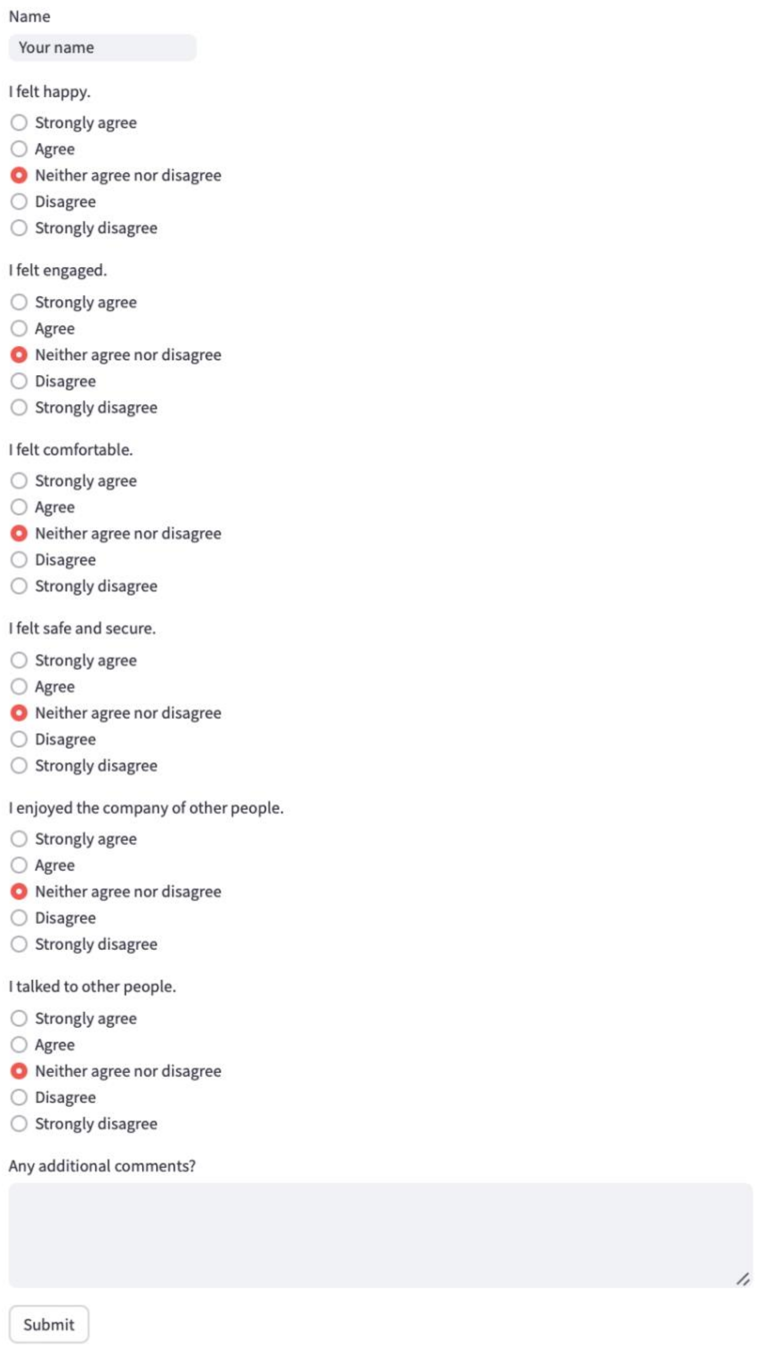
MUSEUM WELLBEING SURVEY APPLICATION
DISPLAYING SURVEY RESULTS with Flask app.py
Index in HTML
Results displayed in brower
ARTISTIC VISUALIZATION KALEIDOSCOPE with p5.js
data.csv in static folder
Artistic Output
data translated
Sentiment Analysis of Comments - no comments = 0
Sentiment Plot
Save as png in static folder
appends with score
COLLECT SURVEY DATA with Streamlit main.py
Get user input: Name Likert Q's Comments
wellbeing_survey.csv
Likert Results Plot
Save as png in static folder

Link to larger version:
https://miro.com/app/board/uXjVKu2uH6w=/?share_link_id=808725857912

There are two sections to this documentation:

1. **Collect survey data**
    a. **Streamlit/python**
2. **View survey results**
    a. **Flask/python**
    b. **P5.js**
    c. **HTML**

1. COLLECT SURVEY DATA with Streamlit main.py


Name

Your name

I felt happy.

- Strongly agree
- Agree
- [✓ Neither agree nor disagree]
- Disagree
- Strongly disagree

I felt engaged.

- Strongly agree
- Agree
- [✓ Neither agree nor disagree]
- Disagree
- Strongly disagree

I felt comfortable.

- Strongly agree
- Agree
- [✓ Neither agree nor disagree]
- Disagree
- Strongly disagree

I felt safe and secure.

- Strongly agree
- Agree
- [✓ Neither agree nor disagree]
- Disagree
- Strongly disagree

I enjoyed the company of other people.

- Strongly agree
- Agree
- [✓ Neither agree nor disagree]
- Disagree
- Strongly disagree

I talked to other people.

- Strongly agree
- Agree
- [✓ Neither agree nor disagree]
- Disagree
- Strongly disagree

Any additional comments?

open new terminal

% cd museum_wellbeing_survey

% streamlit run main.py

```
import streamlit as st import plot_likert import csv import os import numpy as np import
pandas as pd import matplotlib.pyplot as plt from transformers import pipeline import
warnings
```

Suppress FutureWarnings from plot_likert library
```
warnings.filterwarnings("ignore", category=FutureWarning, module="plot_likert")
```

Define file paths

```
file_path = "./data/wellbeing_survey.csv" transformed_file_path = "./static/data.csv"
output_folder = "./static/" likert_output_filename = "likert_plot.png"
sentiment_output_filename = "sentiment_analysis_plot.png"
```

Ensure the output folder exists

```
os.makedirs(output_folder, exist_ok=True)
```

Define the sentiment analysis pipeline

```
model_name = "distilbert-base-uncased-finetuned-sst-2-english" sentiment_pipeline =
pipeline("sentiment-analysis", model=model_name)
```

Initialization

```
if 'init_done' not in st.session_state: st.session_state.init_done = True
```

Define the questions

```
st.session_state.questions = [ "I felt happy.", "I felt engaged.", "I felt comfortable.", "I felt
safe and secure.", "I enjoyed the company of other people.", "I talked to other people.", ]
```

Define the header

st.session_state.header = ["Name"] + st.session_state.questions + ["Comments"]

Define the Likert scale

original_scale = plot_likert.scales.agree5 reversed_scale = original_scale[::-1]
st.session_state.scale = reversed_scale

Define file path

st.session_state.file_path = file_path

Check if the file exists

st.session_state.create_new_data_file = not os.path.exists(st.session_state.file_path)

try:

Load the CSV file

df = pd.read_csv(file_path, quoting=csv.QUOTE_NONNUMERIC, encoding='utf-8')

Clean any newline characters in data fields (if necessary)

df.replace({r'\r': ' ', r'\n': ' '}, regex=True, inplace=True)

Open the CSV file in append mode, creating it if it doesn't exist

st.session_state.fo = open(st.session_state.file_path, "a", newline='', encoding='utf-8')
st.session_state.writer = csv.writer(st.session_state.fo) if
st.session_state.create_new_data_file:

```
st.session_state.writer.writerow(st.session_state.header) except Exception as e:
st.error(f"An error occurred while opening {st.session_state.file_path}: {e}")
```

Inject custom CSS with st.markdown

```
st.markdown(""" <style> .stTextInput>div>div>input { width: 200px; height: 25px; border-
radius: 5px; } .stTextInput>div { width: 200px; } </style> """, unsafe_allow_html=True)
```

Display a label Name with a text input

```
st.text_input(label="Name", value="Your name", key="name")
```

Display the questions and the Likert scale

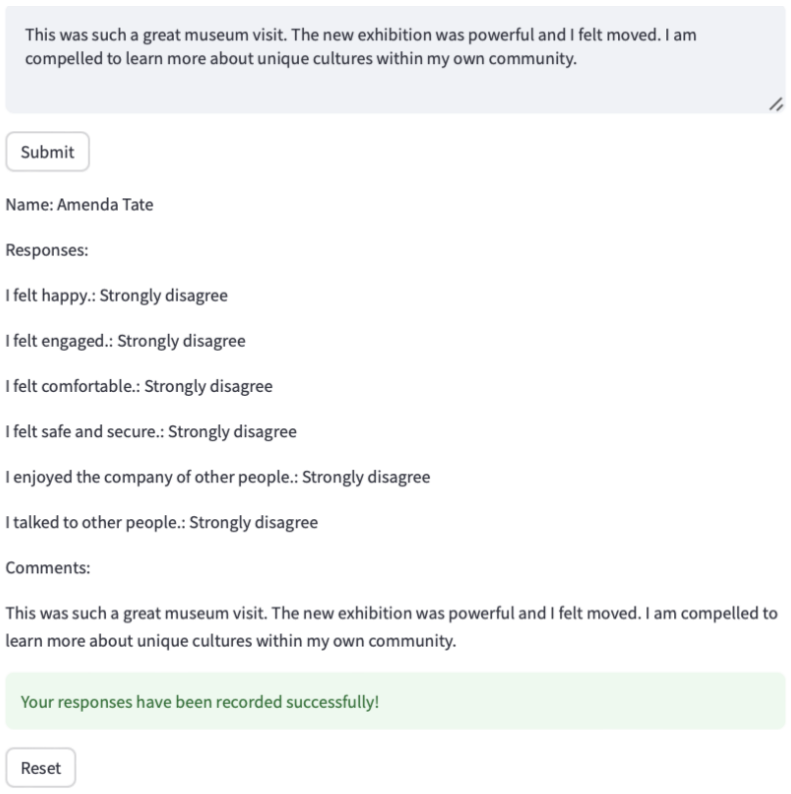
```
for question in st.session_state.questions: st.radio(question,
options=st.session_state.scale, index=2, key=question) # start at neutral
```

Display the text field for comments

```
st.text_area("Any additional comments?", key="comments")
```

When the submit button is pressed, print and record the responses and comments

```
if st.button('Submit'): st.write(f"Name: {st.session_state.name}") st.write("Responses:") for
question in st.session_state.questions: st.write(f"{question}: {st.session_state[question]}")
st.write("Comments:") st.write(st.session_state.comments)
```


This was such a great museum visit. The new exhibition was powerful and I felt moved. I am compelled to learn more about unique cultures within my own community.

Name: Amenda Tate

Responses:

I felt happy.: Strongly disagree

I felt engaged.: Strongly disagree

I felt comfortable.: Strongly disagree

I felt safe and secure.: Strongly disagree

I enjoyed the company of other people.: Strongly disagree

I talked to other people.: Strongly disagree

Comments:

This was such a great museum visit. The new exhibition was powerful and I felt moved. I am compelled to learn more about unique cultures within my own community.

Your responses have been recorded successfully!

Perform sentiment analysis on the comments

comment = st.session_state.comments if not comment: sentiment_score = 0 else: result = sentiment_pipeline(comment) sentiment_score = result[0]['score'] if result[0]['label'] == 'POSITIVE' else -result[0]['score']

Save the responses and sentiment score

record = [st.session_state.name] + [st.session_state[question] for question in st.session_state.questions] + [st.session_state.comments, sentiment_score] try: st.session_state.writer.writerow(record) st.session_state.fo.flush() st.success("Your responses have been recorded successfully!") except Exception as e: st.error(f"An error occurred while writing to {st.session_state.file_path}: {e}")

Reset form fields

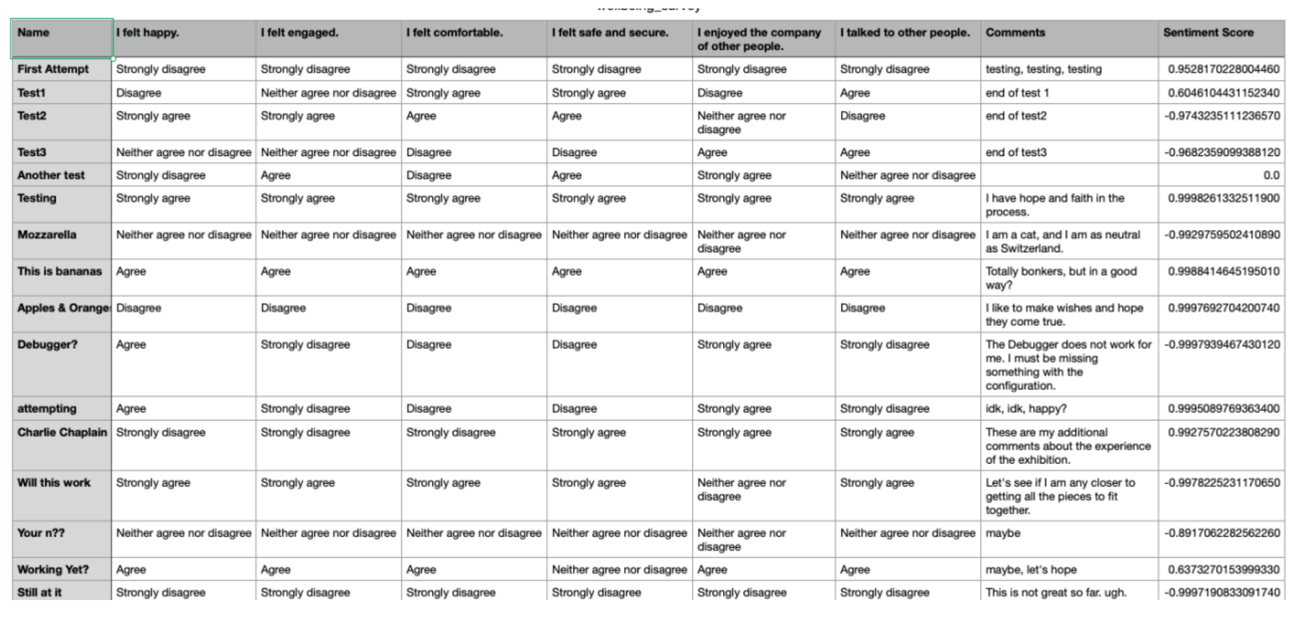
def clear_all(): st.session_state.pop('init_done') st.session_state["name"] = "Your Name" for question in st.session_state.questions: st.session_state[question] = st.session_state.scale[2] st.session_state["comments"] = "" st.button("Reset", on_click=clear_all)

Read and analyze the CSV file

try: df = pd.read_csv(file_path) if 'Sentiment Score' not in df.columns: df['Sentiment Score'] = None new_sentiment_scores = [] comments = [] original_texts = []

for index, row in df.iterrows(): if pd.isna(row['Sentiment Score']): if pd.isna(row['Comments']) or row['Comments'] == "": score = 0 else: data = row['Comments'] results = sentiment_pipeline(data) score = results[0]['score'] if results[0]['label'] == 'POSITIVE' else -results[0]['score'] df.at[index, 'Sentiment Score'] = score new_sentiment_scores.append(score) comments.append(row['Comments']) original_texts.append(data) else: score = row['Sentiment Score'] new_sentiment_scores.append(score) comments.append(row['Comments']) original_texts.append(row['Comments'])

df.to_csv(file_path, index=False) all_scores = df['Sentiment Score'].tolist() average_score = sum(all_scores) / len(all_scores) if all_scores else 0 df_sentiment = pd.DataFrame({'Sentiment Score': all_scores, 'Comment': comments, 'Text': original_texts})



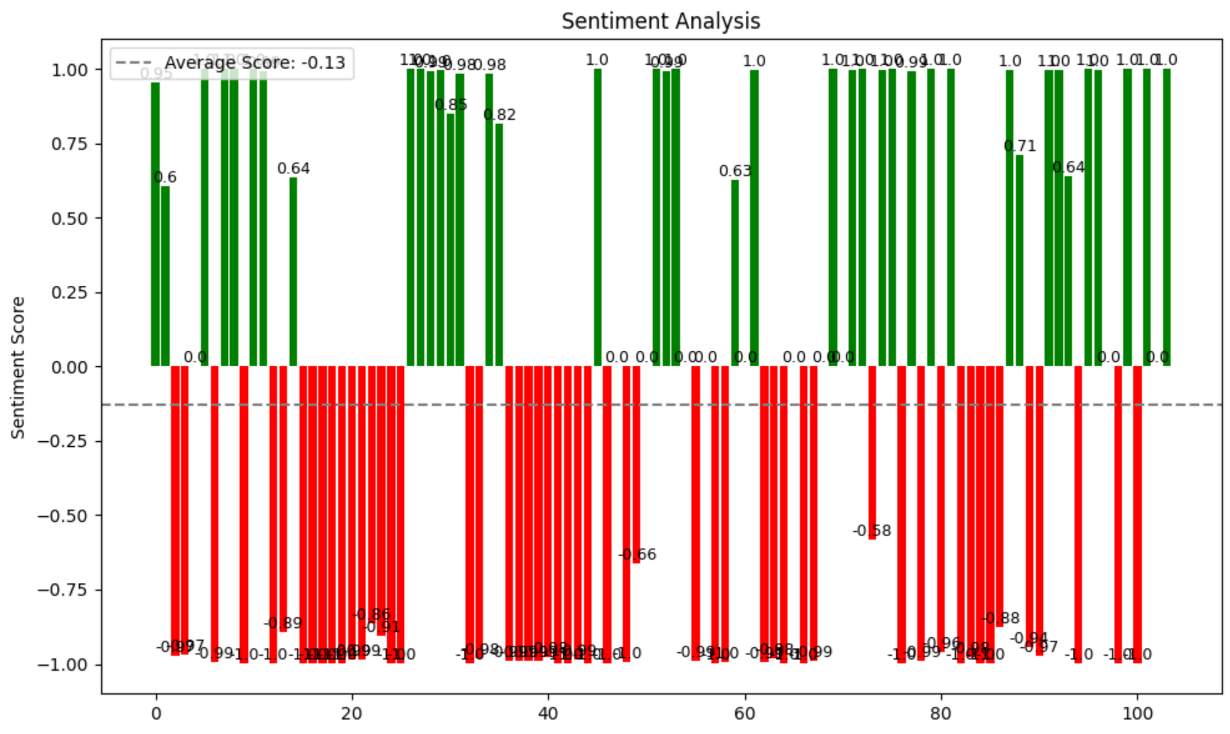
Name	I felt happy.	I felt engaged.	I felt comfortable.	I felt safe and secure.	I enjoyed the company of other people.	I talked to other people.	Comments	Sentiment Score
First Attempt	Strongly disagree	Strongly disagree	Strongly disagree	Strongly disagree	Strongly disagree	Strongly disagree	testing, testing, testing	0.9528170228004460
Test1	Disagree	Neither agree nor disagree	Strongly agree	Strongly agree	Disagree	Agree	end of test 1	0.6046104431152340
Test2	Strongly agree	Strongly agree	Agree	Agree	Neither agree nor disagree	Disagree	end of test2	-0.9743235111236570
Test3	Neither agree nor disagree	Neither agree nor disagree	Disagree	Disagree	Agree	Agree	end of test3	-0.9682359099388120
Another test	Strongly disagree	Agree	Disagree	Agree	Strongly agree	Neither agree nor disagree		0.0
Testing	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Strongly agree	I have hope and faith in the process.	0.9998261332511900
Mozzarella	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	I am a cat, and I am as neutral as Switzerland.	-0.9929759502410890
This is bananas	Agree	Agree	Agree	Agree	Agree	Agree	Totally bonkers, but in a good way?	0.9988414645195010
Apples & Orange	Disagree	Disagree	Disagree	Disagree	Disagree	Disagree	I like to make wishes and hope they come true.	0.9997692704200740
Debugger?	Agree	Strongly disagree	Disagree	Disagree	Strongly agree	Strongly disagree	The Debugger does not work for me. I must be missing something with the configuration.	-0.9997939467430120
attempting	Agree	Strongly disagree	Disagree	Disagree	Strongly agree	Strongly disagree	idk, idk, happy?	0.9995089769363400
Charlie Chaplain	Strongly disagree	Strongly disagree	Strongly disagree	Strongly agree	Strongly agree	Strongly agree	These are my additional comments about the experience of the exhibition.	0.9927570223808290
Will this work	Strongly agree	Strongly agree	Strongly agree	Strongly agree	Neither agree nor disagree	Strongly agree	Let's see if I am any closer to getting all the pieces to fit together.	-0.9978225231170650
Your n??	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	maybe	-0.8917062282562260
Working Yet?	Agree	Agree	Agree	Neither agree nor disagree	Agree	Agree	maybe, let's hope	0.6373270153999330
Still at it	Strongly disagree	Strongly disagree	Strongly disagree	Strongly disagree	Strongly disagree	Strongly disagree	This is not great so far. ugh.	-0.9997190833091740

Create sentiment analysis plot

colors = ['green' if score >= 0 else 'red' for score in df_sentiment['Sentiment Score']] plt.figure(figsize=(10, 6)) bars = plt.bar(df_sentiment.index, df_sentiment['Sentiment Score'], color=colors) plt.title('Sentiment Analysis') plt.ylabel('Sentiment Score') for bar, score in zip(bars, df_sentiment['Sentiment Score']): plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), f'{round(score, 2)}', ha='center', va='bottom', fontsize=9) plt.axhline(y=average_score, color='gray', linestyle='--', label=f'Average Score: {round(average_score, 2)}') plt.legend() plt.tight_layout()

Save the plot as PNG file

plt.savefig(os.path.join(output_folder, sentiment_output_filename)) plt.close() # Close the figure to free up memory



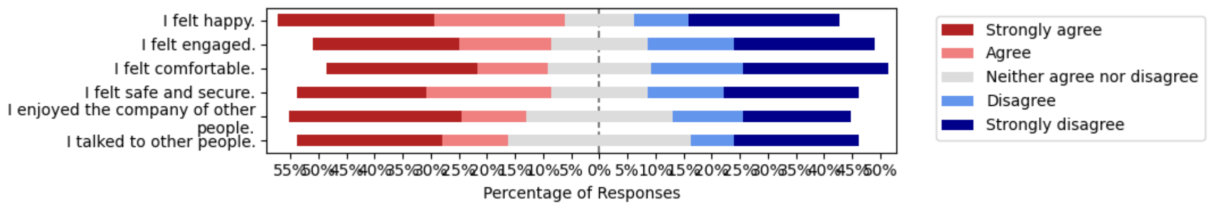
Create Likert scale plot

questions = st.session_state.questions likert_data = df[questions] unique_responses = pd.unique(likert_data.values.ravel('K')) print("Unique responses found in the data:", unique_responses) myscale = ("Strongly agree", "Agree", "Neither agree nor disagree", "Disagree", "Strongly disagree") fig, ax = plt.subplots(figsize=(11, 2)) # Create a new figure and axes plot_likert.plot_likert(likert_data, myscale, plot_percentage=True, ax=ax) fig.tight_layout()

Save the Likert plot as PNG file

fig.savefig(os.path.join(output_folder, likert_output_filename)) plt.close(fig) # Close the figure to free up memory

except FileNotFoundError: st.error(f"Error: The file at {file_path} was not found.") except pd.errors.EmptyDataError: st.error("Error: The CSV file is empty.") except Exception as e: st.error(f"An error occurred: {e}") import pandas as pd



Define the mapping dictionaries for p5.js

response_mapping = { 'Strongly disagree': 100, 'Disagree': 200, 'Neither agree nor disagree': 300, 'Agree': 400, 'Strongly agree': 500 }

width_mapping = { 'Strongly disagree': 1, 'Disagree': 2, 'Neither agree nor disagree': 3, 'Agree': 4, 'Strongly agree': 5 }

noise_mapping = { 'Strongly disagree': 0.0, 'Disagree': 0.25, 'Neither agree nor disagree': 0.5, 'Agree': 0.75, 'Strongly agree': 1.0 }

Function to map sentiment score to 0-359

def map_sentiment_to_color(value): return abs(value) * 359

Function to create the '+ or -' column

def create_plus_minus(value): return 100 if value > 0 else 70

Load the CSV file

```
df = pd.read_csv(file_path)
```

Rename columns

```
df = df.rename(columns={ 'I felt happy.': 'x1', 'I felt engaged.': 'x2', 'I felt comfortable.': 'y1', 'I
felt safe and secure.': 'y2', 'I enjoyed the company of other people.': 'width', 'I talked to
other people.': 'noise' })
```

Print columns to debug

```
print("Columns after renaming:", df.columns)
```

Map responses to numeric values

```
df['x1'] = df['x1'].map(response_mapping) df['x2'] = df['x2'].map(response_mapping) df['y1']
= df['y1'].map(response_mapping) df['y2'] = df['y2'].map(response_mapping) df['width'] =
df['width'].map(width_mapping) df['noise'] = df['noise'].map(noise_mapping)
```

Add the "color" column by mapping sentiment score to color values (0-359)

```
df['color'] = df['Sentiment Score'].apply(map_sentiment_to_color)
```

Create the '+ or -' column

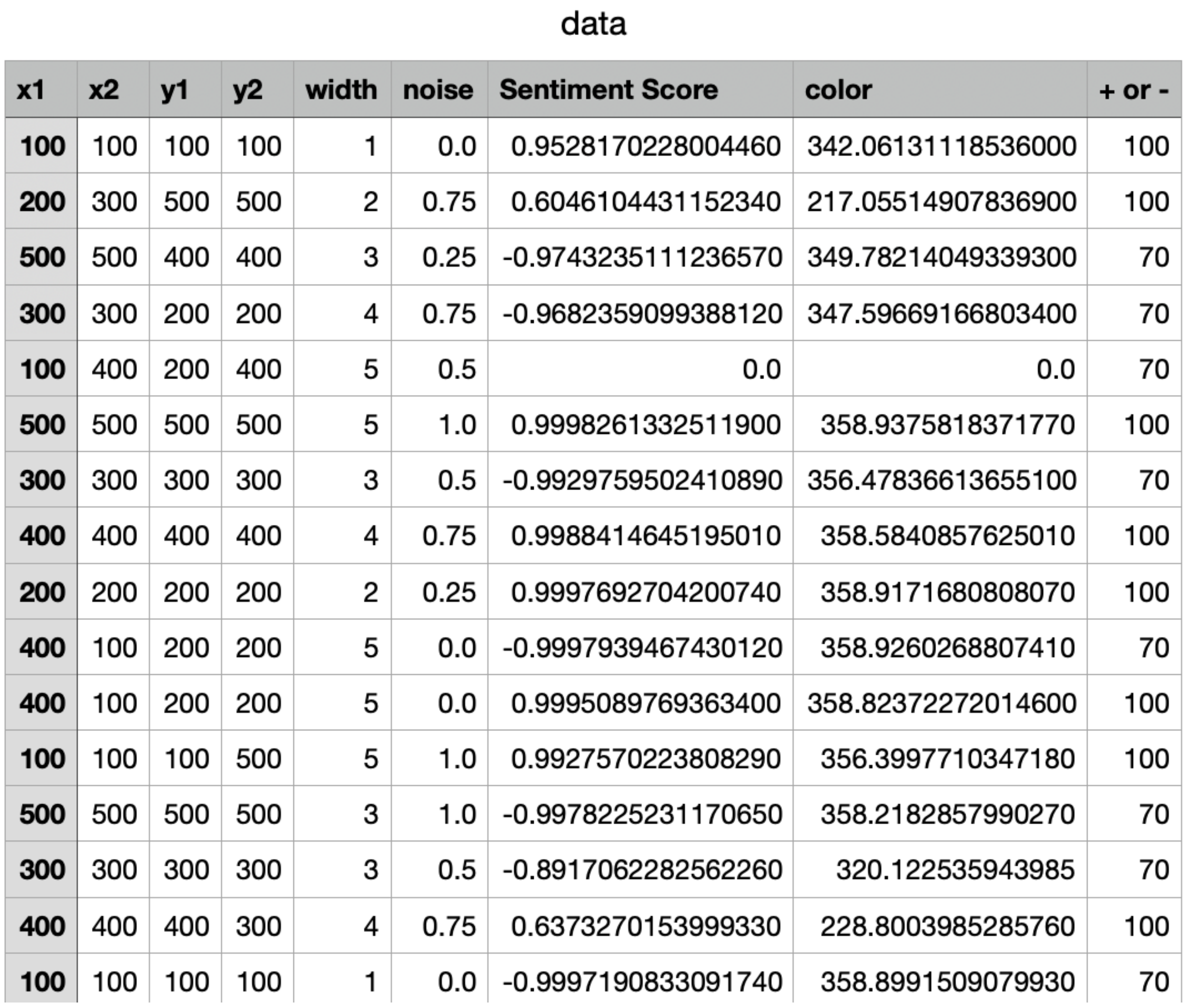
```
df['+ or -'] = df['Sentiment Score'].apply(create_plus_minus)
```

Drop the Name and Comments columns

```
df = df.drop(columns=['Name', 'Comments'])
```

Save the transformed DataFrame to a new CSV file

df.to_csv(os.path.join(transformed_file_path), index=False)



data

x1	x2	y1	y2	width	noise	Sentiment Score	color	+ or -
100	100	100	100	1	0.0	0.9528170228004460	342.06131118536000	100
200	300	500	500	2	0.75	0.6046104431152340	217.05514907836900	100
500	500	400	400	3	0.25	-0.9743235111236570	349.78214049339300	70
300	300	200	200	4	0.75	-0.9682359099388120	347.59669166803400	70
100	400	200	400	5	0.5	0.0	0.0	70
500	500	500	500	5	1.0	0.9998261332511900	358.9375818371770	100
300	300	300	300	3	0.5	-0.9929759502410890	356.47836613655100	70
400	400	400	400	4	0.75	0.9988414645195010	358.5840857625010	100
200	200	200	200	2	0.25	0.9997692704200740	358.9171680808070	100
400	100	200	200	5	0.0	-0.9997939467430120	358.9260268807410	70
400	100	200	200	5	0.0	0.9995089769363400	358.82372272014600	100
100	100	100	500	5	1.0	0.9927570223808290	356.3997710347180	100
500	500	500	500	3	1.0	-0.9978225231170650	358.2182857990270	70
300	300	300	300	3	0.5	-0.8917062282562260	320.122535943985	70
400	400	400	300	4	0.75	0.6373270153999330	228.8003985285760	100
100	100	100	100	1	0.0	-0.9997190833091740	358.8991509079930	70

2. DISPLAYING SURVEY RESULTS with Flask app.py

a. open new terminal

cd museum_wellbeing_survey

museum_wellbeing_survey % python app.py

open http://127.0.0.1:8001/

if it says it is already in use

lsof -ti :8001

kill -9 PID

```
from flask import Flask, render_template, send_from_directory

app = Flask(name)

@app.route('/')

def index():

return render_template('index.html')

if name == 'main':

app.run(debug=True, port=8001)
```

---

b. p5.js documentation

This p5.js sketch draws symmetric lines based on data from a CSV file. It supports adjusting symmetry, line appearance, and noise effects.

Key Variables

let symmetry = 7;
Number of symmetry sections (change to 6 if needed).

let angle = 360 / symmetry;
Angle for each symmetry section.

let table;
Stores the loaded CSV data.

let currentRow = 0;
Keeps track of the current row in the CSV file.

let csvUrl = '/static/data.csv';
URL to the CSV file.

Functions preload()
Loads the CSV file from the static folder.

---

c. javascript HTML documentation for Flask index

This HTML document is designed to display the results of a museum wellbeing survey, including plots generated from the survey data and an artistic rendering using p5.js. It leverages p5.js for visualization and incorporates CSS for styling. Structure HTML

Declaration and Metadata

<!DOCTYPE html>: Defines the document type and version (HTML5).
<html lang="en">: Specifies the language of the document as English.
<meta charset="utf-8" />: Sets the character encoding to UTF-8.
<meta name="viewport" content="width=device-width, initial-scale=1.0">:

External Libraries

p5.js: JavaScript library for creative coding.
p5.sound: Adds sound functionality to p5.js sketches.

CSS Styling

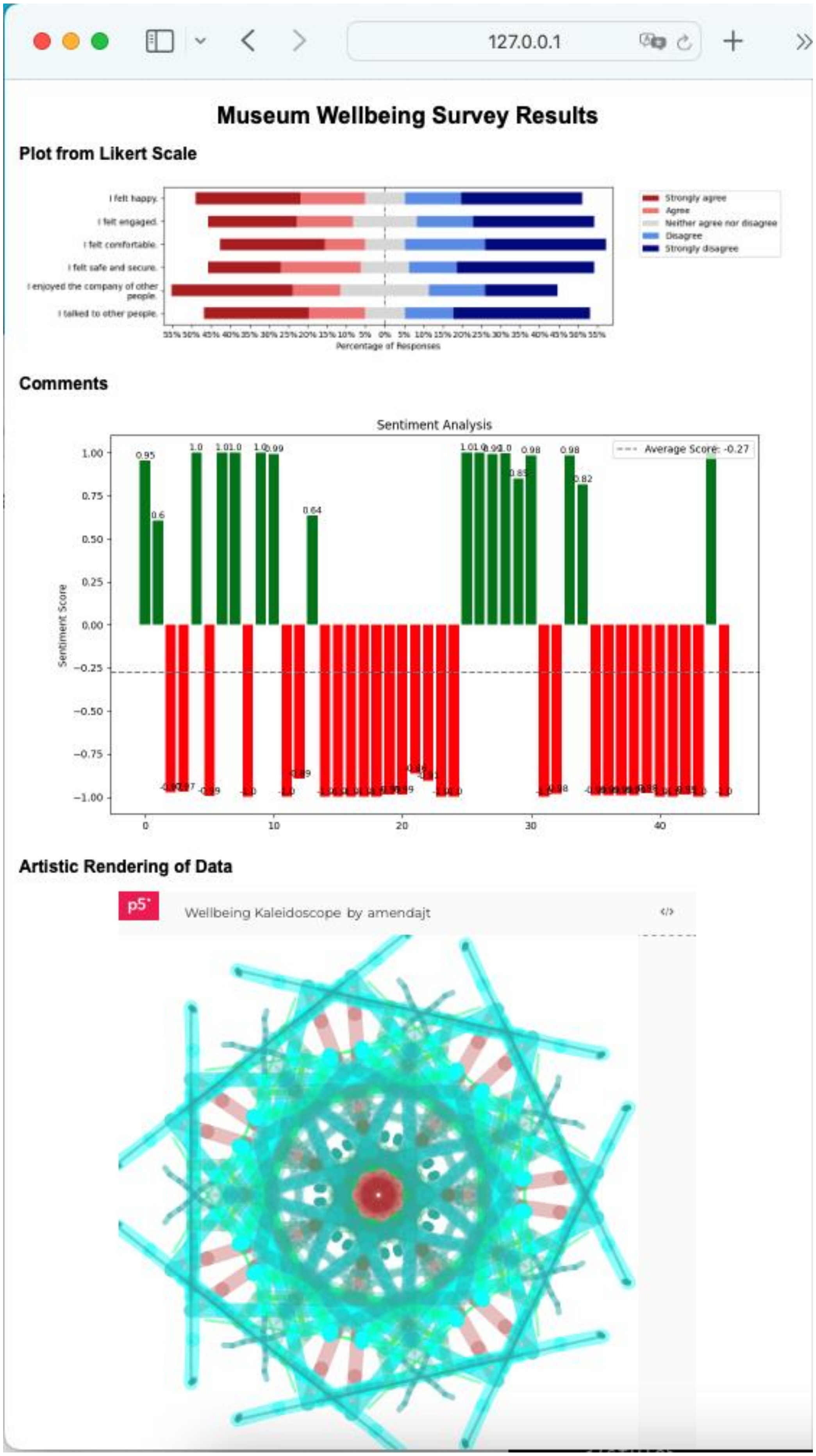
Body: Sets font, margins, and padding for the page.
Headings (h1, h2): Center-aligns the main title and left-aligns subheadings.
Image Container: Flexbox layout for image alignment with spacing.
Images: Responsive design ensuring images maintain aspect ratio and fit within their containers.
Canvas Container: Centers the p5.js canvas and sets its dimensions.
iFrame: Styles the iframe to match the canvas size and removes default border.

Body Content

Main Title (h1): Displays the primary title of the page.
Likert Scale Plot (h2): Shows the plot generated from the Likert scale data.
Comments Plot (h2): Displays the sentiment analysis plot.
Artistic Rendering (h2): Embeds a p5.js sketch using an iframe.

Usage

Images: Ensure likert_plot.png and sentiment_analysis_plot.png are located in the static folder of your Flask application.
p5.js Sketch: Update the iframe src attribute if you have a different p5.js sketch URL.
Customization: Adjust CSS styles and HTML structure as needed to fit your design requirements.



127.0.0.1

Museum Wellbeing Survey Results

Plot from Likert Scale

Comments

Artistic Rendering of Data

Future work

Considerations for improvement or enhancement of the basic application as outlined above:

- update so “blank” submissions do not get recorded
- version that will run on a remote server
- mobile-friendly design
- improved data analysis and report options
- interactive dashboard for analysis and reports
- accessibility considerations
- option to capture user’s email address for mailing list
- data backup and export options
- integration to notify of new survey entries
- optimize for handling larger volumes of data