# Subjective Wellbeing Platform: User Manual

This platform helps you explore trends in subjective wellbeing indicators like health, community connection, and safety. It has three main features:

1. **Dashboard**: Visualize trends and future predictions.
2. **Interactive Map**: Explore geographic patterns.
3. **Data Viewer**: View and filter raw data.

**Steps in this Manual:**

- How to set up the platform.
- How to use the Dashboard.
- How to use the Map.
- How to use the Data Viewer.

**Note**: This platform runs on Flask and Dash, integrated with Google Colab. Follow the instructions step-by-step.

## Step 1: Install Required Libraries

In [ ]:
```
!pip install flask dash pandas folium plotly dash-bootstrap-components
```

## Step 2: Import Required Libraries

In [ ]:
```
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
import dash
from dash import dcc, html, Input, Output
import dash_bootstrap_components as dbc
import folium
from folium.plugins import HeatMap, MarkerCluster
import plotly.graph_objs as go
```

## Step 3: Start the Flask App

Run the Flask app to host the platform locally. This app will allow you to:

1. View the **Dashboard** for trends.
2. Use the **Interactive Map** to explore geographic data.
3. Use the **Data Viewer** to browse raw data.

The following cell starts the server. Copy the URL provided in the output (e.g., `http://127.0.0.1:2022/`) and open it in your browser.

In [ ]:
```
from flask import Flask, render_template, request
import threading

# Flask application setup
app = Flask(__name__)

@app.route('/')
def home():
    return '''
    <h1>Welcome to the Subjective Wellbeing Platform</h1>
    <p>Explore the following:</p>
    <ul>
        <li><a href="/dashboard">Dashboard</a></li>
        <li><a href="/map">Interactive Map</a></li>
        <li><a href="/data">Data Viewer</a></li>
    </ul>
    '''

def run_app():
    app.run(port=2022)

# Run Flask app in a thread
thread = threading.Thread(target=run_app)
thread.start()
```

# Step 4: Using the Dashboard

The dashboard allows you to:

1. Filter data by subtopics, age groups, genders, and suburbs.
2. View trends from 2018–2023.
3. See predictions for 2024–2025.

## Instructions:

1. Open the `/dashboard` link provided by the Flask app (e.g., `http://127.0.0.1:2022/dashboard`).
2. Use the dropdown menus to select filters:
   - **Subtopic**: Choose a wellbeing indicator like Personal Health.
   - **Age Group**: Filter by age group (e.g., 18–24).
   - **Gender**: Filter by Male or Female.
   - **Suburb**: Narrow the results to specific locations.
3. The graph updates automatically based on your filters.
4. To reset filters, refresh the page or select "All" from the dropdowns.

# Step 5: Using the Interactive Map

The interactive map displays geographic patterns in subjective wellbeing data.

## Instructions:

1. Open the `/map` link provided by the Flask app (e.g., `http://127.0.0.1:2022/map`).
2. Use the dropdown filters at the top of the page:
   - **Subtopic**: Select a wellbeing indicator.
   - **Year**: Choose a year or select "All" for all years.
3. Features:
   - **Markers**: Click on markers to view location-specific details.
   - **Heatmap**: Highlights regions with higher or lower wellbeing percentages.
4. Click "Update Map" to apply filters.
5. To reset, select "All" in the filters or refresh the page.

# Step 6: Using the Data Viewer

The Data Viewer allows you to explore the raw dataset and apply filters for specific insights.

## Instructions:

1. Open the `/data` link provided by the Flask app (e.g., `http://127.0.0.1:2022/data`).
2. Use the dropdown filters to narrow down results:
   - **Subtopic**: Filter by wellbeing indicators.
   - **Category**: Filter by Age Group, Gender, or Suburb.
   - **Year**: Choose a specific year.
3. View the filtered data in the table below the filters.
4. Export or copy data directly from the table (if supported).

# Step 7: Troubleshooting

## Common Issues:

1. **Page Not Loading**:
   - Ensure the Flask server is running.
   - Restart the app by running the `run_app()` cell again.
2. **Filters Not Working**:
   - Refresh the page and reapply your filters.
   - Ensure your dropdown selections are valid for the dataset.
3. **Slow Loading**:
   - Large datasets may take a few seconds to render. Be patient.
   - Optimize your data processing scripts if needed.

4. **Data Errors**:
    - Ensure your data files are correctly formatted and placed in the correct directory.