|  |
| --- |
| Project Portfolio  Mai Nguyen |
| Digital technology |
|  |
| Mai Nguyen | xxx | September 9, 2023 |

Table of content

[1. Problem 1](#_Toc165315814)

[2. solutions 1](#_Toc165315815)

[3. Pre-production 1](#_Toc165315816)

[3.1 Development tool selection: 1](#_Toc165315817)

Introduction

# 1. Problem

New Zealand’s universities are facing an identity crisis”

# 2. solutions

A dashboard is a type of graphical user interface which often provides views of key performance indicators (KPIs) that are relevant to a particular objective or data-driven process dashboards aggregate data from multiple databases and files such as Excel spreadsheets. Unlike excel, a dashboard allows you to analyse your data through combination of tables, graphs and interactive user input. The purposes of Dash projects are to show change in data over time to offer insight, trends predictions or inefficiencies and observe relationships between different variables.

# 3. Pre-production

## 3.1 Development tool selection:

I have chosen “Plotly Dash” to develop my data visualisation project. “Plotly” is a Python library which offers multiple data visualisation models such as different plots, charts, and graphs. A library is a collection of pre-written code that makes coding quicker and more efficient, by reducing the amount of code a programmer needs to write.[[1]](#footnote-1) Plotly integrates Python’s leading data visualization and UI (user interface) libraries, making it suitable for a data visualization project.

“Dash” is an open-source library that allows you to create analytics dashboards or data apps - the interface that Plotly’s data visualisation will be displayed. This is developed by Plotly developers to easily implement a web interface and create dashboards without having to learn JavaScript, HTML and other web technologies.[[2]](#footnote-2)

**Business viewpoint:** When combined, Plotly Dash is a platform that offers a high-potential tool to leverage Python in creating data visualisation dashboards. Plotly Dash’s structure is “low-code” because of its low requirement for manual coding, making it well-suited for a beginner to deploy and iterate a functional data application.[[3]](#footnote-3) This “low-code” structure allows flexible customization, quicker development, and scalability to handle growth of a start-up business. Plotly Dash allow users, through the “low-code” structure, to create data apps for their projects and businesses. Data app is the cross-over between web-apps and traditional dashboards, making it easier for decision-makers to understand their data in a corporate environment or for research purposes. This will bring value to end-user of my project. These apps are suited for non-technical users (those that are not data scientists or analysts) to make data-driven decisions, showing that it could resonate with most audience.

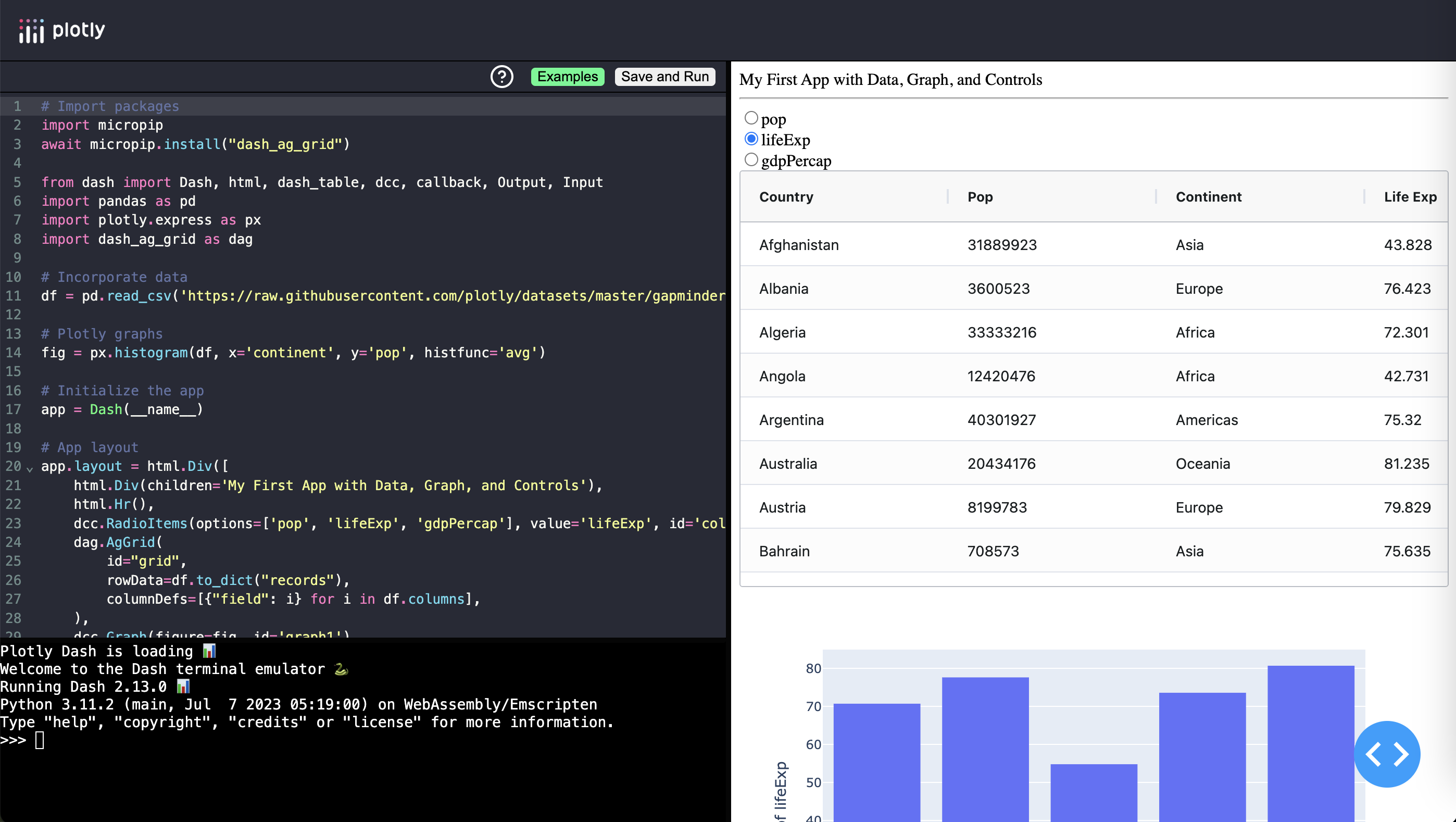
**Technical viewpoint:** Plotly Dash offers the “Dash Open Source Framework” that supports open-source libraries for Python’s leading data visualization and UI (user interface) libraries. Plotly for Python is a free and open-source software, which means the original code is available for anyone to redistribute and modify. Also, Plotly Dash is free to install and use. The commercial, more advanced version of Plotly Dash is Dash Enterprise.[[4]](#footnote-4) An opensource platform largely benefits an industry beginner because it offers opportunities for customization, learning, extend skills and knowledge. Any beginner can reference the open-source code and documentation to make their own app while learning new coding knowledge and practices, that sharpens their skills in the longer term. In contrast, commercial-based platforms such as Tableau, is not beneficial to industry beginners by limiting customization and research capabilities.

To develop my project, I will learn through the Plotly Dash Course from the Plotly YouTube Channel (<https://www.youtube.com/@Plotly>) published on March 2024, including 6 sessions that guide developers through development of their webapp and dashboard.

In the first session, I have discovered the main elements of a Plotly Dash code for a web-app.

The to-be-developed app is recommended to be developed locally, on Visual Studio Code, on my computer because it contains large amounts of data and different files and folders.

The course offers a proof of concept of what a simple data visualization dashboard would look like in session 1 of the course. This proof-of-concept runs on the browser through WasmDash (allowing apps to be developed on the browser). The proof-of-concept connects to the tutorial “Dash in 20 minutes” and is named “dash\_in\_20.py” on WasmDash. I will look through this example to understand more about Plotly Dash code then find the components that I could reference.



Library

Dataset

App layout/component

Callback

Importing library

The first element is the libraries, which are collection of pre-written code that developers can use to perform a specific task. I will be using pandas and plotly express, AG grid (for the proof-of-concept’s data table) For example, the “plotly.express” library offers most common charts and plots such as scatter plot, 1D/2D distributions.

To install libraries, I have to install it using **pip install [library]** as I work on my local computer.

Incorporate data

In this section, I will include the dataset that my data visualization will demonstrate. Data can come from external data databases, APIs, text files, JS files,… If my data is in a CSV sheet stored locally, it must be on the same folder as the code,

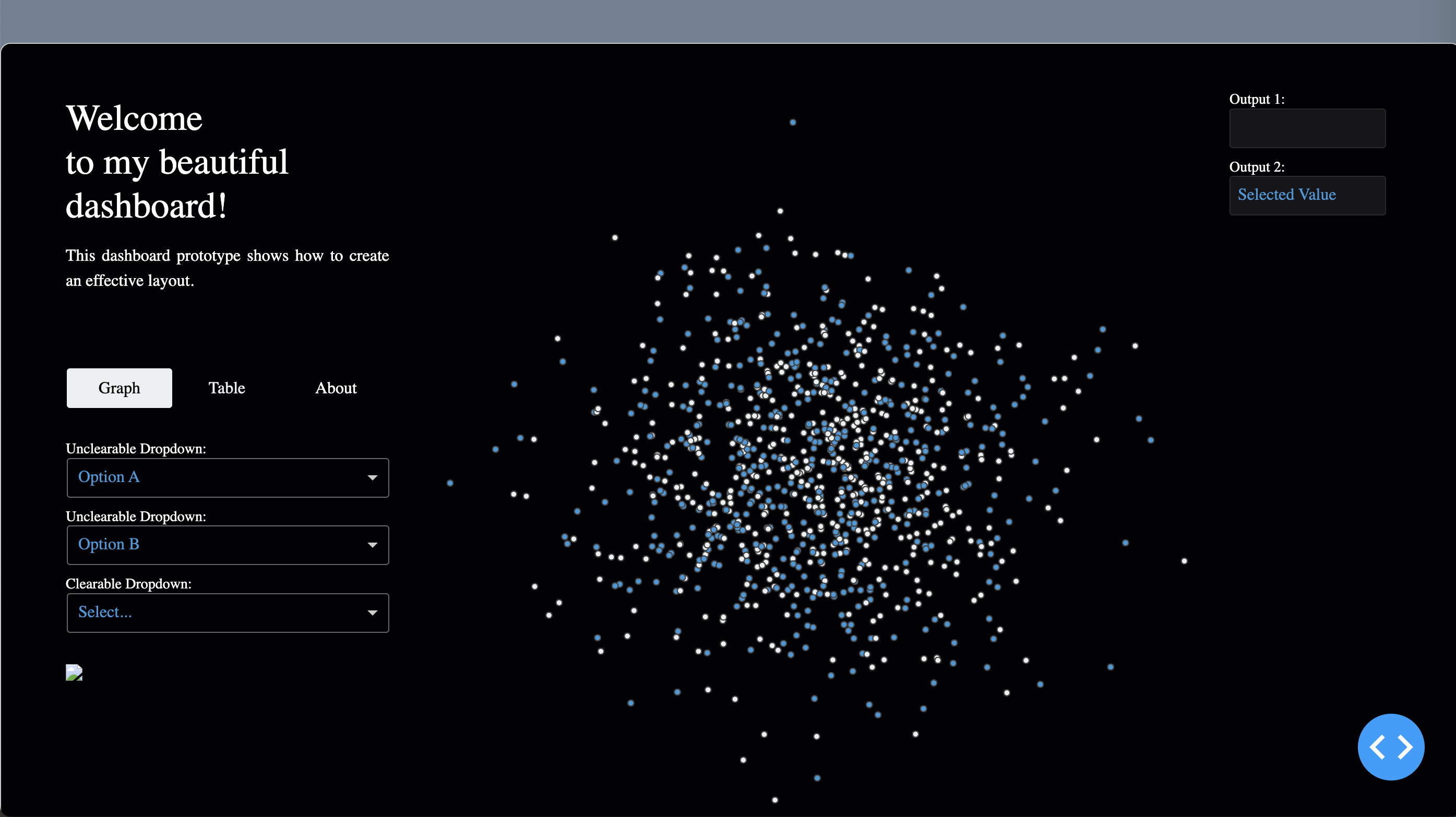
The next component is the layout. This is what allows displaying anything I want on the page. My users will use my app and they will see whatever I want them to see. For example, if I want them to see radio buttons then my app will include a table through AG grid and a histogram. I have to define these in this component for them to show up on my dashboard.

The last component is the callback. This component allows developers to connect between components inside the layout. It’s what gives life to the dashboard, by allowing interactivity. For example, in the proof of concept, the callback is used to connect between the radio button and the histogram. This is probably the most important component of a Dash app. The open-source code to multiple Dash component allows me to include any components that I want such as checklists or clipboards.

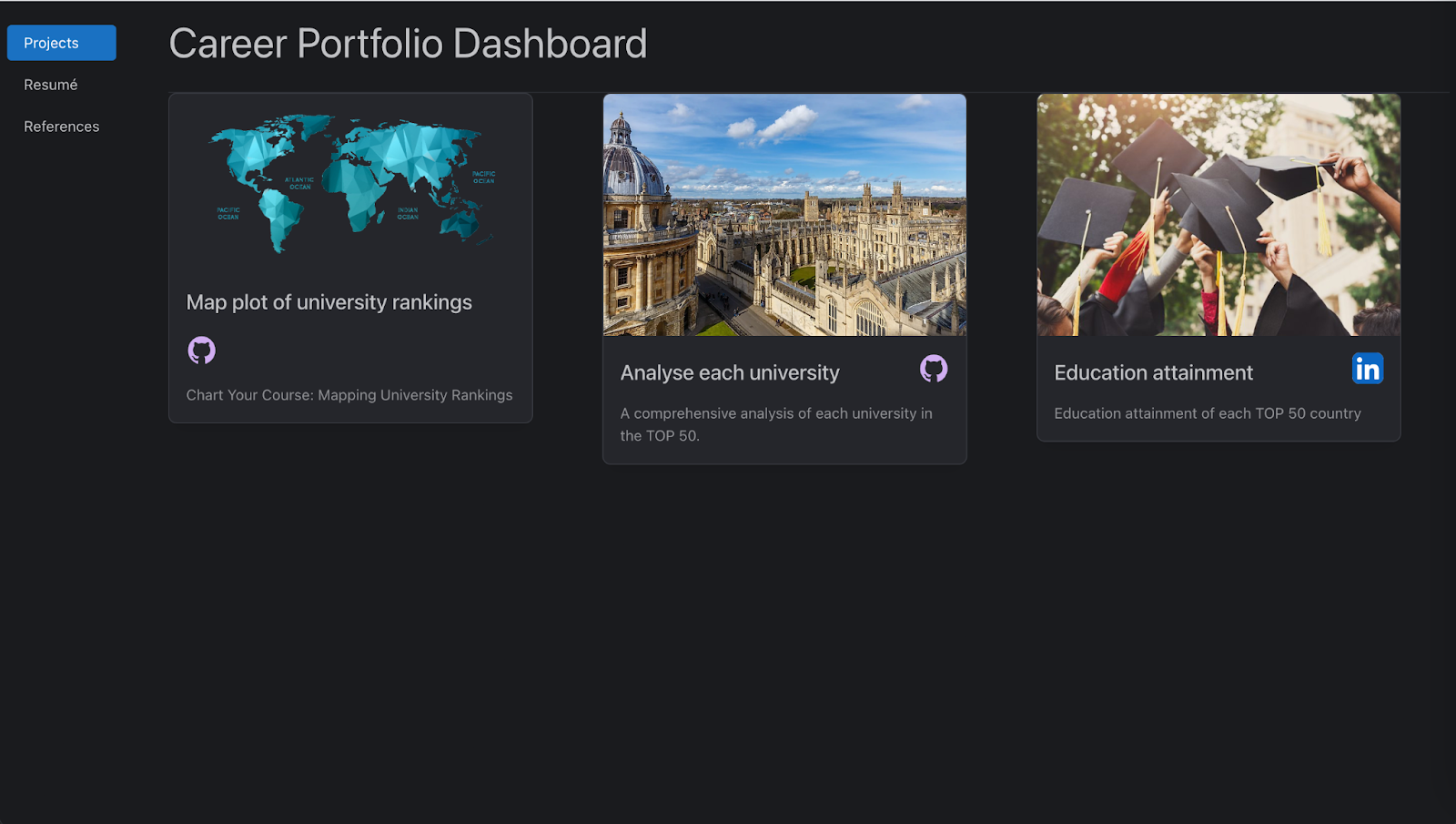
- Plotly Dash course

- IMDB dashboard

- Dashboard layout



- Dashboard/digital-technology (dashboard (Render))



**There are 5 elements of a plotly dash code which is:**

* Library
* Dataset
* App layout
* Components
* Callback

Nice colour theme:

<https://datavizproject.com/>

1. Woke, G. (2023). The difference between libraries and Frameworks. Retrieved from <https://www.red-gate.com/simple-talk/development/other-development/the-difference-between-libraries-and-frameworks/#:~:text=A%20library%20is%20a%20collection,be%20called%20upon%20as%20needed>.

   [↑](#footnote-ref-1)
2. Jack022Jack022                  1, vlizanavlizana                      3, & ruslanivruslaniv                      53011 gold badge66 silver badges1717 bronze badges. (1964). What’s the difference between dash and plotly? Retrieved from <https://stackoverflow.com/questions/53146357/whats-the-difference-between-dash-and-plotly>

   [↑](#footnote-ref-2)
3. Plotly. (n.d.). What’s better: Low-code or no-code data app development? Retrieved from <https://plotly.com/blog/low-code-vs-no-code-data-app-development/> [↑](#footnote-ref-3)
4. Is Plotly for Python Free? (n.d.). Retrieved from <https://plotly.com/python/is-plotly-free/> [↑](#footnote-ref-4)