Coding Challenge

The goal of this coding challenge is to get an impression of how you handle the basics of the technology stack you are likely to encounter. Our stack consists of a TypeScript frontend (Vue.js/React.js) with a Python backend (Django). To streamline the evaluation process, we've simplified the stack for this challenge (i.e., we'll use Flask instead of Django) while retaining the same programming languages. This adjustment eliminates the need for extensive setup, allowing you to focus on showcasing your skills in a more accessible environment. However, feel free to use Vue.js, React.js, or Diango if you prefer.

Project Introduction

This project has three primary objectives:

1) **Build a Simple API:**
   Create a Flask (or Django if you prefer) server to serve player information through REST API endpoints. Define routes for retrieving individual player details, a list of all players, countries with their players, clubs with their players, and a list of all attribute names.
2) **Construct a Table:**
   Develop a frontend using D3 (or Vue.js/React.js if you prefer) that renders a plain HTML table showcasing essential soccer player details. Enable sorting functionality when clicking on column headers. Implement a dropdown for attribute manipulation, allowing users to add or remove columns.
3) **Implement Linked Visualizations:**
   Integrate visualizations to the right of the table to represent skill distributions. Choose a visualization technique (histograms, dot plots, or violin plots) and display distributions for selected attributes from the dropdown. Implement interaction by highlighting selected rows in the table and visualizing the skills of the chosen player in the distributions. Display the player's name on top of the visualizations.

Project Setup

First, please set up a public GitHub repo. This is where your code will live and will allow us to take a look at your code most easily. We'll be taking a monorepo approach with this project, so your frontend and backend code will live in the same repository. Please add a subfolder in the repo for each.

As discussed above, we suggest using Python + Flask and D3 for this coding exercise. Make sure you have Python and Flask installed locally on your machine before proceeding if you choose this stack. If you decide to go with another tool, make sure they're installed.

Once you have the dependencies installed, please set up your frontend and backend tools in their respective folders. We're not being too prescriptive here, so please choose a method for setting up the tools that is most intuitive to you.

Ensure to include a README file in the root of your repository with instructions on how to run your project locally.

Finally, please make sure that your implementation follows good software engineering practices and that your source code is documented.

Data Source

You are given a file that contains a number of soccer players as an array of JSON objects. Your initial objective is to develop a backend that reads this JSON file and outputs data in specific formats, as outlined below. This backend will serve as the data source for your frontend.

Backend

Write a Flask (or Django if you prefer) server which serves the file above from a REST API with the following support:

- /players/ - returns all players and their attributes
- /players/{name} - returns a player and all the player attributes
- /clubs/ - returns all clubs with a list of players playing for those clubs
- /attributes/ - returns a list of all attribute names

You will use some of these end-points in the frontend component. Please add a simple demonstration of example queries for all of these endpoints, independent of whether you use them in the application or not. We suggest you include links that return the JSON response, and a sample of what is returned.

Frontend (Table)

1) Using D3 (or Vue.js/React.js if you prefer), fetch data from the API to retrieve a list of players and their attributes. Render the data as a table with the following columns:
    - Name
    - Nationality
    - National Position
    - Club
    - Height
    - Preferred Foot
    - 2-3 other attributes of your choice

2) Near your table, include a dropdown menu featuring the remaining attributes. This dropdown should allow the addition of selected attributes as columns to the table.
3) Implement an intuitive mechanism for users to remove attributes directly from the table.
4) Implement logic that enables clicking on the table headers to sort the rows based on the selected attribute.This should work for all attributes in the table.
5) Feel free to add some css styling to your table

The plain HTML table will look like this:

Name	Nationality	National Position	Club	Height	Preffered_Foot
Alexis Sánchez	Chile	LW	Arsenal	169 cm	Right
Ángel Di María	Argentina	LW	PSG	180 cm	Left
Antoine Griezmann	France	CAM	Atlético Madrid	176 cm	Left
Arjen Robben	Netherlands	RW	FC Bayern	180 cm	Left
Arturo Vidal	Chile	CDM	FC Bayern	180 cm	Right
Bernd Leno	Germany	Sub	Bayer 04	190 cm	Right
Coutinho	Brazil	RW	Liverpool	171 cm	Right
Cristiano Ronaldo	Portugal	LS	Real Madrid	185 cm	Right
David Alaba	Austria	LM	FC Bayern	180 cm	Left
David Silva	Spain	LM	Manchester City	173 cm	Left
De Gea	Spain	GK	Manchester Utd	193 cm	Right
Diego Costa	Spain	ST	Chelsea	188 cm	Right
Diego Godín	Uruguay	LCB	Atlético Madrid	185 cm	Right
Eden Hazard	Belgium	LF	Chelsea	173 cm	Right
Gareth Bale	Wales	RS	Real Madrid	183 cm	Left
Gianluigi Buffon	Italy	GK	Juventus	192 cm	Right
Giorgio Chiellini	Italy	LCB	Juventus	187 cm	Left
Gonzalo Higuaín	Argentina	Sub	Juventus	184 cm	Right
Henrikh Mkhitaryan	Armenia		Manchester Utd	177 cm	Right

Frontend (Visualization)

Next, create visualizations for the distributions of skills among all players using 2 visualization techniques of your choice. You only need to consider the numerical and categorical variables (i.e. don't worry about visualizing the names).

Place these visualizations to the right of the table.

Finally, Introduce interaction by enabling users to click a row in the table to select it. Upon selection, highlight the chosen row and display the skills of the selected player directly within the previously created distributions.

If you've applied for the software developer position, you're now done. Please share your code with us and be ready to present it in our upcoming interview.