Please use TypeScript as your programming language and D3 as the charting library for the front-end and Python Flask for the server component.

Share your work through a public github repository that contains all the information for us to run and evaluate your code.

**Player Table**

You are [given a file](https://www.google.com/url?q=https://drive.google.com/file/d/18bYLJg1aY0MhSqhDdn9ic246ofOLn7op/view?usp%3Dsharing&sa=D&source=editors&ust=1665231657562588&usg=AOvVaw2828pydMHkcJgYuLNZ7eAb) that contains a number of soccer players as an array of JSON objects. Your task is to create a web app that lists up these soccer players in a plain HTML table.

**Server Component**

Write a Flask server which serves the file above from a REST API with the following support:

* /players/{name} - returns a player and all the player attributes
* /players/ -  returns all players and their attributes
* /countries/ - returns all countries with a list of players from those countries
* /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 client component. Please add a simple demonstration of example queries to your client 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.

**Client Component**

Access the API to retrieve the list of players and the corresponding attributes. and render the following columns:

* Name
* Nationality
* National Position
* Club
* Height
* Preferred Foot
* 2-3 other attributes of your choice

Also show a dropdown with a list of remaining attributes. The dropdown should allow the addition of the attribute as a column to the table.

Clicking on the column header should sort the table according to the attribute (should also work for the above attributes)

Also design a sensible option to allow removal of attributes from the table.

The plain HTML table will look like this:

Table

Description automatically generated with medium confidence

Style the table using CSS or (SCSS).

**Skills Distribution and Player Detail View**

Next, create visualizations for the distributions of skills among all players using a visualization technique of your choice (histograms, dot plots, or violin plots are examples) for the attributes selected from the dropdown. You only need to consider numerical and categorical variables.

Place these visualizations to the right of the table.

Finally, please add interaction: when a row is selected by clicking a table in the row, highlight that row and visualize the skills of the selected player directly in the distributions you created earlier. Also show the name of the player on top of the visualizations.

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