Coding Challenge - Full stack developer

For this vacancy, we are looking for Full Stack Developers who have experience or affinity with the data domain. To speed up and strengthen the data revolution we are searching for the best Development Engineers to join us.

We prepared a coding challenge for you to show your skills to us and understand what kind of challenges you will see if you join us. This assessment could take 3-5 hours.

We would like you to deliver a full stack solution to visualize a small graph data with the following data:

Sample graph data

```
{
    "data":[
        {
            "name":"A",
            "description":"This is a description of A",
            "parent":""
        },
        {
            "name":"B",
            "description":"This is a description of B",
            "parent":"A"
        },
        {
            "name":"C",
            "description":"This is a description of C",
            "parent":"A"
        },
        {
            "name":"D",
            "description":"This is a description of D",
            "parent":"A"
        },
        {
            "name":"B-1",
            "description":"This is a description of B-1",
            "parent":"B"
        },
        {
            "name":"B-2",
            "description":"This is a description of B-2",
            "parent":"B"
        },
        {
            "name":"B-3",
            "description":"This is a description of B-3",
            "parent":"B"
        }
    ]
}
```
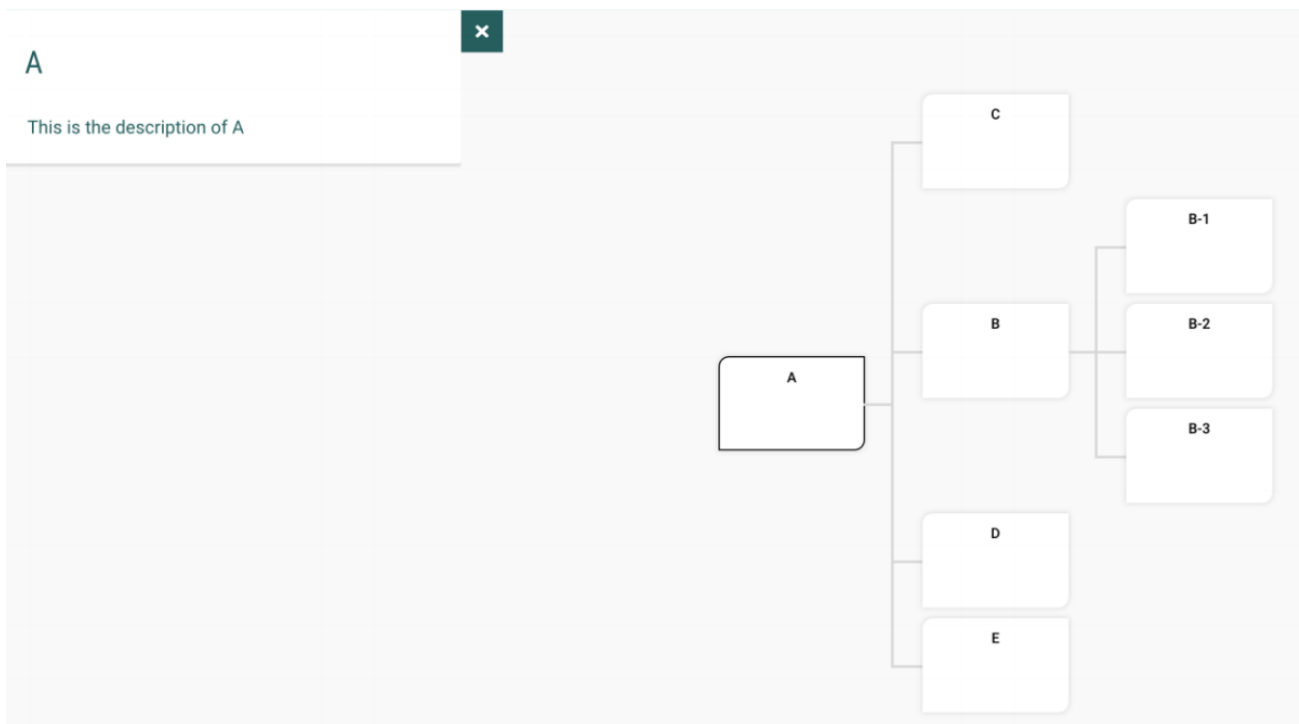
User Story

As a user, I would like to see the hierarchy of the provided data. If I click one of the nodes, I would like to see the detail of that node.

Design

UI and style is not the most important thing for this task. Because of that we don't provideany pixel perfect design.

You can use this image to imagine how you could visualize the data.

Functional Requirements

This task should contain all code in one repository. You can separate this task into 3 phases namely

Data import, Backend API, Frontend.

Migration

- It needs to be a single Node.js file (migration.js) which only stores the data. It would be a separate code which we can run it with "npm run migrate" command.
- You can read the data from the gist link or copy the data into your code.
- As a database we recommend you to use graph database (neo4j). If you are comfortable with other databases you can use them also.

Backend

- We expect from you to create a express.js server which has a GET endpoint which serves the data with correct hierarchy.

- Read the data from database and serve it to frontend with JSON format.

Frontend

- Read the data from Backend via GET endpoint.
- Please feel free to use a front-end visualization library like D3.js.
- You can use plain VanillaJS or VueJS.
- Usage of TypeScript, Sass, Webpack is your consideration.
- Hierarchy should be visible and understandable by the user.
- Tree layout could be from top to right or left to right.
- If user clicks one of the nodes, description and the name should be visible in a sidebar or popup. Selected node should be visible on the layout.
- User should be able to deselect a node. (In the sample design, user can click the cross icon and deselect it)

Additional Points

- Linting rules (we use ESLINT airbnb)
- Unit testing (we use Jest)
- Using principles of SOLID, DRY and Clean Code

Assessment Requirements

- Share a public git repository with source code
- Git Repo should contain a README that describes how to use the application.

If you have any questions please feel free to ask