Task: Data Visualisation

The goal of this technical test is to create a simple visualisation of the attached data using modern web technologies.

The data represents the evolution of the usage for a list of clients over the last 12 months.

For each month, there are 2 values:
- real usage
- estimated usage (i.e. what we expected the client to use)

The visualisation/application must allow a user to:

- at least see a graph of real / estimated usages for a specific client (could be hardcoded)
- **if you have time**, implement a way to change the client we're displaying the data
- **again, if you have time**, any other display you think relevant for these kind of data

Please try to formulate the above information in user story format

Deliverable

What we expect you to deliver is :

- **Documentation**: The user story you formulated
- **Executable**: Something easy to run with instructions (a temporary access to a VM, a docker-compose file to start-up all components, ...)
- **Database**: An access to the database (user / password to use)
- **Sources:** All source files in a single ZIP file sent back by email (or github)

Technology Choice

You may use any technology you think relevant to solve this problem. The only constraints are :
- the data should be stored in a database
- the graphical user interface should be displayed in a modern web browser
- there should be a back-end component in your architecture

Please, select technologies you're comfortable with.

Scope

Limit yourself to not spend **more than 3 hrs** on this problem.

Objectives of the task

- **Agile Practice:** User story formulation and time boxing (3h)
- **UX competencies:** if he included more displays, we'll see what kind of display he chose (table, graphical with d3.js, ...) and why
- **Technical decision making:** we'll ask the candidate during the interview to justify the technical choice
- **Architectural competencies:** we could also ask him which changes he'd do to the architecture to support many more data
- **Deployment:** we'll see how he packaged the solution (is it simple to use ? are the instructions easy to follow ?)
- and of course, how is the code ? Well written ? Are there some tests (even simple) ?