Technical test

Nurea is a start-up developing a tool to help diagnose and monitor vascular diseases, in particular aneurysms. Our software, which operates automatically, extracts geometric measurements of vessels from scanner images to provide the indicators needed by medical practitioners to evaluate the disease.

1. Backend

Development of a REST API for a dashboard tracking vital parameters of hospitalized patients. This API will be used by a front-end application (see 2nd section) to display real-time data to healthcare professionals.

Test objective

- Create a REST API that exposes patient data via a `GET /patients` endpoint.
- Create a server (for example FastAPI for Python). The server must handle HTTP requests and return the appropriate data via the API.

Functional requirements

1. **Endpoint**: **GET /patients**
    - Return a complete list of patients with all their vital data.
    - Data must be structured in JSON, following the example provided below.
2. **Endpoint: GET /patients/:id**
    - Return the details of a specific patient by patient ID.
    - Include all vital data associated with this patient.

Technical Requirements

1. **Language and framework**: use a modern backend language (Python, etc.) and an appropriate REST framework (FastAPI , Django, etc.).
2. **Database** :
    - Simulate a database using a local JSON file.

Example data

Use the following data to populate the JSON file: Example data

General questions :

- What is a REST API and what are its basic principles?
- How do you handle data validation in an API?
- What is middleware and how do you use it?
- How do you manage transactions in a database to ensure data consistency?
- What's the difference between a relational database and a NoSQL database, and when would you use each?

2. Frontend

Development of a dashboard for monitoring vital parameters of hospitalized patients. The goal is to create a simple, intuitive interface enabling healthcare professionals to view vital patient data (such as heart rate, temperature, blood pressure, etc.) in real time.

Test objective

Create a front-end application that allows to:

1. View the list of patients with their vital parameters in real time.
2. Display details of a specific patient with graphs of vital parameters.

Functional Requirements

1. **Display vital parameters in real time:**
    - Display a list of patients with an overview of their vital parameters (heart rate, temperature, blood pressure).
    - Update this data in real time or at regular intervals (every 60 seconds).
2. **Patient details:**
    - When a patient is selected, display his vital parameters in the form of interactive graphs for the last 24 hours.
    - Graphs should include heart rate, body temperature and blood pressure.

3. **Search:**
    - Enable patient search by name or medical record number.

Technical Requirements

1. **Framework**: use a framework of your choice (for example, Vue.js, React, Angular, etc.) the use of Vue.js will be appreciated.
2. **Backend API**: use the REST API you developed in the first section (Backend).
3. **UI/UX**: Make sure the user interface is clear, intuitive and accessible, with good error handling.

General questions :

- What are the advantages of using a front-end framework over traditional JavaScript development?
- What is responsiveness in a front-end framework, and how is it generally implemented?
- How do you handle global state in an application using a front-end framework?
- How do you handle routes in a front-end application?

Deliverables

1. Application source code, hosted on a version management platform (GitHub, GitLab, etc.).
2. A `README.md` file explaining how to configure and launch the application.
3. Optional: A video or screenshots demonstrating how the application works.

Test duration

Around 3 hours.