

Hi there and thank you for your interest in working with us!

We would really like to know more about the way you work, so we think that starting with a small coding challenge is a great way to get an idea of your style. Since we really value your time, we do not think that you should spend more than **4 hours** on this task.

Implementing software is always about trade-offs, so we are interested to see how you balance between clarity and getting-things-done.

The task reflects real-life scenario and challenges typical in our health technology projects, requiring you to demonstrate your skills in areas such as **database management**, **problem-solving** and **clean coding**. By observing how you **approach, manage, and deliver the task**, we aim to gain valuable insights into your capabilities and skills.

The idea of this project is to implement a backend application that will address the User Story:

Document Patient Interactions and View Patient History

User Story Specification

As a doctor,

- I want to **document interactions** with my patient and record the outcome of whether the patient is healthy or not,
- so that I can maintain accurate records and track the health status of my patients over time.

Additionally,

- I want to **see all interactions** with a specific patient,
- so that I can have a comprehensive view of their **medical history** and provide better-informed care.

This ensures that I can effectively document patient interactions, assess patient health outcomes, and review comprehensive patient histories as well as a list of all patients to provide the best possible care.



Acceptance criteria

- Create a technical concept including a testing strategy.
- Create a containerized (**Docker**) setup, so it can run locally, **with Python, a sqlDB and Fast API**.
- Create Endpoints to provide the mentioned functionality, that can be accessed via SwaggerUI.
- Implement example test (does not need full coverage).
- Submit your solution as public GitHub repository and share the link with us.

Remarks:

- Authentication is not required.
- Creation of users is not required, you can create an initial script to add a user from each group to the DB.
- We will really value: **concise and clean code, and testing**.

Have fun!