Test Project for Software Engineer Position

Thank you for considering Neurocage and taking the time to participate in the interview. This is a simple project to test your coding and problem-solving skills. There is no one correct answer and we are interested in new viewpoints. So if something seems reasonable to you, write it down and make sure to **explain** it properly.

What is the task?

In this task you are going to work with some concepts of Neurocage, this will help you get to know us a little better. In Neurocage, we are creating smart cages, that are used to investigate the condition of the rodents inside it. The main goal for a researcher is to find out how well things are. So there is a stream of data to the database, but to simplify things you are expected to do something else.

You are expected to design a web application that a user:

- View list of cages
- Create a new cage
- View details of a cage (on a new page)

Now let's talk details. A cage should have the following fields:

- ID: int
- Label: string
- If you need anything else, feel free, it's your cage!

Then you should consider a table to save sensor data. In the directory of the project, you can find another project, it is here to simulate the sensor. 25% of the time, it will fail to respond, but the rest of the time, it will send you a value as the health of the cage. In the web app you are developing, call its endpoint every 1 minutes and save the corresponding value. In the long run, you are going to have a time series with health values (Yest we need all). In order to test, how well the sensor responds, create a table that **saves the time it took to get an answer from the cage, and if it succeeded or not**.

Cage List:

- Add pagination
- Should contain a link to the detail page of each cage
- Should show the latest health status for each cage

New Cage:

- Nothing fancy, just create the cage!

Detail Page:

- Information about the cage
- Health values for today
- The functionality of the sensor for today

How to develop and deliver?

Now that we are on the same page, let’s talk about the technologies.

- You must use **Python**, but you are welcome to use any web framework or no framework at all, but we strongly suggest **Django**.
- You can use any database you want or just use **SQLite**.
- Create a git repository and deliver the project in it, if you created a private one, please grant access to **amir79esmaeili@gmail.com** and **hojats7731@gmail.com.**
- You must implement a UI (It is a plus if you can create a beautiful one 😉).
- Dockerize the web application and the database.
- Using a web server is a **plus** (Docker-Compose).
- Please write simple **documentation**
- You can access the sensor generator code from this link: Sensor Mock Script

Good Luck.