aidn

Case Assignment

Senior Fullstack Engineer

Introduction

As part of the recruitment process for the position of **Senior Fullstack Engineer**, we ask you to complete a case assignment to better understand your professional expertise and thought-processes in areas that we see as important to succeed and be happy at Aidn.

The case is designed such that there is no one Right Answer™. Rather, **the point of the exercise is to learn more about how you approach problems and communicate your ideas**. Feel free to make any assumptions necessary to complete the assignment—just make sure you tell us about them!

Please be prepared to give a walkthrough of your code and a brief demo of the working solution. Your audience will be comprised of engineers, so you may use as much (or as little) technical language as you wish. We look forward to hearing you draw parallels to your experience and listen to how you use your knowledge, experience, and intuition.

The conversation will be conducted in either **English or Norwegian**, depending on the preferences of the participants.

We have reserved **90** minutes for the interview—**30** minutes are allocated for introductions and your presentation, **45** minutes for discussion of the technical solution, and **15** minutes for follow-up questions from both sides.

Overview

Nursing home patients are at considerable risk for rapid deterioration. To identify important changes in condition as soon as possible, workers perform physical assessments at regular intervals to determine when extra monitoring is necessary to keep a patient stable.

During such assessments, several measurements are feed into an algorithm to produce a single numeric value called a NEWS score. Any score above a certain threshold results in escalated monitoring. This simple heuristic is one of the most important tools used in clinical wards.

Your assignment is to create a micro web application that implements a simplified version of the NEWS calculation and demonstrates your knowledge across both frontend and backend concerns. We want to stress that we do not expect you to spend a ton of time on the assignment. We will outline some basic requirements that must be met—anything you choose to do beyond that is entirely optional. Honestly.

You have complete freedom of choice regarding tech stack, programming languages, frameworks, libraries, tooling, etc. Having written that, our engineers are typically most familiar with C#, .NET and Next.js with TypeScript, so if you happen to be comfortable in those technologies, we would encourage you to choose them for a more thorough and engaged discussion.

Please read on for detailed requirements and expectations.

Expectations

Respect For Your Time

We encourage you to limit yourself to 2-3 hours. We believe that's enough time to meet the basic requirements without monopolising your free time or burdening you unnecessarily. If you want to spend more time, of course that is okay, but it truly is optional.

We know that isn't enough time to do everything exactly the way you would do things in the real world. **We expect you to prioritise some aspects of the solution and to ignore others.** As long as you tell us what choices you made and why, that's fine.

Again, our goal is to establish a practical basis for a detailed, concrete technical discussion. Perfection is not a requirement.

But Requirements are Required

While we offer freedom of choice where we can in an effort to maximise your productivity, **please follow the specification.**

The only way we can evaluate candidates fairly is if they do the same assignment.

What We Care About

1. The **correctness and quality** of your solution
2. The effectiveness of your **communication** with potential colleagues
3. Your critical thinking skills and overall **mindset**

High-Level Requirements

Frontend Requirements

Implement the web form—including styling and interactions—described by these Figma sketches. https://bit.ly/3S14C8D

The form only needs to handle measurement values. Units are standardised for these measurements and can be assumed.

The form should submit the measurements to an API endpoint. Do not handle the calculation in the frontend code.

The form should implement both submit and reset functionality.

Backend Requirements

Implement an API endpoint to accept measurement values and produce a NEWS score. Details are provided on the following slide.

Validate each measurement to ensure it falls within a defined range of acceptable values.

Return structure errors when appropriate.

All authentication of authorisation concerns can be ignored.

API Specification

Input **measurements** should be provided as a type and a value. There are three types of measurements that must be provided to calculate the NEWS score in this simplified example—body temperature (**TEMP**), heart rate (**HR**), and respiratory rate (**RR**). Input measurements should specify the type as a capitalised enumeration and the value as an integer.

For each measurement, your program should calculate an **individual score** by evaluating the input value against a defined set of ranges. All starting values are exclusive; all ending values are inclusive. Values outside of the defined ranges are invalid.

The three individual scores are then summed to produce a final **NEWS score.**

TEMP Range	Score
31..35	3
35..36	1
36..38	0
38..39	1
39..42	2

HR Range	Score
25..40	3
40..50	1
50..90	0
90..110	1
110..130	2
130..220	3

RR Range	Score
3..8	3
8..11	1
11..20	0
20..24	2
24..60	3

API Specification (cont)

For example, given the following input values:

```
{
  measurements: [
    { type: “TEMP”, value: 39 },
    { type: “HR”, value: 43 },
    { type: “RR”, value: 19 },
  ]
}
```

Your API endpoint should return:

```
{ score: 2 }
```