coreplus

Practical Exercise – Software Developer

For the requirements given below, please describe the design you would use to implement it, taking into consideration functionality, future modifications, performance etc. Once you have determined the design, please implement it in code (ie deployable application). Consideration of the following elements would be highly regarded:

1. Visual representation of the proposed solution
2. Deployment, uptime, security and maintenance considerations
3. Consider expandability to (unknown) future requirements
4. Consider testing strategies for backend/frontend code and possibly write any tests for core functions
5. Adding docker support

Introduction

A starter codebase has been provided with some very rough placeholder code for both the backend and frontend to help you get started quickly. Please use this code as a starting point and extend it to implement the business requirements below. You may add or modify the starter code (including to the UI layout as long as business requirements are met), make improvements to it in areas you deem are important and fix any shortcomings in it as you see fit.

Note: You will require dotnet 7 and nodejs to be able to use the starter code. Please run `npm i` *within the* `coreplus-sample-ui` *folder before running the ui project.*

Note: The data required for completing the following business requirements is provided in `/coreplus-sample-api/Data/` and must be used as is without any changes to the json files. While the provided data in the json files must be changed, feel free to make modifications to the existing source code as you see fit.

Business Requirements

1. As a coreplus account supervisor, I need a way to generate a financial report to analyse profitability of my practitioners over the past 3 years. The report should be accessible within the coreplus user interface, where the user can populate criteria and see results.

Report criteria:
* Which practitioners to report (this can be done by selecting a practitioner from the list) on Date range (this can be specified using some sort of datepicker or any other appropriate inputs)

Report output:
* For each practitioner, display cost and revenue per month within the given date range.

2. Once the report had been generated, clicking on a practitioner should display a breakdown of that practitioner's appointments, each with its cost and revenue, on the same screen. The summary report should still be visible on the page.

3. Clicking on an appointment, should display details of that appointment (e.g. date, client's name, appointment type, appointment duration) in another section of the page. The summary and breakdown should still be visible on the page.

4. The practitioners must be displayed in a list from which they can be selected. The list must be separated into two parts, one part for practitioners that are supervisors (i.e. with a practitioner level of `PractitionerLevel.OWNER` or `PractitionerLevel.ADMIN`) and the second part for any remaining practitioners.

5. **The submission must be a git repository link**. The submission, once handed back in, will be run using `dotnet run` for the API and `npm run dev` for the UI. If changes are made require any additional (or different) steps before the code can be run, these instructions must be provided in a README file. Additionally, a brief explanation of what was implemented and how it achieves the business requirements must be provided alongside the code.

Other non-functional requirements:

- The page should not timeout (or otherwise fail), even if the report takes several minutes to complete.
- Consideration must be given to performance and efficiency.