

Thank you for giving me an opportunity to complete the take-home task on ASP.NET and React. Please take a look at this report accompanying the assignment.

Requirements

The Back-end (REST API) was built using C# and .NET Core, which is similar to .NET Framework. An additional framework used is MVC; therefore, the API requests and responses are in JSON format. The API is currently storing data on SQLite, i.e., app.db file.

The application has multiple pages instead of a fully-responsive single-page one (due to the limitation in my current abilities which will be improved in the future).

to substitute my missing skill on TypeScript, I attempted to incorporate authentication but the work is currently incomplete at this moment.

Use cases

- 1. List all blood works:** The index page lists all blood works in a table format including created date, exam date, description, and 'details' to show the details of selected blood work. Users can search for records that matched with the text description and the range of given dates.
- 2. Create new blood work entry:** Users can fill data in a form at the end of the index page. Created date, exam date, results date, and "today" are validated against one another prior to allowing the blood work record to be created. Once a new entry is added, the list is updated right away. The form remains in the main index page.
- 3. Edit blood work entry:** Once clicking the "edit" button specific to certain records in the index table, users will see all the data filled and can change them. Created date, exam date, results date, and "today" are validated against one another prior to allowing the blood work record to be created. After editing, the user will be redirected to the list or index page.
- 4. View blood work details:** From the list of all blood works, once clicking the "details" button, the user can see all blood work details. At the bottom of the page, the user can click a button to edit the data and another button to go back to the index page.
- 5. Blood work report:** On the main page, users can see one-line graphs of their blood work information including hemoglobin, hematocrit, white blood cell count, and red blood cell counts.

Extra – identity users can register and login, but the application is currently not restricting access to the blood work records.

Other resources

I used Visual Studio Code as my code editor for building and debugging the web application. I also use Linux operating System, mainly Ubuntu 20.04 LTS. Since the Entity Framework tool does not function on my main operating system, I switched back and forth to Ubuntu 18.04 LTS on my virtual machine to handle the database update after migrations. Lastly, I used some instructions publicly posted online as a GitBook and many websites such as stackoverflow, c-sharpcorner, w3schools, canvasjs and Microsoft documentation. I also use Git version control to revert and reset the development.

Deliverables

The source code and database files are zipped and attached. Please see the instructions below for how to setup and run the application.

How to setup - after unzipping the submitted folder, please run the following commands.

```
cd BloodProfile
dotnet tool install --global dotnet-ef
dotnet add package Microsoft.EntityFrameworkCore.Design
dotnet add package Microsoft.EntityFrameworkCore.Sqlite
dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore
dotnet add package Microsoft.AspNetCore.Identity.UI
dotnet add package Humanizer
```

How to run the application – in CLI, please run: `dotnet run`, open a web browser and go to your local host that the application is listening to.

Screenshots

The screenshots illustrate the user interface of the 'BloodProfile' application. The top-left screenshot shows the 'Your Blood Work' list, which includes a table with columns for 'Created Date', 'Exam Date', and 'Description'. Each row has 'Details' and 'Edit' buttons. The top-right screenshot shows the 'Blood Work Details' page, which displays the 'Date Created', 'Exam Date', 'Results Date', and 'Description' of a specific record, along with various blood test results. The bottom-left screenshot shows the 'Add New Record' form, which includes input fields for 'Date Created', 'Exam Date', 'Results Date', 'Description', 'Hemoglobin (g/dl)', 'Hematocrit (%)', 'White blood cell counts (10⁹ cells/L)', and 'Red blood cell counts (10¹² cells/L)', along with an 'Add' button. The bottom-right screenshot shows the 'Edit Blood Work' form, which is similar to the 'Add New Record' form but includes a 'Save' button and a validation message for the 'Red blood cell counts' field.