university of groningen

faculty of science and engineering

eHealth Research Tool

Short Programming Project: Final Report

Stefan Malinovski - s4790804

Supervisor: F.F.M. (Fadi) Mohsen, PhD

February 14, 2024

Contents

1 Introduction

The eHealth Research Tool is a full stack web application that is meant to guide medical personnel through administering neurological tests, storing the patient's responses and computing a preliminary outcome. The examinations are also accompanied by videos and pictures to assist in the process.

In this report I will be going through my technology stack, each part of the app, as well as the limitations of my work and where future progress can be made.

2 Overview

The application consists of three parts:

- The Frontend - The view of the user as well and interface they interact with.
- The Backend - Responsible for communicating data between the frontend and the database.
- The Database - The storage for all the acquired data.

s The technologies I used for each component are **React, Express and PostgreSQL** respectively.

After choosing the technologies for my stack, I planned out the responsibilities for each component and began to work on them.

3 Frontend

React is one of the most popular and well developed front-end frameworks. It has been used for applications such as Instagram, Discord and Airbnb. The front-end itself is composed of 4 sections, the login/register forms, the patient information form, the quiz and the outcome tables. Each section has it's own route, with all routes being handled by the main App.tsx component.

After the medical staff logs into the app, the session is assigned a unique id from the Backend that is used to link all data from that session in the database. After a user fill our the login/registration form, an API call is made to he backend to log the request and if all the credentials are proper, they are then rerouted to the patient information form which acts in much the same way. Next is the quiz itself, whose order is dynamically altered depending on the users answers.

Figure 1: User View during quiz

After each answer is submitted, an API call is made to the backend which saves the data in the database

and responds with the following question in the quiz. During each section there are buttons that, when clicked, supply the user with instructions for the particular section, an image or a video.

Finally, after the last question, the user is shown the final page, containing 2 tables, which display the potential outcomes and selected answers.

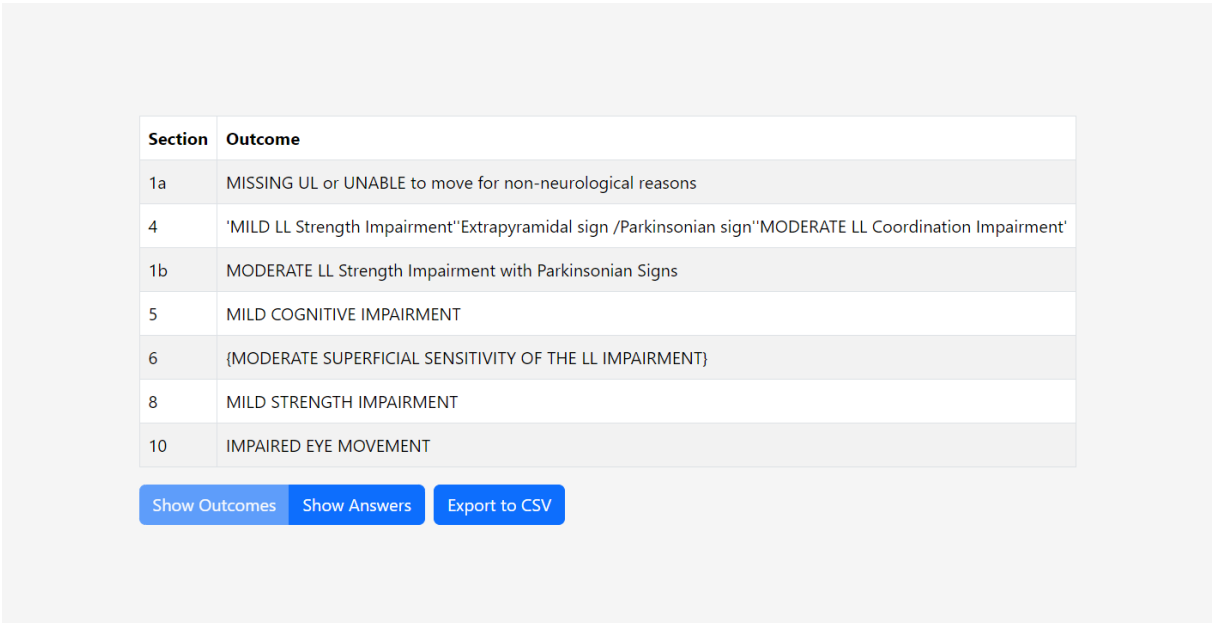

Figure 2: Outcomes table

There is also an additional button to export both tables to .csv files.

4 Backend

The backend is written in Express, a fast web framework for Node.js, with Node.js being a JavaScript runtime environment. The backend acts as a middleman between the frontend and the database and mainly consists of **2 API endpoints for each possible question**. One POST method, for the frontend to post data that needs to be saved, and another GET method for requesting relevant question information.

Additionally, there are endpoints that check whether the correct login information as entered, whether the login information can be registered, an endpoint for saving the patient information and 2 endpoints for requesting the outcomes and answers for that specific session.

5 Database

PostgreSQL is the database framework that is used by many big name companies such as Apple, Spotify and Instagram. For this project, the database consists of **5 tables**, those being **“login_info”**, **“user_entries”**, **“patients”**, **“outcomes”** and **“data”**.

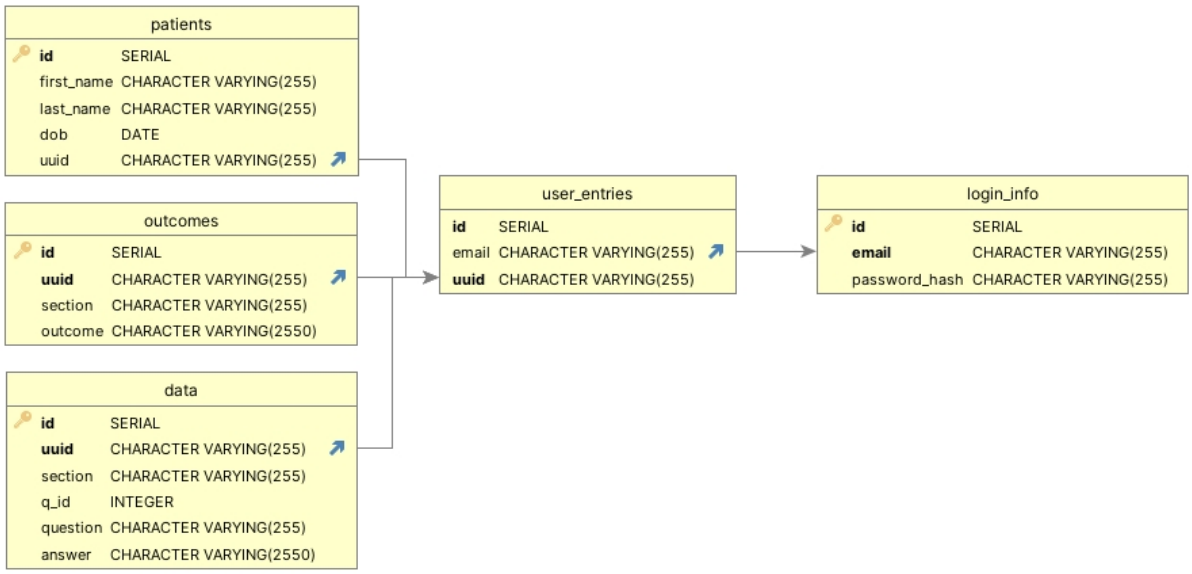

Figure 3: Database

The login info table stores the emails and hashes of the passwords that are used as login credential by the users. The **email** field is **referenced as a foreign key** by the email field in the "user_entries" table. When a user logs into the app, the backend assigns a unique id to that session, this is the **"uuid"** field in the "user_entries" table. The uuid is **referenced as a foreign key** in the "patients", "outcomes" and "data" tables.

The patients table contains the patient information that was obtained through the patient form in the front end. This information is linked to a specific user entry through the uuid. The data table contains all the questions and answers from the user and the outcomes table contains the possible outcomes that are assigned after each section of the quiz.

6 Limitations and Future Directions

Most of the frameworks I used, I worked with for the first time during this project so I would like to go through the entire codebase so see if there are any opportunities to decrease technical debt. Additionally I would add a button to go to the previous section, which technically is already possible by just going back in your browser, this however does not remove the database entry.

Lastly, for the web app to be deployed it needs to be hosted on a web hosting service such as Microsoft Azure or AWS.

7 Conclusion

Overall, I believe I delivered on the requirements set out by the stakeholders and am very pleased with the work I completed. I was granted the opportunity to learn and gain experience with full stack development and all of the frameworks I used to the project, and for that I would like to thank my supervisor for the project, Fadi Mohsen, for the amazing opportunity.