4ME302 Assignment 3: Authentication service and access roles

Annamária Váradyová
School of Computer Science, Social Media and Web Technologies, Linnaeus University, Sweden
av222xc@student.lnu.se

Abstract. In this paper, we develop and implement an application that accepts JSON, XML, and CSV data and manipulates and visualises it using various visualisation techniques to achieve various desired goals.

Keywords: web API, data formats

INTRODUCTION

This document aims to create a web application. The mission is to build an application that aids Parkinson's disease sufferers, medical professionals, and researchers. The objective is to improve the networking infrastructure, authentication mechanisms, and role management for medicine-related processes.

The app supports several third-party authentication systems, such as Facebook, GitHub, and Google. Using links from them, the user can log into the system. Depending on the authorization service they select, the user receives a specific role and access to various resources and functionalities. The user may be a patient, doctor, or researcher.

- *If the user is a patient, a selection of YouTube films describing the symptoms and activities of Parkinson's disease are available, and they can load data using the form.*
- *If the user is a physician, they can view patient data, also have a geographical overview of the patients, and he can choose patients from the map.*
- *If the user is a researcher, they can read the most recent news https://www.news-medical.net, observe patient data and activities and add a note to them and also have a geographical overview of the patients, and he can choose patients from the map*

Firstly, start by going over the development environment that we used to create the application.
Second, we'll explain how the provided application functions.
Finally, we will mention a few enhancements that would be beneficial to include in the future.

DEVELOPMENT ENVIRONMENT

We settled on using Visual Studio Code1[1] as the source code editor along with the Node.js[2] and Express[3] development environment.

Opening a file on the server and returning the content to the client is a frequent activity for a web server that can be accomplished using Node.js. The most crucial tool for interacting with Node apps is npm[4]. Npm can also be used to run tests and tools that are used during the development process. Together, the binary packages, installers, operating system package managers, or source code are used to install Node and the npm package manager. Then, as a dependency of each of your unique Express web applications, npm installs Express.

Npm packages:

- *Express*
- *Express-session: an HTTP server-side framework used to build and administer session middleware*
- *Mongoose: MongoDB object modelling tool designed to work in an asynchronous environment*
- *ejs: simple templating language which is used to generate HTML markup with plain JavaScript*
- *ejs-mate: creates reusable code*
- *Passport: authentication middleware*
- *Passport-facebook*

[1] https://code.visualstudio.com
[2] https://nodejs.dev/en/
[3] https://expressjs.com
[4] https://www.npmjs.com

- *Passport-github*
- *Passport-google-oauth20*
- *Passport-local*
- *Passport-local-mongoose*
- *Usetube: allows you to scrape YouTube data without worrying about getting blacklisted*
- *Axios: implements the promise API, which is native to JS and used to make HTTP requests from Node.js or XML from the browser*
- *Cheerio: parses markup and provides an API for traversing the resulting data structure*
- *Body-parser: process data sent in an HTTP request body, provides express middleware for parsing JSON, Text, URL-encoded, and raw data sets*
- *Csvtojson: a comprehensive nodejs csv parser to convert csv to json or column arrays*
- *Connet-flash: special area of the session used for storing messages*
- *Method-override: lets you use HTTP verbs such as PUT or DELETE in places where the client doesn't support it*

Mongoose[5] is the database management solution we opt for, is a Node.js and MongoDB[6] object data modelling (ODM) library. It facilitates the management of data relationships, offers schema validation, and translates between the representation of objects in code and their MongoDB counterparts.

MapBox[7] is a platform for location data that provides the maps and location services. We used this platform to capture the location of our patients.

APPLICATION DESCRIPTION

On the application's home page, there is a sign-up form and links for Registration, Sign-up with Google, Sign-up with Facebook and Sign-up with GitHub.

- *The Registration button takes you to the form through which you create your profile.*
- *Sign up with Google to create a profile of the role patient.*
- *Sign up with Facebook creates a profile of role physicians.*
- *Sign up with GitHub and create a profile of the role researcher.*

The profile of the patient display your name, role (patient) and email. Three buttons: Show records, Show Data and Show videos are also included.

- *The Show Records provide you with information about your therapy and session details.*
- The Show Data display the data you have loaded. On the page, there is a button Add Data that leads to the form where you insert your data.
- *The Show Videos direct you to a YouTube "Parkison's foundation" playlist.*

The profile of the physician displays your name, role (physician) and email. Additionally button: Show patients.

- *The Show Patients all patients will be displayed. There is also a map on which you can track where the patients come from. After clicking on the patient's name, you will see his records and relevant data. Notes created by the researcher will also be displayed for the data.*

The profile of the researcher displays your name, role (researcher) and email. Two buttons are also included: Show patients and Show news.

- *The Show Patients all patients will be displayed. There is also a map on which you can track where the patients come from. After clicking on the patient's name, you will see his records and relevant data. Next to the notes there is a form through which it is created and attached to the data.*
- *The Show News pulls news for you from the website medical.net.*

The application has a navbar and footer.

In the navbar are located links Home, Profile, Sign Up, Register/ Log out.

- *The Home directs you to the main page, which is the sign-up page and other links (Sign up with Google,...).*
- *The Profile shows your profile.*
- *You see Log out if you are signed up. If not, you will see Sign up and Register.*

In the footer, there is only the creator name.

[5] https://mongoosejs.com
[6] https://www.mongodb.com
[7] https://www.mapbox.com

APPLICATION ENHANCEMENT

The programme is extremely straightforward, but in the near future, they plan to update the look and include new features like chat, therapy notification, and a blog.

The layout ought to be simpler and more contemporary. We want to design a profile that makes everything visible.

Additional communication is crucial, and it would be easier to do it directly in the application.

Notifications for specific patients would be helpful so they wouldn't forget their treatments and appointments.

A blog where users could help and counsel one another on an individual basis.

CONCLUSION

Finally, using the Node.js development environment, we produced an application, with the data being kept in the MongoDB library.

The user can sign up for the eHealth application directly or using Google, GitHub, or Facebook. Patients, doctors, and researchers are users. Various profiles are shown depending on the user's role. The patient is shown his schedule of events and suggested videos. Physicians observe the records and data of their patients and they are able to follow their position on the map. Researchers observe patients, their records and data and their location from where they come from. Be able to write a note on the given data. Read the articles on medical.net.

The programme still has space for improvement, and I would include a chat and a blog. Enhancing the design would be beneficial specifically for patient usability.

It is a terrific beginning for a new application, and the business would benefit customers.

REFERENCE

Short video: https://youtu.be/YYtVuz9_iKA
Code repository: https://github.com/Anna8295/assigment3