

4ME302 Assignment 2: 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 build an authentication system that works with several services, like Twitter and Google. Depending on the access rights of the authenticated user, the system directs the user's request to different places.

Keywords: network infrastructure, authentication mechanisms, role management

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 collection of YouTube videos outlining the activities associated with Parkinson's disease is offered.*
- *If the user is a physician, they can view patient data*
- *If the user is a researcher, they can read the most recent news <https://www.news-medical.net> and observe patient data and activities.*

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 as the source code editor along with the Node.js² and Express³ 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 npm⁴. 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*
- *Passport-github*
- *Passport-google-oauth20*

¹ <https://code.visualstudio.com>

² <https://nodejs.dev/en/>

³ <https://expressjs.com>

⁴ <https://www.npmjs.com>

- *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*

Mongoose⁵ is the database management solution we opt for, is a Node.js and MongoDB⁶ 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.

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. Two buttons: Show records and Show videos are also included.

- *The Show records provide you with information about your therapy and session details.*

(I was unable to upload and open the file that we received. An error was displayed by my programme. I made a form that allows the patient to draught his own treatment plan. He can select a time and the therapy's name, as well as include a comment.)

- *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 show you your patients' information regarding their sessions and therapy.*

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 show you your patients' information regarding their sessions and therapy.*
- *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 my name.

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.

⁵ <https://mongoosejs.com>

⁶ <https://www.mongodb.com>

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 their patients' behaviour. Researchers observe patients, their activities, and medical.net articles.

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/6gEhZXQRGRw>

Code repository: https://github.com/Anna8295/ehealth_app