Personal Fitness (MERN) Site

Harrison Spain  
Undergraduate Computer Science Capstone  
Capstone Advisor: Professor Reale  
SUNY Polytechnic Institute  
Spring 2023

# Introduction

The purpose of this project is to create a full stack MERN application that will serve as an interactive business page allowing clients to contact the owner, purchase plans in the form of memberships, view content and maintain a sense of certainty by gaining details about the gym.

To have a working web page the front end, and backend must properly communicate with one another. For example, it is simple to create a component with React and implement routes on the frontend of the project. Once you begin to fetch data from the backend to populate on the given component with the intended data things can get complicated quickly. As you add more libraries and implement changes you must ensure that you are not going to break the project. To avoid running into complications it can be necessary to use some sort of version control system, in this project we use Git. Simply put, Git allows the programmer to make changes and add them bit by bit to avoid breaking the application.

An online presence allows a small business to connect with customers in their local environment. The use of react utilizes components and routes essentially removes wait times between pages which keeps impatient users from navigating away or choosing alternatives. During the plan selection process, the customer enters his or her credentials which are stored into the Mongo database allowing the business owner to have a sense of current and potential customers. The use of a payment api will allow the business owner to manage active customers, and view payment details all in one place.

# Related work

React - Is a component-based framework that is used to create painless interactive User Interfaces.

Mongo Database - MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need

Node - Is an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications.

Express –is a Node.js framework designed to build API's web applications cross-platform mobile apps quickly and make node js easy.

https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js

https://legacy.reactjs.org

https://nodejs.org/en/about

https://www.mongodb.com/what-is-mongodb

# Method

When creating the website, the basic structure was to have a navigation header that allows the user to access each of the three main components. These components are the Plans, Content, and Bio page. The body of the webpage will consist of each component being routed to at a time. The bottom of the web page will consist of a quick contact link section to schedule a consultation with the business owner and below that a footer. The footer includes links to all off their social media accounts, along with the business name.

# Results

Each customer can find a plan that works best for their current desired outcome. They can easily reach out for additional information via the schedule consultation footer. The content page provides video explanation of workouts that ca be filtered according to muscle group. For customers who are interested in seeing where they will work out, they can navigate to the bio page where previews of the gym are included.

# Discussion

I learned a lot about Redux and the importance of gathering state information. Redux allows you to simply maintain user data as they navigate through the site. This can make keeping track of the customers information along with a plan they may be interested in easier as you can manage it all through Redux.

Creating responsive pages that adjust based on the user’s device went very well along with creating components and routes. On the other hand, Both the Axios Crud operations and payment API have been very difficult to implement and may require more tie than initially intended for this project.

I believe having stronger resources to access would have been beneficial. In this case I was using some documentation along with an outdated e commerce course to create my page. The problem with this is that as frameworks age their methods begin to become deprecated which in turn changes what you can and cannot use along with how they affect other pieces of your code.

# Conclusion and Future Work

During this project I was able to create a personal fitness page that is essentially ready for deployment. The desired customization has been completed on the frontend, and all that is left is to complete the payment processor and get the Axios calls working for posting user credentials to the database. In the coming weeks I plan to implement Stripes payment processing API which will allow users a secure checkout and act as a way for the business owner to manage active members.

# References

[1] https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js

[2] https://legacy.reactjs.org

[3] https://nodejs.org/en/about

[4] https://www.mongodb.com/what-is-mongodb

[5] https://stackoverflow.com/questions/71270157/what-is-the-index-html-file-used-for-in-react#:~:text=public%2Findex.,be%20managed%20by%20React%20DOM.

# Appendix

Mongo DB:

1. Create an account, download, or use online cloud version of database, <https://www.mongodb.com>
2. Obtain API key, in this case the copy is the mongo URI is stored in the .env file in the root directory of the project.

Node:

1. Install node on your device

<https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-6.0.5.zip>

React:

1. Open your text editor’s terminal and enter npx create-react-app

The best way to get this project up and running would be to copy and paste the following dependencies to your “package.json” file open your text editor’s terminal and navigate to each package.jsons directory. You can then enter “npm I” to install them all at once

The Root Package.json’s dependencies are as follows:

    "cors": "2.8.5",

    "dotenv": "^16.0.3",

    "express": "^4.18.2",

    "express-async-handler": "^1.2.0",

    "mongoose": "^7.0.2",

    "node-fetch": "^3.3.1",

    "save": "^2.9.0",

    "uuid": "^9.0.0"

The Front end Package.json’s dependencies are as follows:

"@fortawesome/fontawesome-svg-core": "^6.4.0",

    "@fortawesome/free-regular-svg-icons": "^6.4.0",

    "@fortawesome/free-solid-svg-icons": "^6.4.0",

    "@fortawesome/react-fontawesome": "^0.2.0",

    "@paypal/react-paypal-js": "^7.8.3",

    "@reduxjs/toolkit": "^1.9.3",

    "@testing-library/jest-dom": "^5.16.5",

    "@testing-library/react": "^13.4.0",

    "@testing-library/user-event": "^13.5.0",

    "axios": "^1.3.4",

    "babel-plugin-macros": "^3.1.0",

    "bootstrap": "^5.2.3",

    "react": "^18.2.0",

    "react-bootstrap": "^2.7.2",

    "react-dom": "^18.2.0",

    "react-icons": "^4.8.0",

    "react-redux": "^7.2.9",

    "react-responsive-carousel": "^3.2.23",

    "react-router-bootstrap": "^0.26.2",

    "react-router-dom": "^6.8.1",

    "react-scripts": "5.0.1",

    "redux": "^4.2.1",

    "redux-devtools-extension": "^2.13.9",

    "redux-thunk": "^2.4.2",

    "swiper": "^9.1.0",

Git hub repository: https://github.com/spainh/PF-Training