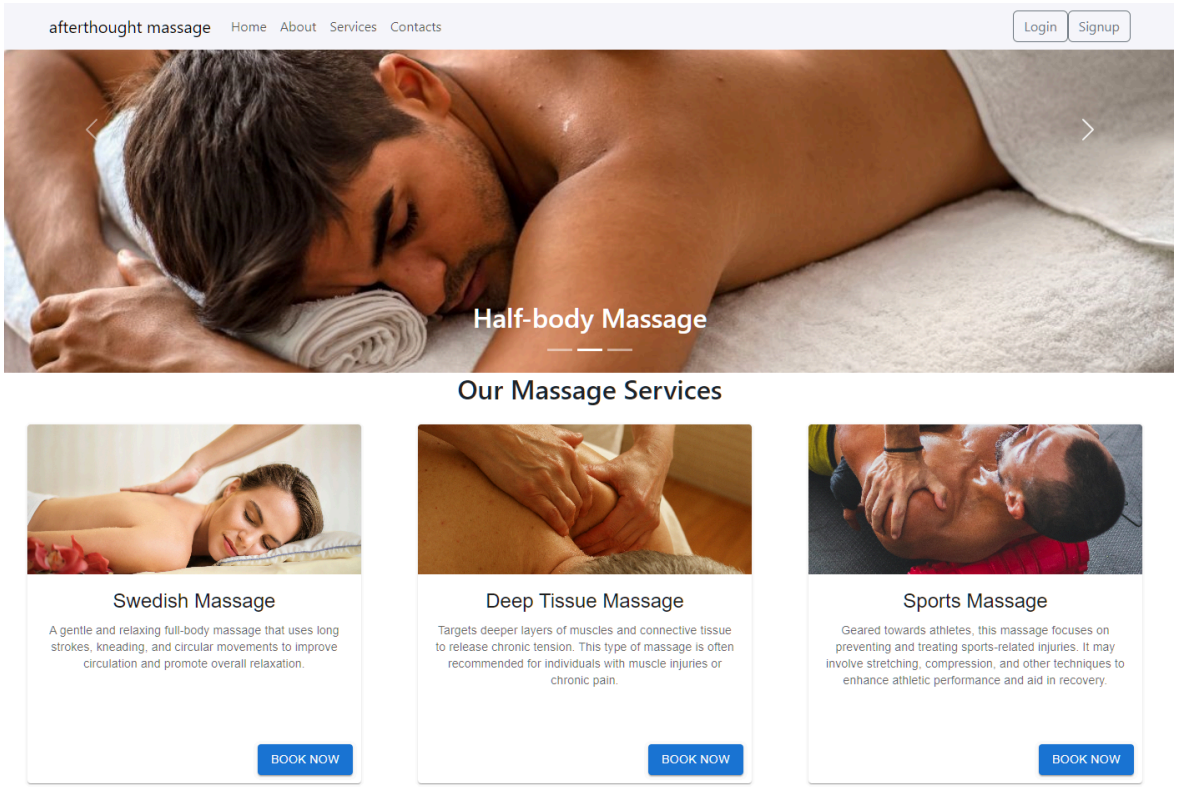
CMPT 496

Afterthought Massage Website

By: Lyndon Yeung, Cedric De Leon



Introduction

This capstone project is a web application for people who wish to gain relaxation through Massage therapy. Our client, Micheal Li, hopes to open a clinic of his own to soothe patients from their stress and tension. The focus of this project is to provide a website with seamless user experience to easily schedule patient appointments. By leveraging web technologies such as Javascript React and Node, we can ensure that the website not only looks visually appealing but also functions smoothly.

Technologies

Design Phase

To start off our project, we started with designing our front-end and back-end infrastructure. During this phase, we experimented with multiple tools such as draw.io, Figma, Azure, and GitHub. First, we wanted to create an interactive website design focusing on our front-end design. The purpose of this is to establish our main theme and UI elements and show them to our client. After approval, we based our entity relationship diagram on the website design, creating it on draw.io. Finding a place to store our database posed a challenge, as we wanted a cheap way to maintain our data while having remote access. We considered both MongoDB and Azure. Since we are already learning new technologies in both front-end and back-end, we decided to use Azure with an MSSQL database for familiarity and more convenient commands through SSMS.

Front end

For our front-end design technologies, we used HTML, CSS and Javascript with the React library for component based website building. We made use of React version 18 tools such as hooks, routers, components and APIs. To create a uniform look and feel, we opted for Material UI with Bootstrap v5 touch ups for inline styling. We also used multiple open source libraries for our features, which we will explain under “Main Features.” JWT token was used here to store logged in user data in the browser local-storage to have site wide access. We also utilized the Axios library, to send asynchronous HTTP requests to our REST API calls retrieved from our Node Server.

Back end

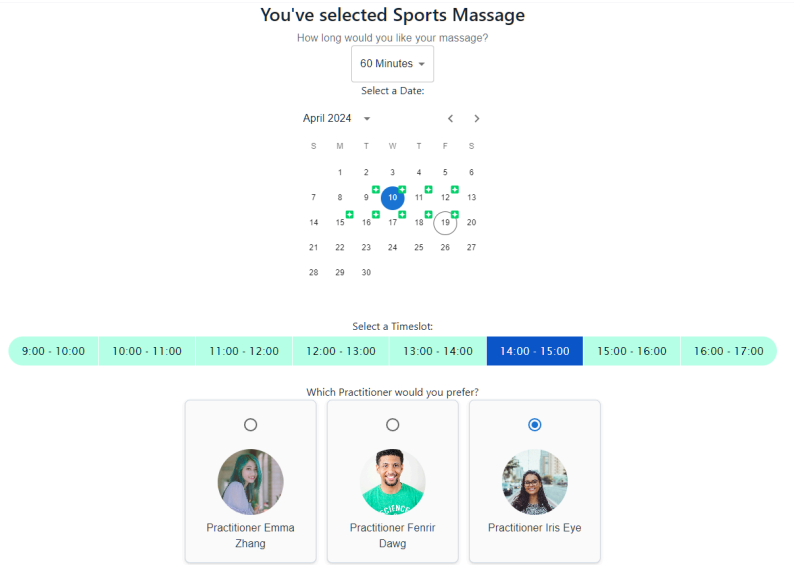
In the back-end we used the express framework and connected it to the MSSQL database. There are several http routes for several API calls such as entering or retrieving information from the different tables in the database. For our design, we have tables for employees and customers, and a separate table for users if customers want to track their booking history after registering their account. Some database operations include registering a new user through the sign in page, adding a customer’s booking when it is created, or viewing the employees scheduled massages. The JWT is used here to create the token and sent to the front end. Another table for the refresh tokens is planned to be added to make the authentication more secure.

Main Features

Service Selection

The service section of the main page shows all the available massage services offered. It includes the name, picture and description of the massage service. The cards use the Material UI library from React for the smooth and responsive animation design. Selecting one of the services will bring you to the appointment booking creation page for that service where you can customize your booking.

Appointment Booking Creation



The main feature of this website will be the massage service booking creation, where clients can easily schedule their appointments online. We've done multiple approaches with our initial design in Figma. One approach was to book the practitioners first and then book the time after. Ultimately we went with our second approach which is to book a time first and then choose the practitioners, this way it allows more opportunities to try out new practitioners since it will be for a new clinic.

After selecting a massage service, the user is then greeted with a booking page with multiple variable components. First is a drop-down menu to select a booking duration split into 75, 60, and 45 minutes, this is defaulted to 60 minutes. Next is a calendar date selector indicating stars for practitioner availability. After selecting a date, a timetable is then shown based on the massage duration. Finally, after selecting a time, a list of practitioners is then displayed with their images showing who is available during that appointment period. After the user chooses their desired practitioner, we then show them a confirmation screen. On this screen, the user would see their appointment details and a two-sectioned accordion.

The first section shows a stripe payment method inputting their credit card as an option to pay ahead, and the second is a textbox with a confirmation button to send the appointment details to.

Stripe Payment Integration

Integrating the stripe API was a challenge since I haven't used it before. Creating the checkout component involved both the front and back-end to safely process transactions. It required a Stripe account to obtain API keys and create product payments. Each product in Stripe was essentially a massage service with 1 booking duration. With 5 services and 3 durations each, this means a total of 15 products to be made. Luckily the documentation with Stripe was vast and easily accessible.

User Authentication

A customer can create his own account with the website by using the signup page and their information will be stored in the database after the information has been validated. For user authentication, we use JSON Web Token to authenticate the user and is created and stored into a cookie every time the user logs in. The token is sent with every request to the backend that needs the authentication where the token is then verified and either fulfills the request or gives an error if the token doesn't match. This allows for features such as customers checking their booking history or admin only features if their token is verified and they are authenticated.

We initially used express-session to authenticate the user, however we found out session wouldn't work well with our design, so we switched to tokens instead.

Booking History

In this feature, a customer can check in the front end all the bookings they have made with the account. Their login credentials must be verified during the page's request to retrieve the information. Here, they can check the information and status of the individual bookings. If the bookings are not marked as finished, then they will have an option to cancel the booking.

Weekly Event Viewer

This is an admin side feature, it is a page to show a selected practitioner's schedule on a weekly basis. There are three components to the page, first is a drop-down menu that displays a list of practitioners, second is a small calendar to navigate selected week dates, and lastly, is a weekly time grid displaying each practitioner's booking events colour coordinated within the grid. We integrated this with the open-source react package from DayPilot lite.

Employee Scheduler

This admin side feature displays a screen to assign practitioners their weekly shift schedule. The page consists of two main components, a form and an employee shifts grid. The form has inputs for the practitioner to be assigned, the date, shift start time and end time. Adding shifts will add a time slot for the selected practitioner with the corresponding date. Unfortunately we didn't get enough time to fully implement this with back-end functionality.

Future Implementations

Some of the small changes for the future are email confirmation for signing up, email notifications for confirmed booking, refresh token for authentication. and restrictions of canceling a booking. These changes add extra security for the web service by preventing a booking from being canceled moments prior to the scheduled time or fake emails from being registered.

Larger changes we want to implement are a rating system for finished bookings, back-end functionality for employee scheduler, and more admin features. These admin features include being able to add/delete and view employees from the front-end instead of from the database, update the massage services on display, and manually create a booking for walk-ins.

Lastly, we also want to look into web deployment when the time comes for when Michael opens up his clinic. We will also delve deeper into search engine optimization strategies for this website, aiming to appear more frequently in searches of Edmontonians looking for massage services.

Conclusion

In conclusion, this web application is a booking service for a future massage clinic for Michael Li, that uses Javascript in React and node. These web technologies were new for both of us, so we learned the different libraries and framework necessary for running and testing the application. It was also new using tools such as figma or draw.io to map out the design. Some of the challenging parts of the project were sharing the variable states across the different Javascript files in the front-end and connecting the front-end to the back-end using Axios. The service features a booking service, payment option, and user authentication. Customers will be able to book an appointment for a massage and pay on the website while tracking their history. Other admin features include viewing the employees booking and viewing their working schedule. Future changes will be aimed towards making the site more accessible for the admins and more user-friendly.