

AZ Bookings

Project Engineering

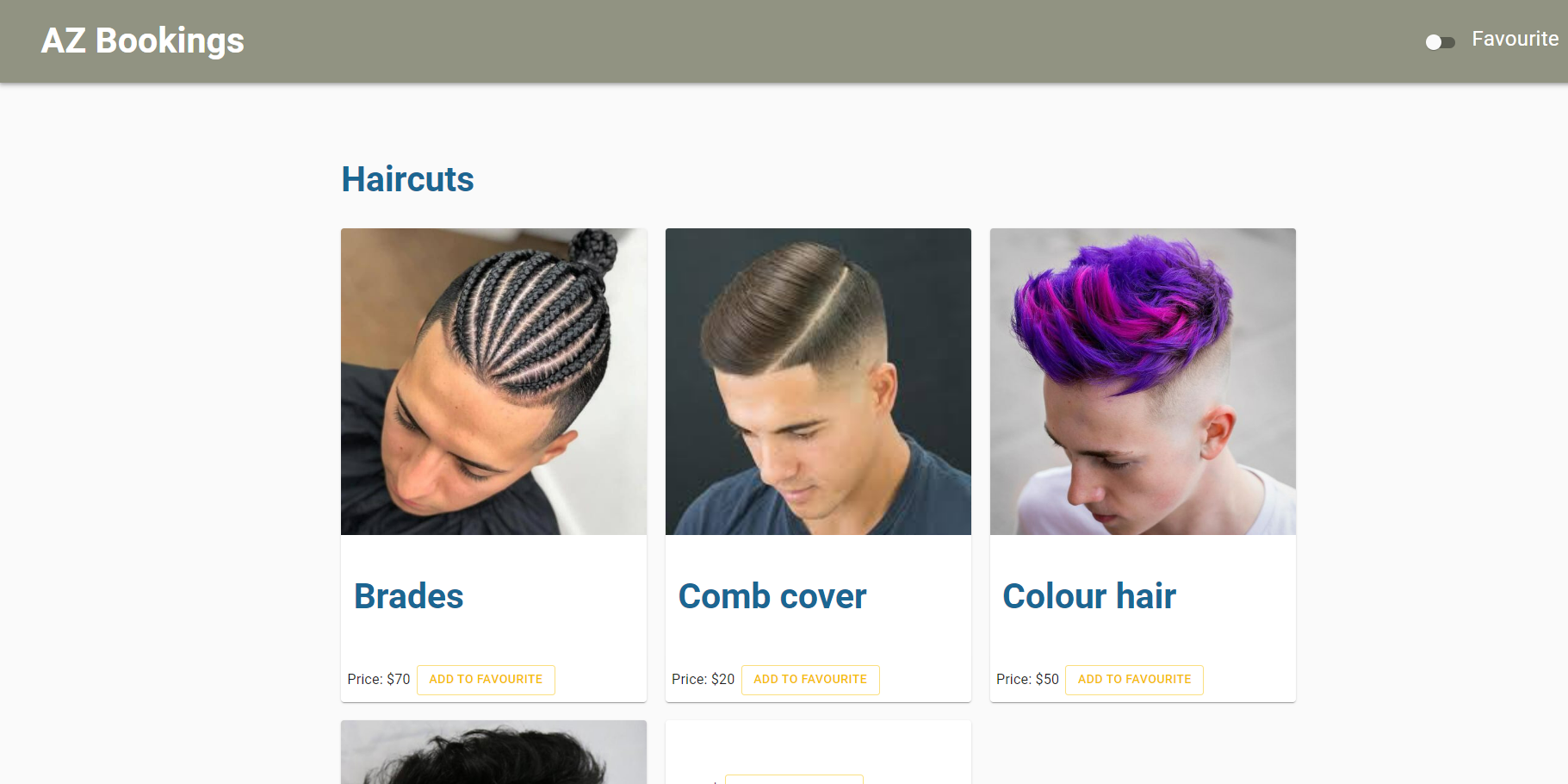
Year 4

Ashar Zafri

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Galway-Mayo Institute of Technology

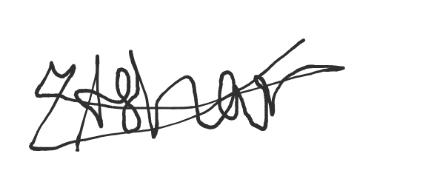
2021/2022

  
**Figure 1: AZ Booking Main Page**

**Declaration**

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgements**

My professors Nail OKeffe, Paul Lennon, Michelle Lynch, and my supervisor Brain O Shea all supported me in the development of my project. I am grateful for all of my lecturers' support and help.

**Table of Contents**

[1 Summary 6](#_Toc112877945)

[2 Poster 7](#_Toc112877946)

[3 Introduction 8](#_Toc112877947)

[4 Project Architecture 9](#_Toc112877948)

[5 Project Plan 10](#_Toc112877949)

[6 Project 11](#_Toc112877950)

[6.1 Frontend 11](#_Toc112877951)

[6.2 Backend 12](#_Toc112877952)

[6.3 AWS (Amazon Web Services) 12](#_Toc112877953)

[7 Frontend 12](#_Toc112877954)

[7.1 React 12](#_Toc112877955)

[7.2 NextJS 13](#_Toc112877956)

[8 Backend 23](#_Toc112877957)

[8.1 NodeJS 23](#_Toc112877958)

[9 AWS (Amazon Web Services) 23](#_Toc112877959)

[10 Challenges 27](#_Toc112877960)

[11 Conclusion 28](#_Toc112877961)

[12 Reference 29](#_Toc112877962)

# Summary

I'm going to develop a barber booking web app for my senior project. During Covid, I had this thought. Due to restrictions, barbers could only have a certain number of customers at a time once the lockdown began. Covid resulted in long lines and overcrowding at barbershops. Az booking will help customers to select a time when the barber is available and avoid waiting time by using this Web app.

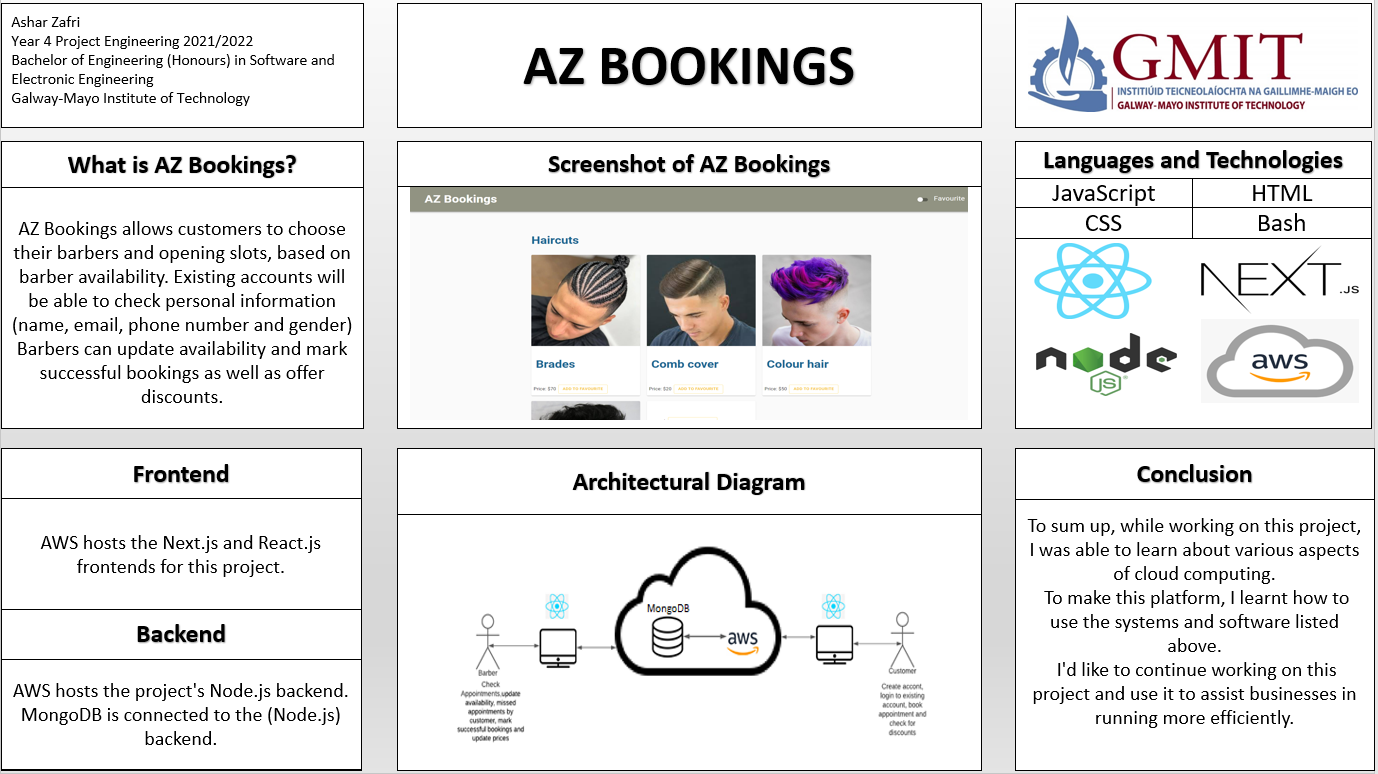
AZ Bookings is an online booking application that allows customers to choose their barbers and opening slots, based on barber availability.

A NextJS frontend and a NodeJS backend are used in this project. AWS (Amazon Web Services). hosts both the frontend and the backend. Az booking has been designed to be as user friendly as possible

On the other hand, Barbers will be able to update availability and mark successful bookings. Barber will also be able see missed appointments by customer. Moreover, Barber should be able to update prices.

To make this possible, I will be using React, HTML, and JavaScript as codebase. NoSQL database through AWS cloud to retain customer data (customer/barber accounts and schedules).

# Poster

  
**Figure 2: AZ Booking Poster**

# Introduction

My barbershop during COVID 19 served as the inspiration for AZ Bookings. There were numerous restrictions during COVID 19, which led to lengthy wait times. This practical issue is what AZ booking is designed to solve.

AZ Bookings is an online booking application that allows customers to choose their barbers and opening slots, based on barber availability. AZ Bookings work with an account where you can create a new one or log in with a current one. Existing accounts will be able to check personal information (name, email, phone number and gender), previous bookings, give discounts (based on the number of bookings).

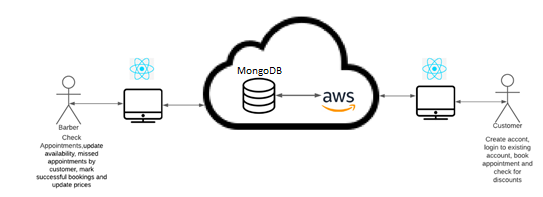
On the other hand, Barbers will be able to update availability and mark successful bookings. Barber will also be able see missed appointments by customer. Moreover, Barber should be able to update prices.

To make this possible, I will be using React, HTML, and JavaScript as codebase. NoSQL database through AWS cloud to retain customer data (customer/barber accounts and schedules).

This report will outline all the procedures, difficulties, and technologies that were used in this project.

# Project Architecture

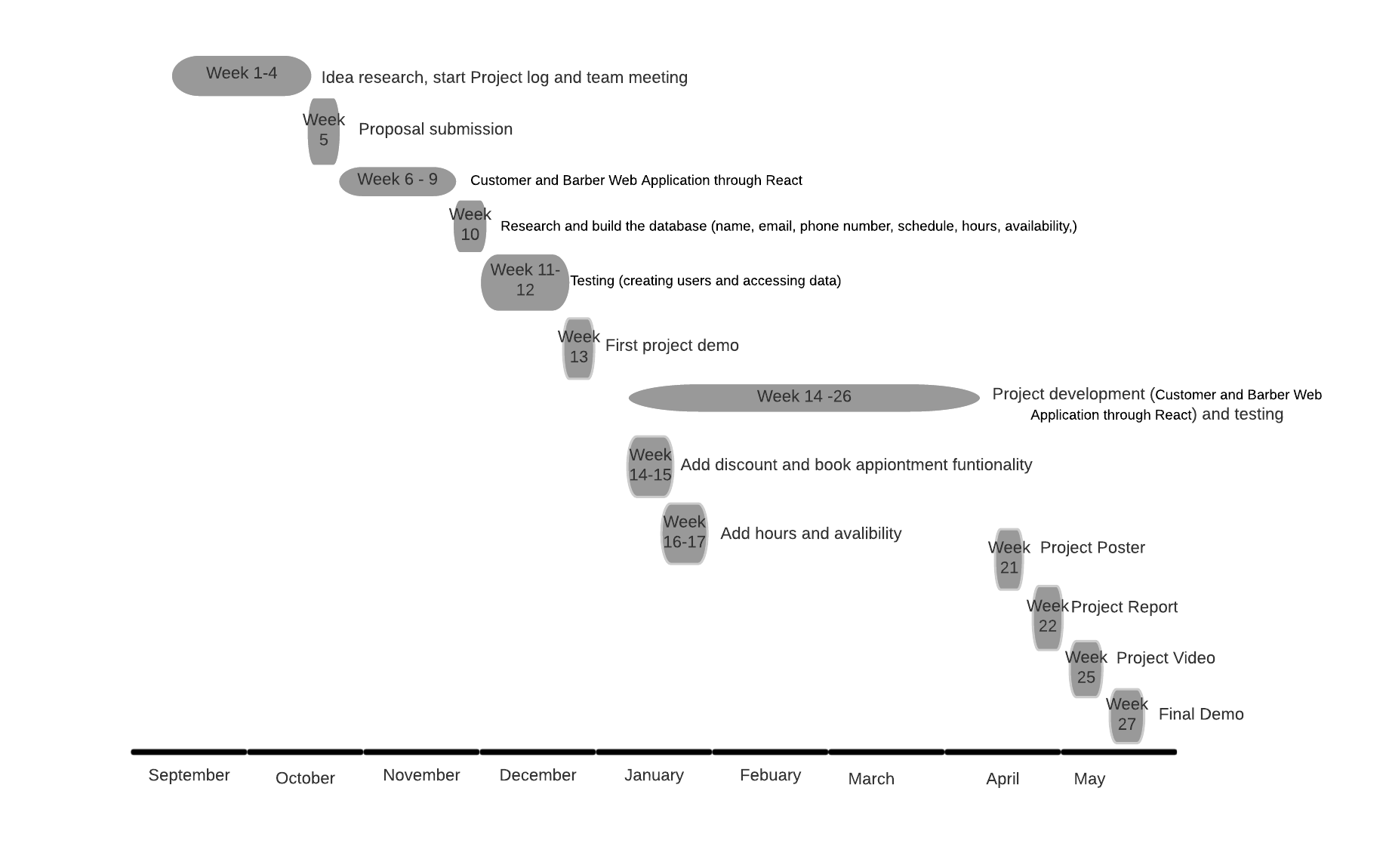
As stated previously, this project is divided into two parts: the frontend and the backend. Both NextJS (frontend) and NodeJS(backend) are hosted on Amazon Web Services. MongoDB is installed so that the backend may send and receive information from the database.



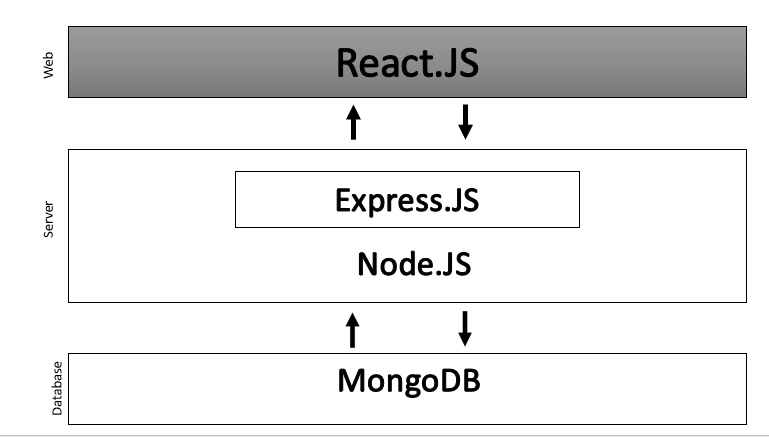
**Figure 3: AZ Booking Architecture diagram**

# Project Plan

This project required a great deal of planning. This enabled me to stay on top of my project. The project's schedule is depicted in the Gantt chart below. In the first semester, I focused on frontend development. In the second semester, I worked on the project's front and back ends. In addition, once we've finished with cloud computing, I'll worked on AWS.

  
**Figure 4: AZ Booking Gantt Chart**

# Project

I have used several cloud computing technologies in my project. For this project, I used MERN Stack, which is shown in the figure below.  
**Figure 5: MERN Stack diagram**

## Frontend

React and NextJS are used in the AZ Booking front end. Both ReactJS and NextJS are component-based JavaScript libraries used in web development. React uses file-based routing to browse around pages, whereas NextJS is more user-friendly and requires less code. In one of my modules on cloud computing, React was used to generate several pages for various components. AZ Booking pages are created using JSX, HTML, CSS, and JavaScript.

## Backend

NodeJS server is used in the AZ Booking backend. The open-source NodeJS runtime environment for JavaScript is made for creating scalable network applications. Using mongoose, this server manages communication with the Mongo database.

## AWS (Amazon Web Services)

On an AWS hosted on EC2 instance running at port 3000, the frontend and the backend. You must enter the IPV4 address with port 3000 at the end in order to access this.

# Frontend

My projects first used ReactJS, but later in the year I learned about NextJS. Both ReactJS and NextJS are JavaScript libraries for web development that are component-based. Because NextJS is user friendly, I switched from NodeJS. As users will be using the frontend, this was done to enhance user experience.

## React

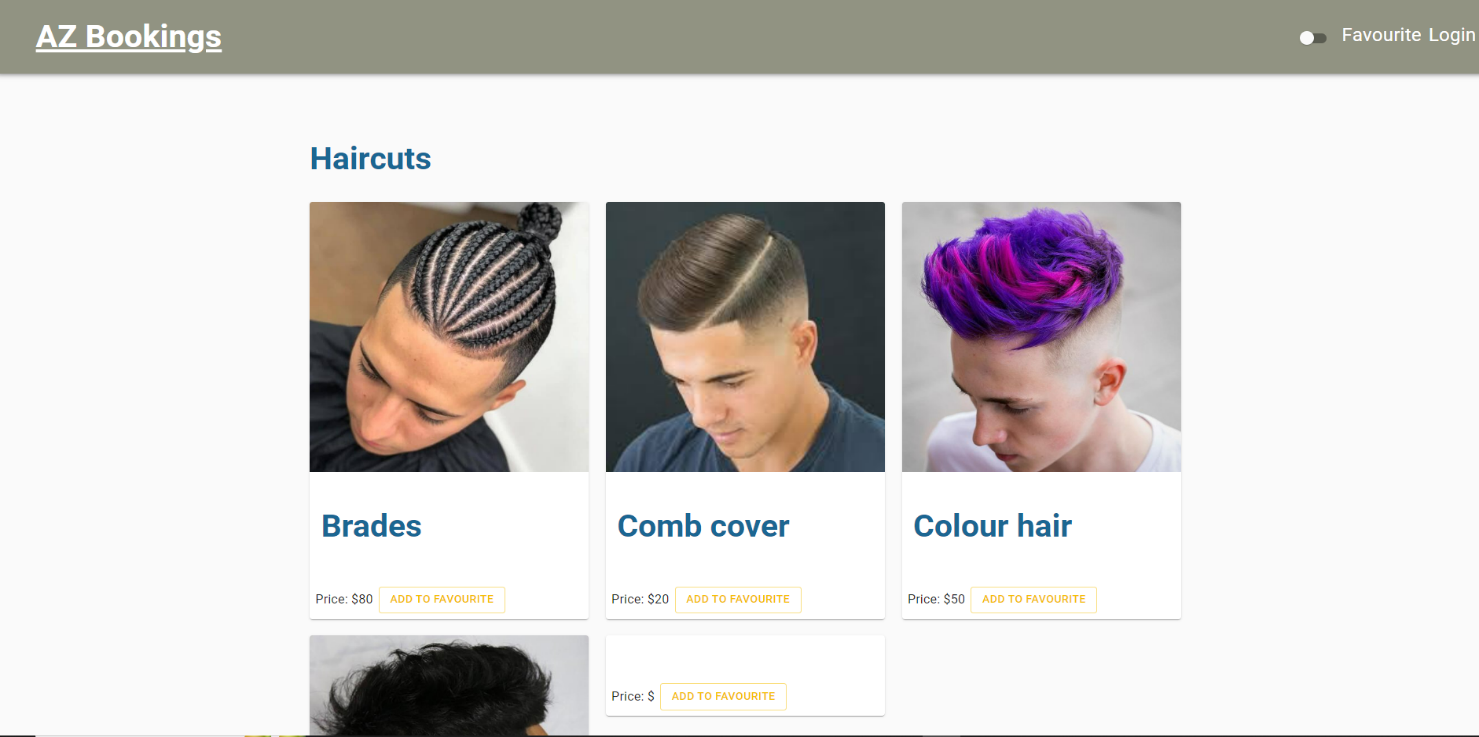
ReactJS was initially used in my projects, however I later discovered NextJS. Both ReactJS and NextJS are component-based JavaScript libraries used in web development. I changed from NodeJS to NextJS since it is more user-friendly. As users will be using the frontend, this was done to enhance user experience.

I was introduced to react in September 2021 in one of my modules cloud computing. In this module we followed YouTube tutorial by Maximillian Schwarzmuller (link of this tutorial is provided in references). I used this tutorial to start developing my AZ Booking while learning React.

Since Mongo DB hasn't yet been covered, everything at this point was static and not connected to a database, and Maximilian used Google Firebase as the backend in this tutorial.

I struggled a lot with this because additional code was needed, and I was trying to position the component correctly on the page using CSS.

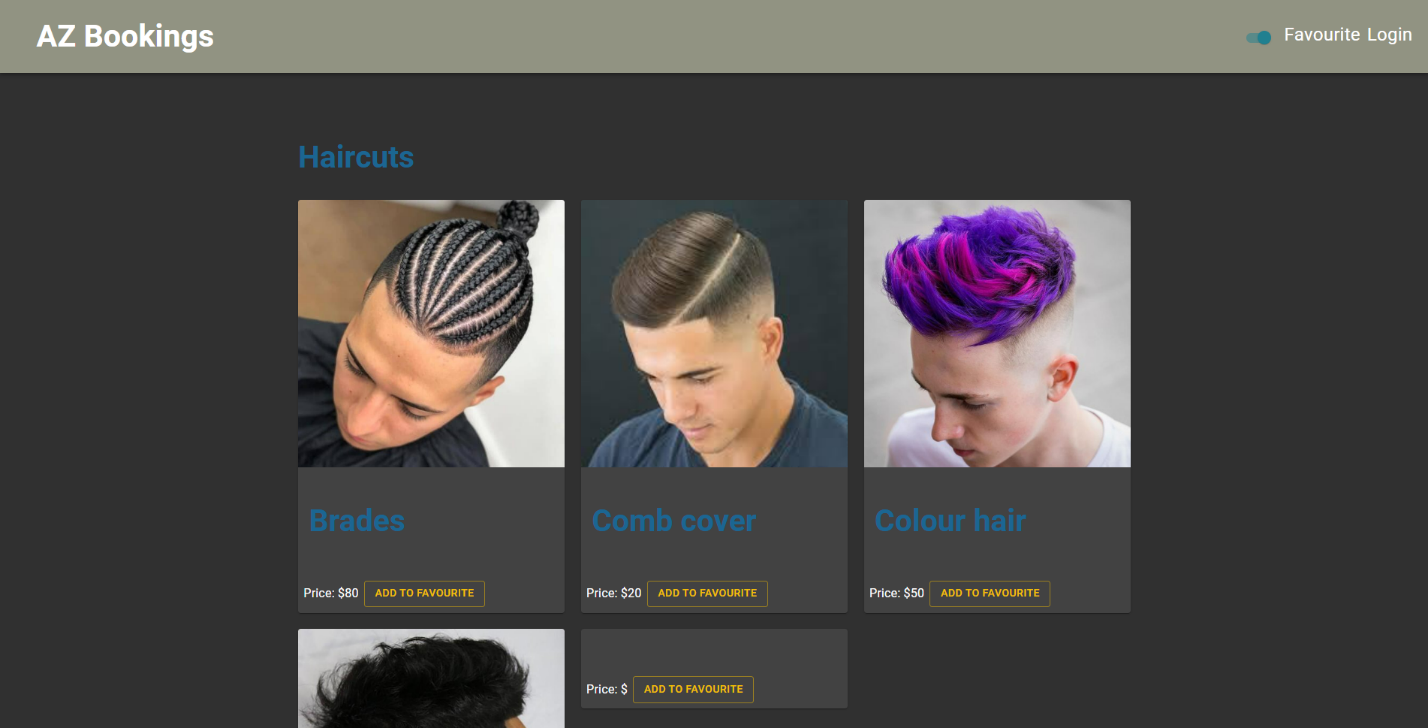
## NextJS

In the development of AZ Booking, January was crucial. We started using NextJS. I discovered that manipulating the data and adding according to my plan to the page was simple. Additionally, I learned about NextUI and MaterialUI. Both "NextUI" and "MaterialUI" are JavaScript/CSS frameworks. With the help of "NextUI" and "MaterialUI" and NextJS, I needed less code and pages, and the ability to discover faults quickly, I was able to make my web app more appealing for the user.

**Figure 6: Main page**

In figure 6, you can see different haircuts options to select. To display these haircuts, I used card component. To place them in the neat display 3 in a row in the centre of the page I used Gird to set items widths in set percentage to fluid the size relative to the parent element and have padding to create the spacing between individual items.

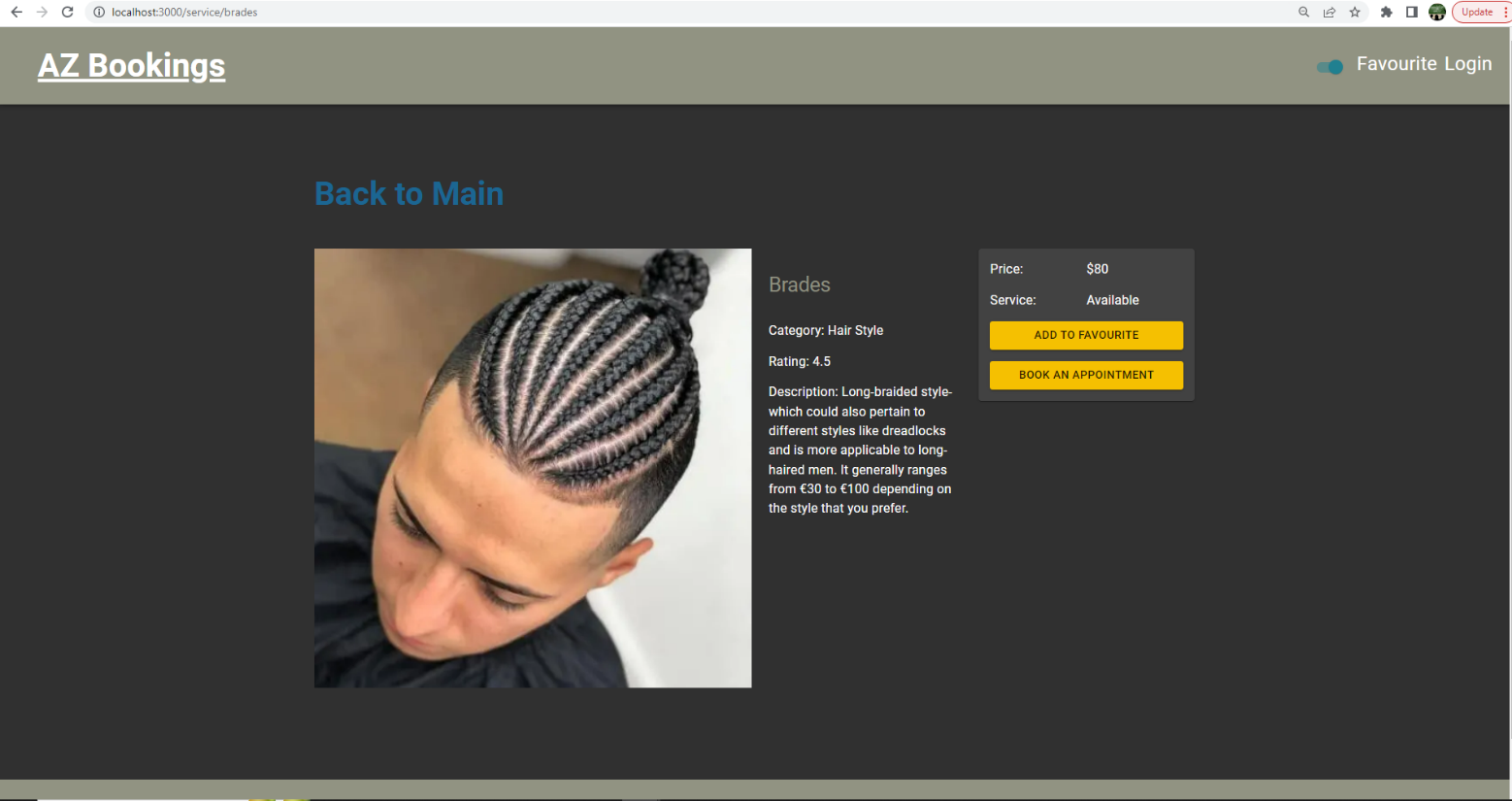
Moreover, the AZ Booking logo is located at the top right corner. A user will be directed to this the main shown in figure 6 if they lick on this logo. By creating a hyperlink, this. This is also for the login and favourite pages; clicking on these will take you to those pages.

Personally, I like to use websites and apps in dark theme. I made the decision to include this feature in my web application so that users could choose their preferred theme. I had to build a react context with the two themes light and dark mode, which can be changed using the toggle button in the upper right corner, in order to accomplish this. The user preference will be saved if the page is refreshed because this change will be kept in the browser's cookies. Figure 7 shows the dark mode, whereas Figure 6 shows the light mode.

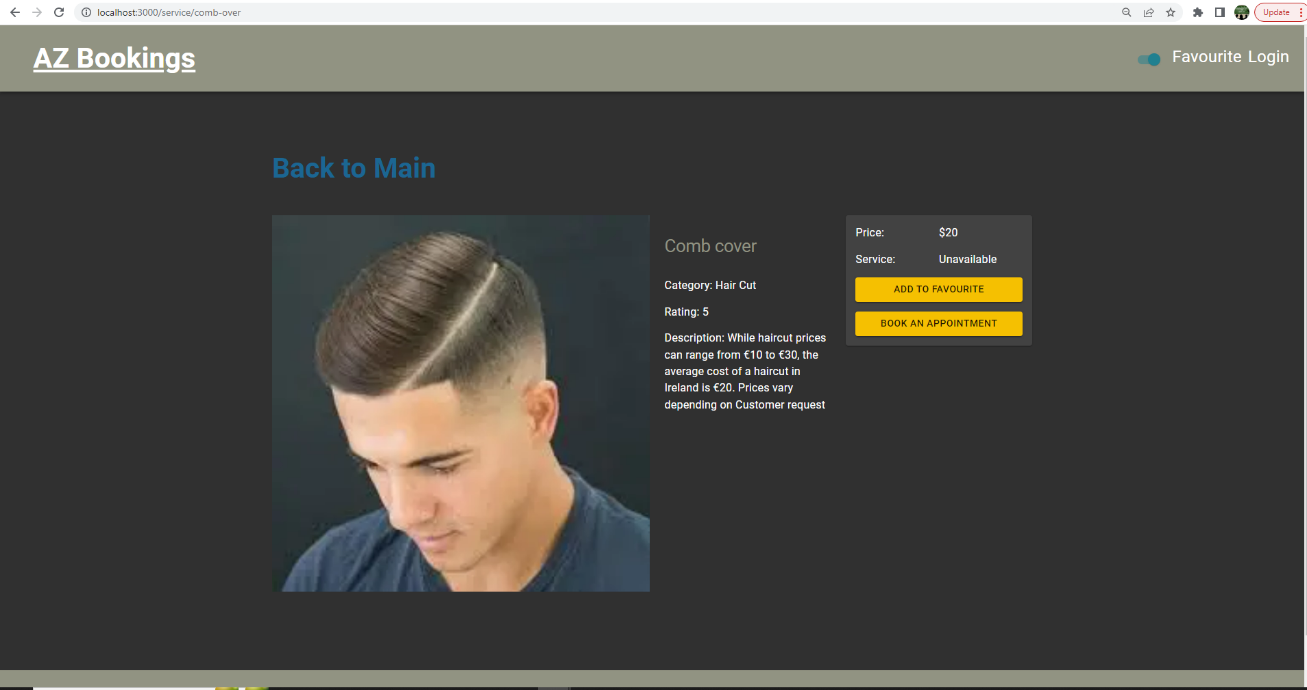
**Figure 7: Main page dark mode**

All of the information shown on the card image, including the name of the haircut and the cost, was fetched from MongoDB. Figure 8 illustrates how to accomplish this by sending a fetch request to the backend.

  
**Figure 8: Fetching data**



**Figure 9: Selecting a haircut**



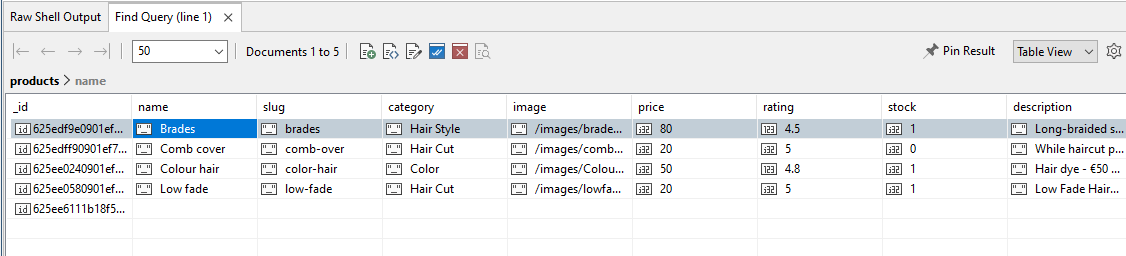
**Figure 10: Selecting a haircut**

Figures 9 and 10 illustrate what the user will see after clicking and choosing a haircut. By sending a request to the backend, the page's image, title, category, description, price, and availability will be pulled from the database.

To make these cards/haircuts link able and pulling the right information for this each product I had to first create the slug of each product. A slug od a product is the URL friendly name of the product. It is the unique identifying part of the web address at the end of the URL show in figure 9 and 10.   
To make a slug for each product or haircut. I had to use NextLink to make the cards on the main page clickable, and I had to set the href to start with service before showing product slug as a dynamic value. This was carried out in the figure 11 index.js page.

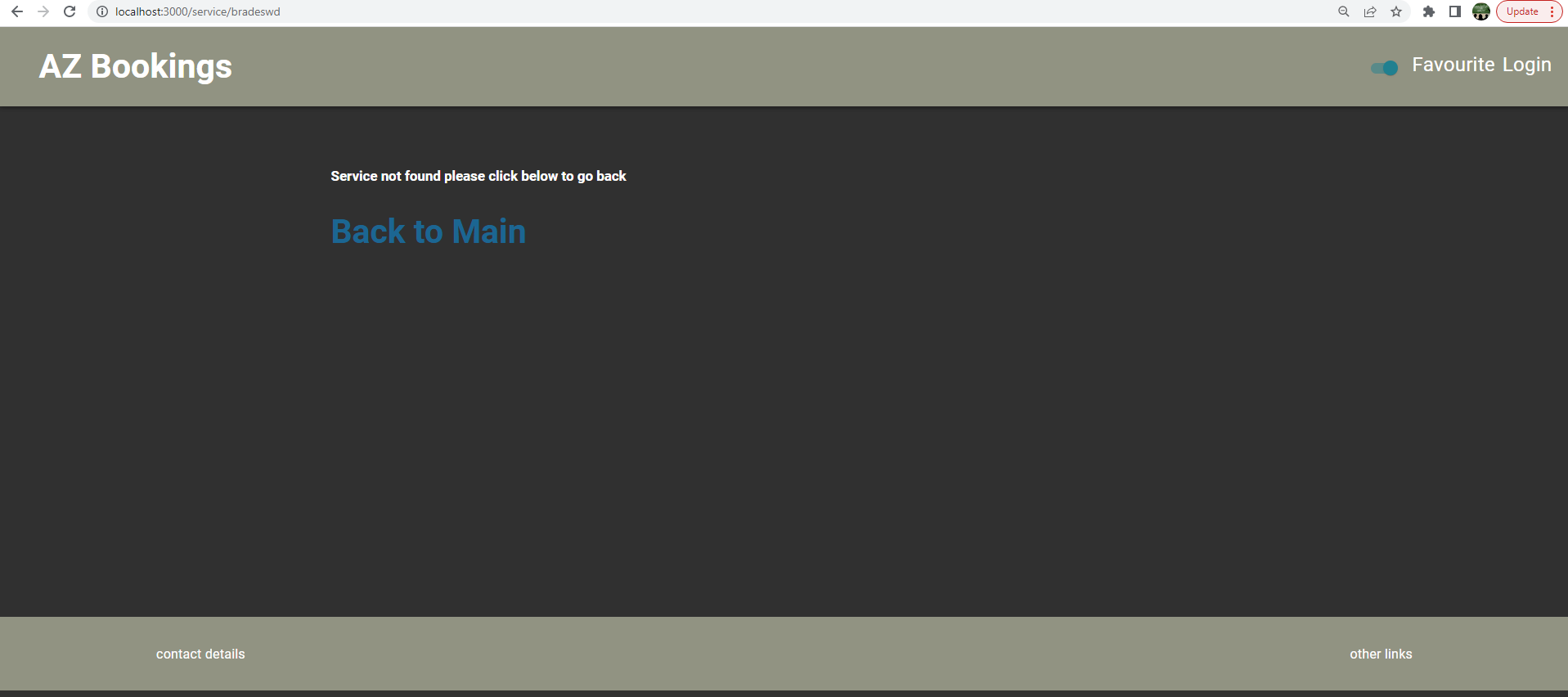
  
**Figure 11: Slug of each product**

I also added a slug field for each product to the database shown in figure 12. As long as the slug is different, this will enable me to have a unique link for each product. It will also let me to fetch and display the correct product information shown in figure 9 and 10.



**Figure 12: Adding slug in database**

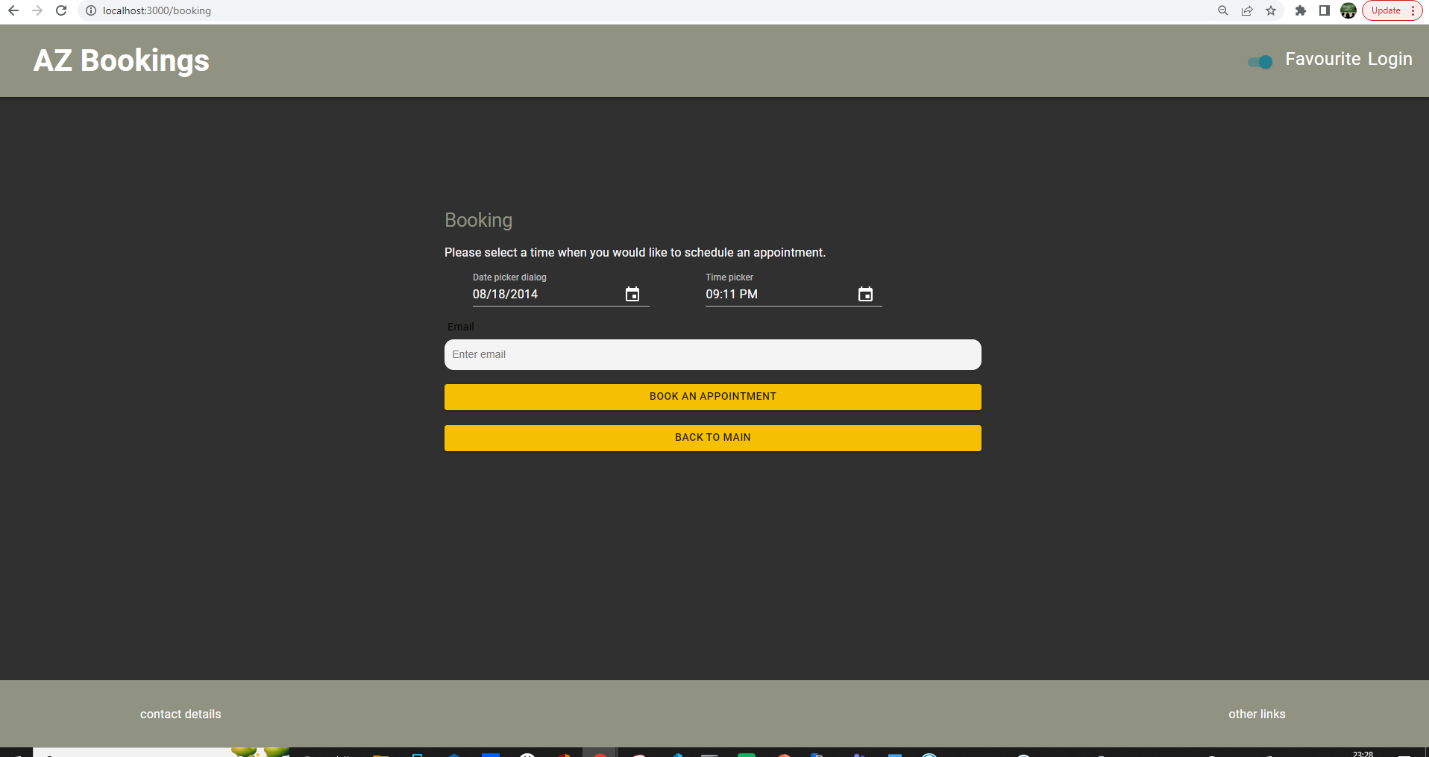
It will display the massage service not found and give the user a link to return to the main page, as shown in figure 13, if a wrong URL is entered in the search field.



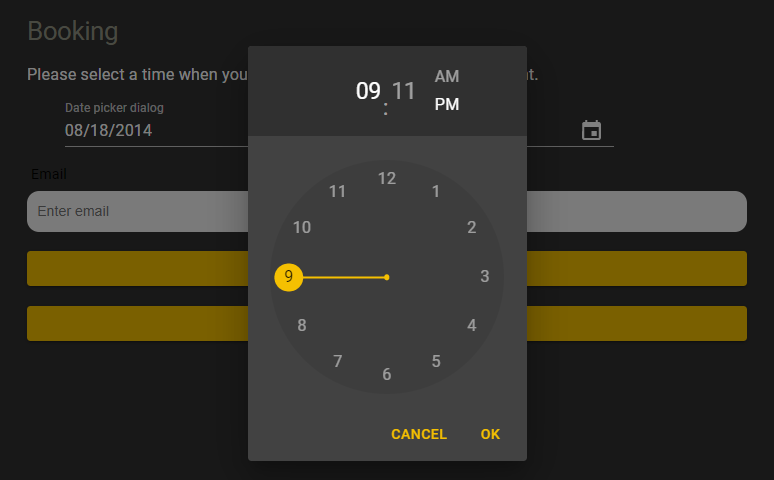
**Figure 13: Service not found**

Right after the user clicks the "Book an appointment" button. They will then arrive at the booking page shown in figure 14. Here, customers can choose the time shown in figure 15, choose the date shown in figure 16, and input a valid email address for validation. If the customer entered a valid email, as illustrated in figure 17, the email field will be green. If an invalid email was entered, the email field will be red.

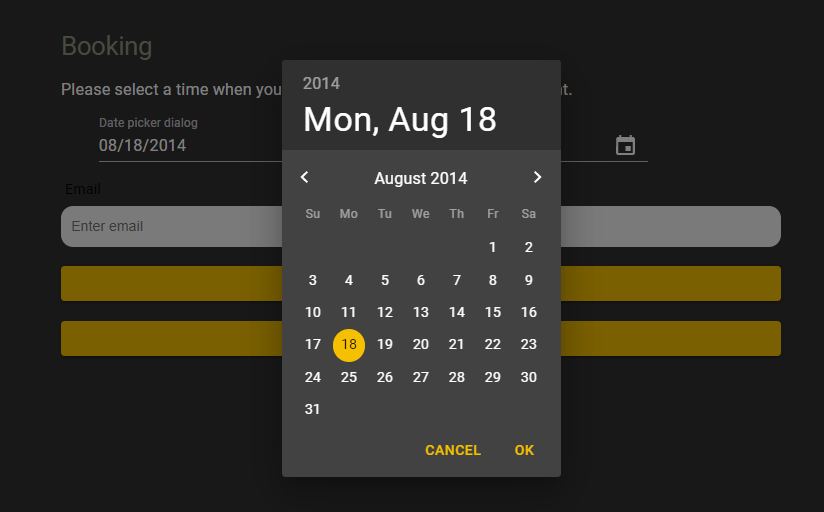
Unfortunately, I was able to preserve data from this page and email the customer to confirm, therefore for the time being, this page is static.

****

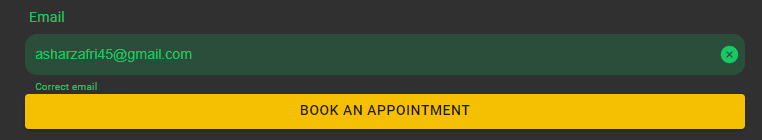
**Figure 14: Booking page**



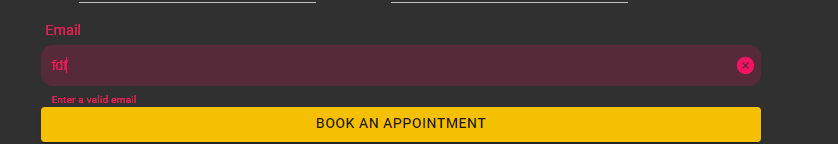
**Figure 15: Select time**

****

**Figure 16: Select date**

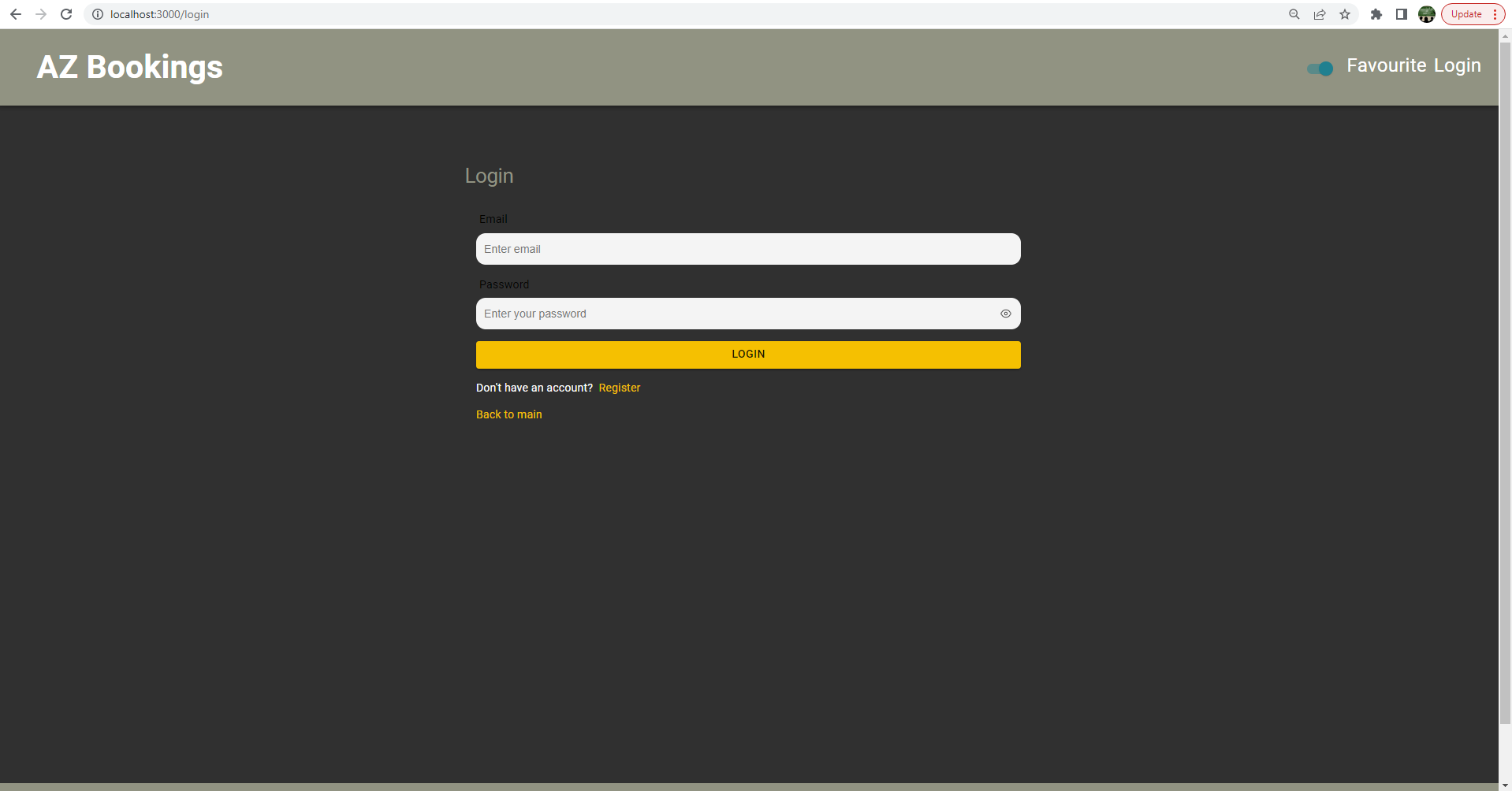
****

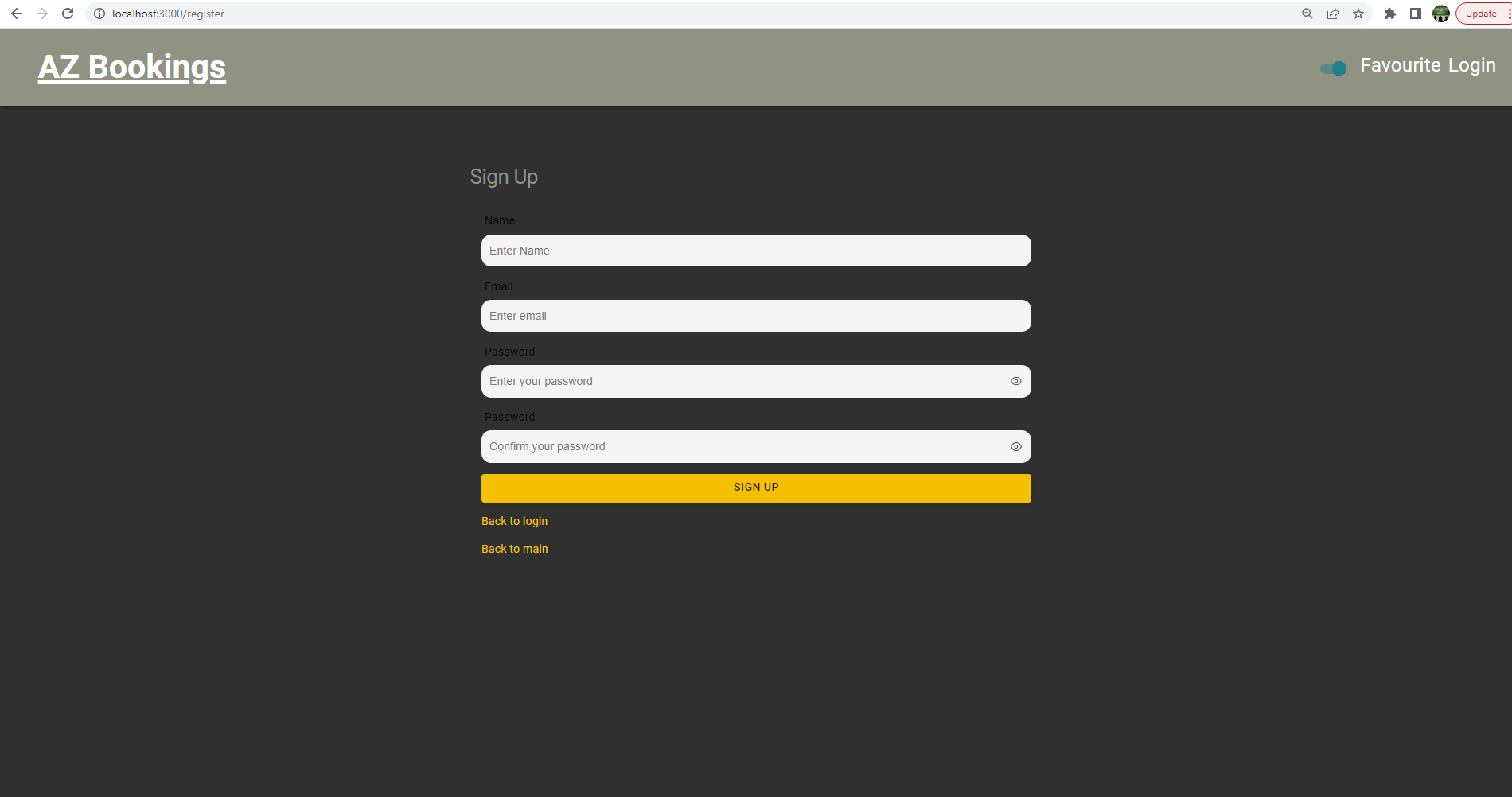
**Figure 17: Valid email**

****

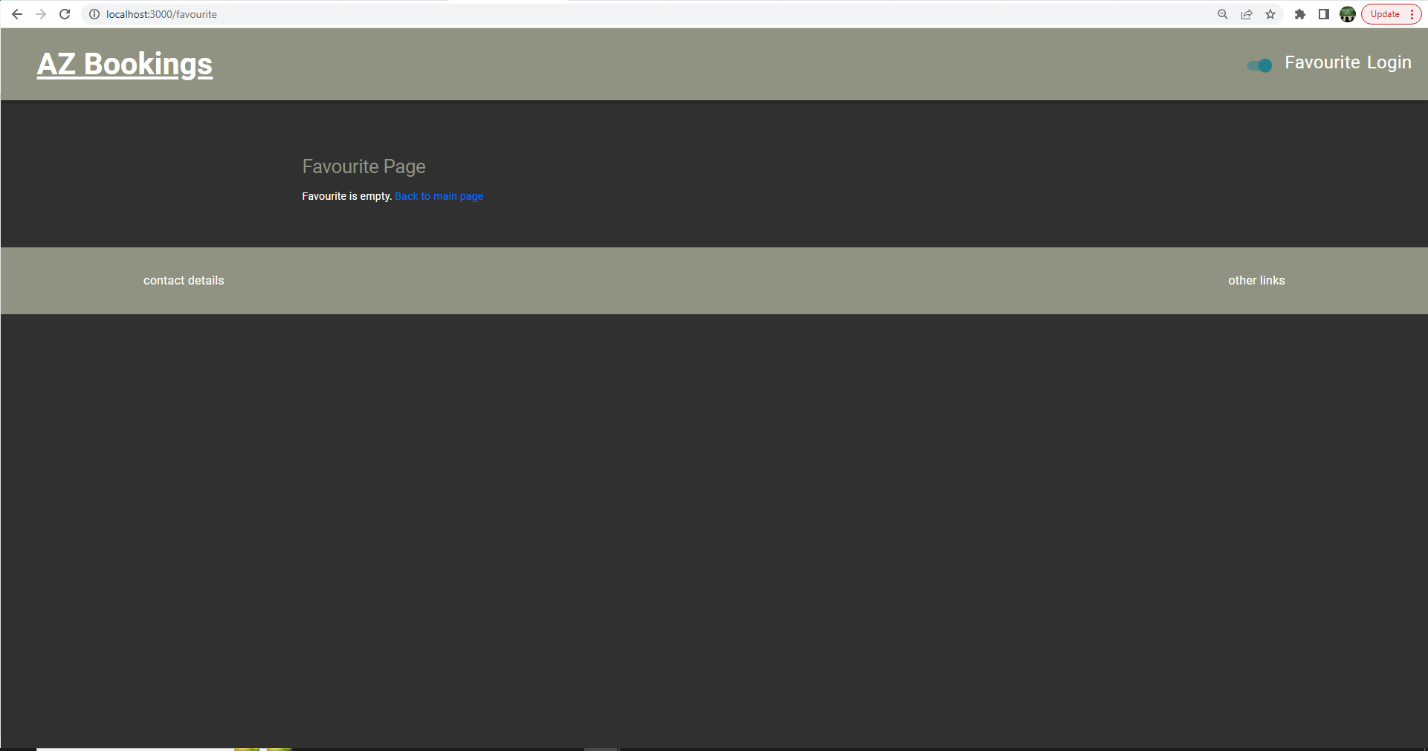
**Figure 18: Invalid email**

A sign-up and login page has been created by me displayed in figures 19 and 20. Due to the fact that I couldn’t work the authentication functionality, these pages were likewise static. You must first visit the login page and select the Register button to access the signup page.

  
**Figure 19: Login page**

  
**Figure 20: sign up page**

The favourite page is shown in Figure 21. Because I wanted to link this page to the database, it is currently presenting an empty page. I wanted the user to adds the haircut to their favourite page, it will stay there until they remove it. How to save information for the favourites page puzzled me. To display the data on the favourites page, I had a work around to save the product/haircut in a global array and read from that global array. This work was not used because adding an item to the favourites page and the refreshing the page. The data will all be gone, and default settings for favourites will show an empty page.



**Figure 21: Favourite page**

# Backend

AZ Booking backend is crucial for this project. Since it is in charge of handling the front end of the request form and any requests made to MongoDB for processing any data calls. For the project's backend, I used NodeJS.

## NodeJS

In January, during my cloud computing module, I learned about NodeJS. We adhered to Maximillian Schwarzmuller's tutorial. I learned how to manipulate databases from this.

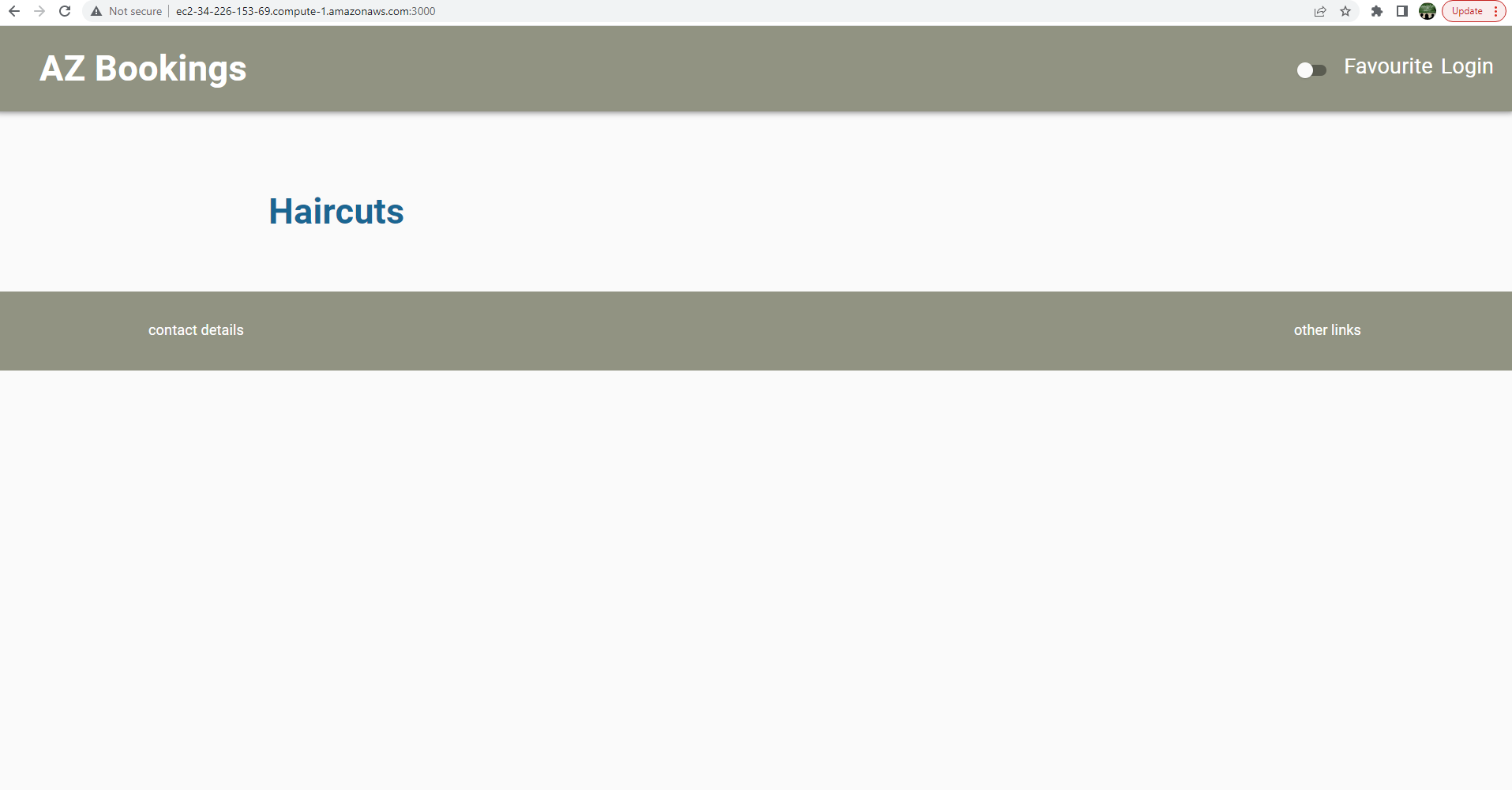
My backend is mostly inspired by Maximillian Schwarzmuller's tutorial. Allowing me to communicate with MongoDB. Using backend, I was able to read and display data such as image, tittle, description, price, and availability on to front end.

# AWS (Amazon Web Services)

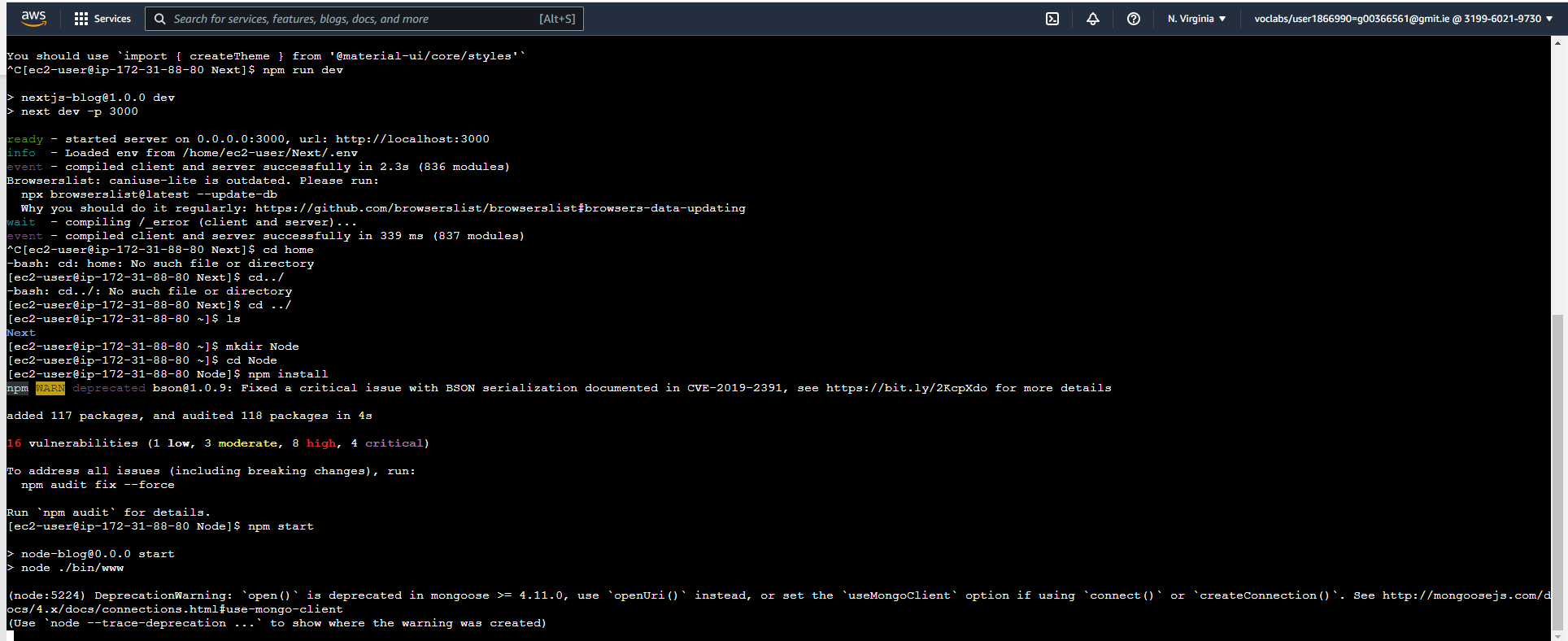
I was also introduced to AWS in my cloud computing module. Use EC2 and create new t2.micro instance using amazon Linux. Next, I had to install NPM, Node, mongo on AWS. Furthermore, I had to create to folders Next for the frontend and Node for the backend. Then transfer the project using WinSCP shown in figure 26. Then I had to edit inbound rule in EC2 security group. Lastly, I had to have the custom TCP at port 3000 anybody on IP4.

Both frontend and backend are hosted on AWS shown in figure 22. I ran into a problem while doing this since npm could not be located, as seen in figure 25. My research led me to [https://exerror.com/node-lib64-libm-so-6-version-glibc 2-27-not-found-required-by-node/](https://exerror.com/node-lib64-libm-so-6-version-glibc%202-27-not-found-required-by-node/), where I discovered the solution. To solve this problem, install Nodejs 16 Simply run this command on the AWS nvm install 16 servers. As I have to install it each time, I try to run the project on AWS, this is not the long-term answer and I will discuss this further with my lecturer to get a permanent solution.

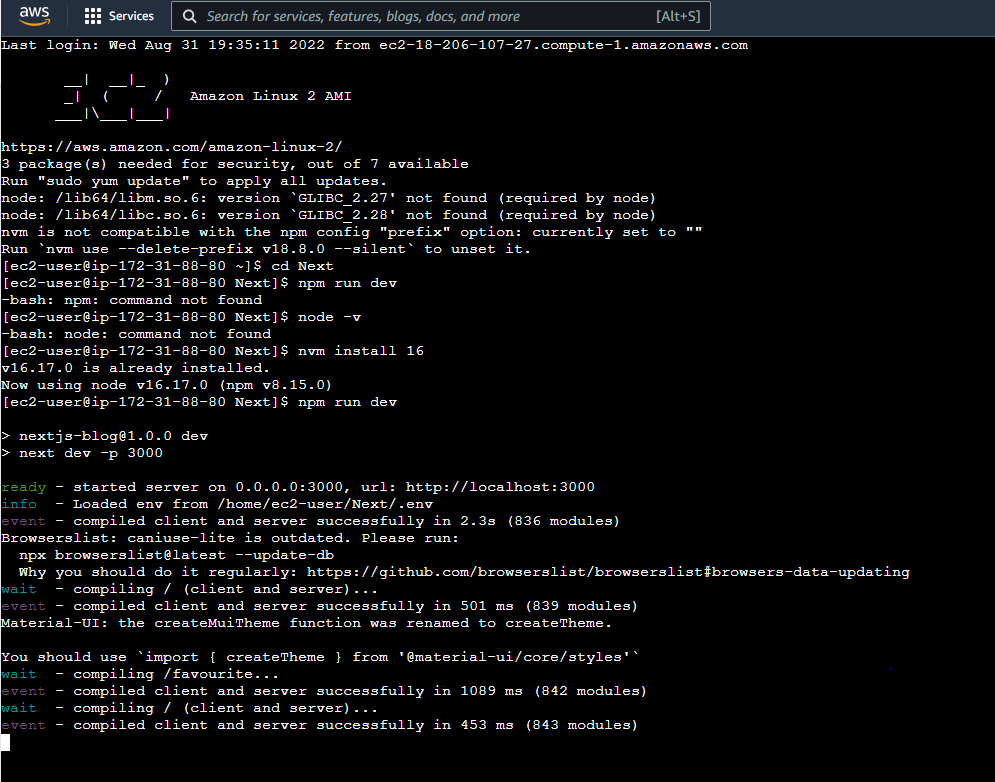
I wasn’t able to display any data on AWS as all the data was erased when moving the project and I did that the functionality in my web to add new data on the app.



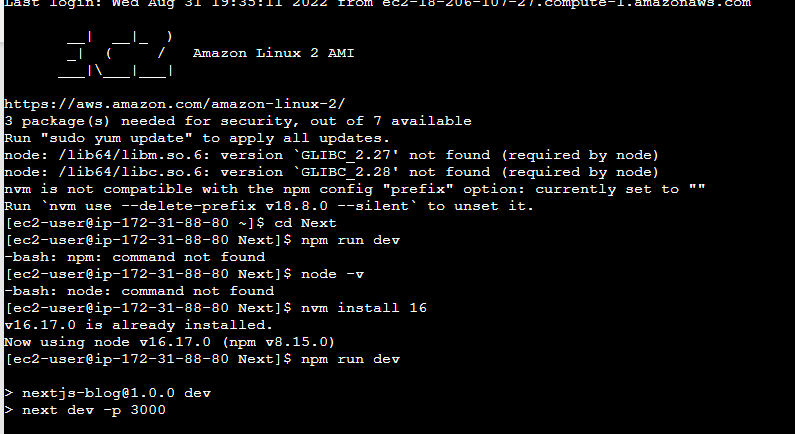
**Figure 22: Frontend and backend on AWS**

****

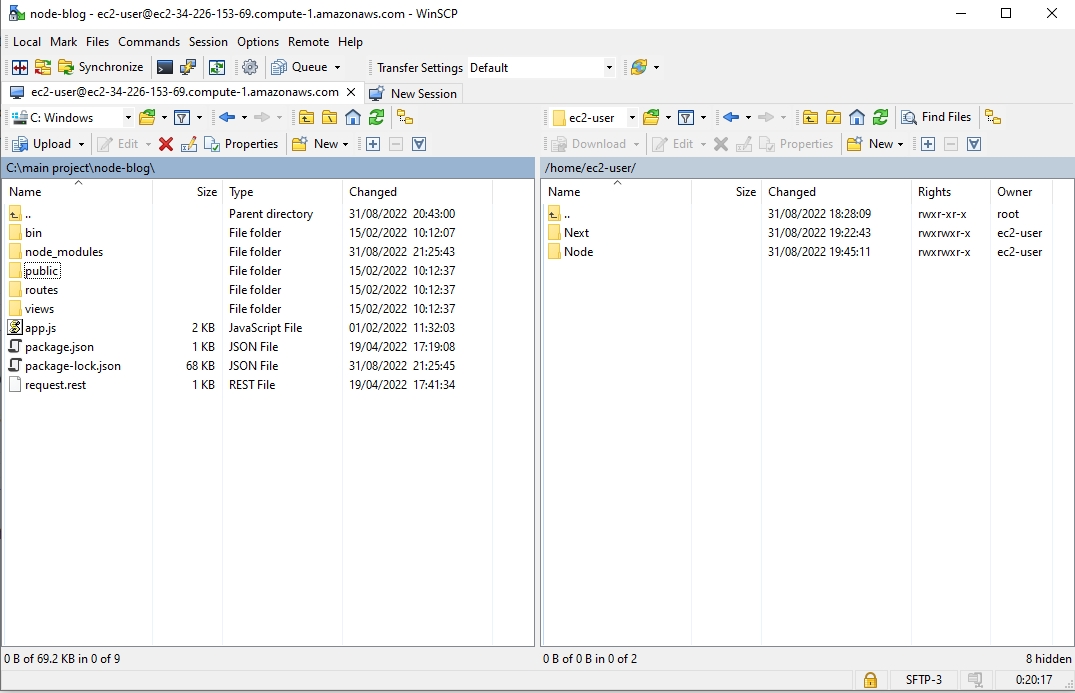
**Figure 23: Backend on AWS**

****

**Figure 24: Frontend on AWS**



**Figure 25: installing Nodejs 16**



**Figure 26: WinSCP transferring project to AWS**

# Challenges

I faced a variety of problems while working on this project. This was my first time working on a full stack project. I needed to learn how to use JavaScript and react. My development was slowed because of this.

In the first semester, my laptop had stop functioning. I had lost all my work on that computer and had to start again with my project.

In addition, in the second semester, I decided to switch from React to NextJS for my project. When I switched from react to NextJS for my pages and components. I came across quite a few errors. After trying to fix the problem for a while, I was suggested to move the files to a new NextJS project. This resolved the problem.

Moreover, I was attempting to connect Robo 3T 1.2 to MongoDB. I was unable to establish a connection to MongoDB. This was frustrating because I was able to add data to MongoDB. I was unable to remove sample data from the database, though. After doing some investigation, I discovered that many other people were experiencing the same problem, and that the remedy was to download Robo 3T 1.3. I was able to miniplate the data and delete the unwanted data because of this solution.

I had a problem when attempting to book an AZ on AWS, and the bash command could not be located. I had to install NodeJS 16 to handle this, shown in figure 25.

Last but not least, I found it difficult to manipulate database dates because I couldn't do user authentication or add favorites.

# Conclusion

To sum up, while working on this project, I was able to learn about various aspects of cloud computing. To make this platform, I learnt how to use the systems and software listed above. I'd like to continue working on this project and use it to assist businesses in running more efficiently.

# Reference

1. [React Crash Course for Beginners 2021 - Learn ReactJS from Scratch in this 100% Free Tutorial!](https://www.youtube.com/watch?v=Dorf8i6lCuk&t=5044s)  Used to start project page navigation and page layout with CSS
2. <https://pretagteam.com/question/how-can-i-create-a-global-array-in-react-when-i-do-separate-operations-for-each-index-i-want-them-to-save-them-in-a-single-array>  Will be used to learn how to store multiple values in a single array
3. <https://nextui.org/> used for adding components
4. <https://mui.com/> used for adding components
5. <https://dev.to/yezyilomo/you-can-definitely-use-global-variables-to-manage-global-state-in-react-17l3> Use to learn about global object
6. <https://dev.to/yezyilomo/you-can-definitely-use-global-variables-to-manage-global-state-in-react-17l3> Use to learn about global object
7. <https://tinloof.com/blog/how-to-build-an-auto-play-slideshow-with-react/>  For a slide show that I can use to  display picture or adds
8. <https://www.w3schools.com/w3css/w3css_modal.asp> How to set up modal
9. <https://www.geeksforgeeks.org/how-to-create-a-simple-responsive-footer-in-react-js/> Used to help set a footer
10. <https://codesource.io/how-to-use-this-props-history-push-on-your-react-project/> How to use push function
11. <https://www.w3schools.com/> Will be used to learn, test and research css and html
12. <https://www.npmjs.com/package/@material/textfield> To be able to use the text field
13. <https://backup.ninja/news/database-backups-101-file-and-database-compression> Method of reducing file size in data bases
14. <https://wisebarber.com/types-of-haircuts-for-men/> colour hair pic
15. styleinterest.com/low-fade-hairstyles/  Fade image
16. <https://www.newstore88.top/products.aspx?cname=taper+cornrows&cid=58>  Brades image
17. <https://pretagteam.com/question/how-can-i-create-a-global-array-in-react-when-i-do-separate-operations-for-each-index-i-want-them-to-save-them-in-a-single-array>  Will be used to learn how to store multiple values in a single array
18. [https://exerror.com/node-lib64-libm-so-6-version-glibc 2-27-not-found-required-by-node/](https://exerror.com/node-lib64-libm-so-6-version-glibc%202-27-not-found-required-by-node/) nvm install 16