Recipe Database - Final Project Client-side Front-End Web Development

By:

Stanislaus Steven Kurniawan / 12202020

Wilbert Dietriech Sutardja / 12202031

STUDENT PROJECT

CLIENT-SIDE FRONT-END WEB DEVELOPMENT

Bachelor of Information Technology Faculty of Engineering & Information Technology

Introduction

The objective of this project is to create a website that showcases different recipes by using react.js, firebase and mealdb. Using react as a framework to create the website, using firebase to enable user registration and log in through email, google and to store user search history and finally by utilising mealdb API to showcase the different recipes on our website.

Job Desk

Our team consists of 2 members: Steven and Wilbert. Steven was in charge of the overall react framework, landing page and recipe catalogue. Wilbert was in charge of integrating firebase for user authentication and registration, also the creation of log in, sign up and recipe page. Both members did the search history function with Wilbert doing the setting up and configuration of the files and components whilst Steven did the integration onto the website.

Problems

The first problem encountered was figuring out the implementation of the .env file onto the firebase configuration file. We used the wrong naming format for the .env file as it had to be REACT_APP_… . We overcame this by looking at solutions online until stumbling upon a stackoverflow forum that had the suggestion of the naming convention.

The next problem we encountered was figuring out error handling for user sign up and login. Initially we did not know of the inbuilt error messages that came with firebase. Hence, we started making our own error handling. We overcame this by checking the console and seeing the different error messages that were in the log and creating our error handling through the firebase errors that were found inside the log.

In the recipe catalogue, implementing from figma proved to be a roadblock due to the larger sizes in figma which proved to be too big for the actual website. To overcome this problem, we had to scale it down using the transform scale but with that the catalogue page appeared way below than intended. Originally, it was solved using transform translate y negative a few pixels. However, that came with its own problems, when searching for prompts with little results it shifts the catalogue page up by a huge margin. Finally, the solution was to use -20% instead.

One problem we did not expect to encounter was the merging/overwriting of files when using github across 2 different users. Due to our inexperience in using github there were times when we were coding on the same page where this at times led to one’s code to overwrite the others. Thus, what we did to solve this problem was through clear communication and the spread of work. If one was needing to code on a certain page the other would work on something as to not mess the other’s work. For example, we both needed to use the App.js and index.css there was once when we were both using App.js and index.css and committed at relatively the same time, due to our inexperience we did not know how to properly merge our files causing a lot of the content to be overwritten. We solved this again by taking turns and commenting each other’s work in App.js and when using index.css we decided to use separate css files when working on it and finally compiling all of it into index.css.

The last and biggest problem we encountered was the configuration of the firestore. It was a lot more complicated than it seemed at first glance. The first problem we encountered was setting up the database itself, setting up the rules were not really clear at first glance and we solved this problem by parsing through the documentation and finding out the rule that we were looking for. The next problem we ran into was the setting up of all the different files needed to get this to work as we only solved it through brute force of trial and error until we finally got the end product that we were looking for this was done through multitude of ways but especially through checking the logs and pushing the error through to the logs.

Conclusion

The project was an eye opener to the world of web development and how simple things can take a lot of complicated turns to achieve such simple functionalities. This project also taught us the importance of strong and concise communication as it had helped us in a multitude of problems where communication was the key to solving a lot of our problems. This project also eoncompasses what we learned across this semester and more where it allowed us to explore the usage of API and web development in general in a lot more depth. Overall, this project was a success for us as we believe we achieved what we have set for ourselves at the start of this target.