**Deli / Chicken Roll Web App Project**

**CIARAN ROCHE - G00376934**

**MILOSZ** **MILEWSKI - G00376658**

**Table of Contents**

1. **Introduction**
2. **System Requirements**
3. **Technology used and why**
4. **Design Methodology Applied**
5. **Features of the Implementation**
6. **Limitations and Known Bugs**
7. **Recommendations for Future Development**
8. **Conclusion**

**Introduction:**

For our project, we decided that we would implement the design of a chicken roll or deli app. We decided on a react application which is connected to an SQL database with a number of different tables. The app will allow the user to create an account, create a review of a deli, search reviews, search for delis, rate and like other reviews and delis. In this document we will talk about the system requirements, technologies used, design methodologies, features, limitations and bugs and finally recommendations for future development.

Our main objectives were as follows:

* Login system based on information stored in SQL
* Submit reviews
* Rate delis
* Search for delis
* Improve our teamworking skills
* Practise project management using different tools such as Jira

**System Requirements:**

As the project is still in early phases and not put online just yet. In order to run the project, you will need the following:

* Open a command prompt by typing cmd
* Clone the repository to your pc or desktop – git clone and the URL
* Install the required modules. You do this by typing npm install
* Then type npm start
* This is needed to be done for both the backend and the front end in order for it to run correctly
* The latest version of node needs to be installed along with an editor such as visual studio code

Once the website is made available online you will not need to follow the steps above as the user will be able to interact with it like any other website on the internet.

We will take you through the user’s perspective of the website in a heading below called features of the implementation.

**Technology Used and Why:**

Below is a list of technologies used. We will give a brief explanation as to why we chose each.

1. **JavaScript**
2. **React**

We decided to use a react application for the project. There were a number of reasons for choosing this. As part of semester one we studied react and both of us have a great interest in it. React seemed the most appealing route to take. Another reason for choosing this was due to our knowledge React. We both feel confident in our abilities to be able to use it. However, there is still a huge learning scope for us. There is a lot of new things we learned when working on the app. This made the application both rewarding and challenging.

1. **Tailwind CSS (Later changed out for Bootstrap CSS)**

We used tailwind CSS as a way of designing the web application. We chose this as we found there was a lot of unique aspects to CSS that were superior to other designs. In terms of front end, it made the styling sleeker and very easy to use. It was very easy to implement as well.

1. **MySQL**

We decided that the best database management system is MySQL. We were originally looking at MongoDB but with all the different tables and relationships between said tables, MySQL was clearly the better option for us. We are also learning MySQL for a number of years, so it was very easy to implement and design.

1. **Jira**

We used Jira as our project management tool. We chose this as we felt it was the most useable and flexible tool available to us. We could set up sprints, we had a roadmap and backlog. It was also very useful for assigning people tasks. This tool is very useful as it was a guide as to where we are and where we should be on the project.

**Design Methodology Applied:**

We started our project by having a meeting and setting out a timeline for the project. We did this by setting up a Jira board. We first set the project into 4 stages:

* Front end
* Database
* Backend
* Documentation

We used these as a basis for our sprints. We then decided on a design or possible design that we would pursue. We did this by creating some freehand designs and ultimately creating a wireframe. We decided to make 3 GitHub repositories. Front end, Back end and documentation repositories. This made it easier to separate out the work. The reasoning for separating the front end and back end was so that there was no clash between the front end and back end. Also, it was so that we could separate them in order to make the website go live.

*Sprint 1:*

We then proceeded to set up the front end. This was done by creating a react app. All the necessary modules were installed and the skeleton for the front end was ready. Setting up and designing the front end was our first sprint. The front-end sprint took approximately 2 to 3 weeks.

In this time, we created the design for:

* Login Page
* Register Page
* Home Page
* Map Page
* Submit a review Page

We collaborated with each other to get the design of the pages up and running. We discussed different designs and decided on the best one to use. Each of us took a page each to design and work over a few days in the week before our scheduled weekly meeting. The design of the front end was submitted to the GitHub repository for the front end.

*Sprint 2:*

Our second sprint was all based around the database. During this time, we designed the Entity Relationship Diagram and then set up the scripts for the SQL database. We both did this over meetings and collaborated to decide on the best design for the SQL tables. This sprint took in total around a week.

*Sprint 3:*

Our third sprint was for the Backend design of the project. This was probably the most complex part of the project with us learning a few new things as we went along. This was essentially were we set up the connection to the front end. We set up the skeleton for the project and then collaborated with each other and took different areas to proceed with. This was to link up the entire project and we could see real progress at this point.

*Sprint 4:*

Our final sprint was where we made this document and tied up all the loose ends of the project. This was essentially where the bulk of the work was done but it was now time to clean it up, comment the code more efficiently and see the final submission of our project.

**Features of the Implementation:**

**Limitations and Known Bugs:**

**Testing Plans:**

**Recommendations for Future Development:**

There are a number of different ways on which we can enhance the project as follows:

On the front end of the application, we can always make the design more modern, sleek and eye catching for the user. At the moment we are very happy with the design, however, in future we may decide that we would like to change the colour scheme. This is easily done but for the moment and sake of the project it was not necessary. We can also add new features in the future. We will undoubtedly have new ideas that we would like to add or pursue and would be a fun way to pass time.

On the database end we can always add more tables. This of course will depend on us adding new features to the website.

We can always look at expanding the website. In the future we don’t necessarily have to limit it to chicken rolls or delis. It is an area that has a wide variety and so many different types of cuisines out there. For example, we can always adopt an approach at looking at restaurants and rating their meals etc.

In terms of software, there is always a possibility for upgrading our code per say. There may be an efficient new way at doing things that we discover and may decide to adopt that in to our project.

**Conclusions - what you learned from this project:**