4/10/2018

Recipe Hub

Project Proposal

# Gavin Hanna

# Higher Diploma in Science in Web Technologies

# Specialisation: Web Technology

## 17164974

## X17164974@student.ncirl.ie

# Table of Contents

|  |  |
| --- | --- |
| 2 | Objectives |
| 3 | Background |
| 4 | Technical Approach |
| 5 | Project Plan |
| 6 | Technical Details |
| 7 | Evaluation |
| 8 | Reference List |

# Objective

To create a social network web application based around creating and sharing cooking recipes. The application should be capable of being saved to a mobile phone as a PWA (Progressive Web App, a web application which can act as a native mobile app), with some offline capabilities: achieved by using a service worker with either an offline.html page being served when a connection is unavailable, or information being served from the browsers indexedDB API in the absence of a connection.

In order to create a smooth, fast and responsive user experience, the application will be created using the MERN stack. The MERN stack incorporates the following technologies:

* MongoDB – A NoSQL database based on JavaScript
* Express – A Node library designed for creating web applications and APIs
* React – A client side library commonly used for creating Single Page Applications
* Node – The JavaScript V8 engine taken from the browser and placed on the server

The application will be separated into two major components:

* The server – A RESTful API
* The client – A React application which will communicate with the server

The application should allow:

Users to be able to register and log in the application, create edit and delete their own recipes.

Users to be able to follow their favourite recipe creators and comment on their own and other users recipes.

Users to be able to save their favourite recipes to easily review.

Users to “like” other users’ recipes.

# Background

## Research

Recipe based websites are an extremely popular corner of the internet, with the top 15 recipe based websites ranging from 2.25 million unique monthly visitors for KraftRecipes, to 25 million visitors to AllRecipes (**ebizimba.com, 2018**).

The word “food” has an average monthly search count of 823,000, with “recipes” having 450,000 and “recipe” with 90,500 (**searchvolume.io, 2018**). In Google Trends, the search term “recipe” never dips below 50% “Interest over time” (**Google Trends, 2018**).

The popularity of recipe based searches on the internet, and the overwhelming popularity of social networks (Facebook alone has over 2,196,000,000 users (**statista.com, 2018**)), lead naturally to the combination of the two, of which there are many such websites currently available online. Two of the most popular are bakespace.com and cucumbertown.com. While these are technically social networks, they either focus more on simply being a recipe search engine, or acting as a blogging platform focused on recipes.

I believe there is a space for a fully realised recipe based social media web application in which people can follow their friends and share there particular cooking styles while interacting with and sampling/critiquing styles of their loved ones.

## Conclusion

This knowledge, coupled with the statistics that suggest that mobile browsing has now taken over desktop browsing (**Eric Enge at stonetemple.com, 2018**), suggests there is a possibility for a successful recipe based social media application. I believe by building the application with it also being a Progressive Web App so that users can easily use the sites functionality on their phones in the same way as a native app, and also on the browser, a large userbase could be obtained.

# Technical Approach

## Research

As in Background, research was conducted on internet search query amounts to determine interest in searching and viewing recipes online, along with research into current social networks and/or blog style websites focused on recipes and the sharing thereof.

Research was also conducted on websites that focus entirely on recipes and providing recipe search functionality, for example **bbcgoodfood.com/recipes**, **easyfood.ie/recipes** and **goodfoodireland.ie/recipes** among others.

## Requirements Capture

From this research the required recipe related data which would need to be stored in the database was formulated, along with potential recipe page layout design and structure. The Recipe database model would need a minimum of the following rows:

* Id (primary key)
* User\_id (foreign key)
* Name
* Cooking time
* Preperation time
* Serves (integer representing amount of people meal should serve)
* Ingredients (stored as an array of strings in MongoDB)
* Method (stored as an array of strings in MongoDB)
* Likes (array of user IDs)

## Implementation

As mentioned in the Objective, the application will be split into two major components: the server and the client.

### The Server

Built using node.js using the express framework to create a RESTful API. The server will store data in a MongoDB instance hosted at **mlab.com**. The server will interact with the database using the Mongoose ORM (Object Relational Mapper). Authentication will be handled with JSON Web Tokens using the jwtwebtoken npm package.

### The Client

Built using React, starting with boilerplate created with the create-react-app npm package. The application will implement Redux to handle app-wide state management.

# Project Plan

Please see attached MS Project Gantt chart for detailed view

# Technical Details

As mentioned in previous sections, the application will be split up in two separate yet communicating applications, the frontend client application and backend server application. The “stack” that was chosen was this application was the MEAN stack.

## Server

The backend of the application will be built using the Express.js web application framework for node. Express will be used to develop a RESTful API with endpoints for create, read, update and delete methods on objects stored in the database.

The database will be a MongoDB instance which will be hosted at mlabs.com. The Object Relational Mapper Mongoose will be used by the app to communicate with the database.

The following is a list of npm modules used by the server, and their purpose:

|  |  |
| --- | --- |
|  |  |
| body-parser | For parsing form data submitted from the client |
| express | The framework upon which the server application is built |
| jsonwebtoken | Used for handling JSON Web Tokens which will be used for authentication |
| bcryptjs | Used to encrypt and decrypt user passwords using hashes and salts |
| mongoose | An ORM to handle communication with the MongoDB instance |
| passport | Handles authentication and authorisation |
| passport-jwt | A passport authentication strategy using JSON Web Tokens |
| validator | A library of string validators and sanitizers, used for form validation |

## Client

The frontend will be implemented using React, with redux to manage application state.

|  |  |
| --- | --- |
|  |  |
| axios | Used for making requests to the backend |
| classnames | Create conditional CSS class names inside React |
| Jwt-decode | Used to extract logged in user info from JSON Web Token |
| moment | Format dates to be human readable |
| react | Library for building interactive client applications |
| react-router-dom | Routing on the client side |
| redux | Application wide state management |

# Evaluation

During the production of the backend of the website, continuous testing will take place using **Postman**. Postman allows developers to test APIs by sending and receiving requests using HTTP verbs and any header or other data they would like to send. This software will therefore be used during the entire development of the server.

During each endpoints creation, requests will be sent to test the endpoint. For example:

If we are creating the user register endpoint, POST data will be sent to the /api/users/register URL for testing.

Once the entire backend has been completed a further suite of tests will confirm whether each endpoint is functioning.

During the production of the frontend of the website, initially data will be requested from the server inside a React component. This will be used for testing the communication between frontend and backend of the application. Once testing has been completed, requests will then be made from inside an action, which will be dispatched to a reducer and added to the application state.

Once this initial testing phase has been completed the system can then be deployed to a server for further testing before release.

# References

Ebizmba 2018, accessed 04/10/2018,   
<http://www.ebizmba.com/articles/recipe-websites>

Searchvolume.io, accessed 04/10/2018,  
<https://searchvolume.io/>

Google Trends 2018, accessed 04/10/2018,  
<<https://trends.google.com/trends/explore?q-recipe>>

BBC Good Food Recipes, accessed 04/10/2018,  
<<https://www.bbcgoodfood.com/recipes>>

Eric Enge, April 27th 2018, accessed 04/10/2018,  
<<https://www.stonetemple.com/mobile-vs-desktop-usage-study/>>

Easy Food, accessed 04/10/2018,  
<<http://easyfood.ie/home/recipe/>>

Good Food Ireland, accessed 04/10/2018,  
<<https://www.goodfoodireland.ie/recipes>>

mLab, accessed 04/10/2018,  
<<https://mlab.com>>

Postman Application, accessed 04/10/2018,  
<<https://www.getpostman.com/>>