Cooking to Accommodate Food Allergies Doesn’t Have to be Expensive

1Taylor Leung, 1Chen-Hsiang Yu

1Department of Computer Science and Networking, Wentworth Institute of Technology

{leungt, yuj6, …}@wit.edu

***Abstract* — Food allergies have increasingly become an issue in children and adults. Along with the fears and worries of having an allergic reaction, we are often burdened with the research of eating allergen-free foods. This study evaluates the current research of software and technology that helps counteract these burdens. This paper also dives deep into my own solution to the problem. What I have proposed is Food Allergy Budgeter. This is a web application that was built with PHP. This application is important in proving that food allergies do not have to be an expensive burden. This application allows users to browse allergy-free recipes at a specified budget. The hope is that this easy-to-use application is something that users will go to when food shopping.**

***Keywords—Food, allergies, budgeting, recipes, web, application***

# **I. Introduction**

Food allergies are a medical condition, not a food preference. Imagine eating a meal and having realizing that you cannot breathe. Food allergies are no laughing matter. It is approximated every year, 30,000 Americans are hospitalized for severe food allergies and it is estimated that 150 to 200 Americans will die from one [6]. Unfortunately, the prevalence of food allergies is quickly on the rise. It is estimated that 4%, or about 12 million, Americans suffer from food allergies [2]. This astonishing number is under-evaluated and the actual amount may be millions more. This number does not even take in account the millions of other people who suffer with this issue across the globe.

Not only does food allergies affect the individual, but those around them. Often times, an entire family will adapt their eating patterns to accommodate the needs of the one with the food allergies. For families whose children suffer from food allergies, they often incur a huge financial burden in order to protect their child. A family with insurance to cover medical costs, still supplements up to $400 on just food [5]. Along with investing in the Epipen, a lifesaving injection, families must pay the high cost of foods and other products.

Food is essential to living. One day, I began having allergic reactions to foods that were a part of my everyday life. From this moment on, I began scanning ingredient labels and asking vendors if their food contained any allergens. Living with food allergies became a hassle for me. Shopping at high-end grocery stores such as Whole Foods ensured that their foods were made allergen-free. Though they were, these foods came with a high price tag, and for every meal, took a toll on my (and my family’s) budget quite quickly . Tired of my wallet taking a hit, I looked into making my own foods from scratch. Combing through the cookbooks and online resources, I found it was possible to make allergen-free foods without breaking the bank!

The idea for the web application, Food Allergy Budgeter, was born from my own personal experiences with food allergies. This application will make the pain in searching for food-safe recipes virtually disappear. This web application will easily be accessible to users on mobile and tablet devices. This project was built using PHP in the back-end and MySQL for the database. This application has a simple interface which allows users to find recipes that fit in their specified budget. There is a variation in types of recipes: breakfast, lunch, or dinner. The recipes are multi-regional and any user is bound to find one that satisfies their diets and taste buds. The hope of this application is that people will realize that they do not need to shop at high-end stores to satisfy dietary needs. Planning and budgeting meals can significantly impact wallets (in a good way!).

In this paper, I will start by introducing other people’s ideas for users with food allergies in Related Work. These people have created their own applications that they believe help those who suffer from food allergies. Next, I will introduce my own solution and explain the design of the application. I will then discuss the implementation of the application and potential issues. Finally, I will conclude with results and future work.

# **II. Related Work**

There are many applications that work towards bettering the lives of individuals living with food allergies. One software idea, proposed by Henricksen et al.[4], allowed users to track their diets via a food diary. Remembering what we ate yesterday can be challenging, and this application solves this problem. It allows users to simply enter their meals, but has no additional benefits to bettering their finances.

Another application, AllergyBot, was proposed by Hsu et al., [7]. This chat-based application links users together. With this application, users are able to communicate whether restaurants accommodate their dietary needs; however, this forces users to repeatedly dine-out in order to get a lot of use.

Grifantini et al., [3], gathered information about the latest applications helping people who suffer from food allergies. One of the applications she wrote about was called Ipiit. This application allowed users to use their mobile devices to scan barcodes of products at grocery stores. The application would return why the scanned product is good or bad for you. The concept of taking this application with you to the grocery store is something that I’ve transferred to my application. Though this application helps a user with selecting products, its downfall is that it is extremely time-consuming and does not put together a recipe for you. The simplicity of the user interface is easy to understand. Users will be able to tell immediately what is safe and what is not.

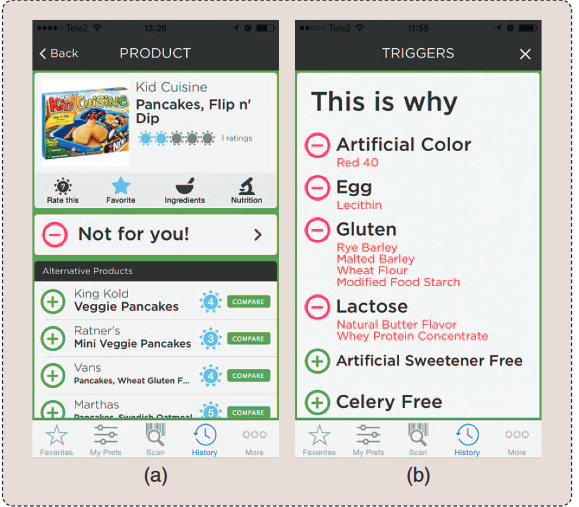


Figure 1. Ipiit user interface for scanned barcode

Those all of these applications serve a purpose for helping people with food allergies, they do not serve the needs that Food Allergy Budgeter will. The web application I have created will make life easier for users. There will be no hassle of digging through websites to find food-safe meals. All the information will be directly accessible to users by filling out a couple of fields in a form.

# **III. Proposed Idea and Research Hypothesis**

Based off research, there is a large socioeconomic disparity in families who live with food allergies and those who do not. Bilaver et al., [1] explains that food allergies do not just affect the person with the allergy, but the entire family. This disparity is especially prevalent in families with a child with food allergies. Food allergies impact the everyday lives of these families. Parents live with a constant fear of sending their children to school or to activities because allergic reactions occur in an instant. This limits the lives of children because they are limited to environments that they can be in. Parents also must spend more time being careful in their preparation of meals. However, the financial burden of food allergies may be the worst for these families.

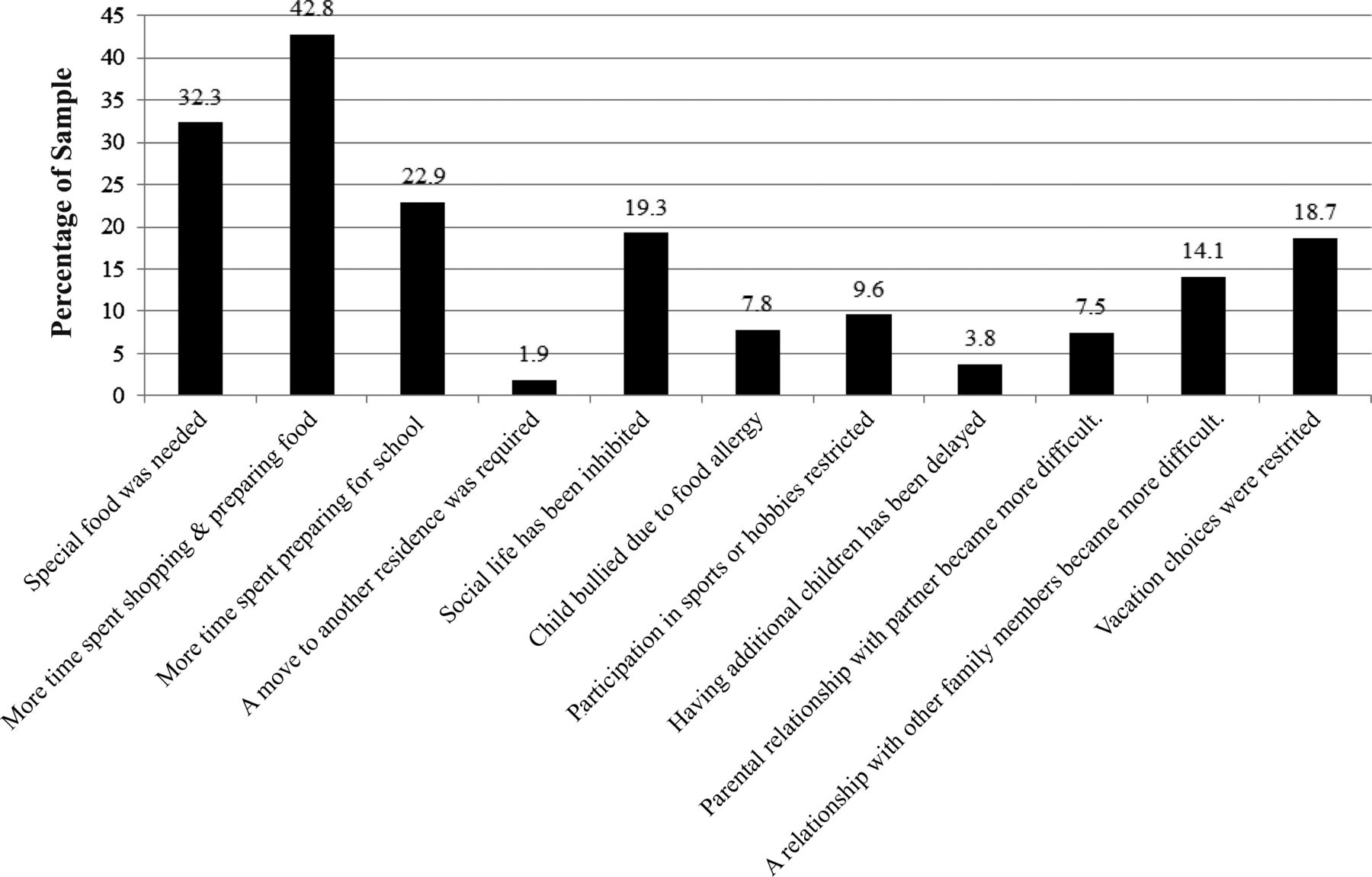


Figure 2. Socioeconomic disparities in the economic impact of children with food allergies

This project aims to revolutionize the everyday lives of individuals, like myself, who suffer from food allergies. Through the use of a simple user interface, users are able to browse allergen-free recipes. Planning out your meals should not be difficult. Instead of scrolling through countless numbers of pages, users will be able to set their search parameters in a form. Whether at home, or on-the-go at the supermarket, shopping for food has never been easier. Through a simple form, appropriately priced and allergen-free recipes will be generated for the user. Tailoring recipes to the user’s budget is vital to the success of this application. Never again, will users have to struggle financially to accommodate their food needs. This application aims to prove that living with food allergies does not have to be a financial burden. The hope is that users will realize that they do not have to go to expensive grocery stores and buy pre-made meals. Adapting to food allergies can be a fun process in the kitchen if you have this application. Within a few clicks, a user can have a custom-tailored recipe. Users can register with the web application and have their favorite recipes saved. Having a place with recipes that the user knows are perfect for their diet will save them time in the future. My hopes for this project is to reduce the stress and financial burden on families who suffer from food allergies.

# **IV. Design of the System**

The Food Allergy Budgeter was originally proposed to be designed using React for the front-end and Node.js as the back-end. Paper prototyping was done to construct a simple, but effective user interface. There were many positives to working with React. The JavaScript library is a newer technology that allows users to dynamically generate and render the page. In HTML pages, the user would continuously make requests for the data at that URL and load resources. For the back-end, Node.js was proposed. Creating my own server in the back-end would give me more flexibility due to its non-blocking nature. The advantage to using React with Node.js is because it makes communication between components with state communicate freely. For the proposed database, I wanted to also incorporate something newer. I wanted to use MongoDB because of its noSQL properties.

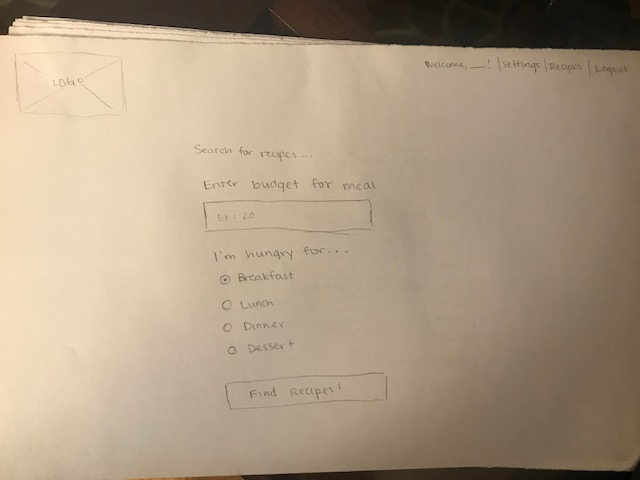


Figure 3. Drawing of user interface done in paper prototyping.

Throughout the paper prototyping phase, my thought process was to make this an easy-to-use application for the user. Figure 3 shows an example of paper prototype page that testers interacted with. Based on the research done on other mobile applications, I knew I had to make the user interface simple. A user should not have to be any searching around a page to get the information that they came for. Paper prototyping proved to be an important step towards development. Drawing out the user interface in steps made it clear where a user may get stuck. Using two test subjects, I gave them scenarios for the application usage. They were prompted to replicate their steps and think through their thought process out-loud. Through these iterations of prototyping, I found that there were holes in the application that could be improved.

The paper prototyping solidified the appearance of the user interface. In the web application, everyone has access to search for recipes. If a user is not registered and logged in, they are automatically redirected to the login page. Their access to these pages are restricted. Once logged in, a user can simply enter their search criteria in a form: the meal that they are looking for, their allergies, and their budget for the meal. After form submission, the appropriate recipes that match the criteria are output. Clicking into a recipe page, users have the option to add this recipes to their favorites. With a user interface decided on, development could begin.

# **V. Implementation**

Although the web application was originally proposed to use React, Node.js, and MongoDB, the implementation changed drastically. There was a high level of complexity to learning three new technologies. Due to time constraints, there was not enough time to become comfortable to program with them. As a solution, I developed the user interface in HTML and Bootstrap framework for CSS. After establishing a solid front end, back-end development began. To run all the code, I used XAMPP. I began back-end development with PHP and a MySQL database.

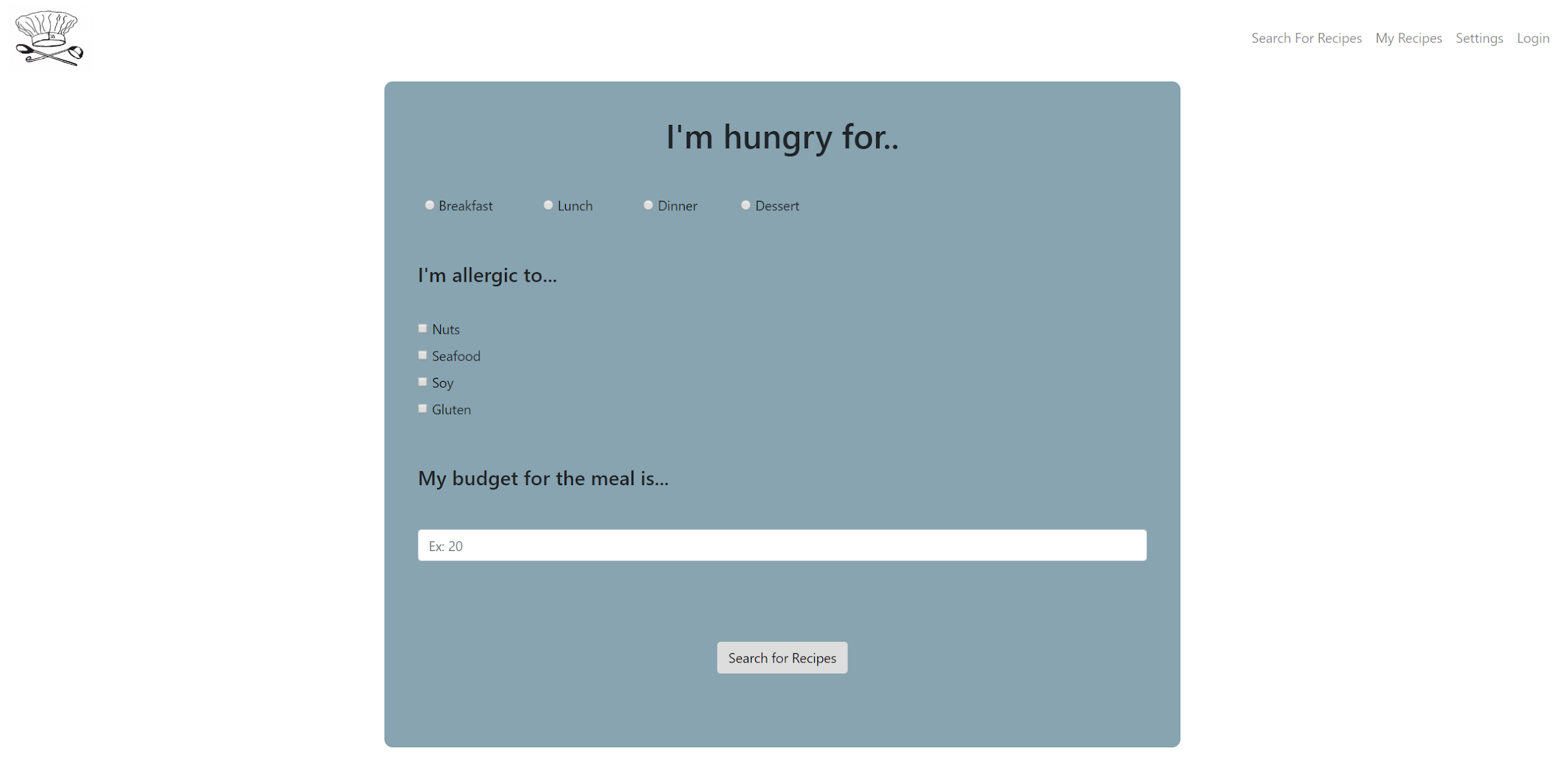


Figure 4. Search form for recipes.

A. *PHP*

Implementing PHP in the back-end allowed me to create a smooth experience for the user. Because the search functionality of recipes was open to both non-registered and registered users, they could bypass the login form. Those who chose to login or sign up for the application went through a simple process of checking if the data entered matched the information in the MySQL database. The login and sign-up forms were submitted via a POST request and sent for processing through another PHP page. I queried the database using a myqsli\_query function call. If the supplied user data was correct, the PHP function session\_start was called. This works like cookies do on a website. Upon session start, a user’s login and ID were saved in the session. For the remainder of the time the user is on the page, the login email and ID are associated with the session. When adding recipes as a favorite, it would know which user was the one who inserted it into the database. In addition to validation of users, I used PHP to return search results. On the recipe search page, users entered their meal of choice, personal allergies, and budget for the meal. The entered values were queried and looped through. For each result from the database, HTML elements about the recipe were added to the results page. If no results were found, no recipes were displayed on the search results page.

B. *MySQL*

Figuring out how to link the database tables together proved to be challenging. The userinfo table held the information for all registered users. This table had columns that held user’s first name, last name, email address, password, and allergies. There was also a user id associated with each registered user. This column was automatically created and incremented by one. Keeping track of allergies was a challenge. Initially, my idea was to comma-separate the allergies in a list. To make it easier for querying, I set the four different allergies to a default value of zero. For example, if the user was allergic to nuts, they would check the box off in the form. Upon form submission, the other three allergies would be left at their default values, but the value for nuts would change to one. Figure 5 shows the current structure of the userinfo table. Users who had allergies have default values changed.

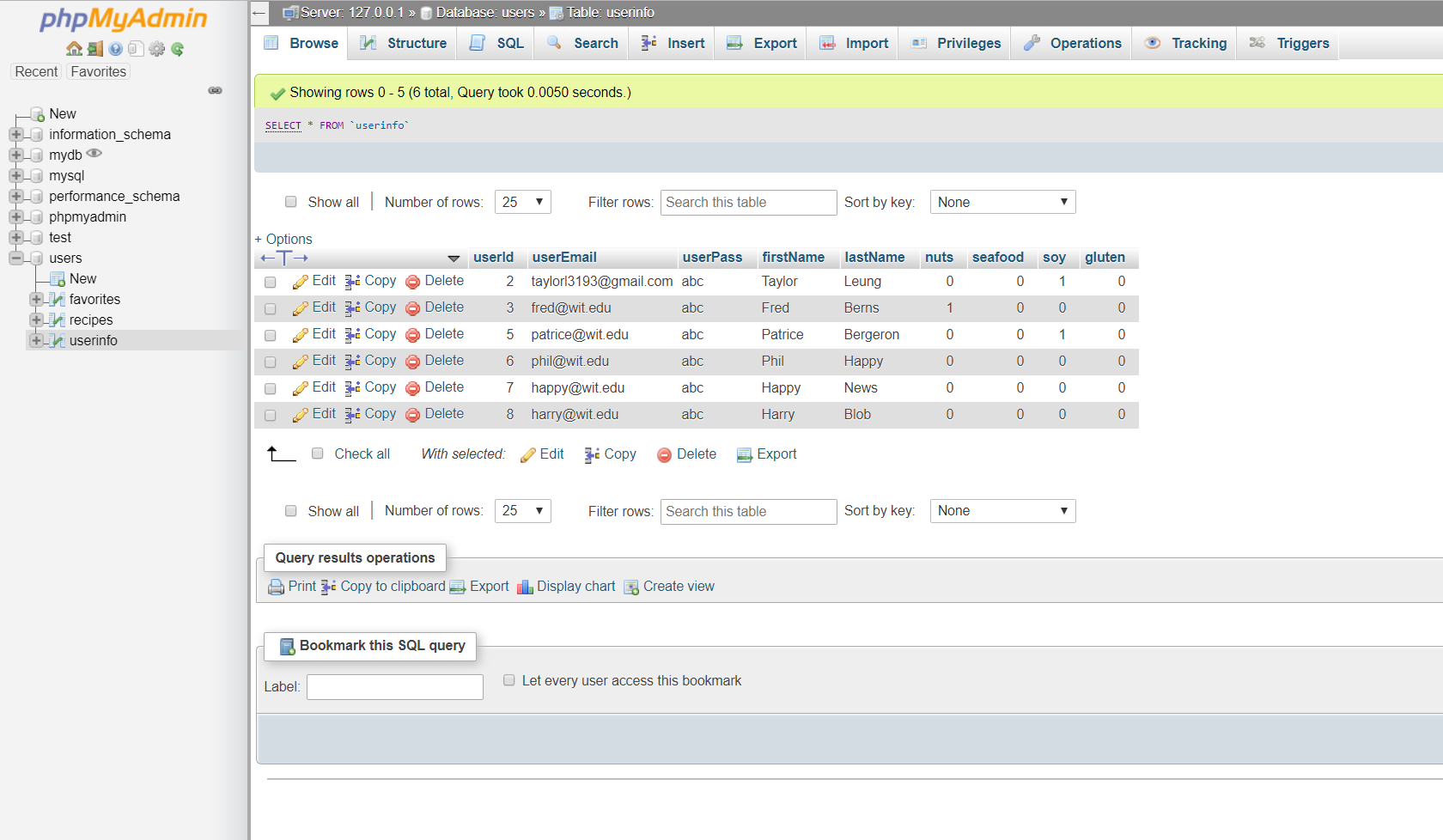


Figure 5. Users database, table userinfo.

Another issue that I ran into during implementation, was dynamically adding buttons to add a recipe to favorites. For the search results, I was able to dynamically generate a list of recipe names and descriptions about the recipe. When it came to dynamically creating a button, I was unsuccessful. Though it added a button to the DOM, clicking on it did not query successfully. Upon searching for answers, I found out that PHP does not have a built in onclick attribute like JavaScript does [8]. To work around this issue, I gave the button a name attribute. If this button was set, it would query the database. After trying to move the processing PHP code to another document, I still was unsuccessful in dynamically creating a button to add to favorites. As a work around, in each recipe’s page, I have hardcoded a button that will insert into the favorites table upon click.

On the topic of hardcoding, I was unsuccessful in dynamically creating recipe pages. Ideally, the appropriate recipe information would have been stored in the database, like in Figure 6. Using PHP, I should have been able to create new pages with this database information for each recipe. Due to time constraints, I was forced to hard code each recipe page. This process was extremely tedious, but was not something I could solve in the time for development.

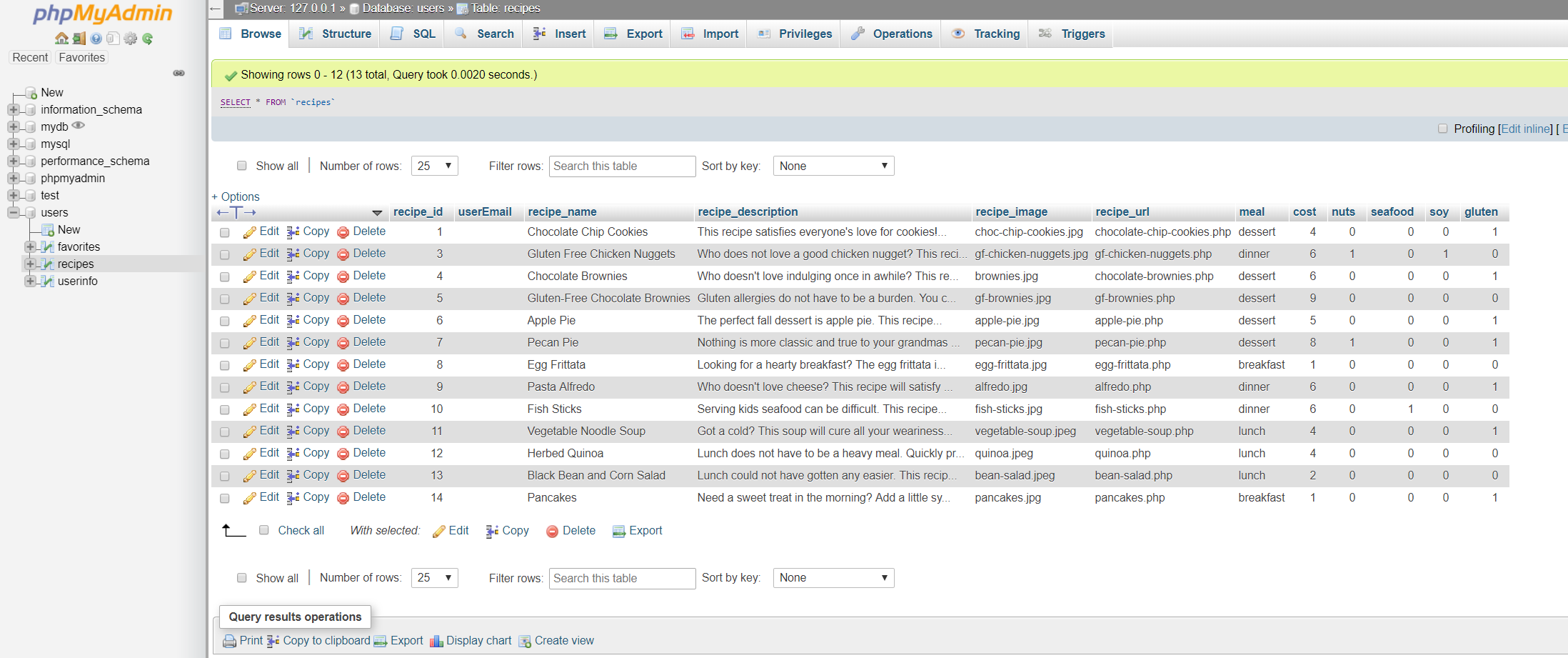


Figure 6. Recipes table with recipe information.

# **VI. Results**

Although the original design was not implemented, the application was still successful. The application still performed its original functionality. The application has optional user login and the ability to perform search on recipes for people living with food allergies.

Figure 7 shows the search results page once the user submits the search form. It shows multiple different recipes that are within the user’s budget. The way that the search results page is displayed keeps to the simplicity of the application. I did not want a complicated and overwhelming experience for the user. The user has the recipe title in a large font. A quick description of the recipe is given to entice the user. Finally, there is a small image that gives a preview of the results of baking.

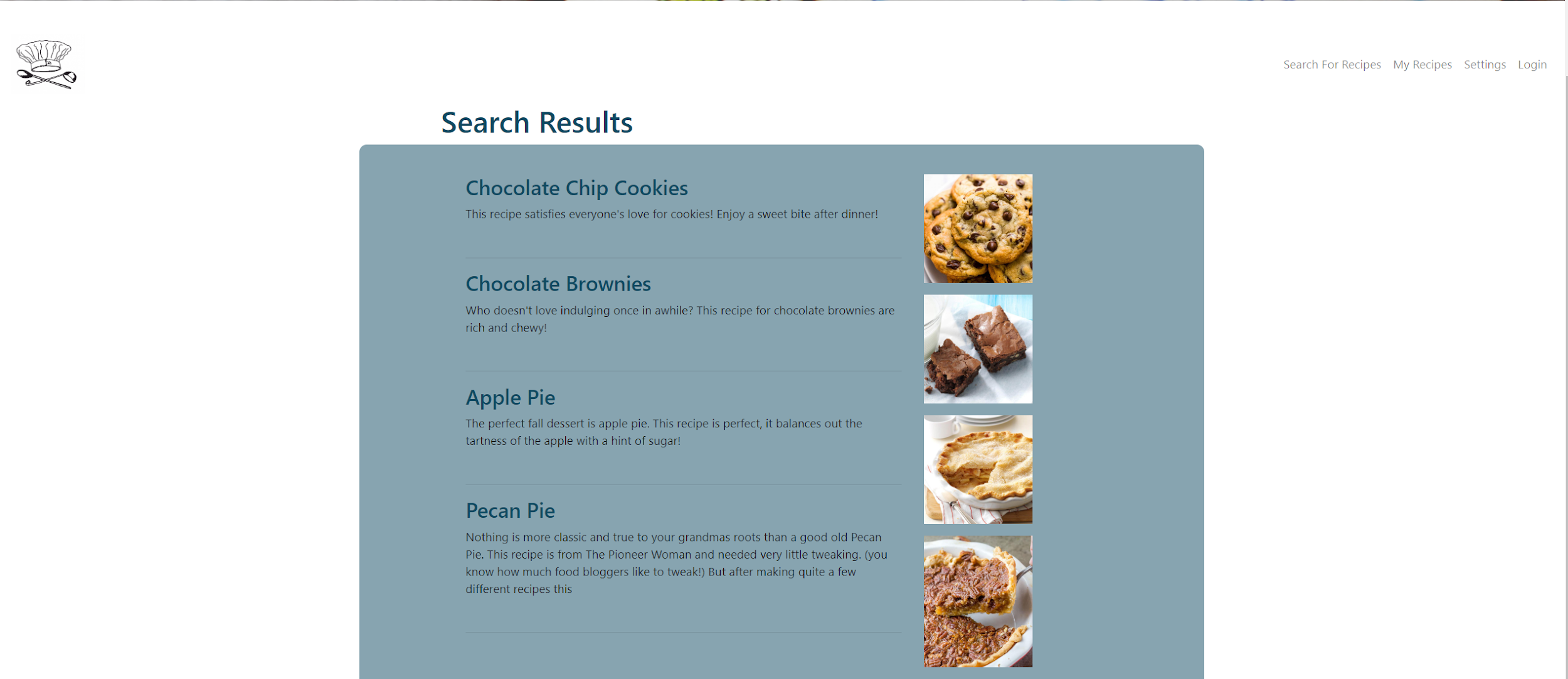


Figure 7. Search results from Food Allergy Budgeter for desserts under $10.

Once the user has a complete list of results, they are able to

view the actual recipe. Figure 8 shows the recipe for chocolate chip cookies. What is successful about the recipe page is it lets the user know right away if they are allergic to its contents or not. The top of the recipe page displays what the recipe is free of. In a large red font, the user is warned about this particular recipe containing gluten. This is an idea I took from the Ipiit application mentioned in the Related Work section. I like how a user can instantly associate danger with the color red. The text pops out on the white page and strays them away from the recipe if they cannot eat it.



Figure 8. Chocolate chip cookies recipe page.

# **VII. Discussion**

There are many things that could have been done better in the development of this application. If there was more time to develop, I would have loved to explore newer technologies. Though PHP and MySQL are a popular set of tools used for development, they are also outdated. It is a pain for users to constantly request and get data for the page because it can be a slow process. Being able to return data without the page reloading is something that would have greatly improved the application.

Another thing I would have addressed with more time is the structure of the database. I used three tables: userinfo, favorites, and recipes. For the My Recipes page, I wanted this to work similar to the search results. I wanted the user to be able to be able to access the recipe name, URL, description, and image. I thought I knew how to query information across all the tables to return the favorite recipes to that user, but I was mistaken. I was getting every recipe in the database, twice, and this is not the desired results. If I had structured the tables to better interact with each other, this could have helped. Figure 9 shows the My Recipes page for a specific user. Though it retrieved some favorited recipes, they had no information except for the recipe name. At the time of adding a recipe to favorites, I also failed to check if the recipe was already favorited in the database.

Due to time constraints, I was unable to dynamically generate recipe pages from the database. If I had more time, I would have gotten this functionality working. It was just as time consuming to manually hard-code each recipe in its own PHP page. Dynamically generating the recipe pages would have been better for development. As the number of recipes in the database grows, the larger the application will become. Because hard-coding the recipes took so long, I was unable to create a large amount of recipe pages at the time of project completion. The search results are very minimal and the user should have more of a selection. Another issue I found after development is the ability to search recipes by more than just one meal type. I currently use radio buttons for the user to select a meal type. Upon entering recipes in the database, I realized some recipes could be classified as lunch *and* dinner. Restructuring the database to accommodate this issue would have generated more results.



Figure 9. Current state of the My Recipes page.

# **VIII. Conclusions and Future Work**

The Food Allergy Budgeter is a useful application to assist those who are affected by the burden of food allergies. Based on the results of development, the application is effective in giving the user the information that they desire. If this application was deployed, I could see the Food Allergy Budgeter as a go-to tool during food shopping. The great thing about this application is that it can be used during the stages of planning meals, or on-the-go at the supermarket. I love how simple of an idea can improve the livelihoods of people with food allergies, myself included. Food shopping no longer has to be a burden and a hit to your wallet.

After development, I am still confident in the Food Allergy Budgeter. It is definitely a useful application that I know people with food allergies would appreciate. Though the application idea is great, the development is lacking. In the future, improvements can be made to the application. I would redo the application in the proposed technologies.

##### **Acknowledgment**

I would like to acknowledge and thank Professor Chen-Hsiang (Jones) Yu at the Wentworth Institute of Technology for conducting this independent study.

##### **References**

1. Bilaver, Lucy A., et al. “Socioeconomic Disparities in the Economic Impact of Childhood Food Allergy.” *Pediatrics*, American Academy of Pediatrics, 1 May 2016, pediatrics.aappublications.org/content/137/5/e20153678.
2. “Food allergies on the rise? Determining the prevalence of food allergies, and how quickly it is increasing, is the first step in tackling the problem” *EMBO reports* vol. 7,11 (2006): 1080-3.
3. Grifantini, Kristina. “Knowing What You Eat.” *IEEE*, 27 Sept. 2017, ieeexplore-ieee-org.ezproxywit.flo.org/stamp/stamp.jsp?tp=&arnumber=7579262&tag=1.
4. Henricksen, Karen, and Stephen Miller. “Design of Software to Support Families with Food-Allergic and Food-Intolerant Children.” *ACM Digital Library*, 26 Nov. 2012, dl-acm-org.ezproxywit.flo.org/citation.cfm?id=2414571.
5. Konrad, Walecia. “Food Allergies Take a Toll on Families and Finances.” *The New York Times*, The New York Times, 14 May 2010, www.nytimes.com/2010/05/15/health/15patient.html?ref=health.
6. Office of the Commissioner. “Consumer Updates - Food Allergies: Reducing the Risks.” *U S Food and Drug Administration Home Page*, Center for Drug Evaluation and Research, 18 Dec. 2017, www.fda.gov/ForConsumers/ConsumerUpdates/ucm089307.htm.
7. Tsu, Paris (Pei-Ting), et al. “AllergyBot: A Chatbot Technology Intervention for Young Adults with Food Allergies Dining Out.” *ACM Digital Library*, 6 May 2017, dl-acm-org.ezproxywit.flo.org/citation.cfm?id=3049270.
8. Web Master World. *onclick php and mysql,* 29 August 2009, https://www.webmasterworld.com/php/3980996.htm.