

## **01 - Proposal Summary (Before) :**

Our proposal was to build “The Calorie Guru”. It is intended to be a calorie tracker that college students use to monitor the nutritional content of what they consume.

The basic expectation of functionality was to scrape the web for average nutritional information of a given food and store it in our database, given that it does not already exist there. We wanted to also cover the basic functionality of a standard calorie tracker app and display the nutritional information on one graphic.

To make ourselves unique, we had planned two things: a nutritional label scanner, which we recognized as a stretch goal, and a recipe book implementation. This way, we could calculate the nutritional value of custom meal entries for the user.

Our inspiration for this project was the “Freshman 15”, which had our group asking the question: how does our diet change before versus after college such that this large of a weight fluctuation comes about? It was our hope that our application could shed some light on the matter.

## **02 - Description of Project (After):**

“The Calorie Guru” did end up having most of the basic functionalities of a standard calorie tracker app. You can add meal entries to a log, which get added to the database, though no check of whether or not the database has an existing entry is done. It also lacks a breakdown into 3 meals + misc snacks in a day, as well as a customized nutritional plan to a user’s body type.

We were able to make a functioning sign in feature that creates tables within our database for users’ logged meals and created recipes. Our recipe book works in that it sends the appropriate queries and sums the nutritional information that is stored in the databases. The same holds for our meal logger. These queries are not to a custom web crawler, but an API we chose to use for the convenience of time.

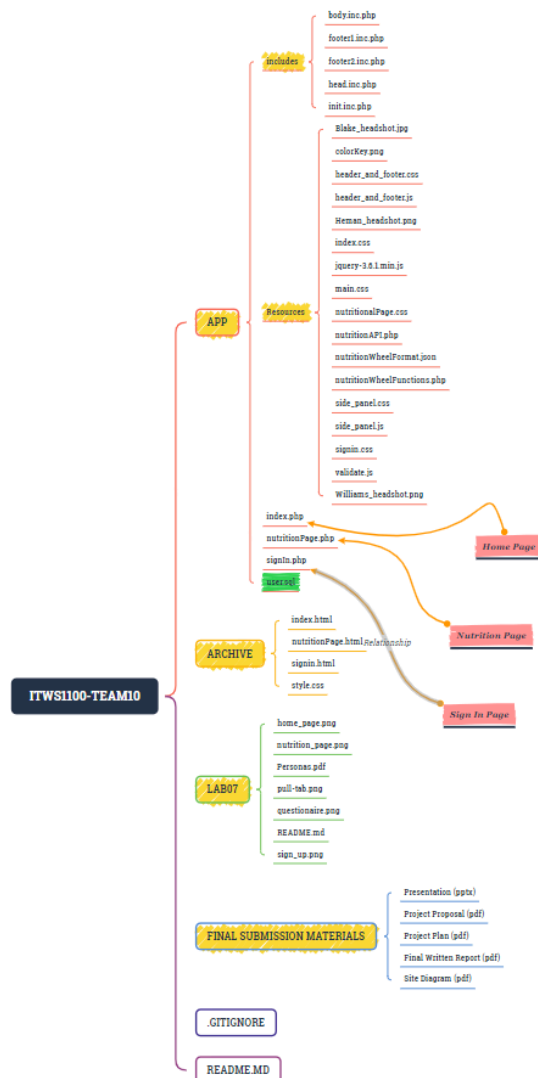
We did manage to get a graphic built that dynamically updates in response to the newly logged meals, which builds the display of our project for nutritional information and a calorie count.

## Group Members: Blake Kessler, Heman Kolla, & Williams Chen

### 03 - Project Plan:

Link: [Project Plan](#)

### 04 - Architecture & Decisions Made:



For the above IA: yellow means folders, red means a web page, and green means setup for the database. We first made our project in HTML with CSS stylesheets and some JS for interactivity. Then, we converted our code into PHP, which created the need for an 'includes' folder to store reusable code modules,

leaving old files in the 'Archive' folder. The remainder of our code to interact with the database and store back-end functions took place in other php files. Everything but the database setup sql file and the php files that represented web pages were stored in the 'Resources folder'—CSS, JS, and other PHP files plus images. All code unrelated to the application is stored outside the 'app' folder to account for miscellaneous files. The submission items, including this file, are in their own folder. Instructions on setup/use/how to demo are in the README.

## **05 - Discussion of Challenges & What we did about them:**

One of the significant challenges we encountered during development was correctly sending API requests. We were originally planning to build up our own database of nutrition facts, but we eventually decided against that because it seemed easier and quicker to just use an API. We settled on the [API-Ninjas Nutrition API](#) (which is probably better than we could do on our own anyway, and allows non-commercial use). We discussed using AJAX to query the API, but settled on curl in PHP because most of the project was already in PHP at that point, and assumed it would be relatively easy to implement. However, figuring out how to properly attach an API key to be able to send queries took far longer than expected. It also didn't help that we got it working, but other issues with our PHP code made it not echo properly (making us think that the API requests were still broken). We eventually figured out how to send an X-API-Key in the header of the query, which resolved the issue.

Another challenge we encountered was getting the Recipe Book feature working properly. We assumed it would be relatively simple because we already had working PHP and SQL code for nearly-identical tasks, but we encountered strange issues in development. At once point, echoing the SQL command in PHP didn't do anything, even though echoing simple strings immediately before and after that did work; once we resolved that, the array it returned worked fine - until any of its elements were accessed (passing it around or getting its length worked, but accessing elements crashed the program), which ended execution. We managed to fix those issues by fixing the SQL command and changing the ordering of the code by moving to different places within the same file.

## **06 - Project Summary:**

Primary Area of Focus : Area 5 - Put real data in a database, cookie, session storage, or local storage

Secondary Area of Focus : Area 4 - Pull real data from a database

Section 02 covered the “Description of the Project”, which covers the current status of our project and its functionalities.

Primarily, we learned a lot about PHP and how SQL interacts with databases. We also learned a lot about how to develop a project together as a team, and how to construct a workflow that follows agile methodologies, such as a sprint-based approach.

There are many things that we would like to add to our project. For one, in the sign in feature, we would like to gather information about the user’s body type to give them customized nutrition values for their ‘daily recommended’ values in place of default values. We would want to log entries to different meals within a day, and different days as well, adding a timestamp to our database. One more idea would be to create a forum where recipes can be shared by users. A stretch goal we had that we would like to accomplish would be the nutritional label scanner idea, using computer vision. Also, we would like to style our page better, and have smoother animation.