

ULTIMATE RECIPE – COOKING AND GROCERY SHOPPING HELPER

Capstone Final Report

Presented to the

Faculty of CST 499 at

California State University, Monterey Bay

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

in

Computer Science

by

Orion Analytics

Carlos de la Roca, Ksenia Koldaeva, Paulo Wu Liu, Miki Fukushima

Fall 2022

EXECUTIVE SUMMARY OF PROPOSAL

Ultimate Recipe - Cooking Grocery Shopping Helper

by

Orion Analytics

Bachelor of Science in Computer Science

California State University Monterey Bay, 2022

The purpose of this project is to provide a web-based application that can change your experience with cooking. Based on a survey done by Huffpost in 2011, approximately one third of Americans do not know how to cook which is ironic considering eating is an essential part of our everyday lives. Luckily, we live in a modern society where food is accessible at places like restaurants and grocery delis, but eating out can be both expensive and unhealthy. This application; Ultimate Recipe will enhance your cooking experience.

Ultimate Recipe will be powered by an extensive nutrition data and analytics database. Users will be able to quickly access the type of dishes they can make based on the ingredients (or keywords of ingredients) they have at home and will be able to save the recipes aggregated across multiple sources with the minimal friction. The application will contain an easy-to-use graphical user interface with a lot of visual graphics so people from any age should feel comfortable navigating through. This project has the potential to impact a large number of Americans who currently feel intimidated by cooking and most importantly, make cooking a fun activity.

TABLE OF CONTENTS

ULTIMATE RECIPE – COOKING AND GROCERY SHOPPING HELPER	1
EXECUTIVE SUMMARY OF PROPOSAL	2
TABLE OF CONTENTS	3
PART I	5
Introduction	5
Project Description	5
End Users: Usage and Accessibility	6
Feasibility Discussion	6
Environmental Scan	6
Ethical and Legal Issues	7
Long Term Life of the End Product	8
Milestones	8
PART II	9
Specifications	9
Potential Users	9
User Requirements	9
Desired Features	10
Platform	10
Application Hosting Infrastructure	10
Development Platform	11
Costs	11
Major Functions	12
Technology	12
Functionality Descriptions	12
Frontend Functions	13
Backend Functions	13
Performance	14
Usability Testing Plans	15
Behavior Driven Testing	15
Focus Group	16
Evaluation Methods	16
Usability Testing and Evaluation	17
User Stories Verification	17

Additional Feedback	18
Action Items and Feedback Followup	18
PART III	19
Timeline/Budget	19
Final Implementation	21
System Design	21
Database Design and Integration	22
Backend Implementation	23
Frontend Implementation	24
Google OAuth Integration	25
Application Deployment and Rollout	26
Discussion	28
Design Phase	28
Implementation Phase	29
Rollout and Future Improvements	29
References	31
Appendix	32
Appendix A	32
Evaluation Test Plan and Usability	32
Appendix B	33
Team Member Role and Division of Responsibilities	33
Appendix C	35
Pertinent Documents	35
Google Cloud Platform Dashboards	35
Ultimate Recipe MySQL Database Schema and Application Level Data Models	37
Database Initialization	39
Development Tools	40
Initial UI/UX Design Proposals	41
Ultimate Application Screenshots	43

PART I

Introduction

Project Description

As a part of CST499 Capstone project, Orion Analytics team is developing an application named Ultimate Recipe. Ultimate Recipe is a web-based application that is built to enhance the cooking experiences for the general end users regardless of their age, gender, ethnicity, or background. We see that the food is something we all enjoy therefore we want to make the application intuitive and available for everyone. While we are designing and developing this application, we will be conscientious that Ultimate Recipe should accomplish below aspects:

- The application will be easily accessible via web-based interface
- The application will contain rich visual representations such as images or gif allowing it to be intuitive for end users from different background who are in wide range of age groups
- Data is retrieved from existing public APIs so that end users can access a wide range of dataset
- The application allows end users to save their favorite item. The favorited items are saved in a backend jawsdb MySQL.
- The application supports Google authentication

End Users: Usage and Accessibility

Another benefit Ultimate Recipe offers is the accessibility. The application will be hosted on cloud servers as a web-based application allowing the end users to:

- Access the application through any web browsers
- The end users can authenticate and log in using the existing Google account via Google authentication method

Feasibility Discussion

Environmental Scan

As discussed in our Capstone Project Proposal, the young adults generation now heavily relies on technology, and cooking is not an exception. In the Capstone Project Proposal we called out that “Millennials are more likely to use technology when they cook” (Matthews, 2018), and identified such popular cooking websites as [Allrecipes](#) and [Tasty](#) (Li & McAuley, 2020). We further looked at other cooking websites and applications such as [SuperCook](#), and the [“BigOven”](#). Despite the number of existing cooking web and mobile applications, the Ultimate Recipe can be valuable in the existing market.

The Ultimate Recipe will be backed by the powerful provider of nutritional data and analytics – [Edamam](#). We have integrated the API provider to the Ultimate Recipe backend service. The backend service has the capabilities to arrange and serve this data in new ways, as well as saving individual users preferences by providing the favoriting functionality. The data coming from the database is labeled with such categories as diet type, health labels, ingredients and nutrients, which allows for the application improvements such as most relevant suggestions to individual users, and accurate searches to a specific demographic.

As discussed in the Capstone Proposal, the main goal for the Ultimate Recipe initial phase would be to reduce friction to use cooking applications for a variety of user groups such as kids, college students, and busy adults. Busy and cluttered UIs are a source of frustration for the end users. Modern UI frameworks make it easier for the developers to create pleasant, responsive and accessible web interfaces. Thus we have researched the frontend frameworks for the view layer of the application. The Ultimate Recipe web application is using [ReactJS](#) framework, as well as libraries such as [MaterialUI](#) and [Bulma](#). The technology stack we have chosen is designed to create a responsive and intuitive web application layout while offering excellent page load times. In addition, the APIs that the Ultimate Recipe is integrated with provides normalized data with the same structure for the data coming from multiple sources. This addresses two major functionalities from the user perspective – the user interface will remain cohesive, despite the data aggregated across multiple sources, and, at the same time, users will have access to the original posters of the recipes, via the hyperlinks attached to time images and from within the description.

Ethical and Legal Issues

We anticipate facing various ethical and legal issues throughout the life cycle of our application. Through market research and environment scans, we were able to determine the best ways to mitigate any problems. Among the ethical and legal issues we anticipate to face, privacy will be a major concern for us as developers and our users. In order to protect the privacy of our users, we will be utilizing Google OAuth to handle our authentication. Google's OAuth will allow users to quickly and painlessly sign up for an account with our app. We will only use email addresses to uniquely identify users, and any other information we capture from Google OAuth will be shown to the user upon sign up. Ultimate Recipe will keep a link to our privacy policy at

the footer of our application so anyone with access to our web application will be able to view it. We ultimately want to keep the trust of our users and will continue to be transparent with what data we collect from our users, how we collect it, and how we store it.

Long Term Life of the End Product

Milestones

For the long-term life of end the product, we aim for the following milestones:

- Present the initial version of the application to the first users – student peer reviews, family members and friends (Closed Beta phase).
- Address initial peer feedback, and announce the application to the public.
- Create a backlog of most desired features, and start iterative project maintenance and new feature development.

At the present moment we identify such additional features as Machine Learning based individual recipe recommendation engine, additional search and filtering functionality, and ingredients photographs uploaded instead of the full text search. Since, we are adopting the Agile methodologies for the product development, we plan to put the product in front of the user as early as possible and further prioritize additional features based on the first adopters' feedback.

PART II

Specifications

We aimed to create an application that would benefit as many end users as possible. We researched criteria for what type of recipe application would not only cater to a wide range of users but also what we would look for in a recipe app. Our research helped us find our potential users, what requirements the users would need for a recipe app, and the desired features we and our users would like.

Potential Users

Our potential users would be people with little time to research new recipes. Our potential users would also be those who have trouble storing recipe information once found. Additionally, a potential user base our application would cater to would be those that want to access recipes on any device they may own. Finally, we think our application would cater to those who are willing to learn about new recipes.

User Requirements

Our users would require to quickly gain access to a large amount of recipe's. Our users would also wish to conveniently store any recipe information they may have enjoyed for their next visits. Our application would need to provide recipe information quickly for those on the go and store the information for easy reference in the future. We envisioned an application that would handle that functionality and also have a clean user interface to give our users a painless way to learn about new recipes. We use current User Interface technologies like Bulma to create a clean and familiar user experience. User privacy is another user requirement we researched

before building this application. We found one the quickest and safest ways to store user information is to use Google Authentication, which provides quick and simple user authentication. We also utilize current front end technology in order to eliminate any potential threat of leaking user information. For example, our frontend and backend are both hosted separately on secure servers.

Desired Features

Once we completed our application, we realized that there were other requirements we could potentially add to create a better user experience for everyone. One example would be adding a “recipe recommendation engine” to easily recommend brand new recipes to users. We would utilize open and free machine learning models to help build out the recommendation engine. Another example of a desired feature for our application would be to add a social media component to our application. Currently, there is no way to follow a friend and view their favorite recipes but we could eventually create this social component to help users learn about new recipes in a more engaging way. Finally, another desired feature would be creating recipes on the application. This feature would allow users to create and share recipes with their friends and all users of **Ultimate Recipe**.

Platform

Application Hosting Infrastructure

Google Cloud Platform is chosen as the primary application hosting platform. Google Cloud Platform provides an array of services from the cloud hosted MySQL database, to Google OAuth Client APIs, and to the declarative application deployment for a selected programming

language and server engine. Google Cloud Platform provides a great convenience with a single projection creation – network connection works seamlessly within a single project and requires a minimum amount of configuration. We also confused the JawsDB MySQL database hosted on AWS. This allows the team to compare functionality of 2 major cloud services providers and to make the best selection in the long run.

To deploy the main application we use [Google App Engine for NodeJS](#) and “gcloud” command line tool that allows us to deploy the application within minutes.

Development Platform

As a starter team of 4 engineers we have extensively used the Github platform and Git Version Control System to develop the application. Git VCS is an excellent tool to ensure higher code quality, and to provide visibility to the code changes to all team members.

Existing deployment with Google App Engine is done manually from the local machine. However and a future improvement to the development process, it is possible to configure this deployment via Github Action and Continuous Deployment pipelines. This will significantly improve the developers productivity and the development experience.

Costs

Initially there are no costs to run both the frontend and backend services, except for a license to the recipe APIs we use. The license costs us \$20 a month to allow us to surpass the rate limiting when querying the API. This cost is necessary to provide a smooth user experience. As our user base grows, we expect to increase the size of our hosting plans and servers in order to keep up with the demand of traffic to our website and the increased need for user data storage. There are many platforms we can utilize to facilitate the process of keeping services up and

running to meet demand, like Google Cloud Platform, Amazon WebServices, Microsoft Azure, etc. Through experience with other projects, we expect the monthly cost to run on a service like Google Cloud Platform to be approximately \$100. Additionally, the initial cost of \$20 for the API may increase as our API queries increase. Finally, if we were to expand the application to include our desired features, we may need to hire additional engineers to help with the development.

Major Functions

Technology

Ultimate Recipe is split up into two components: the frontend and the backend. The frontend is built using ReactJS, a JavaScript framework used to easily create dynamic web applications. The frontend is hosted on a free hosting service on Google Cloud Platform. The backend is built using the NodeJS JavaScript engine and a popular ExpressJS library which allows us to create a server using JavaScript. All of our code is written in TypeScript which is a superset of JavaScript that provides static types to our code to eliminate errors.

Functionality Descriptions

The major functions of Ultimate Recipe can be categorized and explained as two types: frontend and backend. Since Ultimate Recipe is a web-based application that is highly interactive, it is crucial that both sides are integrated together to provide a seamless experience for the end users. The major functions of each respective part are outlined below.

Frontend Functions

- The frontend code is built using ReactJS framework. When the application is launched, the code renders a landing page which consists of pinterest-like tile images of food dishes.
- The images of food returned on the landing page are categorized into different types of dishes; one is for a regular balanced diet and the other is strictly for a vegetarian based diet.
- Each suggested dish contains associated information such as the name, calories, type of cuisine, a URL link to the recipe, and a heart shaped like button.
- The landing page contains a text search box in which end users can enter keywords related to food in order to invoke a search.
- The end users are able to select dishes as favorites and save them to favorites.
- The application performs validation to assess if a user has already been logged in to the application. The end users who are already logged in can utilize the favorite button functionality.
- The end users can navigate to the favorites page where they can view dish items they saved as their favorites.
- The end users are asked to log in using Google log-in to save dish items as their favorites.

Backend Functions

- The keywords entered by the end users must be parsed as a search parameter and the backend code sends a request against the externally hosted recipe API.
- The code receives the results in JSON structure and passes returned variables such as dish name, calories, URL to the recipe to the front end code.

- The backend code validates if an end user is already logged in using the Google log-in. The users must be logged in when adding items as favorites therefore it directs the user to log in before proceeding.
- Any items added as favorites must be inserted into the MySQL jawsdb database hosted on google cloud server.
- It analyzes the user id of the logged in user and returns a list of favorited dish items from the jawsdb (where user id matches).

As explained previously, the backend and frontend code must be integrated to allow end users to have a good experience. Below diagram Figure 1 shows the expected sequence of events as well as the user interactions on the application.

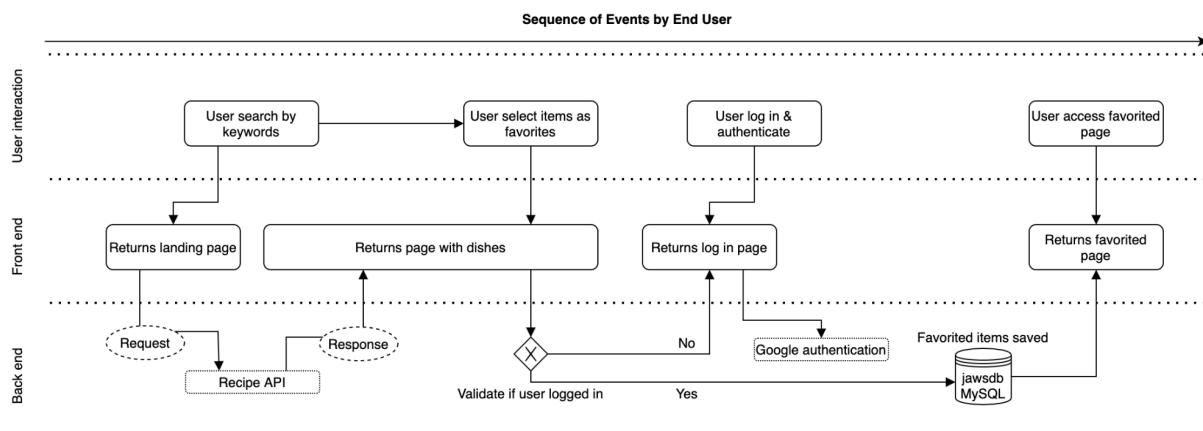


FIGURE 1

Performance

The frontend and backend applications mostly run smoothly with no perceptible performance issues. We do however see a performance issue when querying our database for

recipes. We made a decision to upgrade to the more powerful APIs plan to facilitate the higher number of queries per second and more requests to the external recipes APIs.

Usability Testing Plans

Behavior Driven Testing

To assert that the main project goal of reducing friction of using a cooking application was achieved our team decided to use Behavior Driven Testing methodology.

We identified the following user stories:

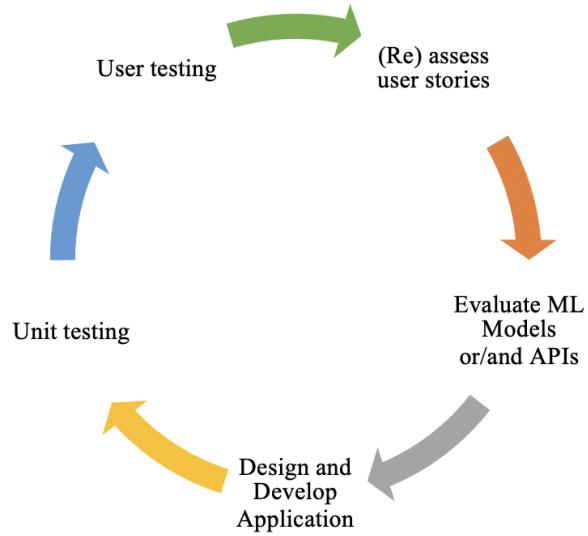
- As a user, I can login using my Google account.
- As a user I can logout.
- As a user, when logged in, I can see my profile page.
- As a user, I can search recipes using the search box on the main pages.
- As a user, I can view the results of the search. Each result must include a recipe image and link to the main recipe page.
- As a user, I can add recipes to Favorites by clicking on the Heart icon. The heart icon must change color to green.
- As a user, I can remove recipes from Favorites by clicking on the Heart Icon again. The icon must change the color back to gray.
- As a user, when logged in, I can view favorite recipes in the Favorites tab.

Focus Group

Our team has specified in the project proposal that the usability testing will be conducted during week 5 and will include participants who are interested in cooking. Our project focus group included 3 participants with different backgrounds: Aisha – computer science student, Ivan – software engineer, Tania – new parent. This aligns with our goal to reach a broad audience of cooking amateurs. In addition, members of this team were similarly involved in testing, and completed the same testing activity. Each team member focused on the part of the application that they did not directly work on, so that they can provide the feedback to other team members and the project as a whole.

Evaluation Methods

Evaluation methods included – in-person testing session, written feedback gathering, feedback analysis, action items identification and prioritization. This testing strategy and evaluation methods aligns with the Agile methodology that our team adopted throughout the project.



Usability Testing and Evaluation

The testing session was conducted online over a Google Hangout meeting, and, additionally asynchronously offline. Each member was presented with the application link <https://cst-499-capstone.uc.r.appspot.com/> available for testing. Since the application runs in testing mode with Google OAuth API, each team member's Google account was added to the test users allow list.

User Stories Verification

Each member of the focus group was provided with a list of the user stories specified above. Each team member was asked to complete the user stories tasks, and to answer if they were able to complete the task successfully.

Any failures of the user stories testing is considered production release blockers, since if a user is not able to complete these actions, that would mean they would be stuck with the

application. Fortunately, no blockers were identified, since all focus group members were able to complete the tasks. The final testing results are evaluated and tracked in the completion table – Figure A1.

Additional Feedback

Focus group members were encouraged to provide additional feedback that was recorded in the testing document. This feedback was considered non-blocking, however it presented a nice opportunity to create a nice-to-have backlog for the project. The additional feedback proved to be a very efficient tool. For example 2 out of the 3 members of the focus group, Ivan and Tanya, provided the feedback that the search bar user experience was not intuitive enough when the search query was cleared. This indicated a clear opportunity for the application usability improvement. Additionally, focus group members provided feedback on the external backend APIs, such as some queries returned links leading to non-existing pages. Valuable feedback was gathered about the project responsiveness and work on the mobile layout. That might be a desired feature, since a lot of users visit cooking websites from mobile and tablet devices.

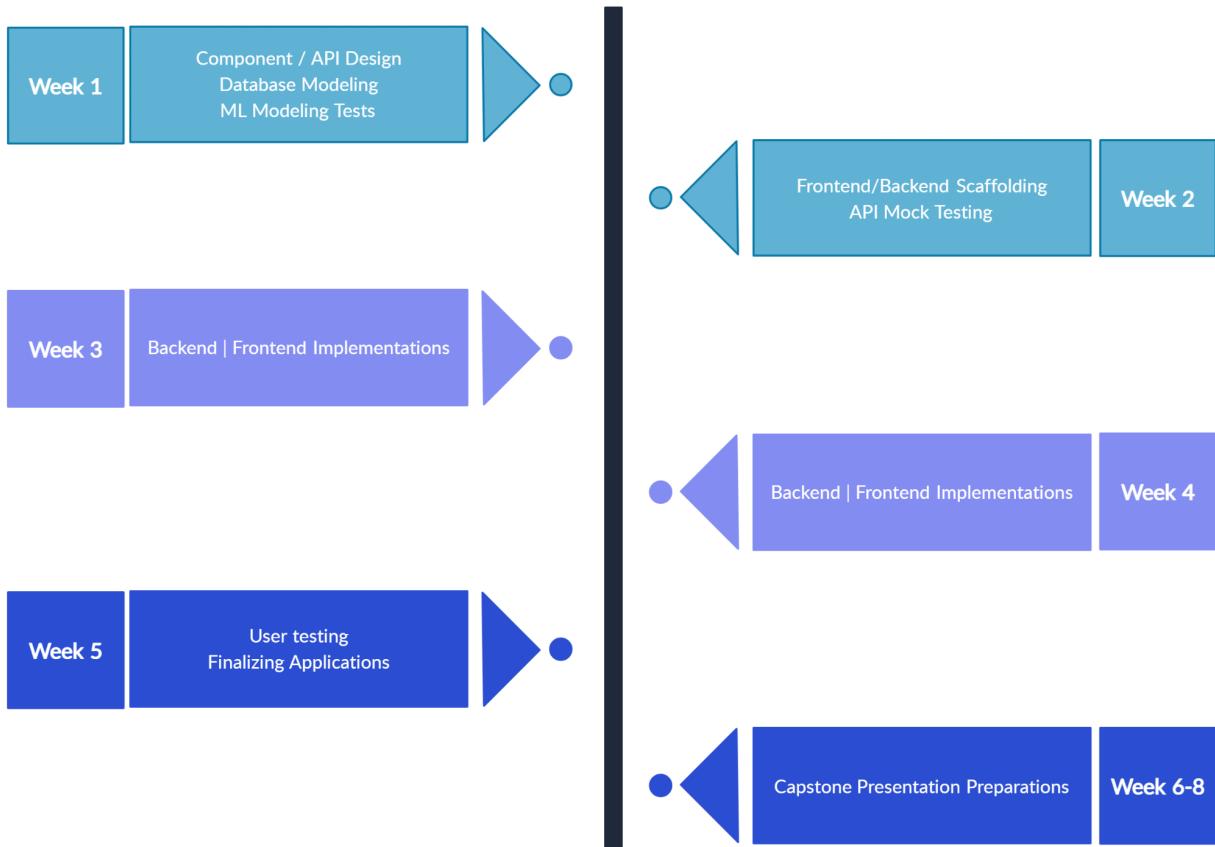
Action Items and Feedback Followup

To conclude our application testing, we have discussed with the team action items to complete for the Capstone Festival and for the public launch. We decided to improve the user experience on mobile devices, address feedback provided by the focus group member considering the search bar, and to fix page refreshes that were called out. We have created tickets on the project weekly boards, which aligns with the team's Agile practices.

PART III

Timeline/Budget

Our initial timeline was initially envisioned as the following diagram:



Our team followed the timeline closely and were able to meet each milestone. We have broken down the team members responsibility based on each team member's preferences, previous experiences, and desire to learn new technologies (See Figure B1). The team used Agile methodologies, and have set up an agile tracking board to have visibility to tasks completion among the team members (See Figure B2). Our team encountered a few hiccups in the beginning of the project work. For instance, while designing the base database model for our application during the first week, we decided to dial back our scope by not including a machine learning

model to handle the recommendations engine. With the scope narrowed down, we were able to focus on developing a product that aligns with our original vision and still to complete it within our timeframe.

In terms of budget, we were focused on utilizing free tier services to host our backend and frontend applications. We use Google Cloud Platform to host our frontend and backend applications for a promotional credit. Despite our efforts to keep the development costs low, we purchased a subscription license from our Recipe API provider in order to prevent any API rate limiting. The cost of the subscription license is \$20 a month.

The Google Cloud Platform offers the initial credit of \$400 and the trial period of 90 days to use it. The current cost of the application usage is about \$4/day, and the cost of running GCP services up to date is \$31.02 that is covered by the initial promotional credit from Google (Refer to Figure C1 for the comprehensive chart with the Google Cloud services cost breakdown).

Google Cloud App Engine charges per hour, per virtual machine (VM) instance, so we don't project a significant cost increase as we release the application to be marketed to the larger audience. As the audience grows we may need to spin up more VM instances, but even then the cost should be within \$16/day – i.e 4 instances running.

The current total cost as well as projected total cost for all services included in the table below.

Service Name	Current Cost / Month	Projected Cost / Month
Edamam Recipe API Provider	\$20	\$20
Google Cloud App Engine Deployment	\$120	\$480
Total for all services	\$140	\$500

Final Implementation

System Design

The main tenets for the final implementation are following:

1. The technology stack should be chosen with responsiveness, performance and privacy goals in mind.
2. The implementation should facilitate a straightforward connection between frontend and backend API contracts.
3. Both backend and frontend technologies should be familiar to all team members, so that everyone can contribute to both codebases.
4. The application should be easy to deploy with available cloud providers – i.e. Google Cloud or Amazon Web Services.

Based on these tenets the team picked [ReactJS](#) library and [Bulma CSS](#) framework for the frontend. The backend utilizes [MySQL](#) database, [NodeJS](#) JavaScript Engine, and [ExpressJS](#) web server. Both backend and frontend are implemented in TypeScript language. To define a consistent contract between backend and frontend, we decided to share types between codebases. Figure C4 demonstrates the types shared between the implementations and Figure C3 shows the same data model on the database. Alignment between models on all 3 levels – database, backend, frontend allowed quicker implementation that is less error prone, since the single contract is established across the entire system.

Database Design and Integration

Any information captured from the end users through Ultimate Recipe is inserted and managed in a cloud hosted MySQL database following the EER diagram in Figure C3. User specific information such as name, email is captured as a part of the user table connected to the favorited_item table through user id. Favorited_item table fulfills the task to keep dish items tagged as favorites by the application users. The two tables favorited_item and recipe are connected through recipe id. The recipe table fulfills the task of providing detailed information about each dish such as ingredients, image, the source URL.

In order to avoid interruption for the end users, a process invokes the .sql file containing the create table statements if these intended tables do not exist already in the cloud MySQL database. Figure 2 indicates the initialization logic (Refer to Figure C5 for the initialization script of the ultimate recipe database).

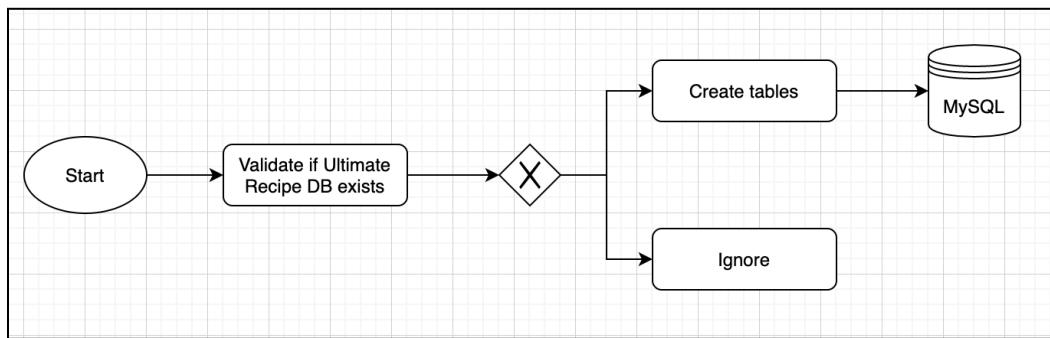


FIGURE 2

For the database connection, the MySQL client for NodeJS creates a pool of connections to ensure the database availability upon a large number of requests. The pool of connections is initialized in the following way.

```

if (env === "PROD") {
  connectionOptions[
    "socketPath"
  ] = `/cloudsql/${process.env.DB_INSTANCE_CONNECTION_NAME}`;
} else {
  connectionOptions["host"] = process.env.DB_HOST;
}

this.connection = await mysql.createPool(connectionOptions);

```

Depending on the environment, SQL connection is established with a different database host and a respective connection pool is created.

Backend Implementation

The backend of Ultimate Recipe performs 2 major functions – acts as a proxy layer to the remote APIs to fetch recipes, and stores information about users' favorites. The backend ExpressJS server exposes the REST API interface for the frontend to consume by calling the APIs with the “/api” prefix. To make the interface easier to consume and predictable for the frontend, the backend follows the [OpenAPI specification](#). According to the Open API Specification website, “The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service” (OpenAPI specification). The OpenAPI specification can be respected by adding API declarations to a file named schema.json. The schema file can be used to create automated documentation with the software called Swagger UI, that we used to communicate across frontend and backend teams. The documentation is displayed on the web UI and updates automatically as the schema declaration changes (See Figure C6). Additionally consumers, such as Ultimate Recipe frontend can fetch the schema declaration and generate a client to call the backend service.

Frontend Implementation

The frontend portion of the Ultimate Recipe application leverages ReactJS library and Bulma CSS framework to create responsive user interface for the application. ReactJS documentation states that the library allows developers to “build encapsulated components that manage their own state, then compose them to make complex UIs” (React – a JavaScript library for building user interfaces).

A fairly simple React Component can look as follows.

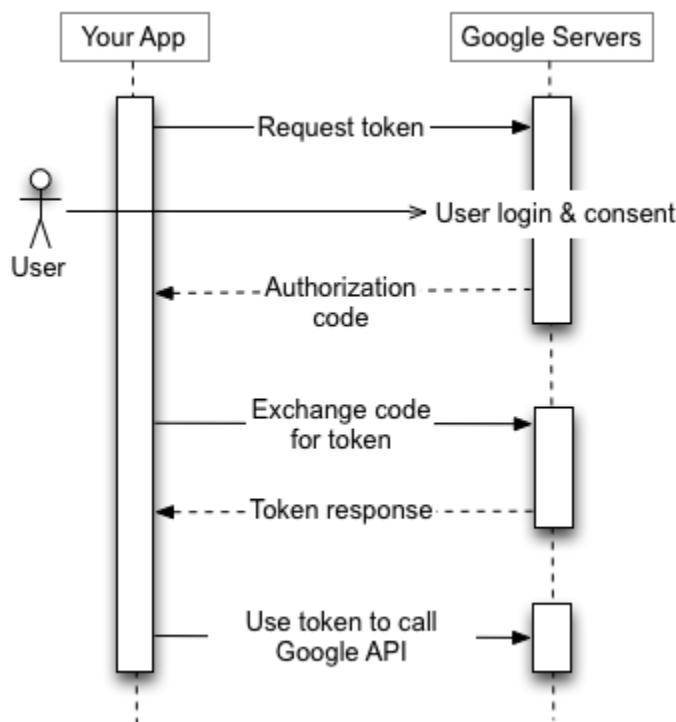
```
You, 6 days ago | 1 author (You)
1 import { FC, ReactNode } from "react";
2
3 import "./CarouselSection.css";
4
5 You, 2 weeks ago | 1 author (You)
6 export interface CarouselSectionProps {
7   heading: string;
8   children: ReactNode;
9 }
10
11 export const CarouselSection: FC<CarouselSectionProps> = ({ heading, children }) => [
12   return (
13     <section className="container is-fullhd carousel-section">
14       <h1 className="carousel-section__header heading_1">{heading}</h1>
15       <div className="carousel-section__carousel">
16         {children}
17       </div>
18     </section>
19   );
]; You, 2 weeks ago • feat: add CarouselSection component
```

In the example the file describes component properties as `CarouselSectionProps`, and the component itself. The component expects properties and returns a JSX, an extension to JavaScript, that describes what the component will look like. Here, the component will render a single heading formatted with CSS as heading 1 and “children” – nested elements passed with properties. Beyond a simple rendering ReactJS, since written completely in JavaScript (as

opposed to traditional HTML markup definition), allows to add complex logic for the UI, such as handling loading states and user errors (See Figure C9 - Figure C12 for the screenshots of the Ultimate Recipe application).

Google OAuth Integration

OAuth is an open access delegation protocol that allows users that already have an account on a major platform such as Google or Facebook to share their profile information with a third party without sharing sensitive data such as passwords. Ultimate Recipe uses Google OAuth to enable user authentication. The diagram below describes the Google OAuth Flow (Google Developer 2022).



Both frontend and backend integration were needed to add Google OAuth flow. The frontend application prompts users to login with Google. The user authenticates with the Google

servers and a one time token is sent to the Ultimate recipe backend server. The backend server then uses the token to obtain the information about the user from the Google servers. In this flow the user's credentials are never shared with the Ultimate Recipe application and the user is in control of their privacy and application access scopes (See Figure C12).

Application Deployment and Rollout

First, we rolled out the frontend and backend independently as initial staging versions. Next, we created a deployment in the Google Cloud Platform and set up a declarative deployment process with a single configuration file. Figure 3 shows an example of a deployment configuration file with the sensitive information redacted.

```
! app.example.yaml ×
!
! app.example.yaml > {} env_variables
all.json | You, 6 days ago | 1 author (You)
1   runtime: nodejs16
2
3   handlers:
4     - url: /api/.*
5       script: auto
6       secure: always
7
8   env_variables:
9     ENV: PROD
10    SESSION_SECRET: "<secret>"
11    DB_HOST: "<dbhost>"
12    DB_USER: "<dbuser>"
13    DB_PASSWORD: "<dbpassword>"
14    DB_NAME: "ultimate_recipe"
15    DB_PORT: 3306
16    EDAMAM_APP_ID: "<app_id>"
17    EDAMAM_APP_KEY: "<app_key>"
18    GOOGLE_CLIENT_ID: "<client_id>"
19    REACT_APP_GOOGLE_CLIENT_ID: "<client_id>"
```

FIGURE 3

The configuration file is then added to the project codebase, and with the help of the command “gcloud app deploy” – part of the “gcloud” command line tool provided by Google, the

application is pushed to Google. Figure C2 demonstrates the view of the Google Cloud Platform Dashboard with multiple versions of the application. The Google App Engine service works well for the Ultimate Recipe, since it allows the team to observe the application status, quickly revert to the previous versions of the application if needed, and to gradually migrate the traffic to the next version, for the large traffic production quality load.

Discussion

Design Phase

In the early stages of our design and development phase, we created mockups to illustrate what we aim to build for production. By creating the UI/UX designs before the development, the team was able to have a common understanding and the expectations for the project (See Figure C7 and Figure C8).

During the design and implementation of the project the major issue was to find the right scope of features needed to be deployed in the web application. At the start of our proposal, we envisioned a feature that would use a machine learning model that would recognize images from a smartphone or PC. Once the image was properly labeled, it would then be automatically inserted on the search bar to start a new recipe search. After researching for week 0 and 1, we realized that creating a machine learning model and implementing it would add significantly more time to the 4-week period of development and focus group testing. For that reason, it was decided to add it as a future improvement. In terms of the project rollout to the users, we believe it was successful as we have a live website that testers and users can navigate to. User rollout and testing was more practical since focus group testers were able to navigate from their end instead of joining a screen share session from our local development machines.

The team designed Ultimate Recipe as a web application that provides users with an uncluttered and visually appealing way to search and save recipes on their unique profiles. Our

main targets were to have 4 major sections: recipe search, recipe listings, authentication, and favorite features. User stories were then created and refined during development and presented to test users. After running focus group tests and reviews, results were positive since all testers were able to accomplish tasks (user stories). User feedback from testers have provided the team great insights for corrections and future improvements.

Implementation Phase

Completion of the capstone project was very challenging and rewarding. Developing a fully working site from design to deployment is a major task to accomplish but having a team made it manageable. Half the team worked on the backend while the other half worked on the frontend. The team was able to contribute by employing a mix of what we learned from our previous computer science classes to new frameworks and APIs we had to learn for the project.

Rollout and Future Improvements

During the rollout phase the team had to solve the challenges specific to the live deployment. For example, the first deployment did not declare the HTTP header needed to invalidate the cache on the main index.html file. Since Ultimate Recipe is a Single Page Application, that caused the scripts to not load properly and the application to fail. We learned how to debug the issue and fixed it by adding needed headers in the response from the backend.

Our vision for the future of Ultimate Recipe is to provide new features that will keep app's user engagement high while also motivating people to cook more and become better cooks. We plan to add machine learning features that will make recipe searching easier with image recognition and better healthy recipe suggestions. We also plan to add other features in regards to user customizability such as the ability for the user to create their own recipes to share to the world.

References

- Google OAuth Flow. (2022). Google Developer. Retrieved December 11, 2022, from
<https://developers.google.com/identity/protocols/oauth2>.
- Huffpost. (2011, September 9). Cooking Survey Reveals That 28% Of Americans Can't Cook. *Huffpost*. Retrieved October 4, 2022 from
https://www.huffpost.com/entry/cooking-survey_n_955600#:~:text=A%20new%20survey%20of%20American,reason%20for%20not%20cooking%20regularly.
- Li, S., & McAuley, J. (2020, July 30). Recipes for Success: Data Science in the home kitchen · issue 2.3, Summer 2020. *Harvard Data Science Review*. Retrieved October 4, 2022, from <https://hdsr.mitpress.mit.edu/pub/wjrl1qsq/release/3>
- Matthews, S. (2018, July 5). BYU students find more taste with technology. *The Daily Universe*. Retrieved October 4, 2022, from
<https://universe.byu.edu/2018/07/03/byu-students-find-more-taste-with-technology/>
- OpenAPI specification. Swagger. (n.d.). Retrieved December 11, 2022, from
<https://swagger.io/specification/>
- React – a JavaScript library for building user interfaces. – A JavaScript library for building user interfaces. (n.d.). Retrieved December 11, 2022, from
<https://reactjs.org/>

Appendix

Appendix A

Evaluation Test Plan and Usability

Task	Test performed
As a user, I can login using my Google account.	<input checked="" type="checkbox"/>
As a user I can logout.	<input checked="" type="checkbox"/>
As a user, when logged in, I can see my profile page.	<input checked="" type="checkbox"/>
As a user, I can search recipes using the search box on the main pages.	<input checked="" type="checkbox"/>
As a user, I can view the results of the search. Each result must include a recipe image and link to the main recipe page.	<input checked="" type="checkbox"/>
As a user, I can add recipes to Favorites by clicking on the Heart icon. The heart icon must change color to green.	<input checked="" type="checkbox"/>
As a user, I can remove recipes from Favorites by clicking on the Heart Icon again. The icon must change the color back to gray.	<input checked="" type="checkbox"/>
As a user, when logged in, I can view favorite recipes in the Favorites tab.	<input checked="" type="checkbox"/>

Usability and evaluation test results tracker.

FIGURE A1

Appendix B

Team Member Role and Division of Responsibilities

Team Member	Primary Responsibility and Tasks
Carlos De La Roca	<ul style="list-style-type: none"> ● Assess technical requirement for frontend development ● Frontend development UI/UX
Paulo Wu Liu	<ul style="list-style-type: none"> ● Assess technical requirement for frontend development ● Frontend development UI/UX
Ksenia Koldaeva	<ul style="list-style-type: none"> ● Assess technical requirement for backend development ● Backend development <ul style="list-style-type: none"> ○ API deployment ○ End point definitions ● Integration with cloud SQL database ● Integration of frontend and backend github code ● Google authentication integration
Miki Fukushima	<ul style="list-style-type: none"> ● UI/UX design proposals ● Deployment of cloud SQL database ● Integration with cloud SQL database ● Testing API endpoints
Orion Analytics (team)	<ul style="list-style-type: none"> ● System integration of frontend and backend code generally ● End user and client testing ● Process documentations

The responsibilities and tasks throughout the design and development of Ultimate Recipe.

FIGURE B1

The team's weekly Agile board with tasks progression.

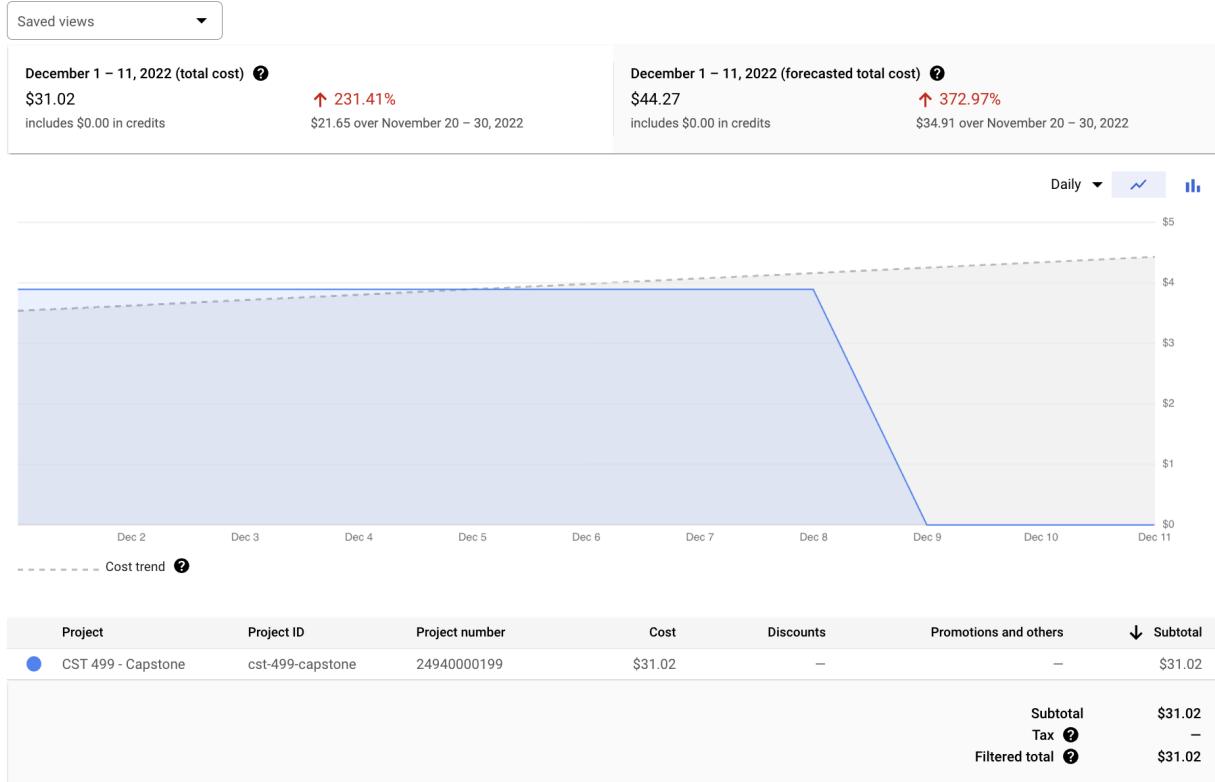
FIGURE B2

Appendix C

Pertinent Documents

All other pertinent documents which contributed to the deployment of Ultimate Recipe are listed below.

Google Cloud Platform Dashboards



Google Cloud Platform Services cost.

FIGURE C1

Free trial status: \$359.62 credit and 78 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

DISMISS **ACTIVATE**

Google Cloud CST 499 - Capstone **Search (/) for resources, docs, products, and more** **Search** **4** **?** **HIDE INFO PANEL**

App Engine **Versions** **REFRESH** **DELETE** **STOP** **START** **MIGRATE TRAFFIC** **SPLIT TRAFFIC** **LEARN**

Dashboard **Services** **Versions** **Instances** **Task queues** **Cron jobs** **Security scans** **Firewall rules** **Quotas** **Memcache** **Search** **Settings** **Release Notes**

Filter Filter versions **?** **III**

Version	Status	Traffic Allocation	Instances	Runtime
20221205t225450	Serving	<div style="width: 100%;">100%</div>	0	nodejs
20221204t175627	Serving	<div style="width: 0%;">0%</div>	0	nodejs
20221128t205640	Serving	<div style="width: 0%;">0%</div>	0	nodejs
20221128t205427	Serving	<div style="width: 0%;">0%</div>	0	nodejs
20221128t203219	Serving	<div style="width: 0%;">0%</div>	0	nodejs
20221128t193016	Serving	<div style="width: 0%;">0%</div>	0	nodejs
20221128t192859	Serving	<div style="width: 0%;">0%</div>	0	nodejs
20221128t192452	Serving	<div style="width: 0%;">0%</div>	0	nodejs

20221205t225450

12/5/22

11:00 PM Completed: Create App Engine ... kkoldaeva@csumb.edu cre...

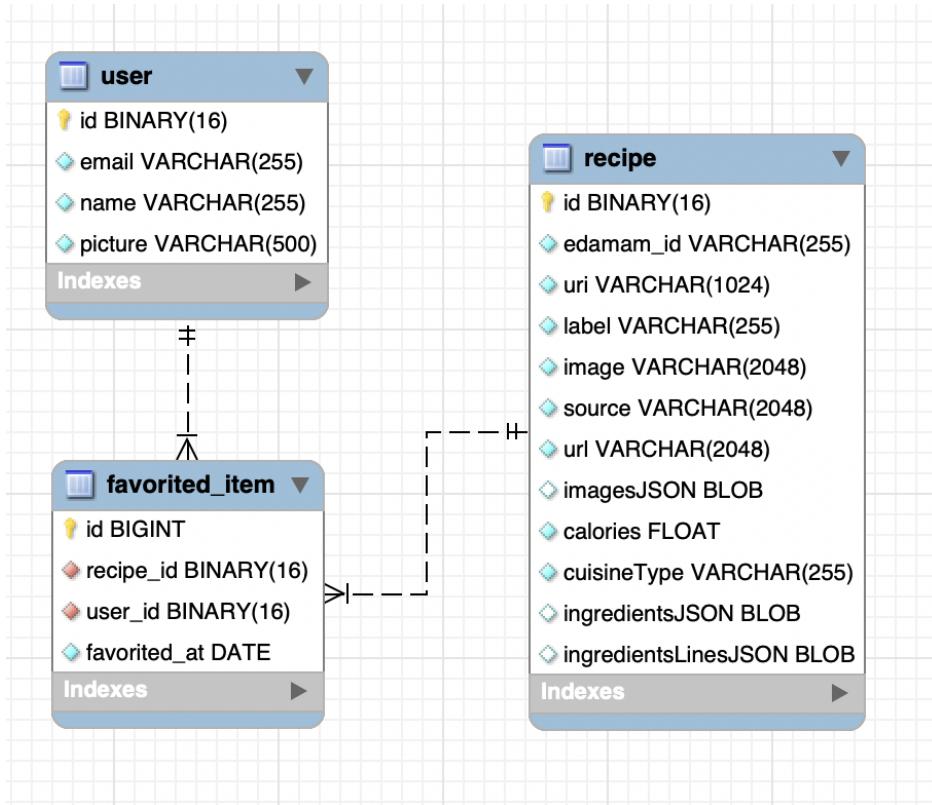
10:59 PM Create App Engine version kkoldaeva@csumb.edu cre...

Loading... searched up to Mar 31, 2022

Google App Engine versions dashboard.

FIGURE C2

Ultimate Recipe MySQL Database Schema and Application Level Data Models



Ultimate Recipe database schema.

FIGURE C3

The screenshot shows the VS Code interface with the following details:

- Title Bar:** types.ts — capstone-backend
- File Explorer:** Shows the project structure: src > server > models > types.ts.
- Editor:** The code editor displays the contents of types.ts:

```
29      You, last month | 1 author (You)
30      export interface Ingredient {
31          text?: string;
32          quantity?: number;
33          measure?: string;
34          food?: string;
35          weight?: number;
36          // shopping isle category
37          foodCategory?: string;
38          foodId?: string;
39          image?: string;
40      }
41
42      You, 3 weeks ago | 1 author (You)
43      export interface User {
44          email: string;
45          name: string;
46          picture: string;
47      }
48      You, last month | 1 author (You)
49      export interface FavoritedItem {
50          recipe: Recipe;
51          favoritedAt: Date;
52      }
```
- Bottom Status Bar:** feat/google-cloud-deploy* | Git Graph -- NORMAL -- | Ln 52, Col 1 | Spaces: 2 | UTF-8 | LF | {} TypeScript | ✓ Prettier

Application level types shared by frontend and backend.

FIGURE C4

Database Initialization

```
1 • CREATE TABLE IF NOT EXISTS recipe (
2     id binary(16) NOT NULL,
3     edamam_id varchar(255) NOT NULL,
4     uri varchar(1024) NOT NULL,
5     label varchar(255) NOT NULL,
6     image varchar(2048) NOT NULL,
7     source varchar(2048) NOT NULL,
8     url varchar(2048) NOT NULL,
9     imagesJSON blob,
10    calories float NOT NULL,
11    cuisineType varchar(255) NOT NULL,
12    ingredientsJSON blob,
13    ingredientsLinesJSON blob,
14    PRIMARY KEY (id)
15 );
16
17 • CREATE TABLE if not exists user (
18     id binary(16) NOT NULL,
19     email varchar(255) NOT NULL,
20     name varchar(255) NOT NULL,
21     picture varchar(500) NOT NULL,
22     PRIMARY KEY (id)
23 );
24
25 • CREATE TABLE if not exists favorited_item (
26     id bigint NOT NULL AUTO_INCREMENT,
27     recipe_id binary(16) NOT NULL,
28     user_id binary(16) NOT NULL,
29     favorited_at date NOT NULL,
30     FOREIGN KEY (recipe_id) REFERENCES recipe(id),
31     FOREIGN KEY (user_id) REFERENCES user(id),
32     PRIMARY KEY (id)
33 );
```

Ultimate Recipe tables creation script.

FIGURE C5

Development Tools

Ultimate Recipe 1.0.0 OAS3

favorites

GET /favorite List all favorited items

POST /favorite Add a favorited item

Parameters

No parameters

Request body application/json

Add recipe to favorites

Example Value | [Schema](#)

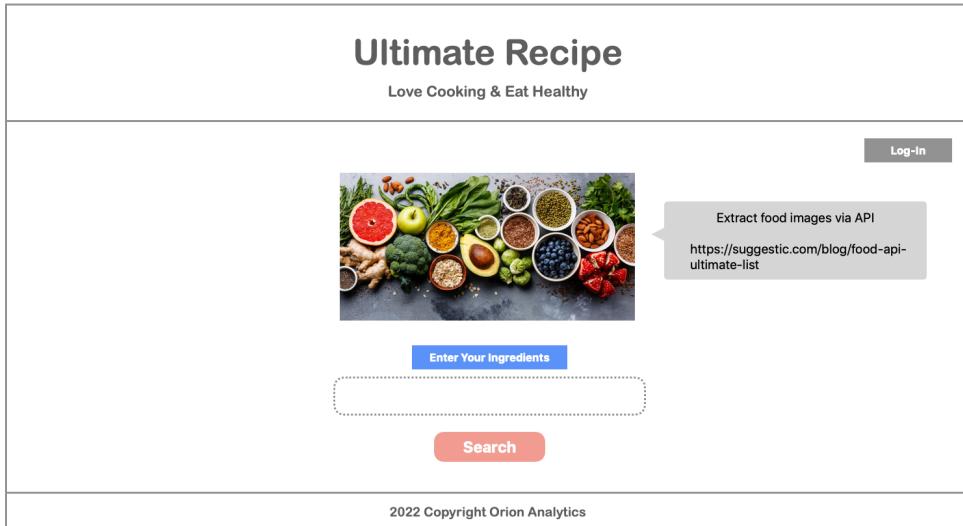
```
Recipe <pre>v {</pre>
  uri* <pre>> [...]</pre>
  label* <pre>> [...]</pre>
  image* <pre>> [...]</pre>
  source* <pre>> [...]</pre>
  url* <pre>> [...]</pre>
  images <pre>> [...]</pre>
  ingredientsLines <pre>> [...]</pre>
  ingredients <pre>> [...]</pre>
}</pre>
```

Open API Swagger UI web interface.

FIGURE C6

Initial UI/UX Design Proposals

Landing Page



The initial landing page design. This page allows end users to perform their search by using keywords.

FIGURE C7

Search Result Page

Ultimate Recipe

Love Cooking & Eat Healthy

[Main Page](#) [Log-in](#)



Baked Cheese & Herb Potato 

Ingredients: potatoes, herb, garlic



Buffalo Wing Potatoes 

Ingredients: potatoes, green onion



Chiles Rellenos Breakfast Bake 

Ingredients: potatoes, cheese

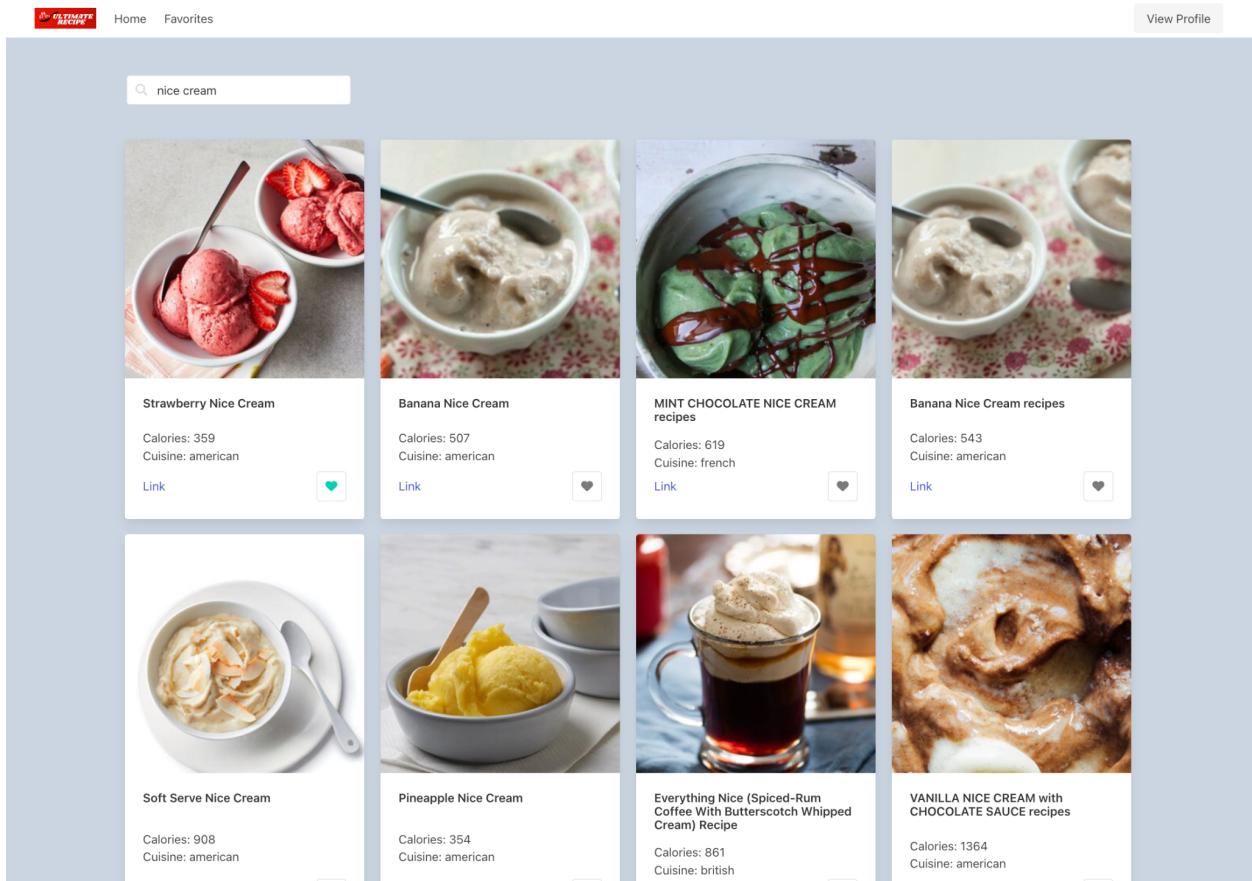
2022 Copyright Orion Analytics

 Food images to return in gallery

Once the users execute their search, the application returns query results related to the entered keywords.

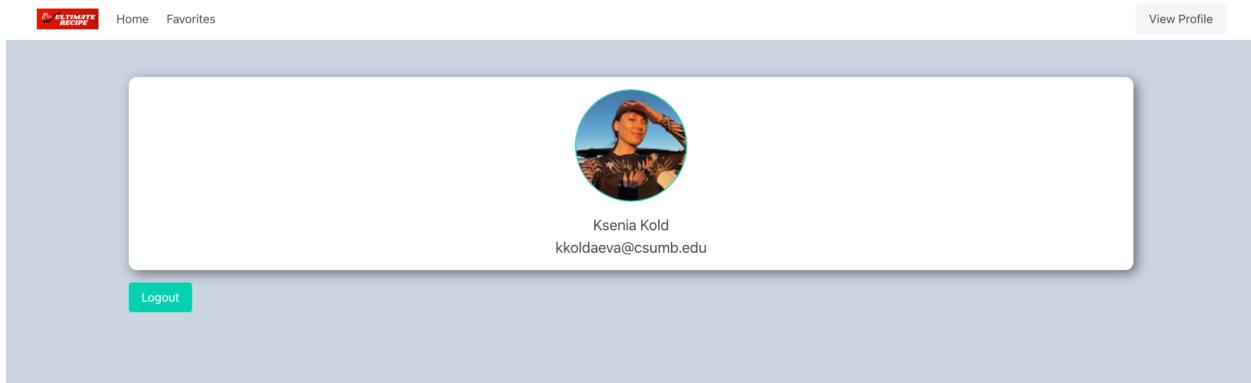
FIGURE C8

Ultimate Application Screenshots



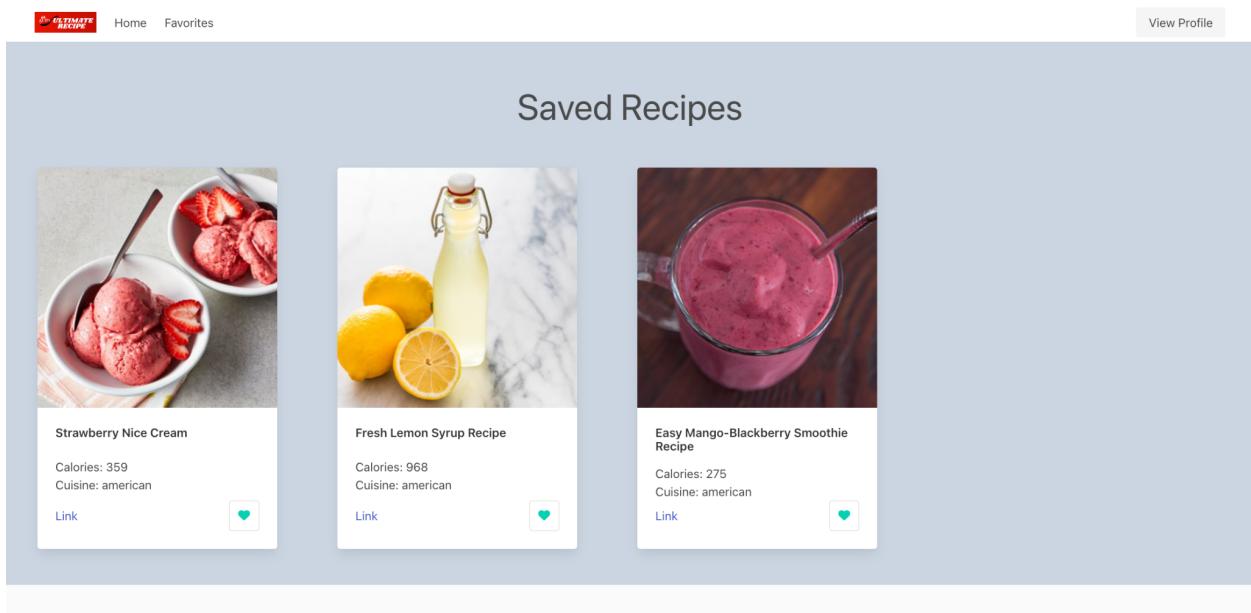
Search bar with the results grid featuring recipe cards for “Nice Cream”.

FIGURE C9



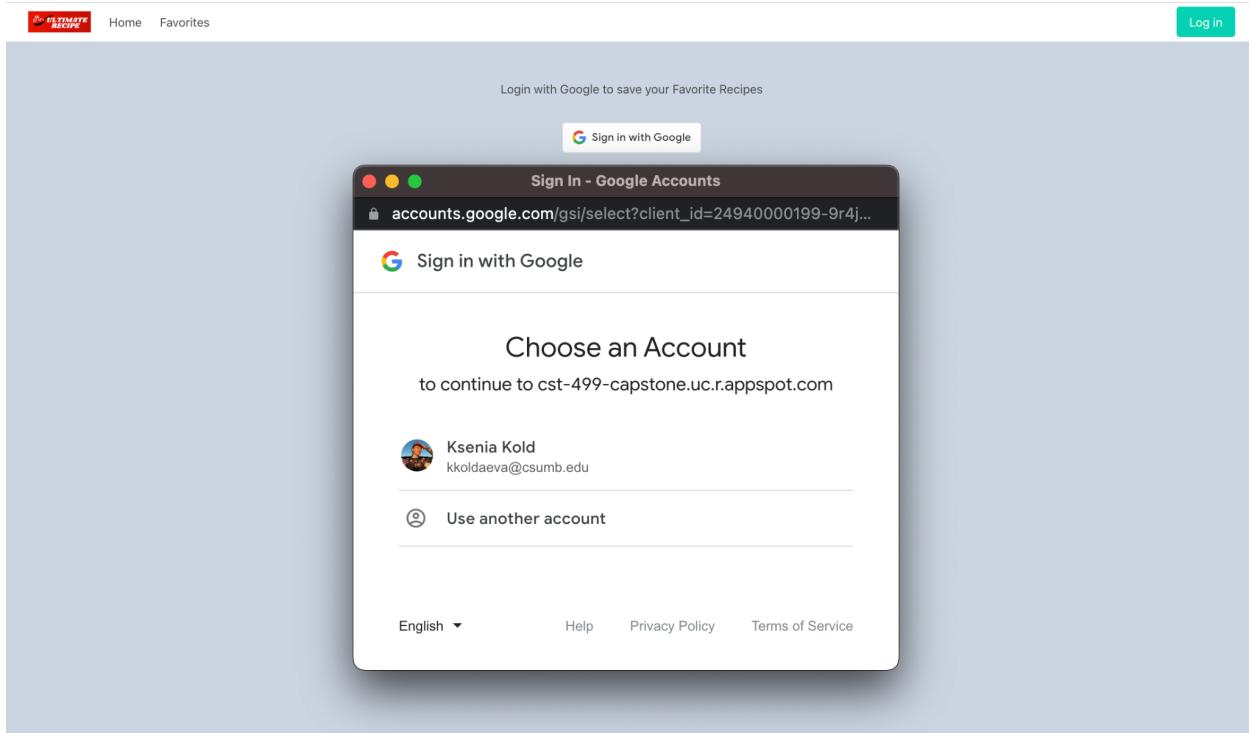
Profile page for a logged in user that displays user details.

FIGURE C10



User's "Saved Recipes" page.

FIGURE C11



Google OAuth Sign In flow.

FIGURE C12