

# FRIDGE-RAIDER

Lujein Nouelaty, Asha Mohamed, Rati Sudha and Sri Teja Challa



## INTRODUCTION

Fridge-Raider is a web-based application that allows users to access thousands of tasty recipes online based on the ingredients available in their fridge. The aim of our project is to make cooking easier for users with a busy lifestyle, or even wanting to search for healthy meals. It provides a lot of useful information including, calorie count, nutrition values, and dietary requirements giving users the opportunity to cater to their needs. The interface is user friendly and easy to use, with a robust search function that allows users to enter random ingredients they have in their kitchen, to then generating suggestions on what they can make with those ingredients. We decided to give the option to use the webpage as a guest or make an account where they can benefit from more features such as saving their favourites recipes and having additional information on the recipes. Our goal is to make searching for meals and cooking quick, simple, and more enjoyable.

## BACKGROUND

With the gradual incline in fast-paced lifestyles, people no longer have the time to plan for meals weeks in advance and are on a constant lookout for new and easy recipes, preferably all in one place. Along with the rise of living costs, what people need is a way to spend less on their grocery shopping and to ensure that no food goes to waste. Our target audience are working parents, students and those who live alone, as they are more likely than others to spend less time in the kitchen due to the time and effort it takes to prepare meals, from going to the supermarket for grocery shopping to the actual preparation time of the meal to be cooked. Naturally, these people find it easier to order a takeaway as it is a quicker option.

With that in mind, we wanted to create a software application program that allows a user to easily search for recipes by using the ingredients they have available at home. It therefore must allow the user to input the ingredients found in their fridge, bringing back several randomly generated recipes with their chosen ingredients. Users are then able to access these recipes by simply clicking on them and they will be redirected to the website of that recipe. The user will also be able to see the image of the recipes, the type of meal and cuisine it is, as well as any important nutritional information relevant to the users, such as dietary information (whether it's vegan, vegetarian or gluten free) and the calorie count. Our program calls on the Edamam API to extract all this information and returns it to the user.

We have also built a web application where visitors of our application can create an account to search for recipes using ingredients that they have at home. They can base their search on any dietary preferences (if any), their preferred meal type as well as a favourite cuisine. We have chosen the name of Fridge-Raider for our project and web application. Fridge for the ingredients found in a user's fridge, and raider for someone 'raiding' a fridge, looking for any scraps of food. Fridge-Raider also sounds like refrigerator, so it's also used as a play on word.

## SPECIFICATIONS AND DESIGN:

Technical requirements:

- Guest users can search for recipes based on ingredients
- Users can click to the recipe link and get the information
- Users can sign up and create an account based on user id, email address and password
- Registered user credentials must be logged into their account
- Registered users can search recipes based on ingredients
- Registered users can do nutrient analysis of their searched recipes
- Registered users can do health analysis of their searched recipes
- Registered users can save their favourite recipes, remove recipes and can view whenever they logged into their account.

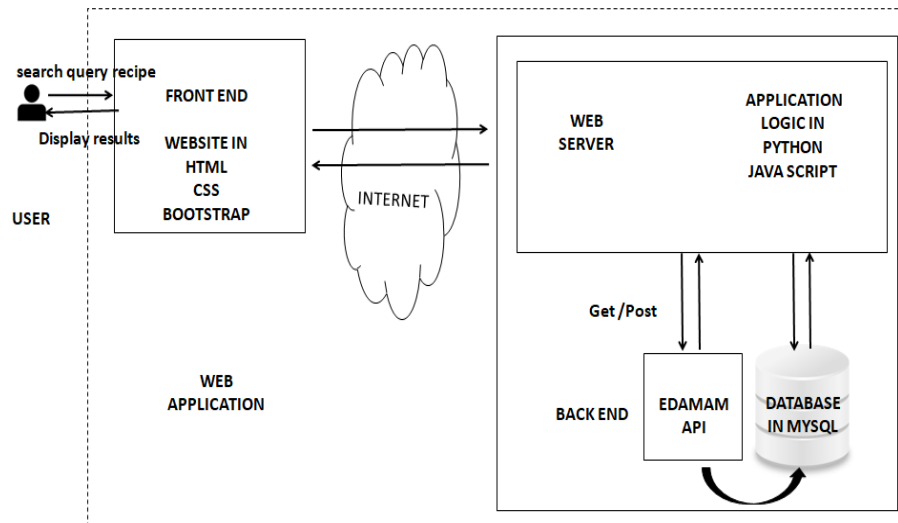
Non-technical:

- To provide users a feel-good interface where they can search
- To provide user smooth navigation through the website
- To provide users with appealing recipe images
- FAQ page has been provided for the users to understand the execution of the website
- We have followed the CFG guidelines, to avoid code duplication, code readability and consistency and also to organize them into a directory and sub-directories

## Design and architecture:

The diagram below illustrates the architecture of our project “fridge-raider”. It has 3 main components:

1. Front-end : For this part we have used HTML, CSS, Bootstrap
2. Back-end: For this part we have used Python and Javascript
3. Database: MySQL is used as a Database Management System for our project



## IMPLEMENTATION AND EXECUTION

### Development approach and team member roles

We used the agile development approach for the development of this project and we assigned ourselves the roles of scrum master (Rati), product owner (Lujein) and 2 developers (Asha, Sri). We also shared each other's responsibilities whenever required and thus, we all worked on the coding and at the end we tested our product.

### Tools and libraries

#### API:

Edamam API (recipe search, food database, Nutritional analysis): We have used Edamam API for recipes search and fetched recipes data to our website

#### Python-3:

- Object oriented programming: we have used class based code writing approach
- Exception handling: We have utilized exceptional handling technique in our code
- Flask (flash, render\_template, redirect) : We have used Flask framework for our project
- mysql-connector: It helped us create link between MySQL and using Python
- database-connection
- requests: The requests module we used to send HTTP requests.
- config: we used SECRET\_KEY to encrypt cookies and save send them to the browser
- bcrypt: We have used password hashing method for user password management

- unit test: In order to check the various functionality of the web application we have used unit test cases.
- Git & Git hub: We used Github repository for our project, so that it can accessible to all and for project submission

### **Web designing:**

- HTML: It has been used to create our website fridge-raider
- CSS: It has been for designing HTML documents
- Bootstrap: Bootstrap framework for developing this website.

### **Database:**

- MySQL: We have used MySQL workbench to create databases for users to save their credentials and to save favourite recipes

### **JavaScript:**

json: It stands for JavaScript Object Notation and it is language independent. In our project json helped us to sent data from Edmam API to a our website fridge-raider.

#### **STRENGTHS:**

- Good knowledge on OOP, classes and testing
- Agile development methods (Scrum and Trello board)
- Collaborative team effort to reach goal

#### **WEAKNESSES:**

- Lack of knowledge of HTML and CSS
- Understanding and using the API in detail

#### **OPPORTUNITIES:**

- Good use of Github, pull and push requests
- Evolving technologies and innovation

#### **THREATS:**

- Managing class homework, revision and project work
- Finding time suitable for everyone to attend regular meetings

- One of the challenges we faced was working on the front end, due to not having much experience on HTML and CSS. We had many ideas on how we could improve our webpage and make it more interactive for users like allowing them to get ingredients out of the fridge and inputting it into the search bar, and giving users more features to benefit from when they make an account with us so it can be user friendly. After discussing ways to do this, by researching and asking around, our lack of experience in the front end limited us to do so. Therefore, we implemented a unique idea on the fridge moving when you click on it so it can be more user friendly. We also didn't have enough time as we had other commitments and external factors we had to juggle alongside doing the project, homework's, and revision for the assessment.
- At the start of the project, our group had many ideas and we were struggling to choose one. We wanted to choose something that challenged us and allowed us to show our problem solving skills as well as something useful we can provide users knowledge with. We had topics to choose from and we decided to limit it to 5 which were: healthcare, media and entertainment, technology, hospitality and leisure, and banking. We all contributed and there was about 10 good project ideas and couldn't decide on what we want the project to be about. We then decided to all take vote for our top 3 and all our answers had 1 idea in common. That was fridge-raider.
- Another challenge we faced as a group was finding the right time for everyone to meet up. As we all had different schedules, with us working and having other commitments, it became a challenge for us. We were having meetings regularly, but not all team members were present at all times. With this, we still made sure we all communicated and worked together as a team by always catching up on anything we missed if we didn't attend any meetings. We also decided to divide the work so it was shared out equally and everyone had an input. We collaborated together if anyone was struggling and needed some extra support, allowing us to manage the workload. We asked around for what our strengths were and what we wanted to work on in the project, kept updating each other on how we were doing and what the progress of the project was.
- Looking for APIs at the start wasn't that challenging, we found many free good ones to use such as: Edamam, Spoonacular, TheMealDB. They all had everything we needed to start working on the project. We did want to use more than one API at first, however, we chose to work with Edamam as it had 3 APIs. These were Food Database API, Nutrition Analysis API, and Recipe Search API. Understanding the API and trying to implement it is something we struggled with as well as figuring out how to make it work through the code. We all chose 1 API to look into, started working on it by reading the documentation, creating an account, looking into how to get a response, and writing out the code. Reading and understanding the documentation to start implementing was a struggle, we needed certain information from one API to retrieve data from another API which meant all 3 were connected. There was 3 different APIs and we thought all 3 retrieved different things, to then finding out there was overlap and one of the APIs had most of the information we needed on it. So we used one API for our project as that provided us with what was needed.

We worked in an agile environment where we created a Trello board (<https://trello.com/b/LpYGXggR/cfg-project>) and split the project work into 4 Sprints. As a Product Owner, Lujain was able to update our Trello board to include what needs to be worked on during each Sprint and what has already been completed. This made working on the project much easier and

straightforward for us to follow as we had a set schedule, and each group member had goals they needed to achieve and tasks to work on. This guaranteed we were able to meet the project deadline.



We collaborated on the project through code review, where we set up weekly team meetings (and often these meetings took place twice or three times per week whenever needed) to go over codes we had separately worked on and to review these codes ensuring they were working properly and were doing what they were supposed to do. As a group, we also used an iterative approach while working on our project from the start. We discussed the main goal of our project and what our software must do. With that in mind, we were able to make improvements to our code along the way, tweaking a few issues which we did not like or which we wanted our software programme to do. For example, we decided that we only wanted certain data to be returned when searching for a recipe, and not everything from the API. We did this until we were satisfied with our end project. We also worked on our web application multiple times, changing it a number of times, such as the background, the pages we wanted to have and the colours and images used until we were all happy with it.

## TESTING AND EVALUATION



Why we do test in the first place:

1. **The need for less manual testing.** Because the test cases are pre-written, there is no real reason to conduct manual testing anymore. Most times, especially with large applications



must run through a lot of stages in the software to test if a certain feature works. This could be avoided with testing. Imagine having to use certain inputs every time your application runs. With testing this would be inputted automatically over each run.

2. **Allows for better structure of code.** It is of a best practice to create test cases before development. This forces us to think of the system before we start developing.
3. **Allows for faster testing in long run.** When coding, the implementation of a function can change however, the output remains the same. Instead of manually testing this every time a function's implementation changes, we can simply implement specific test cases that can run multiple test cases at a time.

There is a module in Python's standard library called unittest which contains tools for testing your code. Unit testing checks if all specific parts of the function's behaviour are correct, which will make integrating them together with other parts much easier.

To test that our application is working, we have decided to create a separate directory with a file to utilise the unittest module. First, we need to create a test file. Then import the unittest module, define the testing class that inherits from unittest.TestCase, and lastly, write a series of methods to test all the cases of your function's behaviour.

We have run test cases for the recipe search with ingredient which should return at least one recipe with the ingredient given.

Another unittest module which consists of test cases which has user tests which checks whether the given email is not in use, or it says unavailable email.

We also checked if the credential given by the user are registered in our database or not. Initially we had errors which says the password is hashed and could not be given but finally made few changes and made that work.

## CONCLUSION

Our aim was to build a software application that allows a user to find recipes based on the ingredients in their fridge. It should help people to avoid food wastage and save money. Indeed, our software can do this and more using the Edamam API. It shows users a variety of information with all recipes that it brings back, where users are then able to select the most suitable recipe for them. Our software application also uses a database to store users' favourite recipes so they can easily access them in the future. We were also able to build a web application using HTML and CSS which links to our backend development. Users can create an account for easy and customised searches. We believe that our software helps people living fast paced lifestyles by offering them an easy solution to meal planning and stops food wastage.

## References:

Image of Sprints used from <https://kruschecompany.com/agile-software-development/>

Image of testing used from <https://medium.com/coderbyte/python-testing-a8156d022eef>

Concepts of testing <https://realpython.com/python-testing/>