RECIPEPAD

Project Report

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT

FOR

Major Project(PRIT-103)

BACHELOR OF TECHNOLOGY

INFORMATION TECHNOLOGY

SUBMITTED BY :-
Shivay Bhandari (1905398)
Tarun Sehgal (1905407)
Raghav Malhotra (1905381)
Branch : D4-IT-B2

Guru Nanak Dev Engineering College
Ludhiana-141006

Abstract

We all want to make projects that make things better, avoiding the bad things that lurk in the shadow of failed efforts. To succeed, we need discipline when the project is designed and built. Many individuals and companies still develop software haphazardly, even as they build systems to service the most advanced technologies of the day. As a result, the quality of the software that we produce suffers and bad things happen.

Food is one of the main human needs. Food is the sole source of energy. People's innovation in food has taken different forms from one place to another. For making a food dish a person needs raw materials and the recipe. A recipe is a set of instructions that tells us how to prepare food from scratch. A recipe includes its name, the series of steps to be followed, and the time taken to prepare the dish.

The emphasis of this report is to document the important concepts and techniques used for the successful development of this project.

I do hope fervently that, through this report, the readers will get a real picture of what the training and project are all about.

ACKNOWLEDGEMENT

We are highly grateful to Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College, Ludhiana, for providing this opportunity to carry out the major project work at Guru Nanak Dev Engineering College, Ludhiana.

The constant guidance and encouragement received from Dr. Kulvinder Singh Mann H.O.D. IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Er. Navdeep Kaur Deol, without his wise counsel and able guidance, would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of Information Technology Department of GNDEC for their intellectual support throughout the course of this work.

Finally, We are indebted to all whosoever have contributed in this report work.

Shivay Bhandari (1905398)

Tarun Sehgal (1905407)

Raghav Malhotra (1905381)

List of Figures

Contents

1 INTRODUCTION

1.1 Introduction to Organization

Guru Nanak Dev Engineering College was established by the Nankana Sahib Education Trust. NSET was founded in memory of the most sacred temple of Nankana Sahib, the birthplace of Guru Nanak Dev Ji. Shiromani Gurudwara Prabandhak Committee, Amritsar, a premier organization of universal brotherhood, was the main force behind the mission of ”Removal of Economic Backwardness through Technology”. With this mission, a Polytechnic was started in 1953 and Guru Nanak Dev Engineering College was established in 1956 and is affiliated with the Punjab Technical University. The college offers undergraduate and postgraduate programs in various fields of engineering, as well as a Ph.D. program. Some of the departments at the college include Computer Science and Engineering, Electrical and Electronics Engineering, Mechanical Engineering, and Civil Engineering. The college has a number of modern facilities, including well-equipped laboratories, a library, and sports facilities. The faculty at Guru Nanak Dev Engineering College is highly qualified and experienced, and the college is known for providing a high-quality education to its students. The Trust deed was registered on 24th February 1953 with a commitment by The Nankana Sahib Education Trust to uplift the vast weaker section of Indian polity comprising Rural India by admitting 70% students every year from Rural Areas. This commitment was made to the nation on 8th April 1956. The day when the foundation stone of the College Building was laid by the Late Dr. Rajendra Prasad Ji, the First President of India. Nearly 10,000 graduate and 3000 Post Graduate Engineers have passed out from this college during the last 50 years and are at present successfully employed in India abroad. The college is now ISO 9001-2008 Certified, NBA accredited, and has signed MoU with IOWA University [USA] for the exchange of students and faculty.

1.2 Introduction to Project

Food is one of the main human needs. Food is the sole source of energy. People’s innovation in food has taken different forms from one place to another. For making a food dish a person needs raw materials and the recipe. A recipe is a set of instructions that tells us how to prepare food from scratch. A recipe includes its name, the series of steps to be followed, and the time taken to prepare the dish.

As a human, our minds cannot remember all the recipes for different dishes so we generally make notes. But in today’s world as everything is available on the internet. But the problem exists what is the recipe for a limited number of ingredients? There are apps that feature searches for recipes based on their names. But what happens if we can get ingredients from a nearby store? This is where our solution comes into the picture. In today’s world as everyone is conscious about their health so we have the feature of a calorie meter which will keep a track of the amount of calories intake for a particular dish on the basis of raw material.

The idea is to be implemented in the form of a web app that has the feature to search for recipes based on the presence of raw materials. The system suggests recipes according to the material available to the person. From different recommendations, a person can select the dish on the basis of taste and health conditions.

Users can input the ingredients they have on hand and the tool will return a list of recipes that include those ingredients. This can be a useful feature for people who want to cook with ingredients they already have in their kitchen, or for those who want to use up ingredients before they go bad.

Overall, it seems like a useful and convenient tool for people who love to cook and are looking for new recipe ideas.

1.3 Project Category

The project falls under the category of Application and System development. It's basically a web band project and also includes the internet for requests to servers and receiving data in the form of response. The project involves the development of a standalone application or the integration of an application with other systems. This project includes hardware components such as servers and networking equipment (hosted on the cloud), as well as software components such as operating systems, databases, and applications (hosted on the cloud). The project also involves the use of agile methodologies to ensure that the project is delivered on time and within budget. In this project, a user-friendly interface is provided to search for recipes based on the presence of raw materials. The system suggests recipes according to the material available to the person. From different recommendations, a person can select the dish on the basis of taste and health conditions.

1.4 Objectives of Project

1. To find the recipe of the dish based on selected ingredients.
2. To count the calories of food being prepared.
3. To plan meals according to diet preference.
4. To get random recipes.

1.5 Identification/Reorganization of Need

- **Improvement of Student Performance:** Most of the recipe management applications contain only those recipes specified by a cook and are uploaded by the admin. The users are allowed to browse and view the recipes. The present application does not clearly provides the number of calories in the particular food item.
- **Effective performance:** Our web application not only provides the users with the recipe of a particular dish while entering the ingredients only but also provides the calorie count of the particular dish. The user can monitor the nutrition needs of himself/herself. Our web application also does not provide a limited number of dishes, there are a number of dishes in our web application, because of the usage of the API (Spoonacular API).

- **Greater economic efficiency:** We may achieve greater economic efficiency (lower cost) and more rapid exchange (high speed accelerated or real-time interaction) with the help of this project.

1.6 Scope Of The Project

The scope of the recipe search project includes the development of a tool that allows users to search for recipes and also based on ingredients. The tool will include a database of recipes, each with a list of ingredients and instructions for preparation. Users will be able to search for recipes by inputting one or more ingredients that they have on hand. The tool will return a list of recipes that include those ingredients. The tool will display all the external tags such as vegetarian/non-vegetarian, contains-gluten/not-contains-gluten, vegan/non-vegan, etc. The tool will be available on a website and/or mobile. The tool may include additional features, such as the ability to save favorite recipes.

1.7 Innovation

One innovation of the recipe search project is the use of ingredient-based search functionality. This allows users to find recipes based on the ingredients they have on hand, rather than searching for recipes based on the dish or cuisine. This can be a useful feature for people who want to use up ingredients they already have in their kitchen, or for those who want to save money by avoiding buying new ingredients. Additionally, the ingredient-based search feature may be able to suggest recipes that use up a large quantity of a particular ingredient, which can be helpful for reducing food waste. Another possible innovation of the project could be the inclusion of additional features that enhance the user experience, such as the ability to save favorite recipes. These features can make it easier for users to plan and prepare meals, and can also help them discover new recipe ideas.

2 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

2.1 Feasibility Study

As the name indicates, the feasibility study aims to assess the proposed capstone project's practicality. This capstone aims to develop an interface that is used to elevate users' kitchen skills, streamline grocery shopping, and minimize food wastage. The project has a function to find recipes with ingredients that match what is owned by the user. This is a personalized website that is useful to those who even don't know about cooking. The project also has a feature to find the recipes according to health and diet preferences which also helps to measure the calorie count. This website provides an easy interface for users to search for recipes.

1. **Economic Feasibility :-** Economic feasibility refers to the potential financial benefits and costs of a project, and it is an important factor to consider when determining the overall feasibility of the recipe search project. This project uses ReactJS, Golang, and Tailwindcss. All the languages used are open-source, so the development cost is zero. To deploy needs to host on a server and data should be stored in the cloud or server. As the project is deployed on the cloud, the cloud maintenance cost should be included.

2. **Technical Feasibility :-** This project is technically feasible as the technology to be used is easily available and open source. The database is collected from an external API. The project is a website, so it can be viewed on a laptop as well as mobile. The technology and related dependencies are up to date with current technical requirements and provide the best possible way for implementation of the project.

3. **Operational Feasibility :-** The project is operationally feasible as the RecipePad is easily customizable to meet the requirements of the users. The project being open source provides horizons to continuous improvements which are beneficial for the users. The project includes normal recipe search, random recipe search, and ingredient recipe search. The project also contains the feature to save a recipe as a favorite so that the user can later see all the favorites.

2.2 System Requirement Specification

This document is meant to delineate the features of RecipePad, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

- **Performance Requirement:** If the system is not connected, it must not add more than two seconds to the time it takes to accomplish an action. There must be no more than a ten-second delay in the logging of researcher data to the research center. The speed with which directives are provided to the system will be affected by the efficiency of the software code. For the proper functioning of the project, only a device with a web browser is required. If the cloud servers on which the project is hosted are working smoothly, it would be a smooth experience for the user.

- **Maintainability Requirement:** The system is as simple to use as feasible, with all the capabilities accessible. The only maintenance required for the recipe search project is the renewal of cloud hosting services, as the project code is hosted on these cloud servers.

- **Security Requirement:** As the system is meant to run on a network like the internet, there are security concerns connected with utilizing it. When evaluating the system, the user must ensure that intruders, such as hacker attempts and third-party invasions, are prevented from gaining access.

3 SYSTEM DESIGN

3.1 Design Approach

In this project, we have adopted the Single Page Application (SPA) approach which is a web app implementation that loads only a single web document, and then updates the body content of that single document via JavaScript APIs such as XMLHttpRequest and Fetch when different content is to be shown. This is in contrast to traditional web applications, which typically require the browser to reload the page from the server each time the user takes an action. One of the key benefits of SPAs is that they provide a more seamless and responsive user experience since the page does not need to be reloaded each time the user takes an action. SPAs are typically built using JavaScript and a front-end framework such as React, Angular, or Vue.js. They rely on making asynchronous requests to a server to retrieve data and update the UI, rather than loading new pages from the server. Overall, SPAs are a popular choice for building modern web applications that require a fast and responsive user experience and are used by a wide range of companies and developers. This can make the application feel more like a native desktop or mobile app, rather than a web page. First, the user is a visitor to our application. After selecting that the user wants to perform a recipe search or ingredient search, recipes can be viewed. After that, they can add recipes to favorites or remove them from favorites.

3.2 Methodology

The project is divided into five modules:

1. **Normal Recipe Search** :

Here the user can search for the recipe name which is a traditional way of remembering recipes. The user will search for recipes like pizza, butter chicken, etc. The user inputs their search criteria (such as ingredients, dish type, cuisine, and dietary restrictions) into the search form on the website or app. The search algorithm processes the user's query and searches the recipe database for matches. The tool returns a list of recipes that match the search criteria. The list may be sorted by relevance, popularity, or some other criterion. The user can browse through the list of recipes and select one to view the full recipe and instructions.

2. **Input ingredients** :

Here, we input the ingredients that are available. On the basis of these ingredients, the model will find different recipes. The user inputs a list of ingredients they have on hand into the search form on the website or app. The search algorithm processes the user's query and searches the recipe database for matches. The tool returns a list of recipes that include the specified ingredients. The list may be sorted by relevance, popularity, or some other criterion. The user can browse through the list of recipes and select one to view the full recipe and instructions.

3. **Random recipe** :

In this module, the user can find random recipes, if no recipes are currently in the mind of the user. The user blank searches in the recipe search column. The search algorithm selects a random recipe from the recipe database. The tool displays the selected recipe and its instructions to the user.

4. **Favourite Recipe** :

The recipe that the user finds useful can save as a favorite recipe which is stored in the device's memory. The user views a recipe and decides that they want to save it as a favorite. The user selects the "add to favorites" option, which is a button on the recipe page. The tool adds the recipe to the user's list of favorite recipes, which is typically stored in the device's memory. The user can view their list of favorite recipes by selecting a "favorites" option on the website.

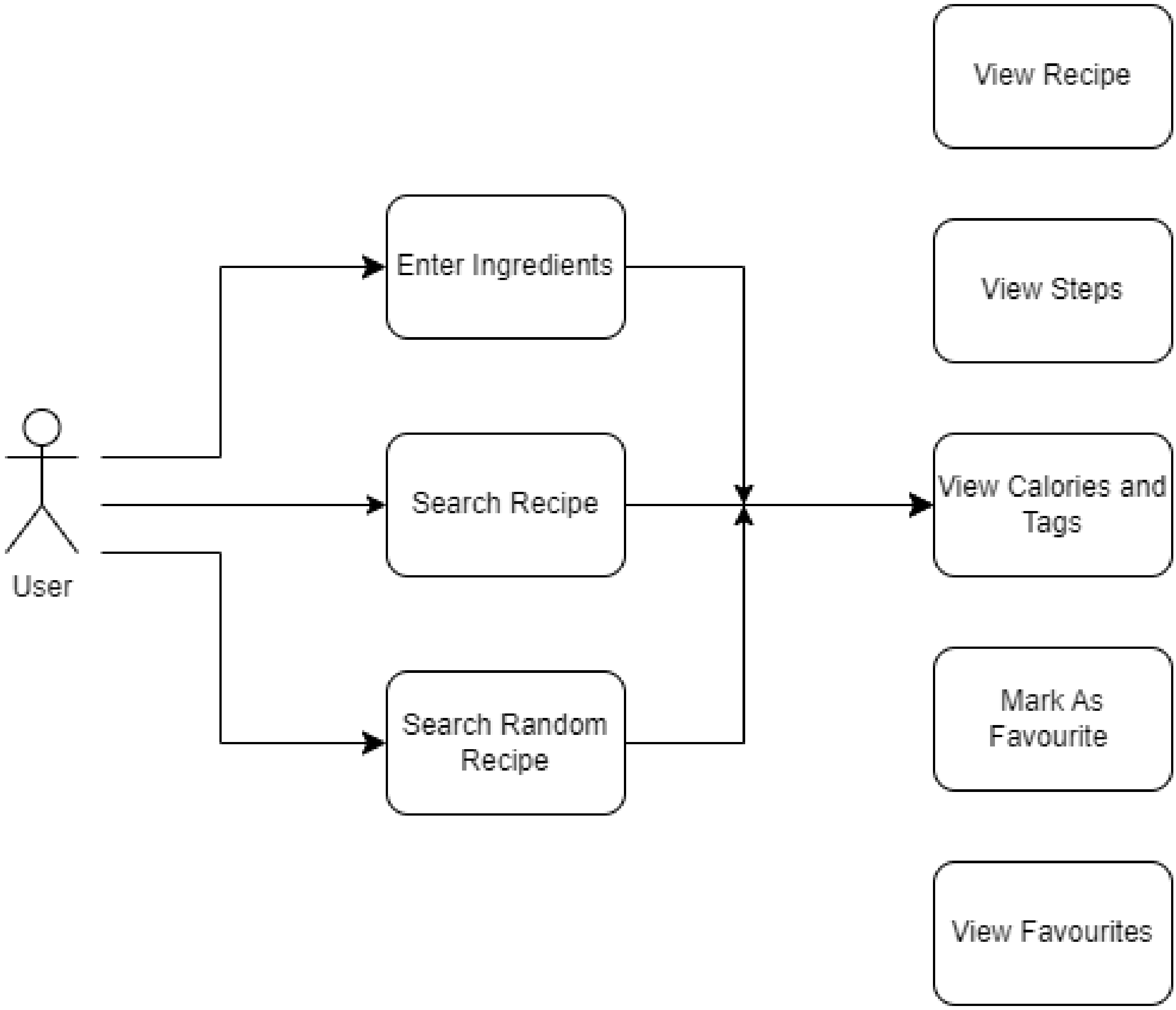


Figure 3.1: Methodology

4 IMPLEMENTATION, TESTING AND MAINTENANCE

4.1 Introduction to Languages, Tools Technologies used for Implementation

- **ReactJS:** React.js, more commonly known as React, is a free, open-source JavaScript library. It works best to build user interfaces by combining sections of code (components) into full websites. Originally built by Facebook, Meta and the open-source community now maintain it. One of the good things about React is that you can use it as much or as little as you want! For example, you can build your entire site in React or just use one single React component on one page. One of the key features of React is its ability to declaratively describe user interfaces using components. A component is a self-contained piece of code that describes a part of a user interface. It receives input in the form of props (short for ”properties”) and returns a description of the UI to be rendered, using JSX, a syntax extension for JavaScript. React allows developers to create reusable UI components, which can significantly speed up the development process and make it easier to maintain large codebases. It also uses a virtual DOM (a lightweight in-memory representation of the actual DOM), which makes it efficient at rendering updates to the user interface. Another key feature of React is its use of a unidirectional data flow, also known as the ”one-way binding” pattern. This means that the state of a parent component can only be passed down to its child components, and not the other way around. This helps to prevent the kind of confusing, hard-to-debug problems that can arise when the state of a UI is modified in multiple places. Overall, React is a powerful and popular tool for building efficient and scalable user interfaces, and is widely used by companies such as Facebook, Netflix, Airbnb, and many others.

- **Golang:** Go (also called Golang or Go language) is an open source programming language used for general purpose. Go was developed by Google engineers to create dependable and efficient software. Most similarly modeled after C, Go is statically typed and explicit. One of the main goals of Go is to provide a simple, efficient, and reliable language for systems programming. It has a small runtime and a fast compiler, making it well-suited for building scalable, high-performance servers and other infrastructure tools. Go has a strong emphasis on readability and simplicity. It has a straightforward syntax, clear naming conventions, and strong support for code organization using packages. It also has a growing standard library that provides a wide range of useful functions for tasks such as networking, cryptography, and data manipulation. In addition to its use in server-side development, Go is also used in the development of command-line tools, distributed systems, and even mobile applications. It has a growing ecosystem of libraries and tools, and an active and welcoming developer community. Overall, Go is a powerful and popular language for building reliable and scalable systems, and is used by companies such as Google, Uber, and Dropbox, among others.

- **API:** APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. For example, the weather bureau's software system contains daily weather data. The weather app on your phone "talks" to this system via APIs and shows you daily weather updates on your phone. API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses. Their API documentation contains information on how developers are to structure those requests and responses. APIs are often used to expose the functionality of a software library or framework to other developers, making it easy for them to build applications that make use of that functionality. They are also used to allow different software systems to communicate with each other, such as when a web application needs to retrieve data from a remote server or when a mobile app needs to access a backend service. APIs are typically defined using a specific protocol, such as HTTP, which specifies the way in which two systems should request and exchange data. They may also be accompanied by documentation, which explains how to use the API and provides examples of how to make different types of requests. Overall, APIs are an essential tool for enabling communication and integration between different software systems, and are widely used in the development of modern applications.

- **GitHub:** GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration. GitHub's interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line. At its core, GitHub is a Git repository hosting service, which means that it provides a place for developers to store and manage their Git repositories (collections of code and other assets). It also offers a range of tools and features that make it easier for developers to collaborate on software projects, including version control, project management, code review, and more. One of the key features of GitHub is its support for distributed version control. This means that developers can work on their own copies of a codebase, or "forks," and then submit changes, or "pull requests," to the main project for review and integration. This makes it easy for developers to contribute to open source projects, and for teams to collaborate on private projects. GitHub also has a large and active community of developers, who use it as a platform for sharing and discussing code, as well as for finding and collaborating on open source projects. It is a popular destination for developers, with over 50 million users and more than 100 million repositories. Overall, GitHub is a powerful and widely-used platform for hosting, collaborating on, and sharing software projects.

- **VS Code:** Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript. VS Code is built on top of the Electron framework, which allows it to run web technologies such as HTML, CSS, and JavaScript in a standalone application. It is a highly extensible code editor, with a large library of extensions that add support for additional languages, debugging tools, and other functionality. VS Code has a number of features that make it particularly well-suited for development work. It has strong support for multiple programming languages, including auto-completion and debugging tools. It also has a built-in terminal and integrations with Git and other version control systems, making it easy to manage code repositories and track changes. Overall, VS Code is a popular and powerful code editor that is widely used by developers for a variety of programming tasks.

- **DigitalOcean:** DigitalOcean is a cloud hosting provider that offers cloud computing services and Infrastructure as a Service (IaaS). Known for pricing and scalability, teams can deploy on DigitalOcean in seconds. One of the main services offered by DigitalOcean is virtual private servers, or "droplets," which are scalable and customizable Linux-based virtual machines that can be used to host websites, applications, and other types of services. DigitalOcean also provides a range of other cloud computing services, including object storage, managed databases, and load balancers, as well as a range of tools and features for managing and deploying applications. One of the key benefits of using DigitalOcean is its simplicity and ease of use. It has a user-friendly control panel and a range of tutorials and documentation to help users get started with its services. It also has a strong focus on developer experience, with a range of tools and features that are designed to make it easy for developers to build, test, and deploy applications. Overall, DigitalOcean is a popular and reliable cloud computing platform that is widely used by developers and businesses for hosting and managing a wide range of applications and services.

- **Vercel:** Vercel is the platform for frontend developers, providing the speed and reliability innovators need to create at the moment of inspiration. We enable teams to iterate quickly and develop, preview, and ship delightful user experiences. Vercel provides an easy-to-use experience for developers and makes it simple to deploy sites that are both fast and delightful for users. It also makes it easy for front-end teams to develop, preview and ship delightful user experiences where performance is the default. One of the key features of Vercel is its focus on simplicity and ease of use. It has a user-friendly control panel and a range of tools and features that make it easy for developers to build and deploy applications, including support for popular frameworks such as Next.js and Gatsby. Vercel also has a strong emphasis on performance and scalability. It uses a global content delivery network (CDN) to ensure that applications and websites are served quickly and reliably, and has automatic scaling capabilities to handle high traffic volumes. In addition to hosting web applications and websites, Vercel also offers a range of other services, including custom domains, SSL certificates, and support for custom serverless functions. Overall, Vercel is a popular and powerful platform for hosting and deploying web applications and websites, and is used by a wide range of companies and developers.

4.2 Coding Standards of Language used

- **Style Guide:** This defines the rules for laying out and organizing code, or at least the parts that js format doesn't handle for you. The style guide also specifies how identifiers are formatted: camelCase, using underscores, etc.

- **Documentation Guide:** This tells you everything you need to know about what goes inside comments. Both doc comments and regular, run-of-the-mill code comments.

- **Usage Guide:** This teaches you how to make the best use of language features to implement behavior. If it's in a statement or expression, it's covered here.

- **Design Guide:** This is the softest guide, but the one with the widest scope. It covers what we've learned about designing consistent, usable APIs for libraries: If it's in a type signature or declaration, this goes over it.

- A library member is a top-level field, getter, setter, or function. Basically, anything at the top level that isn't a type.

- A class member is a constructor, field, getter, setter, function, or operator declared inside a class. Class members can be an instance or static, abstract or concrete.

- A member is either a library member or a class member.

- A variable, when used generally, refers to top-level variables, parameters, and local variables. It doesn't include static or instance fields.

4.3 Testing Techniques and Testing Plans

To check the working of all the functionalities, various tests have been performed. We checked the reliability of all of the functions. The test is built on the user search. The user test proved that a recipe can be generated, favorites can be marked, viewing the favorites where all the favorites and displayed.

5 RESULTS AND DISCUSSIONS

Brief Description of Various Modules of the system:

- **Normal Recipe Search:** Here the user can search for the recipe name which is a traditional way of remembering recipes. The user will search for recipes like pizza, butter chicken, etc. The user inputs their search criteria (such as ingredients, dish type, cuisine, and dietary restrictions) into the search form on the website or app. The search algorithm processes the user's query and searches the recipe database for matches. The tool returns a list of recipes that match the search criteria. The list may be sorted by relevance, popularity, or some other criterion. The user can browse through the list of recipes and select one to view the full recipe and instructions.

- **Ingredient Recipe Search:** Here, we input the ingredients that are available. On the basis of these ingredients, the model will find different recipes. The user inputs a list of ingredients they have on hand into the search form on the website or app. The search algorithm processes the user's query and searches the recipe database for matches. The tool returns a list of recipes that include the specified ingredients. The list may be sorted by relevance, popularity, or some other criterion. The user can browse through the list of recipes and select one to view the full recipe and instructions.

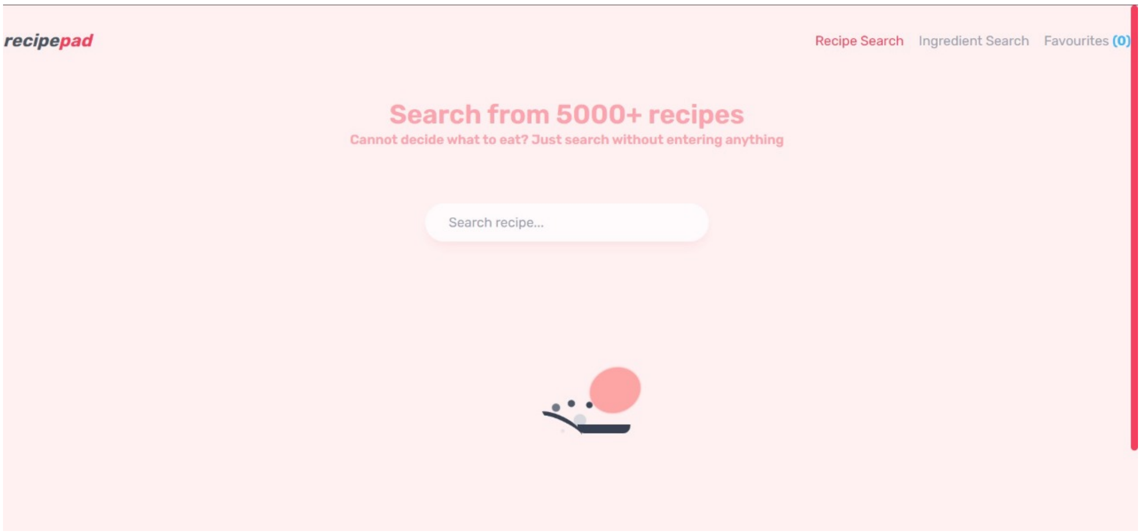
- **Random Search:** In this module, the user can find random recipes, if no recipes are currently in the mind of the user. The user blank searches in the recipe search column. The search algorithm selects a random recipe from the recipe database. The tool displays the selected recipe and its instructions to the user.

- **Mark Favourite Recipe:** The recipe that the user finds useful can save as a favorite recipe which is stored in the device's memory. The user views a recipe and decides that they want to save it as a favorite. The user selects the "add to favorites" option, which is a button on the recipe page. The tool adds the recipe to the user's list of favorite recipes, which is typically stored in the device's memory. The user can view their list of favorite recipes by selecting a "favorites" option on the website.

- **View Favourites:** The recipe that the user finds useful can save as a favorite recipe which is stored in the device's memory. The user views a recipe and decides that they want to save it as a favorite. The user selects the "add to favorites" option, which is a button on the recipe page. The tool adds the recipe to the user's list of favorite recipes, which is typically stored in the device's memory. The user can view their list of favorite recipes by selecting a "favorites" option on the website.

- **Calorie Meal Preference:** In this module user can select the recipe on the basis of the calorie count and meal preference like gluten-free, keto, carbohydrates etc.

5.1 User Interface Representation

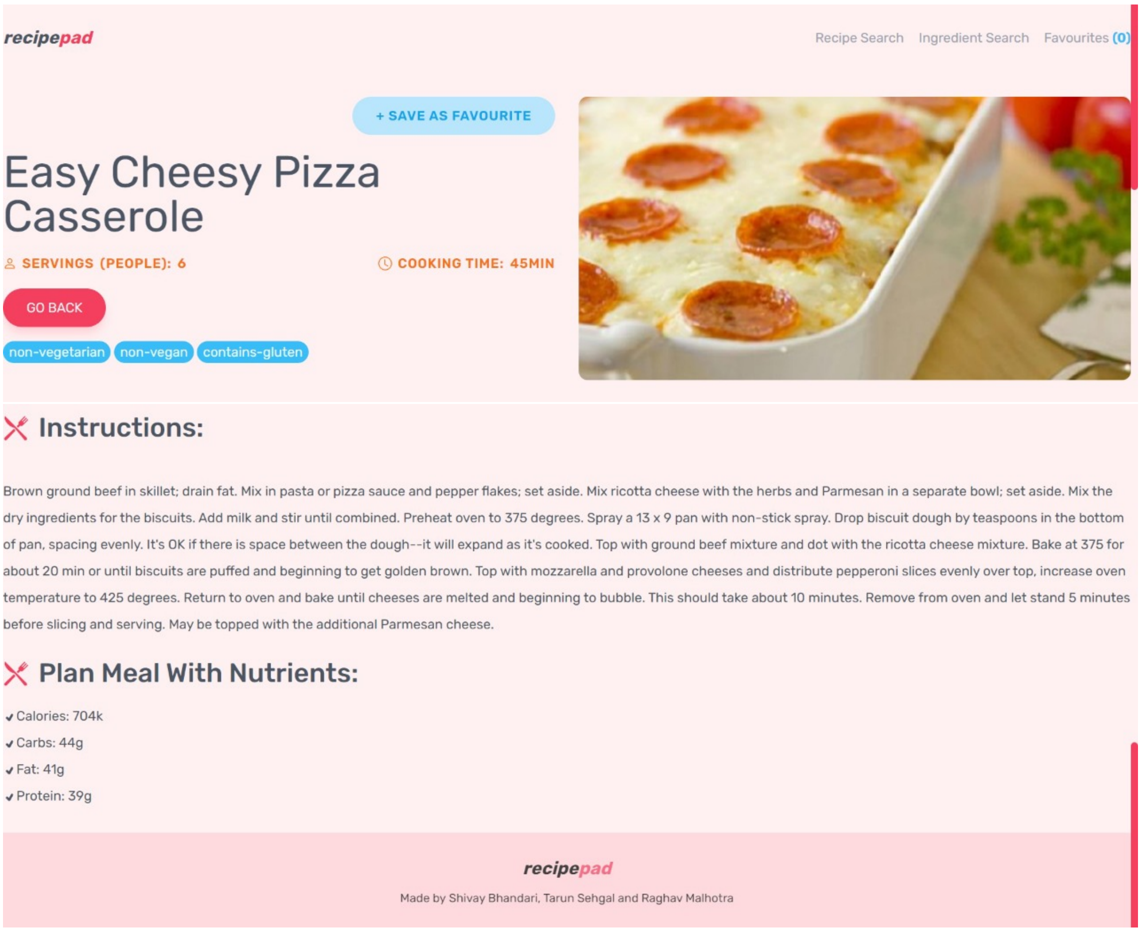


recipepad

Search from 5000+ recipes

Cannot decide what to eat? Just search without entering anything

Figure 5.1: Main interface



Easy Cheesy Pizza Casserole

SERVINGS (PEOPLE): 6

COOKING TIME: 45MIN

Instructions:

Brown ground beef in skillet; drain fat. Mix in pasta or pizza sauce and pepper flakes; set aside. Mix ricotta cheese with the herbs and Parmesan in a separate bowl; set aside. Mix the dry ingredients for the biscuits. Add milk and stir until combined. Preheat oven to 375 degrees. Spray a 13 x 9 pan with non-stick spray. Drop biscuit dough by teaspoons in the bottom of pan, spacing evenly. It's OK if there is space between the dough--it will expand as it's cooked. Top with ground beef mixture and dot with the ricotta cheese mixture. Bake at 375 for about 20 min or until biscuits are puffed and beginning to get golden brown. Top with mozzarella and provolone cheeses and distribute pepperoni slices evenly over top, increase oven temperature to 425 degrees. Return to oven and bake until cheeses are melted and beginning to bubble. This should take about 10 minutes. Remove from oven and let stand 5 minutes before slicing and serving. May be topped with the additional Parmesan cheese.

Plan Meal With Nutrients:

- ✓ Calories: 704k
- ✓ Carbs: 44g
- ✓ Fat: 41g
- ✓ Protein: 39g

recipepad

Made by Shivay Bhandari, Tarun Sehgal and Raghav Malhotra

Figure 5.2: Recipe information

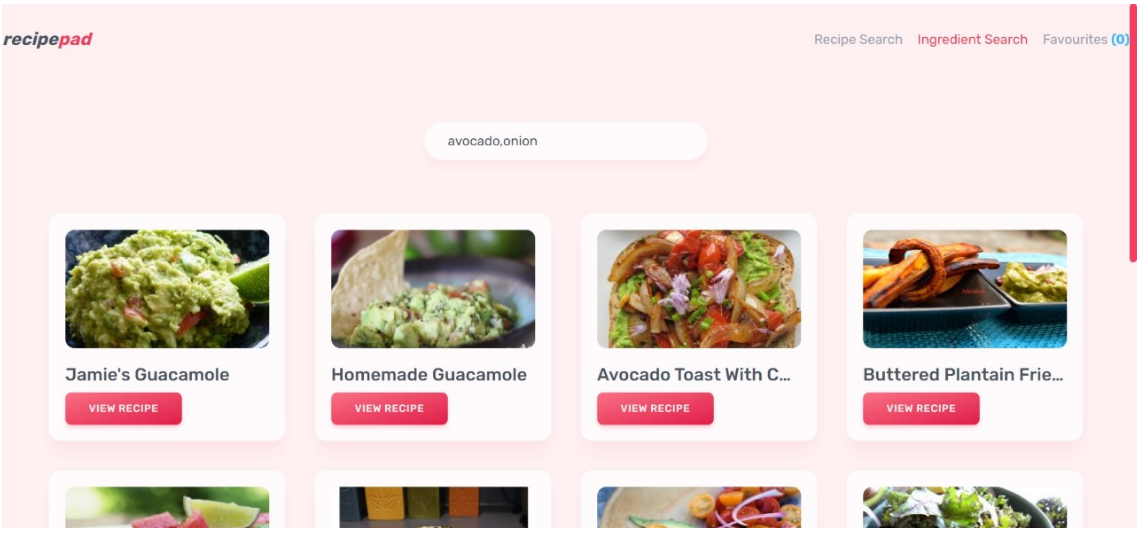


Figure 5.3: Get recipe from ingredients

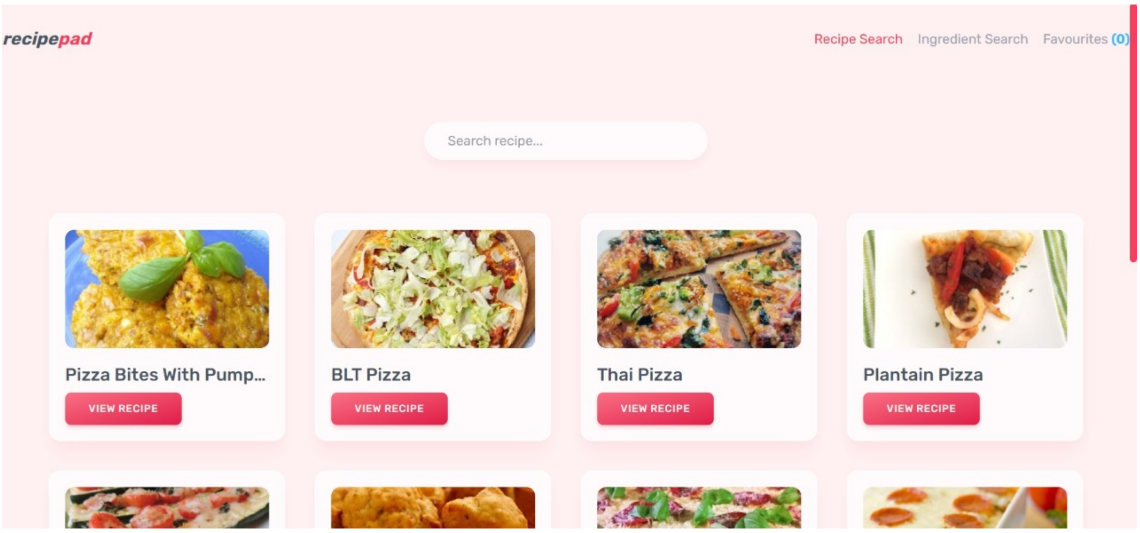


Figure 5.4: Random search

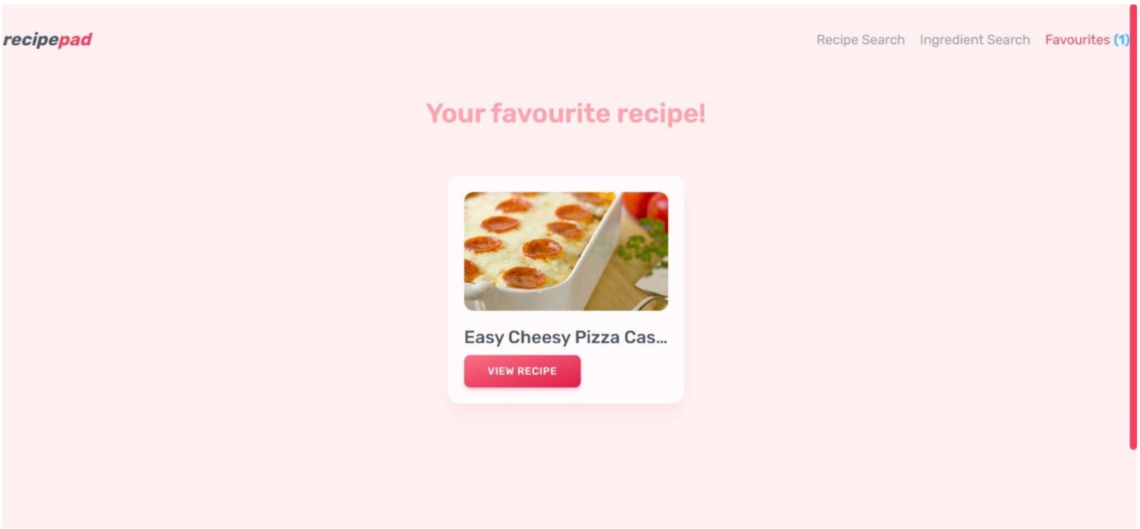
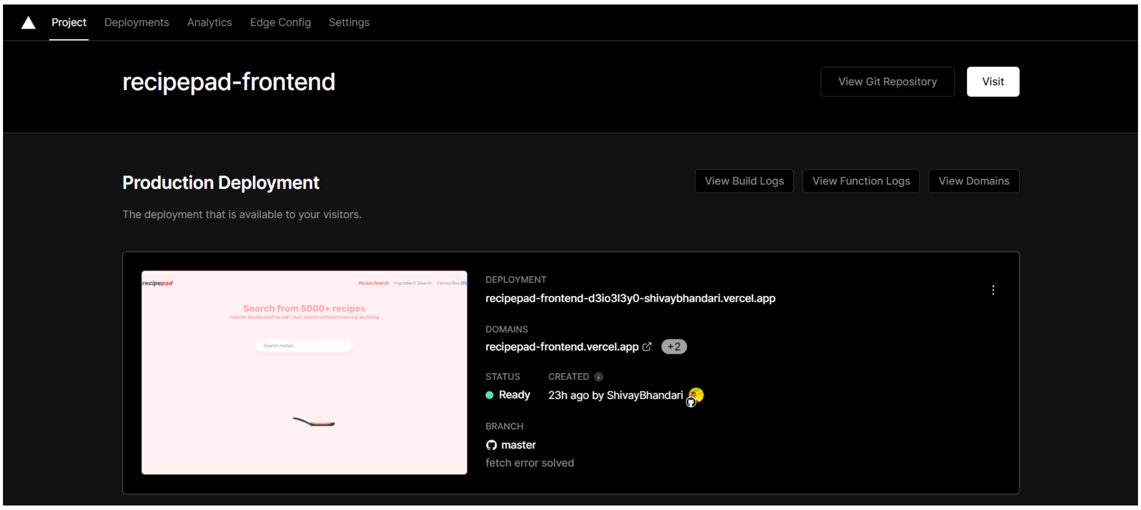


Figure 5.5: Favourite Recipes



recipepad-frontend

Production Deployment

The deployment that is available to your visitors.

DEPLOYMENT
recipepad-frontend-d3io3l3y0-shivaybhandari.vercel.app

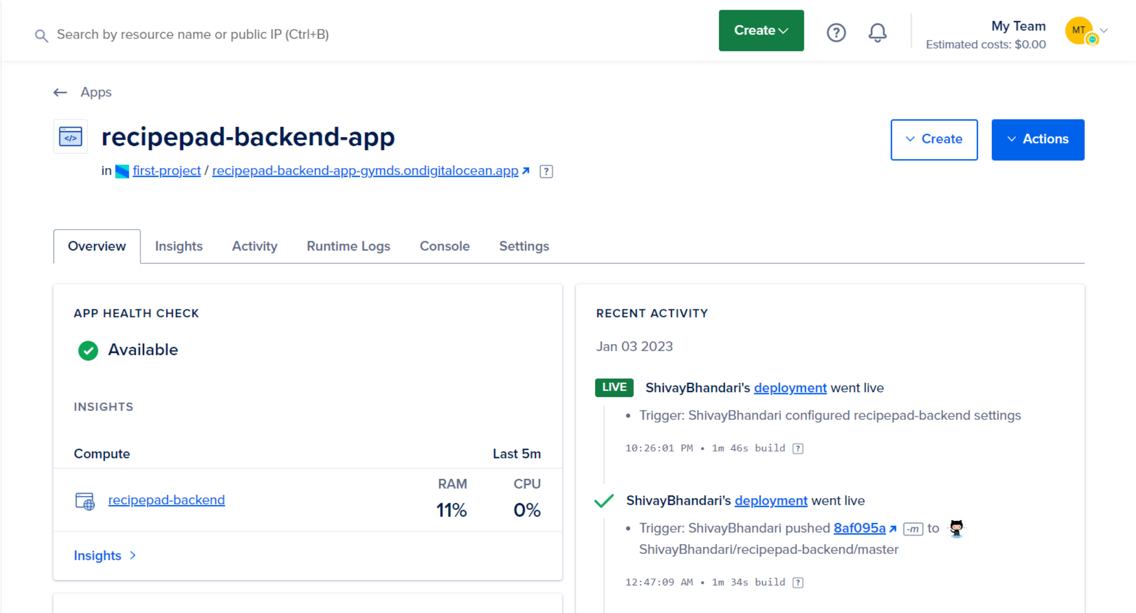
DOMAINS
recipepad-frontend.vercel.app +2

STATUS
Ready

CREATED
23h ago by ShivayBhandari

BRANCH
master
fetch error solved

Figure 5.6: Vercel Deployment



recipepad-backend-app

in first-project / recipepad-backend-app-gymds.ondigitalocean.app

APP HEALTH CHECK

Available

INSIGHTS

Compute	RAM	CPU
recipepad-backend	11%	0%

Last 5m

RECENT ACTIVITY

Jan 03 2023

LIVE ShivayBhandari's deployment went live

- Trigger: ShivayBhandari configured recipepad-backend settings

10:26:01 PM • 1m 46s build

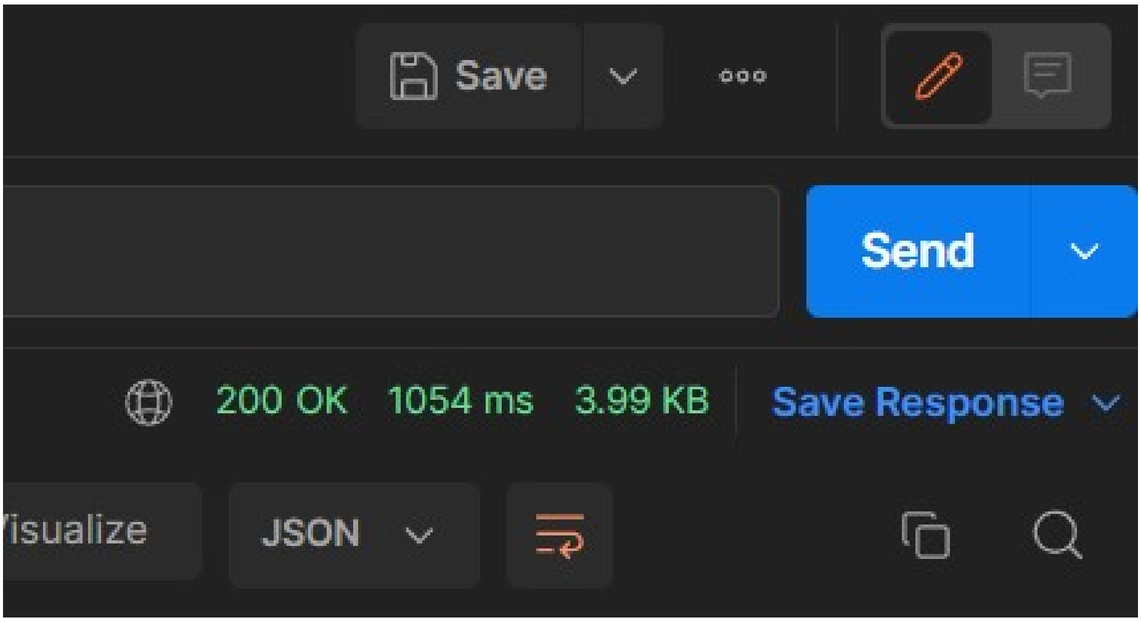
✓ ShivayBhandari's deployment went live

- Trigger: ShivayBhandari pushed 8af095a to ShivayBhandari/recipepad-backend/master

12:47:09 AM • 1m 34s build

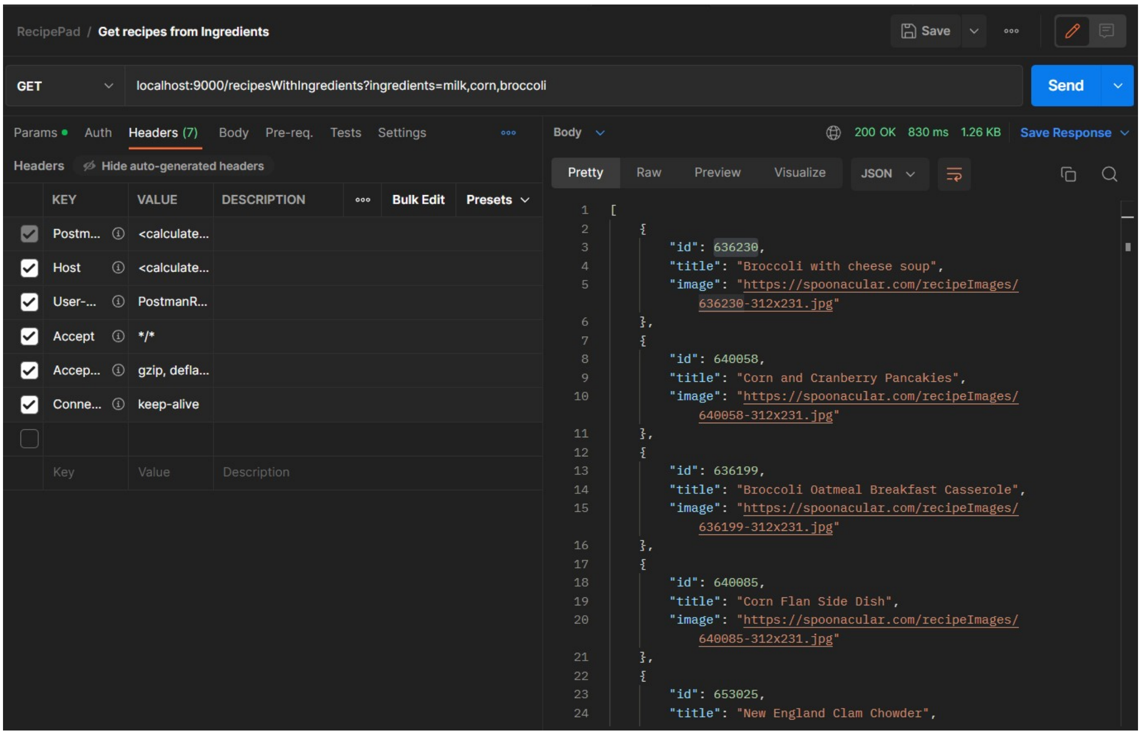
Figure 5.7: Digital Ocean Deployment

5.2 Performance of Project



200 OK 1054 ms 3.99 KB

Figure 5.8: Recipe information performance



GET localhost:9000/recipesWithIngredients?ingredients=milk,corn,broccoli

Headers

	KEY	VALUE	DESCRIPTION
☑	Postm...	<calculate...	
☑	Host	<calculate...	
☑	User-...	PostmanR...	
☑	Accept	*/*	
☑	Accep...	gzip, defla...	
☑	Conne...	keep-alive	
☐			
	Key	Value	Description

200 OK 830 ms 1.26 KB

```
[
    {
        "id": 636230,
        "title": "Broccoli with cheese soup",
        "image": "https://spoonacular.com/recipeImages/
            636230-312x231.jpg"
    },
    {
        "id": 640058,
        "title": "Corn and Cranberry Pancakies",
        "image": "https://spoonacular.com/recipeImages/
            640058-312x231.jpg"
    },
    {
        "id": 636199,
        "title": "Broccoli Oatmeal Breakfast Casserole",
        "image": "https://spoonacular.com/recipeImages/
            636199-312x231.jpg"
    },
    {
        "id": 640085,
        "title": "Corn Flan Side Dish",
        "image": "https://spoonacular.com/recipeImages/
            640085-312x231.jpg"
    },
    {
        "id": 653025,
        "title": "New England Clam Chowder",
```

Figure 5.9: Recipe from ingredients

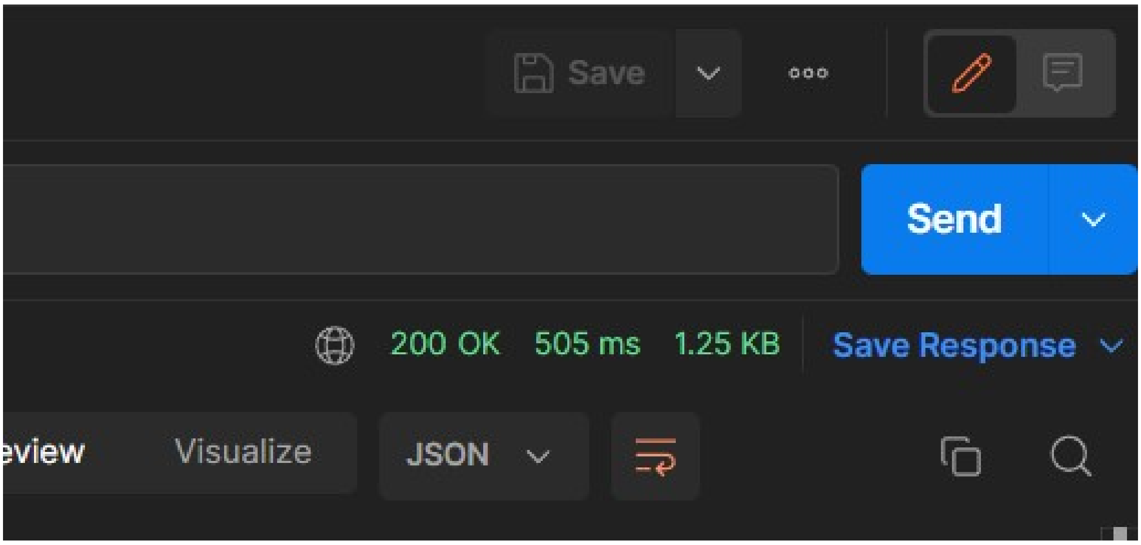


Figure 5.10: Random recipe performance

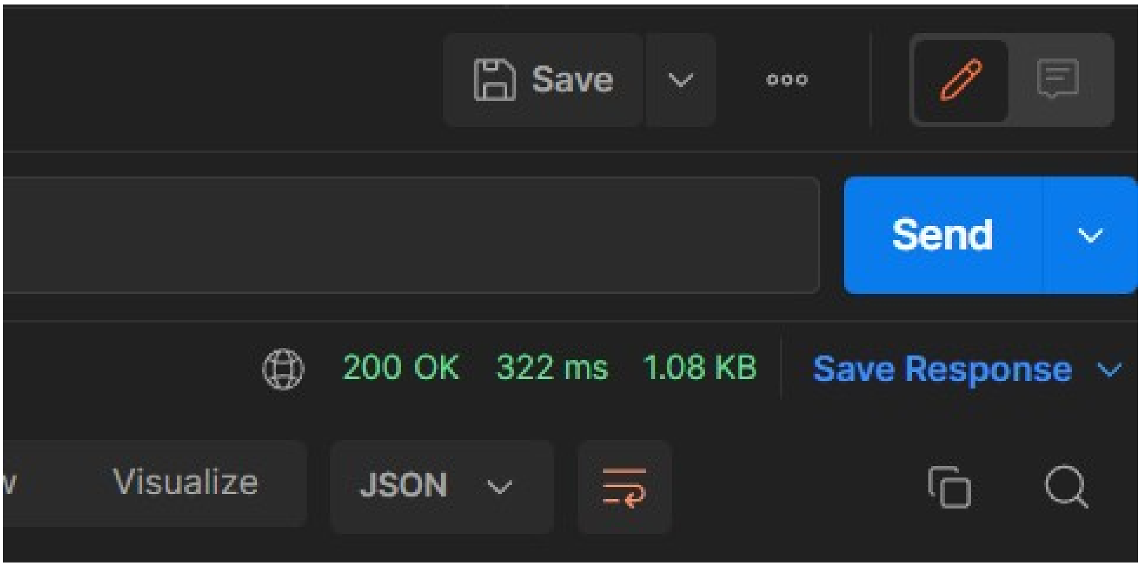


Figure 5.11: Normal recipe performance

6 CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The idea is to be implemented in the form of a web app that has the feature to search for recipes based on the presence of raw materials. The system suggests recipes according to the material available to the person. From different recommendations, a person can select the dish on the basis of taste and health conditions. Also, the user can search for random recipe if there is no particular recipe in mind. Every time there will be different result.

6.2 Future Scope of Project

- **Expanding the recipe database:** The project could potentially include more recipes from a wider variety of cuisines and dietary preferences. This could make the tool more appealing to a wider audience.

- **Adding new search criteria:** he project could potentially include additional criteria that users can use to search for recipes, such as cooking time, skill level, or the presence of specific nutrients..

- **Adding new features:** The project could potentially include additional features that enhance the user experience, such as the ability to create and share shopping lists, meal plans, or recipes with friends.

- **Expanding to new platforms:** The project could potentially be made available on additional platforms, such as smart home devices or virtual assistants.

- **Integration with grocery delivery or meal kit services:** The project could potentially include integration with services that allow users to have ingredients delivered to their doorstep or to receive pre-measured ingredients in a meal kit. This could make it easier for users to follow the recipes and save time on shopping and meal prep.

- **Customized recipe recommendations:** The project could potentially include a feature that suggests recipes to users based on their previous searches, ingredients they have on hand, or their dietary preferences. This could help users discover new recipes that they might not have found otherwise.

- **Social features:** The project could potentially include social features that allow users to share recipes with friends, rate and review recipes, or participate in cooking-related discussions. This could create a sense of community and encourage users to return to the tool for recipe ideas and inspiration.

- **Translation:** The project could potentially include translation functionality that allows users to view recipes in different languages. This could make the tool more accessible to users who speak languages other than English.

- **Personalization:** The project could potentially include features that allow users to personalize their experience, such as setting preferred cuisines or dietary preferences.

- **Integration with kitchen appliances:** The project could potentially include integration with smart kitchen appliances, such as ovens or sous-vide machines. This could allow users to send recipes directly to their appliances and make it easier to follow the instructions.

- **Recipe creation:** The project could potentially include a feature that allows users to create and save their own recipes. This could encourage users to share their own cooking creations and help build a larger and more diverse recipe database.

7 REFERENCES/BIBLIOGRAPHY

[1] https://code.visualstudio.com/
Accessed: 20-10-2022

[2] https://godoc.org/
Accessed: 01-11-2022

[3] https://reactjs.org/docs/getting-started.html
Accessed: 05-11-2022

[4] https://www.git-scm.com/docs/git-commit.
Accessed: 10-11-2022

[5] https://vercel.com/docs
Accessed: 10-11-2022

[6] https://docs.digitalocean.com/
Accessed: 10-11-2022