A Mid-Term Progress Report

On

“RecipePad”

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Information Technology

Submitted by :

Shivay Bhandari (1905398)

Tarun Sehgal (1905407)

Raghav Malhotra (1905381)

Guru Nanak Dev Engineering College,
Ludhiana-141006

Table of Contents

1 Introduction

1.1 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 material 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. If the particular ingredient is not available then it will show the nearby grocery store on the map where you can get that item.

1.2 Objectives

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

2 System Requirements

2.1 Software Requirements:

- Golang
- React JS
- Visual Studio
- API's

  • 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 centre. The speed with which directives are provided to the system will be affected by the efficiency of the software code.

  • Maintainability Requirement: The system is as simple to use as feasible, with all the capabilities accessible

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

2.2 Hardware Requirements:

- RAM: A minimum of 4GB is required, but it is suggested using 8GB RAM as training any algorithm will require some heavy lifting, less than 4GB can cause problems while multitasking
- Processor: A processor (CPU) is the logic circuitry that responds to and processes the basic instructions that drive a computer. The CPU is seen as the main and most crucial integrated circuitry (IC) chip in a computer, as it is responsible for interpreting most of computers commands.
- Operating : An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.

  Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

3 Software Requirement Analysis

3.1 Problem Definition:

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.

3.2 Modules and their Functionalities:

- **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.

- **React JS:** 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.

- **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.

- **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.

4 Software Design

Golang for backend and React JS for frontend is used in the development of the project Go is an open-source programming language focused on simplicity, reliability, and efficiency. Go was originally designed at Google in 2007. At the time, Google was growing quickly, and code being used to manage their infrastructure was also growing quickly in both size and complexity. Some Google cloud engineers began to feel that this large and complex codebase was slowing them down. So they decided that they needed a new programming language focused on simplicity and quick performance. Robert Griesemer, Rob Pike, and Ken Thompson designed Go. Go became an open-source project and was released publicly in 2012. It quickly gained a surprising level of popularity and has become one of the leading modern programming languages.

The React.js framework is an open-source JavaScript framework and library developed by Facebook. It's used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript. In React, you develop your applications by creating reusable components that you can think of as independent Lego blocks. These components are individual pieces of a final interface, which, when assembled, form the application's entire user interface. React's primary role in an application is to handle the view layer of that application just like the V in a model-view-controller (MVC) pattern by providing the best and most efficient rendering execution. Rather than dealing with the whole user interface as a single unit, React.js encourages developers to separate these complex UIs into individual reusable components that form the building blocks of the whole UI. In doing so, the ReactJS framework combines the speed and efficiency of JavaScript with a more efficient method of manipulating the DOM to render web pages faster and create highly dynamic and responsive web applications.

5 Coding/Core Module

```
import React from ’react’;
import ListOfIngredients from ’./ListOfIngredients’
import ’./style.css’
import ShowRecipes from ’./ShowRecipes’;

export default function cookBookInterface() {
    return(
        <div className="container">

            <div className=’sidebar’>
                <h3>Pantry</h3>
                <p>You have <span>0</span> ingredients.</p>
                <div className=’searchbox’>
                    <input type="search" placeholder=’add/remove ingredi
                </div>
                <div className=’ingredients-list’>
                    <ListOfIngredients />
                </div>
            </div>

            <div className=’show-recipes’>
                <ShowRecipes/>
            </div>

        </div>
    );
}
App.js

import Login from ’./Components/Login’;
import {
  Routes,
  Route,
  BrowserRouter,
} from "react-router-dom";
import Signup from ’./Components/Singup’;
import CookBookInterface from ’./Components/CookBookInterface’;

class App extends Component {
  render() {
    return (
    <BrowserRouter>
```

```
      <Routes>
        <Route path="/" element={<Login />} />
        <Route path="signup" element={<Signup />} />
        <Route path="interface" element={<CookBookInterface />} />
      </Routes>
    </BrowserRouter>
    // <CookBookInterface></CookBookInterface>
    );
  }
}

export default App;
```

Backend Code

```
package main

import (
        "log"
        "os"

        routes "github.com/ShivayBhandari/recipepad-backend/routes"
        "github.com/joho/godotenv"

        "github.com/gin-gonic/gin"
)

func main(){
        err := godotenv.Load(".env")
        if err != nil{
                log.Fatal("Error loading .env file")
        }

        port := os.Getenv("PORT")

        if port == ""{
                port = "8000"
        }

        router := gin.New()
        router.Use(gin.Logger())

        routes.RecipeSearch(router)
        routes.AuthRoutes(router)
        routes.UserRoutes(router)

        router.GET("/api-1", func(c *gin.Context){
                c.JSON(200, gin.H{"success":"Access granted for api-1"})
        })

        router.GET("/api-2", func(c *gin.Context){
                c.JSON(200, gin.H{"success":"Access granted for api-2"})
        })

        router.Run(":" + port)
}
```

6 Performance of the Project Developed

1. The performance for our particular searches for recipes are depicted in the below pictures as taken from our postman responses.

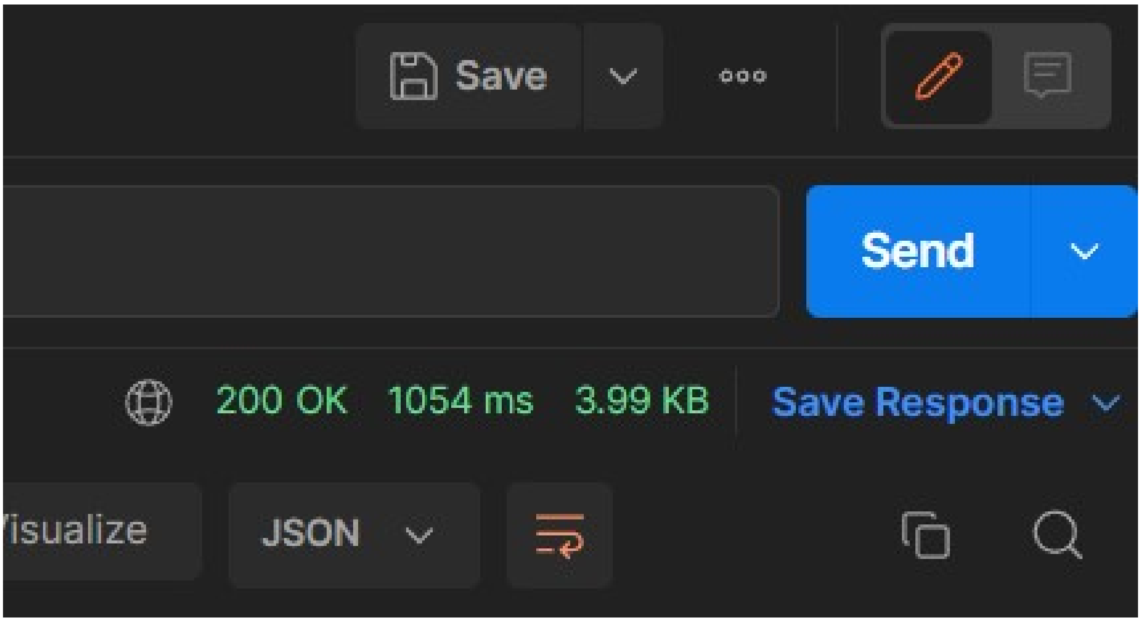


Figure 6.1: *Recipe information performance*

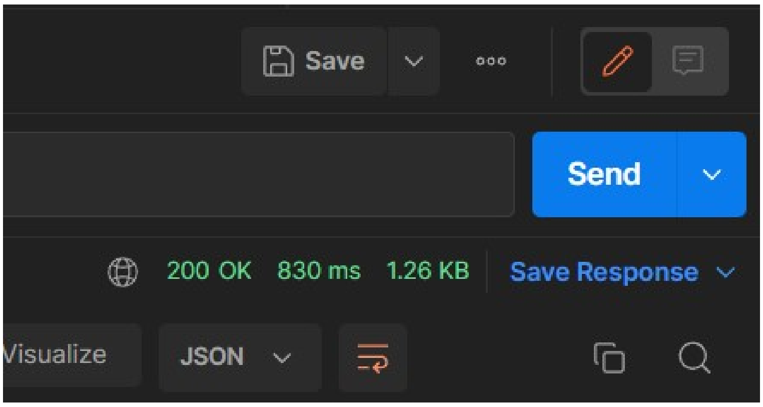
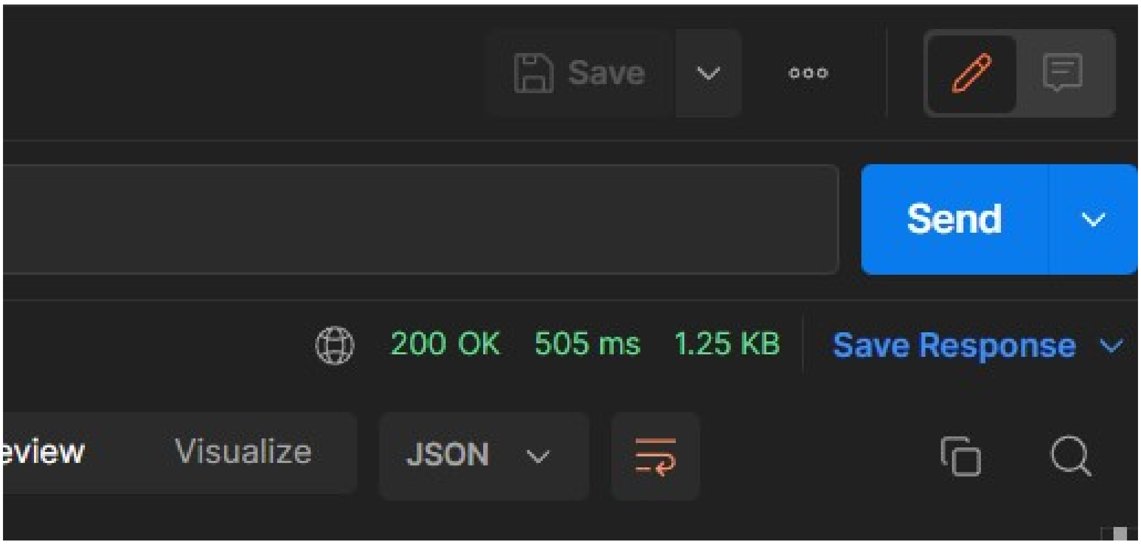
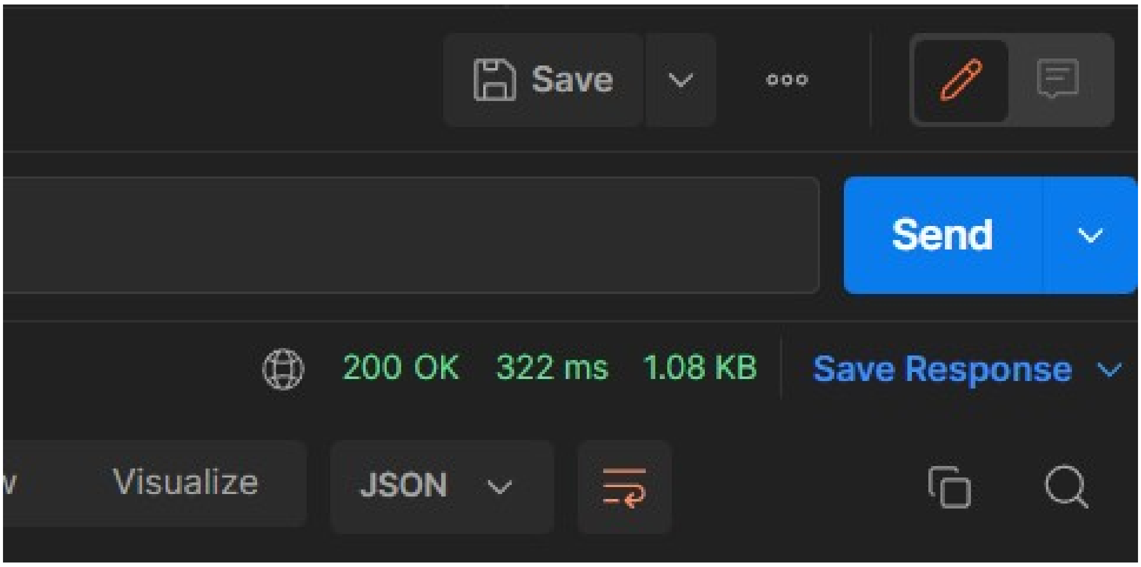


Figure 6.2: *Recipe from ingredients*



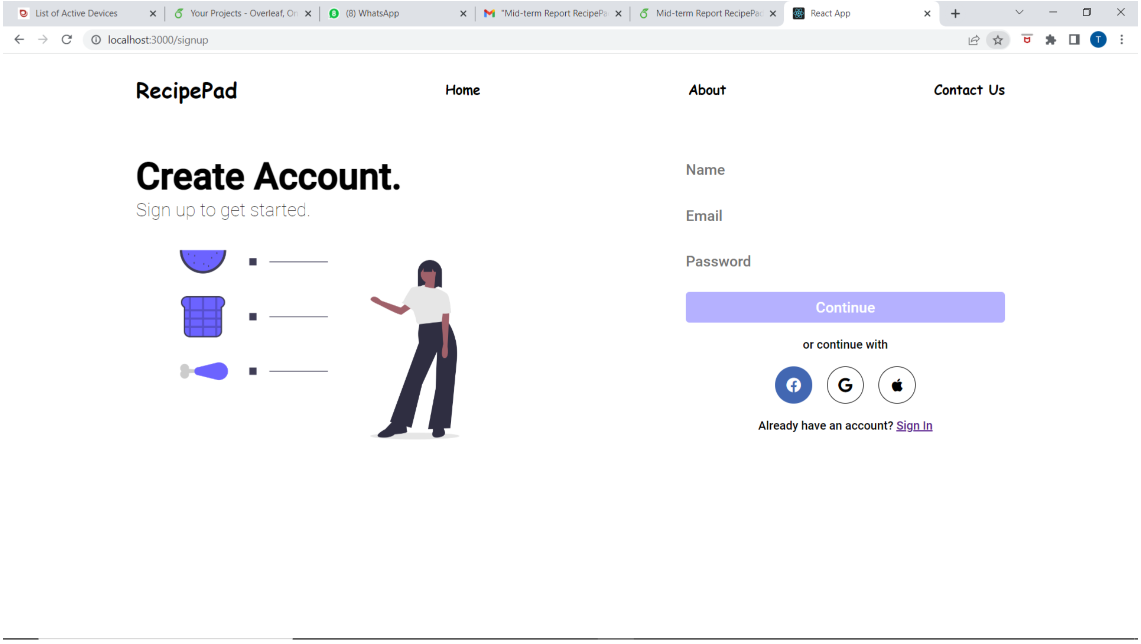
200 OK 505 ms 1.25 KB

Figure 6.3: *Random recipe performance*



200 OK 322 ms 1.08 KB

Figure 6.4: *Normal recipe performance*



RecipePad

Home

About

Contact Us

Create Account.

Sign up to get started.

Name

Email

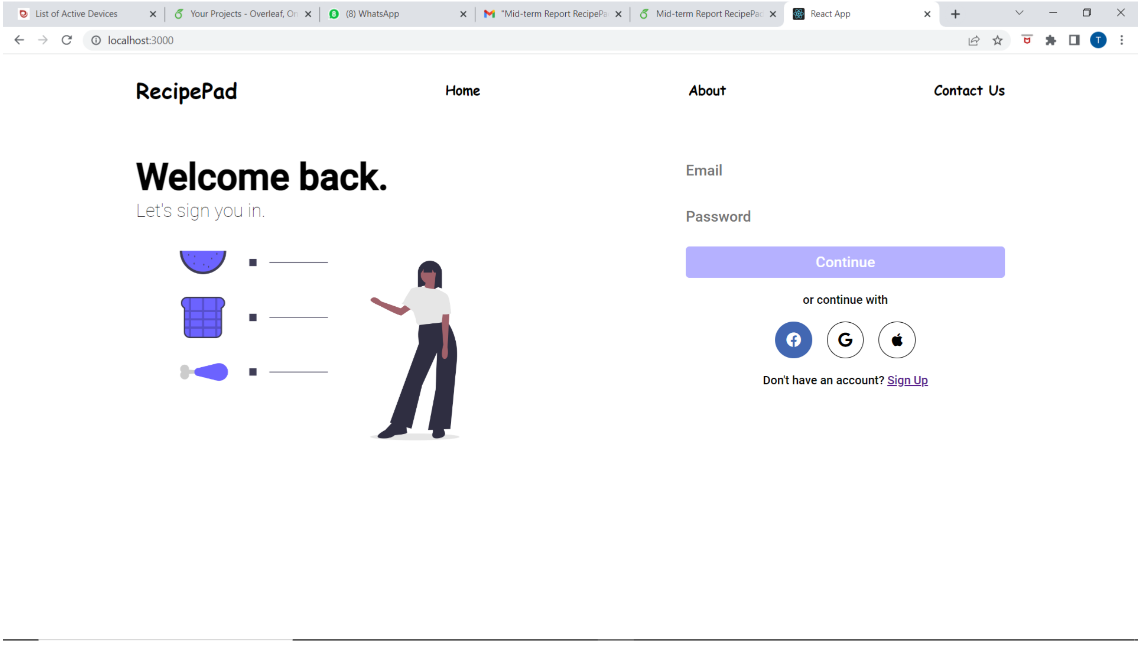
Password

Continue

or continue with

Already have an account? Sign In

Figure 6.5: Signup Page



RecipePad

Home

About

Contact Us

Welcome back.

Let's sign you in.

Email

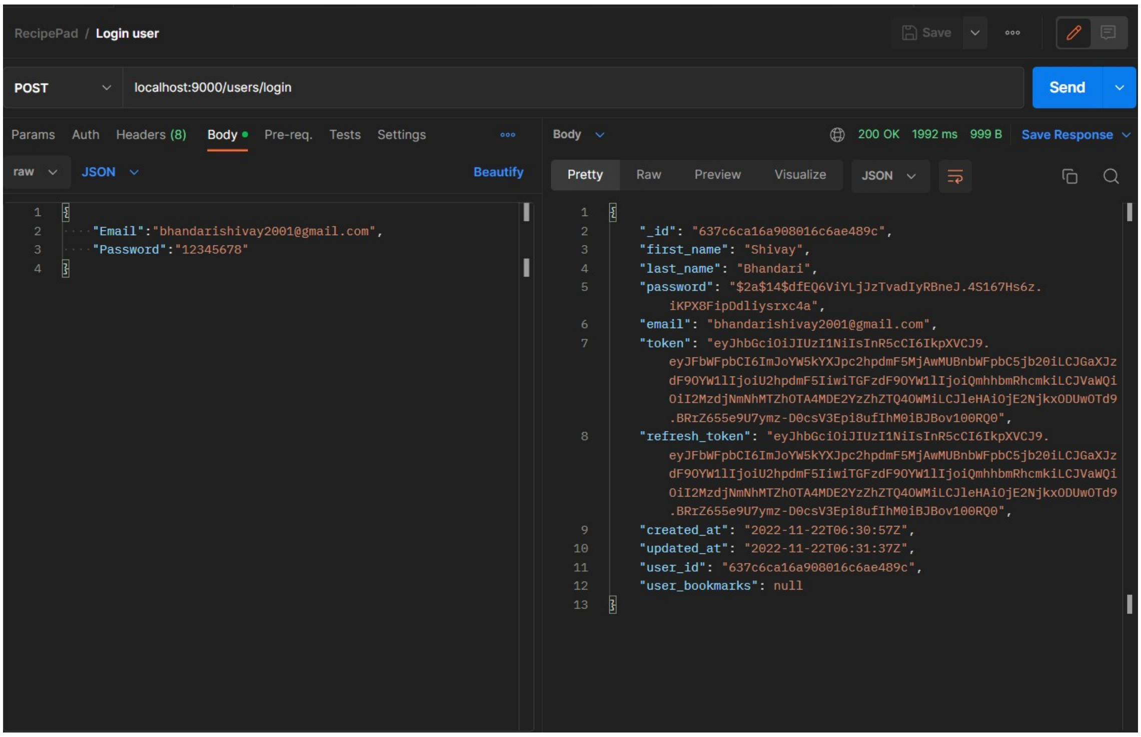
Password

Continue

or continue with

Don't have an account? Sign Up

Figure 6.6: Signin Page



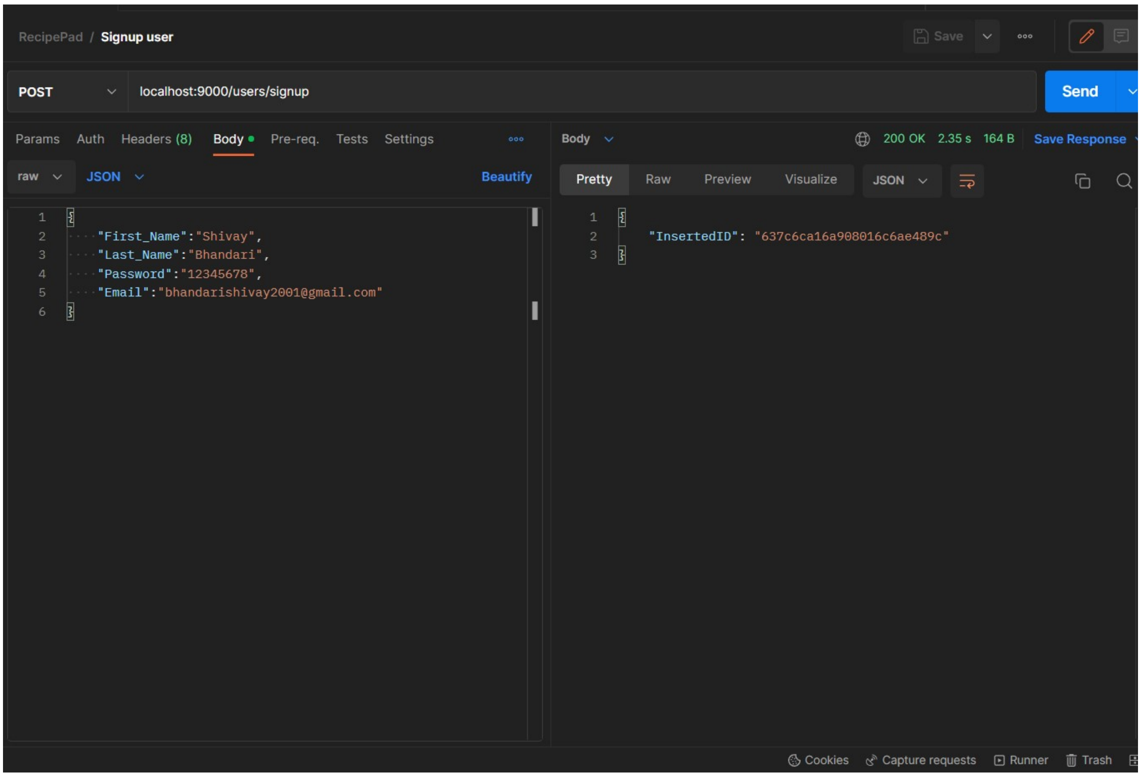
POST localhost:9000/users/login

```
{
····"Email":"bhandarishivay2001@gmail.com",
····"Password":"12345678"
}
```

200 OK 1992 ms 999 B

```
{
    "_id": "637c6ca16a908016c6ae489c",
    "first_name": "Shivay",
    "last_name": "Bhandari",
    "password": "$2a$14$dfEQ6ViYLjJzTvadIyRBneJ.4S167Hs6z.
        iKPX8FipDdliysrxc4a",
    "email": "bhandarishivay2001@gmail.com",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJFbWFpbCI6ImJoYW5kYXJpc2hpdmF5MjAwMUBnbWFpbC5jb20iLCJGaXJz
        dF9OYW1lIjoiU2hpdmF5IiwiTGFzdF9OYW1lIjoiQmhhbmRhcmkiLCJVaWQi
        OiI2MzdjNmNhMTZhOTA4MDE2YzZhZTQ4OWMiLCJleHAiOjE2NjkxODUwOTd9
        .BRrZ655e9U7ymz-D0csV3Epi8ufIhM0iBJBov100RQ0",
    "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJFbWFpbCI6ImJoYW5kYXJpc2hpdmF5MjAwMUBnbWFpbC5jb20iLCJGaXJz
        dF9OYW1lIjoiU2hpdmF5IiwiTGFzdF9OYW1lIjoiQmhhbmRhcmkiLCJVaWQi
        OiI2MzdjNmNhMTZhOTA4MDE2YzZhZTQ4OWMiLCJleHAiOjE2NjkxODUwOTd9
        .BRrZ655e9U7ymz-D0csV3Epi8ufIhM0iBJBov100RQ0",
    "created_at": "2022-11-22T06:30:57Z",
    "updated_at": "2022-11-22T06:31:37Z",
    "user_id": "637c6ca16a908016c6ae489c",
    "user_bookmarks": null
}
```

Figure 6.7: *Backend response for login*



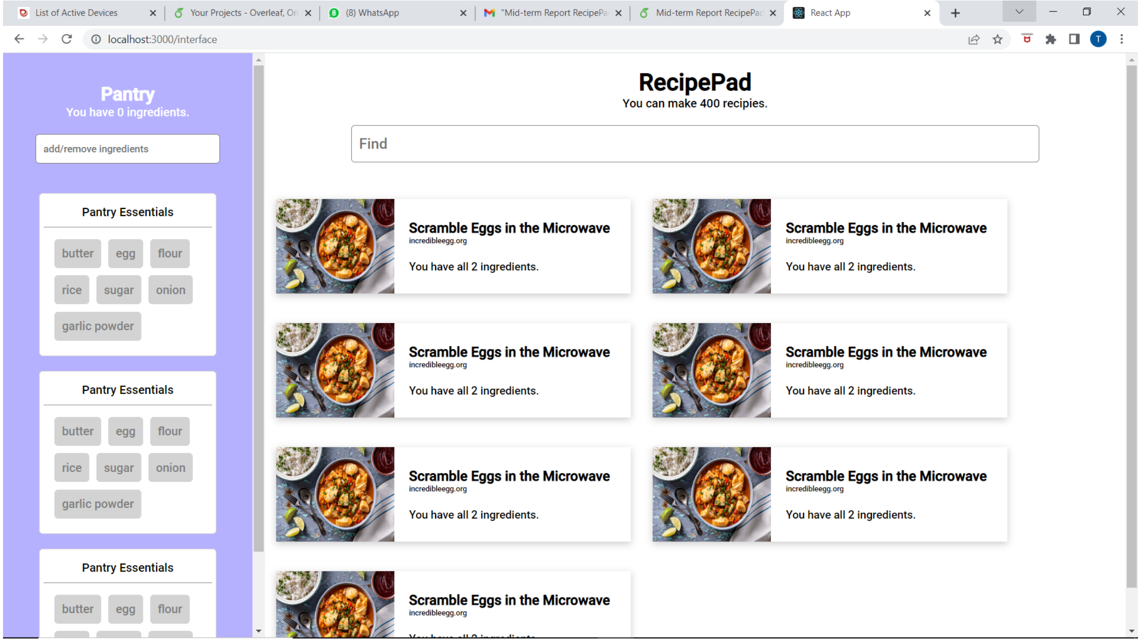
POST localhost:9000/users/signup

```
{
····"First_Name":"Shivay",
····"Last_Name":"Bhandari",
····"Password":"12345678",
····"Email":"bhandarishivay2001@gmail.com"
}
```

200 OK 2.35 s 164 B

```
{
    "InsertedID": "637c6ca16a908016c6ae489c"
}
```

Figure 6.8: *Backend response for login*



Pantry
You have 0 ingredients.

add/remove ingredients

Pantry Essentials
butter egg flour
rice sugar onion
garlic powder

Pantry Essentials
butter egg flour
rice sugar onion
garlic powder

Pantry Essentials
butter egg flour

RecipePad
You can make 400 recipies.

Find

Scramble Eggs in the Microwave
incredibleegg.org
You have all 2 ingredients.

Scramble Eggs in the Microwave
incredibleegg.org
You have all 2 ingredients.

Scramble Eggs in the Microwave
incredibleegg.org
You have all 2 ingredients.

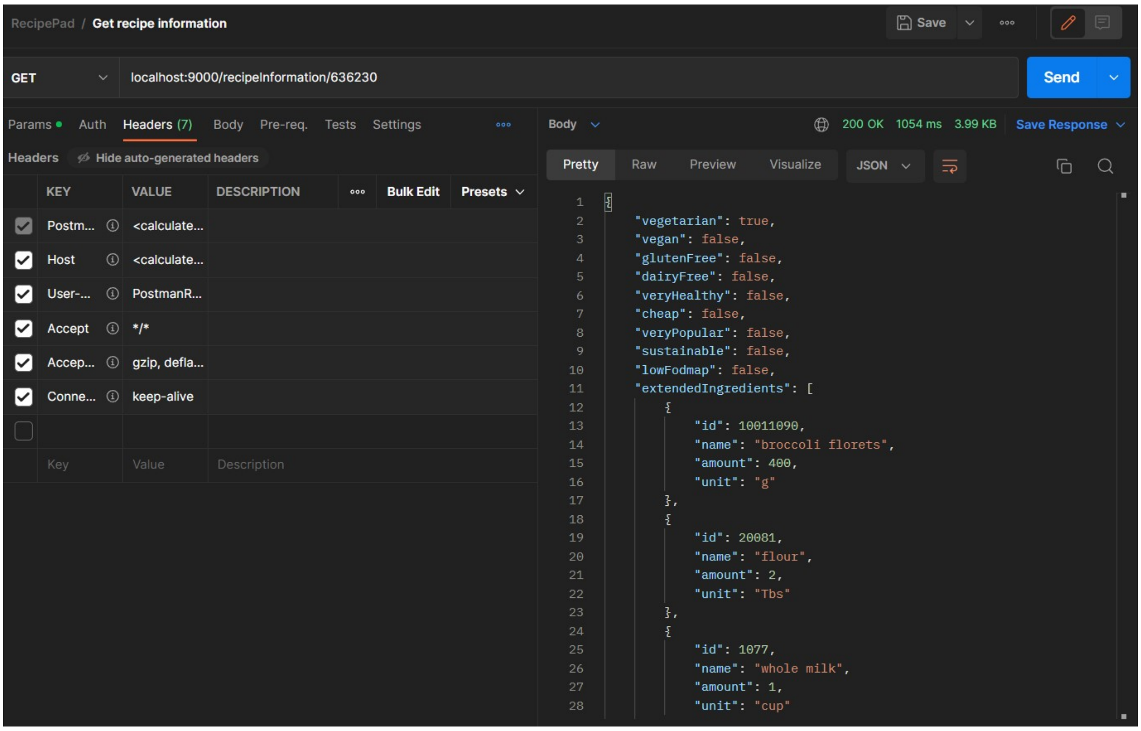
Scramble Eggs in the Microwave
incredibleegg.org
You have all 2 ingredients.

Scramble Eggs in the Microwave
incredibleegg.org
You have all 2 ingredients.

Scramble Eggs in the Microwave
incredibleegg.org
You have all 2 ingredients.

Scramble Eggs in the Microwave
incredibleegg.org

Figure 6.9: *Main interface*



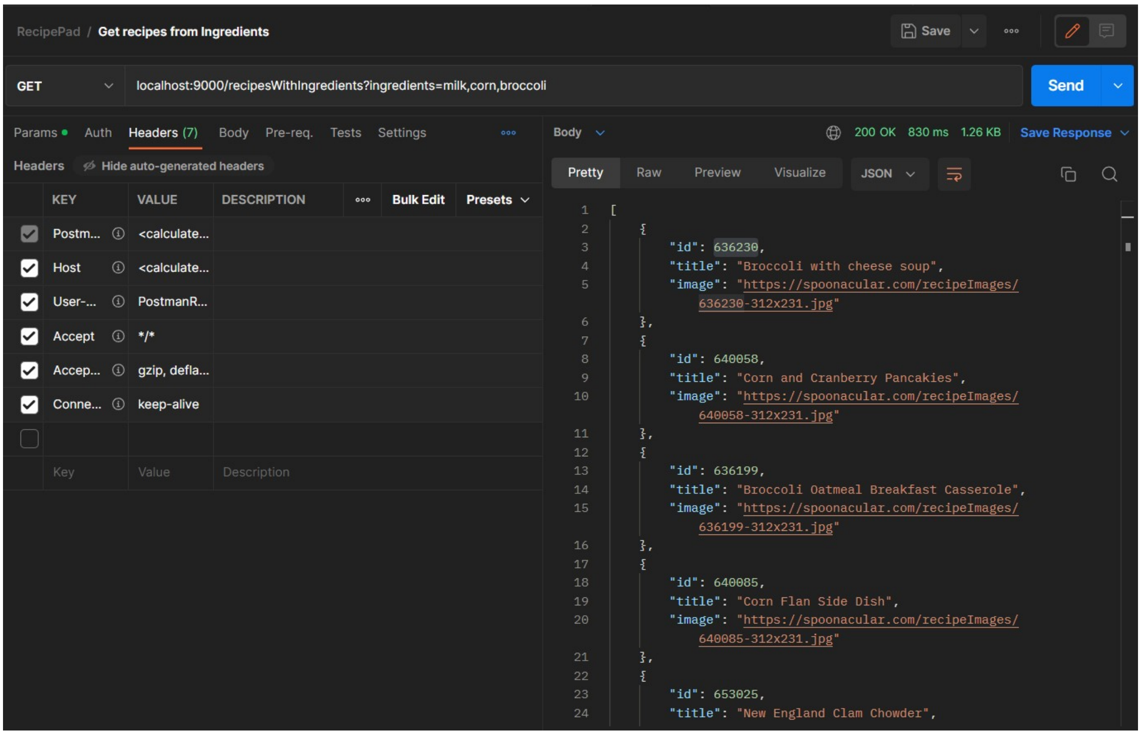
localhost:9000/recipeInformation/636230

200 OK 1054 ms 3.99 KB

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

```
{
    "vegetarian": true,
    "vegan": false,
    "glutenFree": false,
    "dairyFree": false,
    "veryHealthy": false,
    "cheap": false,
    "veryPopular": false,
    "sustainable": false,
    "lowFodmap": false,
    "extendedIngredients": [
        {
            "id": 10011090,
            "name": "broccoli florets",
            "amount": 400,
            "unit": "g"
        },
        {
            "id": 20081,
            "name": "flour",
            "amount": 2,
            "unit": "Tbs"
        },
        {
            "id": 1077,
            "name": "whole milk",
            "amount": 1,
            "unit": "cup"
```

Figure 6.10: *Recipe information*



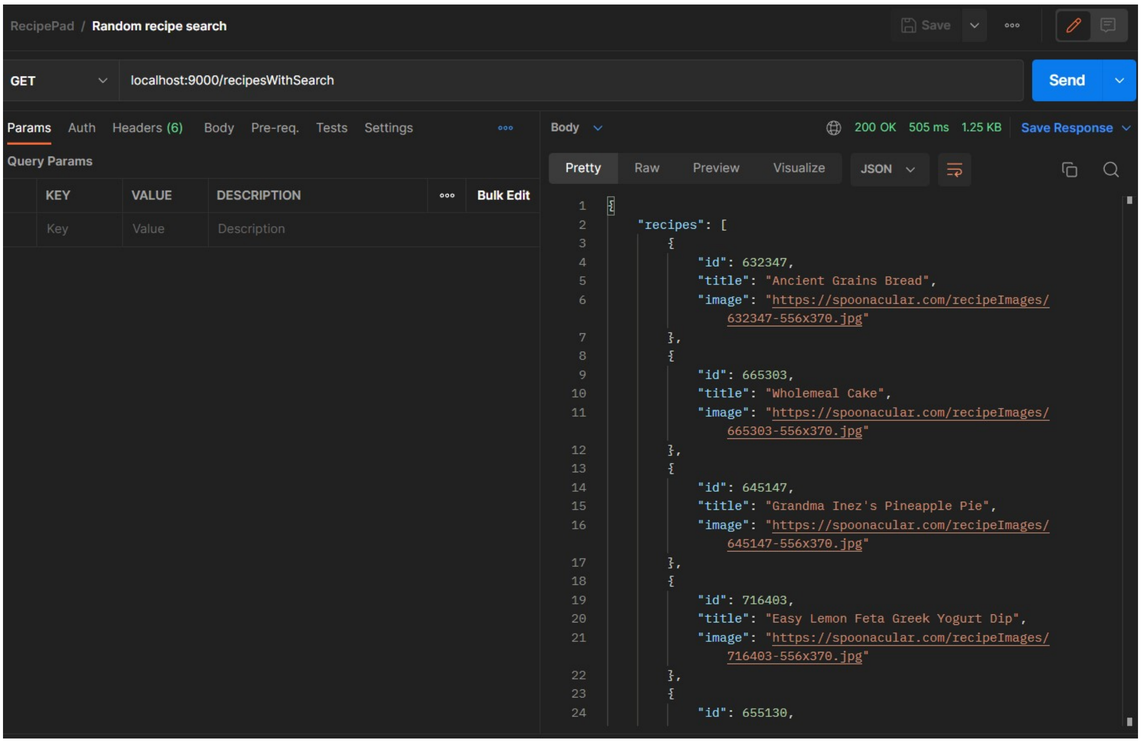
GET localhost:9000/recipesWithIngredients?ingredients=milk,corn,broccoli

Headers

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

```
[
    {
        "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 6.11: *Get recipe from ingredients*



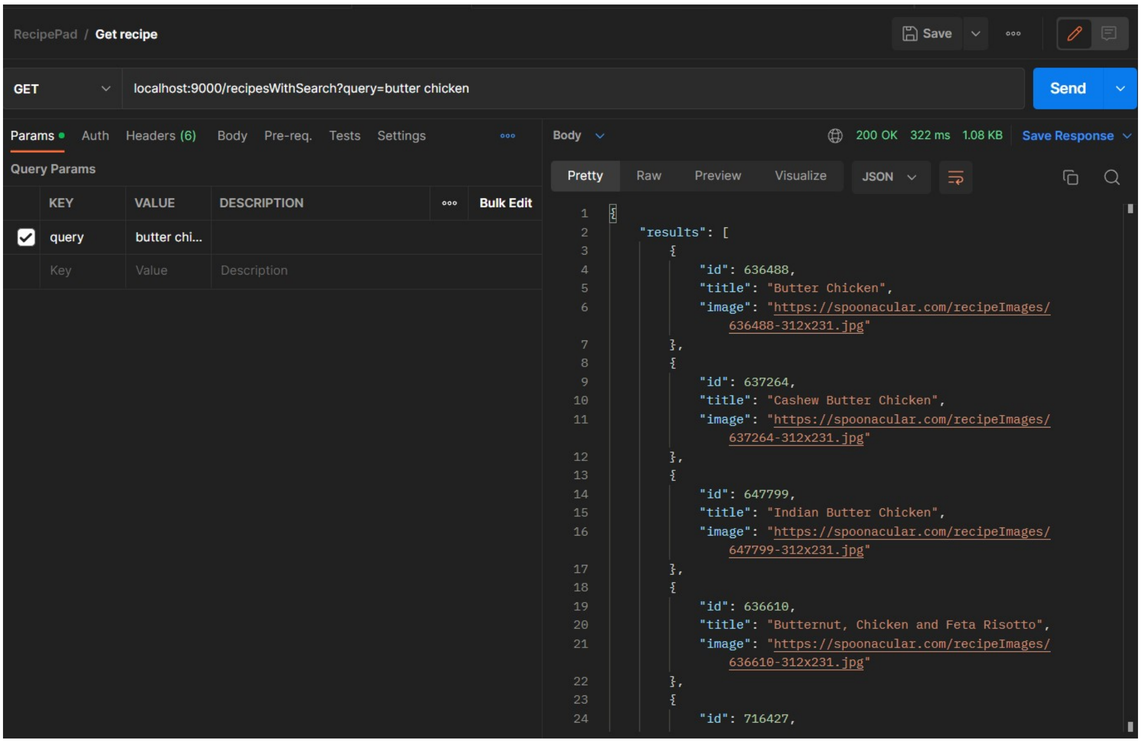
GET localhost:9000/recipesWithSearch

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
{
    "recipes": [
        {
            "id": 632347,
            "title": "Ancient Grains Bread",
            "image": "https://spoonacular.com/recipeImages/
                632347-556x370.jpg"
        },
        {
            "id": 665303,
            "title": "Wholemeal Cake",
            "image": "https://spoonacular.com/recipeImages/
                665303-556x370.jpg"
        },
        {
            "id": 645147,
            "title": "Grandma Inez's Pineapple Pie",
            "image": "https://spoonacular.com/recipeImages/
                645147-556x370.jpg"
        },
        {
            "id": 716403,
            "title": "Easy Lemon Feta Greek Yogurt Dip",
            "image": "https://spoonacular.com/recipeImages/
                716403-556x370.jpg"
        },
        {
            "id": 655130,
```

Figure 6.12: *Random search*



GET localhost:9000/recipesWithSearch?query=butter chicken

	KEY	VALUE	DESCRIPTION
☑	query	butter chi...	
	Key	Value	Description

200 OK 322 ms 1.08 KB

```
{
    "results": [
        {
            "id": 636488,
            "title": "Butter Chicken",
            "image": "https://spoonacular.com/recipeImages/
                636488-312x231.jpg"
        },
        {
            "id": 637264,
            "title": "Cashew Butter Chicken",
            "image": "https://spoonacular.com/recipeImages/
                637264-312x231.jpg"
        },
        {
            "id": 647799,
            "title": "Indian Butter Chicken",
            "image": "https://spoonacular.com/recipeImages/
                647799-312x231.jpg"
        },
        {
            "id": 636610,
            "title": "Butternut, Chicken and Feta Risotto",
            "image": "https://spoonacular.com/recipeImages/
                636610-312x231.jpg"
        },
        {
            "id": 716427,
```

Figure 6.13: *Normal search*

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