



4epicure

(Recipe & Meals App)
Software Engineering

CS301

[A CURE FOR EPICURE]

TOC

Overview

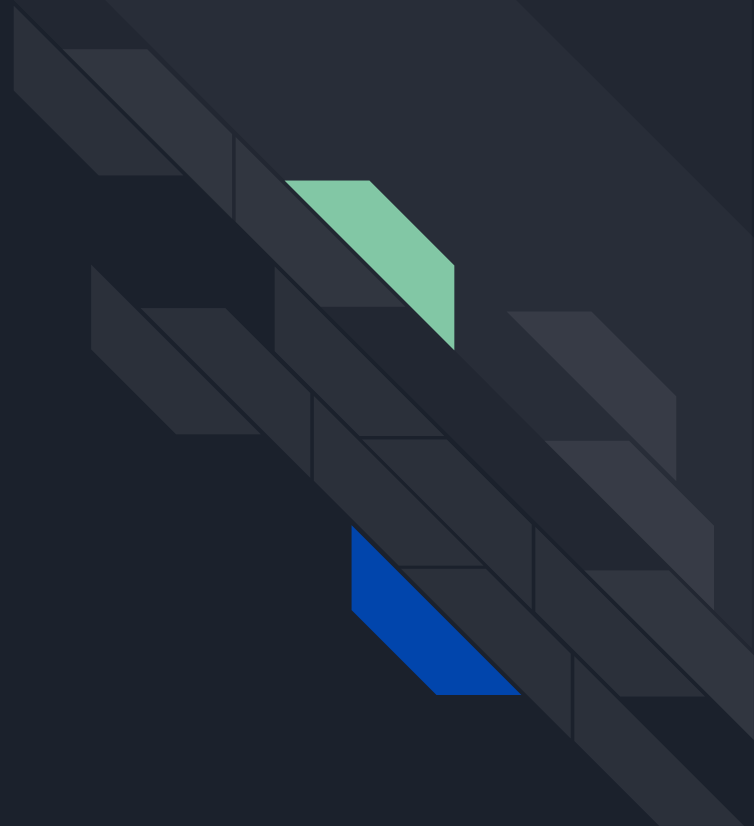
Features

Requirement analysis

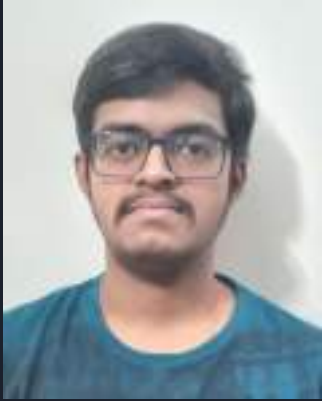
Prototype and demonstration

Tech stack

Testing



Our Team



Karthik Avinash
21bcs052



Devesh Kumar
21bcs032



Devarshi Dubey
21bcs031



Sameed Yallur
21bcs068

CSE
Section A
Semester 4



Overview

Why 4epicure?

No centralized platform to access all information about meals

- An app to let people find restaurants in their locality
- Helpful for students and travelers
- Connect people with their regional or native restaurants in different states.



Features

- Integrated map to display restaurants and their reviews within a given radius.
- Youtube integration to search recipe video tutorials
- Create and search recipes
- Nutritional graphical chart for each recipe
- Voice assistant and,
- Accelerometer used for navigation (completely touchless UI): accessibility
- ChatGPT integration

RA: Lets ask ChatGPT: 😊 (Not only this. Survey, Visited other apps)



We have made a recipe and meals app (without ordering) I will give you a few details about that:

1. We feel that: No centralized platform to access all information about meals
2. An app to let people find restaurants in their locality
3. Helpful for students and travelers
4. Connect people with their regional or native restaurants in different states.

FEATURES:

1. Integrated map to display restaurants under a given radius
2. Youtube integration to search recipe video tutorials
3. Create and search recipes
4. Nutritional graphical chart for each meal
5. Voice assistant and,
6. Accelerometer used for navigation (completely touchless UI): accessibility
7. ChatGPT integration

Based on this data..

Can you tell to our audience if any mobile app exists with all these ideas in the system (atleast till the time you were learning).

If any mobile app exists, kindly let us know... :)



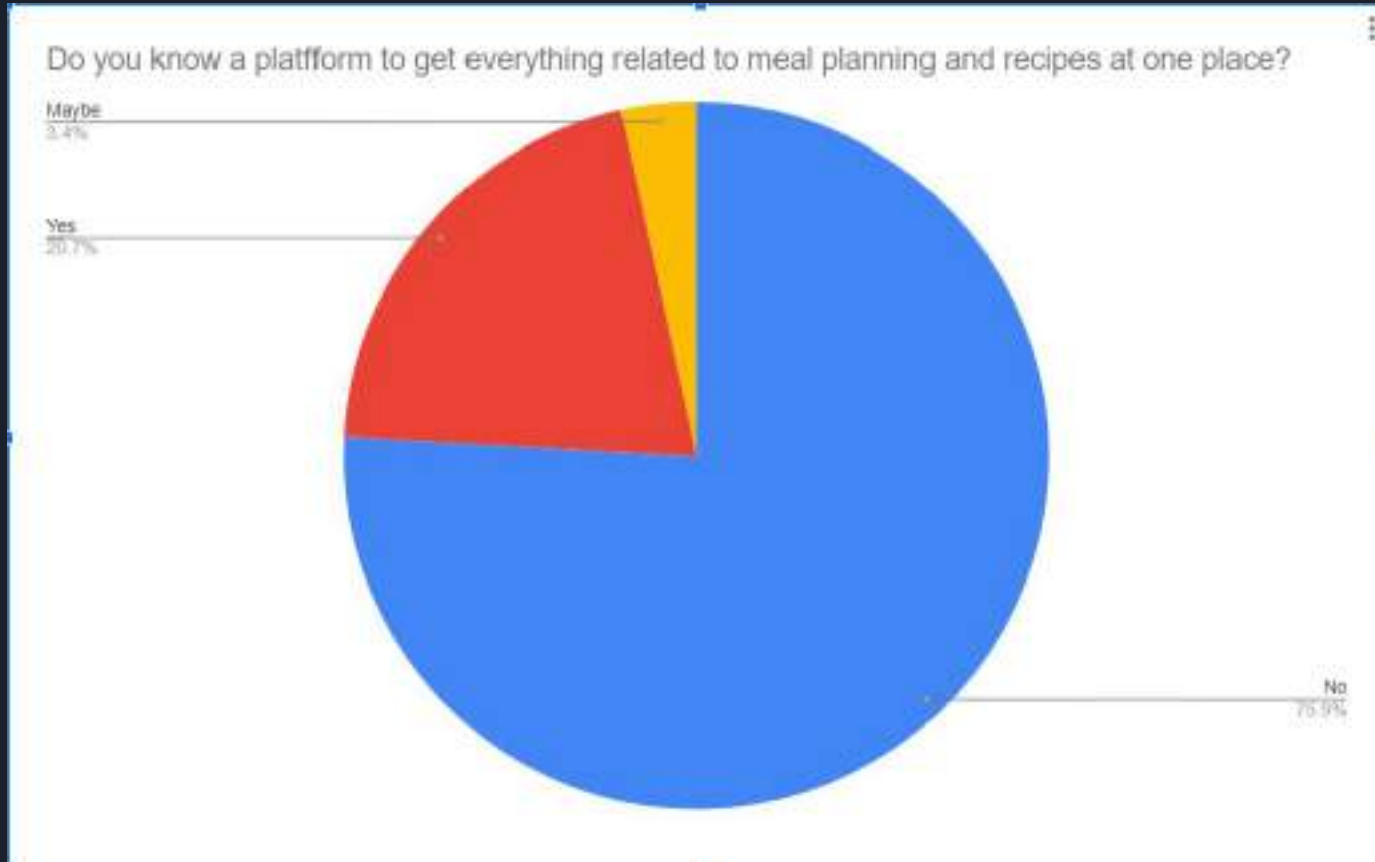
Reply



Based on the information you have provided, it does not appear that there is an existing mobile app that incorporates all of the features you have listed. While there are many apps that offer some similar features, such as restaurant search, recipe creation, and nutritional information, it is rare to find an app that combines all of these features along with unique features such as YouTube integration, voice assistant, accelerometer navigation, and ChatGPT integration.

It's important to note, however, that the mobile app market is constantly evolving and new apps with innovative features are being released all the time. So while there may not be an existing app that incorporates all of the features you have listed, it's always possible that a similar app could be released in the future. Nonetheless, your app appears to offer a unique and valuable combination of features that could set it apart from existing apps and provide significant value to users.

Requirement analysis

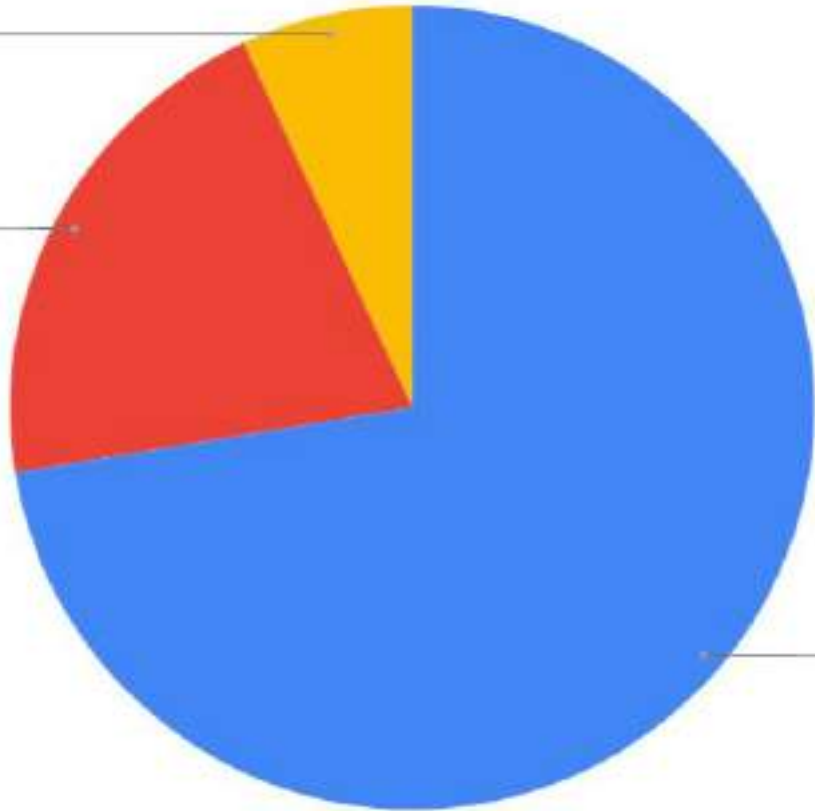


Do you think that having all necessary information (location, nutrition, recipe) about meal planning on a single app would help you?

Maybe
6.9%

No
20.7%

Yes
72.4%





PROTOTYPE

[Prototype Design Link](#)



DEMONSTRATION (LIVE DEMO)

JIRA workflow (Board)

PM

The screenshot displays the Jira Software interface for the 'MEALS_APP' project. The top navigation bar includes 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'Teams', 'Apps', and a 'Create' button. A search bar on the right shows 'G. Board'. A banner at the top states: 'You're eligible for a free 14-day trial of our Jira Software Standard plan. Start now!'. The left sidebar contains a 'PLANNING' section with 'Roadmap', 'Backlog', and 'Board' (selected), and a 'DEVELOPMENT' section with 'Code', 'Project pages', 'Add shortcut', and 'Project settings'. The main area shows the 'Sprint - Authentication pages' board, which is a Kanban board with columns: 'TO DO 4 ISSUES', 'IN PROGRESS 1 ISSUE', and 'DONE 3 ISSUES'. The 'TO DO' column contains four issues: 'splash screen' (MF-4), '2nd screen' (MF-5), 'otp verification screen' (MF-7), and 'start backend' (MF-8). The 'IN PROGRESS' column contains one issue: 'forgot password screen' (MF-6). The 'DONE' column contains three issues: 'Signup Screen' (MF-1), 'Login Screen' (MF-2), and 'First screen' (MF-3). Each issue card shows a status icon (blue, orange, or purple) and a 'Complete sprint' button. A 'Quickstart' panel on the right offers options to 'Create a project', 'Customize your board', 'Create an issue', and 'Invite your teammates', along with a video titled 'What is Jira?' and a 'Dismiss Quickstart' button.

Project Management - JIRA workflow(Backlog)

The screenshot displays the JIRA Software interface for a project named 'MEALS_APP'. The left sidebar shows the navigation menu with options like 'Roadmap', 'Backlog', 'Board', and 'Code'. The main area shows the 'Backlog' view for the 'MEALS_APP' project. A banner at the top indicates a free 14-day trial of Jira Software Standard plan. The 'Backlog' view shows a list of tasks organized by a sprint named 'Authentication pages' (24 Feb - 4 Mar, 8 hours). The tasks are listed with their status and priority:

Task	Status	Priority
MI-3 Signup Screen	done	High
MI-2 Login Screen	done	High
MI-4 First screen	done	High
MI-4 splash screen	TO DO	Medium
MI-1 2nd screen	TO DO	Medium
MI-6 forgot password screen	IN PROGRESS	High
MI-7 otp verification screen	TO DO	High
MI-8 start backward	TO DO	Medium

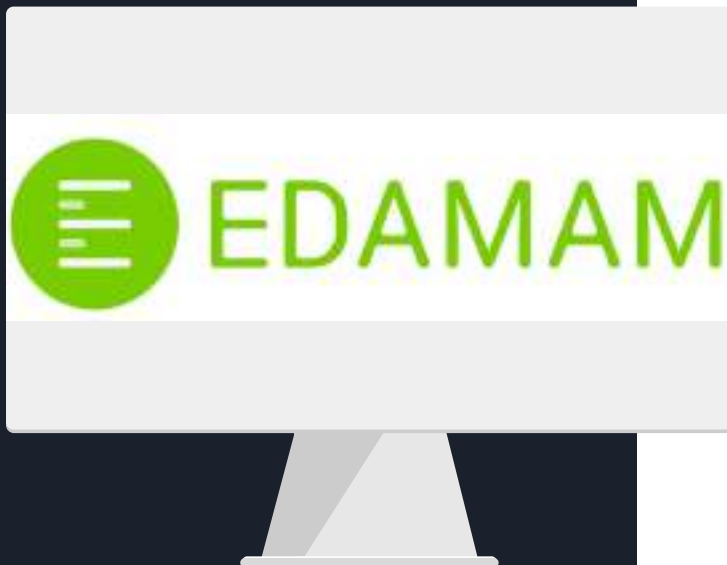
The right sidebar shows a 'Quickstart' panel with options to 'Create a project' and 'Create an issue'. Below this is a video titled 'What is Jira?' and a section for 'Issues' with a 'Show me' button and a 'View issue tutorial' link. At the bottom, there are links to 'Invite your teammates' and 'Connect your tools'.

URL: <https://www.atlassian.com/jira/software/projects/ME/boards/2/backlog>

APIs

Nutrition Analysis API

The Nutrition Analysis API uses Natural Language Processing and semantically structured data.

A mobile app interface for creating a recipe. The top bar is orange with the text "Create Recipe". Below it is a photo of a bowl of Shahi Paneer. Under the photo is an orange button labeled "Add Photo". Below that is a form with a "Title" field containing "Shahi Paneer" and a "Description" field containing a paragraph about the dish. Below the description is an "Ingredients" section with an "Add Ingredient" button and a list of ingredients: "200 gram paneer", "100 gram pea", and "2 gram garam masala". At the bottom is a "Steps" section with an "Add Step" button and a list of steps: "1. Take it until you make it!!".A mobile app interface for getting nutrient information. The top bar is orange with the text "Create Recipe". Below it is a description of the dish: "that is made with paneer (a type of Indian cheese), a rich and creamy tomato-based gravy, and a variety of aromatic spices. The dish has a smooth, creamy texture and is often garnished with nuts and fresh cream." Below the description is an "Ingredients" section with an "Add Ingredient" button and a list of ingredients: "200 gram paneer", "100 gram pea", and "2 gram garam masala". Below the ingredients is a "Steps" section with an "Add Step" button and a list of steps: "1. Take it until you make it!!". At the bottom is a "Get Nutrient Information" button. Below the button is a "Nutrient Information" section with the following data: "Calories: 109.0 kcal", "Protein: 22.7 g", "Carbs: 8.6 g", and "Fat: 5.6 g".

```
GET /api/1/

HTTP/1.1 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 1,
  "name": "Spaghetti Carbonara",
  "recipe": {
    "id": 1,
    "name": "C",
    "full": 1,
    "category": "Carbonara",
    "ingredients": [
      {
        "name": "Spaghetti",
        "quantity": 100,
        "unit": "grams"
      },
      {
        "name": "Sausage",
        "quantity": 150,
        "unit": "grams"
      }
    ]
  },
  "deleted": false,
  "id": 1,
  "category": "Carbonara",
  "deleted": false,
  "id": 2,
  "category": "Carbonara"
}
```

Localization of 4Epicure in
REST API using python
googletrans library



The text you provided appears to be a string of Unicode escape sequences for Devanagari script characters used in Hindi language. Here is the string in its unescaped form:

श्रीगुरु कृष्णाय नमः

The above text is in the Hindi language and could be translated to English as "Shri Gurus Krishna Namah".



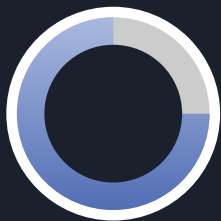
Tech Stack 01

Django

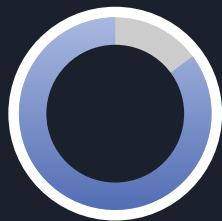
django

It has been used to create the backend of our project.

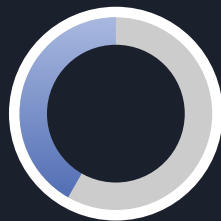
Batteries-included.



Fast development



Security: CSRF,
hashing



Highly scalable:
load balancing,
databases



Tech Stack



Fast Api

Containers

Images

Volumes

Dev Environments **BETA**

Learning Center

Extensions

Add Extensions

Docker Desktop

Upgrade plan

Search for local and remote images, containers, and more...

Ctrl+K

Sign in





Containers

Give feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Search

Only show running containers

	Name	Image	Status	Port(s)	Last started	Actions
<input type="checkbox"/>	 fastapi	-	Running (1/1)		30 minutes ago	  

Recipe with nutrition api:

```
GET /recipes/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 16,
  "user": 3,
  "title": "Dal Makhani",
  "steps": "1. Soak the black gram and rajma overnight. 2. Cook the soaked dal and rajma with",
  "image_url": "https://www.wavesvision.com/wp-content/uploads/2019/06/dal-makhani.jpg",
  "description": "Dal Makhani is a classic Punjabi dish made with black lentils (urad dal) and a",
  "short_description": "A rich and flavorful Indian dish made with black lentils and kidney beans",
  "ingredients": [
    {
      "name": "Black gram",
      "quantity": 1,
      "unit": "cup"
    }
  ]
}
```

pythonanywhere
by ANACONDA

Warning: You have not confirmed your email address yet. This means that you will not be able to reset your password anymore, send yourself a new one [here](#).

RecipeNutrition.pythonanywhere.co...

[Add a new web app](#)

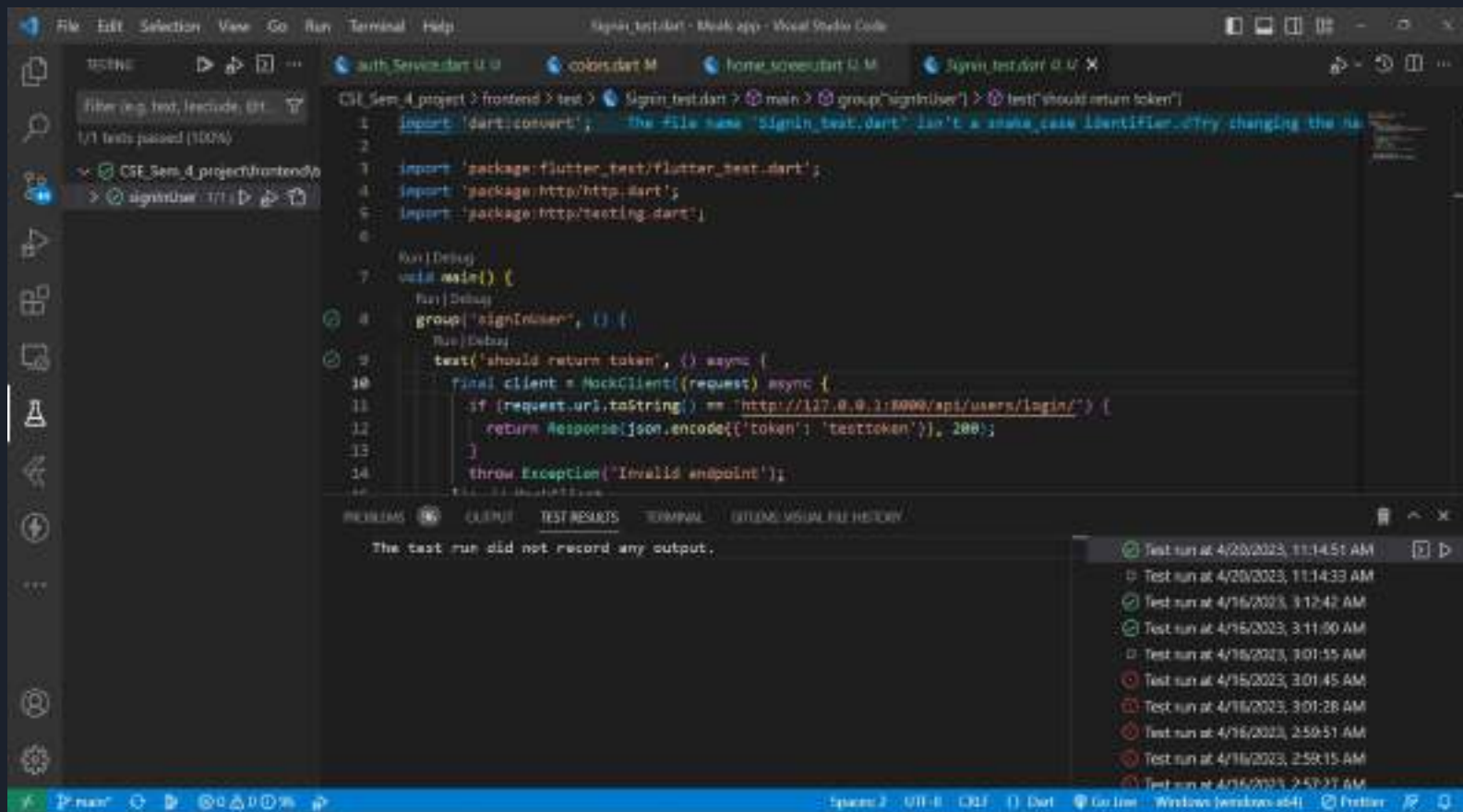
Configuration for
RecipeNutrition.pythonanywhere.com

Reload:

[Reload RecipeNutrition.pythonanywhere.com](#)

HOSTED AND TESTED API

Flutter Unit Testing



The screenshot displays an IDE window titled "Signin_test.dart - Visual Studio Code". The editor shows the following Dart code:

```
1 import 'dart:convert'; // the file name "Signin_test.dart" isn't a snake_case identifier. Try changing the name to "signin_test.dart"
2
3 import 'package:flutter_test/flutter_test.dart';
4 import 'package:http/http.dart';
5 import 'package:http/testing.dart';
6
7 Run|Debug
8 void main() {
9   Run|Debug
10   group('signInUser', () {
11     Run|Debug
12     test('should return token', () async {
13       final client = MockClient((request) async {
14         if (request.url.toString() == 'http://127.0.0.1:8000/api/users/login/') {
15           return Response.json.encode({'token': 'testtoken'}), 200;
16         }
17         throw Exception('Invalid endpoint');
18       });
19     });
20   });
21 }
```

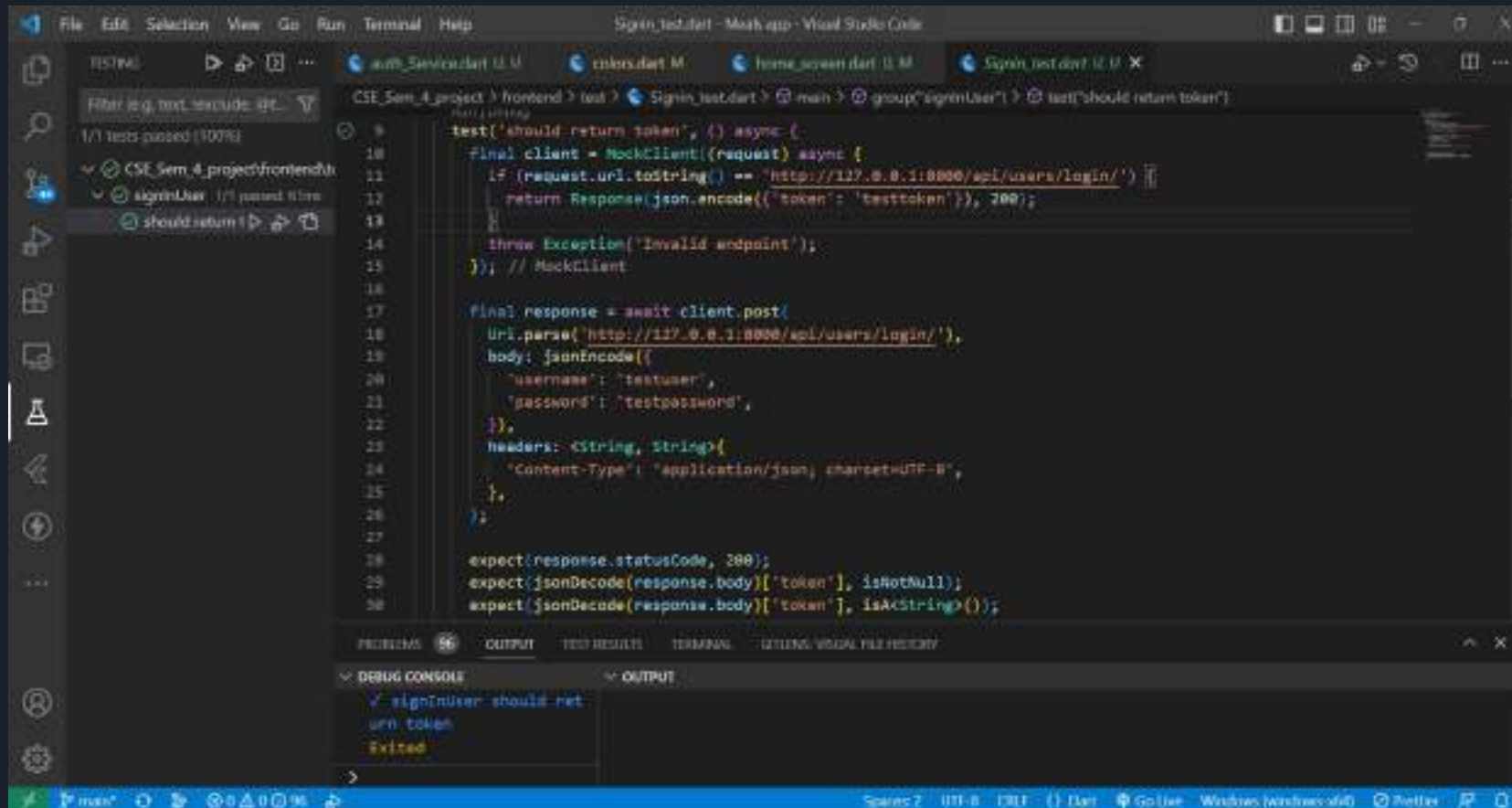
The left sidebar shows the file explorer with the project structure:

- CSE_Sem_4_project/frontend/
- signInUser / 1/1

The bottom panel shows the "TEST RESULTS" tab with the message: "The test run did not record any output." The right sidebar shows a list of test runs:

- Test run at 4/20/2023, 11:14:51 AM
- Test run at 4/20/2023, 11:14:33 AM
- Test run at 4/16/2023, 3:12:42 AM
- Test run at 4/16/2023, 3:11:00 AM
- Test run at 4/16/2023, 3:01:55 AM
- Test run at 4/16/2023, 3:01:45 AM
- Test run at 4/16/2023, 3:01:28 AM
- Test run at 4/16/2023, 2:59:51 AM
- Test run at 4/16/2023, 2:59:15 AM
- Test run at 4/16/2023, 2:57:21 AM

Flutter Unit Testing



CI/CD: Continuous Integration/Continuous Delivery (or Continuous Deployment)

[CHECK RELEASES HERE](#)

The screenshot displays the GitHub Actions interface for the repository **KarthikAvinash / 4epicure-releases**. The **Actions** tab is active, showing a list of workflow runs under the heading **All workflows**. The left sidebar shows the **Releases** section with 8 releases, the latest being **v1.0.9** (19 hours ago).

Workflow Runs:

Workflow	Status	Event	Branch	Actor	Time
chat_bot reject other questions	Success	Push	main	KarthikAvinash	2 days ago, 5m 12s
Update main.yml	Success	Push	main	KarthikAvinash	5 days ago, 4m 57s
youtube integrated	Success	Push	main	KarthikAvinash	5 days ago, 5m 14s
Update README.md	Success	Push	main	KarthikAvinash	last week, 5m 13s

Django unit testing

```
PS C:\Users\digitshikeb\Downloads\tests> & C:/Python311/python.exe c:/User
```

```
.....
```

```
Ran 4 tests in 5.399s
```

```
OK
```

```
1 import requests
2 import unittest
3
4 class TestGroceryItems(unittest.TestCase):
5     def setUp(self):
6         self.base_url = 'https://recipenutrition.pythonanywhere.com/'
7
8     def test_create_grocery_item(self):
9         url = self.base_url + 'users/1/grocery-items/'
10        data = {
11            'title': 'Milk',
12            'quantity': 2,
13            'unit': 'l'
14        }
15        response = requests.post(url, data=data)
16
17        self.assertEqual(response.status_code, 201)
18
19    def test_get_grocery_items(self):
20        url = self.base_url + 'grocery-items/get-user-grocery/1/'
21        response = requests.get(url)
22        self.assertEqual(response.status_code, 200)
23
24    def test_patch_grocery(self):
25        url = self.base_url + 'grocery-items/1/56/modify/'
26        data = {
27            'id': 56,
28            'user': 1,
29            'title': 'Mango Dalna',
30            'quantity': 2,
31            'unit': 'l'
32        }
33        response = requests.patch(url, data=data)
34        self.assertEqual(response.status_code, 200)
35
36    def test_get_grocery_items(self):
37        url = 'https://recipenutrition.pythonanywhere.com/grocery-items/'
38        response = requests.get(url)
39        self.assertEqual(response.status_code, 200)
40
41    def test_delete_grocery_item(self):
42        url = 'http://recipenutrition.pythonanywhere.com/grocery/delete/57/'
43        response = requests.delete(url)
44        self.assertEqual(response.status_code, 204)
45
46    if __name__ == '__main__':
47        unittest.main()
```




PS C:\Users\digitshikeb\Downloads\tests> & C:/Python

....

Ran 4 tests in 4.753s

OK

```
class TestRecipeAPI(unittest.TestCase):

    def setUp(self):

        self.url = 'https://recipenutrition.pythonanywhere.com/recipes/'
        self.headers = {
            'Content-Type': 'application/json'
        }
        self.data = {
            'user': 2,
            'title': 'corvan',
            'steps': 'Example steps for making the recipe',
            'image_url': 'https://example.com/recipe.jpg',
            'description': 'Example description of the recipe',
            'short_description': 'Example short description of the recipe',
            'ingredients': ['example ingredient 1', 'example ingredient 2'],
            'nutrition': {'calories': 200, 'protein': 10},
            'steps_with_images': [{ 'step': 'Example step 1', 'image_url': 'https://example.com/step1.jpg' },
                                  { 'step': 'Example step 2', 'image_url': 'https://example.com/step2.jpg' }],
            'rating': 4.5,
            'cook_time': '00:30:00'
        }

    def test_get_all_recipes(self):
        url = 'https://recipenutrition.pythonanywhere.com/recipes/'
        response = requests.get(url)
        self.assertEqual(response.status_code, 200)

    def test_get_recipe_by_title(self):
        url = 'https://recipenutrition.pythonanywhere.com/recipes/name/Carrot'
        response = requests.get(url)
        self.assertEqual(response.status_code, 200)

    def test_get_popular_recipes(self):
        url = 'https://recipenutrition.pythonanywhere.com/recipes/popular/'
        response = requests.get(url)
        self.assertEqual(response.status_code, 200)

    def test_create_recipe(self):
        response = requests.post(self.url, headers=self.headers, data=json.dumps(self.data))
        self.assertEqual(response.status_code, 201)
        self.assertEqual(response.json().get('title'), 'corvan')

if __name__ == '__main__':
    unittest.main()
```




PROJECT RESOURCES AND CONCLUSION

Locate to link: [App Releases](#)

