



4epicure

(Recipe & Meals App)
Software Engineering

CS301

[A CURE FOR EPICURE]

TOC

Overview

Features

Requirement analysis

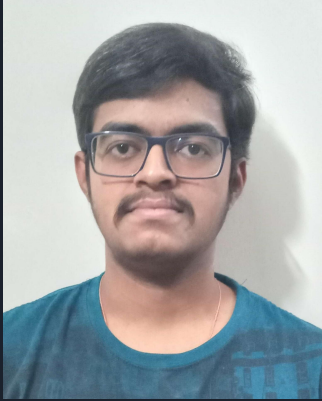
Prototype and demonstration

Tech stack

Testing



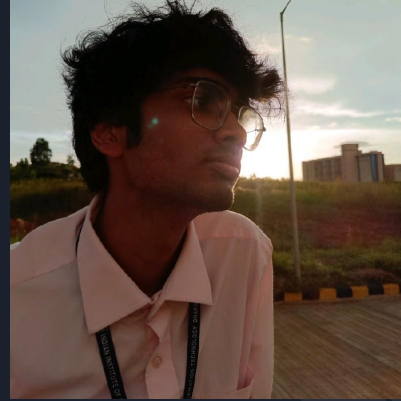
Our Team



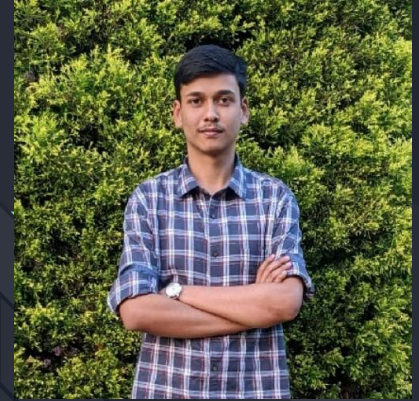
Karthik Avinash
21bcs052



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.

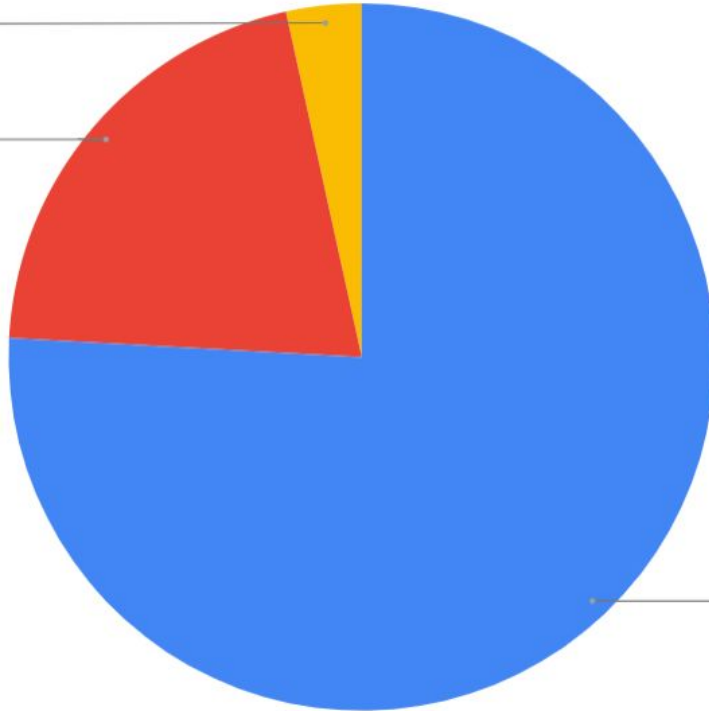
Requirement analysis

Do you know a platform to get everything related to meal planning and recipes at one place?

Maybe
3.4%

Yes
20.7%

No
75.9%



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

Maybe

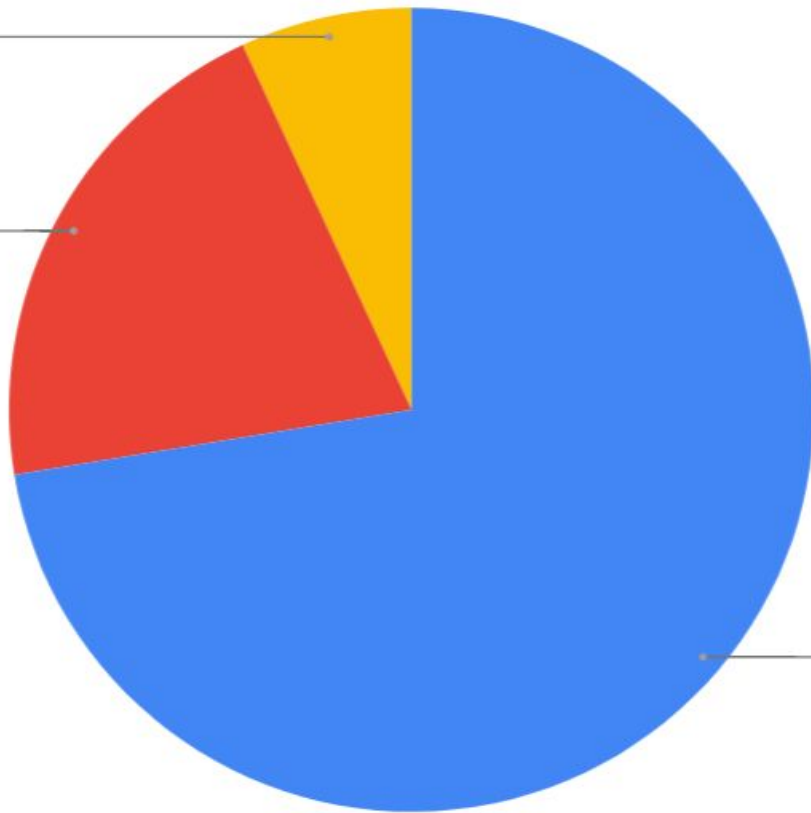
6.9%

No

20.7%

Yes

72.4%





PROTOTYPE

[Prototype Design Link](#)



DEMONSTRATION (LIVE DEMO)

JIRA workflow (Board)

PM

Jira Software Your work ▾ Projects ▾ Filters ▾ Dashboards ▾ Teams ▾ Apps ▾ **Create**

MEALS_APP
Software project

PLANNING

- Roadmap
- Backlog
- Board**

DEVELOPMENT

- Code
- Project pages
- Add shortcut
- Project settings

You're eligible for a free 14-day trial of our Jira Software Standard plan. [Start now!](#)

Projects / MEALS_APP

Sprint - Authentication pages

Complete backend and frontend for authentication part

0 days remaining **Complete sprint**

GROUP BY: None ▾ Insights

TO DO 4 ISSUES

- splash screen
MF-4 (K)
- 2nd screen
MF-5 (K)
- otp verification screen
MF-7 (SY)
- start backend
MF-8 (K)

IN PROGRESS 1 ISSUE

- forgot password screen
MF-6 (SY)

DONE 3 ISSUES

- Signup Screen
MF-3 (SY)
- Login Screen
MF-2 (SY)
- First screen
MF-1 (SY)

Quickstart

- ✓ Create a project ▾
- ✓ Customize your board ▾
- ✓ Create an issue ▲

What is an issue in Jira?

Issues are individual pieces of work that you assign to teammates.

Issues can be tasks or stories.

Show me View issue tutorial- ✓ Invite your teammates ▾

Dismiss Quickstart

Project Management - JIRA workflow(Backlog)

Jira Software Your work ▾ **Projects** ▾ Filters ▾ Dashboards ▾ Teams ▾ Apps ▾ **Create**

MEALS_APP
Software project

▼ **PLANNING**

- Roadmap
- Backlog**
- Board

DEVELOPMENT

- Code
- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
[Learn more](#)

Backlog

Projects / MEALS_APP

Q Search

SY K

Epic ▾

Insights

▼ **Sprint - Authentication pages** 24 Feb – 4 Mar (8 issues) **Complete sprint**

Complete backend and frontend for authentication part

✓ MF-3 Signup Screen	DONE ▾ SY
✓ MF-2 Login Screen	DONE ▾ SY
✓ MF-4 First screen	DONE ▾ SY
✓ MF-4 splash screen	TO DO ▾ K
✓ MF-5 2nd screen	TO DO ▾ K
✓ MF-6 forgot password screen	IN PROGRESS ▾ SY
✓ MF-7 otp verification screen	TO DO ▾ SY
✓ MF-8 start backend	TO DO ▾ K

+ Create issue

Quickstart

- ✓ Create a project ▾
- Create an issue ▴

What is an issue in Jira?

Issues are individual pieces of work that you assign to teammates.

Issues can be tasks or stories.

Show me View issue tutorial

- Invite your teammates ▾
- Connect your tools ▾

[Dismiss Quickstart](#)

<https://sam.atlassian.net/jira/software/projects/MF/boards/2/backlog>


APIs



Nutrition Analysis API

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

3:26 Create Recipe



Add Photo

Title
Shahi Paneer

Description
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.

Ingredients

Add Ingredient

Add

Ingredients:

- 200 gram paneer
- 100 gram pea

3:26 Create Recipe

Description
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.

Ingredients

Add Ingredient

Add

Ingredients:

- 200 gram paneer
- 100 gram pea
- 2 gram garam masala
- 100 gram curd

Steps

Add Step

Add

Steps:

1. fake it until you make it!!

Get Nutrient Information

Nutrient Information:

Calories: 196.0 kcal
Protein: 22.2 g
Carbs: 6.8 g
Fat: 8.6 g

```
GET /api/

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

[
  {
    "id": 1,
    "name": "Spaghetti Carbonara",
    "steps": [],
    "user": {
      "id": 2,
      "username": "kendrick"
    },
    "ingredients": [
      {
        "name": "spaghetti",
        "quantity": 500,
        "unit": "grams"
      },
      {
        "name": "pancetta",
        "quantity": 150,
        "unit": "grams"
      }
    ],
    "liked_by": [
      {
        "id": 1,
        "username": "devarshi"
      }
    ],
    "disliked_by": [
      {
        "id": 2,
        "username": "kendrick"
      }
    ]
  }
]
```

Localization of 4Epicure in REST API using python googletrans library

Django REST framework

devesh

Recipe List / Recipe Detail

Recipe Detail

DELETE

OPTIONS

GET

GET /api/1/

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

```
{\"id\": 1, \"name\": \"\\u0938\\u094d\\u092a\\u0947\\u0918\\u091f\\u0940\\u0915\\u093e\\u0930\\u092c\\u094b\\u0928\\u093e\\u0930\\u093e\\\", \"steps\": [],
```

DK

```
\\u0938\\u094d\\u092a\\u0947\\u0918\\u091f\\u0940\\u0915\\u093e\\u0930\\u092c\\u094b\\u0928\\u093e\\u0930\\u093e\\
```



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:

स्पेघटी कारबोनारा

The above text is in the Hindi language and could be translated to English as "Spegathi Carbonara".



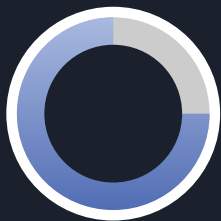
Tech Stack 01

Django

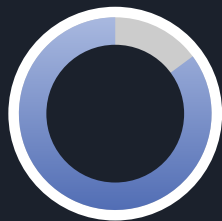
django

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

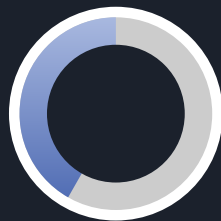
Batteries-included.



Fast development



Security: CSRF,
hashing



Highly scalable:
load balancing,
databases



Tech Stack



Fast Api

Docker Desktop

Upgrade plan

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

Ctrl+K



Sign in



Containers



Images



Volumes



Dev Environments **BETA**



Learning Center



Extensions



Add Extensions

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



[fastapi](#)

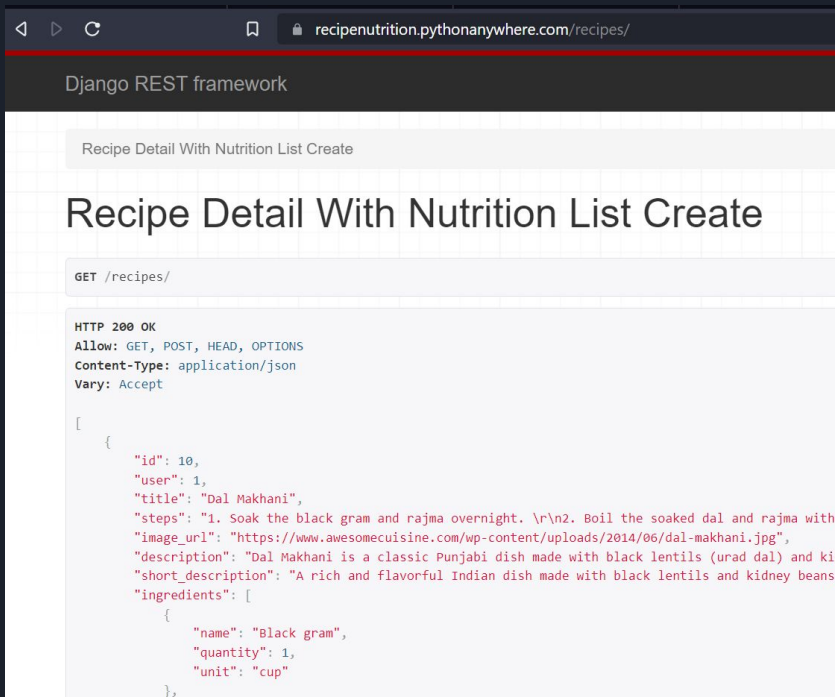
-

Running (1/1)

30 minutes ago



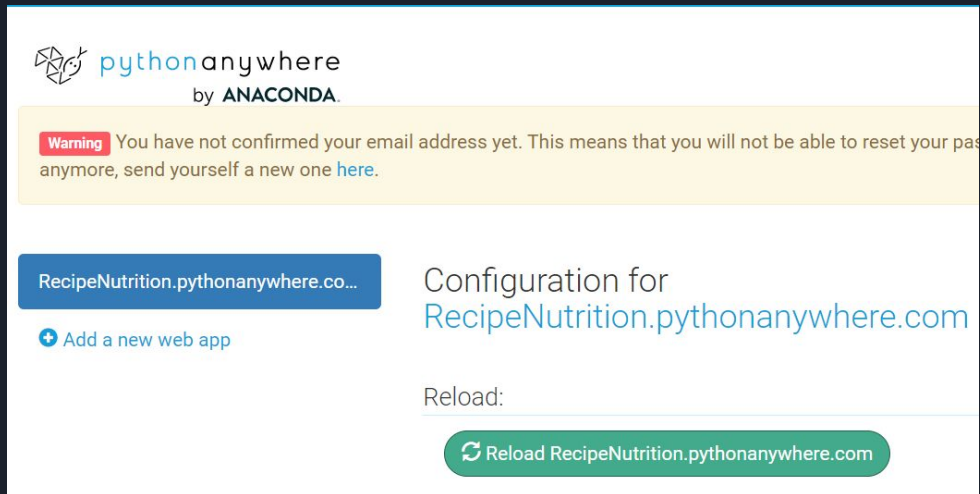
Recipe with nutrition api:



```
GET /recipes/

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

[
  {
    "id": 10,
    "user": 1,
    "title": "Dal Makhani",
    "steps": "1. Soak the black gram and rajma overnight. \r\n2. Boil the soaked dal and rajma with",
    "image_url": "https://www.awesomecuisine.com/wp-content/uploads/2014/06/dal-makhani.jpg",
    "description": "Dal Makhani is a classic Punjabi dish made with black lentils (urad dal) and kidney beans",
    "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

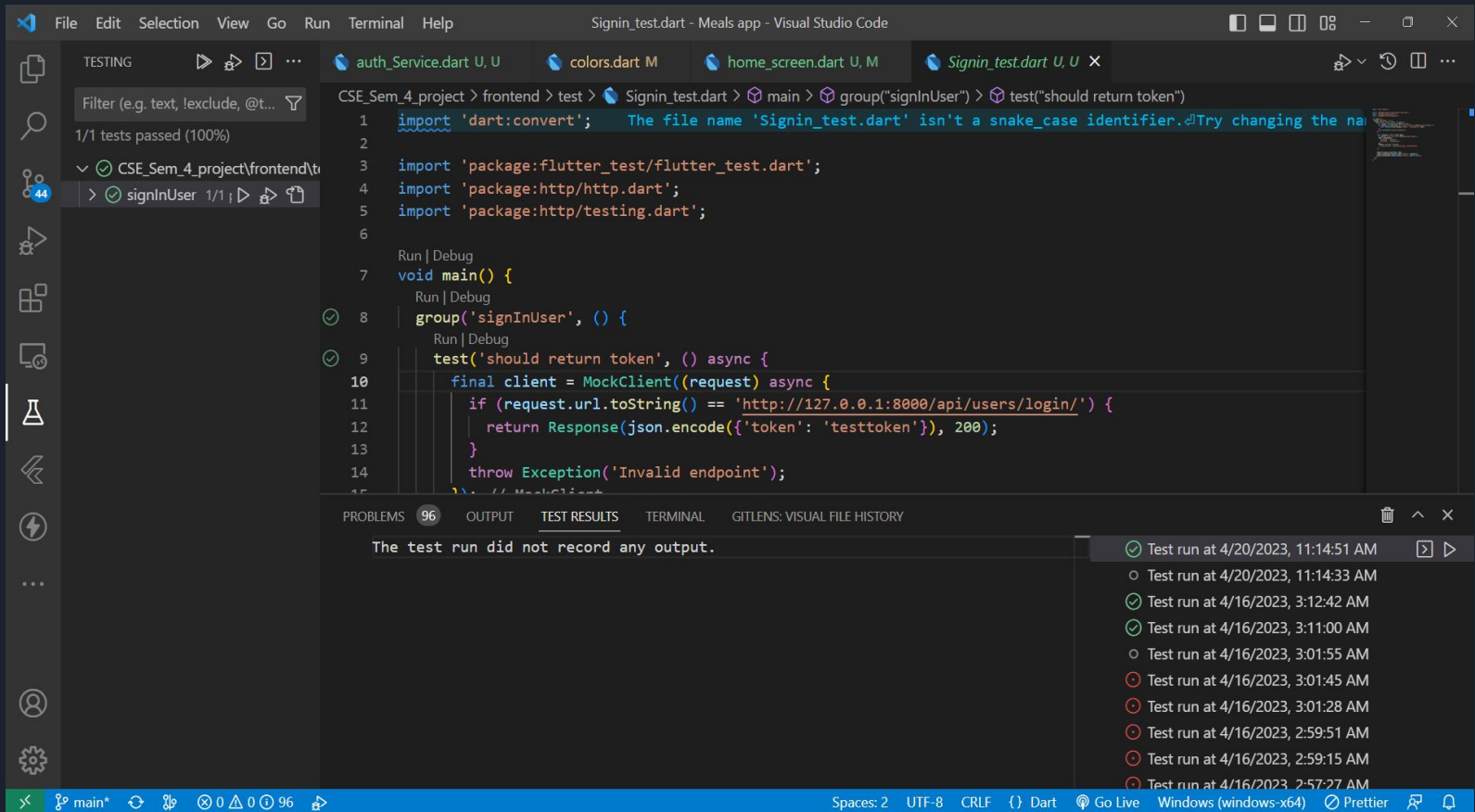
Configuration for
RecipeNutrition.pythonanywhere.com

Reload:

Reload RecipeNutrition.pythonanywhere.com

HOSTED AND TESTED API

Flutter Unit Testing



Visual Studio Code interface showing Flutter unit testing results for the Signin app.

TESTING

Filter (e.g. text, !exclude, @t...)

1/1 tests passed (100%)

✓ CSE_Sem_4_project(frontend\tr...)

> ✓ signInUser 1/1

Signin_test.dart

```
1 import 'dart:convert';
2
3 import 'package:flutter_test/flutter_test.dart';
4 import 'package:http/http.dart';
5 import 'package:http/testing.dart';
6
7 void main() {
8   group('signInUser', () {
9     test('should return token', () async {
10       final client = MockClient((request) async {
11         if (request.url.toString() == 'http://127.0.0.1:8000/api/users/login/') {
12           return Response(json.encode({'token': 'testtoken'}), 200);
13         }
14         throw Exception('Invalid endpoint');
15       });
```

PROBLEMS 96 **OUTPUT** **TEST RESULTS** **TERMINAL** **GITLENS: VISUAL FILE HISTORY**

The test run did not record any output.

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, 2:57:27 AM

Spaces: 2 UTF-8 CRLF Dart Go Live Windows (windows-x64) Prettier

Flutter Unit Testing

The screenshot displays the Visual Studio Code editor with the following components:

- TESTING Panel (Left Sidebar):** Shows a filter input and test results. Under "CSE_Sem_4_project(frontend)", the "signInUser" test is listed with "1/1 passed: 61ms". The specific test "should return token" is expanded and marked as passed.
- Editor (Center):** Displays the file `Signin_test.dart` within the "Meals app" workspace. The code is a unit test for the `signInUser` function, using a `MockClient` to simulate a successful login response. The test is structured as follows:

```
test('should return token', () async {  
  final client = MockClient((request) async {  
    if (request.url.toString() == 'http://127.0.0.1:8000/api/users/login/') {  
      return Response(json.encode({'token': 'testtoken'}), 200);  
    }  
    throw Exception('Invalid endpoint');  
  }); // MockClient  
  
  final response = await client.post(  
    Uri.parse('http://127.0.0.1:8000/api/users/login/'),  
    body: jsonEncode({  
      'username': 'testuser',  
      'password': 'testpassword',  
    }),  
    headers: <String, String>{  
      'Content-Type': 'application/json; charset=UTF-8',  
    },  
  );  
  
  expect(response.statusCode, 200);  
  expect(jsonDecode(response.body)['token'], isNotNull);  
  expect(jsonDecode(response.body)['token'], isA<String>());  
});
```
- Bottom Panel:** Contains three tabs: "PROBLEMS" (showing 96 errors), "OUTPUT", and "DEBUG CONSOLE". The "DEBUG CONSOLE" tab is active, showing the output of the test: `✓ signInUser should return token` and `Exited`.
- Status Bar (Bottom):** Indicates the current file is `main`, the workspace has 2 spaces, and the editor is using UTF-8 encoding with CRLF line endings. It also shows the Dart SDK path and the Go Live extension.

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

[CHECK RELEASES HERE](#)

KarthikAvinash / 4epicure-releases Private

Unwatch 1 Fork 0 Star 0

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

Actions

New workflow

All workflows

Build & Release

Management

Caches

Releases 8

v1.0.9 Latest 19 hours ago

+ 7 releases

All workflows

Showing runs from all workflows

Filter workflow runs

7 workflow runs

	Event	Status	Branch	Actor
✓ chat_bot reject other questions Build & Release #8: Commit 9d7c1c4 pushed by KarthikAvinash	main	2 days ago 6m 12s	...	
✓ Update main.yml Build & Release #7: Commit 06b8715 pushed by KarthikAvinash	main	5 days ago 4m 51s	...	
✓ youtube integrated Build & Release #6: Commit 69e70f3 pushed by KarthikAvinash	main	5 days ago 5m 14s	...	
✓ Update README.md Build & Release #5: Commit 93c9dab pushed by KarthikAvinash	main	last week 5m 15s	...	



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-usr-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/1/58/modify/'
26        data = {
27            'id': 56,
28            'user': 1,
29            'title': 'Mango Juice',
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': 'carvan',
    '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'), 'carvan')
```

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




PROJECT RESOURCES AND CONCLUSION

Locate to link: [App Releases](#)

