Cookbook - Virtual Kitchen

A project work
submitted for the partial fulfillment for the
award of degree in

NAAN MUDHALVAN – PROJECT DEVELOPMENT COURSE

COLLEGE CODE : UNM1441

BACHELOR OF SOFTWARE APPLICATIONS

BY

BASHEER.I	**- 222208906**
CHARU KUMAR.S	**- 222208907**
DEENADAYANT.P	**- 222208908**
DEEPAK.A	**- 222208909**
HARISH.S	**- 222208911**

SREE MUTHUKUMARASWAMY COLLEGE
(AFFILIATED TO UNIVERSITY OF MADRAS)
KODUNGAIYUR, CHENNAI – 600 118

APRIL 2025 EXAMINATIONS

BONAFIDE CERTIFICATE

This is to certify that the project entitled “**YOUR PROJECT TOPIC**” being submitted to Sree Muthukumaraswamy College, College Code: UNM1441 Kodungaiyur, Chennai – 600118, by group of students in partial fulfillment for the award of the degree of B.Sc., Software Applications is a bonafide record of the work carried out by her under my guidance and supervision.

Internal Guide	**Head of the Department**
(Mr. M.Kalaimani)	**(Mrs.T.Merlin Jaba)**

DECLARATION

I hereby declare that this project titled “**Your Project Topic**” submitted by me in partial fulfillment of the requirements for the Bachelor Degree of Software Applications has not formed a basis for the award of any other degree, diploma, associate, fellowships or other similar titles and this project was fully developed by us.

NAME OF THE STUDENT	REGISTER NO	SIGNATURE
1.BASHEER.I	222208906	
2. CHARU KUMAR.S	222208907	
3.DEENA DHAYANTH.P	222208908	
4.DEEPAK.A	222208909	
5.HARISH.S	222208911	

Place : Chennai - 600 118

Date : 08-03-2025

Sree Muthukumaraswamy College Kodungaiyur Ch-118

Project Documentation: Cookbook - Virtual Kitchen

Team Members:

- Basheer I - Backend Development

- Charu Kumar S - Data Handling

- Deena Dhayanth P - Database Management

- Deepak A - Frontend Developer

- Harish S - User Interface

1. Introduction

Project Title: Cookbook: Virtual Kitchen

Welcome to Cookbook: A Virtual Kitchen, your ultimate destination for all things food. Explore, create, and share recipes with a community of fellow food enthusiasts.

---

2. Project Overview

- Purpose:

The Cookbook: Virtual Kitchen is a Python-based application designed to manage, search, and suggest recipes based on available ingredients. It acts as a virtual assistant for home cooks, providing step-by-step guidance for various dishes.

- Features:

- Search recipes by ingredients.

- Add/Edit/Delete personal recipes.

- Categorized recipes (Breakfast, Lunch, Dinner, Snacks).

- Favorite recipe bookmarking.

- Automatic ingredient list generator for shopping.

---

3. Architecture

### Overall Architecture

- Core Application: Python (Flask/FastAPI optional for web interface).

- Data Storage: JSON files or SQLite.

- Interface: CLI (initial), optional Web UI.

Key Modules

- `recipes_manager.py`: CRUD operations for recipes.

- `search_engine.py`: Search recipes based on input.

- `shopping_list.py`: Generate shopping lists.

- `user_preferences.py`: Handle favorites/preferences.

---

4. Setup Instructions

Prerequisites

- Python 3.10+
- Pip

Installation

```bash
git clone https://github.com/your-repo/virtual-kitchen.git
cd virtual-kitchen
python -m venv venv
source venv/bin/activate   # Windows: venv\Scripts\activate
pip install -r requirements.txt
```

---

5. Folder Structure

```
/virtual-kitchen
|-- app/
|   |-- recipes_manager.py
|   |-- search_engine.py
|   |-- shopping_list.py
|   |-- user_preferences.py
|-- data/
|   |-- recipes.json
|   |-- categories.json
|-- tests/
|-- docs/
|-- main.py
|-- requirements.txt
```

---

6. Running the Application

Run main application:

```bash
python main.py
```

Run tests:

```bash
pytest tests/
```

---

7. Module Documentation

Module	Purpose
recipes_manager.py	Manages loading, saving, updating recipes.
search_engine.py	Handles recipe search.
shopping_list.py	Generates shopping lists.
user_preferences.py	Manages favorites and user preferences.

---

8. State Management

- Recipes and user data are stored in `JSON`.
- Future: Upgrade to SQLite/PostgreSQL for multi-user support.

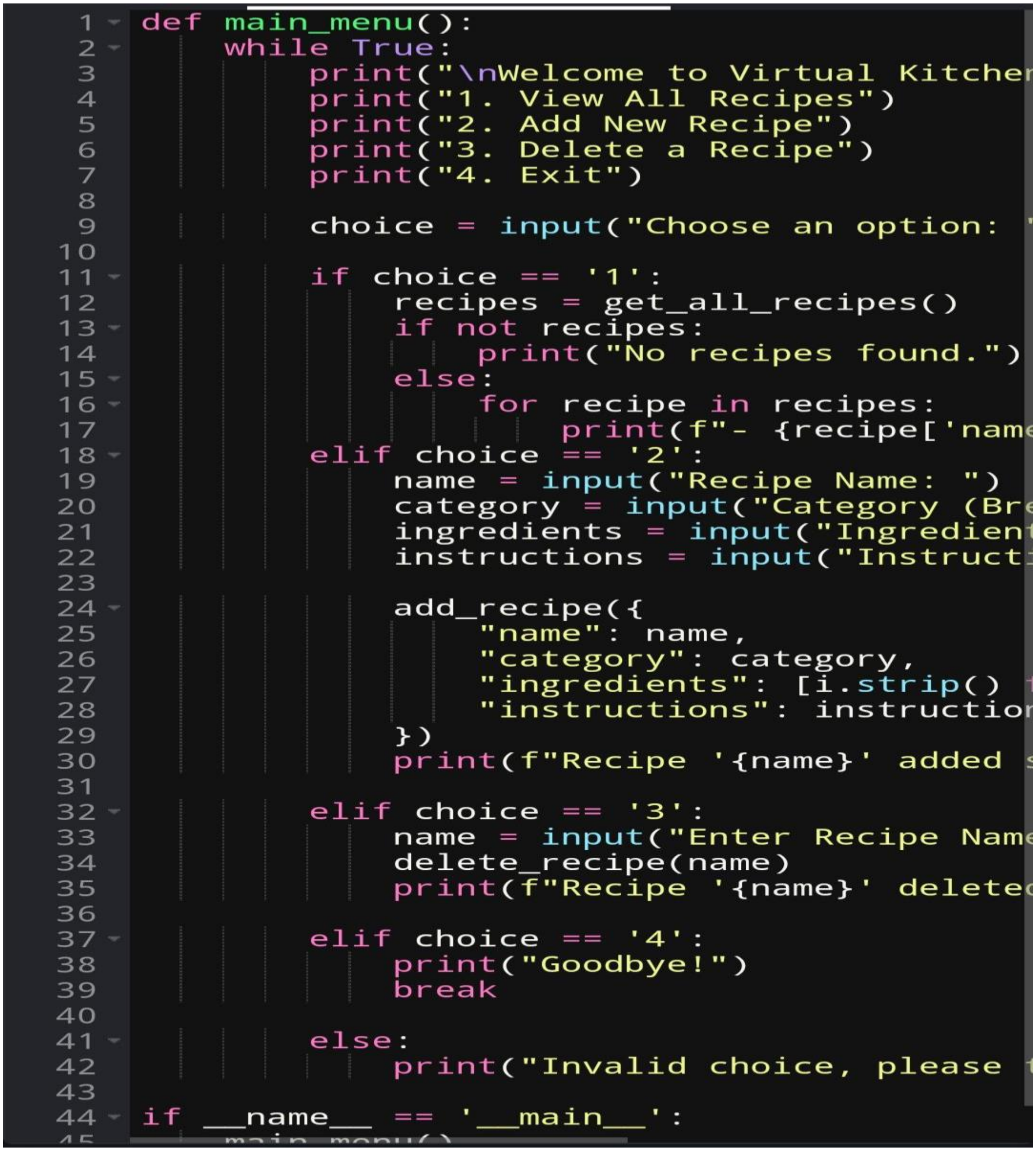
---

9. User Interface

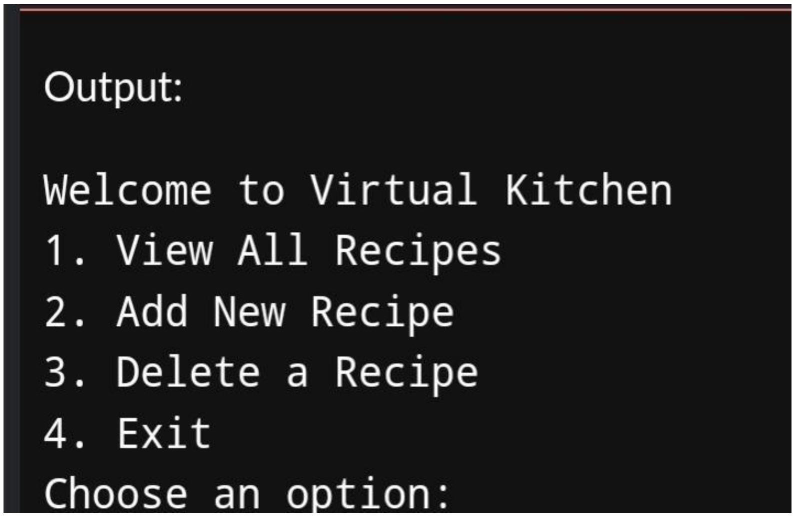
- CLI Menu-based interaction.

Example:

```
Welcome to Virtual Kitchen!

1. Search Recipe

2. Add New Recipe

3. View Favorites

4. Generate Shopping List

5. Exit
```



```
def main_menu():
    while True:
        print("\nWelcome to Virtual Kitcher
        print("1. View All Recipes")
        print("2. Add New Recipe")
        print("3. Delete a Recipe")
        print("4. Exit")

        choice = input("Choose an option: '

        if choice == '1':
            recipes = get_all_recipes()
            if not recipes:
                print("No recipes found.")
            else:
                for recipe in recipes:
                    print(f"- {recipe['name
        elif choice == '2':
            name = input("Recipe Name: ")
            category = input("Category (Bre
            ingredients = input("Ingredient
            instructions = input("Instructi

            add_recipe({
                "name": name,
                "category": category,
                "ingredients": [i.strip() f
                "instructions": instructior
            })
            print(f"Recipe '{name}' added s

        elif choice == '3':
            name = input("Enter Recipe Name
            delete_recipe(name)
            print(f"Recipe '{name}' deleted

        elif choice == '4':
            print("Goodbye!")
            break

        else:
            print("Invalid choice, please t

if __name__ == '__main__':
    main_menu()
```



Output:

```
Welcome to Virtual Kitchen
1. View All Recipes
2. Add New Recipe
3. Delete a Recipe
4. Exit
Choose an option:
```

10. Styling (optional if web UI added)

- Text colors via `colorama`.
- If Flask frontend added: Bootstrap or Tailwind.

---

11. Testing

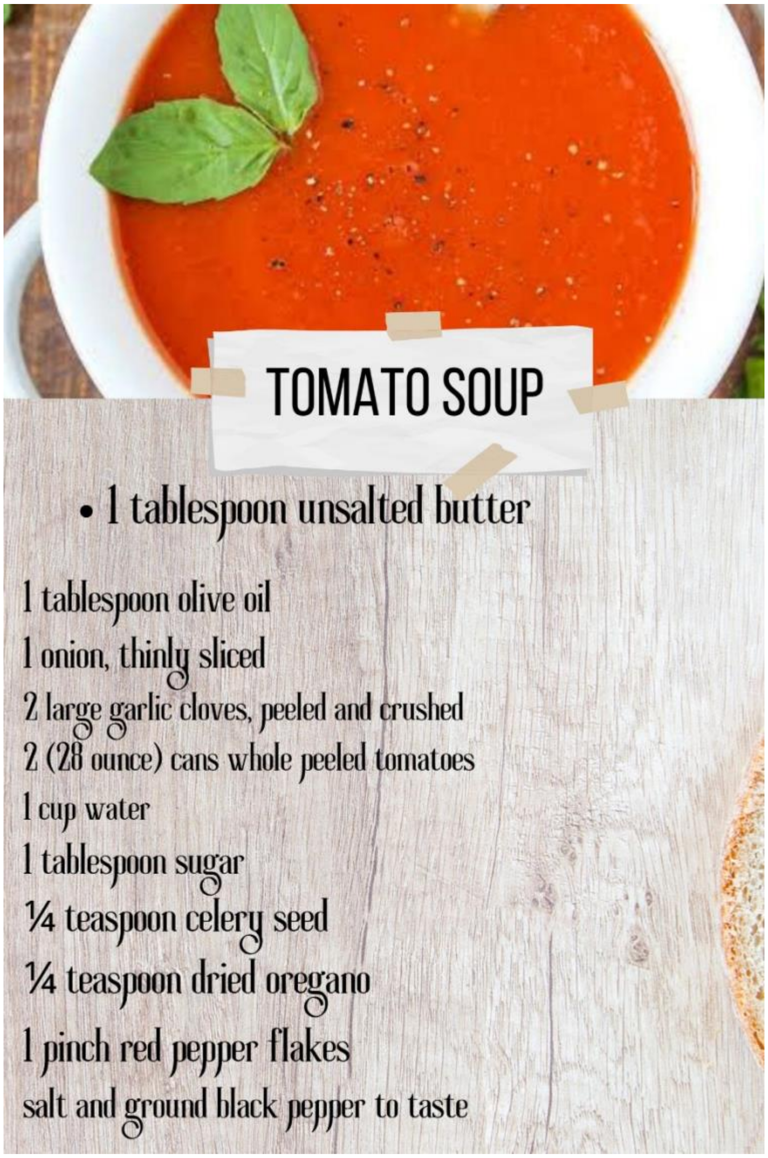
- `pytest` for unit tests.
- `unittest.mock` for mocking file handling.

---

12. Screenshots or Demo

CLI Example:

```
Enter Ingredient: Tomato

Found Recipes: Tomato Soup, Tomato Rice
```



TOMATO SOUP

- 1 tablespoon unsalted butter

1 tablespoon olive oil
1 onion, thinly sliced
2 large garlic cloves, peeled and crushed
2 (28 ounce) cans whole peeled tomatoes
1 cup water
1 tablespoon sugar
¼ teaspoon celery seed
¼ teaspoon dried oregano
1 pinch red pepper flakes
salt and ground black pepper to taste

---

13. Known Issues

- Case-sensitive ingredient matching.
- No authentication system.

---

14. Future Enhancements

- Full Flask/React web interface.
- API connection to Spoonacular.
- Meal planner.
- Multi-language support.
- AI-based recipe suggestions.

Sample Folder Structure + `recipes_manager.py`

Folder Tree Example

```
virtual-kitchen/
├── app/
|   ├── recipes_manager.py
|   ├── search_engine.py
|   ├── shopping_list.py
|   ├── user_preferences.py
├── data/
|   ├── recipes.json
|   ├── categories.json
├── docs/
├── tests/
├── main.py
```

├── requirements.txt
~~~

app/recipes_manager.py (Sample Code)

```python
import json
import os

DATA_PATH = os.path.join(os.path.dirname(__file__), '..', 'data', 'recipes.json')

def load_recipes():
    if not os.path.exists(DATA_PATH):
        return []
    with open(DATA_PATH, 'r') as file:
        return json.load(file)
~~~

```
def save_recipes(recipes):
    with open(DATA_PATH, 'w') as file:
        json.dump(recipes, file, indent=4)


def add_recipe(recipe):
    recipes = load_recipes()
    recipes.append(recipe)
    save_recipes(recipes)


def get_all_recipes():
    return load_recipes()


def delete_recipe(recipe_name):
    recipes = load_recipes()
    recipes = [r for r in recipes if r['name'].lower() !=
recipe_name.lower()]
    save_recipes(recipes)


if __name__ == '__main__':
    # Sample usage for testing
    add_recipe({
        "name": "Tomato Soup",
        "ingredients": ["Tomato", "Salt", "Pepper"],
        "category": "Soup",
        "instructions": "Boil tomatoes, add salt and pepper, blend and serve."
    })

    print("All Recipes:")
    for recipe in get_all_recipes():
        print(recipe['name'])
```

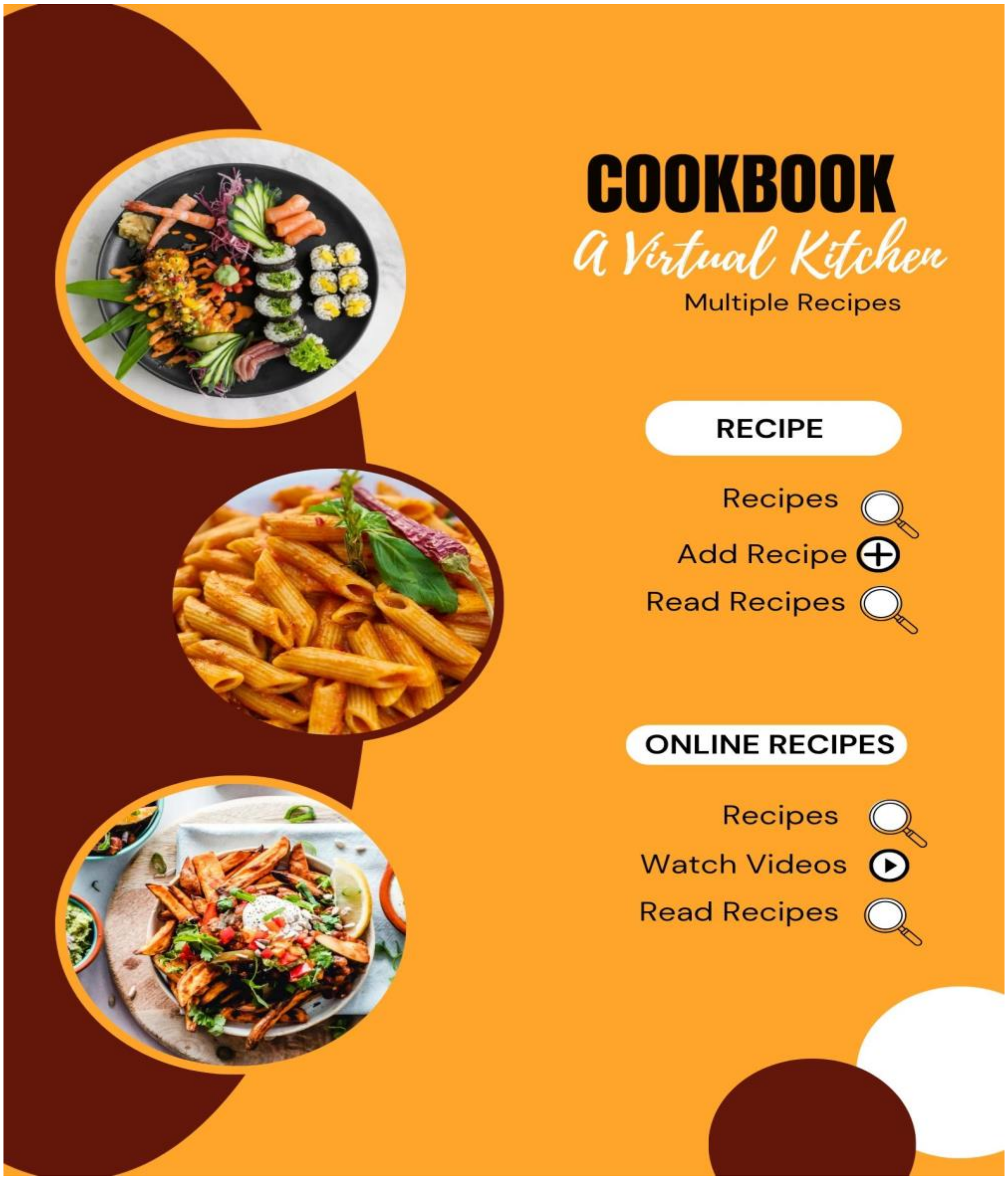
Bonus: requirements.txt

Market Research & User Feedback

Identify Target Audience: Understand the needs of home cooks, health-conscious individuals, and food enthusiasts.

Competitive Analysis: Study similar apps to determine unique features and improvements.

User Testing: Gather early feedback to refine features before full-scale deployment.


COOKBOOK
A Virtual Kitchen
Multiple Recipes
RECIPE
Recipes
Add Recipe
Read Recipes
ONLINE RECIPES
Recipes
Watch Videos
Read Recipes