



LLM driven recipe recommendation application design

Doan Duc Anh Dinh

A thesis report submitted in partial fulfilment of the

requirements for the award of the degree of

BACHELOR OF COMPUTER SCIENCE

CHARLES DARWIN UNIVERSITY

COLLEGE OF ENGINEERING, INFORMATION TECHNOLOGY AND ENVIRONMENT

October 2024

DECLARATION

I hereby declare that the work herein, now submitted as a thesis report for the degree of Bachelor of Computer Science at Charles Darwin University, is the result of my own investigations, and all references to ideas and work of other researchers have been specifically acknowledged. I hereby certify that the work embodied in this thesis report has not already been accepted in substance for any degree and is not being currently submitted in candidature for any other degree.

A handwritten signature in black ink, appearing to read 'Duck', written over a horizontal line.

Signature:

Date: 17 October 2024

DEDICATION

This thesis report paper is sincerely dedicated to my family, who has always been there to support me. Their love and encouragement helped me throughout my research and writing. I am forever grateful for their love and support.

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to my first supervisor, Dr. Yakub Sebastian, for his constant efforts to supporting me in my research and development process. I am deeply grateful for his guidance and knowledge in the field of large language models leading my path to a successful project.

TABLE OF CONTENTS

1. TABLE OF CONTENTS

DECLARATION	I
DEDICATION	II
ACKNOWLEDGEMENTS	III
LIST OF FIGURES	2
LIST OF TABLES	5
LIST OF ABBREVIATIONS	6
I. INTRODUCTION	7
BACKGROUND	7
AIM OF RESEARCH	7
STRUCTURE OF PAPER	7
II. APPROACH	8
METHODOLOGY	8
PROJECT CONCEPTION AND RESEARCH	8
COLLECTING DATA MODEL	8
APP DEVELOPMENT	8
DATA ANALYSIS	9
LIMITATION	9
III. RESULTS	9
INSTALLATION AND SET UP	9
FEATURES OF THE FOOD RECOMMENDER	10
USER INPUT AND OUTPUT EXAMPLES	11
FOOD WASTE AUDIT FEATURES	12
TESTING AND GAINING FEEDBACK	13
IV. ANALYSIS AND DISCUSSION OF RESULTS	13
V. CONCLUSIONS	16
VI. FUTURE WORK	16
APPENDIX A: LITERATURE REVIEW	17
INTRODUCTION	17
TRADITIONAL METHODS OF MACHINE LEARNING RECIPE RECOMMENDATIONS	18
LLM - GENERATED RESPONSE CURRENT USAGE AND STUDIES	20
CONCLUSION	22
APPENDIX B: DATA AND PROGRAMS USED FOR THIS REPORT	23
APPENDIX C: FIGURES	24
APPENDIX D: TABLES	37
REFERENCES	41

LIST OF FIGURES

Figure 1, Model driven design interface	8
Figure 2, Control flow of MVT	9
Figure 3, pip install command	9
Figure 4, Installing the virtual environment	10
Figure 5, Set up the virtual environment	10
Figure 6, Starting the virtual environment	10
Figure 7, Install Django package	10
Figure 8, Install Django haystack package	10
Figure 9, Install OpenAI package	10
Figure 10, Install whoosh package	10
Figure 11, Running the server	10
Figure 12, Basic interface of the recommender	10
Figure 13, Mobile view of the recommender	11
Figure 14, Logged in view	11
Figure 15, Input example 1	11
Figure 16, Output example 1	11
Figure 17, Input example 2	11
Figure 18, Input example 2	11
Figure 19, Input example 3	12
Figure 20, Output example 3	12
Figure 21, Input example 4	12
Figure 22, Output example 4	12
Figure 23, Adding new food items	12
Figure 24, Food item log	12
Figure 25, Adding a new waste audit item	12
Figure 26, Waste audit log	13
Figure 27, Recent waste items	13
Figure 28, RecipeIS home page upload image	13
Figure 29, Recognized the ingredient and use that as the main ingredient in the recipe recommender	13

Figure 30, Food recommendation and explanation based on nutrition effect from the ChatDiet	14
Figure 31, Personalized feedback from ChatDiet	14
Figure 32, Supercook web application	15
Figure 33, List of suggested recipes	15
Figure 34, Appendix photo Model driven design interface	24
Figure 35, Appendix photo Control flow of MVT	24
Figure 36, Appendix photo pip install command	24
Figure 37, Appendix photo Installing the virtual environment	25
Figure 38, Appendix photo Set up the virtual environment	25
Figure 39, Appendix photo Starting the virtual environment	25
Figure 40, Appendix photo Install Django package	25
Figure 41, Appendix photo Install Django haystack package	26
Figure 42, Appendix photo Install OpenAI package	26
Figure 43, Appendix photo Install whoosh package	26
Figure 44, Appendix photo Running the server	26
Figure 45, Appendix photo Basic interface of the recommender	27
Figure 46, Appendix photo Mobile view of the recommender	27
Figure 47, Appendix photo Logged in view	28
Figure 48, Appendix photo Input example 1	29
Figure 49, Appendix photo Output example 1	29
Figure 50, Appendix photo Input example 2	30
Figure 51, Appendix photo Input example 2	30
Figure 52, Appendix photo Input example 3	31
Figure 53, Appendix photo Output example 3	31
Figure 54, Appendix photo Input example 4	32
Figure 55, Appendix photo Output example 4	32
Figure 56, Appendix photo Adding new food items	32
Figure 57, Appendix photo Food item log	32
Figure 58, Appendix photo Adding a new waste audit item	33
Figure 59, Appendix photo Waste audit log	33
Figure 60, Appendix photo Recent waste items	33
Figure 61, Appendix photo RecipeIS home page upload image	34

Figure 62, Appendix photo Recognized the ingredient and use that as the main ingredient in the recipe recommender	34
Figure 63, Appendix photo Food recommendation and explanation based on nutrition effect from the ChatDiet	35
Figure 64, Appendix photo Personalized feedback from ChatDiet	35
Figure 65, Appendix photo Supercook web application	36
Figure 66, Appendix photo List of suggested recipes	36

LIST OF TABLES

Table 1, RecipeIS usability analysis	13
Table 2, ChatDiet usability analysis	14
Table 3, Supercook usability analysis	15
Table 4, Food waste recipe recommender usability analysis	15
Table 5, Appendix table RecipeIS usability analysis	37
Table 6, Appendix table ChatDiet usability analysis	38
Table 7, Appendix table Supercook usability analysis	39
Table 8, Appendix table Food waste recipe recommender usability analysis	40

LIST OF ABBREVIATIONS

Abbreviation 1, LLM(s) = large language model(s)

Abbreviation 2, AI = artificial intelligence

Abbreviation 3, MVC = Model-View-Controller

Abbreviation 4, MVT = Model-View-Template

Abbreviation 5, UX = user experience

Abbreviation 5, UI = user interface

LLM Driven Recipe Recommendation Application Design

Abstract— This research will aim to address key contributors such as insufficient shopping and meal planning, overbuying and improper storage to the food waste problem in Australia. The approach to mitigate this issue will be to reduce food waste through a recipe recommendation system that utilizes leftover ingredients. In the past other recipe recommendation systems have used the traditional method of creating models for their recommender system which limits output due to the lower data size. The aim is to leverage advanced large language models (LLMs) to build a web application that recommends user's recipes based on entered ingredients. The programmed application utilizes GPT-4 model which is a popular generative AI model used as the backend for ChatGPT to take ingredients that the user inputs and generate a food recipe based on those inputs. By integrating GPT-4, a popular generative AI model, the web application can generate more creative and context-aware recipe recommendations based on user input, allowing for more flexibility and personalization. The system not only recommends recipes based on entered ingredients but also includes a waste audit feature that helps users log food waste and select recent waste items for use in the recipe generation process. Through this innovative approach, the application aims to empower users to make more informed decisions about their food consumption, reduce food waste, and ultimately contribute to sustainability efforts.

Keywords — Generative AI, Large Language Model (LLM), Recipe recommendation, Food recipe.

I. INTRODUCTION

Background

Food waste has become a problematic issue in recent years, according to the Australia Institute; 7.6m tonnes of food each year costing households \$19.3 billion. Some of the largest contributing factors to the amount of food waste in Australia include insufficient shopping and meal planning, overbuying or improper storage. This research paper will look to address the issue of insufficient shopping and meal planning and overbuying by using machine learning to give users recipes based on leftover ingredients so that the ingredients can be used without wastage.

Currently, there are multiple studies done on using image recognition to identify ingredients and recommend based on the recognized ingredient. Rodrigues et al (2023) study of 'Recipe recommendation system based on recognition of food ingredients' details them using a traditional approach to machine learning, creating a dataset with 36 food ingredients classes: separating the data into test, train and validation and training a model to 96% accuracy. Similarly, a different study done by Moral et al (2022)

involved collecting data for image recognition and generating a database in the structure of a 2D matrix with 19 rows containing recipes and columns containing 32 food ingredients with one-to-many relationships. With a database of 9856 images the data was then split into 70% training, 20% for testing and 10% validation and using their model achieved 99.71% accuracy for the training dataset and 92.6% for the validation dataset (Moral et al, 2022). Alternatively, there are a few existing studies that used a language learning model for recommending recipes. Nilesch et al (2019) study on 'Recommendation of Indian Cuisine Recipes based on ingredients' starts with data collection using scrapy 1.5.1 framework for web scraping, this data is then cleaned to only include ingredients and the recipes name. The recommendation system uses bags of word methodology which is a way of rearranging data into a multi-set of words the keywords are then ranked against other recipes in decreasing order of their similarity using a cosine similarity matrix (Nilesch et al, 2019). Therefore, the recommendation system works by taking the user's input, creating a series of similarity scores and outputting the two most similar recipes within the database.

Previous studies have covered utilizing more traditional machine learning methods to recommend recipes based on a single ingredient. However, with the continued development of AI technology it has opened the door to using the vast database of language-based AI models to create a model for this purpose. Therefore, in this project, we fill that missing gap in the field by investigating the creation of a recipe recommendation model that uses AI language models which borrows from a much larger database and uses more user input than previous studies.

Aim of Research

The aim of this project research is to research and develop an app that uses machine learning taking the user's input of ingredients and recommending meal recipes based on the ingredients. This will then be implemented into a food waste web application allowing users to add waste audit and select recent waste items to be part of the recipe recommendation.

Structure of Paper

This thesis project paper is structured to provide a comprehensive account of methodology, implementation and results of the development of a web application recipe recommendation system aimed at mitigating food waste using a large language model (LLM). The approach section will discuss the project's conception and method of research and the methodology for developing the application. The development process is detailed using the

model-driven design framework which also utilizes the Model-View-Controller framework (MVC). Additionally, this section provides insight into the models and programs used such as Django-Haystack and OpenAI. The results section outlines the steps involved in setting up the application, including virtual environment set up, packages install and process to run the application. User interaction elements and application features are also included in this section showing sections of the application and how a user would use the app. The result and analysis section compares the developed application with existing systems to demonstrate its innovative features and how the app will fill a gap in the field. Finally, limitations and conclusions are discussed, focusing on the impact of the application on reducing food waste.

II. APPROACH

Methodology

The project aims to address the most common contributors to food waste in Australia including insufficient shopping and meal planning, overbuying, and improper storage. This will be achieved by developing a web application that utilizes large language models (LLMs) to recommend recipes based on the user's entered leftover ingredients. The research will compare the system's performance of traditional machine learning models, emphasizing the advantages of LLMs in generating a wider and more diverse range of recipes. By implementing LLMs, the project seeks to provide a practical tool for mitigating food waste, empowering users to make better use of leftover ingredients through recipe regenerated recommendations. The project's research was conducted qualitatively due to the development focus of the project. Approach this research project within a 3-month timeframe the project will be split into different phases detailed as follows:

Project conception and research

In preparation for the project a research proposal and research plan was documented detailing the project's aim, scope, approach and important deliverables. The project aim and scope has mostly stayed the same with the key difference in scope of the project which was originally to develop a machine learning based app that recommends recipes based on user's available leftover ingredients and any additional dietary needs, implementing an AI recommendation backend system to match user's input with a suitable recipe. This scope has then been readjusted to cover a wider range to implement the application into a food waste web application that also allows users to add waste audit and select recent waste items to be part of the recipe recommendation.

Collecting Data model

As mentioned the research conducted will be qualitative research that first looks at similar projects to form the approach and methodology of the project. The literature review covers the basis of data gathering for the development of the project. The review consists of

researching thesis or report papers covering the same or similar topics. Wide readings help to develop a deep understanding of the given field and allow the development of what model will be used and possible implementation methods. Additionally, research will need to be conducted to find suitable AI models and pipelining programs that can implement the LLMs into the application. Upon completion of the application this paper will look to compare features of the application to other projects in the same field of study to determine if the application fills the gap in the field.

App development

AI application design is centred around model-driven design since the AI model is the primary focus of the application that processes data and generates responses. Feng et al (2020) states the fundamental behind model-driven design approach is to utilise several linked models to state and create the interface and its components at different phases of the design process, hence using models to drive the entire user interface design process. Instead of tackling the implementation level right away, model-based design seeks to discover high-abstraction level models that enable designers to specify and analyse interactive software systems from a more semantically oriented level. This allows designers to concentrate on more crucial elements, such as the interface's logical design instead of concentrating on implementation methods (Feng et al, 2020).

Django architecture utilises the Model-View-Template (MVT) framework, a variation of the popular MVC model but adapted to fit the Django design philosophy. The MTV model breaks down the application into three components. The MVT pattern includes the model used to store, maintain data and provide data access methods. It is the logical structure behind the whole application and is represented by a database such as MySQL or Postgres. The view is the user interface and is responsible for displaying the data and making user requests to the model. Django template is made up of static parts that describes how dynamic content will be inserted.

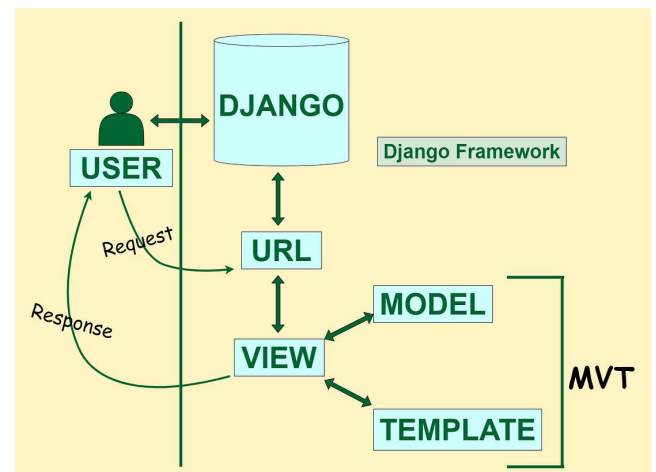


Fig. 1: Model driven design architecture

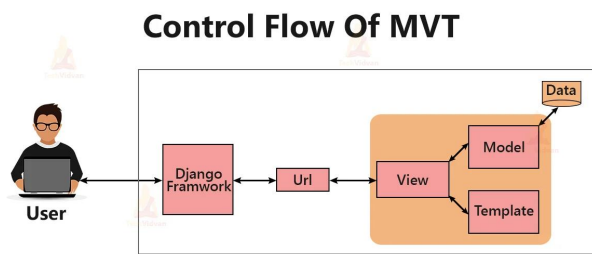


Fig. 2: Control flow of MVT

The above figures show the model hierarchy within the framework. Within the application the model integrates the GPT-4 Open AI model providing the app's main functionality and separates functions and interactive data from the view layer. Haystack was used as the implementation pipeline seeding the Open AI API key and prompt.

The view model handles the user's interaction between the model and the template. Using Python code, this layer defines Django HTTP POST and GET requests, directing the user's input to the right templates and providing dynamic elements to the templates. This is also where the Open AI model is defined and pipelined forming the interaction between the user's data and AI model at this layer.

The template is the static presentation layer of the interface and involves HTML, CSS and Django template language to display dynamic elements like forms to gather user's input. The forms then post the user's entered data to the AI prompt which then reads and configures a response which is then posted back to the views.

Data analysis

At the core of the application, users provide input through a form for an AI generated response. The AI prompt was tested with variable inputs to ensure that a response still fits with the user's input. This is the main functional focus of the application and by qualitatively comparing to other applications in the same field it can be inferred how the application will fit into the field. The application will be compared to other applications listing advantages and disadvantages. Even though there are now many principles and guidelines for application design, Jakob Nielsen '10 usability heuristic' is still considered the standard for UX design. Each application will be analysed to see if each of the heuristics are met.

- Visibility of system status: Keeping users informed about what is going on, through on-screen feedback.
- Match between the system and the real world: Follow real world conventions for design and use concepts and phrases already familiar to the user
- User control and freedom: If user's make a mistake have clear emergency exists
- Consistency and standards: Follows the platform and industry conventions

- Error prevention: check for errors and have alternatives when an error occurs
- Recognition rather than recall: make elements, action and option clear and visible. Users should not have to remember information from one part of the interface to another.
- Flexibility and efficiency of use - include shortcuts to speed up interactions for advanced users.
- Aesthetic and minimalist design: Avoid including information that is rarely needed or diminishes the value of other information
- Help users recognize, diagnose and recover from errors: Error messages should be expressed in plain language not error codes
- Help and documentation: Provide necessary documentation and how to complete their task as needed.

Limitation

The biggest limitation of the project will be time constraint requiring the project to be completed in the period of 10 weeks. This factor will then impact other parts of the project such as limiting the amount of features implemented due to the short time span, and optimization of data pipelining.

The GPT-4 model, although powerful and versatile may not be trained to specific niche dataset resulting in some unwanted outputs. There are also challenges that arise from implementing the OpenAI API as this makes the application dependent on another external system. This means that if there are any outages, latency issues or API limits it will affect the application.

III. RESULTS

Installation and set up

With python installed run the Windows PowerShell command prompt with admin access and run the pip install command: `python -m pip install -U pip`

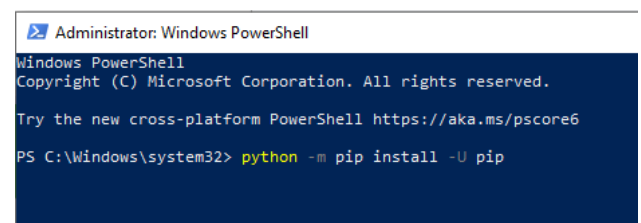


Fig. 3: pip install command

Set up a python virtual environment using command: `pip install virtualenv` and set up a virtual environment for the app with the command: `virtualenv [name of environment]` e.g. `virtualenv env_foodapp`

```
Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> pip install virtualenv
```

Fig. 4: Installing the virtual environment

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Duck> cd ..
PS C:\Users> cd ..
PS C:\> cd .\assessment4_proj\
PS C:\assessment4_proj> virtualenv env_foodapp
```

Fig. 5: Set up the virtual environment

Start the virtual environment command: `.\[name of env]\Scripts\activate.ps1`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Duck> cd ..
PS C:\Users> cd ..
PS C:\> cd .\assessment4_proj\
PS C:\assessment4_proj> .\env_assessment4\Scripts\activate.ps1
(env_assessment4) PS C:\assessment4_proj>
```

Fig. 6: Starting the virtual environment

Multiple packages are required to be installed before running the application including:

- django
- django-haystack
- openai
- whoosh

Therefore, in the environment run command: `pip install [package]`

`pip install Django`

```
(env_assessment4) PS C:\assessment4_proj> pip install django
Collecting django
  Using cached Django-5.1.1-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: asgiref<4,>=3.8.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from django) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from django) (0.5.1)
Requirement already satisfied: tzdata in c:\assessment4_proj\env_assessment4\lib\site-packages (from django) (2024.1)
Using cached Django-5.1.1-py3-none-any.whl (8.2 MB)
Installing collected packages: django
Successfully installed django-5.1.1
```

Fig. 7: Install Django package

`pip install django-haystack`

```
(env_assessment4) PS C:\assessment4_proj> pip install django-haystack
Collecting django-haystack
  Using cached django_haystack-3.3.0-py3-none-any.whl
Requirement already satisfied: Django>=3.2 in c:\assessment4_proj\env_assessment4\lib\site-packages (from django-haystack) (5.1.1)
Requirement already satisfied: packaging in c:\assessment4_proj\env_assessment4\lib\site-packages (from django-haystack) (24.1)
Requirement already satisfied: asgiref<4,>=3.8.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from Django>=3.2->django-haystack) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from Django>=3.2->django-haystack) (0.5.1)
Requirement already satisfied: tzdata in c:\assessment4_proj\env_assessment4\lib\site-packages (from Django>=3.2->django-haystack) (2024.1)
Installing collected packages: django-haystack
Successfully installed django-haystack-3.3.0
```

Fig. 8: Install Django haystack package

`pip install openai==0.28`

```
(env_assessment4) PS C:\assessment4_proj> pip install openai==0.28
Collecting openai==0.28
  Using cached openai-0.28.0-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: requests>=2.20 in c:\assessment4_proj\env_assessment4\lib\site-packages (from openai==0.28) (2.32.0)
Requirement already satisfied: tqdm in c:\assessment4_proj\env_assessment4\lib\site-packages (from openai==0.28) (4.66.5)
Requirement already satisfied: aiohttp in c:\assessment4_proj\env_assessment4\lib\site-packages (from openai==0.28) (3.10.5)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (2024.8.30)
Requirement already satisfied: aiohappyhll<=2.3.0 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (2.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (25.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (1.11.1)
Requirement already satisfied: colorama in c:\assessment4_proj\env_assessment4\lib\site-packages (from tqdm->openai==0.28) (0.4.6)
Using cached openai-0.28.0-py3-none-any.whl (76 kB)
Installing collected packages: openai
Successfully installed openai-0.28.0
```

Fig. 9: Install OpenAI package

`pip install whoosh`

```
(env_assessment4) PS C:\assessment4_proj> pip install whoosh
Collecting whoosh
  Using cached Whoosh-2.7.4-py2.py3-none-any.whl.metadata (3.1 kB)
Using cached Whoosh-2.7.4-py2.py3-none-any.whl (468 kB)
Installing collected packages: whoosh
Successfully installed whoosh-2.7.4
```

Fig. 10: Install whoosh package

After installing the necessary packages, run the Django server using command: `python manage.py runserver`

```
(env_assessment4) PS C:\assessment4_proj> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 06, 2024 - 13:08:40
Django version 5.1.1, using settings 'assessment4_proj.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig. 11: Running the server

The application can then be accessed via <http://localhost:8000/> as long as Windows PowerShell is running.

Features of the food recommender

When the user is not logged in the recipe recommender will appear with an interface that includes a user input form and a get recipe button. The user would use the app by entering an ingredient or a list of ingredients separated by commas and press get recipe. Three recipes will then be displayed that include the entered ingredients with directions to make the meal.

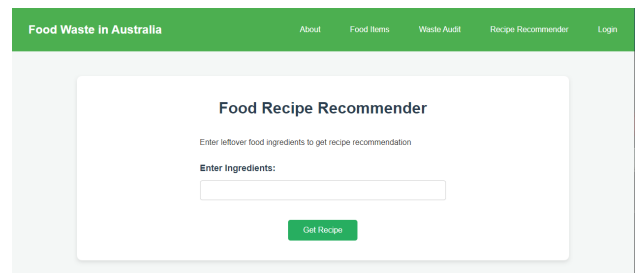


Fig. 12: Basic interface of the recommender

In order to ensure that the application is accessible to a wider group of people, dynamic content has been integrated, that scales with the application size to provide a usable mobile view. This allows the site to look and scale delivering the same experience on different screen sizes.

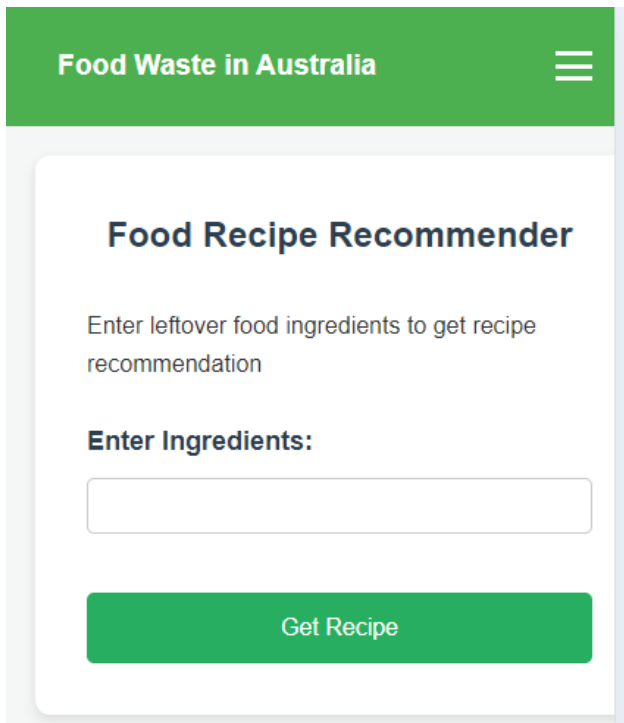


Fig. 13: Mobile view of the recommender

After the user has logged in and creates waste audits to track their food waste. The recent waste items logged into the waste audit will appear as checkbox options that will be selected to be included in the recipe recommendation processing.

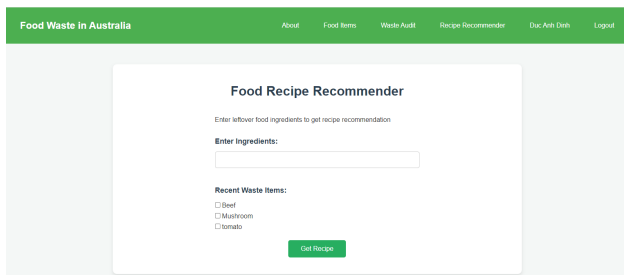


Fig. 14: Logged in view

User input and output examples

This section will include example input and output of the application testing different inputs and analysing the output to see whether the output is a correct response to the input.

Enter Ingredients:

Apple Sugar Flour Butter

Recent Waste Items:

- ☐ Beef
☐ Mushroom
☐ tomato

Get Recipe

Fig. 15: Input example 1

Recommended Recipes:

Apple Crisp

Ingredients:
 - 4-5 apples, peeled and sliced
 - 1 cup sugar
 - 1 cup flour
 - 1/2 cup butter, softened

Directions:
 1. Preheat the oven to 350°F (175°C).
 2. In a bowl, mix the sliced apples with half of the sugar and spread in a baking dish.
 3. In another bowl, mix the remaining sugar, flour, and butter until crumbly.
 4. Sprinkle the crumb mixture over the apples.
 5. Bake for 40-45 minutes or until the topping is golden brown and the apples are soft.
 6. Serve warm with ice cream or whipped cream.

Apple Pie

Ingredients:
 - 6-7 apples, peeled and sliced
 - 1 cup sugar
 - 1 1/2 cups flour
 - 1/2 cup butter, cold and cubed

Directions:
 1. Preheat the oven to 375°F (190°C).
 2. In a bowl, mix the sliced apples with sugar and spread in a pie dish lined with pie crust.
 3. In a separate bowl, mix flour and butter until crumbly.
 4. Sprinkle the crumb mixture over the apples in the pie crust.
 5. Cover the pie with another pie crust or create a lattice pattern with strips of pie crust.
 6. Bake for 45-50 minutes or until the crust is golden brown and the filling is bubbly.
 7. Let it cool before serving.

Apple Cinnamon Muffins

Ingredients:
 - 2 cups flour
 - 1/2 cup sugar
 - 2 tsp baking powder
 - 1/2 tsp salt
 - 1/2 cup butter, melted
 - 1 cup chopped apples
 - 1 tsp cinnamon
 - 1/2 cup milk
 - 2 eggs

Directions:
 1. Preheat the oven to 375°F (190°C) and line a muffin tin with paper liners.
 2. In a bowl, mix flour, sugar, baking powder, and salt.
 3. In another bowl, mix melted butter, milk, and eggs.
 4. Combine the wet and dry ingredients, then fold in the chopped apples and cinnamon.
 5. Fill each muffin cup 2/3 full with the batter.
 6. Bake for 20-25 minutes or until a toothpick inserted in the center comes out clean.
 7

Fig. 16: Output example 1

Enter Ingredients:

Chicken breast, Lettuce, Tomato, Cucumber

Recent Waste Items:

- ☐ Beef
☐ Mushroom
☐ tomato

Get Recipe

Fig. 17: Input example 2

Recommended Recipes:

Chicken Caesar Salad

Ingredients:
 - Chicken breast
 - Lettuce
 - Tomato
 - Cucumber
 - Caesar dressing
 - Parmesan cheese
 - Croutons

Directions:
 1. Grill or cook the chicken breast until fully cooked, then slice into strips.
 2. Wash and chop the lettuce, tomato, and cucumber.
 3. In a large bowl, combine the lettuce, tomato, cucumber, chicken strips, and croutons.
 4. Add Caesar dressing and toss the salad until everything is well coated.
 5. Sprinkle Parmesan cheese on top before serving.

Chicken and Veggie Stir-Fry

Ingredients:
 - Chicken breast
 - Lettuce
 - Tomato
 - Cucumber
 - Soy sauce
 - Garlic
 - Ginger
 - Bell peppers
 - Onion

Directions:
 1. Cut the chicken breast into bite-sized pieces and marinate in a mixture of soy sauce, minced garlic, and grated ginger.
 2. Heat some oil in a pan or wok and stir-fry the chicken until cooked through.
 3. Add chopped bell peppers, onion, tomato, and cucumber to the pan and continue stir-frying until the veggies are tender.
 4. Serve the stir-fry over a bed of lettuce.

Grilled Chicken Pita Pockets

Ingredients:
 - Chicken breast
 - Lettuce
 - Tomato
 - Cucumber
 - Greek yogurt
 - Lemon juice
 - Dill
 - Pita bread

Directions:
 1. Marinate the chicken breast in a mixture of Greek yogurt, lemon juice, and dill for at least 30 minutes.
 2. Grill the chicken until fully cooked and slice into strips.
 3. Cut the lettuce, tomato, and cucumber into small pieces.
 4. Warm the pita bread, then fill each pocket with the grilled chicken, lettuce, tomato, and cucumber.
 5. Drizzle extra Greek yogurt sauce over the filling before serving.

Fig. 18: Output example 2

Using the food items from the recent waste audit items as part of the prompt

Enter Ingredients:

Potato, Cheese

Recent Waste Items:

☒ Beef

☒ Mushroom

☐ tomato

Get Recipe

Fig. 19: Input example 3

Recommended Recipes:

Beef and Mushroom Stuffed Potatoes

Ingredients:

- 4 large potatoes
- 1 lb ground beef
- 1 cup mushrooms, chopped
- 1 cup shredded cheese
- Salt and pepper to taste
- Olive oil

Directions:

1. Preheat oven to 400°F (200°C).
2. Wash and dry the potatoes, then rub them with olive oil and sprinkle with salt. Bake for about 45-50 minutes until tender.
3. In a skillet, cook the ground beef until browned. Add the chopped mushrooms and cook until softened.
4. Cut the tops off the baked potatoes and scoop out the flesh, leaving a thin layer inside the skins.
5. Mix the potato flesh with the beef and mushroom mixture, then stuff the potato skins with the filling.
6. Top each stuffed potato with shredded cheese and bake for another 10-15 minutes until the cheese is melted and bubbly.

Cheesy Beef and Mushroom Skillet

Ingredients:

- 1 lb ground beef
- 1 cup mushrooms, sliced
- 1 onion, diced
- 1 cup shredded cheese
- Salt, pepper, and your favorite seasonings
- Olive oil

Directions:

1. Heat olive oil in a skillet over medium heat. Add the diced onion and cook until translucent.
2. Add the ground beef to the skillet and cook until browned.
3. Stir in the sliced mushrooms and season with salt, pepper, and your favorite seasonings.
4. Once the mushrooms are cooked, sprinkle shredded cheese over the beef and mushroom mixture.
5. Cover the skillet and let the cheese melt. Serve hot and enjoy!

Mushroom and Cheese Stuffed Beef Burgers

Ingredients:

- 1 lb ground beef
- 1 cup mushrooms, finely chopped
- 1 cup shredded cheese
- 1 egg
- Bread crumbs
- Salt, pepper, and seasonings of your choice
- Burger buns and toppings of your choice

Directions:

1. In a bowl, mix together the ground beef, chopped mushrooms, shredded cheese, egg, bread crumbs, salt, pepper, and seasonings.
2. Form the mixture into burger patties.
3. Heat a grill or skillet over medium-high heat. Cook the burger patties for about 4-5 minutes per side, or until cooked to your desired doneness.
4. Assemble your burgers with your favorite toppings and enjoy the delicious mushroom and

Fig. 20: Output example 3

If the input is not food ingredients, then the AI will respond with an error not being able to recommend recipes based on the input.

Enter Ingredients:

Shoe, Fan, Couch, Plant, Table, Shirt

Fig. 21: Input example 4

Recommended Recipes:

I'm sorry, but it seems like the ingredients you provided are not suitable for cooking. Can you please provide a list of food ingredients that you have on hand?

Fig. 22: Output example 4

Food waste audit features

The website is aimed to educate the masses on the topic of food waste and mitigation methods to prevent the amount of food waste in the regular household. To this end, the website includes a food item log and a waste audit log for users to keep track of their food waste and try to mitigate this wastage.

The food item log is used to create common food items that user's use on a weekly basis. Users input a food item and select from a list of food groups and can tick if it is biodegradable. These food items also then appear on a list for the waste audit that the user can choose from.

Add A New Food Item

Food name:

Food groups:

Bio degradable:

Add Submit

Fruit

Vegetable

Grains

Protein

Dairy

Fig. 23: Adding new food items

Food Waste in Australia

AboutFood ItemsWaste AuditRecipe RecommenderDuc Anh DinhLogout

Food Items Log

Create Food items that you would like to use to enter the waste audits

Create new food item

Food Name: Mushroom

Food Group: vegetable

Bio-degradable: True

Add Edit

Food Name: Beef

Food Group: protein

Bio-degradable: True

Add Edit

Food Name: Tomato

Food Group: vegetable

Bio-degradable: True

Add Edit

Food Name: Capsicum

Food Group: vegetable

Bio-degradable: True

Add Edit

Fig. 24: Food item log

The waste audit items are a designed database that users can use to store and track their food wastage. Upon creating a new audit users would select from a list of food items that comes from the food item log, select a waste type and enter an amount. This form is then saved as an item and displayed on the Waste Audit page letting the user view and edit the item as needed.

Add A New Waste Audit

Entry date:

2024-10-08

User:

Duc Anh Dinh

Food item:

Mushroom

Beef

tomato

capsicum

Waste type:

Amount:

Expired Food Waste

Preparation Waste

Uneaten Waste

Add Submit

Fig. 25: Adding a new waste audit item

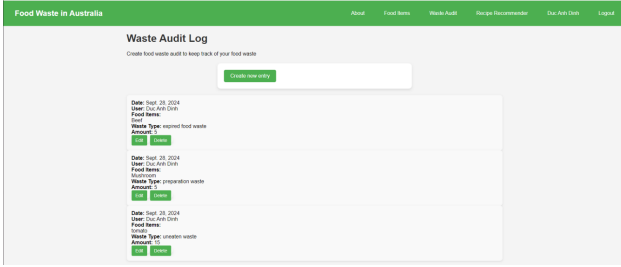


Fig. 26: Waste audit log

Additionally, when a new waste audit that includes a new food item is created the food item will also appear in the recommender for quick access to be included in the recipe recommendation. They appear with checkboxes for the user to select and quickly use those ingredients as part of the recipe recommendation.

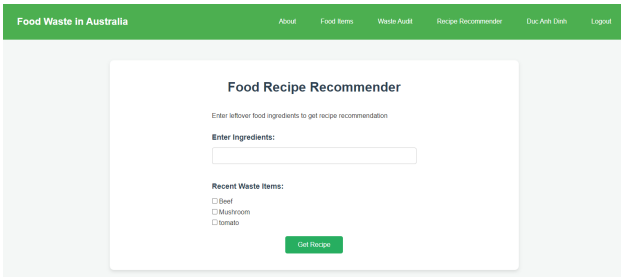


Fig. 27: Recent waste items

Testing and gaining feedback

Testing was done whenever changes were made to the application to ensure everything works smoothly and improves the overall experience. This helps to quickly identify and fix issues like logic errors, and performance problems. After conducting these tests, feedback was continuously sought from supervisors to check if the changes aligned with the goals and scope of the project. The supervisors with an extensive knowledge of data science and large language models (LLMs), provide invaluable insights and direct the program used in the backend of the application. The AI application was also tested using a variety of inputs and error checks for in the case where users enter non ingredient's input.

IV. ANALYSIS AND DISCUSSION OF RESULTS

In this analysis section each application will be rated based on the factors of the 10 Jakob Nielsen Usability Heuristic for UX design.

RecipeIS is an application developed using an open-source database on recipe and food data called Edamam API. This database included thousands of food recipes and food

items also containing images files that were then separated into train, test and validation folders and classified into 36 classes of food ingredients. The ResNet-50 model was used to analyse and classify the ingredients into classes of food ingredients. The model would identify the ingredient and match the recognised ingredient with the vast collection of recipes provided by Edamam API.



Fig. 28: RecipeIS home page upload image

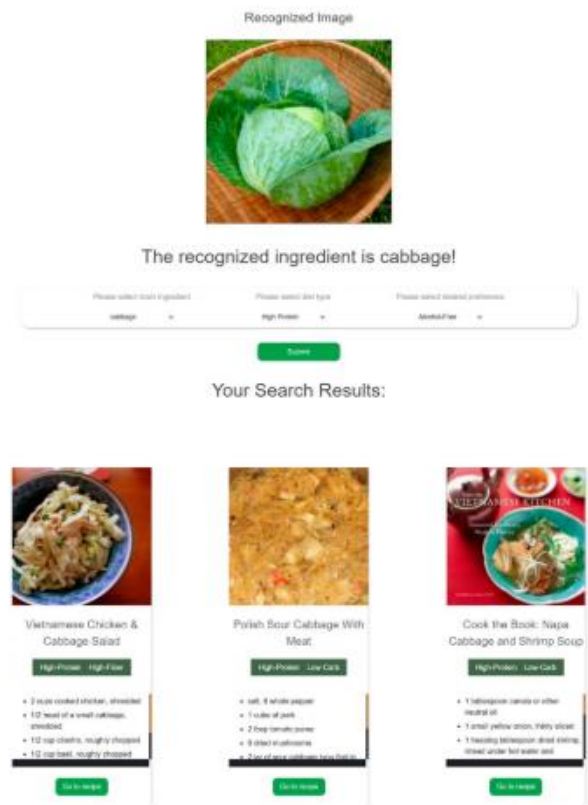


Fig. 29: Recognize the ingredient and use that as main ingredient in the recipe recommender

Table 1: RecipeIS usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input checked="" type="checkbox"/>	The system provides constant feedback, first providing feedback on the image recognition and telling the user the ingredient recognised.
Match between the system and the real world	<input type="checkbox"/>	The website starts on the upload image page but with the colour contrast to the background it makes very clear from first look that this page is used to upload pictures.

User control and freedom	<input type="checkbox"/>	Most of the navigation of the site is found in the navigation bar which is clear. However, after the user uploads an image and the model recognizes there is no clear exits to go back to the page to upload a different image in the case that the user uploads a wrong image.
Consistency and standards	<input type="checkbox"/>	The website's homepage is not what you would expect following the standards of website design. There are also no instructions that leads to the user to process to use the recipe recommender.
Error prevention	<input checked="" type="checkbox"/>	The model was trained and analysed using a correlation matrix obtaining approximately a value of 96% accuracy and a loss rate of about 4%.(Rodrigues, 2023).
Recognition rather than recall	<input checked="" type="checkbox"/>	Important data from each section of the page is displayed and easily recognisable.
Flexibility and efficiency of use	<input type="checkbox"/>	The site is quite simplistic, so the user must follow all the whole process in order.
Aesthetic and minimalist design	<input checked="" type="checkbox"/>	Simple and minimalist design only including key features of the website. The colour contrast works against the website with sections not remaining consistent with their design.
Help users recognize, diagnose, and recover from errors	<input type="checkbox"/>	No error message was not shown when uploading an image, the app doesn't recognise.
Help and documentation	<input checked="" type="checkbox"/>	Section in the thesis paper showing example usage of the recipe recommendation system.

ChatDiet is an LLM-powered chatbot that integrates personalized data for personalised nutrition-oriented food recommender. The framework includes an Orchestrator, which pulls relevant data from personal and population models in response to user inquiries. This information is then combined with the LLM's internal knowledge to provide personalized nutrition recommendations (Yang, 2024)

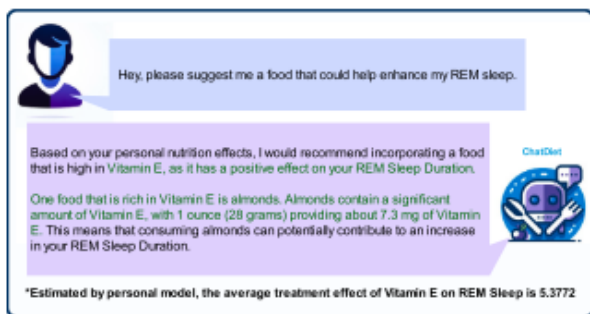


Fig. 30: Food recommendation and explanation based on nutrition effect from ChatDiet

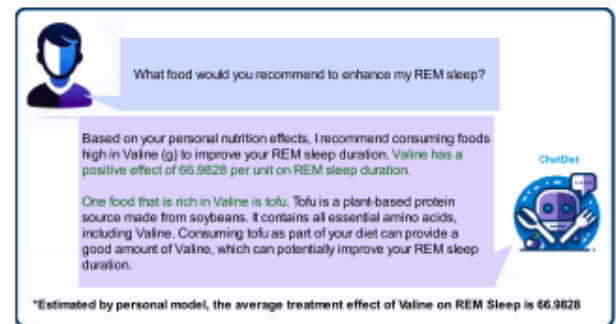


Fig. 31: Personalized feedback from ChatDiet

Table 2: ChatDiet usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input checked="" type="checkbox"/>	The chatbot provides a human like conversation always giving feedback to the user regarding their inquiries.
Match between the system and the real world	<input checked="" type="checkbox"/>	As chatbot assistant are similar provide a similar experience to conversation and is easy for anyone to use.
User control and freedom	<input type="checkbox"/>	Limited user freedom as when a prompt has been entered the chatbot will response and the user doesn't have a way to prevent that data sent.
Consistency and standards	<input checked="" type="checkbox"/>	As chatbot assistant have become popularized the app has a good match between system and real world having a design like other application in the industry.
Error prevention	<input checked="" type="checkbox"/>	The model is effective being tested with 400 queries and the response analysis for each section with no less than 85% accuracy in each section.
Recognition rather than recall	<input checked="" type="checkbox"/>	Previous queries are still displayed when the user enters another query so it can be looked back on.
Flexibility and efficiency of use	<input type="checkbox"/>	Simplistic AI chatbot design with no shortcuts or flexibility of usage.
Aesthetic and minimalist design	<input type="checkbox"/>	The interface is simple and easy to use, however does not contain much information on the usage of the chatbot before the user uses it.

Help users recognize, diagnose, and recover from errors	<input type="checkbox"/>	Unable to be tested.
Help and documentation	<input type="checkbox"/>	The documentation shows a case study inputs and output, but they are quite specific and other example inputs are not explained.

SuperCook is an AI-powered cooking app that enhances the kitchen experience through customized recipe suggestions, intuitive ingredient selection, and interactive meal planning. It provides personalized recommendations based on user preferences, profiles, and available ingredients.

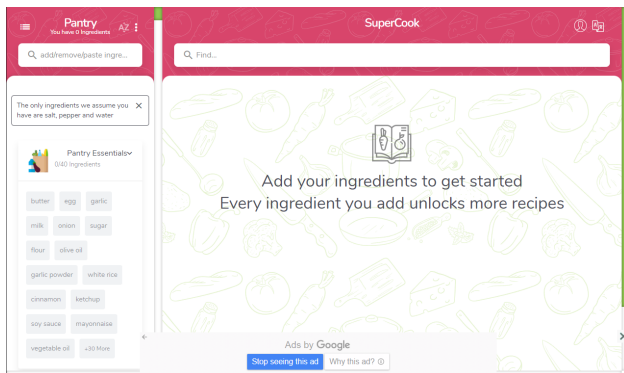


Fig. 32: Supercook web application

The Supercook application works by the user selecting ingredients in the pantry list and the recipe recommender will suggest recipes based on the available ingredients selected. The suggested recipe appears in a database that the user and filter and search through. Recipes will not appear if the recipe has ingredients not selected in the pantry list.

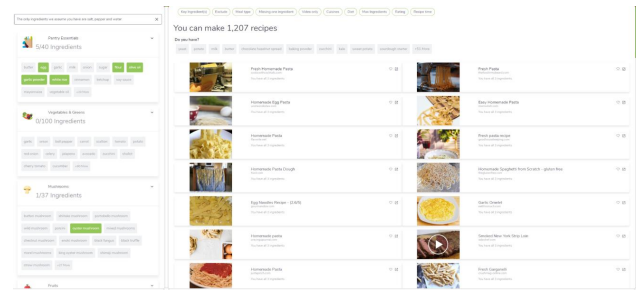


Fig. 33: List of suggested recipes

Table 3: Supercook usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input checked="" type="checkbox"/>	The application is responsive whenever pantry ingredients are selected providing clear feedback of recipes to the user.
Match between the system and the real world	<input checked="" type="checkbox"/>	The system takes the approach of giving the user a database storage in the form of the pantry like the real world.
User control and freedom	<input checked="" type="checkbox"/>	The user selects from a list of ingredients and can easily deselect the ingredient if they accidentally select the wrong ingredient.
Consistency and standards	<input type="checkbox"/>	The app layout and features functionality does not follow the standards for an AI application which involve user's queries input.
Error prevention	<input checked="" type="checkbox"/>	Supercook has each ingredient as selector limiting the user input but also preventing user error.
Recognition rather than recall	<input type="checkbox"/>	The pantry list is long and for first time user's it would be hard to select all the ingredients and remember the ingredients selected in the pantry.
Flexibility and efficiency of use	<input type="checkbox"/>	The web application doesn't have shortcuts, and the user must select the ingredients one by one.
Aesthetic and minimalist design	<input checked="" type="checkbox"/>	The design is simple and effective with the pantry list taking the user's input and the recipes will be displayed and can be searched for.
Help users recognize, diagnose, and recover from errors	<input checked="" type="checkbox"/>	Error check prevention in place and the application handles user input very well.
Help and documentation	<input checked="" type="checkbox"/>	The research paper explains the application well with descriptions on app usage and performance.

Table 4: Food waste recipe recommender usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input type="checkbox"/>	The application is sometimes unclear with user feedback when the AI is loading a response. With longer responses if the user does not notice the loading sign on the page.
Match between the system and the real world	<input checked="" type="checkbox"/>	The basis of the project was built around users' leftover ingredients and allowing them to enter those ingredients to recommender recipes so that they can make use of those ingredients
User control and freedom	<input type="checkbox"/>	Limited user freedom as when a prompt has been entered the chatbot will response and the user doesn't have a way to prevent that data sent
Consistency and standards	<input checked="" type="checkbox"/>	The website fits the consistency and standards of the industry using cards

		to display important information with multiple navigation to the same places
Error prevention	✓	The GPT 4 model is effective at recognising invalid input and will response with the appropriate error message if the user doesn't enter ingredients
Recognition rather than recall	✓	When the user uses other functionality of the site each item are interconnected and displayed when using another feature
Flexibility and efficiency of use	✓	Waste audit items are shortcut to a checklist that the logged in user can use making it easier for users that have used the app multiple times
Aesthetic and minimalist design	✓	The application has a clean and distinctive design with each web page complementing each other
Help users recognize, diagnose, and recover from errors	✓	The GPT 4 model is effective at recognising invalid input and will response with the appropriate error message and notice if the user doesn't enter ingredients
Help and documentation	✓	Documentation shows how the users would use the app and also shows potential errors that could occur with the GPT model

Compared to other recipe recommenders in the industry the food waste recipe recommender performed better overall in terms of Jakob Nielsen's Usability heuristic. The application was designed with the UX design in mind to provide users with the key recipe recommendation feature of the application. RecipeIS uses the traditional approach to machine learning borrowing the dataset from Edamam API that has a variety of approved recipes allowing for accurate and precise recipe recommendations that aligns with the image recognized food ingredient. However, this more traditional approach limits the effectiveness of the recommender to the size of the available database compared to using the LLM model. ChatDiet functionality works quite differently to the developed recipe recommender, but the app uses a similar LLM model. This model is prompted to use a different set of data to provide a different functionality. The app takes the user's personalized feedback to be added into the AI model which allows the prompt to better generate personalized responses to the user's queries. However, the recognised queries are quite limited and niche to specific health recommendations potentially making it harder for new users to use. Lastly, Supercook is the most similar application, using AI to recommend recipes based on selected ingredients. The application is well developed including many useful features including the pantry list, filter and search function. The recipe result shows a wide variety of options that the user can choose from that include the selected ingredient, but this can also play to the detriment of the app where it takes new users a long time to select all the required ingredients. Recipes also require all ingredients included in that recipe to be selected and will not be displayed otherwise. Additionally, when selecting a wide array of food ingredients, the list of recipes is quite vast making it hard for user's to find and decide on the recipe they want to cook. Supercook is also aimed at a different purpose to the developed recipe recommender aiming to enhance the kitchen and cooking experience through these personalized recommendations while the developed recipe recommender is aimed to reduce the amount of food waste adopting less features to not overwhelm user's the number of options.

V. CONCLUSIONS

This research project focuses on development of a recipe recommendation app utilising large language models (LLMs). Research for LLMs recipe recommender are undeveloped so more focused was put on traditional methods and LLM usage in other fields to find potential models and methodology that could later be implemented. Many existing recipe recommender studies leveraged image recognition and traditional machine learning techniques to recommend recipes based on single ingredients. This project not only aims to develop an advanced recommendation system but also integrates it into a broader food waste web application that enables users to track food waste and utilize leftover ingredients effectively. The methodology, development, and results outlined in this paper emphasize the project's contribution to mitigating food waste through the intelligent application of AI technology. This system builds upon previous research while filling a gap in the field by offering more dynamic and comprehensive user input and leveraging a larger database of recipe possibilities.

VI. FUTURE WORK

Due to a few limitations with the development of the app as discussed above there are a few future work and improvements that can be considered. Firstly, since the application uses the GPT 4 model, while diverse and functional cannot provide personalized recommendation. Therefore, future work can try to implement other LLMs models to better personalize the AI recipe recommendation responses. Another consideration include implementation of an exclude list or filter option where the user can select ingredients to not include or select dietary restrictions that they follow or have, and the AI model will take that into consideration when giving recipe recommendations. Finally, a feature for the LLMs is to have a feedback system for the AI for the user to rate recipes responses that the AI recommends and save this personalized data. The data would help the LLMs make better personalized recommendations for the user based on previous liked recipes.

APPENDIX A: LITERATURE REVIEW

Introduction

Food waste has become a problematic issue in recent years, according to the Australia Institute; 7.6m tonnes of food each year costing households \$19.3 billion. Some of the largest contributing factors to the amount of food waste in Australia include insufficient shopping and meal planning, overbuying or improper storage. These factors are exacerbated by modern consumer habits, where people's busy lifestyles and lack of awareness of food management lead to excessive waste. Indecisiveness or lack of recipe knowledge is a big contributing factor to insufficient shopping and meal planning or overbuying problems among the masses. When individuals lack recipe knowledge or are unsure about what meals to prepare or how to use ingredients they already have, they are more likely to purchase unnecessary items or let food go unused.

Recommendation systems have been ubiquitous employed, especially in areas like e-commerce, streaming services to help users make informed decisions from the overwhelming number of options. These systems use data-driven algorithms to suggest options based on user preferences and past behaviours. However, there has not been much research and employment of recipe recommendations using Large Language Models (LLMs), which could be used to greatly expand the field. Currently, recipe recommendation studies have been mostly limited to traditional machine model building which has involved data collection or web scraping to build a specific dataset. This dataset is then trained by splitting the data into training and testing sets and run using a machine learning model. Although these traditional models are effective to some extent, they have limitations to the amount of data available in the trained dataset and are not capable of adapting to the user's needs. Unlike traditional recommendation systems, which rely on structured data and predefined rules, LLMs can process and understand natural language, making them capable of providing more context-aware and adaptive suggestions. This capability opens new possibilities for enhancing recipe recommendation systems, making them more responsive to individual user needs and preferences. By leveraging LLM's expansive database we can enhance these systems through advanced techniques such as zero-shot sequential recommendation, generative recommendation, and text-based collaborative filtering (Wang, 2024).

In this literature review, we will discuss studies of traditional methods of machine learning for recipe recommendation and LLMs recommendations systems current usage and studies. This will lay the foundation for research to address the issue of insufficient shopping and meal planning and overbuying by using machine learning large language model (LLMs) to recommend users recipes based on leftover ingredients so that purchased ingredients can be used without wastage. Studies show that personalized recommendations can encourage healthier food choices, increase dietary adherence, and improve overall user satisfaction [12]. This intersection of personal well-being and culinary delight underscores the profound societal and individual benefits of effective food recommendation systems.

Traditional methods of machine learning recipe recommendations

Research for recipe recommendations can hardly be seen but there are few existing studies that used a language learning model for recommending recipes in this niche field. The research on recipe recommendations is relatively small mainly due to their development largely relying on research of technology such as machine learning, deep learning or image recognition (Liang, Y et al, 2024). In this field, published studies have often utilized traditional machine learning techniques to employ recipe recommendations systems based on user inputs or ingredient recognition. These studies feature various approaches to data collection, data storing, model training, and content-based filtering but build a more limited database compared to large language models. The project by Rodrigues (2023) on the RECIPEIS recommendation system and by Gim on RecipeBowl and Moral study on 'Food recipe recommendation-based ingredient detection' are notable examples.

According to Liang, Y et al (2024) most applications that involve food recognition require a very accurate recognition of the food ingredient photographed by the user so algorithms like convolutional neural networks (CNN) are often used. All the above listed systems employ CNNs to first recognize ingredients through image recognition and then recommend recipes based on these recognized ingredients (Rodrigues, 2023; Gim, 20; Moral,). Rodrigues's (2023) started by utilizing an open-source database on recipe and food data called Edamam API, this dataset consists of image files that was then separated into train, test and validation folders and classified into 36 classes of food ingredients. ResNet-50 model structure was used to create the model structure which achieved an accuracy of 96%. Similarly, a study done by Moral et al (2022) involved collecting

data for image recognition and generating a database in the structure of a 2D matrix with 19 rows containing recipes and columns containing 32 food ingredients with one-to-many relationships. With a database of 9856 images, the data was then split into 70% training, 20% for testing and 10% validation and using their model achieved 99.71% accuracy for the training dataset and 92.6% for the validation dataset (Moral et al, 2022). However, Moral (2022) study recognizes multiple ingredients input and returns multiple suggestion recipes that include those recognized ingredients which improves upon Rodrigues's web application which only takes one user input. Additionally, Moral's study did not use the commonly used ResNet50 model since this pre-trained model does not classify images into specific object categories and had built their own model from the ResNet50 structure. ResNet50 early layer's can still be used as the early layers are used to identify shapes and visual patterns but the last layer which makes predictions was changed.

Gim's (2021) RecipeBowl takes this a step further by employing a set transformer model, which encodes the ingredient set into a context representation, enabling more sophisticated recommendations. This enables RecipeBowl not only to predict recipes based on the ingredients but also suggests additional ingredients that might complement the existing set. RecipeBowl uses an extended version of Receptor which includes a dataset of 507,834 recipes which has been split into 80% including 373,760 training recipes, 10% including 47104 validation and 10%, 47104 test recipes. Nilesh et al (2019) study on 'Recommendation of Indian Cuisine Recipes based on ingredients' starts with data collection using scrapy 1.5.1 framework for web scraping, this data is then cleaned to only include ingredients and the recipes name. The recommendation system uses bags of word methodology which is a way of rearranging data into a multi-set of words the keywords are then ranked against other recipes in decreasing order of their similarity using a cosine similarity matrix (Nilesh et al, 2019). Therefore, the recommendation system works by taking the user's input, creating a series of similarity scores and outputting the 2 most similar recipes within the database.

These traditional methods, while effective, have limitations. Content-based filtering can miss the culinary context and the potential for ingredient substitutions, leading to less flexible recommendations. Pre-trained CNN models like ResNet-50 require substantial data and fine-tuning to adapt to specific culinary domains, and while they are powerful in recognizing ingredients, their performance in recipe recommendation is heavily dependent on the quality and structure of the underlying recipe database. Traditional machine learning approaches to recipe

recommendation have laid the groundwork for more advanced systems. They demonstrate the effectiveness of content-based filtering and transfer learning but also highlight the challenges of adapting general-purpose models to the specific and varied domain.

LLM - generated response current usage and studies

Large Language learning models (LLM) are models trained with an immense amount of data making them capable of understanding and predicting relationships of language text. They are increasingly being integrated into various applications, from customer service chatbots to advanced research tools, due to their ability to generate human-like text. In the context of recipe recommendation, LLMs techniques such as zero-shot sequential recommendation, generative recommendation and text based collaborative filtering have opened a new avenue in the industry. (Wang et al, 2024). The GPT model, the backend behind ChatGPT has emerged as the most popular AI model, trained with a wide variety of data opening various capabilities for user queries and interactions. GPT model has also become the most common model utilized in research studies with a wide variety of LLMs studies such as enhancing recommender system, nutritional recommendations for dialysis patients and personalized food-based recommendation all utilizing ChatGPT AI model.

Roumeliotis et al (2024) and Lubos et al (2024) studies both utilised the ChatGPT model to investigate and understand certain functions of LLMs model that can be further developed in the field. Roumeliotis et al (2024) research has found that the most effective unsupervised LLMs recommender model evaluated by GPT model is content-based filtering which achieved 72.93% accuracy, correctly recommending 1094 out of 1500 products. K-means clustering-trained model performs very closely behind with an accuracy of 72.33% equalling 1085 correct recommendation out of 1500 (Roumeliotis et al, 2024). These two models stand out from the hierarchical clustering-trained model that has a much lower prediction rate at 63.07% (Roumeliotis et al, 2024).

In Lubos' et al (2024) study on "LLM generated explanations for recommender systems", they seek to improve the overall user experience with recommender systems by providing explanations and why certain items were suggested. Results show that in some domains ChatGPT inherently can perform as a recommendation system that's on par with other models only needing minimal input, such as short descriptions and directions to provide relevant recommendations (Lubos et al, 2024). This capability allows the model to act as an effective initial recommendation tool allowing the

model to be useful when user data is limited. Furthermore, tested users expressed an overall positive for the recommender explanations improving the transparency and perceptions of the recommender system (Lubos et al, 2024).

There have also been studies conducted by Rostami (2024) and Wang et al (2024) that hopes to implement the broad applicability of LLMs to food recommendation domains. Rostami (2024) states there are several challenges to this implementation as we seek to mitigate inherent flaws like logical inaccuracies and hallucinations. To this challenge, recent advancements have seen food recommendation systems integrate a hybrid of content-based filtering and collaborative filtering leveraging the strengths of both methods to improve recommendation accuracy and downside (Rostami, 2024). This approach will also enhance personalization and contextual awareness of the model by combining user preferences with LLMs understanding of linguistics, description and context (Rostami, 2024). Similarly, Wang et al (2024) has integrated Large Language Models (LLMs) like ChatGPT for personalized nutritional guidance in dialysis patients. The study utilizes GPT-4 to generate diet plans for virtual patients of varying demographics and clinical characteristics revealed that while the model can produce culturally tailored menus and reliable translations, it falls short in accurate nutritional analysis. Discrepancies, particularly in estimations of protein, calories, and key electrolytes, highlight the need for cautious application in clinical settings. Despite these limitations, the study suggests that with further refinement, LLMs could become valuable tools in supporting renal dietitians by providing diverse and personalized dietary recommendations.

Yang et al (2024) furthers personalized dietary recommendations with their nutrition-oriented food recommender chatbot, ChatDiet. The ChatDiet framework includes 4 components including personal model, population model, orchestrator and LLM. The personal and population model are user input models with the personal model gathering user's food log data and other health information from a wearable ring (Yang et al, 2024). The population model gathers data from a food nutrition list created by Cronometer Food Logger (Yang et al, 2024). The orchestrator interacts with the personal and population model to extract relevant information based on the user's queries (Yang et al, 2024). This data is then sent to the LLM which will make personalized nutrition-oriented food recommendation for the user to incorporate into their diet (Yang et al, 2024). The capacity for ChatDiet to demonstrate the reasoning and decision-making process

behind recommendation makes the chatbot an efficient recipe recommender for health outcome interest Yang et al (2024).

Peng .Y (2024) study on SuperCook is an intuitive design project of a recipe recommender developed using Figma to foster the concept of using AI-driven models to generate recipes to cater to individual culinary tastes and needs. The development of SuperCook resulted in a multi featured mobile application that integrates artificial intelligence (AI) to provide a personalized and innovative culinary experience for the user (Peng. Y, 2024). The design was rooted by the user-centred approach emphasizes the user experience (UX) and user interface (UI) design. The final product includes a wide range of features including a recipe gallery, ingredient selection, cuisine exploration and flavour adjustment. These features and benefits have potential to revolutionize the industry being a useful tool for anyone enhance their culinary experience.

Conclusion

The literature review has noted many related studies pertaining to the research topic, comparing methodology and results of the different studies. The review first started with articles that developed recipe recommender using traditional methods using techniques like machine learning, content-based filtering, and image recognition. Their reliance on structured data and predefined algorithms restricted their ability to offer personalized, context-aware recommendations. They also demonstrated the slow advancement of AI integration in recipe recommendations. The review then explored the integration of Large Language Models (LLMs) into recipe recommendation systems, highlighting the potential of LLMs leveraging vast amounts of unstructured data and are capable of processing natural language inputs, enabling more dynamic, flexible, and personalized recipe recommendations. The potential of LLMs in advancing recipe recommendation systems is evident, offering new opportunities to address the issues of insufficient shopping, meal planning, and food waste. By harnessing the power of LLMs, future research can focus on creating systems that are more adaptable to user needs, capable of suggesting recipes based on leftover ingredients, and minimizing food waste in households. As research in this area evolves, LLMs could play a pivotal role in transforming how individuals plan meals, manage food leading users to manage their purchase ingredients effectively reducing the amount of food that goes to waste.

APPENDIX B: DATA AND PROGRAMS USED FOR THIS REPORT

Programs used

- IDE/Text editor such as Visual Code studio
- Python software
- Haystack
- Django
- Django-haystack
- Whoosh
- OpenAI GPT 4

APPENDIX C: FIGURES

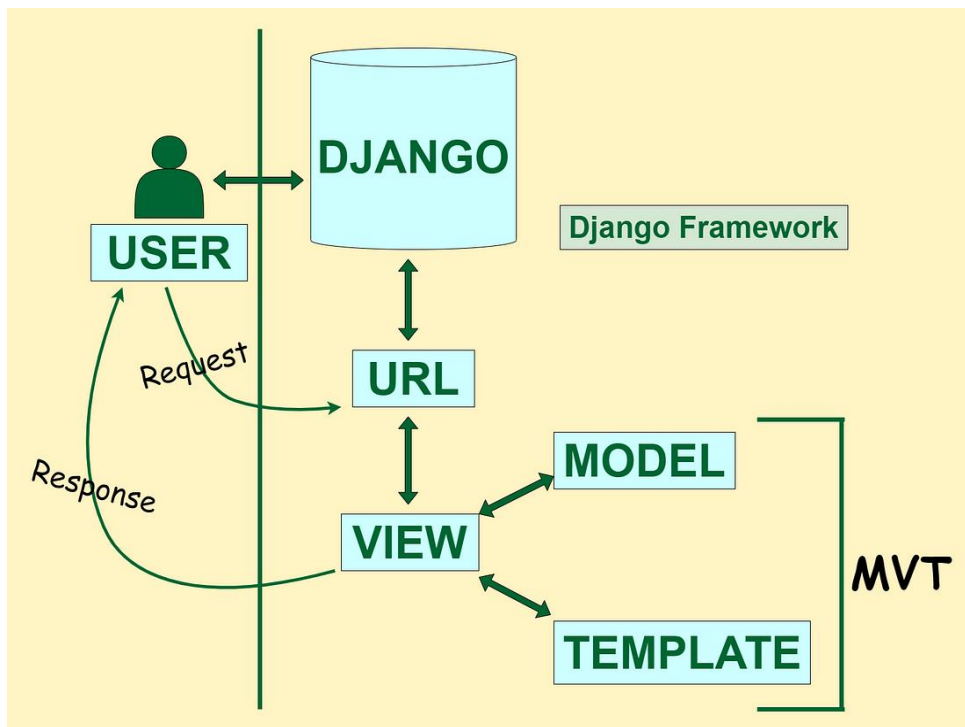


Fig. 34: Model driven design architecture

Control Flow Of MVT

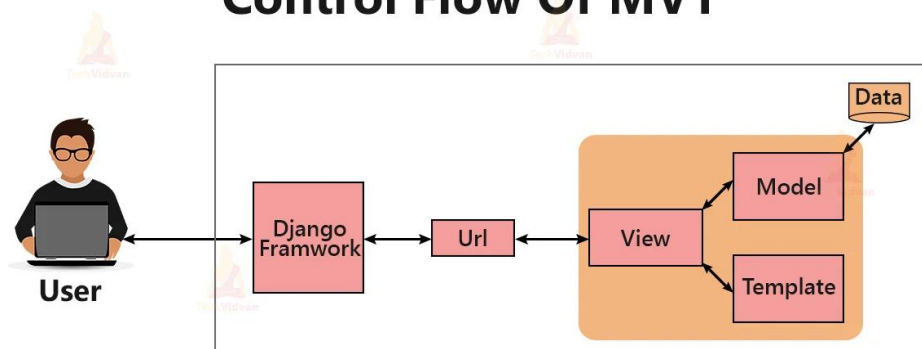


Fig. 35: Control Flow of MVT

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> python -m pip install -U pip
```

Fig. 36: pip install command

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> pip install virtualenv
```

Fig. 37: Installing the virtual environment

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Duck> cd ..
PS C:\Users> cd ..
PS C:\> cd .\assessment4_proj\
PS C:\assessment4_proj> virtualenv env_foodapp
```

Fig. 38: Set up the virtual environment

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Duck> cd ..
PS C:\Users> cd ..
PS C:\> cd .\assessment4_proj\
PS C:\assessment4_proj> .\env_assessment4\Scripts\activate.ps1
(env_assessment4) PS C:\assessment4_proj>
```

Fig. 39: starting the virtual environment

```
(env_assessment4) PS C:\assessment4_proj> pip install django
Collecting django
  Using cached Django-5.1.1-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: asgiref<4,>=3.8.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from django) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from django) (0.5.1)
Requirement already satisfied: tzdata in c:\assessment4_proj\env_assessment4\lib\site-packages (from django) (2024.1)
Using cached Django-5.1.1-py3-none-any.whl (8.2 MB)
Installing collected packages: django
Successfully installed django-5.1.1
```

Fig. 40: Install Django package

```
(env_assessment4) PS C:\assessment4_proj> pip install django-haystack
Collecting django-haystack
  Using cached django_haystack-3.3.0-py3-none-any.whl
Requirement already satisfied: Django>=3.2 in c:\assessment4_proj\env_assessment4\lib\site-packages (from django-haystack) (5.1.1)
Requirement already satisfied: packaging in c:\assessment4_proj\env_assessment4\lib\site-packages (from django-haystack) (24.1)
Requirement already satisfied: asgiref<4,>=3.8.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from Django>=3.2->django-haystack) (3.8.1)
Requirement already satisfied: sqlparse>=0.3.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from Django>=3.2->django-haystack) (0.5.1)
Requirement already satisfied: tzdata in c:\assessment4_proj\env_assessment4\lib\site-packages (from Django>=3.2->django-haystack) (2024.1)
Installing collected packages: django-haystack
Successfully installed django-haystack-3.3.0
```

Fig. 41: install Django Haystack package

```
(env_assessment4) PS C:\assessment4_proj> pip install openai==0.28
Collecting openai==0.28
  Using cached openai-0.28.0-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: requests>=2.20 in c:\assessment4_proj\env_assessment4\lib\site-packages (from openai==0.28) (2.32.3)
Requirement already satisfied: tqdm in c:\assessment4_proj\env_assessment4\lib\site-packages (from openai==0.28) (4.66.5)
Requirement already satisfied: aiohttp in c:\assessment4_proj\env_assessment4\lib\site-packages (from openai==0.28) (3.10.5)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\assessment4_proj\env_assessment4\lib\site-packages (from requests>=2.20->openai==0.28) (2024.8.30)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (2.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\assessment4_proj\env_assessment4\lib\site-packages (from aiohttp->openai==0.28) (1.11.1)
Requirement already satisfied: colorama in c:\assessment4_proj\env_assessment4\lib\site-packages (from tqdm->openai==0.28) (0.4.6)
Using cached openai-0.28.0-py3-none-any.whl (76 kB)
Installing collected packages: openai
Successfully installed openai-0.28.0
```

Fig. 42: Install OpenAI package

```
(env_assessment4) PS C:\assessment4_proj> pip install whoosh
Collecting whoosh
  Using cached Whoosh-2.7.4-py2.py3-none-any.whl.metadata (3.1 kB)
Using cached Whoosh-2.7.4-py2.py3-none-any.whl (468 kB)
Installing collected packages: whoosh
Successfully installed whoosh-2.7.4
```

Fig. 43: Install Whoosh package

```
(env_assessment4) PS C:\assessment4_proj> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 06, 2024 - 13:08:40
Django version 5.1.1, using settings 'assessment4_proj.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig. 44: Running the server

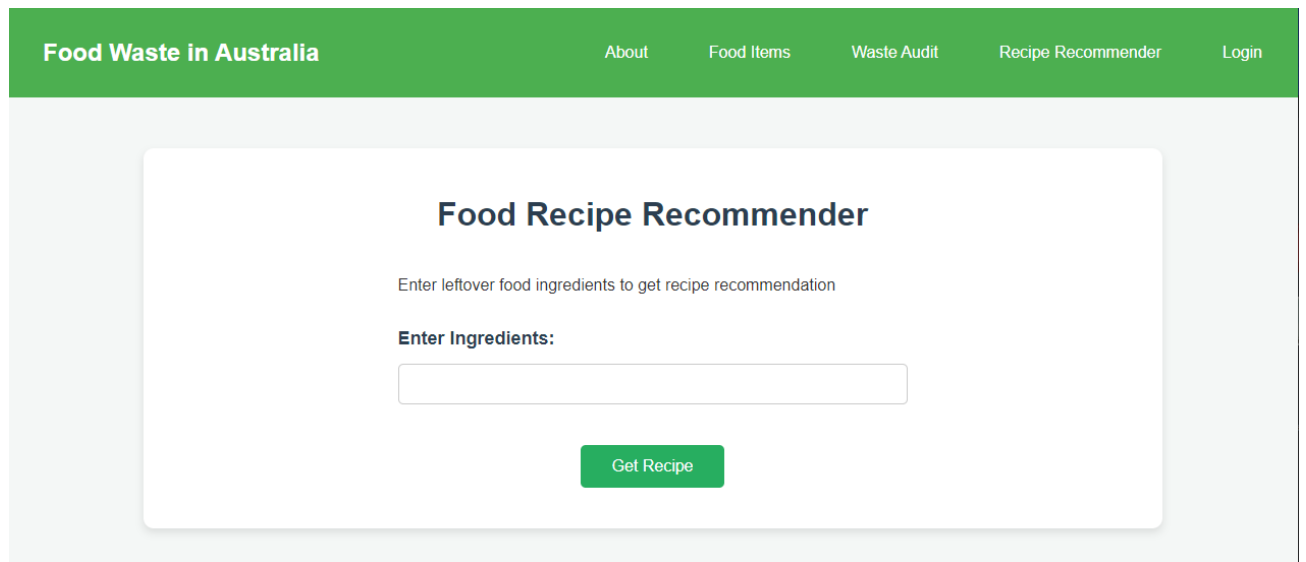


Fig. 45: Basic interface of the recommender

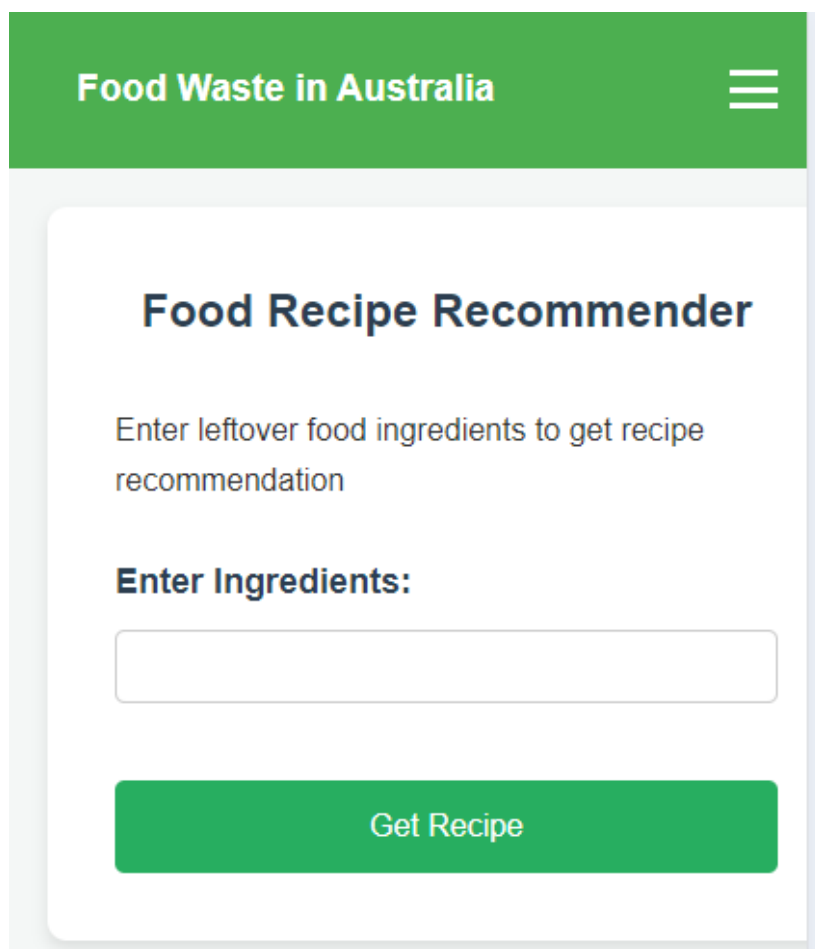


Fig. 46: Mobile view of the recommender

Food Waste in Australia

About

Food Items

Waste Audit

Recipe Recommender

Duc Anh Dinh

Logout

Food Recipe Recommender

Enter leftover food ingredients to get recipe recommendation

Enter Ingredients:

Recent Waste Items:

☐ Beef

☐ Mushroom

☐ tomato

Get Recipe

Fig. 47: Logged in view

--- Continue on next page ---

Enter Ingredients:

Apple Sugar Flour Butter

Recent Waste Items:

- ☐ Beef
- ☐ Mushroom
- ☐ tomato

Get Recipe

Fig. 48: Input example 1

Recommended Recipes:

<p>Apple Crisp</p> <p>Ingredients:</p> <ul style="list-style-type: none">- 4-5 apples, peeled and sliced- 1 cup sugar- 1 cup flour- 1/2 cup butter, softened <p>Directions:</p> <ol style="list-style-type: none">1. Preheat the oven to 350°F (175°C).2. In a bowl, mix the sliced apples with half of the sugar and spread in a baking dish.3. In another bowl, mix the remaining sugar, flour, and butter until crumbly.4. Sprinkle the crumb mixture over the apples.5. Bake for 40-45 minutes or until the topping is golden brown and the apples are soft.6. Serve warm with ice cream or whipped cream.	<p>Apple Pie</p> <p>Ingredients:</p> <ul style="list-style-type: none">- 6-7 apples, peeled and sliced- 1 cup sugar- 1 1/2 cups flour- 1/2 cup butter, cold and cubed <p>Directions:</p> <ol style="list-style-type: none">1. Preheat the oven to 375°F (190°C).2. In a bowl, mix the sliced apples with sugar and spread in a pie dish lined with pie crust.3. In a separate bowl, mix flour and butter until crumbly.4. Sprinkle the crumb mixture over the apples in the pie crust.5. Cover the pie with another pie crust or create a lattice pattern with strips of pie crust.6. Bake for 45-50 minutes or until the crust is golden brown and the filling is bubbly.7. Let it cool before serving.	<p>Apple Cinnamon Muffins</p> <p>Ingredients:</p> <ul style="list-style-type: none">- 2 cups flour- 1/2 cup sugar- 2 tsp baking powder- 1/2 tsp salt- 1/2 cup butter, melted- 1 cup chopped apples- 1 tsp cinnamon- 1/2 cup milk- 2 eggs <p>Directions:</p> <ol style="list-style-type: none">1. Preheat the oven to 375°F (190°C) and line a muffin tin with paper liners.2. In a bowl, mix flour, sugar, baking powder, and salt.3. In another bowl, mix melted butter, milk, and eggs.4. Combine the wet and dry ingredients, then fold in the chopped apples and cinnamon.5. Fill each muffin cup 2/3 full with the batter.6. Bake for 20-25 minutes or until a toothpick inserted in the center comes out clean. <p>7</p>
--	---	---

Fig. 49: Output example 1

Enter Ingredients:

Chicken breast, Lettuce, Tomato, Cucumber

Recent Waste Items:

- ☐ Beef
- ☐ Mushroom
- ☐ tomato

Get Recipe

Fig. 50: Input example 2

Recommended Recipes:

<p>Chicken Caesar Salad</p> <p>Ingredients:</p> <ul style="list-style-type: none">- Chicken breast- Lettuce- Tomato- Cucumber- Caesar dressing- Parmesan cheese- Croutons <p>Directions:</p> <ol style="list-style-type: none">1. Grill or cook the chicken breast until fully cooked, then slice into strips.2. Wash and chop the lettuce, tomato, and cucumber.3. In a large bowl, combine the lettuce, tomato, cucumber, chicken strips, and croutons.4. Add Caesar dressing and toss the salad until everything is well coated.5. Sprinkle Parmesan cheese on top before serving.	<p>Chicken and Veggie Stir-Fry</p> <p>Ingredients:</p> <ul style="list-style-type: none">- Chicken breast- Lettuce- Tomato- Cucumber- Soy sauce- Garlic- Ginger- Bell peppers- Onion <p>Directions:</p> <ol style="list-style-type: none">1. Cut the chicken breast into bite-sized pieces and marinate in a mixture of soy sauce, minced garlic, and grated ginger.2. Heat some oil in a pan or wok and stir-fry the chicken until cooked through.3. Add chopped bell peppers, onion, tomato, and cucumber to the pan and continue stir-frying until the veggies are tender.4. Serve the stir-fry over a bed of lettuce.	<p>Grilled Chicken Pita Pockets</p> <p>Ingredients:</p> <ul style="list-style-type: none">- Chicken breast- Lettuce- Tomato- Cucumber- Greek yogurt- Lemon juice- Dill- Pita bread <p>Directions:</p> <ol style="list-style-type: none">1. Marinate the chicken breast in a mixture of Greek yogurt, lemon juice, and dill for at least 30 minutes.2. Grill the chicken until fully cooked and slice into strips.3. Cut the lettuce, tomato, and cucumber into small pieces.4. Warm the pita bread, then fill each pocket with the grilled chicken, lettuce, tomato, and cucumber.5. Drizzle extra Greek yogurt sauce over the filling before serving.
---	--	---

Fig. 51: Output example 2

Enter Ingredients:

Potato, Cheese

Recent Waste Items:

- ☒ Beef
- ☒ Mushroom
- ☐ tomato

Get Recipe

Fig. 52: Input example 3

Recommended Recipes:

Beef and Mushroom Stuffed Potatoes

Ingredients:

- 4 large potatoes
- 1 lb ground beef
- 1 cup mushrooms, chopped
- 1 cup shredded cheese
- Salt and pepper to taste
- Olive oil

Directions:

1. Preheat oven to 400°F (200°C).
2. Wash and dry the potatoes, then rub them with olive oil and sprinkle with salt. Bake for about 45-50 minutes until tender.
3. In a skillet, cook the ground beef until browned. Add the chopped mushrooms and cook until softened.
4. Cut the tops off the baked potatoes and scoop out the flesh, leaving a thin layer inside the skins.
5. Mix the potato flesh with the beef and mushroom mixture, then stuff the potato skins with the filling.
6. Top each stuffed potato with shredded cheese and bake for another 10-15 minutes until the cheese is melted and bubbly.

Cheesy Beef and Mushroom Skillet

Ingredients:

- 1 lb ground beef
- 1 cup mushrooms, sliced
- 1 onion, diced
- 1 cup shredded cheese
- Salt, pepper, and your favorite seasonings
- Olive oil

Directions:

1. Heat olive oil in a skillet over medium heat. Add the diced onion and cook until translucent.
2. Add the ground beef to the skillet and cook until browned.
3. Stir in the sliced mushrooms and season with salt, pepper, and your favorite seasonings.
4. Once the mushrooms are cooked, sprinkle shredded cheese over the beef and mushroom mixture.
5. Cover the skillet and let the cheese melt. Serve hot and enjoy!

Mushroom and Cheese Stuffed Beef Burgers

Ingredients:

- 1 lb ground beef
- 1 cup mushrooms, finely chopped
- 1 cup shredded cheese
- 1 egg
- Bread crumbs
- Salt, pepper, and seasonings of your choice
- Burger buns and toppings of your choice

Directions:

1. In a bowl, mix together the ground beef, chopped mushrooms, shredded cheese, egg, bread crumbs, salt, pepper, and seasonings.
2. Form the mixture into burger patties.
3. Heat a grill or skillet over medium-high heat. Cook the burger patties for about 4-5 minutes per side, or until cooked to your desired doneness.
4. Assemble your burgers with your favorite toppings and enjoy the delicious mushroom and

Fig. 53: Output example 3

Enter Ingredients:

Shoe, Fan, Couch, Plant, Table, Shirt

Fig. 54: Input example 4

Recommended Recipes:

I'm sorry, but it seems like the ingredients you provided are not suitable for cooking. Can you please provide a list of food ingredients that you have on hand?

Fig. 55: Output example 4

Add A New Food Item

Food name:

Food groups:

Bio degradable: ☐

Add Submit

- Fruit
- Vegetable
- Grains
- Protein
- Dairy

Fig. 56: Adding new food items

Food Waste in Australia [About](#) [Food Items](#) [Waste Audit](#) [Recipe Recommender](#) [Duc Anh Dinh](#) [Logout](#)

Food Items Log

Create Food items that you would like to use to enter the waste audits

[Create new food item](#)

Food Name: Mushroom Food Group: vegetable Bio-degradable: True Edit Delete
Food Name: Beef Food Group: protein Bio-degradable: True Edit Delete
Food Name: tomato Food Group: vegetable Bio-degradable: True Edit Delete
Food Name: capsicum Food Group: vegetable Bio-degradable: True Edit Delete

Fig. 57: Food item log

Add A New Waste Audit

Entry date: 2024-10-08

User: Duc Anh Dinh

Food item:

Mushroom
Beef
tomato
capsicum

Waste type:

Expired Food Waste

Preparation Waste

Uneaten Waste

Amount:

Add Sub

Fig. 58: Adding a new waste audit item

Food Waste in Australia		About	Food Items	Waste Audit	Recipe Recommender	Duc Anh Dinh	Logout
<h3>Waste Audit Log</h3> <p>Create food waste audit to keep track of your food waste</p> <div>Create new entry</div>							
<div> Date: Sept. 28, 2024 User: Duc Anh Dinh Food Items: Beef Waste Type: expired food waste Amount: 5 Edit Delete </div>							
<div> Date: Sept. 28, 2024 User: Duc Anh Dinh Food Items: Mushroom Waste Type: preparation waste Amount: 5 Edit Delete </div>							
<div> Date: Sept. 28, 2024 User: Duc Anh Dinh Food Items: tomato Waste Type: uneaten waste Amount: 15 Edit Delete </div>							

Fig. 59: Waste audit log

Food Waste in Australia		About	Food Items	Waste Audit	Recipe Recommender	Duc Anh Dinh	Logout
<h3>Food Recipe Recommender</h3> <p>Enter leftover food ingredients to get recipe recommendation</p> <p>Enter Ingredients:</p> <input type="text"/> <p>Recent Waste Items:</p> <div> <input type="checkbox"/> Beef <input type="checkbox"/> Mushroom <input type="checkbox"/> tomato </div> <div>Get Recipe</div>							

Fig. 60: Recent waste items



Fig. 61: RecipeS home page upload page

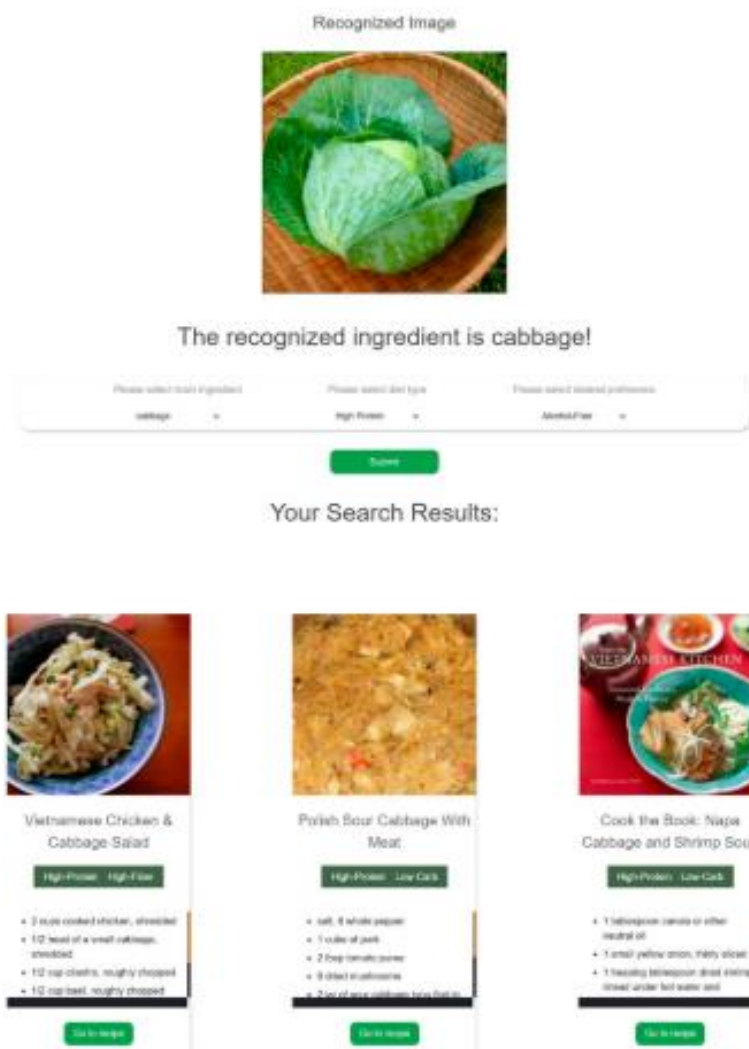


Fig. 62: Recognize the ingredient and use that as the main ingredient in the recipe recommender

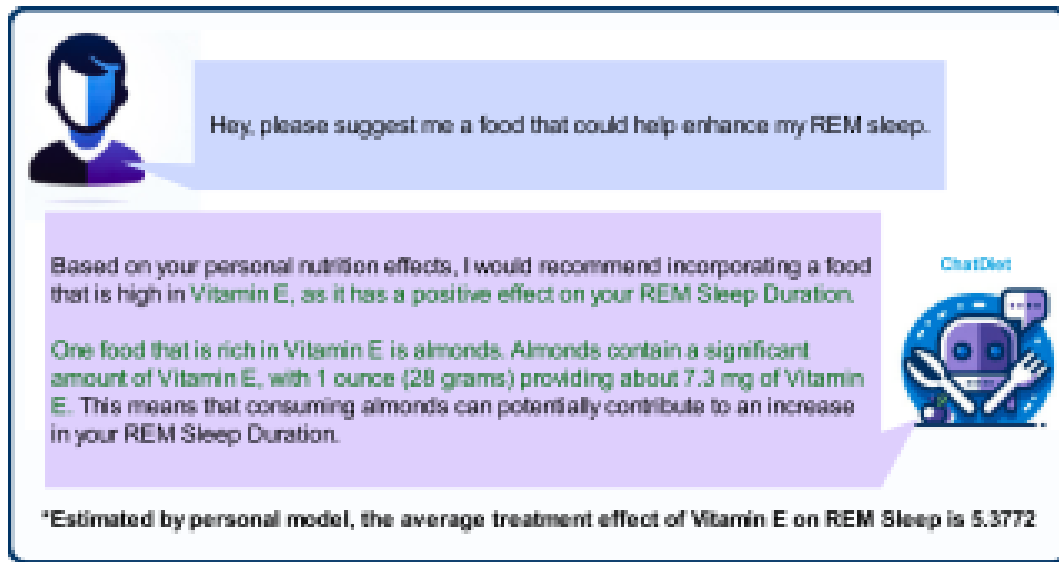


Fig. 63 Food recommendation and explanation based on nutrition effect from ChatDiet

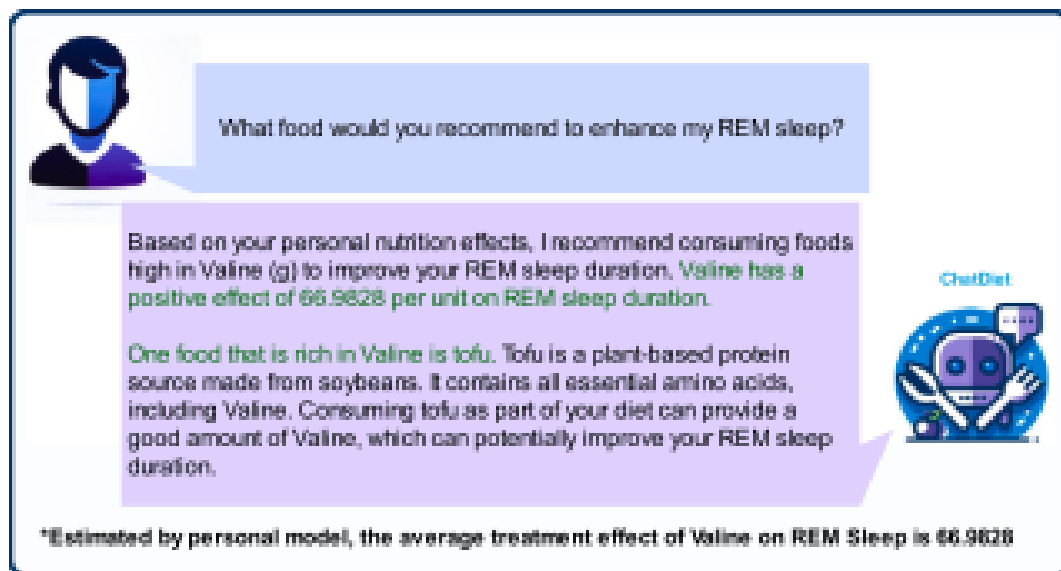


Fig. 64: Personalized feedback from ChatDiet

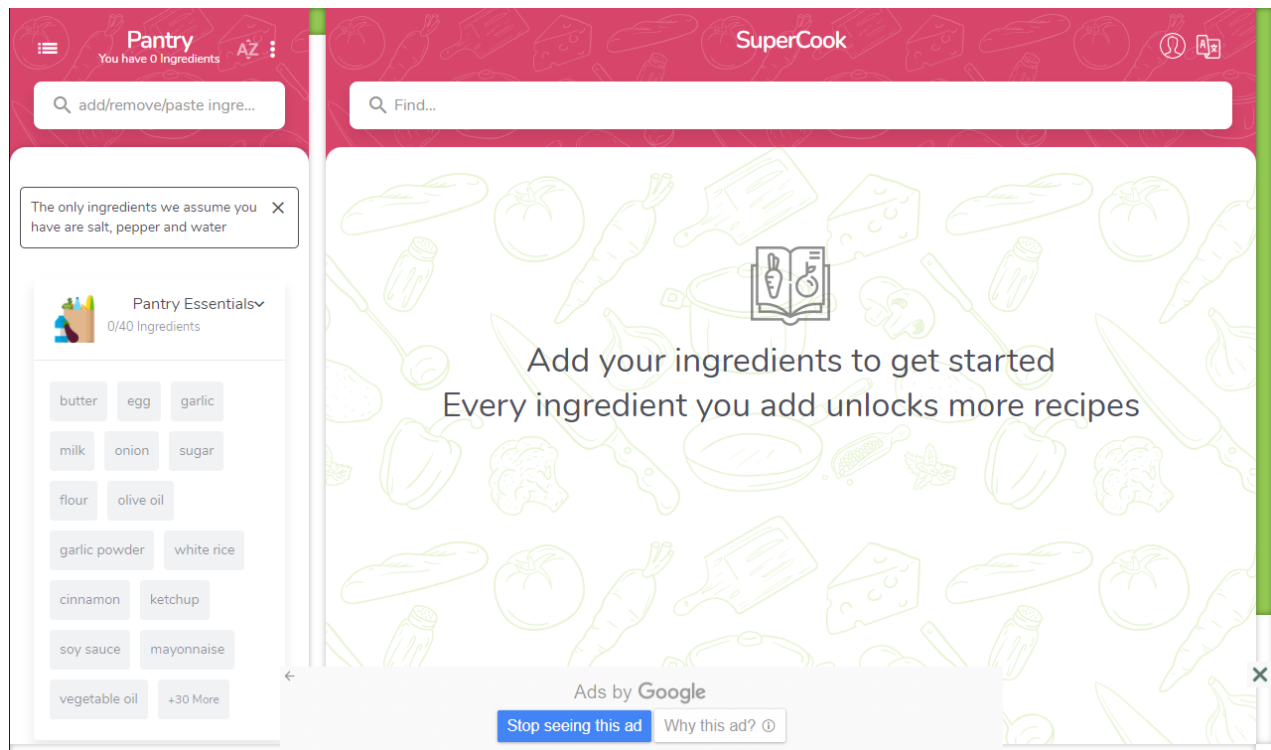


Fig. 65: Supercook web application

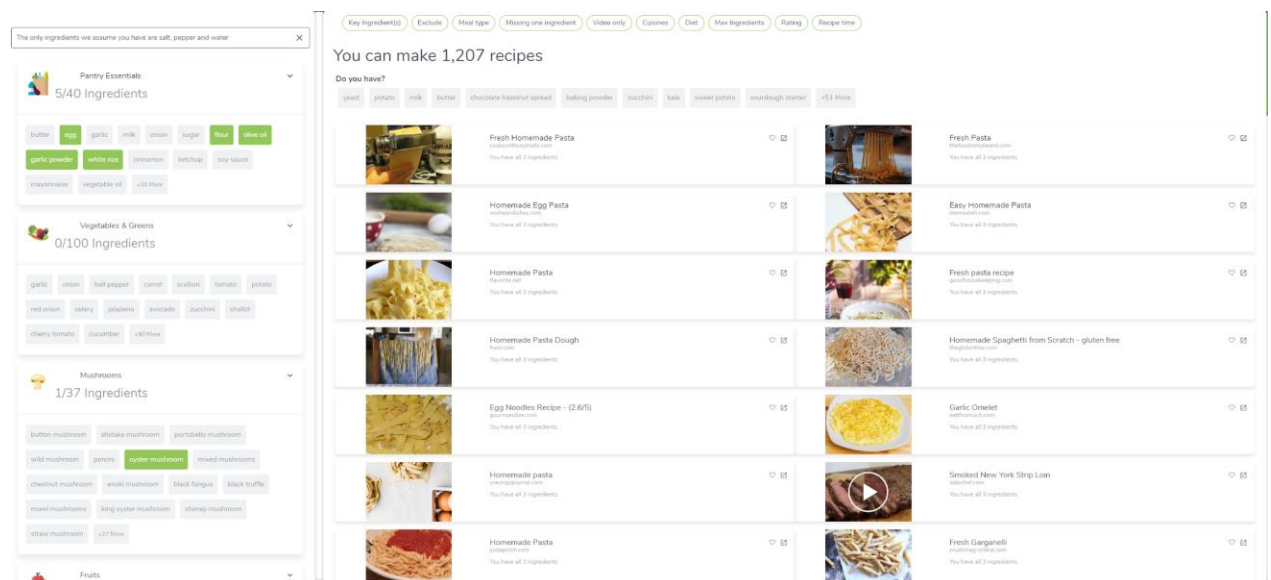


Fig: 66: list of suggested recipes

APPENDIX D: TABLES

Table 5: RecipeIS usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input checked="" type="checkbox"/>	The system provides constant feedback, first providing feedback on the image recognition and telling the user the ingredient recognised.
Match between the system and the real world	<input type="checkbox"/>	The website starts on the upload image page but with the colour contrast to the background it makes very clear from first look that this page is used to upload pictures.
User control and freedom	<input type="checkbox"/>	Most of the navigation of the site is found in the navigation bar which is clear. However, after the user uploads an image and the model recognizes there is no clear exits to go back to the page to upload a different image in the case that the user uploads a wrong image.
Consistency and standards	<input type="checkbox"/>	The website's homepage is not what you would expect following the standards of website design. There are also no instructions that leads to the user to process to use the recipe recommender.
Error prevention	<input checked="" type="checkbox"/>	The model was trained and analysed using a correlation matrix obtaining approximately a value of 96% accuracy and a loss rate of about 4%.(Rodrigues, 2023).
Recognition rather than recall	<input checked="" type="checkbox"/>	Important data from each section of the page is displayed and easily recognisable.
Flexibility and efficiency of use	<input type="checkbox"/>	The site is quite simplistic, so the user must follow all the whole process in order.
Aesthetic and minimalist design	<input checked="" type="checkbox"/>	Simple and minimalist design only including key features of the website. The colour contrast works against the website with sections not remaining consistent with their design.
Help users recognize, diagnose, and recover from errors	<input type="checkbox"/>	No error message was not shown when uploading an image, the app doesn't recognise.
Help and documentation	<input checked="" type="checkbox"/>	Section in the thesis paper showing example usage of the recipe recommendation system.

Table 6: ChatDiet usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input checked="" type="checkbox"/>	The chatbot provides a human like conversation always giving feedback to the user regarding their inquiries.
Match between the system and the real world	<input checked="" type="checkbox"/>	As chatbot assistant are similar provide a similar experience to conversation and is easy for anyone to use.
User control and freedom	<input type="checkbox"/>	Limited user freedom as when a prompt has been entered the chatbot will response and the user doesn't have a way to prevent that data sent.
Consistency and standards	<input checked="" type="checkbox"/>	As chatbot assistant have become popularized the app has a good match between system and real world having a design like other application in the industry.
Error prevention	<input checked="" type="checkbox"/>	The model is effective being tested with 400 queries and the response analysis for each section with no less than 85% accuracy in each section.
Recognition rather than recall	<input checked="" type="checkbox"/>	Previous queries are still displayed when the user enters another query so it can be looked back on.
Flexibility and efficiency of use	<input type="checkbox"/>	Simplistic AI chatbot design with no shortcuts or flexibility of usage.
Aesthetic and minimalist design	<input type="checkbox"/>	The interface is simple and easy to use, however does not contain much information on the usage of the chatbot before the user uses it.
Help users recognize, diagnose, and recover from errors	<input type="checkbox"/>	Unable to be tested.
Help and documentation	<input type="checkbox"/>	The documentation shows a case study inputs and output, but they are quite specific and other example inputs are not explained.

Table 7: Supercook usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input checked="" type="checkbox"/>	The application is responsive whenever pantry ingredients are selected providing clear feedback of recipes to the user.
Match between the system and the real world	<input checked="" type="checkbox"/>	The system takes the approach of giving the user a database storage in the form of the pantry like the real world.
User control and freedom	<input checked="" type="checkbox"/>	The user selects from a list of ingredients and can easily deselect the ingredient if they accidentally select the wrong ingredient.
Consistency and standards	<input type="checkbox"/>	The app layout and features functionality does not follow the standards for an AI application which involve user's queries input.
Error prevention	<input checked="" type="checkbox"/>	Supercook has each ingredient as selector limiting the user input but also preventing user error.
Recognition rather than recall	<input type="checkbox"/>	The pantry list is long and for first time user's it would be hard to select all the ingredients and remember the ingredients selected in the pantry.
Flexibility and efficiency of use	<input type="checkbox"/>	The web application doesn't have shortcuts, and the user must select the ingredients one by one.
Aesthetic and minimalist design	<input checked="" type="checkbox"/>	The design is simple and effective with the pantry list taking the user's input and the recipes will be displayed and can be searched for.
Help users recognize, diagnose, and recover from errors	<input checked="" type="checkbox"/>	Error check prevention in place and the application handles user input very well.
Help and documentation	<input checked="" type="checkbox"/>	The research paper explains the application well with descriptions on app usage and performance.

Table 8: Food waste recipe recommender usability analysis

Usability Heuristic	Fulfilled	Analysis
Visibility of system status	<input type="checkbox"/>	The application is sometimes unclear with user feedback when the AI is loading a response. With longer responses if the user does not notice the loading sign on the page.
Match between the system and the real world	<input checked="" type="checkbox"/>	The basis of the project was built around users' leftover ingredients and allowing them to enter those ingredients to recommender recipes so that they can make use of those ingredients
User control and freedom	<input type="checkbox"/>	Limited user freedom as when a prompt has been entered the chatbot will response and the user doesn't have a way to prevent that data sent
Consistency and standards	<input checked="" type="checkbox"/>	The website fits the consistency and standards of the industry using cards to display important information with multiple navigation to the same places
Error prevention	<input checked="" type="checkbox"/>	The GPT 4 model is effective at recognising invalid input and will response with the appropriate error message if the user doesn't enter ingredients
Recognition rather than recall	<input checked="" type="checkbox"/>	When the user uses other functionality of the site each item are interconnected and displayed when using another feature
Flexibility and efficiency of use	<input checked="" type="checkbox"/>	Waste audit items are shortcut to a checklist that the logged in user can use making it easier for users that have used the app multiple times
Aesthetic and minimalist design	<input checked="" type="checkbox"/>	The application has a clean and distinctive design with each web page complementing each other
Help users recognize, diagnose, and recover from errors	<input checked="" type="checkbox"/>	The GPT 4 model is effective at recognising invalid input and will response with the appropriate error message and notice if the user doesn't enter ingredients
Help and documentation	<input checked="" type="checkbox"/>	Documentation shows how the users would use the app and also shows potential errors that could occur with the GPT model

REFERENCES

- Feng, X., Liu, W., & Jiang, Y. (2020). A model-driven system interface design method based on MVC pattern. *Journal of Physics. Conference Series*, 1549(4). <https://doi.org/10.1088/1742-6596/1549/4/042075>
- Gim, M., Park, D., Spranger, M., Maruyama, K., & Kang, J. (2021). RecipeBowl: A Cooking Recommender for Ingredients and Recipes Using Set Transformer. *IEEE Access*, 9, 143623–143633. <https://doi.org/10.1109/ACCESS.2021.3120265>
- Liang, Y., Xiao, R., Huang, F., Lin, Q., Guo, J., Zeng, W., & Dong, J. (2024). AI nutritionist: Intelligent software as the next generation pioneer of precision nutrition. *Computers in Biology and Medicine*, 178, 108711-. <https://doi.org/10.1016/j.compbimed.2024.108711>
- Lubos, S., Tran, T. N. T., Felfernig, A., Polat Erdeniz, S., & Le, V.-M. (2024). LLM-generated Explanations for Recommender Systems. *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, 276–285. <https://doi.org/10.1145/3631700.3665185>
- Morol, Md. K., Rokon, Md. S. J., Hasan, I. B., Saif, A. M., Khan, R. H., & Das, S. S. (2022). Food Recipe Recommendation Based on Ingredients Detection Using Deep Learning. *Proceedings of the 2nd International Conference on Computing Advancements*, 191–198. <https://doi.org/10.1145/3542954.3542983>
- Nielson, J. (1994, April 24). 10 usability heuristics for user interface design. NNgroup. <https://www.nngroup.com/articles/ten-usability-heuristics/>
- Nilesh, N., Kumari, M., Hazarika, P., & Raman, V. (2019). Recommendation of Indian Cuisine Recipes Based on Ingredients. *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*, 96–99. <https://doi.org/10.1109/ICDEW.2019.00-28>
- Nishant, V. (2020, August 20). Django MVT architecture: A fresh take on Classic MVC. AskPython. <https://www.askpython.com/django/django-mvt-architecture>

Peng, Y. (2024). *SuperCook: Design Interactive Culinary Experiences With AI Technology*. ProQuest Dissertations & Theses Global. <https://www.proquest.com/dissertations-theses/supercook-design-interactive-culinary-experiences/docview/3057563071/se-2>

Rodrigues, M. S., Fidalgo, F., & Oliveira, Â. (2023). RecipeIS—Recipe Recommendation System Based on Recognition of Food Ingredients. *Applied Sciences*, 13(13), 7880-. <https://doi.org/10.3390/app13137880>

Rostami, A. (2024). *An Integrated Framework for Contextual Personalized LLM-Based Food Recommendation*. ProQuest Dissertations & Theses Global. <https://www.proquest.com/dissertations-theses/integrated-framework-contextual-personalized-llm/docview/3060126764/se-2>

Roumeliotis, K. I., Tselikas, N. D., & Nasiopoulos, D. K. (2024). Precision-Driven Product Recommendation Software: Unsupervised Models, Evaluated by GPT-4 LLM for Enhanced Recommender Systems. *Software*, 3(1), 62–80. <https://doi.org/10.3390/software3010004>

Teng, C.-Y., Lin, Y.-R., & Adamic, L. A. (2012). Recipe recommendation using ingredient networks. *Proceedings of the 4th Annual ACM Web Science Conference*, 298–307. <https://doi.org/10.1145/2380718.2380757>

Wang, L.-C., Zhang, H., Ginsberg, N., Ban, A. N., & Kotanko, P. (2024). 729 Application of ChatGPT to support nutritional recommendations for dialysis patients. *Nephrology, Dialysis, Transplantation*, 39(Supplement_1). <https://doi.org/10.1093/ndt/gfae069.861>

Wang, T., Wang, C., & Wang, C. (2024). Embracing LLMs for Point-of-Interest Recommendations. *IEEE Intelligent Systems*, 39(1), 56–59. <https://doi.org/10.1109/MIS.2023.3343489>

Yang, Z., Khatibi, E., Nagesh, N., Abbasian, M., Azimi, I., Jain, R., & Rahmani, A. M. (2024). ChatDiet: Empowering personalized nutrition-oriented food recommender chatbots through an LLM-augmented framework. *Smart Health (Amsterdam)*, 32, 100465-. <https://doi.org/10.1016/j.smhl.2024.100465>