ChefFly

Submitted in partial fulfillment of the requirements of the degree
of

BACHELOR OF COMPUTER ENGINEERING

by

Anushree Salunke (20102179)

Eisha Saini (20102025)

Pooja Tumma (20102126)

Sanskruti Shinde (21202018)

Guide:

Prof. D.S. Khachane

Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

(2022-2023)

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

CERTIFICATE

This is to certify that the Mini Project 2B entitled "**ChefFly**" is a bonafide work of **"Anushree Salunke (20102179), Eisha Saini (20102025), Pooja Tumma (20102126), Sanskruti Shinde (21202018)"** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Engineering.**

Guide:	Project Coordinator:	Head of Department
Prof. D.S. Khachane	Prof. D.S. Khachane	Prof. S.H. Malave

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

Project Report Approval for Mini Project-2B

This project report entitled "**ChefFly**" by **Anushree Salunke, Eisha Saini, Pooja Tumma, Sanskruti Shinde** is approved for the partial fulfillment of the degree of ***Bachelor of Engineering*** in ***Computer Engineering***, ***2022-23***.

Examiner Name	Signature
1.__________________	______________
2.___________________	______________

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Anushree Salunke-20102179

Eisha Saini-20102025

Pooja Tumma-20102126

Sanskruti Shinde-21202018

Date:

Abstract

Recipe generators are a new and innovative tool that can help home cooks overcome the challenges of meal planning and come up with exciting and delicious recipes. These generators use machine learning algorithms to analyze a vast array of ingredients and suggest personalized recipe recommendations based on individual preferences and dietary restrictions.The purpose of this abstract is to highlight the benefits and limitations of recipe generators as a tool for meal planning. While recipe generators can be a great resource for home cooks, there are still challenges to be addressed, such as cultural differences in cuisine, the need for detailed instructions, and the accuracy of the recipe recommendations. Overall, recipe generators have the potential to revolutionize the way that we plan and prepare meals, making it easier and more enjoyable for everyone to experiment with new ingredients and cooking techniques. As technology continues to advance, we can expect to see even more sophisticated and effective recipe generators that cater to an even wider range of preferences and dietary restrictions. To solve this issue, we have developed a web application called ChefFly that will offer recipes based on the ingredients that are readily available as well as a number of other features. ChefFly is a software that can extract information from existing recipes and learn how to make new dishes. It uses artificial intelligence (AI) to analyze the ingredients and instructions, and then creates recipes based on this information.

Keywords: Machine-Learning, NLP, Technology, AI.

CONTENTS

Sr. No.	Chapter Name	Page No.
1	**Introduction**	8
2	**Literature Survey**	9
3	**Problem Statement, Objective & Scope**	12
4	**Proposed System**	14
5	**Project Plan**	21
6	**Experimental Setup**	22
7	**Implementation Details**	24
8	**Results**	34
9	**Conclusion**	38
10	**References**	39

LIST OF FIGURES

Sr. No.	Figure Name	Page No.
1	Gantt Chart	21
2	Architecture Diagram	14
3	Data Flow Diagram	15
4	Use Case Diagram	18
5	Sequence Flow Diagram	19
6	Activity Diagram	20

Chapter 1
Introduction

A recipe generator is an online tool or application that uses various algorithms and data sources to generate recipes based on specific criteria such as dietary restrictions, cuisine, cooking time, and ingredients available. These tools can assist users in finding new and creative meal ideas, as well as helping them to plan their grocery shopping and meal preparation more efficiently. Recipe generators typically operate by allowing users to input various parameters, such as the type of dish they are looking to make, the ingredients they have on hand, and any dietary restrictions or preferences they may have. The generator then searches its dataset and produces a recipe that meets the user's criteria. It's a very challenging task to encode this information into a machine-understandable format, and it's even harder to learn from this information to create new recipes by combining ingredients and spices from different cuisines. We're going to use some methods called "word embedding" and "neural networks" to see if we can create recipes that taste like the real thing. When cooking, sometimes we can't find the ingredients we need for a recipe. Sometimes we can replace the ingredients with other ingredients, or we can use a different recipe. There is a model called a neural network that can help us choose an alternative recipe or ingredients when we can't find the ones we need. This model takes into account the style of cuisine, the importance of the ingredients, and how popular the recipe is.We used natural language processing (NLP) techniques to help people choose alternative ingredients or recipes.The first step was acquiring training data which includes existing recipes and data gathered from cooking websites. This data was then analyzed to understand how recipes are constructed. The results of this analysis were used to clean the data, so texts that were ungrammatical, irrelevant or lacked crucial features (like ingredients) were removed. This data was then used to train a deep learning language model. The model was then evaluated using NLG (Natural Language Generation) metrics and human-based study. Recipe generators that use Natural Language Processing (NLP) have become increasingly popular due to their ability to process and understand natural language input from users. By analyzing the user's input, NLP-based recipe generators can generate recipe suggestions that are tailored to the user's preferences, dietary restrictions, and available ingredients. NLP-based recipe generators can also take into account various factors, such as cooking time, complexity, and nutritional information, to provide a more comprehensive recipe suggestion. They can even analyze user feedback to improve their suggestions over time. A recipe generator with NLP can provide personalized recipe suggestions based on user preferences, dietary restrictions, and other relevant factors. This makes it a useful tool for people with specific dietary needs or preferences. The recipe generator can analyze user feedback, such as ratings and reviews, to improve its recipe suggestions over time. This allows the recipe generator to learn from user behavior and preferences, making its suggestions more accurate and relevant.Surrogate network derived from user-created suggestions for modifications, can be broken down into many communities of functionally equivalent ingredients and captures user preferences for healthier recipe variants. Our experiments show that recipe ratings can be well predicted with properties derived from combinations of ingredient networks and nutritional information.

Overall, NLP-based recipe generators provide a convenient and user-friendly way to discover new and unique recipes, simplify the meal planning process, and make use of ingredients that may be available in your pantry. As NLP technology continues to evolve, we can expect recipe generators to become even more sophisticated and accurate in their suggestions, making them an essential tool for anyone looking to explore new culinary horizons.

Chapter 2

Literature Survey

1. **Recipe recommendation using ingredient networks (Published : 22 June 2012)**

   The paper proposes a method for recommending recipes based on ingredient networks.The proposed system uses a dataset of over 60,000 recipes and their ingredients to construct a network where each ingredient is represented as a node and edges connect ingredients that commonly appear together in recipes. The authors use network analysis techniques to identify clusters of ingredients that tend to co-occur, and use these clusters to recommend recipes. The system takes a user-specified set of ingredients as input and uses the ingredient network to identify related ingredients and recipe clusters. The authors evaluate their system using a dataset of user ratings and report promising results in terms of recommendation accuracy and diversity.

2. **Food Recipe Alternation and Generation with Natural Language Processing Technique. (Published : 2020)**

   The paper proposes a natural language processing-based approach for recipe alternation and generation.The proposed system consists of three main components: pre-processing, alteration, and generation. In the pre-processing phase, the system parses the input recipe and extracts its ingredients, cooking methods, and instructions. The alteration phase then uses a word embedding model to replace ingredients and cooking methods with alternative options, based on user preferences or dietary restrictions. Finally, the generation phase uses a language model to generate new recipes based on the altered ingredients and methods. The authors evaluate their system using a dataset of 500 recipes and report promising results in terms of recipe similarity and coherence.

3. **Improving text-to-image generation with object layout guidance. (Published : 20 May 2021)**

The paper proposes a method for improving the generation of images from text descriptions by incorporating object layout guidance.The proposed system uses a neural network to generate an initial image based on the input text description. The system then uses a separate neural network to predict the locations and sizes of objects in the image based on the input text description. The two networks are trained jointly using a combination of adversarial and perceptual loss functions.The authors evaluate their system using several benchmark datasets and report significant improvements in terms of image quality and object layout accuracy compared to existing state-of-the-art methods. The paper also provides a detailed analysis of the factors that contribute to the effectiveness of the proposed method.

4. **User-item content awareness in matrix factorization based collaborative recommender systems. (Published 21 May 2020)**

The paper proposes a method for enhancing collaborative filtering recommender systems by incorporating user and item content information.The proposed system uses matrix factorization techniques to model user-item interactions in a low-dimensional latent space. In addition, the system uses a separate matrix factorization approach to model user and item content information, which is then combined with the user-item interaction model to improve recommendation accuracy. The paper provides a useful contribution to the field of collaborative filtering recommender systems, demonstrating the potential of incorporating user and item content information for improving recommendation accuracy. The proposed system has implications for a wide range of applications, including recipe recommendation, where the ability to incorporate user and item content information such as dietary preferences and ingredient lists could enhance the relevance and personalization of recommendations.

Research Paper	ANALYSIS
1. Recipe recommendation using ingredient network.	This paper explains the working of a recipe recommendation. It is used for recipe recommendation.
2. Food Recipe Alternation and Generation with Natural Language Processing Technique. .	This paper explains the NLP techniques and recipe generation techniques. It is used for natural language processing techniques.
3. User-item content awareness in matrix factorization based collaborative recommender systems	This explains the pattern and mindset of the user with their food choices. It is used for recommending each active user by comparing with the preferences.
4. User-item content awareness in matrix factorization based collaborative recommender systems	This explains the pattern and mindset of the user with their food choices. It is used for recommending each active user by comparing with the preferences.
5. Improving text-to-image generation with object layout guidance	This paper explains the technique of text-to-image generation. It is used for text to image generation with object layout design.

Chapter 3

Problem Statement, Objective & Scope

Problem Statement: -

To create a web app that suggests recipes based on available ingredients to help individuals overcome the challenge of cooking with limited supplies during pandemics, lockdowns or in day-to-day life of hostelers.

Our project is intended to address the issue that arises when attempting to prepare a specific recipe but running out of some ingredients. To solve this issue, we have developed a web application called ChefFly that will offer recipes based on the ingredients that are readily available as well as a number of other features.

Objective: -

- To present users with either pre-existing recipes or innovative dishes based on the ingredients they have specified.

- To come up with recipes that include both major and minor ingredients.

- It may possibly happen that a user may find trouble to spell the ingredients so our web app will recommend the ingredients based on the characters given as input.

- Based on ingredients our app will generate the name of the dish, the quantity of the ingredients required for the dish and the step by step procedure of making that dish.

- Based on the recipe generated an downloadable image of that particular recipe will be generated for the use.

Scope: -

- Taking Ingredients as an input from the user and considering some minor ingredients if not specified.

- Recommendation of ingredients based on the data of available ingredients.

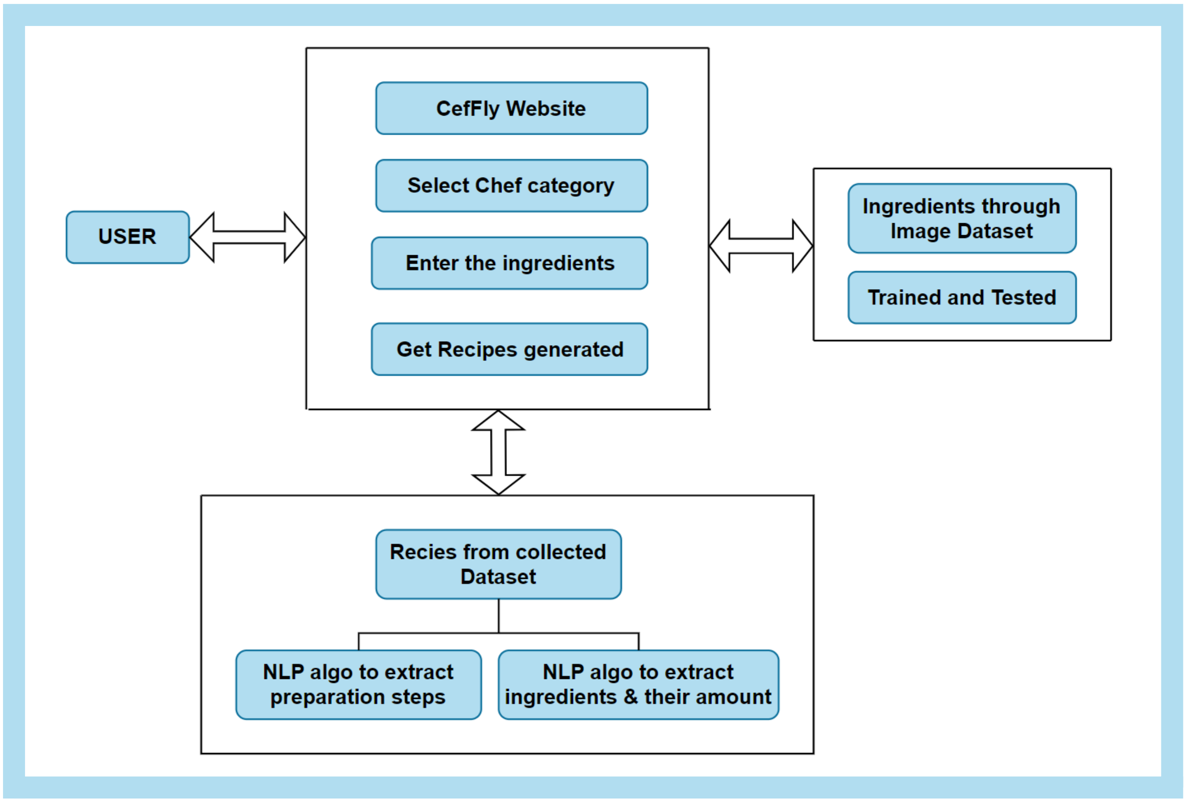
- Generation of the recipes based on major and minor ingredients.

- Image generation of the recipe generated which can be downloaded.

Chapter 4

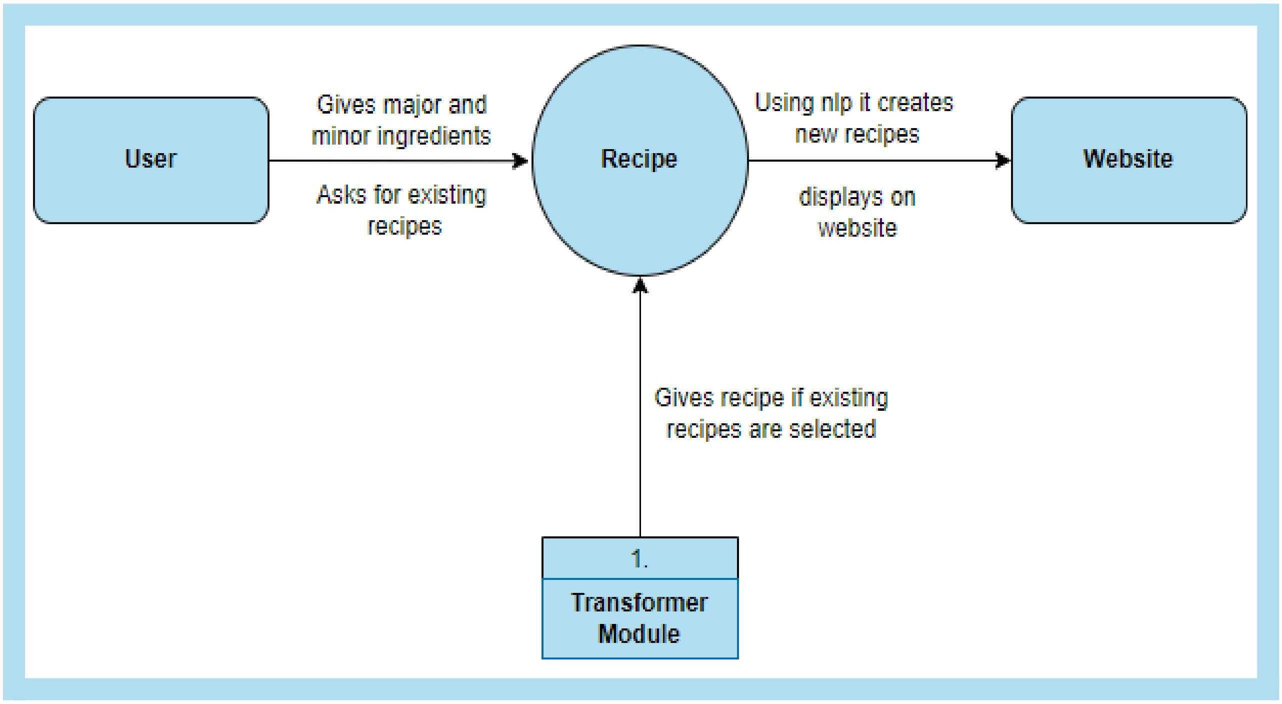
Proposed System

Architecture Diagram :

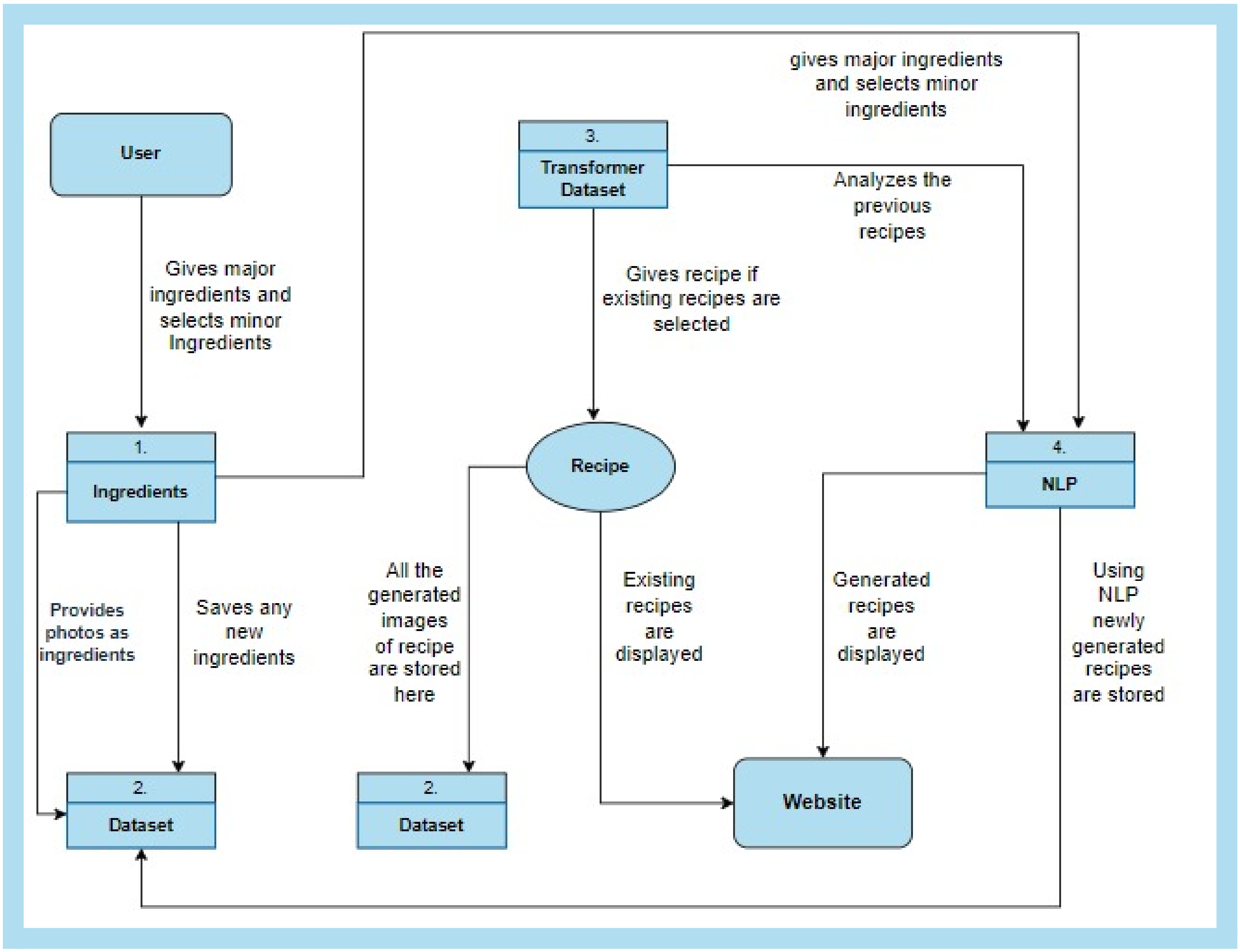

USER
CefFly Website
Select Chef category
Enter the ingredients
Get Recipes generated
Ingredients through Image Dataset
Trained and Tested
Recies from collected Dataset
NLP algo to extract preparation steps
NLP algo to extract ingredients & their amount

Data Flow Diagram (Level 0) :-

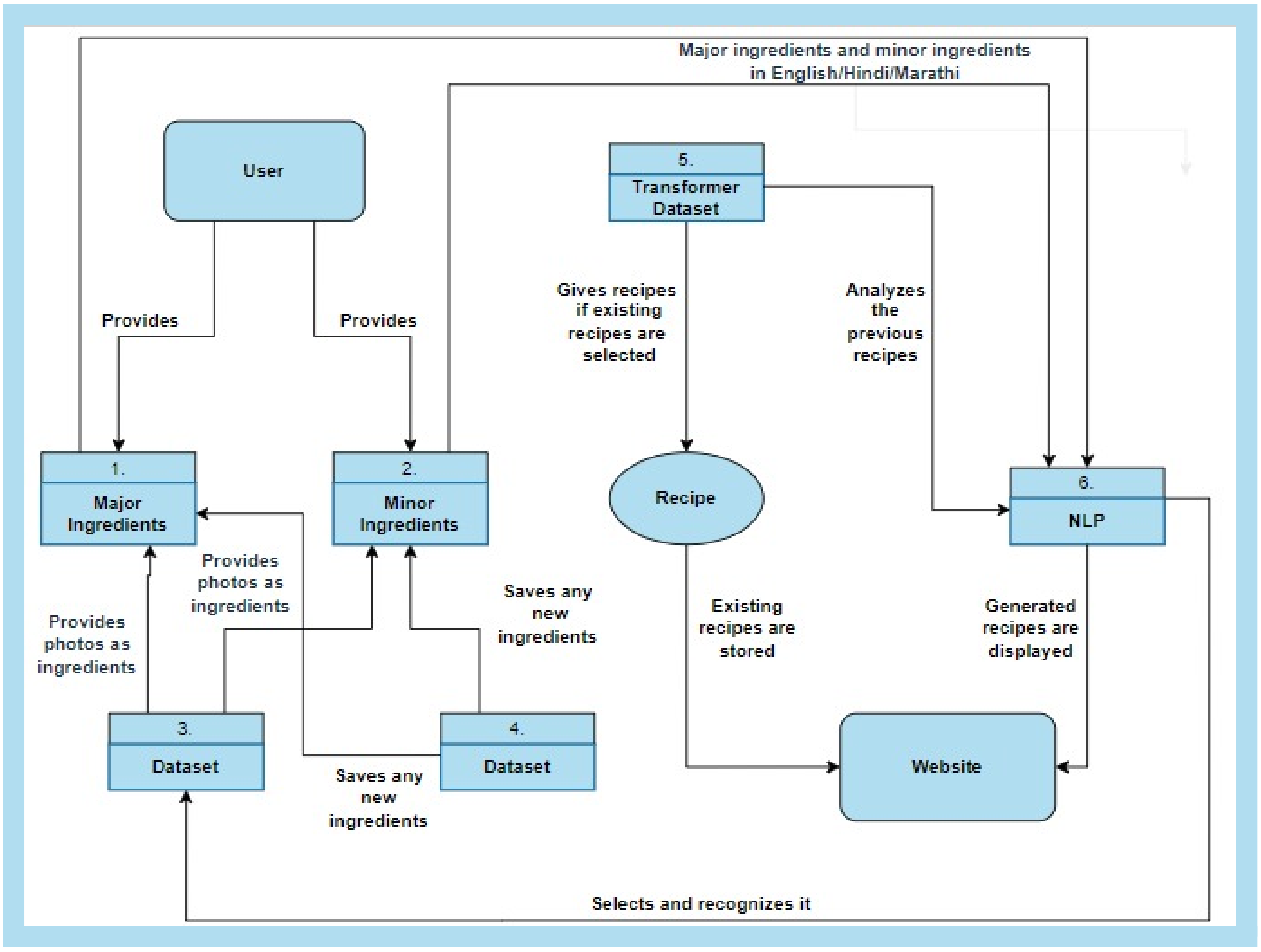
A data flow Diagram is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.


User
Gives major and minor ingredients
Asks for existing recipes
Recipe
Using nlp it creates new recipes
displays on website
Website
Gives recipe if existing recipes are selected
1.
Transformer Module

Data Flow Diagram (Level 1) :-

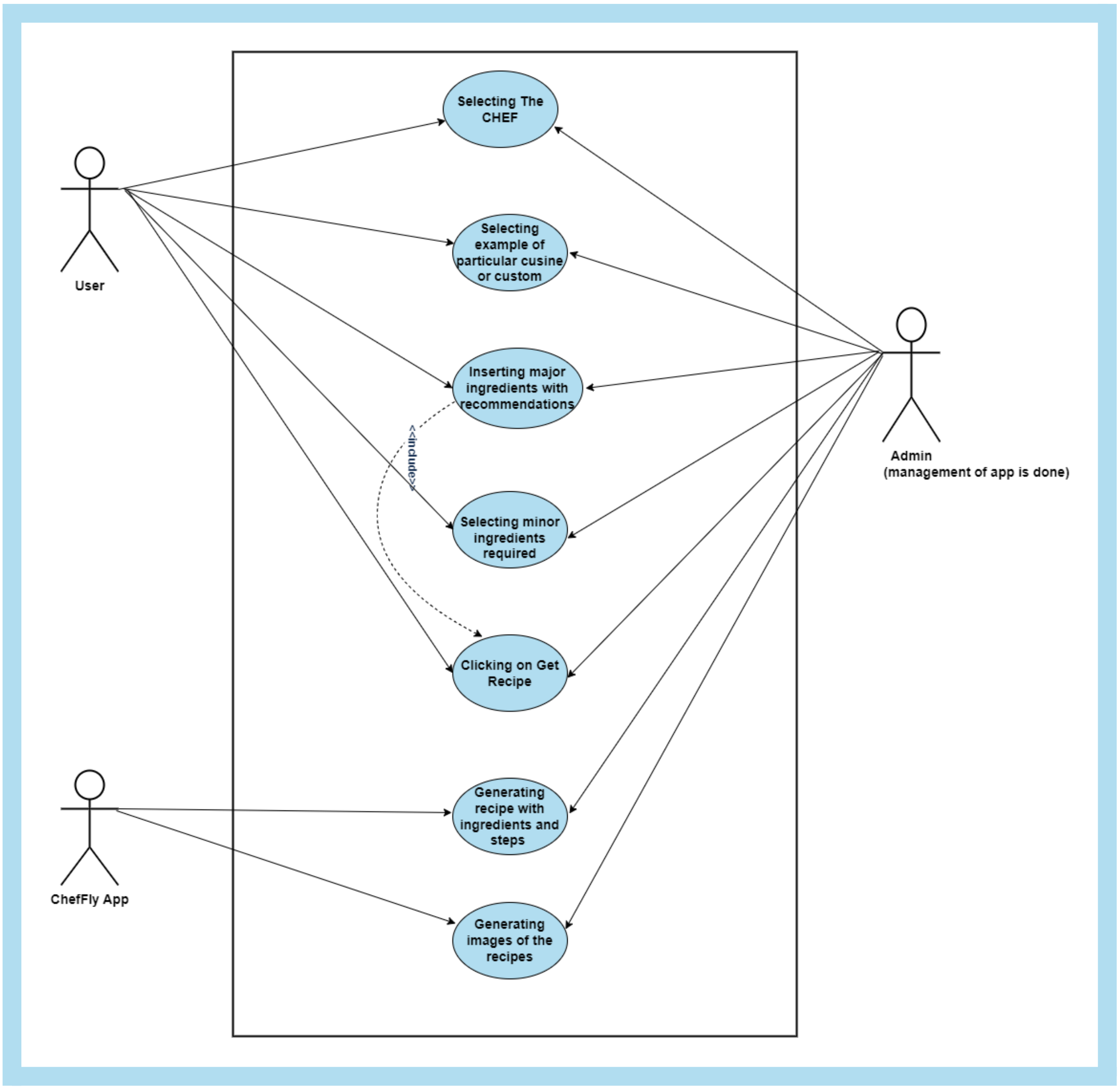

User
Gives major ingredients and selects minor Ingredients
1. Ingredients
Provides photos as ingredients
Saves any new ingredients
2. Dataset
gives major ingredients and selects minor ingredients
3. Transformer Dataset
Analyzes the previous recipes
Gives recipe if existing recipes are selected
Recipe
All the generated images of recipe are stored here
2. Dataset
Existing recipes are displayed
Website
4. NLP
Generated recipes are displayed
Using NLP newly generated recipes are stored

Data Flow Diagram (Level 2) :-


Major ingredients and minor ingredients
in English/Hindi/Marathi
User
5.
Transformer Dataset
Provides
Provides
Gives recipes if existing recipes are selected
Analyzes the previous recipes
1.
Major Ingredients
2.
Minor Ingredients
Recipe
6.
NLP
Provides photos as ingredients
Saves any new ingredients
Existing recipes are stored
Generated recipes are displayed
Provides photos as ingredients
3.
Dataset
4.
Dataset
Website
Saves any new ingredients
Selects and recognizes it

Use Case Diagram :-

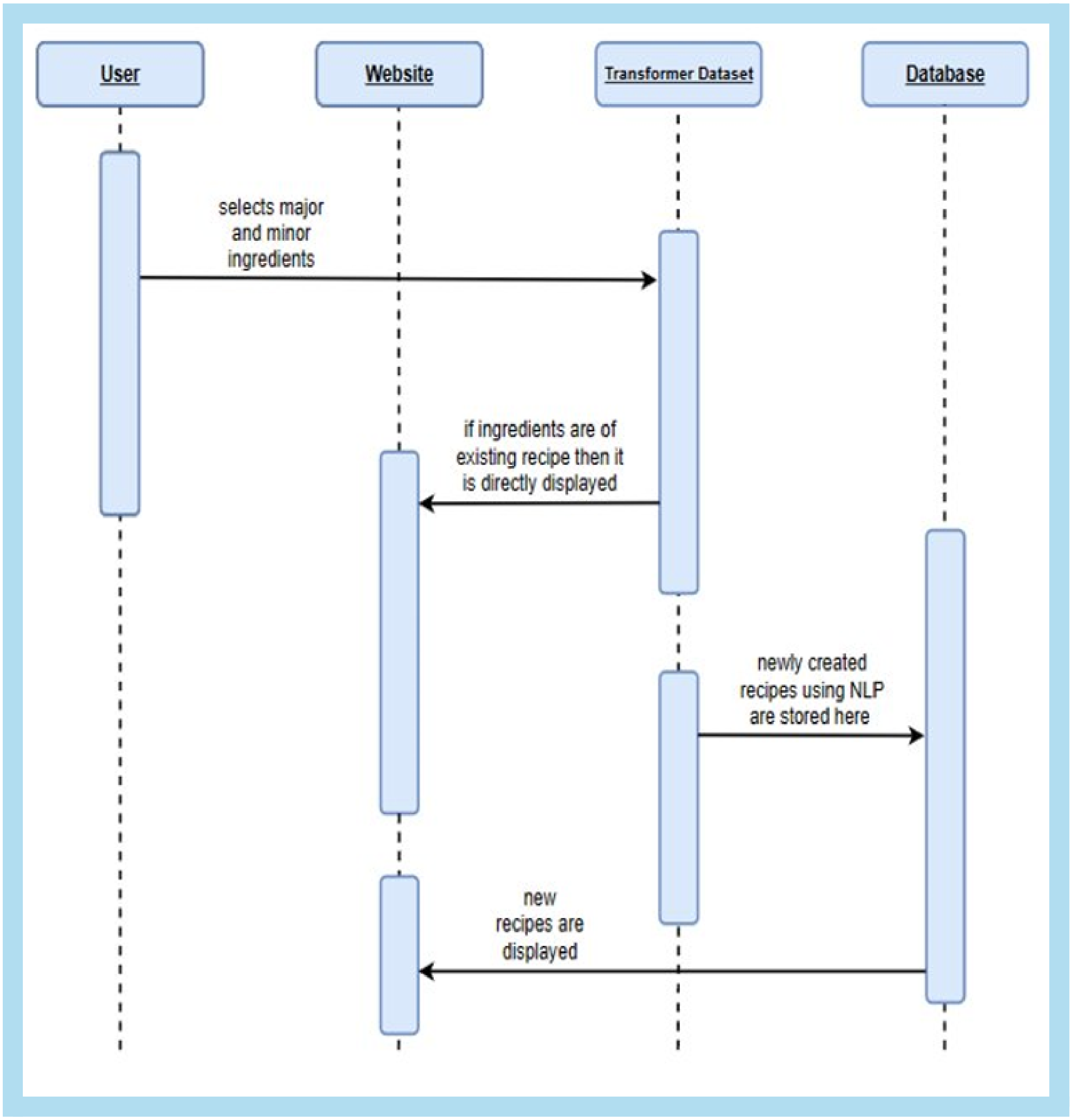
Use case diagram is a graphic depiction of the interactions among the elements of a system. Use cases will specify the expected behavior, and the exact method of making it happen.



User
ChefFly App
Admin
(management of app is done)
Selecting The CHEF
Selecting example of particular cusine or custom
Inserting major ingredients with recommendations
<<include>>
Selecting minor ingredients required
Clicking on Get Recipe
Generating recipe with ingredients and steps
Generating images of the recipes

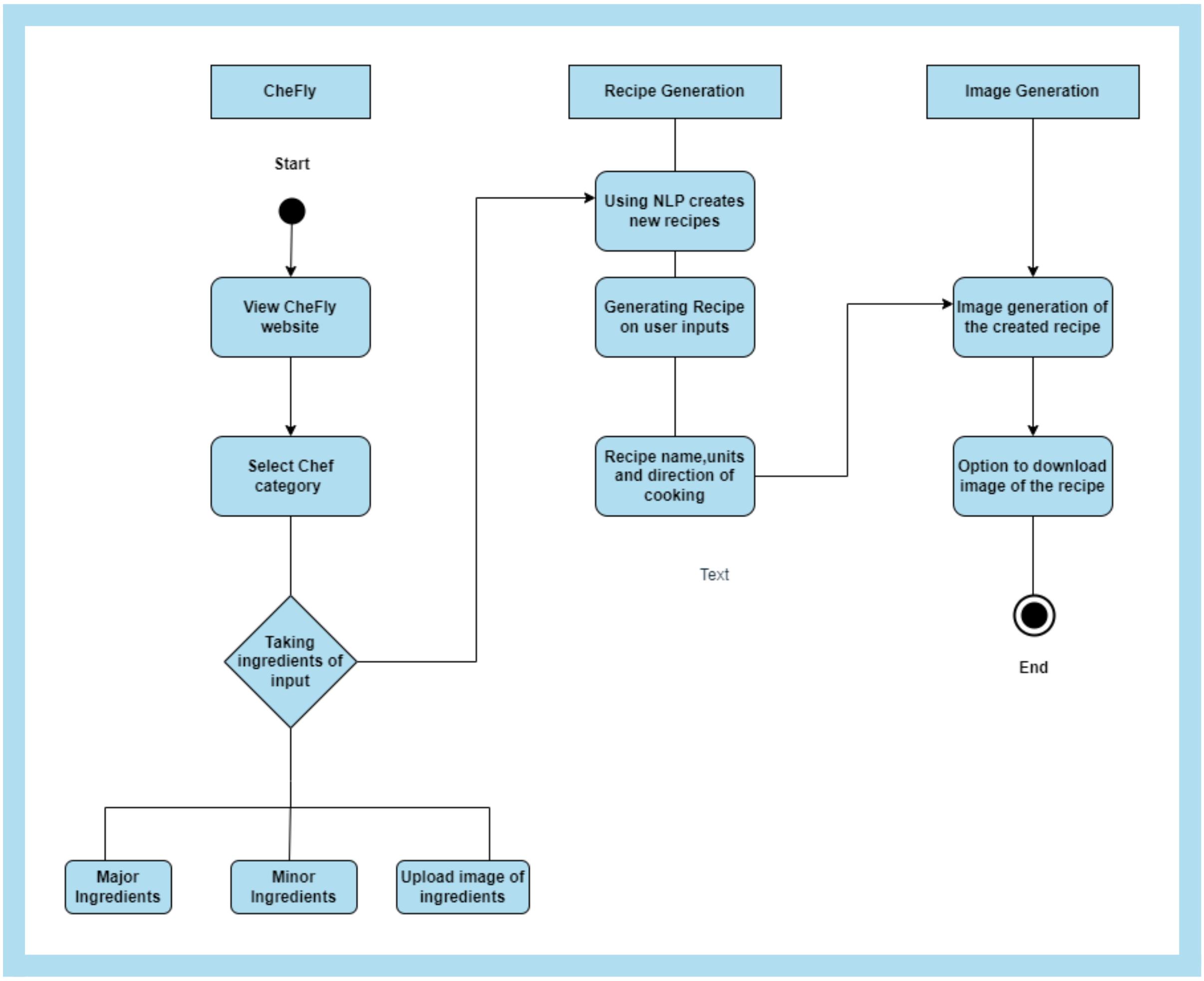
Sequence Diagram :-

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when.


User
Website
Transformer Dataset
Database
selects major and minor ingredients
if ingredients are of existing recipe then it is directly displayed
newly created recipes using NLP are stored here
new recipes are displayed

Activity Diagram :-

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity

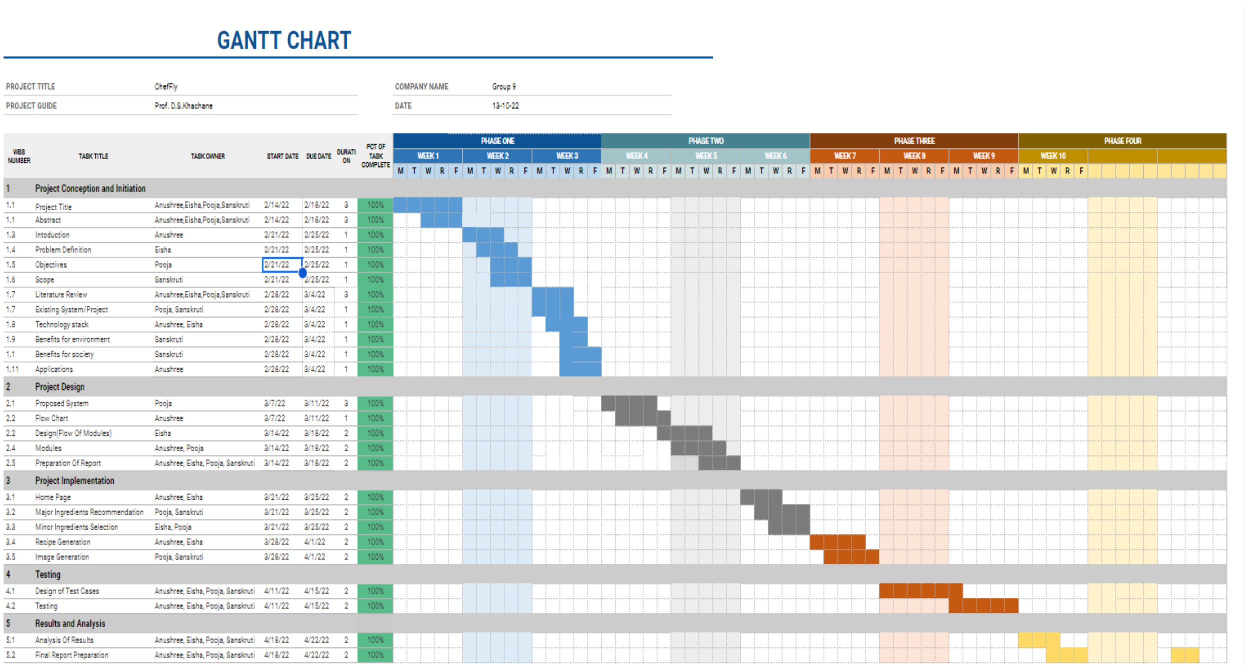

CheFly
Recipe Generation
Image Generation
Start
Using NLP creates new recipes
View CheFly website
Generating Recipe on user inputs
Image generation of the created recipe
Select Chef category
Recipe name,units and direction of cooking
Option to download image of the recipe
Text
Taking ingredients of input
End
Major Ingredients
Minor Ingredients
Upload image of ingredients

Chapter 5

Project Planning

<u>Gantt Chart</u>

Gantt Chart is a chart in which a series of horizontal lines shows the amount of work done or production completed in certain periods of time in relation to the amount planned for those periods



GANTT CHART

PROJECT TITLE	ChefFly	COMPANY NAME	Group 9
PROJECT GUIDE	Prof. D.S.Khachane	DATE	13-10-22

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Conception and Initiation**					
1.1	Project Title	Anushree,Eisha,Pooja,Sanskruti	2/14/22	2/18/22	3	100%
1.1	Abstract	Anushree,Eisha,Pooja,Sanskruti	2/14/22	2/18/22	3	100%
1.3	Intoduction	Anushree	2/21/22	2/25/22	1	100%
1.4	Problem Definition	Eisha	2/21/22	2/25/22	1	100%
1.5	Objectives	Pooja	2/21/22	2/25/22	1	100%
1.6	Scope	Sanskruti	2/21/22	2/25/22	1	100%
1.7	Literature Review	Anushree,Eisha,Pooja,Sanskruti	2/28/22	3/4/22	3	100%
1.7	Existing System/Project	Pooja, Sanskruti	2/28/22	3/4/22	1	100%
1.8	Technology stack	Anushree, Eisha	2/28/22	3/4/22	1	100%
1.9	Benefits for environment	Sanskruti	2/28/22	3/4/22	1	100%
1.1	Benefits for society	Sanskruti	2/28/22	3/4/22	1	100%
1.11	Applications	Anushree	2/28/22	3/4/22	1	100%
2	**Project Design**					
2.1	Proposed System	Pooja	3/7/22	3/11/22	3	100%
2.2	Flow Chart	Anushree	3/7/22	3/11/22	1	100%
2.2	Design(Flow Of Modules)	Eisha	3/14/22	3/18/22	2	100%
2.4	Modules	Anushree, Pooja	3/14/22	3/18/22	2	100%
2.5	Preparation Of Report	Anushree, Eisha, Pooja, Sanskruti	3/14/22	3/18/22	2	100%
3	**Project Implementation**					
3.1	Home Page	Anushree, Eisha	3/21/22	3/25/22	2	100%
3.2	Major Ingredients Recommendation	Pooja, Sanskruti	3/21/22	3/25/22	2	100%
3.3	Minor Ingredients Selection	Eisha, Pooja	3/21/22	3/25/22	2	100%
3.4	Recipe Generation	Anushree, Eisha	3/28/22	4/1/22	2	100%
3.5	Image Generation	Pooja, Sanskruti	3/28/22	4/1/22	2	100%
4	**Testing**					
4.1	Design of Test Cases	Anushree, Eisha, Pooja, Sanskruti	4/11/22	4/15/22	2	100%
4.2	Testing	Anushree, Eisha, Pooja, Sanskruti	4/11/22	4/15/22	2	100%
5	**Results and Analysis**					
5.1	Analysis Of Results	Anushree, Eisha, Pooja, Sanskruti	4/18/22	4/22/22	2	100%
5.2	Final Report Preparation	Anushree, Eisha, Pooja, Sanskruti	4/18/22	4/22/22	2	100%

Chapter 6

Experimental Setup

- **Software Requirements: -**

  1. Frontend : HTML, CSS, JS
  2. Backend: Python
  3. For ML model training transformers module
  4. For NLP and text processing nltk

- **Hardware Requirements: -**

  1. CPU : Basic 64 bit System with i3 core processor or higher.
  2. GPU : T4
  3. RAM: 8 GB
  4. Storage: 128 GB or higher ;
  5. OS : Windows 7, MAC or Linux-based OS

HTML : The Hyper Text Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

CSS : Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside html and JavaScript.

JSS : JSS is an authoring tool for CSS which allows you to use JavaScript to describe styles in a declarative, conflict-free and reusable way. It can compile in the browser, server-side or at build time in Node. JSS is framework agnostic. It consists of multiple packages: the core, plugins, framework integrations and others.

PYTHON : Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule.

MACHINE LEARNING (ML) : Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

NATURAL LANGUAGE PROCESSING (NLP) : Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLTK (Natural Language Toolkit) : NLTK -- the Natural Language Toolkit -- is a suite of open source Python modules, data sets, and tutorials supporting research and development in Natural Language Processing. NLTK requires Python version 3.7, 3.8, 3.9 or 3.10.

Chapter 7

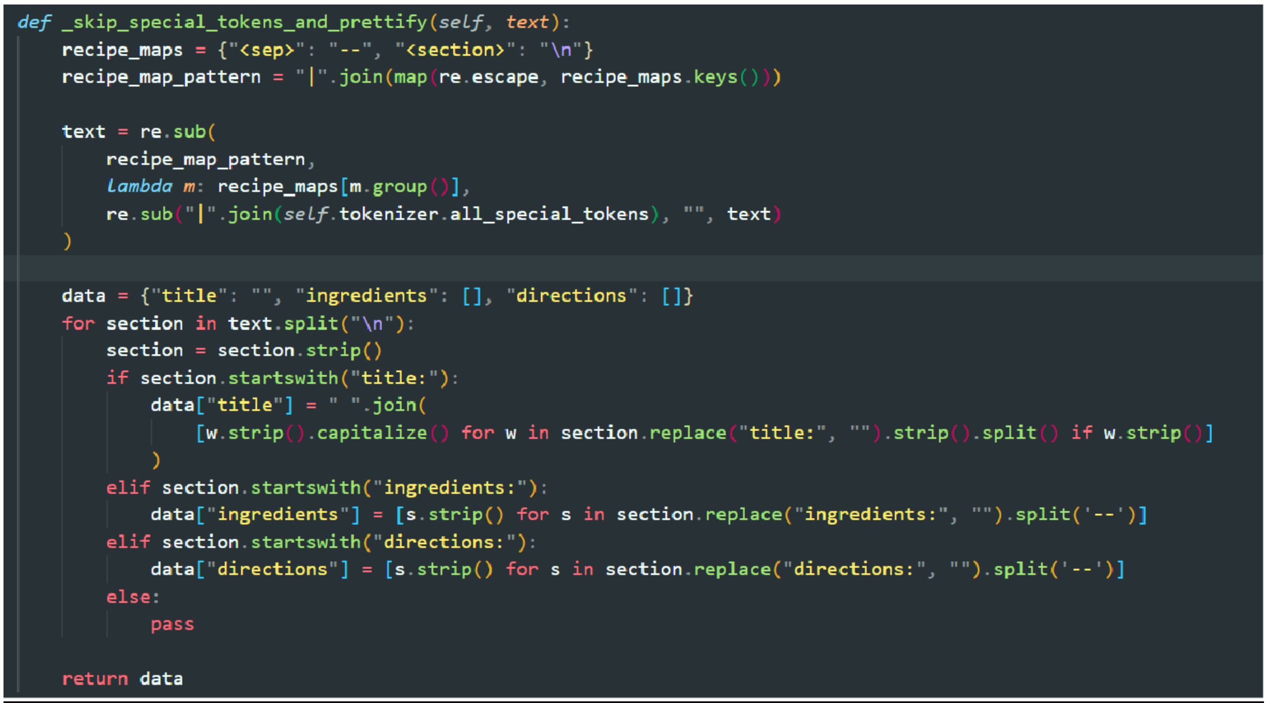
Implementation Details

Modules :

Module 1: Natural Language Processing

Natural Language Processing (NLP) focuses on enabling machines to understand, interpret, and generate human language. NLP involves several tasks such as text classification, sentiment analysis, named entity recognition,language translation, and language generation. In our project we have used NLP concepts of text classification and tokenization for identifying ingredients, the steps of recipes and the title of the recipes.

Code Snippet:



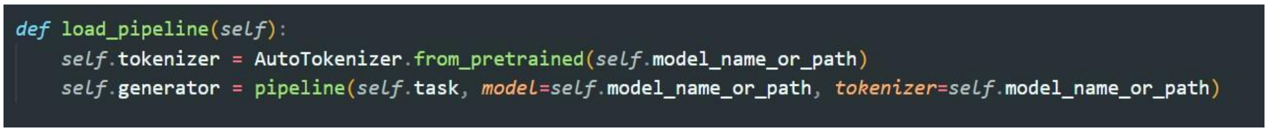
```python
def _skip_special_tokens_and_prettify(self, text):
    recipe_maps = {"<sep>": "--", "<section>": "\n"}
    recipe_map_pattern = "|".join(map(re.escape, recipe_maps.keys()))

    text = re.sub(
        recipe_map_pattern,
        lambda m: recipe_maps[m.group()],
        re.sub("|".join(self.tokenizer.all_special_tokens), "", text)
    )

    data = {"title": "", "ingredients": [], "directions": []}
    for section in text.split("\n"):
        section = section.strip()
        if section.startswith("title:"):
            data["title"] = " ".join(
                [w.strip().capitalize() for w in section.replace("title:", "").strip().split() if w.strip()]
            )
        elif section.startswith("ingredients:"):
            data["ingredients"] = [s.strip() for s in section.replace("ingredients:", "").split('--')]
        elif section.startswith("directions:"):
            data["directions"] = [s.strip() for s in section.replace("directions:", "").split('--')]
        else:
            pass

    return data
```

This is a method in a class that takes in a string of text as an argument and returns a dictionary that represents a recipe with a title, list of ingredients, and list of directions. The method first defines a dictionary called recipe_maps which maps certain special tokens in the text to specific characters. For example, <sep> is mapped to "--" and <section> is mapped to a new line character ("\n").The method then uses the re module to perform regular expression operations on the text. Specifically, it first removes all special tokens using re.sub and then applies the recipe_maps dictionary to replace the remaining special tokens with their corresponding characters. Next, the method initializes a dictionary called data with empty strings for the title and empty lists for the ingredients and directions. It then splits the text into sections based on new line characters and processes each section. If a section starts with "title:", the method extracts the title from the section and stores it in the data dictionary. It first removes the "title:" prefix from the section using replace and then capitalizes each word in the resulting string using a list comprehension. If a section starts with "ingredients:", the method extracts the ingredients from the section and stores them as a list in the data dictionary. It first removes the "ingredients:" prefix from the section using replace and then splits the remaining text using the "--" separator. The resulting list is stored in the data dictionary. If a section starts with "directions:", the method extracts the directions from the section and stores them as a list in the data dictionary, following a similar process as for the ingredients.Finally, the method returns the data dictionary.

Module 2: ML using Transformer Module

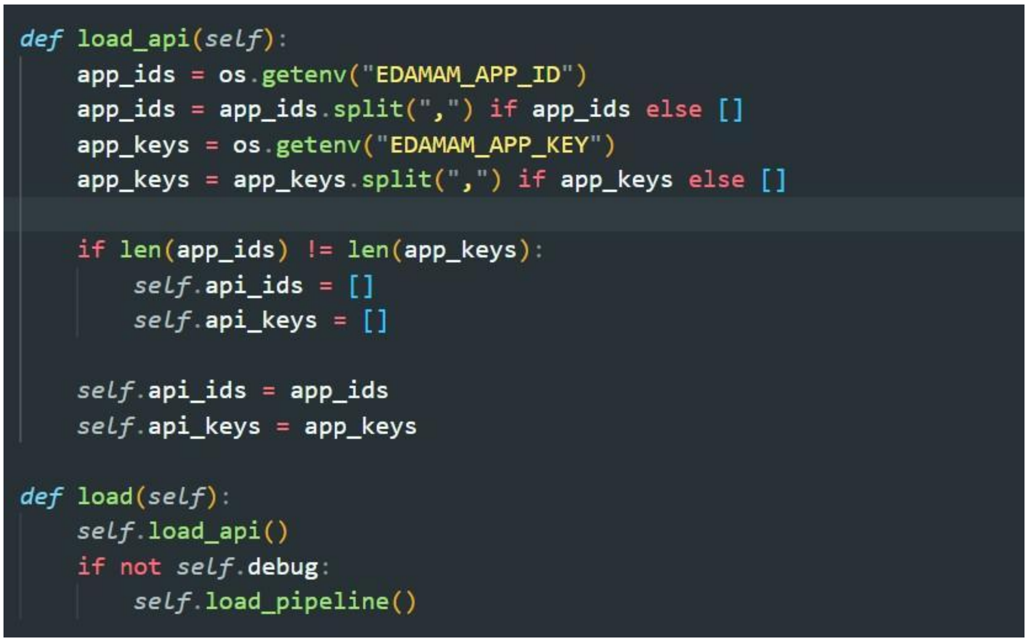
The Transformers module is a key component of the state-of-the-art natural language processing (NLP) models, including the popular BERT, GPT-2, and RoBERTa. The module is based on the self-attention mechanism, which enables the model to capture the dependencies between all input tokens, regardless of their relative positions. In our project we are using a predefined ML model for generating the recipe based on ingredients.

Code Snippet :



```
def load_pipeline(self):
    self.tokenizer = AutoTokenizer.from_pretrained(self.model_name_or_path)
    self.generator = pipeline(self.task, model=self.model_name_or_path, tokenizer=self.model_name_or_path)
```

The first line loads a tokenizer from a pre-trained model using the Hugging Face Transformers library. The AutoTokenizer class automatically detects the appropriate tokenizer for the specified model name or path. The second line creates a pipeline object for a specific NLP task, such as text generation or sentiment analysis, using the pipeline function from the Transformers library. The pipeline object uses the pre-trained model specified by the model_name_or_path argument, along with the tokenizer loaded in the previous step.



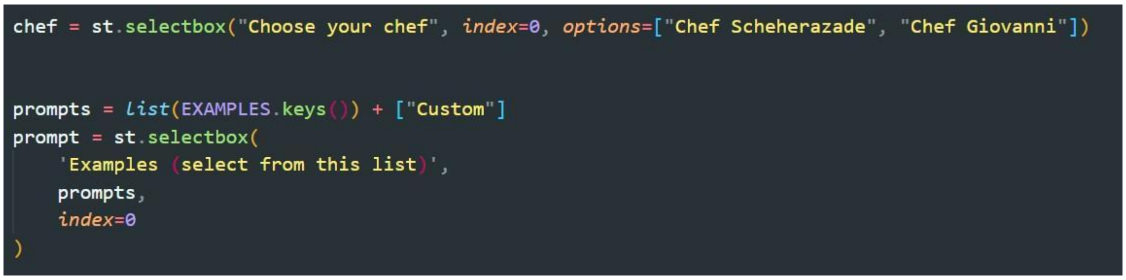
```python
def load_api(self):
    app_ids = os.getenv("EDAMAM_APP_ID")
    app_ids = app_ids.split(",") if app_ids else []
    app_keys = os.getenv("EDAMAM_APP_KEY")
    app_keys = app_keys.split(",") if app_keys else []

    if len(app_ids) != len(app_keys):
        self.api_ids = []
        self.api_keys = []

    self.api_ids = app_ids
    self.api_keys = app_keys

def load(self):
    self.load_api()
    if not self.debug:
        self.load_pipeline()
```

The load_api function loads API IDs and keys from environment variables using the os.getenv function. It splits the IDs and keys into separate lists if they are comma-separated strings. If the length of the ID and key lists does not match, both lists are set to empty. Otherwise, the lists are stored in the class attributes api_ids and api_keys. The load function calls the load_api function and then conditionally calls the load_pipeline function if debug is False. This suggests that the class is used to load and configure various components required for natural language processing and API access.

Module 3: Major Ingredients Recommendation

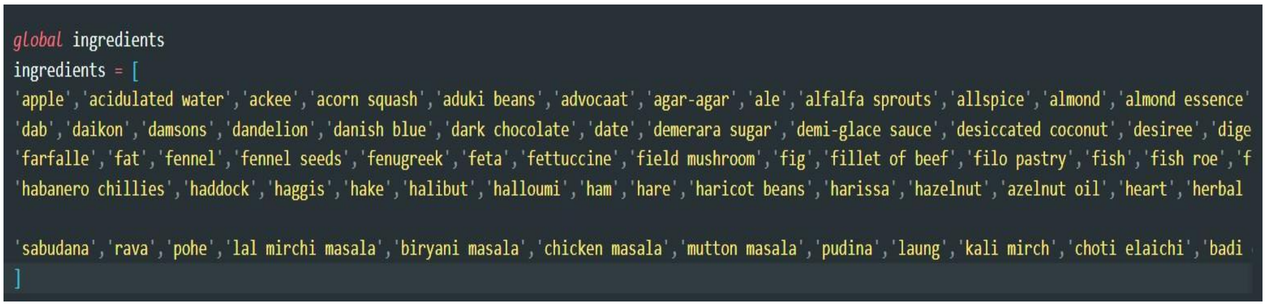
Here we are selecting the ingredients which are required for the dishes or randomly any ingredients. While selecting the ingredients we are also providing the recommendations while typing the ingredients for the user to get easy access. Also some cuisines are also provided which users can access

Code Snipet :



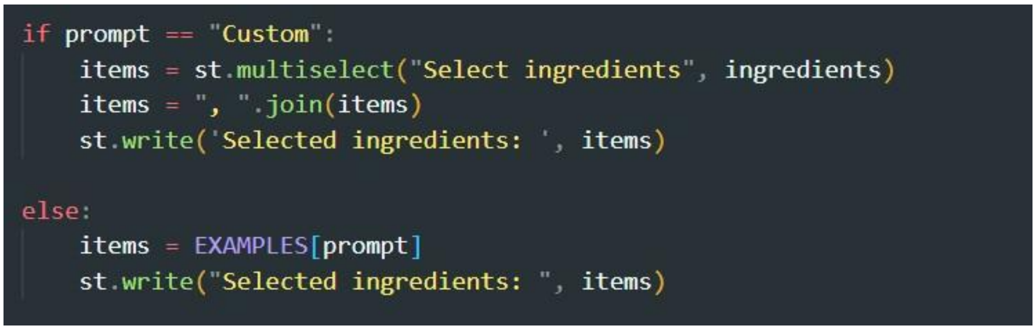
```
chef = st.selectbox("Choose your chef", index=0, options=["Chef Scheherazade", "Chef Giovanni"])


prompts = list(EXAMPLES.keys()) + ["Custom"]
prompt = st.selectbox(
    'Examples (select from this list)',
    prompts,
    index=0
)
```

Here we select the chef from the above two. Then we can either select the cuisines from the EXAMPLES or we can select custom.



```
global ingredients
ingredients = [
'apple','acidulated water','ackee','acorn squash','aduki beans','advocaat','agar-agar','ale','alfalfa sprouts','allspice','almond','almond essence'
'dab','daikon','damsons','dandelion','danish blue','dark chocolate','date','demerara sugar','demi-glace sauce','desiccated coconut','desiree','dige
'farfalle','fat','fennel','fennel seeds','fenugreek','feta','fettuccine','field mushroom','fig','fillet of beef','filo pastry','fish','fish roe','f
'habanero chillies','haddock','haggis','hake','halibut','halloumi','ham','hare','haricot beans','harissa','hazelnut','azelnut oil','heart','herbal

'sabudana','rava','pohe','lal mirchi masala','biryani masala','chicken masala','mutton masala','pudina','laung','kali mirch','choti elaichi','badi
]
```

This are the list of some of the many ingredients we have listed for the recommendation



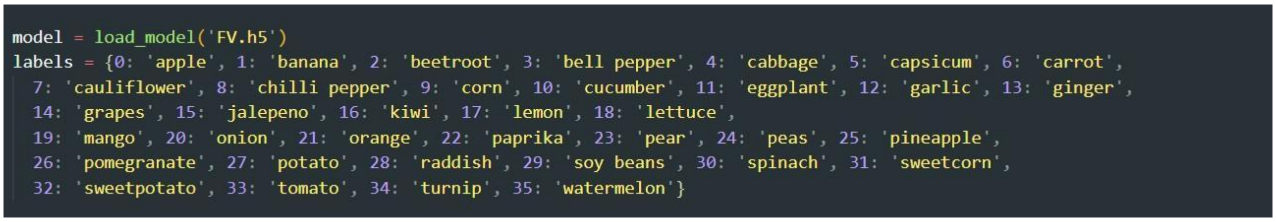
```
if prompt == "Custom":
    items = st.multiselect("Select ingredients", ingredients)
    items = ", ".join(items)
    st.write('Selected ingredients: ', items)

else:
    items = EXAMPLES[prompt]
    st.write("Selected ingredients: ", items)
```

Here if we select custom then we can select from the above list of ingredients else if we select any cuisine example we will get ingredients automatically for that cuisine.

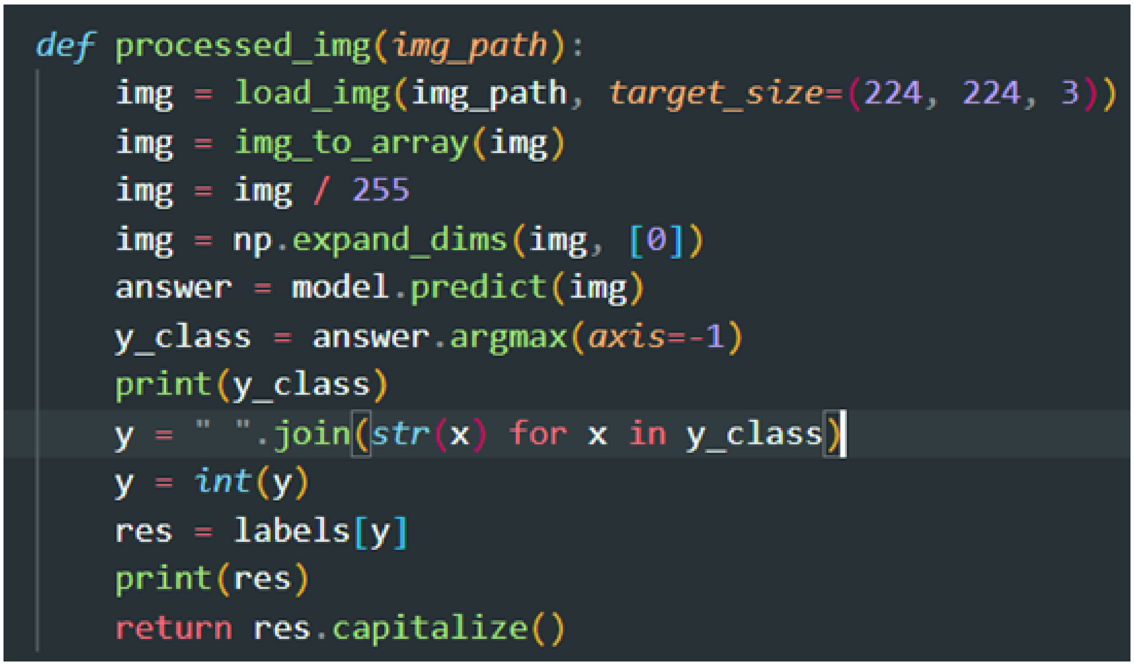
Module 4: Prediction of ingredient from image

We have also provided a feature in which if someone wants to upload the image of the ingredients then they can upload and our model will predict it and use that ingredient. Multiple images can be uploaded at a time. We trained this model on a list of ingredients with our dataset created and used CNN for training and prediction. CNN is convolutional neural networks which is used for image prediction, classification etc.

Code Snippet:



```
model = load_model('FV.h5')
labels = {0: 'apple', 1: 'banana', 2: 'beetroot', 3: 'bell pepper', 4: 'cabbage', 5: 'capsicum', 6: 'carrot',
  7: 'cauliflower', 8: 'chilli pepper', 9: 'corn', 10: 'cucumber', 11: 'eggplant', 12: 'garlic', 13: 'ginger',
  14: 'grapes', 15: 'jalepeno', 16: 'kiwi', 17: 'lemon', 18: 'lettuce',
  19: 'mango', 20: 'onion', 21: 'orange', 22: 'paprika', 23: 'pear', 24: 'peas', 25: 'pineapple',
  26: 'pomegranate', 27: 'potato', 28: 'raddish', 29: 'soy beans', 30: 'spinach', 31: 'sweetcorn',
  32: 'sweetpotato', 33: 'tomato', 34: 'turnip', 35: 'watermelon'}
```

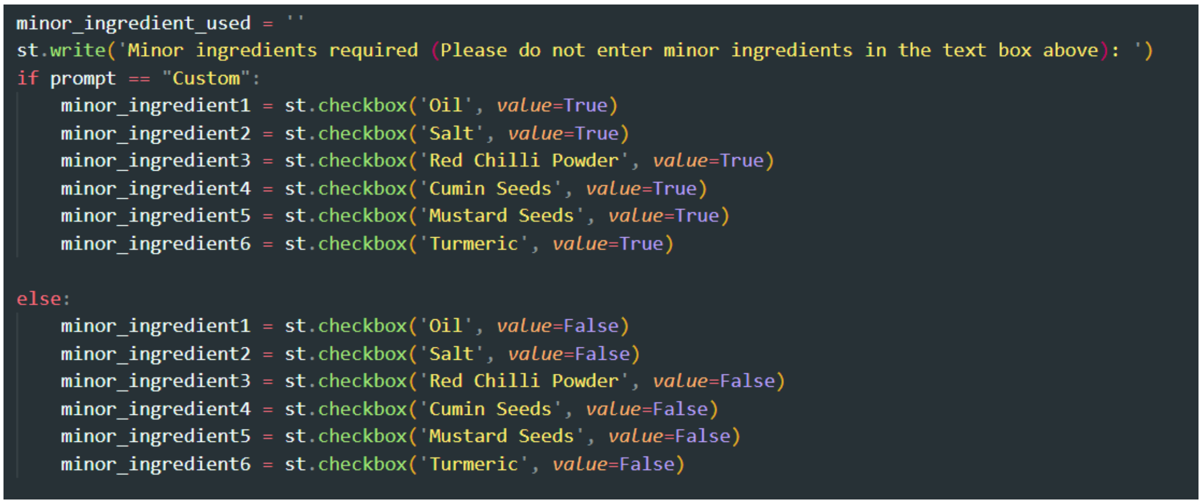


```
def processed_img(img_path):
    img = load_img(img_path, target_size=(224, 224, 3))
    img = img_to_array(img)
    img = img / 255
    img = np.expand_dims(img, [0])
    answer = model.predict(img)
    y_class = answer.argmax(axis=-1)
    print(y_class)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    print(res)
    return res.capitalize()
```

We load our model being trained and these are the labels and the ingredients we can predict.This function processes the image being taken as an input and converts it into an array and predicts the answer.Also we are joining all the predictions being done here. We have given a fixed size of the image. So any size image uploaded will get converted to the fixed size mentioned.

Module 5: Minor Ingredients Selection

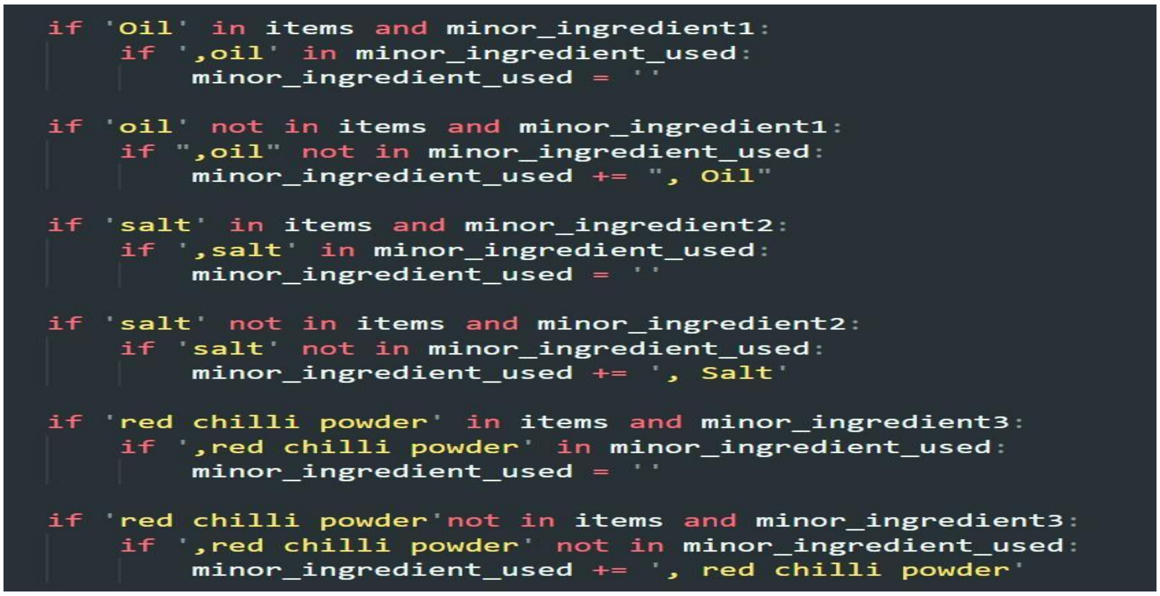
We provided a feature of minor ingredients prediction where most common ingredients used in many cuisines and dishes can be selected and they are joined with the major ingredients being selected.



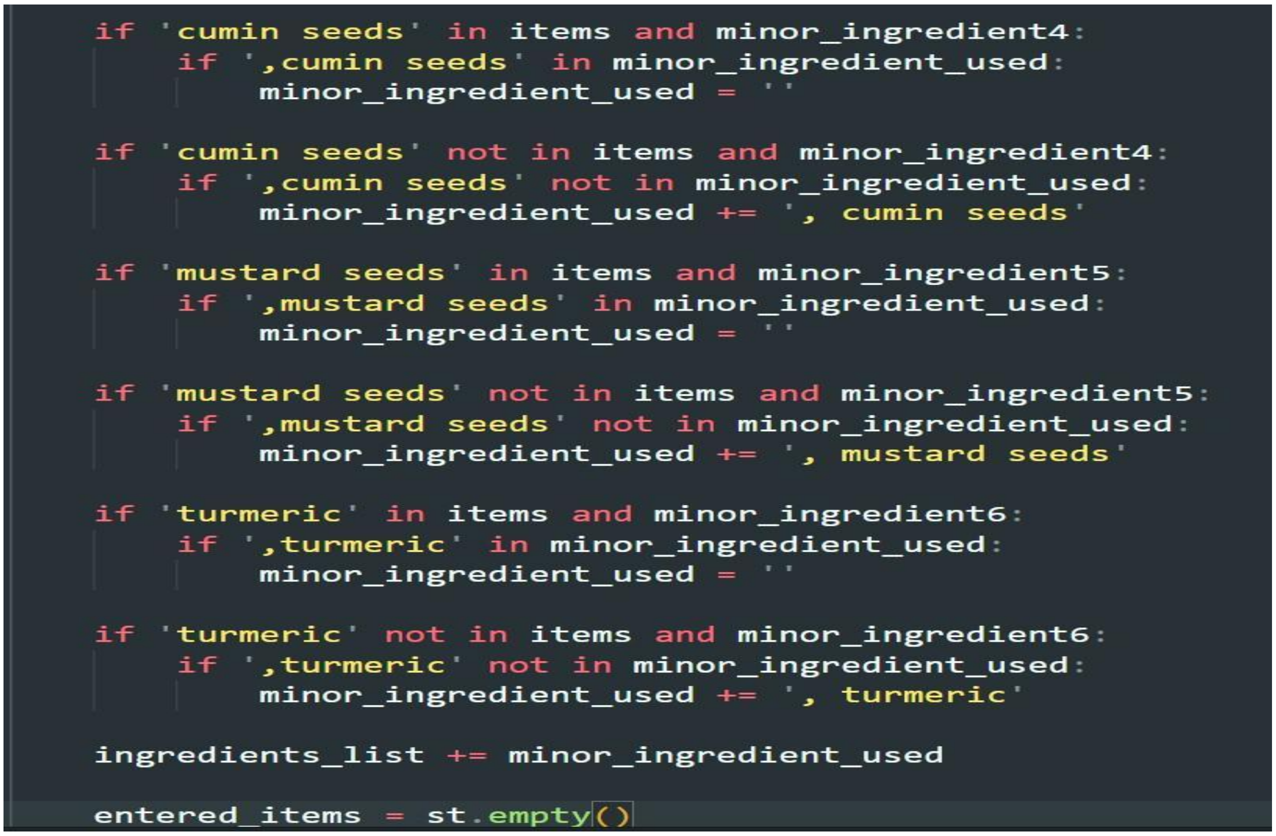
```
minor_ingredient_used = ''
st.write('Minor ingredients required (Please do not enter minor ingredients in the text box above): ')
if prompt == "Custom":
    minor_ingredient1 = st.checkbox('Oil', value=True)
    minor_ingredient2 = st.checkbox('Salt', value=True)
    minor_ingredient3 = st.checkbox('Red Chilli Powder', value=True)
    minor_ingredient4 = st.checkbox('Cumin Seeds', value=True)
    minor_ingredient5 = st.checkbox('Mustard Seeds', value=True)
    minor_ingredient6 = st.checkbox('Turmeric', value=True)

else:
    minor_ingredient1 = st.checkbox('Oil', value=False)
    minor_ingredient2 = st.checkbox('Salt', value=False)
    minor_ingredient3 = st.checkbox('Red Chilli Powder', value=False)
    minor_ingredient4 = st.checkbox('Cumin Seeds', value=False)
    minor_ingredient5 = st.checkbox('Mustard Seeds', value=False)
    minor_ingredient6 = st.checkbox('Turmeric', value=False)
```

Code Snippet:

Here the ingredients remain selected if the box is custom else they remain unselected. The common ingredients we have mentioned are oil, salt , Red chili powder, cumin seeds, mustard seeds, turmeric.



```
if 'Oil' in items and minor_ingredient1:
    if ',oil' in minor_ingredient_used:
        minor_ingredient_used = ''

if 'oil' not in items and minor_ingredient1:
    if ",oil" not in minor_ingredient_used:
        minor_ingredient_used += ", Oil"

if 'salt' in items and minor_ingredient2:
    if ',salt' in minor_ingredient_used:
        minor_ingredient_used = ''

if 'salt' not in items and minor_ingredient2:
    if 'salt' not in minor_ingredient_used:
        minor_ingredient_used += ', Salt'

if 'red chilli powder' in items and minor_ingredient3:
    if ',red chilli powder' in minor_ingredient_used:
        minor_ingredient_used = ''

if 'red chilli powder'not in items and minor_ingredient3:
    if ',red chilli powder' not in minor_ingredient_used:
        minor_ingredient_used += ', red chilli powder'
```



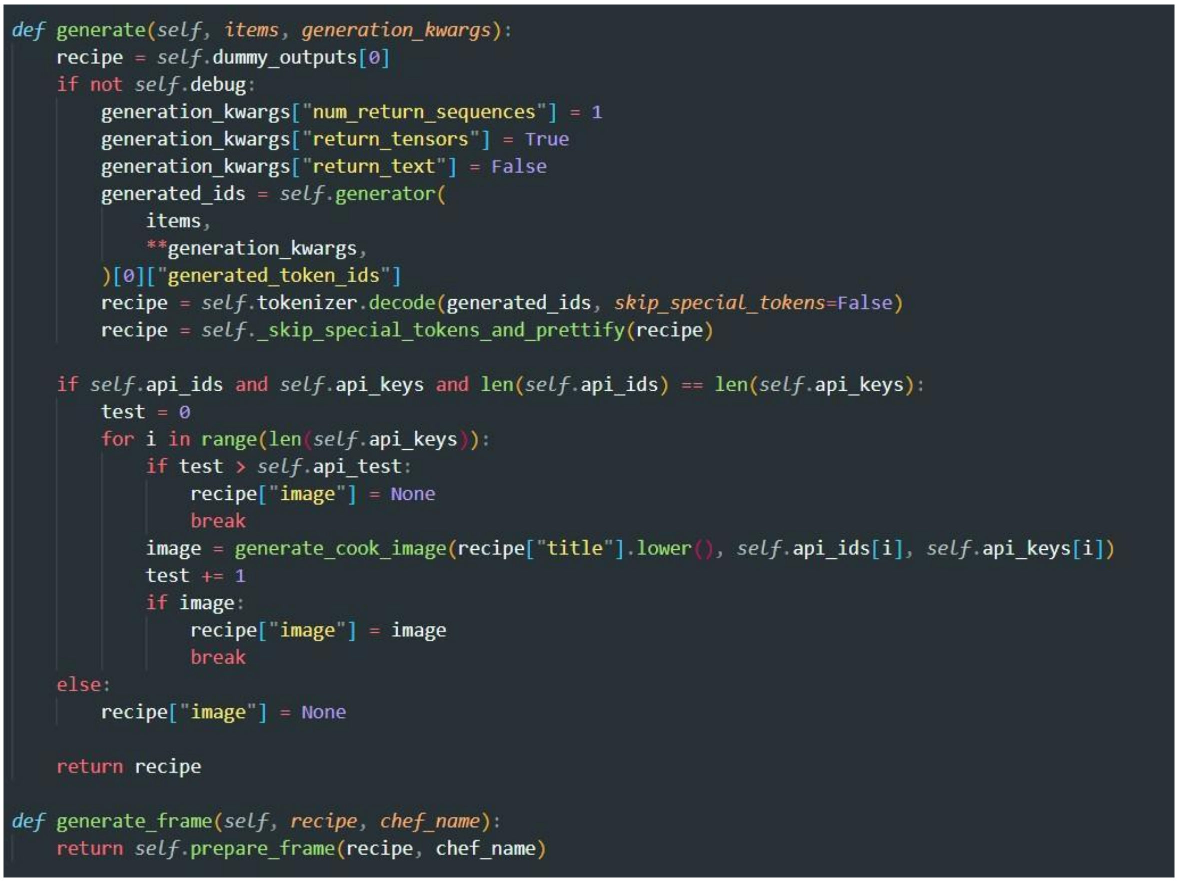
```
if 'cumin seeds' in items and minor_ingredient4:
    if ',cumin seeds' in minor_ingredient_used:
        minor_ingredient_used = ''

if 'cumin seeds' not in items and minor_ingredient4:
    if ',cumin seeds' not in minor_ingredient_used:
        minor_ingredient_used += ', cumin seeds'

if 'mustard seeds' in items and minor_ingredient5:
    if ',mustard seeds' in minor_ingredient_used:
        minor_ingredient_used = ''

if 'mustard seeds' not in items and minor_ingredient5:
    if ',mustard seeds' not in minor_ingredient_used:
        minor_ingredient_used += ', mustard seeds'

if 'turmeric' in items and minor_ingredient6:
    if ',turmeric' in minor_ingredient_used:
        minor_ingredient_used = ''

if 'turmeric' not in items and minor_ingredient6:
    if ',turmeric' not in minor_ingredient_used:
        minor_ingredient_used += ', turmeric'

ingredients_list += minor_ingredient_used

entered_items = st.empty()
```

These are the conditions written for the ingredients being selected. For one ingredient selected. For two ingredients selected and so on and finally they are added along with the major ingredients and the final ingredients list is being generated.

Module 6: Recipe Generation

After all the ingredients being uploaded from the image, minor ingredients selected and major ingredients written then the final list of ingredients is prepared and then send to the ML model for generating the recipe

Code Snippet:



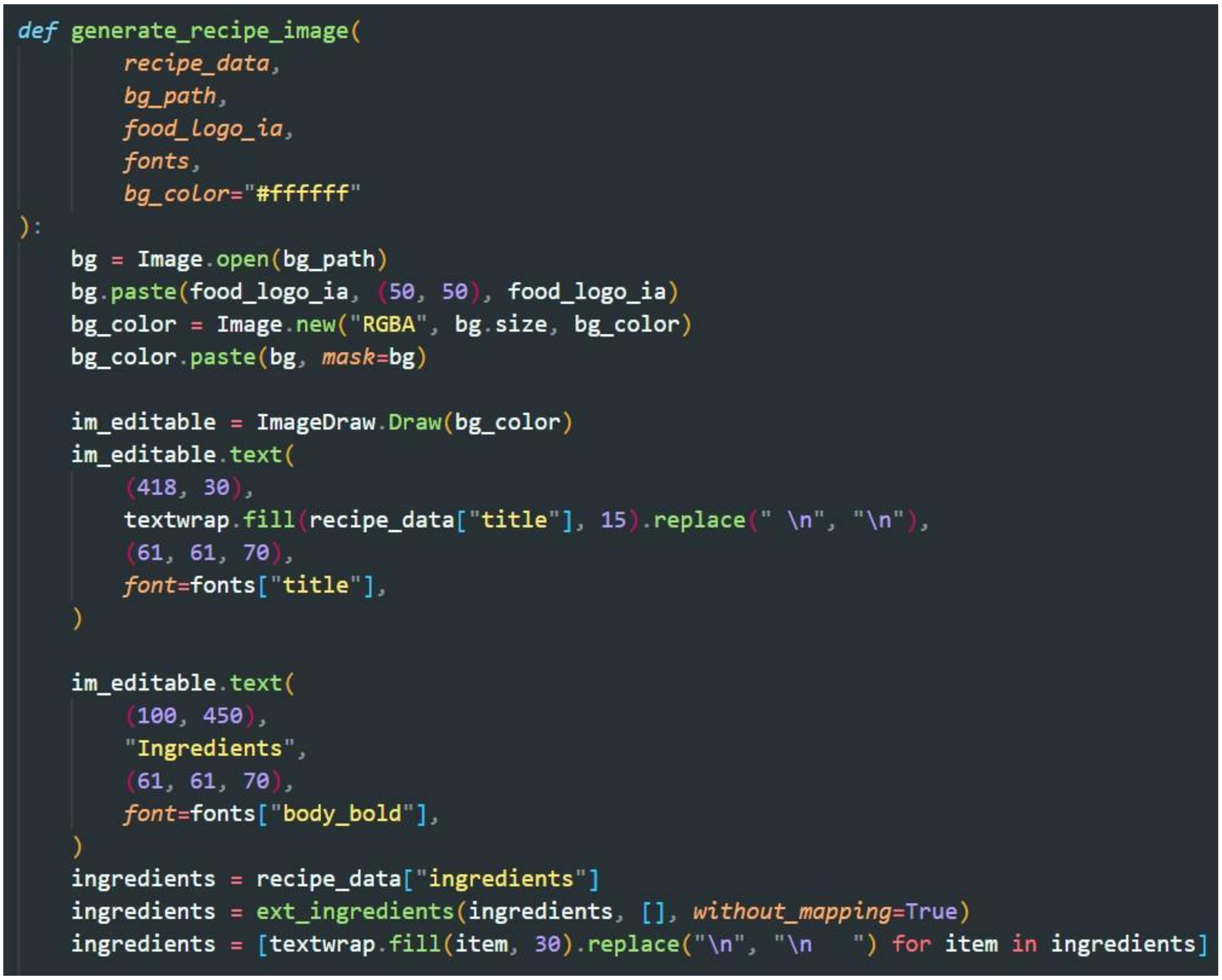
```python
def generate(self, items, generation_kwargs):
    recipe = self.dummy_outputs[0]
    if not self.debug:
        generation_kwargs["num_return_sequences"] = 1
        generation_kwargs["return_tensors"] = True
        generation_kwargs["return_text"] = False
        generated_ids = self.generator(
            items,
            **generation_kwargs,
        )[0]["generated_token_ids"]
        recipe = self.tokenizer.decode(generated_ids, skip_special_tokens=False)
        recipe = self._skip_special_tokens_and_prettify(recipe)

    if self.api_ids and self.api_keys and len(self.api_ids) == len(self.api_keys):
        test = 0
        for i in range(len(self.api_keys)):
            if test > self.api_test:
                recipe["image"] = None
                break
            image = generate_cook_image(recipe["title"].lower(), self.api_ids[i], self.api_keys[i])
            test += 1
            if image:
                recipe["image"] = image
                break
    else:
        recipe["image"] = None

    return recipe

def generate_frame(self, recipe, chef_name):
    return self.prepare_frame(recipe, chef_name)
```

The function takes two arguments: items, which appears to be a list of input text items, and generation_kwargs, which is a dictionary of keyword arguments that will be passed to the pipeline function when generating output. The function starts by setting the recipe variable to the first item in the dummy_outputs list. This appears to be a default recipe that is returned if the debug flag is set to True. If debug is False, the function modifies the generation_kwargs dictionary to ensure that only one output sequence is generated and that the output is returned in a format that can be decoded by the tokenizer. The function then generates a sequence of tokens using the pre-trained model specified by the generator attribute of the class. It decodes the generated tokens into text using the tokenizer decode function and formats the text using the _skip_special_tokens_and_prettify method. If API IDs and keys have been loaded and api_test is greater than 0, the function loops over each set of API credentials and attempts to generate a corresponding image for the recipe. The generate_cook_image function appears to be a custom function that takes the recipe title and API credentials as arguments and returns an image URL. The first successful image URL is assigned to the recipe["image"] attribute and the loop is terminated. If no image URLs are successfully generated, recipe["image"] is set to None. Finally, the function returns the recipe object, which contains a title, instructions, ingredients, and image (if available).

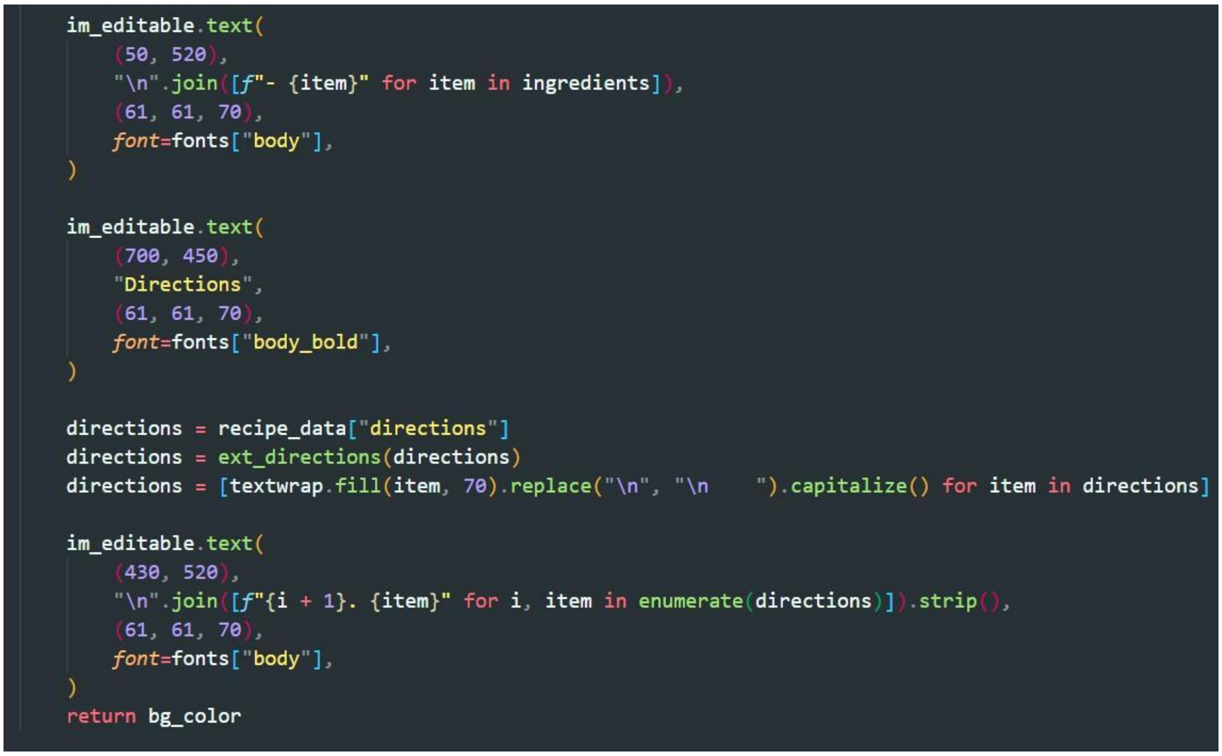
Module 7: Image Generation

Finally after the recipe gets generated, we then also make a downloadable image of the dish along with the ingredients and the recipe which can be shared with people around the world and new cuisines can be tried out.

Code Snippet:



```python
def generate_recipe_image(
        recipe_data,
        bg_path,
        food_logo_ia,
        fonts,
        bg_color="#ffffff"
):
    bg = Image.open(bg_path)
    bg.paste(food_logo_ia, (50, 50), food_logo_ia)
    bg_color = Image.new("RGBA", bg.size, bg_color)
    bg_color.paste(bg, mask=bg)

    im_editable = ImageDraw.Draw(bg_color)
    im_editable.text(
        (418, 30),
        textwrap.fill(recipe_data["title"], 15).replace(" \n", "\n"),
        (61, 61, 70),
        font=fonts["title"],
    )

    im_editable.text(
        (100, 450),
        "Ingredients",
        (61, 61, 70),
        font=fonts["body_bold"],
    )
    ingredients = recipe_data["ingredients"]
    ingredients = ext_ingredients(ingredients, [], without_mapping=True)
    ingredients = [textwrap.fill(item, 30).replace("\n", "\n   ") for item in ingredients]
```



```
    im_editable.text(
        (50, 520),
        "\n".join([f"- {item}" for item in ingredients]),
        (61, 61, 70),
        font=fonts["body"],
    )

    im_editable.text(
        (700, 450),
        "Directions",
        (61, 61, 70),
        font=fonts["body_bold"],
    )

    directions = recipe_data["directions"]
    directions = ext_directions(directions)
    directions = [textwrap.fill(item, 70).replace("\n", "\n    ").capitalize() for item in directions]

    im_editable.text(
        (430, 520),
        "\n".join([f"{i + 1}. {item}" for i, item in enumerate(directions)]).strip(),
        (61, 61, 70),
        font=fonts["body"],
    )
    return bg_color
```

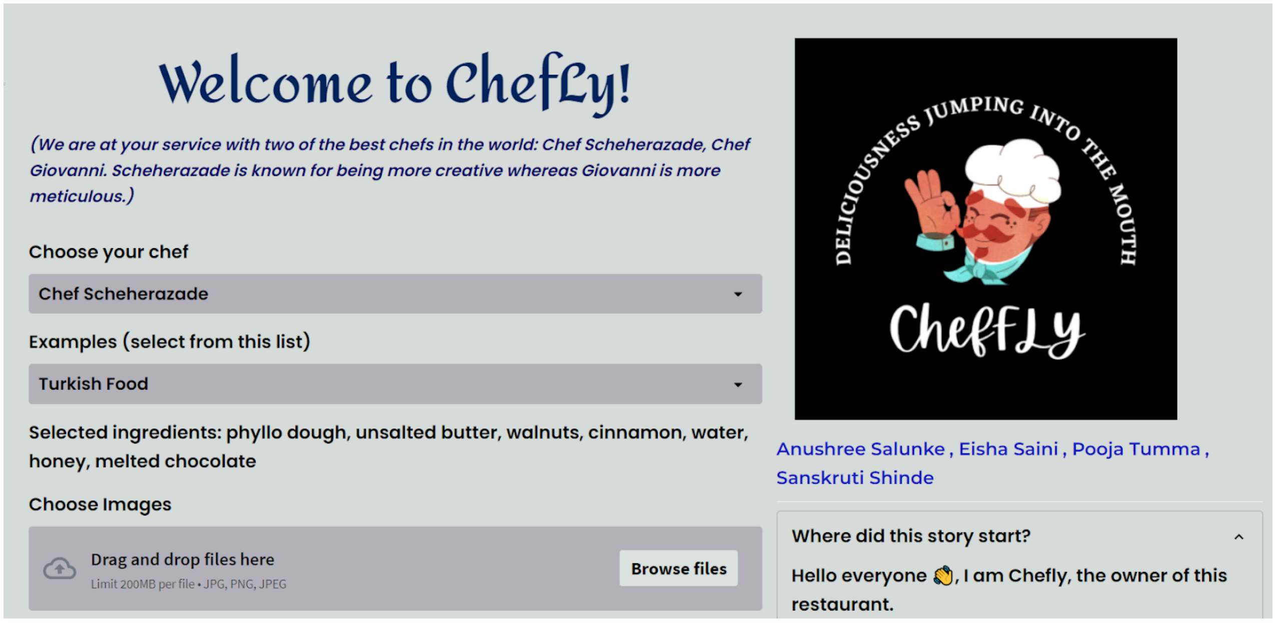
This is a function called generate_recipe_image that takes in various inputs to create an image of a recipe.

<u>The inputs are:</u>

recipe_data: a dictionary containing the title, ingredients, and directions of the recipe.
bg_path: a file path to the background image for the recipe image.
food_logo_ia: an image of a food logo to be placed in the recipe image.
fonts: a dictionary containing various fonts for the different elements of the recipe image.
bg_color: a background color for the recipe image (default is white) The function first opens the background image and pastes the food logo image onto it. Then, it adds the recipe title to the image using a specified font. Next, it adds the ingredients to the image by first extracting the ingredients from the recipe data, formatting them, and then adding them to the image using a specified font. Finally, the function adds the recipe directions in a similar manner to how it added the ingredientsThe function returns the final recipe image

Chapter 8

Result



Welcome to ChefLy!

(We are at your service with two of the best chefs in the world: Chef Scheherazade, Chef Giovanni. Scheherazade is known for being more creative whereas Giovanni is more meticulous.)

Choose your chef

Chef Scheherazade

Examples (select from this list)

Turkish Food

Selected ingredients: phyllo dough, unsalted butter, walnuts, cinnamon, water, honey, melted chocolate

Choose Images

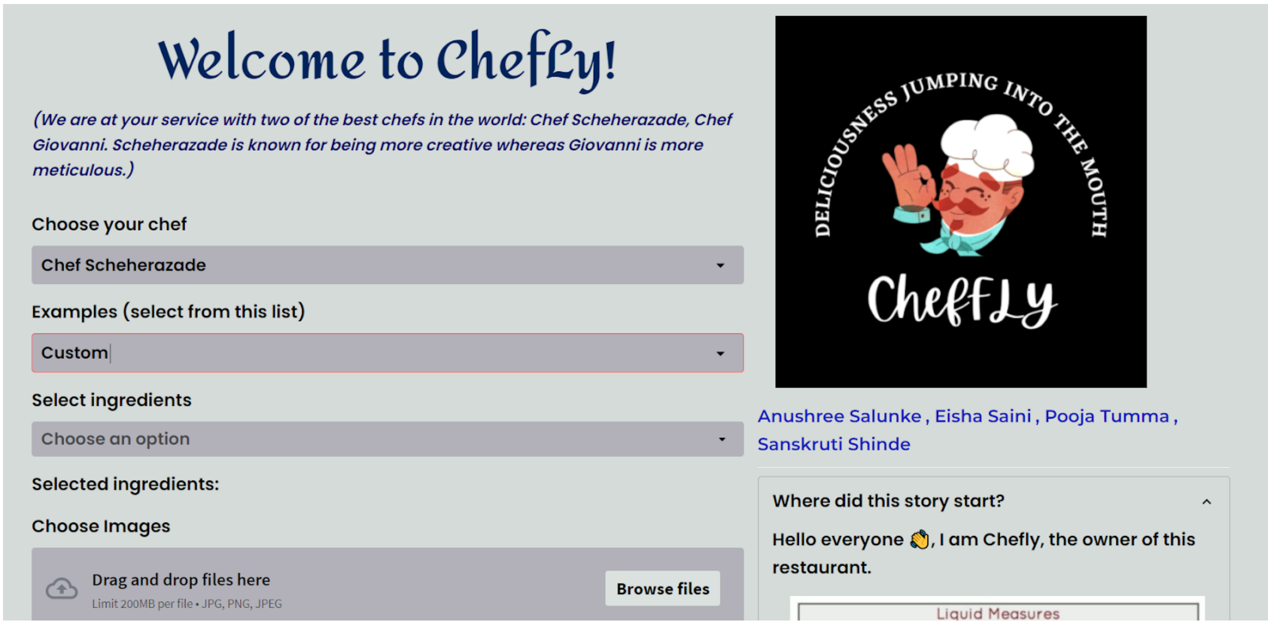
Drag and drop files here
Limit 200MB per file • JPG, PNG, JPEG
Browse files
DELICIOUSNESS JUMPING INTO THE MOUTH
ChefLy

Anushree Salunke , Eisha Saini , Pooja Tumma , Sanskruti Shinde

Where did this story start?

Hello everyone 👋, I am Chefly, the owner of this restaurant.

In the above output we first select the chef and then we can select the food examples given in the drop down box.After selecting the examples it displays the ingredients which are to be used for generating the recipes.



Welcome to ChefLy!

(We are at your service with two of the best chefs in the world: Chef Scheherazade, Chef Giovanni. Scheherazade is known for being more creative whereas Giovanni is more meticulous.)

Choose your chef: Chef Scheherazade

Examples (select from this list): Custom

Select ingredients: Choose an option

Selected ingredients:

Choose Images

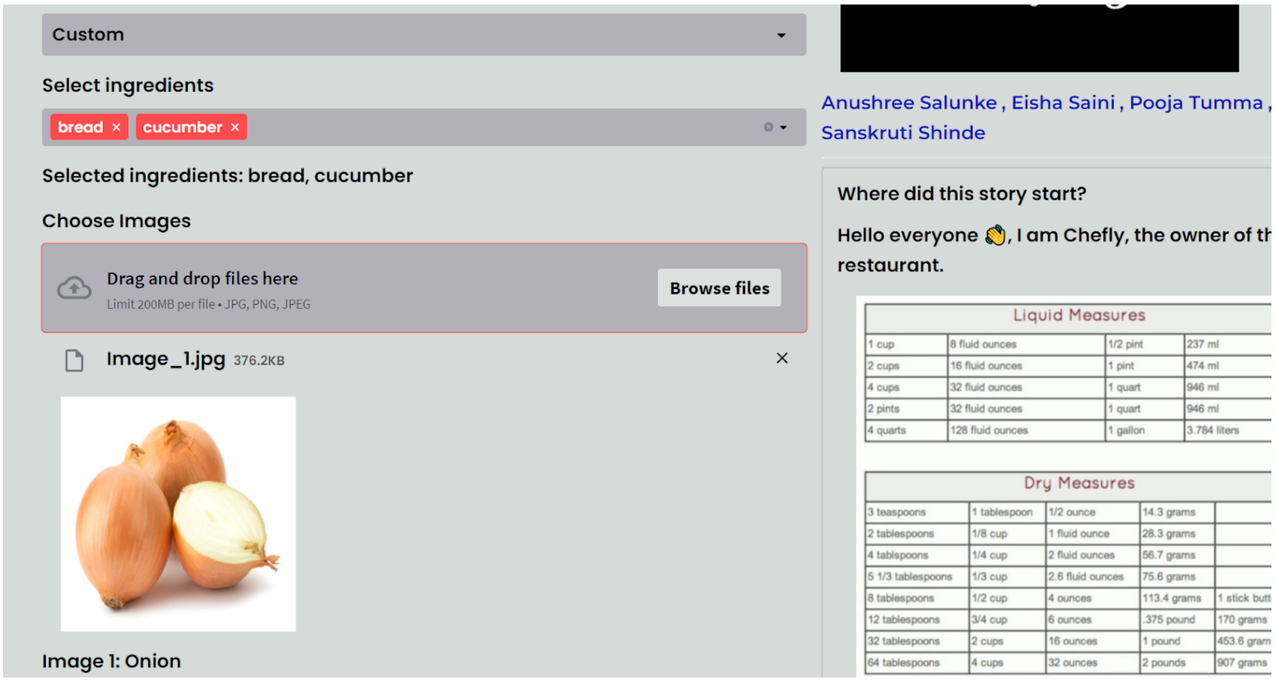
Drag and drop files here
Limit 200MB per file • JPG, PNG, JPEG
Browse files

Anushree Salunke , Eisha Saini , Pooja Tumma , Sanskruti Shinde

Where did this story start?

Hello everyone 👋, I am Chefly, the owner of this restaurant.

Liquid Measures



Custom

Select ingredients: bread ×, cucumber ×

Selected ingredients: bread, cucumber

Choose Images

Drag and drop files here
Limit 200MB per file • JPG, PNG, JPEG
Browse files

Image_1.jpg 376.2KB

Image 1: Onion

Anushree Salunke , Eisha Saini , Pooja Tumma , Sanskruti Shinde

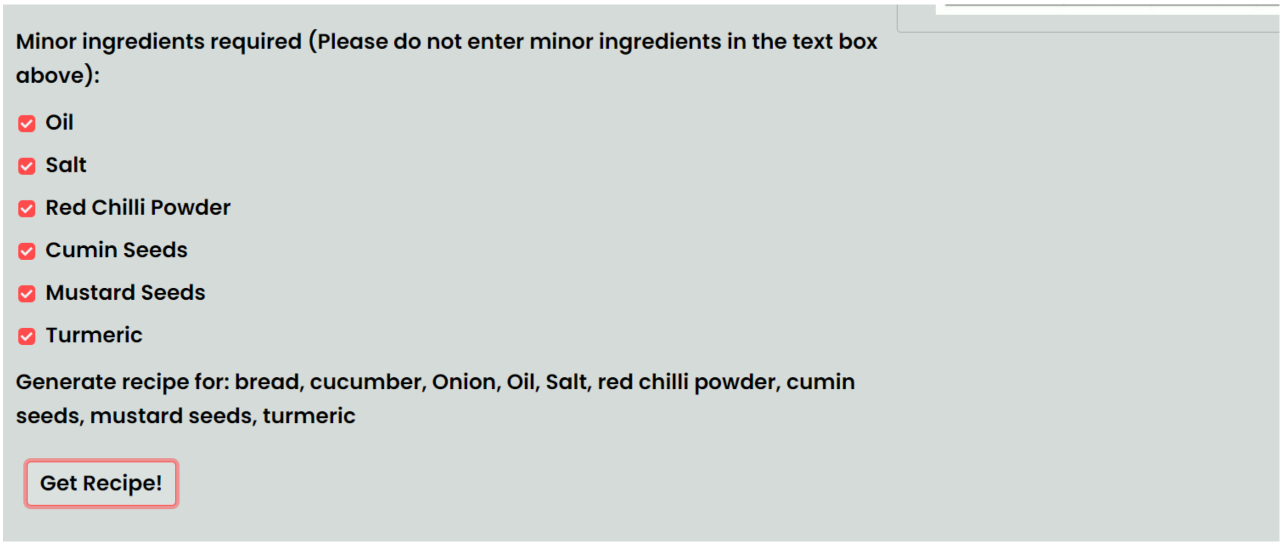
Where did this story start?

Hello everyone 👋, I am Chefly, the owner of th restaurant.

Liquid Measures			
1 cup	8 fluid ounces	1/2 pint	237 ml
2 cups	16 fluid ounces	1 pint	474 ml
4 cups	32 fluid ounces	1 quart	946 ml
2 pints	32 fluid ounces	1 quart	946 ml
4 quarts	128 fluid ounces	1 gallon	3.784 liters

Dry Measures				
3 teaspoons	1 tablespoon	1/2 ounce	14.3 grams	
2 tablespoons	1/8 cup	1 fluid ounce	28.3 grams	
4 tablspoons	1/4 cup	2 fluid ounces	56.7 grams	
5 1/3 tablespoons	1/3 cup	2.6 fluid ounces	75.6 grams	
8 tablespoons	1/2 cup	4 ounces	113.4 grams	1 stick butt
12 tablespoons	3/4 cup	6 ounces	.375 pound	170 grams
32 tablespoons	2 cups	16 ounces	1 pound	453.6 gram
64 tablespoons	4 cups	32 ounces	2 pounds	907 grams

In the above two outputs we first select the chef and then we can select the custom option given in the drop down box.After selecting the custom option it provides us an ingredients list box where we can select all the ingredients and also recommends it. Also we can upload the image of any ingredient and it will predict and use it further.

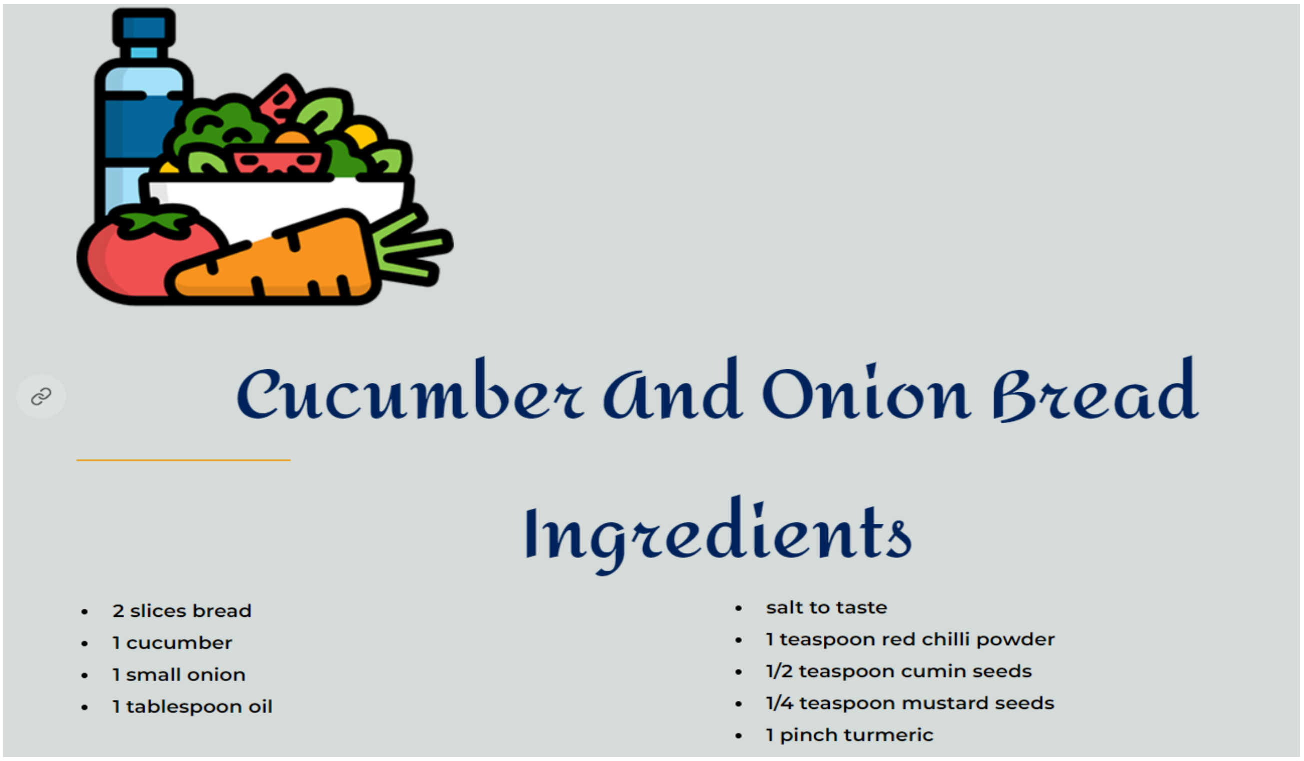


Minor ingredients required (Please do not enter minor ingredients in the text box above):

- ✓ Oil
- ✓ Salt
- ✓ Red Chilli Powder
- ✓ Cumin Seeds
- ✓ Mustard Seeds
- ✓ Turmeric

Generate recipe for: bread, cucumber, Onion, Oil, Salt, red chilli powder, cumin seeds, mustard seeds, turmeric

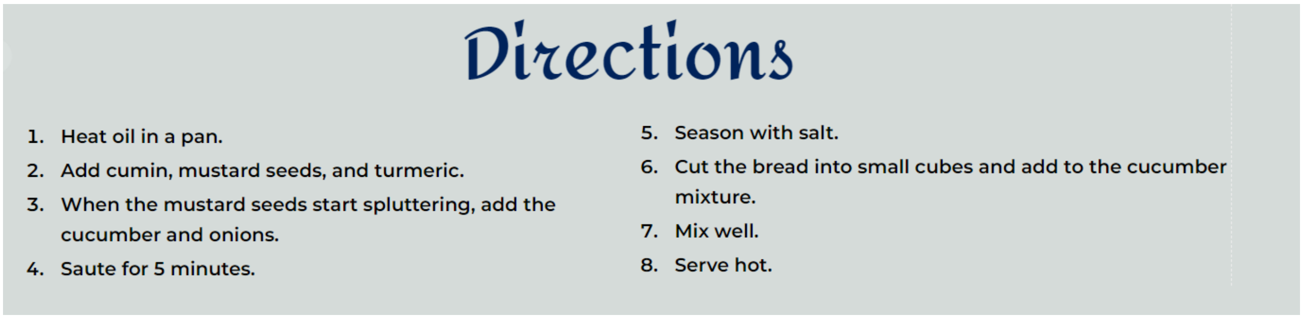
Here we are selecting the minor ingredients which are required for preparing the dish. Finally all the ingredients selected are put together separated by commas and considered together for generating the recipe after clicking the Get Recipe! button.



Cucumber And Onion Bread

Ingredients

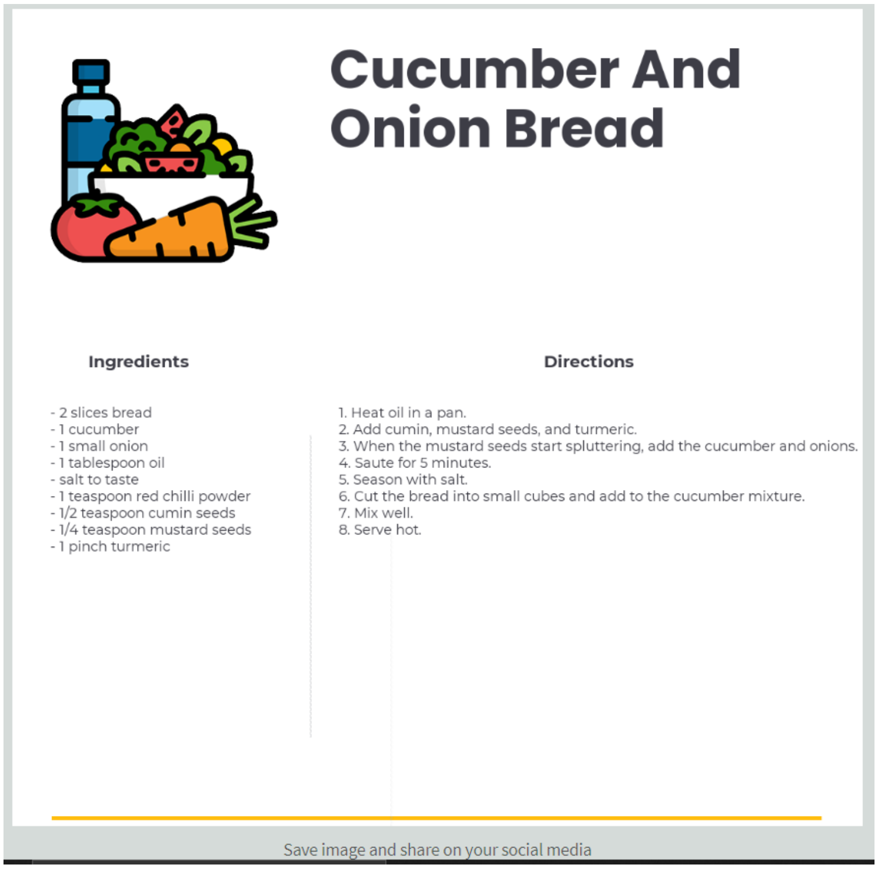
- 2 slices bread
- 1 cucumber
- 1 small onion
- 1 tablespoon oil
- salt to taste
- 1 teaspoon red chilli powder
- 1/2 teaspoon cumin seeds
- 1/4 teaspoon mustard seeds
- 1 pinch turmeric



Directions

1. Heat oil in a pan.
2. Add cumin, mustard seeds, and turmeric.
3. When the mustard seeds start spluttering, add the cucumber and onions.
4. Saute for 5 minutes.
5. Season with salt.
6. Cut the bread into small cubes and add to the cucumber mixture.
7. Mix well.
8. Serve hot.

The above two outputs give us the name of the dish, the ingredients required for preparing the dish and the steps of preparing the dish.



Cucumber And Onion Bread

Ingredients

- 2 slices bread
- 1 cucumber
- 1 small onion
- 1 tablespoon oil
- salt to taste
- 1 teaspoon red chilli powder
- 1/2 teaspoon cumin seeds
- 1/4 teaspoon mustard seeds
- 1 pinch turmeric

Directions

1. Heat oil in a pan.
2. Add cumin, mustard seeds, and turmeric.
3. When the mustard seeds start spluttering, add the cucumber and onions.
4. Saute for 5 minutes.
5. Season with salt.
6. Cut the bread into small cubes and add to the cucumber mixture.
7. Mix well.
8. Serve hot.

Save image and share on your social media

The generated recipe is then provided in the form of an image which can be shared on social media after being downloaded.

The output of our project is being displayed above. In this project we select the chef we want to generate the recipe based on the ingredients. Then we have options to select the foods of different countries which will gives us the ingredients and generates the recipe from that list of ingredients.If we select the custom option then we will receive a select ingredients box which will ask us to select the ingredients.While typing the ingredients it gives us the recommendations of the ingredients which start from that letter. We have also provided a file uploader where we can upload the image of the ingredients and then our website will predict the ingredients. We can upload multiple images and then that can be predicted. We have also provided the minor ingredients which are the most common ingredients used in the dishes. We can select those ingredients such as salt, oil, turmeric etc. After that a final list of ingredients is being generated which is used for generating the recipe. After we click on the Get Recipe button we will get the name of the dish along with the quantity of ingredients and the steps of the ingredients. Finally the generated recipe can be downloaded as an image and then shared with everyone on social media.

Chapter 9

Conclusion

In conclusion, ChefFly is a great tool for those who are looking to cook new and exciting meals but may not have the time or creativity to come up with recipes on their own. With the help of machine learning algorithms, it can analyze a wide range of ingredients and suggest recipes that are tailored to individual preferences and dietary restrictions. We were able to achieve our scope and goals with which we started. Web crawlers and transformer dataset were used for recipes which were created. Data preprocessing was necessary to have the right kind of datasets. However, it's important to keep in mind that recipe generators are not perfect and may not always produce the most accurate or delicious recipes. Additionally, they may not take into account cultural or regional differences in cuisine, and may not always provide detailed instructions for more complex dishes. Overall, ChefFly can be a useful resource for home cooks, but should be used in conjunction with personal knowledge and experience in the kitchen for the best results.

Chapter 10

References

[1] Chun-Yuen Teng, Yu-Ru Lin, and Lada A. Adamic. 2012. Recipe recommendation using ingredient networks. Proceedings of the 4th Annual ACM Web Science Conference (WebSci '12). Association for Computing Machinery, New York, NY, USA, 298–307. https://dl.acm.org/doi/10.1145/2380718.2380757

[2] Y. Pan, Q. Xu and Y. Li, "Food Recipe Alternation and Generation with Natural Language Processing Techniques," 2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW), Dallas, TX, USA, 2020, pp. 94-97,
https://ieeexplore.ieee.org/document/9094119

[3] Maryam Mohammadi, Somaye Arabi Naree, and Mahsa Lati. 2020. User-item content awareness in matrix factorization based collaborative recommender systems. Intell. Data Anal. 24, 3 (2020), 723–739.
https://doi.org/10.3233/IDA-194599

[4]Zakraoui, J., Saleh, M., Al-Maadeed, S. et al. Improving text-to-image generation with object layout guidance. Multimed Tools Appl 80, 27423–27443 (2021).
https://doi.org/10.1007/s11042-021-11038-0

[5] Developer Documentation for Web Development
HTTPS://DEVDOCS.IO

Acknowledgement

We have great pleasure in presenting the mini project report on **“ChefFly- Recipe Generator**”. We take this opportunity to express our sincere thanks towards our guide **Prof. Deepak Khachane**,Department of Computer Engineering, APSIT Thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of the project.

We thank **Prof. Sachin Malave, Head of Department**, Computer Engineering, APSIT for his encouragement during the progress meeting and providing guidelines to write this report.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

SR No	Name	Moodle Id
1.	Anushree Salunke	20102179
2.	Eisha Saini	20102025
3.	Pooja Tumma	20102126
4.	Sanskruti Shinde	20202018