



# UNIVERSITY OF SIALKOT

---

**Faculty of Computing & IT University of Sialkot**

**“COOKMANIA”**

**Session: BS-Information Technology**

**Spring 2020-2023**

**Submitted By:**

**Hasnaat Abdullah**

**20201003-037**

**Amaar Ali**

**20201003-023**

---

**Faculty of Computing & IT University of Sialkot**

<b>PROJECT NO</b>	<b>06</b>
-------------------	-----------

<b>Number of Group members</b>	<b>2</b>
--------------------------------	----------

<b>TITLE</b>	<b>COOKMANIA</b>
--------------	------------------

<b>SUPERVISOR NAME</b>	<b>Ma'am Wajeeha Fareed</b>
------------------------	-----------------------------

<b>Sr. No.</b>	<b>Name</b>	<b>Roll Number</b>	<b>Email Address</b>
1	Ammar Ali	20201003-023	<a href="mailto:20201003-023@uskt.edu.pk">20201003-023@uskt.edu.pk</a>
2	Hasnaat Abdullah	20201003-037	<a href="mailto:20201003-037@uskt.edu.pk">20201003-037@uskt.edu.pk</a>

#### *CHECKLIST*

Number of pages in this report

We have enclosed the soft copy of this document along with the codes and scripts created by myself/ourselves

**YES/NO**

My/Our supervisor and co-supervisor have attested to the attached document  
**YES/NO**

**I/We confirm that this project is free from any type of plagiarism**

**and misuse of copyrighted material**

**YES/NO**

#### **Members Signatures**

Ammar Ali\_\_\_\_\_

Hasnaat Abdullah\_\_\_\_\_

## **STATEMENT OF SUBMISSION**

This is certified that **Hasnaat Abdullah** Roll No. **20201003-037** and **Ammar Ali** Roll No. **20201003-023** and have successfully completed the final year project “**COOKMANIA**” at the Department of Information Technology, University of Sialkot, to fulfill the requirement of the degree **BS-IT** in **Information Technology**.

---

**Project Supervisor**

---

**Project Management**

Faculty of CS & IT-USKT

---

**Head of Department**

### **Acknowledgement**

We truly acknowledge the cooperation and assistance by the **Dr.Azeem**, Head of the Department of Information Technology, University of Sialkot. He has been a constant source of guidance throughout the course of this project. We would also thank **Miss, Wajeaha Fareed** for her assistance and guidance throughout this project. We are thankful to our families whose immense support emotionally and financially led us to complete our project.

- |                     |              |
|---------------------|--------------|
| 1. Hasnaat Abdullah | 20201003-037 |
| 2. Ammar Ali        | 20201003-023 |

Date: \_\_\_\_\_

## **DECLARATION**

**Ammar Ali S/O Imran Iqbal 20201003-023** and **Hasnaat Abdullah S/O Shahid Nawaz 20201003-037** are students of Bachelor of Information Technology, Department of Computing & Information Technology, University of Sialkot, Pakistan, hereby solemnly declare that the data quoted in this report titled “**COOKMANIA**” is based on our original work, and has not yet been submitted or published elsewhere.

Ammar Ali  
20201003-023  
15-02-2024

Hasnaat Abdullah  
20201003-037  
15-02-2024

## CERTIFICATIONS

I certify that the project titled “**COOKMANIA**” is under my supervision with students **Ammar Ali Roll# 20201003-023** and **Hasnaat Abdullah Roll# 20201003-037** of Computing & Information Technology, University of Sialkot, Pakistan.

**Supervisor’s Name:** Wajeeha Fareed  
Department of Information Technology  
Faculty of Computing & Information Technology  
University of Sialkot, Punjab, Pakistan.  
Email: [wajeeha.fareed@uskt.edu.pk](mailto:wajeeha.fareed@uskt.edu.pk)  
Date: 15-02-2024



## ABSTRACT

A mobile application termed “**COOKMANIA**” utilizes Java, Python, XML, SQLite, and Flask. Currently, food waste is a global concern, a problem that arises mainly at the consumption level and generates environmental, economic, and social impacts. One way to reduce the food waste problem is to use the food we already have at home. However, this causes another concern, which is what to cook with certain foods. Sometimes we do not know what recipes can be made. Knowing which ingredients can be mixed and how to mix them can be a difficult task for a beginner cook, so selecting the right ingredients for a recipe is essential. Therefore, it is proposed to develop a recipe recommendation system through food ingredients. Presently, the system is a mobile application that recommends recipes containing the recognized ingredient. Due to the fast growth of the technology and the environment we can see the people take help in daily life. For the betterment of their cooking skills, they try every day new dishes and try to make them better and tasty. For this in old times people buy the recipes books called ‘**Magazine**’ and there we can see the recipes are written for the help of people. But due to the time moving fast the recipes, system totally come digitally. Now we can see this in our mobile phone but in our project, we can get the recommendation from our responding app according to our ingredients and this is very new and unique idea that can be possible with the help of Machine learning Algorithm that are used in it.



## Table of Contents

CHAPTER 1 INTRODUCTION .....	3
1.1 Introduction .....	3
1.2 Problem Statement .....	3
1.3 Objectives:.....	4
1.4 Project Scope .....	4
1.5 Summary .....	5
CHAPTER 2 LITERATURE REVIEW AND SYSTEM SPECIFICATIONS .....	6
2.1 Literature Review: .....	6
2.1.1 User centric design and experience: .....	7
2.1.2 Minimizing Food Waste through Ingredient.....	7
2.1.3 Machine Learning for Enhanced Recommendations:.....	7
CHAPTER 3: DESIGN AND METHODOLOGY .....	8
3.1 Recommendation model:.....	8
3.2 Methodology.....	8
3.3 Tools .....	8
3.3.1 Software/IDE: .....	8
3.3.2 Android Studio.....	9
3.3.3 Pycharm.....	9
3.3.4 Postman .....	9
3.3.5 SQLite .....	9
3.3.6 XML.....	9
3.3.7 JAVA.....	10
3.3.8 Flask:.....	10
3.3.9 Google collab.....	10
3.3.10 Python .....	10
3.3.10 Ngrok:.....	10
3.4 SYSTEM MODEL.....	11
3.5 Main Modules: .....	11
3. 5.1 User: .....	11
3.6 .....	11
3.7 Architecture Diagram.....	12
3.7 Flow Diagram .....	13
CHAPTER 4 RESULTS.....	14

4.1 Recommendation model .....	14
4.2 Start page .....	15
4.2.1 Login .....	15
4.2.2 Signup.....	16
4.2.3 Select ingredients no.....	16
4.2.4 Three ingredients .....	17
4.2.5 five ingredients.....	17
4.2.6 Eight ingredients:.....	18
4.2.7 Recipe results: .....	18
CHAPTER 5 CONCLUSIONS AND FUTURE WORK .....	19
5.1 Conclusions:.....	19
5.2 Enhanced Machine Learning Model.....	19
5.2.1 Saving Recommended recipes.....	19
5.2.2 Multi-language recipes:.....	19
5.2.3 Adding images of ingredients: .....	19
5.2.4 Adding recipes from different countries .....	19
REFERENCES .....	20
APPENDIX A.....	22

## List of Figures

FIGURE 3.1 system model.....	-17
FIGURE 3.2 use case.....	18
FIGURE 3.3 Architecture diagram.....	19
FIGURE 3.4 Flowchart diagram.....	20
FIGURE 4.1 Accuracy of the model.....	17
FIGURE 4.2 Result of the model.....	18
FIGURE 4.3 Start page of the app.....	19
FIGURE 4.3.1 login page of the app.....	20
FIGURE 4.3.2 Signup page of the app.....	19
FIGURE 4.3.3 select ingredients page of the app.....	20
FIGURE 4.3.4 three ingredients page.....	19
FIGURE 4.3.5 five ingredients page.....	20
FIGURE 4.3.6 eights ingredients page.....	20
FIGURE 4.3.7 results page.....	20

# **CHAPTER 1 INTRODUCTION**

## **1.1 Introduction:**

Currently, it is possible to find several websites or web applications that have user profiles for their use. These websites or web applications use a recommendation system that is based on the information given by users in their initial registration, such as preferences or tastes. In the culinary industry, there are several web apps and websites that are used to find recipes based on keywords such as type of cuisine, or culture. These applications attend to the needs and interests of their users but fail to identify restrictions, such as the maximum limit on the number of ingredients. In such cases, users end up missing ingredients or replacing them with similar ones. To help users avoid such adjustments, ingredient recognition can be used to identify the ingredients that the user has at his or her disposal at that moment and, through that ingredient or those ingredients, recommend various recipes. The main objective of the recipe recommendation system through the identification of ingredients is to help users who wish to cook with the ingredients that they have available at home to avoid food waste

The recipe system is very well known in now a day if we want to see any kind of recipe, we can easily get from the google or any recipe app which will help you and even guide us from start to end how can we make our dish.

In our project we are also doing the same thing but we make an innovation in it so that it helps more often for that we have the recommendation system addition in it it is a very unique and innovative idea. It is not expensive even it is not considering any cost [1]. So, in this system we add some points and add more data about it. So, the app can work like this we can find out any kind of recipe according to ingredients and if we have not completed our ingredients then the recommendation system will work for us [2]. The main thing that we must done is to write the name of that ingredients that we are have in our home at that time and after clicking on search the system will automatically recommend us the recipe according to our ingredients [3].

The project integrates various technologies to create a robust and efficient system. The frontend of the mobile application is built in android studio using XML, for providing an intuitive and interactive user interface. The backend is developed using java and Flask, which enable seamless communication between the different components of the system. Data storage and retrieval are managed using SQLite, ensuring efficient and secure management of user and recommend data. The system consists of three key panels: user, ingredients, and recipes. The user panel allows users to register, log in, and write the ingredients. Upon the ingredients, the machine learning model analyzes it and provides detailed recipes about the ingredients through API.

## **1.2 Problem Statement:**

There are many kinds of apps that are already exist in the market and the people are already used these apps. Mostly apps are very common around the people. And the market is already set the very high hopes in these apps and they provide good type of result. We are going to target this segment to provide the good service. There is the different between already exist application and our application which is that our application will recommend the recipe based on ingredients which we have and this is the major task because before our application we haven't find any application which is similar to us. Those application that are already in the market just gives us the recipes and

cuisine but not ingredient-based recipes so that it is too difficult to fetch information regarding for the recipes according to our ingredients from those app[4].

Another problem is that the current landscape of recipe-related websites and applications caters to users' preferences and tastes, there is a significant gap in addressing the issue of ingredient restrictions and availability. Many existing platforms fail to recognize users' limitations, such as the maximum ingredient count or the absence of specific items, leading to users either missing ingredients or substituting them. This often results in a less-than-ideal cooking experience and contributes to food waste. To address this gap, our innovative recipe recommendation system aims to leverage ingredient recognition technology. The primary objective is to empower users to efficiently discover and prepare recipes based on the ingredients available to them at home, ultimately minimizing food waste and enhancing the overall cooking experience. The system combines user-friendly interfaces, machine learning models, and advanced recommendation algorithms to provide personalized and resourceful recipe suggestions tailored to users' ingredient inventory [5].

### **1.3 Objectives:**

The cook mania has been developed with the following objectives in mind, aiming to address the limitations of the existing system and enhance the overall process of recipe recommendation [6]:

- The objective of our project is to design according the people need and help them in mean time and useful for the maximum numbers of people.
- Our project work under the new technologies and train by these new technologies.
- Basically, it can give the recipes that we need according to our ingredients.
- For the betterment result of our project, we must store the large scale of recipes data in the dataset that we are going to use in it [7].
- The focus of our is the recommendation that he can recommend or give recipe option according to our ingredients that we have in our home.
- The application offers a user-friendly interface that is easy to navigate for user. Intuitive feature is ingredients entering that ensure a smooth user experience, encouraging active participation and collaboration to get various recipes.

### **1.4 Project Scope:**

The system focuses on providing an intuitive and user-friendly mobile-application that facilitates the user-friendly interaction. The following areas fall within the project scope:

- Our project main scope is (Recommendation System) according what we have.
- It is used to try the new dishes[8].
- It helps us to make the good and healthy food base on our ingredients.
- It also works on what we have in hand (ingredients) like Meat, Chicken, Rice ETC.
- It provides us maximum numbers of dishes according to our ingredients.
- Bit change of ingredients change the recipes.
- we will finish that concept which is handed Magazine for recipe and take them into totally digital [9].
- The system is developed using a combination of technologies, including Java, Python,

XML, SQLite, and Flask. These technologies facilitate the development of a user-friendly and interactive mobile application that can be accessed from various devices.

### **1.5 Summary:**

The logical instruction is provided by using this app is basically, help for those people who do not aware about their daily use or any other recipes. It may help to those people who are away from home and alone. This app helps them and provide them convenient way to select and make the dishes because it recommends according to their own ingredients the only thing, they must do is just enter the ingredients which we have in our home and system will recommend them recipe according to our ingredients. It is very helpful for those people too who loved to try different type of dishes, cuisine, and recipes.

The recipe recommendation System is developed using a combination of technologies, including Android Studio, Xml, JavaScript, SQLite, and Flask. These technologies enable the creation of a robust and interactive mobile application that can be accessed from various android devices. In summary, the “**COOKMANIA**” System aims to revolutionize the process of recommend, recipes, and managed cuisine. By combining machine learning, Mobile application technologies, and user-friendly interfaces, the system improves accessibility to, enhances the accuracy of recipes, streamlines communication between user and system data, and ensures efficient data management. Ultimately, the system strives to improve recipes outcomes, with a comprehensive and reliable platform for managing their recommended recipes

## **CHAPTER 2 LITERATURE REVIEW AND SYSTEM SPECIFICATIONS**

### **2.1 Literature Review:**

The culinary world has witnessed a surge in cooking apps, yet many fall short in addressing practical cooking challenges. A significant drawback is the oversight of ingredient constraints, leading to frustration and unnecessary food waste [10]. In response to this gap in the market, we propose a revolutionary cooking app designed to consider your available ingredients and provide tailored recipe suggestions.

Our research has identified a common flaw in existing cooking apps – a lack of user-friendly design. This often results in missed items and the need for ingredient substitutions. Recognizing this issue, our app prioritizes user experience by allowing you to input the ingredients you have at home. This approach not only simplifies the cooking process but also minimizes food waste, creating a seamless and enjoyable cooking experience.

Leveraging the power of technology, our app is developed using Android Studio, XML, Java, and Flask to ensure efficiency and a visually appealing interface. These technologies work together to provide a smooth user experience, making the app accessible and user-friendly for individuals of all skill levels.

A key feature that sets our app apart is the incorporation of machine learning. This technology analyzes the ingredients available in your kitchen and generates accurate recipe recommendations. By understanding your unique ingredient constraints, the app can suggest recipes that not only suit your preferences but also minimize the need for additional grocery shopping [11].

Food waste is a global issue with significant environmental, economic, and social implications. As the world grapples with the consequences of overconsumption and inefficient resource utilization, innovative solutions are emerging to address these challenges. This literature review explores the existing research and technological developments related to minimizing food waste through the use of cooking apps, with a focus on tailoring recipe suggestions to available ingredients.

Numerous studies highlight the alarming rates of food waste globally and the associated consequences. According to the Food and Agriculture Organization (FAO), approximately one-third of the world's food produced for human consumption is lost or wasted annually. Food waste contributes to environmental degradation, greenhouse gas emissions, and economic losses. Understanding the multifaceted nature of this issue is crucial for developing effective strategies to mitigate its impact. The proliferation of smartphones and advancements in mobile technology have led to the development of cooking apps designed to assist users in meal planning and preparation. However, many existing apps fall short in addressing food waste concerns, as they often provide generic recipes that may not align with the ingredients users have at hand. Recent literature emphasizes the importance of user-centric design in cooking apps to enhance user experience and encourage sustainable cooking practices. Apps that allow users to input available ingredients and receive tailored recipe suggestions are gaining attention for their potential to minimize food waste. These approaches align with the principles of user empowerment, cost reduction, and environmental sustainability [12].

Advancements in technology, such as machine learning and artificial intelligence, have been pivotal in developing cooking apps that can analyze and recommend recipes based on available ingredients. These technologies enable more accurate and personalized suggestions, addressing the limitations of traditional apps and contributing to a more efficient and environmentally friendly cooking process. Our app goes beyond generic recipe suggestions by providing personalized recommendations based on your available ingredients. This customization ensures that the recipes align with your taste preferences and dietary restrictions, creating a more enjoyable and satisfying cooking experience. To further enhance the user experience, our app includes features that encourage community engagement. Users can share their favorite recipes, tips, and ingredient combinations, fostering a sense of culinary camaraderie. This collaborative aspect adds a social dimension to the app, creating a vibrant community of home cooks.

In conclusion, our cooking app addresses the shortcomings of existing apps by prioritizing user-friendly design, leveraging advanced technologies, and focusing on personalized recipe recommendations. By reducing food waste and simplifying the cooking process, we aim to revolutionize the way individuals approach home cooking. Our app is not just a tool; it's a culinary companion that empowers users to create delicious meals with the ingredients they already have at their fingertips [13].

### **2.1.1 User centric design and experience:**

Successful applications in the culinary domain prioritize user-centric design. That's why we have prioritize the user experience as our first priority.

### **2.1.2 Minimizing Food Waste through Ingredient :**

Recognizing the importance of minimizing food waste, researchers and developers are exploring innovative solutions. The integration of ingredient technology into recipe recommendation systems represents a promising approach. By allowing users to input available ingredients, these systems aim to tailor recipe suggestions to the user's specific inventory, thereby reducing the likelihood of unused items [14].

### **2.1.3 Machine Learning for Enhanced Recommendations:**

Machine learning models play a pivotal role in analyzing user-provided ingredient information and generating relevant recipe recommendations. Leveraging APIs and advanced algorithms, these models enhance the accuracy and personalization of suggested recipes, aligning more closely with the user's preferences and available ingredients.



## **CHAPTER 3: DESIGN AN METHODOLOGY**

### **3.1 Recommendation model:**

A recommendation model refers to a type of machine learning model that is trained to get recommendation on the base of output to develop the model firstly we preprocess the data by cleaning it and erasing the null values in our dataset after that we change the data into string to make sure the model will detect only string type values for recommendation we use sklearn function TF-IDF with cosine similarity moreover ballroot is also use which is also the functionality of sklearn to make the model efficient and accurate [15].

### **3.2 Methodology:**

- Gather a diverse dataset of food recipes that includes the different types of recipes such as Indian, south Indian, vegetarian, and non-vegetarian, moreover it's also including healthy and junk food recipes. The dataset should contain enough recipes for each dish category.
- Prepare the dataset for model training by performing various preprocessing steps. This includes cleaning the data, excluding the null values of dataset, selection of the require data for recommendation system and changing the data into string for the better accuracy.
- Choose an appropriate machine learning model architecture for recommendation system. TF-IDF vectorization with cosine similarity is best fit for the model moreover ballroot is also use for the fast working of model under low equipment device
- Evaluate the trained model on the validation set to assess its accuracy. Fine-tune the model further if necessary to improve its performance and accuracy.
- Develop the mobile application using the chosen technologies, such as XML for the front-end, JAVA for the back-end, and SQL for the database. Implement user authentication and authorization functionalities. Integrate the trained recommendation model into the system, allowing users to input ingredients, obtain recipe recommendation with the help of volley and models API.
- Conduct thorough testing of the system to ensure its functionality, usability, and performance. Fix any bugs or issues that arise during testing. Once the system is deemed stable, deploy it on a suitable hosting platform or server to make it accessible to users.
- Gather user feedback, monitor system performance, and continuously improve the system based on user needs and technological advancements. Consider incorporating additional features or expanding the dataset to enhance the accuracy and capabilities of the recipe recommendation system.

### **3.3 Tools**

#### **3.3.1 Software/IDE:**

android Studio used to develop android application.

### **3.3.2 Android Studio:**

Android Studio is an integrated development environment (IDE) designed specifically for Android app development. It provides a comprehensive set of tools and features that streamline the process of creating, testing, and deploying Android applications. Android Studio offers a user-friendly interface, code editing, debugging, and various tools for designing graphical user interfaces, making it the primary development environment for Android developers [16].

### **3.3.3 Pycharm:**

PyCharm is an integrated development environment (IDE) tailored for Python programming. Developed by JetBrains, it offers a range of tools and features to enhance the Python development experience. PyCharm provides code editing, debugging, testing, and project management capabilities, along with intelligent code completion and navigation. Its user-friendly interface and robust set of functionalities make it a popular choice among developers working with Python for web development, data science, and other applications [17].

### **3.3.4 Postman:**

Postman is a popular API development and testing tool that simplifies the process of working with APIs. It provides a user-friendly interface for sending HTTP requests, inspecting responses, and testing API endpoints. With Postman, developers can easily create and organize requests, set headers and parameters, and analyze the API responses. It supports various request types, including GET, POST, PUT, DELETE, and more, allowing developers to interact with APIs and test different scenarios. Postman also provides features for generating API documentation, collaborating with team members, and automating tests. Overall, Postman streamlines the API development workflow, making it easier to build and test APIs effectively.

### **3.3.5 SQLite:**

SQLite is a lightweight, serverless, self-contained, and open-source relational database management system (RDBMS). It is embedded directly into the application, making it suitable for scenarios where a full-fledged database server is unnecessary. SQLite is widely used in mobile applications, embedded systems, and small-scale projects due to its simplicity, ease of use, and minimal setup requirements [18].

### **3.3.6 XML:**

XML, or eXtensible Markup Language, is a versatile and human-readable markup language used to structure and store data in a format that is both machine-readable and easy for humans to understand. It uses a set of rules for encoding documents in a format that is both hierarchical and customizable. XML is commonly employed for data interchange between different systems and platforms, facilitating the exchange of structured information in a standardized way.

### **3.3.7 JAVA:**

Java is a high-level, object-oriented programming language designed to be platform-independent. It is known for its simplicity, versatility, and the "write once, run anywhere" principle, as Java applications can run on any device with a Java Virtual Machine (JVM). Java is widely used in web development, mobile app development (Android), enterprise applications, and various other domains due to its portability, strong community support, and extensive libraries.

### **3.3.8 Flask:**

Flask is a lightweight and flexible web framework for building web applications using the Python programming language. It provides a simple and intuitive interface for developers to create web applications quickly and efficiently. Flask follows the model-view-controller (MVC) architectural pattern, allowing developers to separate concerns and maintain a clean code structure. With Flask, developers can easily define routes, handle requests, and render dynamic HTML templates. It also supports various extensions for adding additional functionality, such as handling database operations, authentication, and API development [19]. Flask's simplicity and modularity make it a popular choice for developing small to medium-sized web applications and APIs. It has a supportive community and extensive documentation, enabling developers to get started with Flask easily and find solutions to common development challenges. Whether building a simple web application or a more complex project, Flask provides a flexible foundation for Python developers to create powerful and scalable web applications.

### **3.3.9 Google collab:**

Google Colab, short for Colaboratory, is a cloud-based platform provided by Google for free, enabling users to write and execute Python code in a collaborative environment. It runs on Google's servers and offers access to powerful hardware resources, including GPUs, without requiring local installations [20]. Colab is commonly used for data analysis, machine learning, and deep learning tasks. It provides an interactive and shareable notebook interface, making it easy for users to collaborate on coding projects in real-time.

### **3.3.10 Python:**

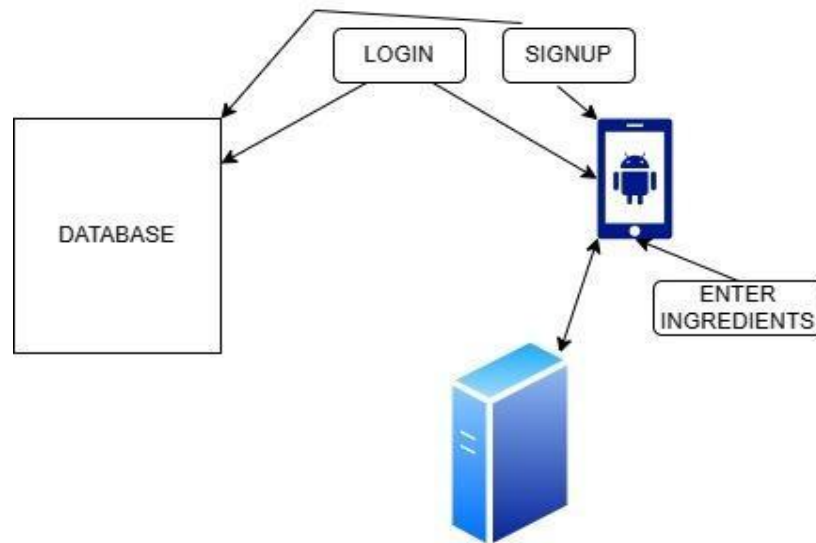
Python is a high-level, interpreted, and versatile programming language known for its readability and ease of use. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming [21]. Python is widely used for web development, data analysis, artificial intelligence, machine learning, and automation. Its simplicity and extensive standard libraries make it a popular choice for beginners and experienced developers alike.

### **3.3.10 Ngrok:**

Ngrok is a tunneling software that creates a secure connection between a local server and the internet, allowing external access to the local server. It provides a public URL that can be used to access a web server, API, or any other local service without exposing the server directly to the internet. Ngrok is commonly used during development and testing to share and showcase projects, as it simplifies the process of creating secure tunnels for local services [22].

### 3.4 SYSTEM MODEL:

A system model is a conceptual representation or abstraction of a system, created to understand, describe, or simulate the structure, behavior, and interactions within that system. It serves as a tool for system analysts, designers, and stakeholders to visualize and communicate the different aspects of a system in a systematic and organized manner. There are various types of system models, and the choice of a particular model depends on the specific needs and goals of the analysis or development process



*Figure 3.1 system model*

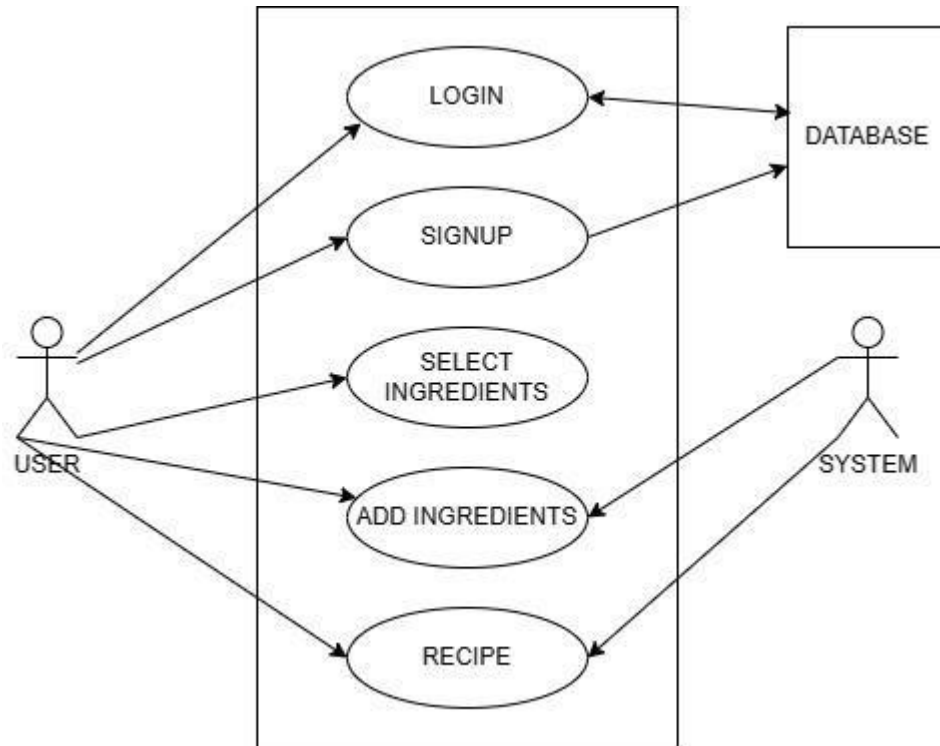
### 3.5 Main Modules:

#### 3. 5.1 User:

- Login
- Register
- Select ingredients quantity to recommend
- Recommends recipes on the base of ingredients

### 3.6 Use Case Diagram:

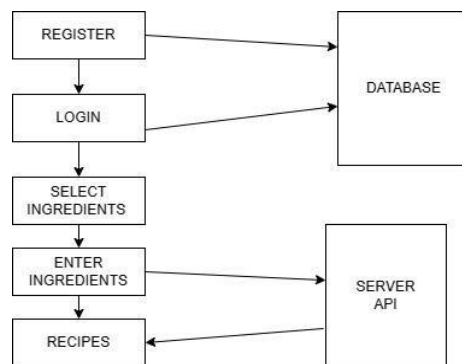
A use case diagram is a visual representation of the functional requirements of a system, showcasing the interactions between actors (users or external systems) and the system itself. It illustrates the various use cases or actions that actors can perform within the system and how they are interconnected. The use case diagram helps to identify and understand the system's functionality, boundaries, and relationships, providing a high-level overview of the system's behavior from a user's perspective.



*Figure 3.2 usecase of the model*

### 3.7 Architecture Diagram:

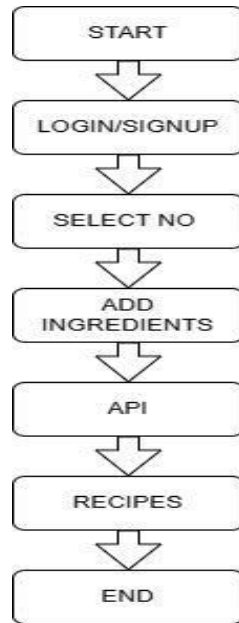
An architecture diagram provides a high-level visual representation of the components and their interactions within a system or software application. It depicts the structure, relationships, and flow of data or control between different modules or layers of the system. The architecture diagram highlights the key components, such as the user interface, business logic, data storage, and external interfaces, and illustrates how they connect and communicate with each other. It helps in understanding the overall system design, including the distribution of responsibilities, dependencies, and interfaces between various components.



*Figure 3.3 Architecture diagram*

### 3.7 Flow Diagram:

A flow diagram, also known as a flowchart, is a visual representation of a process or system using various symbols to illustrate the sequence of steps or activities. It provides a clear and structured overview of how tasks are performed or how information flows within a system. Flow diagrams use standardized shapes and arrows to depict decision points, actions, inputs, outputs, and the order in which they occur. This graphical representation aids in understanding, analyzing, and communicating complex processes or systems, making it a valuable tool in various fields, including business, engineering, and software development.



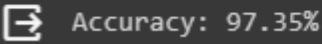
**Figure 3.4 Flowchart diagram**

## CHAPTER 4 RESULTS

In the chapter results which is all about the working of all modules of the mobile-application. Therefore, we will start from the disease this web-application detects which are shown in the following. As the user or patient can use it without a signup or login, we will directly move on to the model trained and detection of the diseases shown in the following images.

```
accuracy = evaluate_recommendation_system(test_data, recommend)

print(f'Accuracy: {accuracy * 100:.2f}%')
```



*Figure 4.1 Accuracy of the model*

### 4.1 Recommendation model:

```
Recipe Name: Carrot And Capsicum Rice Recipe
Instructions: To prepare Carrot And Capsicum Rice Recipe, firstly heat ghee in a Kadai and add
Recipe Name: Carrot Puree Recipe - First Food For Babies
Instructions: To begin making carrot puree recipe, first wash and peel the carrots. Cut them i
Recipe Name: Cheese Stuffed Mashed Potato Balls Recipe (Non Fried)
Instructions: To begin making Cheese Stuffed Mashed Potato Balls Recipe (Non Fried), boil and
Recipe Name: Bharwa Karela With Sweet Potato Filling Recipe
Instructions: To begin making the Bharwa Karela with Sweet Potato Filling Recipe, first peel t

Recipe Name: Sweet Potato Flatbread Recipe - Sweet Potato Chapati
Instructions: To begin making the Sweet Potato Flatbread recipe, boil sweet potatoes in a pres
Recipe Name: Carrot Chutney Recipe
Instructions: To begin with carrot chutney, soak the dry red chillies and mustard seeds in jus
Recipe Name: Sesame Potato Toast Recipe
Instructions: To begin making the Sesame Potato Toast, first boil the potatoes until soft. Pee

Recipe Name: Vegetarian Malai Kebab Recipe
Instructions: To begin making the Vegetarian Malai Kebab Recipe, keep all the above things rea
Recipe Name: Potato Stuffed Dal Dhokli Recipe
Instructions: To prepare Potato Stuffed Dal Dhokli Recipe, get prep with all the ingredients r
Recipe Name: Carrot Tadka Raita Recipe- Crunchy Grated Carrot In Yogurt
Instructions: To begin making the Carrot Tadka Raita Recipe, in a mixing bowl, combine yogurt,
```

*Figure 4.2 Result of the model*

We can see in this picture model is recommending recipes on the base of ingredients.

## 4.2 Start page:

By clicking this user can enter into login and sign-up page to get the recommendations.



*Figure 4.3 start of the app*

### 4.2.1 Login:

This is used to login with your authentic data



**Figure 4.3.1 login page of the app**



### 4.2.2 Signup:

This is used to signup in the app to get recommendation.



*Figure 4.3.2 Sign-up page of the app*

### 4.2.3 Select ingredients no:

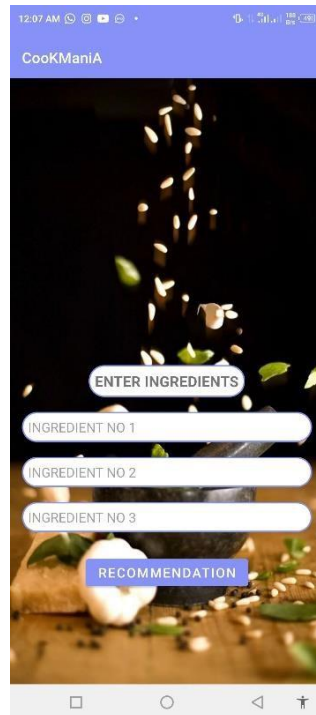
This is used to select ingredients number. On the base of ingredients user have.



*Figure 4.3.3 ingredient selection page of the app*

#### 4.2.4 Three ingredients:

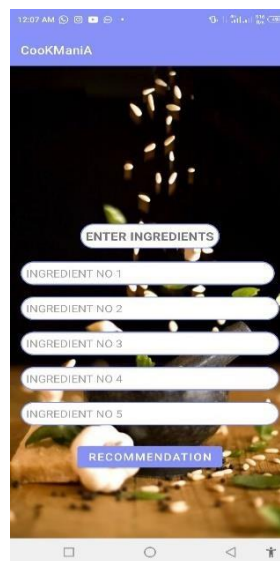
This is used to enter three ingredients. If user have three ingredients he will use this.



*Figure 4.3.4 three ingredient page of the app*

#### 4.2.5 five ingredients:

This is used to enter five ingredients. If user have five ingredients he will use this to enter.



*Figure 4.3.5 five ingredients page of the app*

#### 4.2.6 Eight ingredients:

This is used to enter eight ingredients. If user have eight ingredients he will use this page and enter one by one ingredients in order.

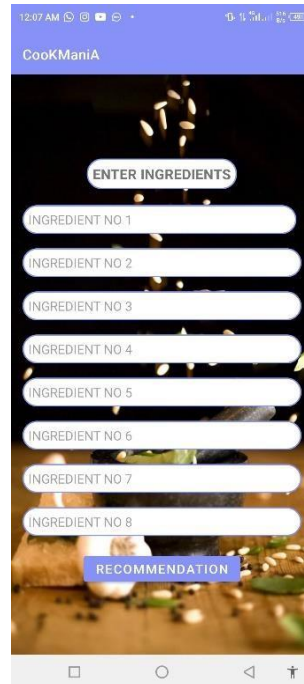


Figure 4.3.6 eights ingredients page of the app

#### 4.2.7 Recipe results:

This is used to display recipe results. All recommended recipes will display here in this page.

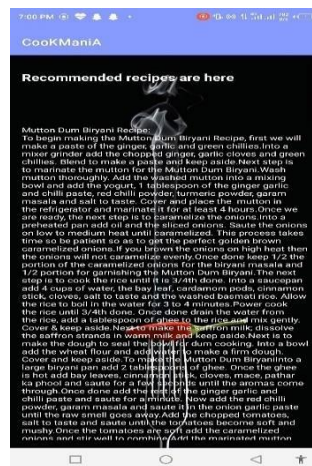


Figure 4.3.7 result page of the app

## **CHAPTER 5 CONCLUSIONS AND FUTURE WORK**

In the future, several enhancements and additional features can be developed into the cook mania to further improve its functionality, usability, and effectiveness.

### **5.1 Conclusions:**

Hence, a mobile application is developed which can take ingredients from the user, apply recommendation model and give recommendation of recipes. This makes easier for the users to cook recipe on the base of ingredients they have and making it easier for them to try new dishes every day. It is not just budget friendly for the users but also less time taking and if we look from the user point of view it is easier for them who cook occasionally because they are not experienced enough to cook good food.

### **5.2 Enhanced Machine Learning Model:**

Continuously improving and updating the machine learning models used for recommendation can significantly enhance the accuracy and reliability of recommending recipes. Future work could involve integrating advanced algorithms, expanding the dataset with more diverse recipes to try or make different cuisine of the world.

#### **5.2.1 Saving Recommended recipes:**

Implementing a system that provides saving recommended recipes for future use will increase the user interest. By implementing this feature user can save the recipe in the database of app to make it next time without using the recommendation system to recommend.

#### **5.2.2 Multi-language recipes:**

Adding support for multiple languages recipes can improve accessibility for users from diverse backgrounds. This feature can be implemented by incorporating multilingual interfaces, dataset, translation services, and language-specific content.

#### **5.2.3 Adding images of ingredients:**

Some people are from the diverse background they don't know the name of the ingredients they have in English so by implementing this feature will help those people to use our app without any mediate help.

#### **5.2.4 Adding recipes from different countries:**

At this point we only target the Asian people by implementing this different user from different part of the world will also use that app for the recommendation of the recipes.

## REFERENCES:

- [1] Ueta, Tsuguya, Masashi Iwakami, and Takayuki Ito. "A recipe recommendation system based on automatic nutrition information extraction." In Knowledge Science, Engineering and Management: 5th International Conference, KSEM 2011, Irvine, CA, USA, December 12-14, 2011. Proceedings 5, pp. 79-90. Springer Berlin Heidelberg, 2011.
- [2] Maruyama, Takuma, Yoshiyuki Kawano, and Keiji Yanai. "Real-time mobile recipe recommendation system using food ingredient recognition." In Proceedings of the 2nd ACM international workshop on Interactive multimedia on mobile and portable devices, pp. 27-34. 2012.
- [3] Forbes, Peter, and Mu Zhu. "Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation." In Proceedings of the fifth ACM conference on Recommender systems, pp. 261-264. 2011.
- [4] Elsweiler, David, Hanna Hauptmann, and Christoph Trattner. "Food Recommender Food recommender Systems." In Recommender Systems Handbook, pp. 871-925. New York, NY: Springer US, 2012.
- [5] Zhang, Jieyu, Miqing Li, Weibo Liu, Stanislao Lauria, and Xiaohui Liu. "Many-objective optimization meets recommendation systems: A food recommendation scenario." Neurocomputing 503 (2022): 109-117.
- [6] Vivek, M. B., N. Manju, and M. B. Vijay. "Machine learning based food recipe recommendation system." In Proceedings of International Conference on Cognition and Recognition: ICCR 2016, pp. 11-19. Springer Singapore, 2018.
- [7] Maruyama, Takuma, Yoshiyuki Kawano, and Keiji Yanai. "Real-time mobile recipe recommendation system using food ingredient recognition." In Proceedings of the 2nd ACM international workshop on Interactive multimedia on mobile and portable devices, pp. 27-34. 2012.
- [8] Pecune, Florian, Lucile Callebort, and Stacy Marsella. "A Recommender System for Healthy and Personalized Recipes Recommendations." In HealthRecSys@ RecSys, pp. 15-20. 2020.
- [9] Toledo, Raciél Yera, Ahmad A. Alzahrani, and Luis Martinez. "A food recommender system considering nutritional information and user preferences." IEEE Access 7 (2019): 96695-96711.
- [10] Rostami, Mehrdad, Vahid Farrahi, Sajad Ahmadian, Seyed Mohammad Jafar Jalali, and Mourad Oussalah. "A novel healthy and time-aware food recommender system using attributed community detection." Expert Systems with Applications 221 (2023): 119719.
- [11] Ferretto, Luciano Rodrigo, Cristiano Roberto Cervi, and Ana Carolina Bertoletti de Marchi. "Recommender systems in mobile apps for health a systematic review." In 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-6. IEEE, 2017. 15
- [12] Ribeiro, David, João Machado, Jorge Ribeiro, Maria João M. Vasconcelos, Elsa F. Vieira, and Ana Correia De Barros. "SousChef: Mobile meal recommender system for older adults."

In International Conference on Information and Communication Technologies for Ageing Well and eHealth, vol. 2, pp. 36-45. SCITEPRESS, 2017.

[13] Maia, Rui, and Joao C. Ferreira. "Context-aware food recommendation system." Context-aware food recommendation system (2018): 349-356.

[14] De Croon, Robin, Leen Van Houdt, Nyi Htun, Gregor Štiglic, Vero Vanden Abeele, and Katrien Verbert. "Health recommender systems: systematic review." Journal of Medical Internet Research 23, no. 6 (2021): e18035.

[15] Gavalas, Damianos, and Daphne Economou. "Development platforms for mobile applications: Status and trends." IEEE software 28, no. 1 (2010): 77-86.

[16] Mascolo, Cecilia, Licia Capra, and Wolfgang Emmerich. "An XML-based Middleware for Peerto-Peer computing." In Proceedings First International Conference on Peer-to-Peer Computing, pp. 69-74. IEEE, 2001.

[17] Portugal, Ivens, Paulo Alencar, and Donald Cowan. "The use of machine learning algorithms in recommender systems: A systematic review." Expert Systems with Applications 97 (2018): 205-227.

[18] Akbarnejad, Javad, Gloria Chatzopoulou, Magdalini Eirinaki, Suju Koshy, Sarika Mittal, Duc On, Neoklis Polyzotis, and Jothi S. Vindhiya Varman. "SQL QueRIE recommendations." Proceedings of the VLDB Endowment 3, no. 1-2 (2010): 1597-1600.

[19] DiMarzio, Jerome. Beginning android programming with android studio. John Wiley & Sons, 2016. [20] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[21] Van Horn, Richard L. "Validation of simulation results." Management Science 17, no. 5 (1971): 247-258.

[22] Wong, Paul TP. "Toward a dual-systems model of what makes life worth living." The human quest for meaning: Theories, research, and applications 2 (2012): 3-22

## APPENDIX A

### CODE:

#### MODEL CODE

```
import numpy as np
import pandas as pd
set=pd.read_csv('set11.csv')
set['TranslatedIngredients']=set['TranslatedIngredients']+' '+set['Diet']
set=set[['TranslatedRecipeName','TranslatedIngredients','TranslatedInstructions']]
set.isnull().sum()
set.dropna(inplace=True)
set.duplicated().sum()
set.iloc[1].TranslatedIngredients
set['TranslatedIngredients']=set['TranslatedIngredients'].apply(lambda x:x.split() )
set['TranslatedIngredients']= set['TranslatedIngredients'].apply(lambda x:[i.replace(" ","")for i
in x] )
set['tag']=set['TranslatedIngredients']
new_df=set[['TranslatedRecipeName','TranslatedInstructions','tag']]
new_df['tag']=new_df['tag'].apply(lambda x:" ".join(x) )
new_df['tag']=new_df['tag'].apply(lambda x:x.lower() )
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.neighbors import BallTree
import numpy as np
new_df['CombinedFeatures'] = new_df['tag'] + ' ' + new_df['TranslatedInstructions']
vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = vectorizer.fit_transform(new_df['CombinedFeatures'].values.astype('U'))
dense_tfidf_matrix = tfidf_matrix.toarray()
# Calculate cosine similarity
similarity = cosine_similarity(dense_tfidf_matrix, dense_tfidf_matrix)

# Use BallTree for efficient nearest neighbors search
balltree = BallTree(dense_tfidf_matrix)

def recommend(ingredients, new_df=new_df, vectorizer=vectorizer, balltree=balltree):
    # Convert input ingredients to a TF-IDF vector
    input_features = ' '.join(ingredients)
    input_tfidf_vector = vectorizer.transform([input_features]).toarray()

    # Query BallTree for nearest neighbors
    _, indices = balltree.query(input_tfidf_vector, k=10)
```

```

# Get the recommended recipes details
recommended_recipes = new_df.iloc[indices[0]][['TranslatedRecipeName',
'TranslatedInstructions']]

return recommended_recipes
from sklearn.model_selection import train_test_split
train_data, test_data = train_test_split(new_df, test_size=0.2, random_state=42)
train_data, test_data = train_test_split(new_df, test_size=0.2, random_state=42)
def evaluate_recommendation_system(test_data, recommendation_function):
    total_recipes = 0
    relevant_recipes = 0

    for _, row in test_data.iterrows():
        ingredients = row['tag'].split() # Use translated ingredients from the test set

        # Get recommended recipes
        recommended_recipes = recommendation_function(ingredients)

        # Assess relevance: check if the actual recipe name is in the recommended set
        actual_recipe_name = row['TranslatedRecipeName']
        if actual_recipe_name in recommended_recipes['TranslatedRecipeName'].values:
            relevant_recipes += 1

    total_recipes += 1

    accuracy = relevant_recipes / total_recipes if total_recipes != 0 else 0
    return accuracy

# Evaluate the recommendation system
accuracy = evaluate_recommendation_system(test_data, recommend)

print(f'Accuracy: {accuracy * 100:.2f}%')
API code
from flask import Flask, request, jsonify
import pickle
import pandas
import numpy as np
app = Flask(__name__)

```



```

# Load the dictionary from the pickle file
with open('model11.pkl', 'rb') as file:
    loaded_objects = pickle.load(file)

    new_df = loaded_objects['new_df']
    vectorizer = loaded_objects['vectorizer']
    balltree = loaded_objects['balltree']

# Your existing code for the recommend function
def recommend(ingredients, new_df, vectorizer, balltree):
    # Convert input ingredients to a TF-IDF vector
    input_features = ' '.join(ingredients)
    input_tfidf_vector = vectorizer.transform([input_features]).toarray()

    # Query BallTree for nearest neighbors
    _, indices = balltree.query(input_tfidf_vector, k=10)

    # Get the recommended recipes details
    recommended_recipes = new_df.iloc[indices[0]][['TranslatedRecipeName',
'TranslatedInstructions']]

    return recommended_recipes

# API endpoint for recommendation
@app.route('/')
def hello_world():
    return 'hello hasnat'

@app.route('/recommend', methods=['POST'])
def get_recommendations():
    # Get ingredients from the form data
    ingre1 = request.form.get('ingre1')
    ingre2 = request.form.get('ingre2')
    ingre3 = request.form.get('ingre3')
    ingredients = [ingre1, ingre2, ingre3]

    # Call the recommend function with the required arguments

```

```

recommended_recipes = recommend(ingredients, new_df, vectorizer, balltree)

# Format recommendations as a list of strings
recommendations_list = []
for index, row in recommended_recipes.iterrows():
    recommendation_string
    f"{row['TranslatedRecipeName']}:\n{row['TranslatedInstructions']}\n"
    recommendations_list.append(recommendation_string)

# Join the list into a single string with line breaks
recommendations_text = '\n'.join(recommendations_list)

return jsonify({'recommendations': recommendations_text})
@app.route('/recommend1', methods=['POST'])
def recommendations():
    # Get ingredients from the form data
    ingre1 = request.form.get('ingre1')
    ingre2 = request.form.get('ingre2')
    ingre3 = request.form.get('ingre3')
    ingre4 = request.form.get('ingre4')
    ingre5 = request.form.get('ingre5')
    ingredients = [ingre1, ingre2, ingre3, ingre4, ingre5]

    # Call the recommend function with the required arguments
    recommended_recipes = recommend(ingredients, new_df, vectorizer, balltree)

    # Format recommendations as a list of strings
    recommendations_list = []
    for index, row in recommended_recipes.iterrows():
        recommendation_string
        f"{row['TranslatedRecipeName']}:\n{row['TranslatedInstructions']}\n"
        recommendations_list.append(recommendation_string)

    # Join the list into a single string with line breaks
    recommendations_text = '\n'.join(recommendations_list)

    return jsonify({'recommendations': recommendations_text})
@app.route('/recommend2', methods=['POST'])
def getrecommendations():
    # Get ingredients from the form data
    ingre1 = request.form.get('ingre1')

```

```

ingre2 = request.form.get('ingre2')
ingre3 = request.form.get('ingre3')
ingre4 = request.form.get('ingre4')
ingre5 = request.form.get('ingre5')
ingre6 = request.form.get('ingre6')
ingre7 = request.form.get('ingre7')
ingre8 = request.form.get('ingre8')
ingredients = [ingre1, ingre2, ingre3, ingre4, ingre5, ingre6, ingre7, ingre8]

recommended_recipes = recommend(ingredients, new_df, vectorizer, balltree)

# Format recommendations as a list of strings
recommendations_list = []
for index, row in recommended_recipes.iterrows():
    recommendation_string
    f"{row['TranslatedRecipeName']}:\n{row['TranslatedInstructions']}\n"
    recommendations_list.append(recommendation_string)

# Join the list into a single string with line breaks
recommendations_text = '\n'.join(recommendations_list)

return jsonify({'recommendations': recommendations_text})

if __name__ == '__main__':
    app.run(debug=True)

```

## XML CODE

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:layout_centerInParent="true"
    android:background="@drawable/b"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".choice">

    <!-- Second Button -->

    <TextView
        android:id="@+id/titleTextView"

```

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="16dp"
    android:layout_marginTop="250dp"
    android:layout_marginEnd="9dp"
    android:background="@drawable/lavender_border"
    android:padding="8dp"
    android:text="SELECT HOW MANY INGREDIENTS YOU HAVE"
    android:textSize="20sp"
    android:textStyle="bold"
    android:textAlignment="center"/>

```

```

<Button
    android:id="@+id/button1"
    android:layout_width="244dp"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:layout_marginTop="30dp"
    android:text="THREE INGREDIENTS"
    android:textSize="18sp" />

```

<!-- Third Button -->

```

<Button
    android:id="@+id/button2"
    android:layout_width="238dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/button1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="32dp"
    android:text="FIVE INGREDIENTS"
    android:textSize="18sp" />

```

```

<Button
    android:id="@+id/button3"
    android:layout_width="239dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/button2"
    android:layout_centerHorizontal="true"

```

```

        android:layout_marginTop="30dp"
        android:text="EIGHT INGREDIENTS"
        android:textSize="18sp" />
</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:background="@drawable/c"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".display">

    <!-- ScrollView -->
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <!-- Title -->
            <TextView
                android:id="@+id/titleTextView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Recommended recipes are here"
                android:textStyle="bold"
                android:textColor="@color/white"
                android:textSize="20sp"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="16dp"/>

            <!-- Recipe 1 -->
            <TextView
                android:id="@+id/recipeTextView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

```

```

        android:textColor="@color/white"

        android:textColorHint="@color/white"
        android:layout_marginTop="16dp"/>
<TextView
    android:id="@+id/recipe1TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/white"

    android:textColorHint="@color/white"
    android:layout_marginTop="16dp"/>
<TextView
    android:id="@+id/recipe2TextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/white"

    android:textColorHint="@color/white"
    android:layout_marginTop="16dp"/>

<!-- Recipe 2 -->

<!-- Add more recipes as needed -->

</LinearLayout>
</ScrollView>
</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:layout_centerInParent="true"
    android:background="@drawable/b"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".entering">

```

```
<!-- Title -->
<TextView
    android:id="@+id/titleTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ENTER INGREDIENTS"
    android:textStyle="bold"
    android:textSize="20sp"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="30dp"
    android:padding="8dp"
/>
```

```
<!-- First Text Field -->
<EditText
    android:id="@+id/editText1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/titleTextView"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 1"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"
/>
```

```
<!-- Second Text Field -->
<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText1"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 2"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

```
<!-- Third Text Field -->
```

```

<EditText
    android:id="@+id/editText3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText2"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 3"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>

<!-- Button -->
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText3"
    android:layout_centerHorizontal="true"
    android:text="RECOMMENDATION"
    android:textSize="18sp"
    android:layout_marginTop="30dp"
/>

</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:layout_centerInParent="true"
    android:background="@drawable/b"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".entering12">

<TextView
    android:id="@+id/titleTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"

```



```
    android:layout_marginStart="16dp"
    android:layout_marginTop="102dp"
    android:background="@drawable/lavender_border"
    android:padding="8dp"
    android:text="ENTER INGREDIENTS"
    android:textSize="20sp"
    android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="388dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/titleTextView"
    android:layout_alignParentEnd="true"
    android:layout_centerInParent="true"
    android:layout_marginTop="22dp"
    android:layout_marginEnd="7dp"
    android:background="@drawable/lavender_border"
    android:hint="INGREDIENT NO 1"
    android:padding="8dp" />
```

```
<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText1"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 2"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

<!-- Third Text Field -->

```
<EditText
    android:id="@+id/editText3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText2"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 3"
    android:background="@drawable/lavender_border"
```

```
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

```
<!-- Fourth Text Field -->
```

```
<EditText
    android:id="@+id/editText4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText3"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 4"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

```
<!-- Fifth Text Field -->
```

```
<EditText
    android:id="@+id/editText5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText4"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 5"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

```
<!-- Sixth Text Field -->
```

```
<EditText
    android:id="@+id/editText6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText5"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 6"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

```
<!-- Seventh Text Field -->
```

```
<EditText
    android:id="@+id/editText7"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText6"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 7"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

<!-- Eighth Text Field -->

```
<EditText
    android:id="@+id/editText8"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText7"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 8"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

<!-- Button -->

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText5"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="203dp"
    android:text="RECOMMENDATION"
    android:textSize="18sp" />
```

</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:layout_centerInParent="true"
    android:background="@drawable/b"
```

```
xmlns:tools="http://schemas.android.com/tools"
tools:context=".entering2">
```

```
<!-- Title -->
```

```
<!-- First Text Field -->
```

```
<!-- Second Text Field -->
```

```
<TextView
    android:id="@+id/titleTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="16dp"
    android:layout_marginTop="255dp"
    android:background="@drawable/lavender_border"
    android:padding="8dp"
    android:text="ENTER INGREDIENTS"
    android:textSize="20sp"
    android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="388dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/titleTextView"
    android:layout_alignParentEnd="true"
    android:layout_centerInParent="true"
    android:layout_marginTop="22dp"
    android:layout_marginEnd="7dp"
    android:background="@drawable/lavender_border"
    android:hint="INGREDIENT NO 1"
    android:padding="8dp" />
```

```
<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
    android:layout_below="@id/editText1"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 2"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

<!-- Third Text Field -->

```
<EditText
    android:id="@+id/editText3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText2"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 3"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

<!-- Fourth Text Field -->

```
<EditText
    android:id="@+id/editText4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText3"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 4"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
    android:padding="8dp"/>
```

<!-- Fifth Text Field -->

```
<EditText
    android:id="@+id/editText5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText4"
    android:layout_centerHorizontal="true"
    android:hint="INGREDIENT NO 5"
    android:background="@drawable/lavender_border"
    android:layout_marginTop="20dp"
```

```

        android:padding="8dp"/>

<!-- Button -->
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editText5"
    android:layout_centerHorizontal="true"
    android:text="RECOMMENDATION"
    android:textSize="18sp"
    android:layout_marginTop="30dp"
/>

</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/login"
    android:orientation="vertical"
    tools:context=".LoginActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="500dp"
                android:orientation="vertical"
                android:padding="20dp">

```

```
<EditText
    android:id="@+id/login_email"
    android:layout_width="350dp"
    android:layout_height="60dp"
    android:layout_marginTop="30dp"
    android:background="@drawable/lavender_border"
    android:drawableLeft="@drawable/ic_baseline_email_24"
    android:drawablePadding="8dp"
    android:hint="Email"
    android:padding="8dp"/>
```

```
<EditText
    android:id="@+id/login_password"
    android:layout_width="350dp"
    android:layout_height="60dp"
    android:layout_marginTop="20dp"
    android:background="@drawable/lavender_border"
    android:drawableLeft="@drawable/ic_baseline_lock_24"
    android:drawablePadding="8dp"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="8dp" />
```

```
<Button
    android:id="@+id/login_button"
    android:layout_width="350dp"
    android:layout_height="70dp"
    android:layout_marginTop="30dp"
    android:text="Login"
    android:textSize="18sp"
    app:cornerRadius="30dp" />
```

```
<TextView
    android:id="@+id/signupRedirectText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="14dp"
    android:padding="8dp"
    android:text="Not yet registered? Signup"
```

```

        android:textColor="@color/lavender"
        android:textSize="18sp" />

    </LinearLayout>
</LinearLayout>
</ScrollView>
</LinearLayout>
<?xml version="1.0" encoding="utf-8"?>

<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/login"
    tools:context=".SignupActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_marginTop="380dp"
            android:padding="20dp">

            <EditText
                android:layout_width="350dp"
                android:layout_height="60dp"
                android:id="@+id/signup_email"
                android:background="@drawable/lavender_border"
                android:layout_marginTop="30dp"
                android:padding="8dp"
                android:hint="Email"

```



```
android:drawableLeft="@drawable/ic_baseline_email_24"  
android:drawablePadding="8dp"/>
```

```
<EditText  
    android:layout_width="350dp"  
    android:layout_height="60dp"  
    android:id="@+id/signup_password"  
    android:background="@drawable/lavender_border"  
    android:layout_marginTop="20dp"  
    android:padding="8dp"  
    android:hint="Password"  
    android:inputType="textPassword"  
    android:drawableLeft="@drawable/ic_baseline_lock_24"  
    android:drawablePadding="8dp"/>
```

```
<EditText  
    android:layout_width="350dp"  
    android:layout_height="60dp"  
    android:id="@+id/signup_confirm"  
    android:background="@drawable/lavender_border"  
    android:layout_marginTop="20dp"  
    android:padding="8dp"  
    android:hint="Confirm Password"  
    android:inputType="textPassword"  
    android:drawableLeft="@drawable/ic_baseline_password_24"  
    android:drawablePadding="8dp"/>
```

```
<Button  
    android:layout_width="350dp"  
    android:layout_height="70dp"  
    android:text="Sign Up"  
    android:id="@+id/signup_button"  
    android:textSize="18sp"  
    android:layout_marginTop="30dp"  
    app:cornerRadius = "30dp"/>
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/loginRedirectText"
```

```

        android:text="Already an user? Login"
        android:layout_gravity="center"
        android:padding="8dp"
        android:textSize="18sp"
        android:layout_marginTop="8dp"
        android:textColor="@color/lavender"/>

</LinearLayout>

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView" />
</LinearLayout>
</ScrollView>
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/login"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="COOKMANIA"
        android:textSize="36sp"
        android:textColor="@color/white"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

    android:text="RECIPES ON THE BASE OF INGREDIENTS"
    android:textColor="@color/white"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.598" />

```

```

<Button
    android:id="@+id/button12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TO PROCEED"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.732" />

```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

#### JAVA CODE:

```

package com.example.cookmania;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.content.Intent;
import android.view.View;
import android.widget.Button;

public class choice extends AppCompatActivity {

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_choice);

    // Assuming you have a button with ID "button1" in your layout
    Button button1 = findViewById(R.id.button1);

    // Set OnClickListener for the button
    button1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Create an Intent to start the TargetActivity
            Intent intent = new Intent(choice.this, entering.class);

            // You can also pass data to the new activity if needed
            // intent.putExtra("key", "value");

            // Start the new activity
            startActivity(intent);
        }
    });

    // Assuming you have two more buttons with IDs "button2" and "button3"
    Button button2 = findViewById(R.id.button2);
    Button button3 = findViewById(R.id.button3);

    // Set OnClickListener for button2
    button2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(choice.this, entering2.class);
            startActivity(intent);
        }
    });

    // Set OnClickListener for button3
    button3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(choice.this, entering12.class);

```

```

        startActivity(intent);
    }
});
}
}
package com.example.cookmania;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {

    public static final String databaseName = "SignLog.db";

    public DatabaseHelper(@Nullable Context context) {
        super(context, "SignLog.db", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase MyDatabase) {
        MyDatabase.execSQL("create Table users(email TEXT primary key, password TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase MyDB, int i, int i1) {
        MyDB.execSQL("drop Table if exists users");
    }

    public Boolean insertData(String email, String password){
        SQLiteDatabase MyDatabase = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("email", email);
        contentValues.put("password", password);
        long result = MyDatabase.insert("users", null, contentValues);

        if (result == -1) {
            return false;
        } else {

```

```

        return true;
    }
}

public Boolean checkEmail(String email){
    SQLiteDatabase MyDatabase = this.getWritableDatabase();
    Cursor cursor = MyDatabase.rawQuery("Select * from users where email = ?", new
String[]{email});

    if(cursor.getCount() > 0) {
        return true;
    }else {
        return false;
    }
}
}

```