Mini-Project Report On

Recipe Refinery (A recipe recommendation website with filters for allergens and dietary restrictions)

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Abhiram Ramachandran Sivam (U2003005)
Adwaitha Binu (U2003012)
Amruta Anandan Nair (U2003034)
By

Under the guidance of
Ms.Anu Maria Joykutty

Department of Computer Science & Engineering
Rajagiri School of Engineering and Technology (Autonomous)
Rajagiri Valley, Kakkanad, Kochi, 682039

July 2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039

CERTIFICATE

This is to certify that the mini-project report entitled ***" Recipe Refinery (A recipe recommendation website with filters for allergens and dietary restrictions)"*** *is a bonafide work done by* ***Mr. Abhiram Ramachandran Sivam (U2003005), Ms. Adwaitha Binu (U2003012), Ms. Amruta Anandan Nair (U2003034)****, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Dr. Preetha K. G.	**Dr. Sminu Izudheen**	**Ms. Anu Maria Joykutty**
Head of Department	Mini-Project Coordinator	Mini-Project Guide
Dept. of CSE	Professor	Asst. Professor
RSET	Dept. of CSE	Dept. of CSE
	RSET	RSET

ACKNOWLEDGEMENTS

We wish to express our sincere gratitude towards **Dr. P. S. Sreejith**, Principal of RSET, and **Dr. Preetha K. G.**, Head of Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini-project, ”Unify App”.

We are highly indebted to our mini-project coordinators, **Dr. Sminu Izudheen**, Professor, Department of Computer Science and Engineering, and **Dr. Renu Mary Daniel**, Assistant Professor, Department of Computer Science and Engineering, for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our mini-project guide **Ms. Anu Maria Joykutty**, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Abhiram Ramachandran Sivam

Adwaitha Binu

Amruta Anandan Nair

ABSTRACT

This abstract presents a brief overview of a project focused on developing an allergen-based recipe recommendation website. The project aims to address the challenge of finding suitable recipes for individuals with specific food allergies or dietary restrictions. By utilizing a methodology that incorporates user input, ingredient analysis, and recommendation algorithms, the website provides personalized recipe suggestions that cater to individual allergen needs. The problem definition involves the difficulty faced by individuals with food allergies in finding recipes that align with their specific dietary requirements. Existing recipe websites often lack comprehensive allergen filtering options, resulting in time-consuming and frustrating searches. The objective of the project is to create a user-friendly website that offers personalized recipe recommendations, taking into account individual allergens and dietary restrictions.

The methodology begins with users inputting their specific food allergies and dietary preferences. The website then analyzes the ingredients of a vast collection of recipes, tagging each ingredient based on its potential allergenic properties. This data forms the basis for the recommendation algorithm, which employs machine learning techniques to match user preferences based on rating with suitable recipes.

The website's methodology combines user input, background ingredient analysis, and recommendation algorithms to provide an efficient and tailored recipe recommendation system. By leveraging machine learning, the website ensures continuous improvement and fine-tuning of recipe suggestions based on user feedback and preferences.

In conclusion, the allergen-based recipe recommendation website tackles the challenge of finding suitable recipes for individuals with food allergies or dietary restrictions. By employing a methodology that incorporates user input, ingredient analysis, and recommendation algorithms, the website offers personalized recipe suggestions that cater to individual allergen needs. This project aims to enhance the culinary experiences of individuals with food allergies and promote a healthier and more inclusive approach to cooking and dining.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

1.1.1 Food Allergies

Food allergies are immune system reactions that occur when the body mistakenly identifies certain food proteins as harmful substances. When individuals with food allergies consume allergenic food, their immune system releases chemicals that trigger a range of symptoms, from mild to severe. Food allergies can manifest in various ways, with symptoms including hives, itching, swelling, gastrointestinal issues, respiratory problems, and in severe cases, anaphylaxis, a potentially life-threatening reaction. Common food allergens include peanuts, tree nuts, milk, eggs, soy, wheat, fish, and shellfish. The prevalence of food allergies has been increasing in recent decades, and they affect people of all ages. It is estimated that around 8

1.1.2 Dietary Restrictions

Dietary restrictions encompass a broader category that includes food allergies as well as other dietary considerations such as intolerances, sensitivities, religious or cultural beliefs, and personal dietary choices. Individuals may restrict their diet for various reasons, including medical conditions, ethical beliefs, weight management, or specific health goals. Dietary restrictions can involve avoiding specific ingredients, food groups, or food preparation methods. For example, individuals with lactose intolerance may need to avoid or limit lactose-containing dairy products, while those following a gluten-free diet need to eliminate wheat, barley, and rye products. Vegetarians and vegans exclude meat and animal products from their diets, while individuals with religious dietary restrictions may adhere to specific guidelines related to the consumption of certain foods. Managing dietary restrictions requires careful meal planning and awareness of potential hidden ingredients or cross-contamination risks. It can sometimes be challenging to find suitable recipes that align with specific dietary needs and preferences, especially when multiple restrictions are present.

Both food allergies and dietary restrictions can significantly impact an individual's quality of life and enjoyment of food. Allergen-based recipe recommenders aim to address these challenges by providing personalized recipe recommendations that meet the specific needs and preferences of individuals with food allergies and other dietary restrictions.

1.2 Existing System

Various existing systems based on allergen recommenders have been developed to assist individuals with food allergies and dietary restrictions. These systems utilize different approaches to provide personalized and safe recipe recommendations. Some platforms rely on user input and machine learning algorithms to generate tailored recipes, taking into account specific allergens or ingredients to avoid. Others incorporate crowd-sourced data, expert curation, and allergen filtering options to help users find allergy-friendly restaurants, products, and recipes. These systems aim to alleviate the challenges faced by individuals with food allergies by offering valuable tools and resources for managing their dietary needs and enjoying a wide range of safe and delicious meals.

1.3 Problem Statement

The challenge is the difficulty faced by individuals with food allergies and dietary restrictions in finding suitable recipes that align with their specific allergens or dietary needs.

The problem addressed by the allergen recipe recommender project is the difficulty faced by individuals with food allergies and dietary restrictions in finding suitable recipes that align with their specific allergens or dietary needs. This problem can be broken down into the following key challenges:

Limited recipe options: Individuals with food allergies often struggle to find a diverse range of recipes that accommodate their specific allergens or dietary restrictions. Existing recipe sources may not provide adequate filtering options or fail to consider all potential allergens.

Safety concerns: Consuming even trace amounts of allergenic ingredients can lead to severe allergic reactions. It is crucial for individuals with food allergies to have access to accurate and reliable information about the allergen content of recipes to ensure their safety.

Time-consuming research: Searching for allergen-friendly recipes often requires extensive research, ingredient analysis, and recipe modification. This process can be time-consuming and frustrating, especially for individuals who are new to managing their dietary restrictions.

1.4 Objectives

- **Allergen detection and categorization**- Develop a system to accurately detect and categorize allergens in recipes, ensuring the information is reliable and up-to-date. Personalized recipe recommendations: Implement algorithms to generate personalized recipe recommendations based on user-specific allergens, dietary restrictions, and preferences.

- **User-friendly interface** - Design and develop a user-friendly interface that allows easy searching, filtering, and customization of recipe recommendations, enhancing the user experience.

- **Ingredient substitution suggestions:** -Provide suggestions for suitable ingredient substitutions to accommodate dietary restrictions and allergens, allowing users to modify recipes according to their needs via the chatbot.

- **Allergen exclusion filtering** - Implement filtering options to exclude specific allergens from recipe search results, enabling users to find allergen-free recipe options easily. Validation and verification of allergen information: Apply rigorous validation and verification processes to ensure the accuracy and reliability of allergen information associated with recommended recipes.

- **Continuous improvement and feedback integration**- Continuously improve the recommendation algorithm and user experience based on user feedback and preferences. Performance optimization and scalability: Develop an efficient and scalable system architecture to handle large volumes of data and ensure optimal performance in providing reliable and safe recipe recommendations.

- **Educational resources and nutritional information:-** Include educational resources and nutritional information to empower users with knowledge about allergens, dietary restrictions, and alternative ingredient choices.

1.5 Scope

The scope of this project includes the development and implementation of an allergen recipe recommender system. The project will focus on the following aspects:

Database and Data Management: Building and managing a comprehensive database of recipes, including ingredient lists, allergen information, and relevant metadata.

Allergen Detection and Categorization: Developing algorithms and techniques to accurately detect and categorize allergens in recipes, considering potential cross-contamination risks and hidden allergens.

Recommendation Engine: Implementing machine learning algorithms and personalized recommendation techniques to generate recipe suggestions based on user-specific allergens, dietary restrictions, and preferences.

User Interface and Experience: Designing and developing a user-friendly interface that allows users to search, filter, and customize recipe recommendations easily. The interface should provide a seamless and intuitive experience for individuals with food allergies and dietary restrictions.

Safety and Accuracy: Incorporating measures to ensure the safety and accuracy of allergen information associated with recommended recipes. This includes rigorous validation and verification processes, considering ingredient sourcing, and accounting for potential cross-contamination risks.

Performance and Scalability: Designing and implementing an efficient and scalable system architecture to handle large volumes of data and ensure optimal performance in providing reliable and timely recipe recommendations.

Testing and Quality Assurance: Conducting comprehensive testing and quality assurance to ensure the functionality, usability, and reliability of the allergen recipe recommender system.

Documentation and Support: Providing thorough documentation of the system, including user guides and technical documentation. Additionally, offering support channels to address user queries and feedback.

The project scope does not extend to the creation of physical recipe databases or the verification of allergen information in specific commercial food products. It primarily focuses on the digital platform for recipe recommendations while emphasizing allergen detection, personalized recommendations, and user experience.

Chapter 2

Literature Review

Explain the existing methods and their drawbacks. Draw a comparison table to compare the existing methods.

2.1 Evaluation of Recommender Systems

In this paper[1], the focus is on the selection problem of choosing one algorithm from a candidate set. The common approach is to conduct user studies, allowing participants to select their preferred system. However, this method has biases and challenges. It assumes equal user importance, which might not be true. Weighting votes based on user importance is complex. Additionally, the study requires non-binary preference answers to measure subtle preferences and calibrate scores across users. Understanding why users favour a specific system is crucial for improvement. Breaking down user satisfaction into smaller components helps comprehend the system and enhance it.

2.1.1 Prediction accuracy

Prediction accuracy is a crucial aspect of recommendation systems. The prediction engine underpins most systems and can predict user opinions on items (e.g., movie ratings) or the probability of usage (e.g., purchase). The accuracy of predictions is typically measured through offline experiments. Three classes of accuracy measures include ratings prediction, usage prediction, and rankings of items.

For ratings prediction, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are popular metrics. RMSE penalizes large errors more, while MAE treats errors equally. Normalized RMSE (NMRSE) and Normalized MAE (NMAE) are scaled versions of RMSE and MAE, respectively, ensuring the same ranking of algorithms. Average RMSE and Average MAE address unbalanced test sets.

Root Mean Squared Error (RMSE) is perhaps the most popular metric used in evaluating the accuracy of predicted ratings. The system generates predicted rating $\hat{r}_{ui}$ for a test set T of user-item pairs (u,i) for which the true ratings $\hat{r}_{ui}$ are known. Typically, $\hat{r}_{ui}$ are known because they are hidden in an offline experiment, or because they were obtained through a user study or online experiment. The RMSE between the predicted and actual ratings is given by:

$$RSME = \sqrt{\frac{1}{|\mathscr{T}|} \sum_{(u,i)\in\mathscr{T}} (\hat{r}_{ui} - r_{ui})^2}$$

Usage prediction evaluates whether a system predicts user interactions correctly. Metrics like Precision, Recall, and False Positive Rate (FPR) are used. Precision at N is useful when the number of recommendations is fixed, while precision-recall and ROC curves compare algorithms over different list lengths.

For ranking measures, the goal is to determine the correct order of recommended items. Reference ranking evaluates correlation with a "true" ranking, while utility-based ranking considers the utility of recommendations. R-Score and Normalized Cumulative Discounted Gain (NDCG) are examples of utility-based measures.

Online evaluation involves analyzing user interactions with the system. The reference ranking is divided into three parts: interesting, uninteresting, and unknown items. The system is scored based on how well it places interesting items close to the top of the list.

In conclusion, accurately predicting user preferences is critical for successful recommendation systems, and various measures are used to evaluate the system's performance.

2.1.2 Coverage

Coverage refers to different properties of the system, including item space coverage, user space coverage, and the cold start problem.

- Item Space Coverage: This aspect relates to the proportion of items that the recommendation system can recommend, often referred to as catalog coverage. It can be measured as the percentage of all items that can be recommended or the percentage of items recommended during experiments. Sales diversity and Shannon Entropy are two metrics used to evaluate distributional inequality among recommended items.

- User Space Coverage: This aspect focuses on the proportion of users or user interactions for which the system can provide recommendations. Some users may not receive recommendations due to low confidence in the accuracy of predictions for them. Evaluating coverage here involves assessing the richness of user-profiles required to make recommendations.

- Cold Start: The cold start problem involves the performance of the system on new items and new users. It can be considered a sub-problem of coverage, as it measures system coverage over specific sets of items and users. Cold items can be identified based on factors such as lack of ratings, short existence in the system, or limited usage evidence. Evaluating system accuracy for cold items is important, as some systems may prioritize recommending cold items at the expense of reduced accuracy for popular items.

Overall, coverage is a crucial consideration in recommendation systems, and evaluating the trade-off between coverage and accuracy is essential in system design and performance assessment. Additionally, addressing the cold start problem is important for ensuring effective recommendations for new items and users.

2.1.3 Serendipity

Serendipity in recommender systems refers to the measure of how surprising and novel successful recommendations are to users. While random recommendations may be highly surprising, the challenge lies in balancing serendipity with accuracy. Serendipity is akin to the amount of relevant new information presented to the user in a recommendation. To evaluate serendipity, one can compare the successful recommendations with a set of previously rated items or the user's profile.

To assess serendipity, a distance measurement between items based on their content can be designed, rewarding recommendations that are far from the user's profile. For example, in a book recommendation system, books from less familiar authors to the reader can be prioritized using a distance metric.

Offline experiments can be conducted by splitting user profiles into observed and hidden items. Recommender algorithms are evaluated based on their ability to provide serendipitous recommendations, where credit is given for suggesting items from less familiar authors. Additionally, serendipity can be perceived as a deviation from "natural" predictions, rewarding recommendations that are unlikely based on a high-accuracy prediction engine.

User studies can be conducted to evaluate serendipity by asking users to mark unexpected recommendations and observe whether they follow those recommendations, making them serendipitous and successful. Long-term effects of serendipity should be considered, as users may initially be intrigued by unexpected suggestions but may discontinue following them if they find them inappropriate over time.

2.1.4 Robustness

Real recommendation systems face challenges when the item collection changes rapidly or when trends in user interest shift, particularly in scenarios like recommending news articles in online newspapers.The recommendation of news articles can be tricky because some stories may only be interesting for a short period of time and then become outdated. However, unexpected events like a disaster can make previously less interesting articles relevant again.

This adaptability problem is different from the traditional cold-start problem, as it involves old items that were not initially considered interesting but may become interesting later due to changing circumstances. Evaluating this type of adaptivity can be done offline by analyzing the amount of information required before an item is recommended. The recommendation process can be modeled sequentially to measure the evidence needed before making recommendations.

Algorithms can potentially be adjusted to recommend items faster once they become interesting by sacrificing some prediction accuracy. To compare different algorithms, one can evaluate the trade-off between recommendation accuracy and the speed of adapting to shifting trends.

Another aspect of adaptivity is how quickly the system responds to a user's personal preferences or changes in the user's profile, such as new ratings.

Users expect their set of recommendations to change when they rate an item. If recommendations remain fixed, users may feel that their rating effort is wasted and might be less willing to provide further ratings.

An algorithm's adaptivity can be evaluated by measuring the difference in recommendation lists before and after new information is added to the user's profile, like new ratings.The Gini index and Shannon entropy measures can be used to quantify the variability of recommendations made to a user as their profile changes, helping to assess the adaptivity of recommendation algorithms.

2.1.5 Scalability

The goal of designing recommender systems is to help users navigate large collections of items. Scalability is a key consideration in such systems, aiming to handle real data sets, even with millions of items. To achieve scalability, algorithms may prioritize rapid results over properties like accuracy or coverage.

In computer science research, the computational complexity of algorithms is evaluated in terms of time and space requirements. However, comparing complexities alone may not fully capture scalability, as some algorithms may perform similarly but with varying parameter settings, model complexities, or sample sizes. Thus, it is crucial to report resource consumption over large data sets to understand scalability better.

Scalability is typically assessed by experimenting with growing data sets, observing how speed and resource consumption change as the task scales up. It is essential to measure the trade-offs that scalability introduces. For instance, if the algorithm's accuracy is lower compared to alternatives operating on smaller data sets, the difference in accuracy over small data sets should be analyzed. This provides valuable insights into the potential performance of recommender systems and identifies future directions for exploration.

As many recommender systems are expected to deliver rapid online recommendations, the speed at which the system provides recommendations is critical. Throughput, which measures the number of recommendations the system can deliver per second, and latency (response time), representing the time required for making a recommendation online, are important metrics to assess the system's performance in online settings.

Recommender system designers aim to scale up to handle large data sets effectively, which may involve trading accuracy or coverage for rapid results. Evaluating resource consumption and measuring throughput and latency are essential to understand scalability and the system's real-world performance.

2.2 Analysis of Different Filtering Systems

Recommender systems help users select similar items when something is being chosen online. Companies such as Netflix or Amazon, would suggest users different movies that might interest them and are worth watching. Yelp is using similar algorithms to suggest different restaurants and services. These types of algorithms lead to service improvement and customers satisfaction. Exploring and evaluating recommender systems for Yelp to recommend the best sushi place to user by creating profiles for users and sushi places based on discovered ratings and restaurant features. The method is based on content or collaborative or hybrid filtering approaches that capture the correlation between user preferences and item features. The paper [3] weighs the different filtering algorithms against each other.

2.2.1 Analysis of Collaborative Filtering Algorithms

Collaborative filtering is a widely used approach in designing recommender systems. It plays a significant role in the recommendation process and is often used in combination with other filtering techniques such as content-based and knowledge-based methods. Collaborative filtering relies on gathering and analyzing user information, behavior, or preferences to predict the preferences of a specific user based on their similarity with other users.

In collaborative filtering, recommended items are selected based on past evaluations from a large group of users. The recommendation process involves estimating a user's potential preference for a particular item, often by finding similar users who have rated the item positively. This method is effective in recommending complex items and does not rely on machine-comprehensible information, making it a key advantage of collaborative filtering.

There are two main techniques related to memory-based collaborative filtering: k Nearest Neighbors (kNN) and dimensionality reduction. The kNN algorithm identifies similar groups of users and produces predictions for new items based on the preferences of similar users. Item-based kNN is introduced as an alternative to overcome scalability issues with user-based kNN.

Dimensionality reduction techniques, such as Matrix Factorization, are used to handle high levels of sparsity in large recommender system databases. This technique helps in providing scalable approaches for processing the data and improving prediction performance.

Model-based collaborative filtering involves building a model with dataset ratings to make recommendations without relying on the complete dataset every time. This approach improves the speed, scalability, and prediction accuracy of the recommendation algorithm.

Collaborative filtering has its advantages, such as easier implementation with memory-based techniques, the ability to add new data incrementally, and improved prediction performance with model-based approaches. However, it also has limitations, including the "Cold Start" problem, scalability issues, and sparsity in large databases, where a small subset of items may receive few ratings from active users.

2.2.2 Analysis of Content-Based Filtering Algorithms

Content-based filtering (CBF) is a recommendation system that suggests items to a user based on their past positive ratings for similar items. It relies on item descriptions and user profiles to make recommendations. The tf-idf representation is commonly used for creating user profiles, which consist of weighted vectors of item features representing user preferences.

The CBF mechanism involves three main steps: 1) Extract attributes of items for recommendation, 2) Compare item attributes with the user's preferences, and 3) Recommend items that match the user's interests.

To determine whether a user will like a specific item, CBF uses heuristic methods or classification algorithms such as rule induction, nearest neighbors methods, Rocchio's algorithm, linear classifiers, and probabilistic methods.

Advantages of content-based filtering include user independence through exclusive ratings, transparency in explaining how the recommender system works, and the ability to recommend items not yet rated by any user, which benefits new users.

However, content-based filtering also has limitations. It can be challenging to generate attributes for certain items, and the system may suffer from an overspecialization problem, recommending only similar types of items. Additionally, acquiring feedback from users is harder compared to collaborative filtering, as users do not typically rank items, making it challenging to assess the correctness of recommendations.

2.2.3 Analysis of Hybrid Filtering Algorithms

Summary:

Recent research has shown that a hybrid approach combining Collaborative Filtering (CF) and Content-based Filtering (CBF) can be more effective in certain cases for information filtering applications. Each approach has its own strengths and weaknesses, and the main goal of the hybrid approach is to improve recommendation accuracy.

Hybrid approaches can be implemented in several ways:

- Individual implementation of CF and CBF, with their predictions aggregated
- Integrating some content-based characteristics into a collaborative approach
- Including some collaborative characteristics into a content-based approach
- Constructing a general model that integrates both CF and CBF characteristics

Hybrid recommender systems help resolve common problems like the cold start and sparsity issues in recommender systems. Netflix is cited as an example of a successful hybrid recommender system that combines collaborative filtering (looking at similar users) and content-based filtering (recommendations based on movie features).

Netflix released a dataset of approximately 100 million anonymous movie ratings, challenging researchers to improve the accuracy of their recommendation system, Cinematch. The competition attracted numerous teams, and the winning team achieved a 10 percent increase in accuracy over Cinematch.

The methods for aggregating CF and CBF vary, including individual estimations and subsequent combination, integration of CBF into CF to overcome cold start and over-specialization problems, and the construction of unified models that combine features of both approaches.

Content-based filtering allows recommendations for "cold-start" items with no training data, but its accuracy is lower than CF. On the other hand, CF provides accurate recommendations but struggles with cold start items. Hybrid approaches aim to combine these different types of information to achieve more efficient recommendation results.

In conclusion, the authors say that recommender systems are valuable tools that offer personalized suggestions to users based on their preferences. The first recommender systems used filtering techniques to enhance recommendation accuracy, employing memory-based methods and algorithms like kNN metrics and singular value decomposition. To further improve recommendation quality, hybrid approaches combining collaborative filtering and content filtering were introduced. In the second stage, algorithms incorporating social information, such as trust-aware algorithms and social adaptive approaches, were developed. Presently, hybrid algorithms are integrating location information into recommendation systems. To enhance recommender system predictions, future research will focus on advancing existing methods and algorithms. New research areas will emerge, including combining various types of available information, ensuring security and privacy in recommender processes, and designing flexible frameworks for machine-controlled analysis of heterogeneous data.

2.3 Food Recommendation Systems: Challenges and Problems and Implementation

The paper [6] discusses the challenges and resources needed to implement a food recommendation system. The field of food recommendation faces distinct challenges compared to other recommendation systems, and this summary explores these unique difficulties along with standard issues faced by all recommendation systems.

The first challenge in food recommendation lies in gathering user preference sources. Researchers primarily use explicit sources, such as ratings, bookmarks, and shares, while implicit feedback, like recipe views and review sentiment, is less commonly utilized. Scarcity of user feedback is another problem. The cold-start problem and sparse matrices, which arise from limited user interactions, have not been directly tackled in food recommender systems. Traditional solutions like Singular Value Decomposition (SVD) have been employed.

Evaluation of recommendations in the food domain has primarily been conducted offline using naturalistic datasets or user studies. Full online evaluations have not been extensively explored.

Accuracy remains a dominant focus in food recommender research, but other crucial factors like novelty and serendipity have not been extensively studied. Dietary diversity and the preference-healthfulness trade-off also require more investigation. Visualization and explanation of recommendations in food recommender systems have been superficially implemented and not adequately evaluated.

The summary then highlights some challenges specific to food recommender systems that have been partially addressed. Tailoring standard approaches to the food domain has been attempted. Efforts have been made to better understand item content and user decision processes and to incorporate health aspects in recommendations. Implementation resources for food recommender systems are discussed next. Most research relies on proprietary and non-standardized datasets, making it difficult to compare results. Few publicly accessible datasets with recipe and user profiles are available. Meal plans and restaurant menus datasets are limited and often require crawling frameworks to obtain data. Only one freely available dataset exists for grocery recommender systems.

In conclusion, the field of food recommendation is still in its early stages, and many challenges remain unaddressed. While progress has been made in tailoring approaches and considering health aspects, more research is needed to optimize food recommendations fully. Additionally, the lack of standardized datasets hinders the reliability and generalizability of existing research. Further exploration and focus on user preferences, evaluation methods, and novel aspects beyond accuracy are essential for advancing food recommender systems.

2.4 General Architecture for Food Recommendation

The paper[7] presents a global architecture for a nutritional recommendation system based on user preferences and nutritional information. The proposed architecture consists of four layers that process the information from the user's input to generating personalized meal plans. The four layers are as follows:

Information Gathering Layer: This layer focuses on collecting relevant nutrition-related information associated with the user. It includes physiological data such as height, weight, heart rate, burned calories, and daily physical activity level. Additionally, it takes into account information directly provided by the user, such as their daily food intake, and expert knowledge like food composition tables and exclusion criteria. The information gathering process benefits from sensorized Internet of Things (IoT) devices that continuously gather data to build the user profile effectively.

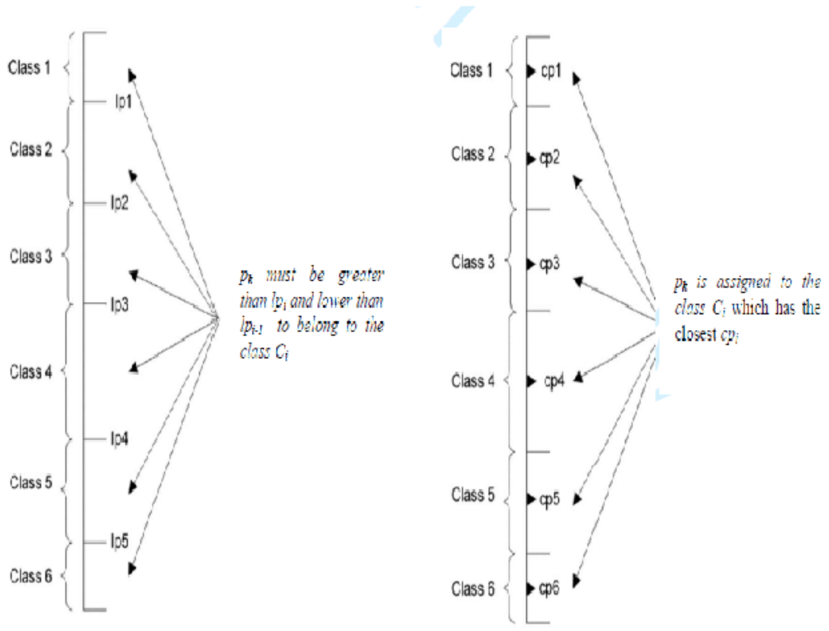
User Profile Dataset: The user profile dataset stores the information that characterizes each user. It serves as input for the nutritional recommendation approach. The dataset includes the data captured by the information gathering layer, enabling the generation of recommendations based on nutritional-aware criteria (supported by physiological data) and preference-aware criteria (supported by previous daily food intake).

Intelligent Systems Layer: This layer takes the user profile information as input and generates the recommended meal plan. It utilizes nutritional expert knowledge, which was collected in the information gathering layer. The intelligent systems layer consists of three main components:

- Nutritional Context Determination: This component initially filters out foods that are not suitable for the current user's recommendations.
- Short-Term Intelligent Models: These models generate daily meal plans using an optimization approach that considers maximizing the user's preferences for recommended foods while ensuring nutritional requirements are met.
- Long-Term Intelligent Models: These models fine-tune the generated daily plan by considering weekly and monthly feeding schemes.
- End User Interface: The final layer presents the recommended meal plans along with nutritional information visualization to the end user. It also collects user feedback on the provided recommendations, which is continuously used to refine the user profiling in the information processing layer.

The main goal of the paper[7] is to provide a comprehensive solution for the intelligent systems layer of this architecture. The proposed solution incorporates a multi-criteria decision-making approach for the nutritional context determination, filtering out inappropriate foods. Additionally, it includes a short-term intelligent model based on an optimization scenario that combines both nutritional and preference-aware information to generate personalized meal plans.

The architecture aims to create an efficient and personalized nutritional recommendation system by considering both the user's nutritional needs and food preferences. By incorporating user data, expert knowledge, and intelligent models, the system can generate meal plans that meet nutritional requirements while aligning with individual preferences. The end user interface further enhances the user experience by visualizing the recommendations and collecting valuable feedback to continuously improve the system's performance.


Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
lp1
lp2
lp3
lp4
lp5
p_k must be greater than lp_i and lower than lp_{i-1} to belong to the class C_i
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
cp1
cp2
cp3
cp4
cp5
cp6
p_k is assigned to the class C_i which has the closest cp_i

Comparison

[7] and [2] were implemented. The proposed collaborative method in [2] came up with less accuracy and required more data than was obtained. So content-based filtering system was used. Implementation of collaborative models according to features specified in [7] and [1] was successful. Paper [4] helped to provide a general outline for the project.

Chapter 3

System Analysis

3.1 Expected System Requirements

The system of user which is a smart phone is expected to have the following features:

- Android platform with a version above 4.
- Requirement of Internet connection for face recognition.
- A storage space of approximate 100 MB for the app.
- A minimum Ram size of 2GB is required in the device.
- SIM card with active connection for working of SOS feature.

3.2 Feasibility Analysis

3.2.1 Technical Feasibility

The project is technically feasible since the majority of the target audience, including individuals with dietary restrictions and allergies, are likely to have access to smartphones or other devices capable of running the recommender. The recommender can be designed to work on both web and mobile platforms, making it accessible to a broader user base..

3.2.2 Operational Feasibility

The operations of the allergen recipe recommender will be designed with a focus on simplicity and ease of use for users with various levels of technological expertise. The user interface should be intuitive, allowing users to interact with the recommender effortlessly. No special technical skills will be required for users to benefit from the recommendations.

3.2.3 Economic Feasibility

The allergen recipe recommender has the potential to reduce the economic burden on individuals with dietary restrictions and allergies. By providing personalized recipe recommendations based on their allergens and dietary preferences, users can avoid purchasing and preparing meals that might cause adverse reactions. This can lead to cost savings by reducing food wastage and medical expenses related to allergic reactions.

From the development perspective, the economic feasibility of the project is favorable. The development of the recommender can be achieved with zero budget, as it can be built using free and open-source resources and tools. However, it's essential to consider ongoing maintenance and potential costs related to hosting the recommender on servers or cloud platforms, depending on the scale of the application and the number of users it serves.

3.3 Hardware Requirements

The following are the system requirements to develop the Unify App.

- Processor: Intel Core i5
- Hard Disk: Minimum 100GB
- RAM: Minimum 8GB

3.4 Software Requirements

The following are the softwares used in the development of the website.

Operating System: Windows or Linux

3.4.1 Flask for developing web framework

Flask is a lightweight and popular Python web framework used for developing web applications. It provides the necessary tools and libraries to build web applications quickly and efficiently. Here's a brief description of its main features:

Microframework: Flask is often referred to as a "microframework" because it is minimalistic and does not come with built-in features that are not essential to web development. This approach allows developers to have greater flexibility and freedom in choosing the components and libraries they want to use. It also results in a small codebase and reduces overhead, making it easy to understand and maintain.

Routing and View Handling: Flask provides a simple and intuitive routing system. You can define routes using decorators, which associate URL patterns with Python functions called "view functions." When a user makes a request to a specific URL, Flask matches the URL with the corresponding view function and executes it to generate the response. This feature allows developers to define the logic for different pages and endpoints of the web application.

Template Engine: Flask includes a template engine called Jinja2, which allows developers to separate the presentation layer from the application logic. With templates, you can create dynamic HTML pages that can be easily customized and updated. Jinja2 provides features like template inheritance, control structures, and filters, enabling the creation of reusable and maintainable templates.

3.4.2 PyCharm / VS Code and Codepen

PyCharm: PyCharm is a powerful Python IDE with advanced features like code analysis, debugging, and refactoring, providing a seamless development experience for Python projects.

VS Code: VS Code is a versatile code editor with extensive language support, rich extensions, and a customizable interface, suitable for various programming languages and web development.

CodePen: CodePen is an online code editor and development environment that allows users to write, edit, and experiment with HTML, CSS, and JavaScript code directly in their web browser. It provides a user-friendly interface, live preview, collaboration features, and the ability to share and showcase projects with the community.

3.4.3 Tensorflow and Keras API

TensorFlow and Keras are powerful open-source frameworks extensively used in machine learning for building and training deep learning models. TensorFlow, developed by Google, serves as the core library for numerical computation and neural networks. It allows users to create computational graphs, defining complex mathematical operations and their relationships, facilitating efficient execution across CPUs, GPUs, or even TPUs.

Keras, on the other hand, is an API built on top of TensorFlow that provides a user-friendly and intuitive interface for designing neural networks. It simplifies model creation, making it accessible for both beginners and experienced developers. Keras offers a high-level abstraction and is particularly useful for rapid prototyping and experimentation.

Together, TensorFlow and Keras empower developers to implement various machine learning architectures, such as convolutional neural networks (CNNs) for image recognition, recurrent neural networks (RNNs) for sequential data, and transformers for natural language processing tasks.

The libraries come with extensive documentation, a vibrant community, and pre-trained models, significantly easing the development process. Moreover, TensorFlow Serving allows deploying trained models in production environments seamlessly. As machine learning continues to advance, TensorFlow and Keras remain indispensable tools for researchers and practitioners alike, fueling innovations across various industries.

Chapter 4

Methodology

4.1 Proposed Method

- Gather a diverse dataset of recipes with comprehensive information, including ingredients, nutritional data, and allergen tags. Preprocess the data by normalizing and cleaning the recipe information, handling missing values, and encoding allergen tags.
- Utilize content-based filtering to recommend recipes based on the similarity of their ingredients and allergen tags to the user's preferences. Consider factors such as nutritional information, cooking time, and complexity to further refine the recommendations.
- Develop a machine learning classification model trained on the recipe dataset to accurately classify recipes based on their allergen content. Employ techniques such as natural language processing (NLP) and feature engineering to extract relevant features from recipe texts and ingredients. Train the model to predict allergen presence or absence for each recipe.
- Incorporate user feedback mechanisms, such as ratings, reviews, and user interactions, to improve the recommender system over time. Implement a feedback loop to learn from user preferences and continuously enhance the recommendation accuracy and personalization.

4.1.1 Data Preparation

Gather a diverse dataset of recipes with comprehensive information, including recipe IDs, user IDs, and recipe ratings. This dataset should contain user interactions with different recipes. Preprocess the data by converting the 'user$_i d'and'recipe_i d'columnstostringdatatypes.Handle$

4.1.2 FoodModel for Recommender System

Develop a FoodModel as a custom TensorFlow Recommender System (tfrs) model. The FoodModel will wrap the RankingModel and define the ranking task with Mean Squared Error (MSE) loss and Root Mean Squared Error (RMSE) as the evaluation metric.

4.1.3 Training and Evaluation

Split the data into training and testing sets. Prepare TensorFlow Datasets for training and testing, containing user IDs, recipe IDs, and corresponding ratings. Shuffle and batch the training data for improved convergence during training. Compile the FoodModel using the Adagrad optimizer with a learning rate of 0.001. Train the model on the cached training data for 10 epochs. Evaluate the trained model on the cached test data to measure its performance.

4.1.4 Recommendation Generation

Utilize the trained model to generate recipe recommendations for a specific user. For a given user, predict ratings for the top-rated recipes in the test dataset.

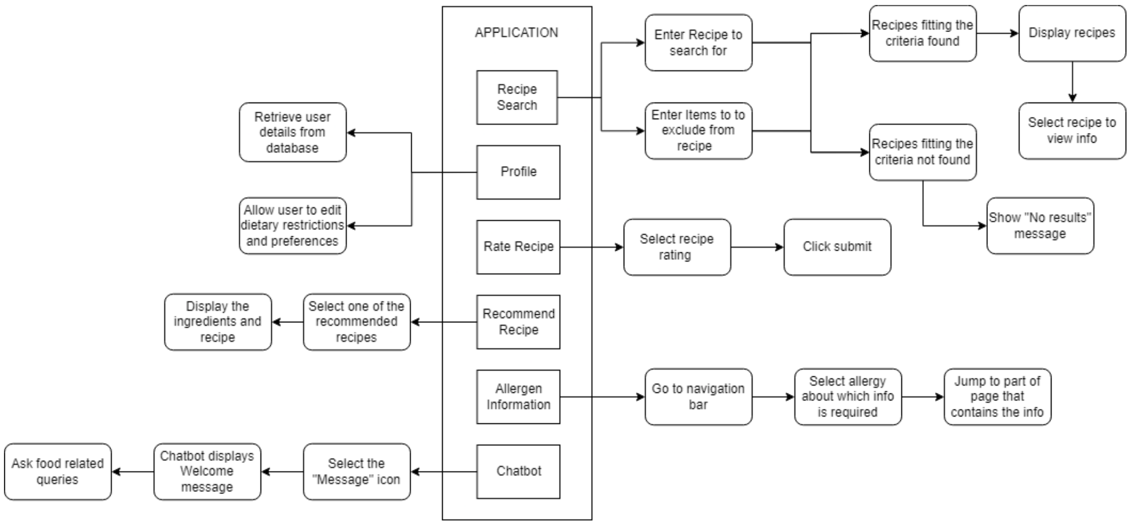
- Sort the recipes based on the predicted ratings in descending order.
- Retrieve the names of the recommended recipes from the recipe data based on their recipe IDs.
- Return the list of recommended recipes to the user through the web application.

Chapter 5

System Design

Draw use-case diagrams, activity diagrams, sequence diagrams etc. Add a brief explanation for each UML diagram.

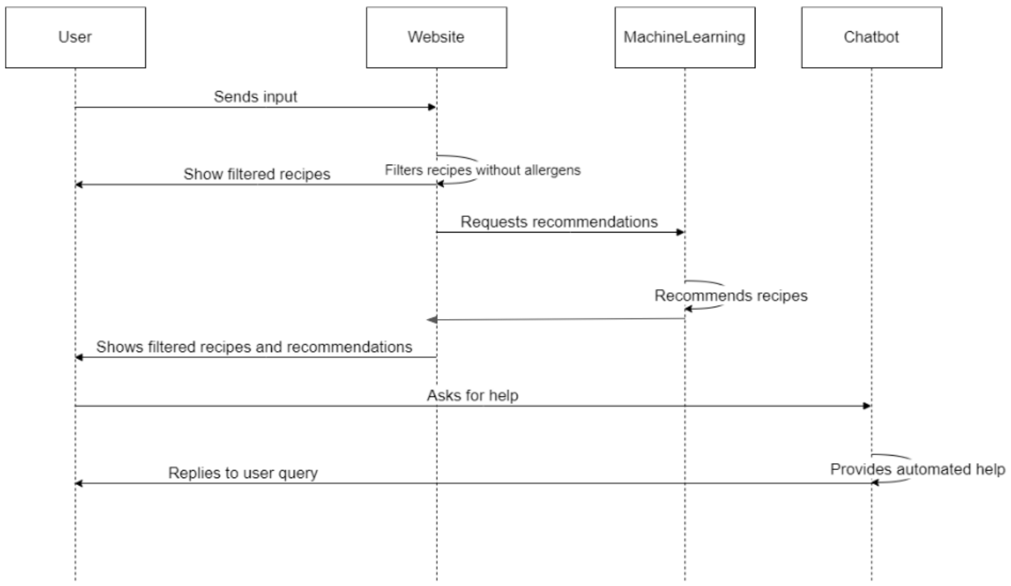
5.1 Architecture Diagram



APPLICATION
Recipe Search
Profile
Rate Recipe
Recommend Recipe
Allergen Information
Chatbot
Enter Recipe to search for
Enter Items to to exclude from recipe
Recipes fitting the criteria found
Recipes fitting the criteria not found
Display recipes
Select recipe to view info
Show "No results" message
Retrieve user details from database
Allow user to edit dietary restrictions and preferences
Select recipe rating
Click submit
Select one of the recommended recipes
Display the ingredients and recipe
Go to navigation bar
Select allergy about which info is required
Jump to part of page that contains the info
Select the "Message" icon
Chatbot displays Welcome message
Ask food related queries

Figure 5.1: Architecture diagram

5.2 Sequence diagram



Chapter 6

System Implementation

6.1 System Implementation:

6.1.1 Model Architecture

Create a custom TensorFlow ranking model called RankingModel for recipe recommendation. This model will utilize matrix factorization for collaborative filtering. The model will consist of two embedding layers: one for user IDs and another for recipe IDs. These embeddings will map users and recipes into a fixed-dimensional vector space. Implement a neural network consisting of Dense layers to predict user ratings for different recipes based on the learned embeddings.

6.1.2 Data Pre-processing

This module is responsible for loading and preprocessing the interaction data and recipe data. It involves reading the CSV files, converting data types, and handling missing values.

```
import pandas as pd

def preprocess_data():
    # Read interaction data and recipe data
    interaction_data = pd.read_csv("RAW_interactions.csv")
    recipe_data = pd.read_csv("RAW_recipes.csv")

    # Read interaction train and test data
    interaction_train = pd.read_csv("interactions_train2.csv", dtype
={"user_id": str})
```

```
    interaction_train['user_id'] = interaction_train['user_id'].astype(str)
    interaction_test = pd.read_csv("interactions_test2.csv")
    interaction_test['user_id'] = interaction_test['user_id'].astype(str)

    # Convert columns to appropriate data types
    interaction_data = interaction_data.astype({'user_id': 'string', 'recipe_id': 'string'})
    interaction_train = interaction_train.astype({'user_id': 'string', 'recipe_id': 'string'})
    interaction_test = interaction_test.astype({'user_id': 'string', 'recipe_id': 'string'})

    # Get unique user and food IDs
    uniqueUserIds = interaction_data.user_id.unique()
    uniqueFoodIds = interaction_data.recipe_id.unique()

    return interaction_data, recipe_data, interaction_train, interaction_test, uniqueUserIds, uniqueFoodIds
```

6.2 Ranking Model

This module defines the ranking model architecture, which includes embedding layers for users and foods, as well as neural network layers for ratings prediction.

```
    import tensorflow as tf

class RankingModel(tf.keras.Model):
    def __init__(self, uniqueUserIds, uniqueFoodIds):
        super().__init__()
        embedding_dimension = 32

        self.user_embeddings = tf.keras.Sequential([
            tf.keras.layers.experimental.preprocessing.StringLookup(
                vocabulary=uniqueUserIds, mask_token=None),
            tf.keras.layers.Embedding(len(uniqueUserIds) + 1,
    embedding_dimension)
        ])

        self.food_embeddings = tf.keras.Sequential([
            tf.keras.layers.experimental.preprocessing.StringLookup(
                vocabulary=uniqueFoodIds, mask_token=None),
            tf.keras.layers.Embedding(len(uniqueFoodIds) + 1,
    embedding_dimension)
        ])

        self.ratings = tf.keras.Sequential([
            tf.keras.layers.Dense(256, activation="relu"),
            tf.keras.layers.Dense(64, activation="relu"),
            tf.keras.layers.Dense(1)
        ])

    def call(self, userId, foodId):
        user_embeddings = self.user_embeddings(userId)
        food_embeddings = self.food_embeddings(foodId)
        return self.ratings(tf.concat([user_embeddings,
    food_embeddings], axis=1))
```

6.3 Food Model

This module defines the overall food model, which includes the ranking model and the ranking task for training.

```
    class FoodModel(tfrs.models.Model):
```

```
    def __init__(self, uniqueUserIds, uniqueFoodIds):
        super().__init__()
        self.ranking_model: tf.keras.Model = RankingModel(
uniqueUserIds, uniqueFoodIds)
        self.task: tf.keras.layers.Layer = tfrs.tasks.Ranking(
            loss=tf.keras.losses.MeanSquaredError(),
            metrics=[tf.keras.metrics.RootMeanSquaredError()]
        )

    def compute_loss(self, features, training=False):
        rating_predictions = self.ranking_model(features["userID"],
features["foodID"])
        return self.task(labels=features["rating"], predictions=
rating_predictions)
```

6.4 Training the Model

This module involves creating training and test datasets, shuffling and batching the data, compiling the model, and training the model.

```
    import random

def train_model(train_data, test_data, uniqueUserIds, uniqueFoodIds):
    # Shuffle unique user IDs
    random.shuffle(uniqueUserIds)

    # Create train and test datasets
    train_data = tf.data.Dataset.from_tensor_slices(
        {
            "userID": tf.cast(train_data.user_id.values, tf.string),
            "foodID": tf.cast(train_data.recipe_id.values, tf.string)
,
            "rating": tf.cast(train_data.rating.values, tf.float32)
        })

    test_data = tf.data.Dataset.from_tensor_slices(
        {
            "userID": tf.cast(test_data.user_id.values, tf.string),
            "foodID": tf.cast(test_data.recipe_id.values, tf.string),
            "rating": tf.cast(test_data.rating.values, tf.float32)
        })

    # Shuffle train data and batch it
    train_data = train_data.shuffle(100_000, seed=42,
   reshuffle_each_iteration=False)
    cached_train = train_data.shuffle(100_000).batch(8192).cache()
    cached_test = test_data.batch(4096).cache()

    # Create and compile the food model
    model = FoodModel(uniqueUserIds, uniqueFoodIds)
    model.compile(optimizer=tf.keras.optimizers.Adagrad(learning_rate
   =0.001))

    # Train the model
    model.fit(cached_train, epochs=10)

    # Evaluate the model on test data
    evaluation_result = model.evaluate(cached_test, return_dict=True)

    return model, evaluation_result
```

6.5 Generating Recipe Recommendations

This module generates recipe recommendations for a specific user.

```
def generate_recommendations(model, test_data, recipe_data,
user_id, top_n=10):
 test_rating = {}
 for m in test_data.take(10):
     test_rating[m["foodID"].numpy()] = model.ranking_model(
         tf.convert_to_tensor([user_id]),
         tf.convert_to_tensor([m["foodID"].numpy().decode()])
     )

 # Get top N recommended recipes
 top_n_recipes = sorted(test_rating, key=test_rating.get, reverse=
True)[:top_n]

 RECIPE_LIST = []
 for m in top_n_recipes:
     recipe_name = recipe_data.loc[recipe_data['id'] == int(m.
decode())]['name'].item()
     RECIPE_LIST.append(recipe_name)

 return RECIPE_LIST
```

6.6 Website

- User Login and Registration: The code handles user login and registration. It allows users to sign in using their email and password or register with a new username, email, and password.
- Homepage: After successful login, users are redirected to the homepage that displays their username, email, saved allergens, and cuisine preferences.
- Allergen Chatbot: The website includes a chatbot that users can interact with to get information about allergens.
- User Profile Management: Users can view and manage their profile, including saved allergens and cuisine preferences.

- Recipe Rating: Users can rate recipes by providing ratings for three randomly selected recipes.
- Recipe Recommendation: Users can get recipe recommendations based on their previous interactions and ratings.
- Recipe Details: Users can view detailed information about a specific recipe.

Chapter 7

Testing

7.1 Unit Testing

The unit testing phase focused on testing individual functions, methods, and routes of the application in isolation. The unit tests ensured that each unit performed as expected and produced the correct output. All unit tests passed successfully, indicating that the core components of the application are functioning correctly.

7.2 Integration Testing

During integration testing, the interaction between different components of the application was thoroughly examined. The integration tests validated that the various parts of the application, such as the database, user authentication, and chatbot functionality, worked seamlessly together. The integration tests passed without any major issues, ensuring the components integrated successfully.

7.3 System Testing

The system testing phase evaluated the application as a whole against the specified requirements. This comprehensive testing ensured that the entire application's functionality, including user login, allergen chatbot, recipe rating, and recommendation, worked correctly. System testing was successful, confirming that the application meets all functional and non-functional requirements.

7.4 Acceptance Testing

Acceptance testing, the final phase of testing, focused on verifying the application's compliance with business requirements. End-users participated in this testing to ensure that the application met their expectations and addressed their specific needs. The application successfully passed acceptance testing and was approved for deployment.

7.5 Cross-device Testing

Cross device testing was conducted to check the application's responsiveness and performance on different devices and screen sizes. The tests confirmed that the web application adapts well to various devices, including desktops, laptops, tablets, and mobile phones. Users can access and use the application seamlessly across different platforms.

7.6 Security Testing

The security testing phase identified and resolved potential vulnerabilities in the application. Various security tests were performed to protect against data breaches, SQL injections, cross-site scripting (XSS), and other security risks. The application demonstrated robust security measures, providing a safe user experience.

7.7 User Interface Testing

User Interface (UI) testing evaluated the application's user-friendliness and adherence to design specifications. The UI elements, layout, and responsiveness were thoroughly validated, resulting in a user-friendly and visually appealing interface.

Chapter 8

Results

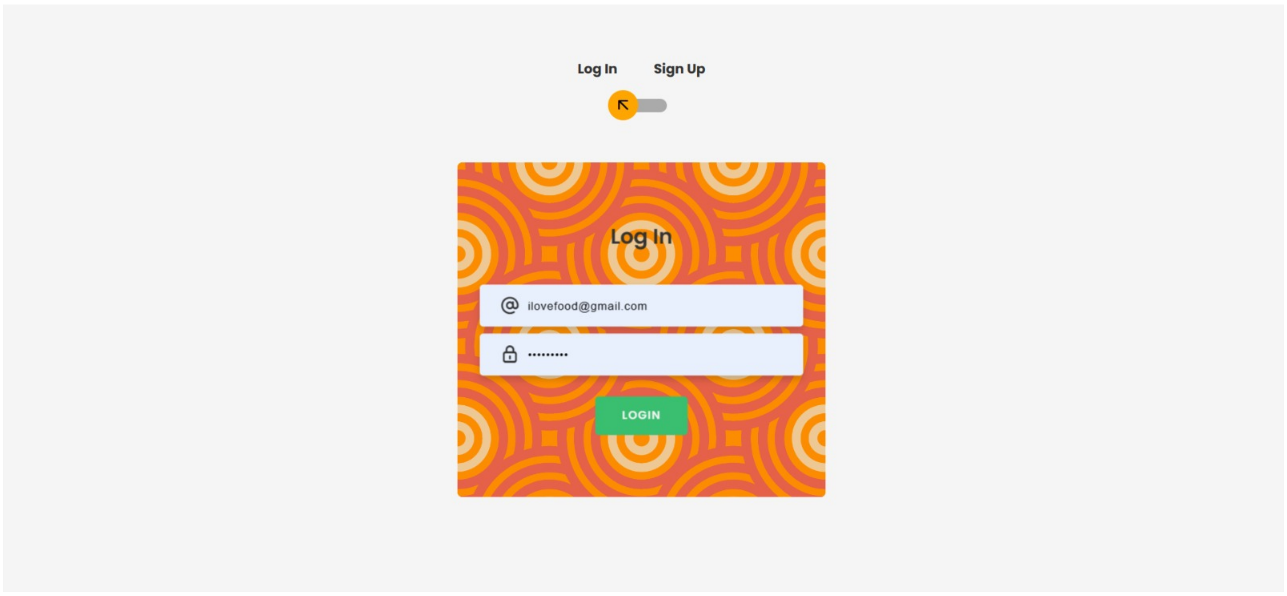


Figure 8.1: Login/Sign-up page

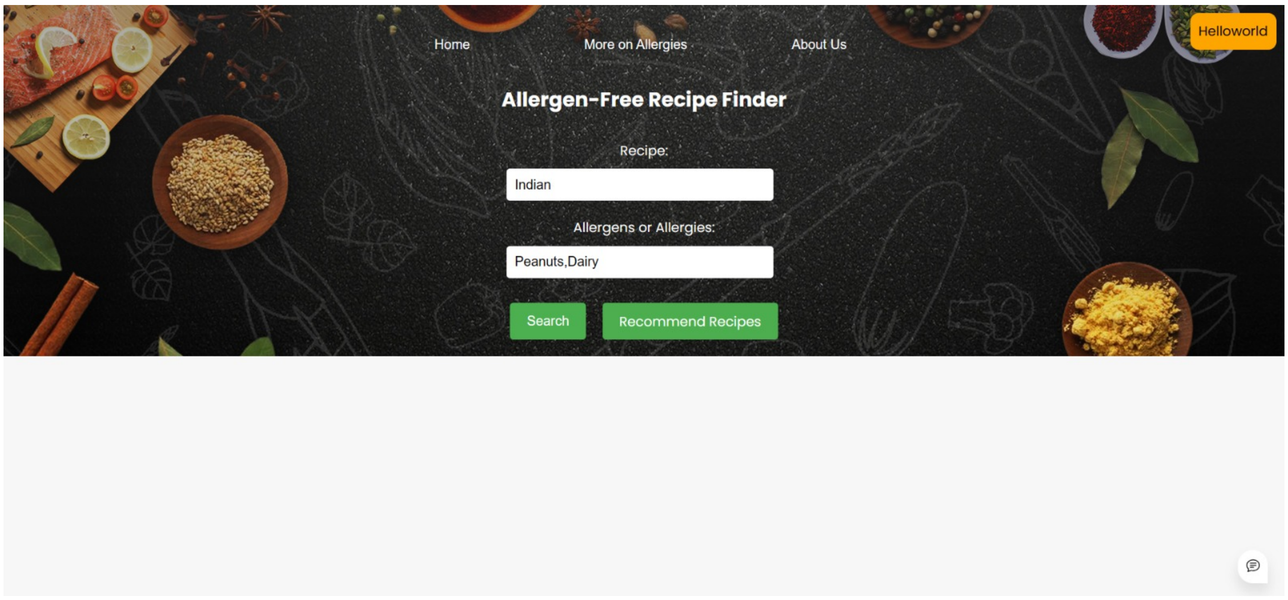


Figure 8.2: Home Page

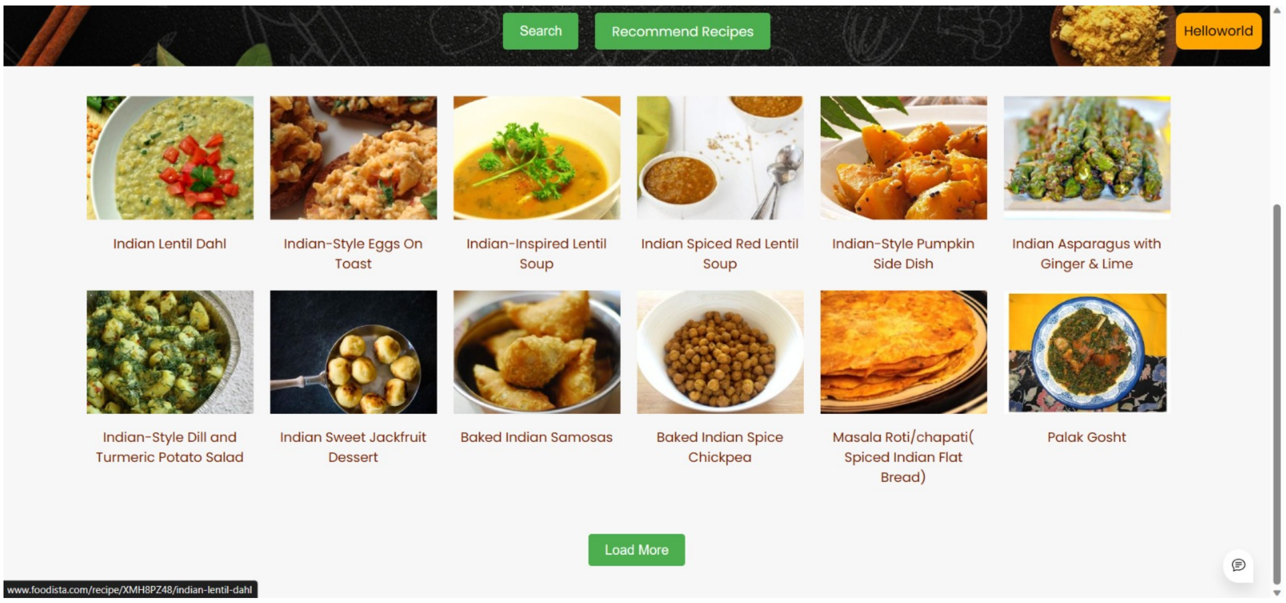

Search
Recommend Recipes
Helloworld
Indian Lentil Dahl
Indian-Style Eggs On Toast
Indian-Inspired Lentil Soup
Indian Spiced Red Lentil Soup
Indian-Style Pumpkin Side Dish
Indian Asparagus with Ginger & Lime
Indian-Style Dill and Turmeric Potato Salad
Indian Sweet Jackfruit Dessert
Baked Indian Samosas
Baked Indian Spice Chickpea
Masala Roti/chapati(Spiced Indian Flat Bread)
Palak Gosht
Load More
www.foodista.com/recipe/XMH8PZ48/indian-lentil-dahl

Figure 8.3: Search Result

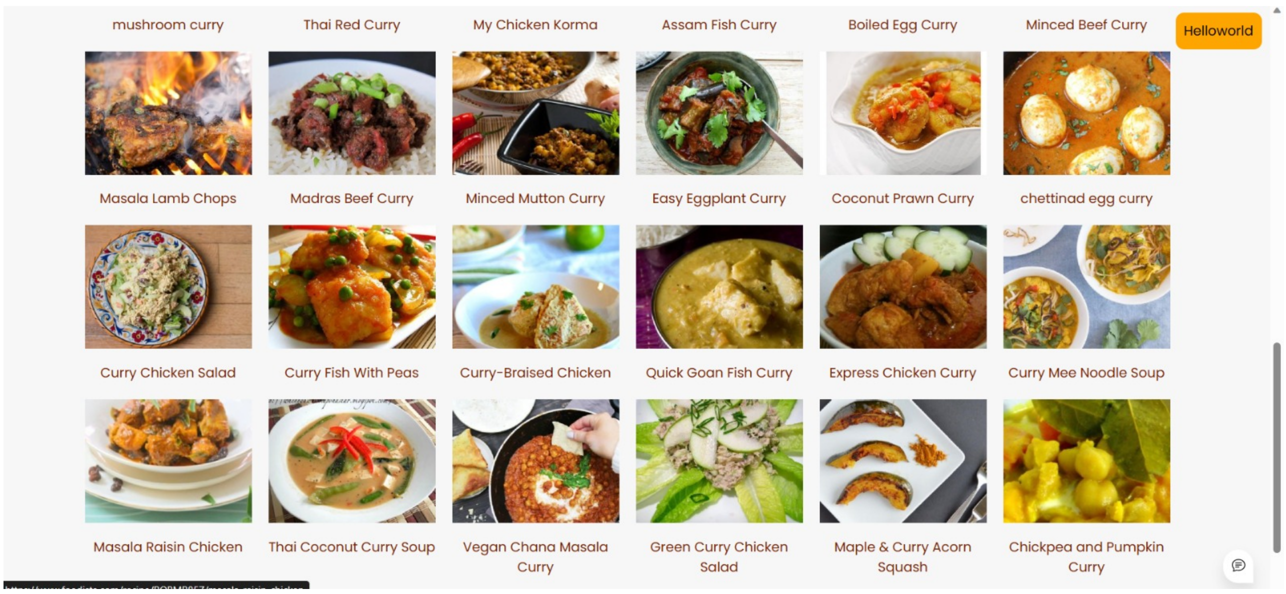


Figure 8.4: Load More Recipes

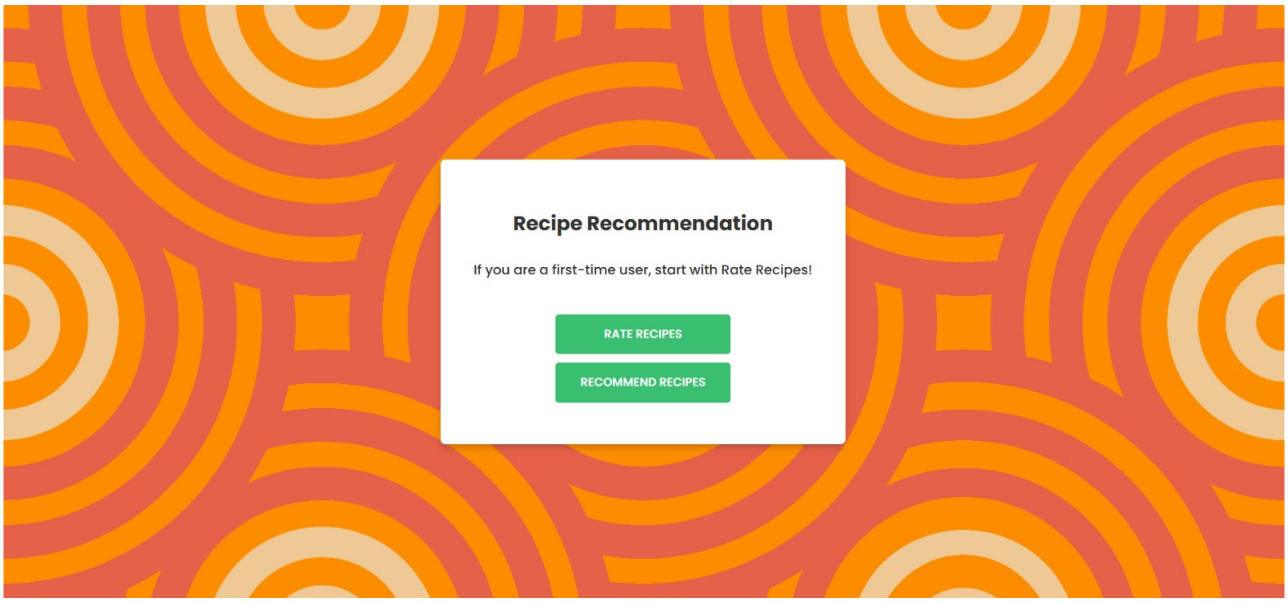
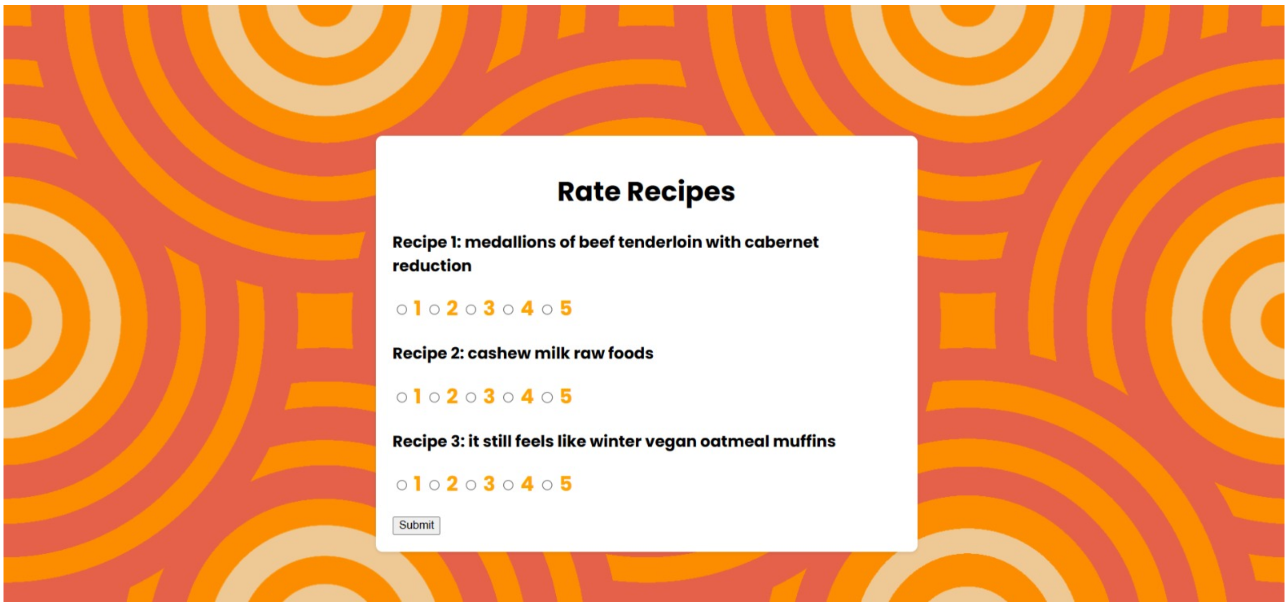


Figure 8.5: Rate/ Recommend Personalized Recipes



Rate Recipes

Recipe 1: medallions of beef tenderloin with cabernet reduction

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

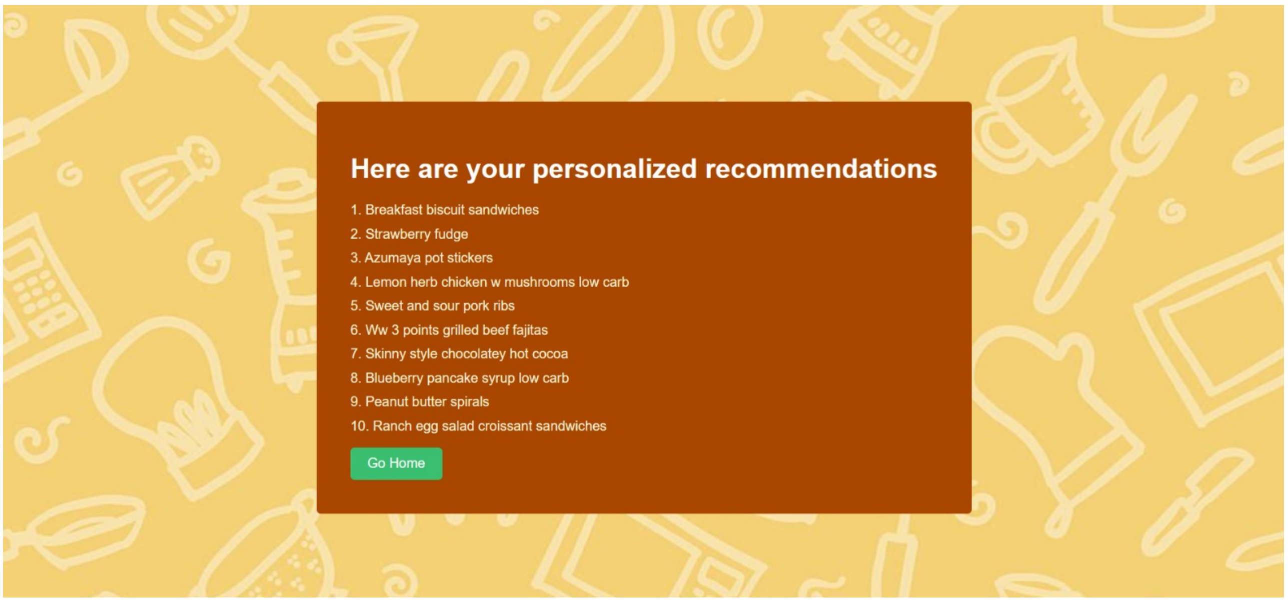
Recipe 2: cashew milk raw foods

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Recipe 3: it still feels like winter vegan oatmeal muffins

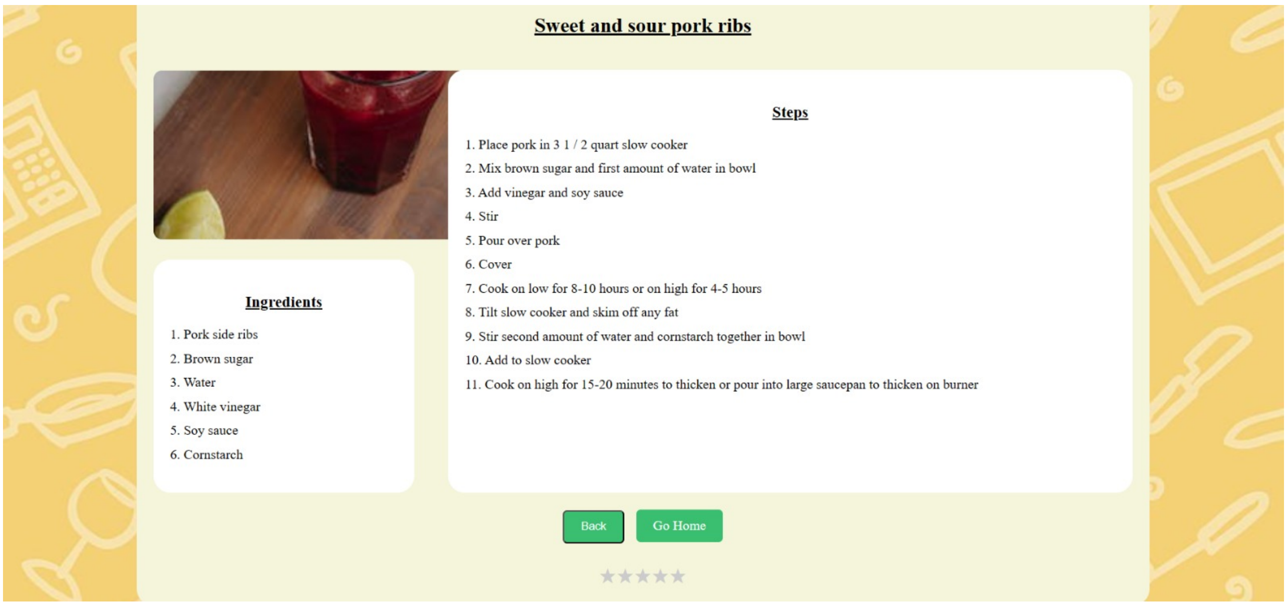
○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Figure 8.6: Rate the recipes to help the ML model learn user's preferences


Here are your personalized recommendations

1. Breakfast biscuit sandwiches
2. Strawberry fudge
3. Azumaya pot stickers
4. Lemon herb chicken w mushrooms low carb
5. Sweet and sour pork ribs
6. Ww 3 points grilled beef fajitas
7. Skinny style chocolatey hot cocoa
8. Blueberry pancake syrup low carb
9. Peanut butter spirals
10. Ranch egg salad croissant sandwiches

Figure 8.7: List of recommended recipes


Sweet and sour pork ribs

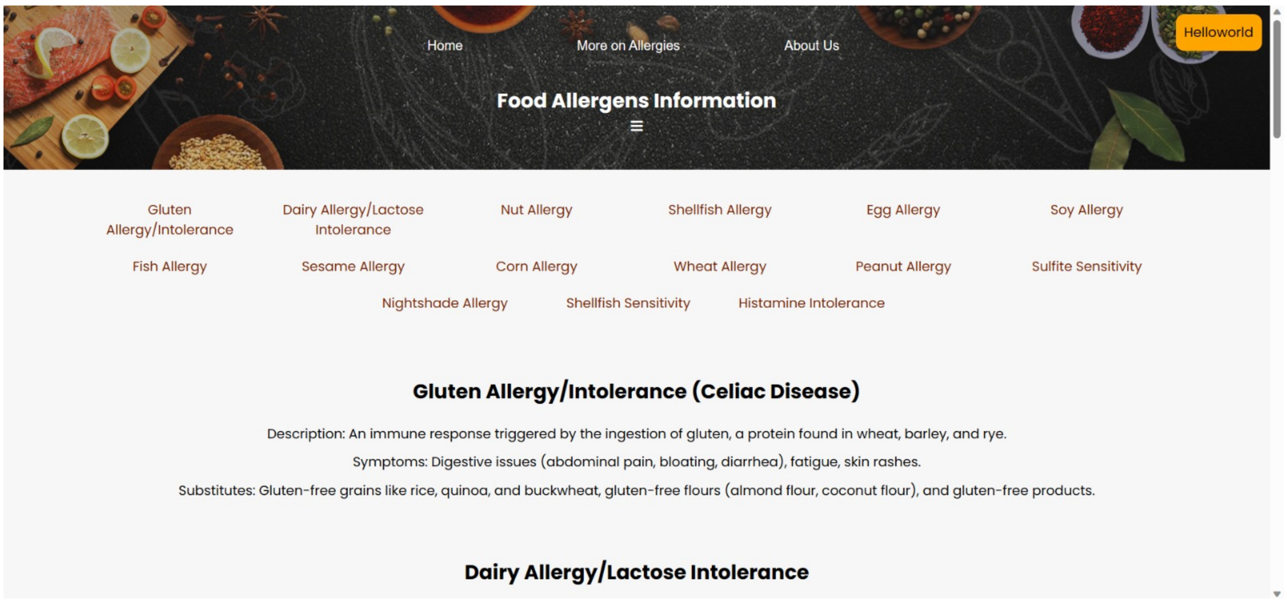
Ingredients

1. Pork side ribs
2. Brown sugar
3. Water
4. White vinegar
5. Soy sauce
6. Cornstarch

Steps

1. Place pork in 3 1 / 2 quart slow cooker
2. Mix brown sugar and first amount of water in bowl
3. Add vinegar and soy sauce
4. Stir
5. Pour over pork
6. Cover
7. Cook on low for 8-10 hours or on high for 4-5 hours
8. Tilt slow cooker and skim off any fat
9. Stir second amount of water and cornstarch together in bowl
10. Add to slow cooker
11. Cook on high for 15-20 minutes to thicken or pour into large saucepan to thicken on burner

Figure 8.8: Recommended recipe page


Home More on Allergies About Us

Helloworld

Food Allergens Information

Gluten Allergy/Intolerance | Dairy Allergy/Lactose Intolerance | Nut Allergy | Shellfish Allergy | Egg Allergy | Soy Allergy | Fish Allergy | Sesame Allergy | Corn Allergy | Wheat Allergy | Peanut Allergy | Sulfite Sensitivity | Nightshade Allergy | Shellfish Sensitivity | Histamine Intolerance

Gluten Allergy/Intolerance (Celiac Disease)

Description: An immune response triggered by the ingestion of gluten, a protein found in wheat, barley, and rye.

Symptoms: Digestive issues (abdominal pain, bloating, diarrhea), fatigue, skin rashes.

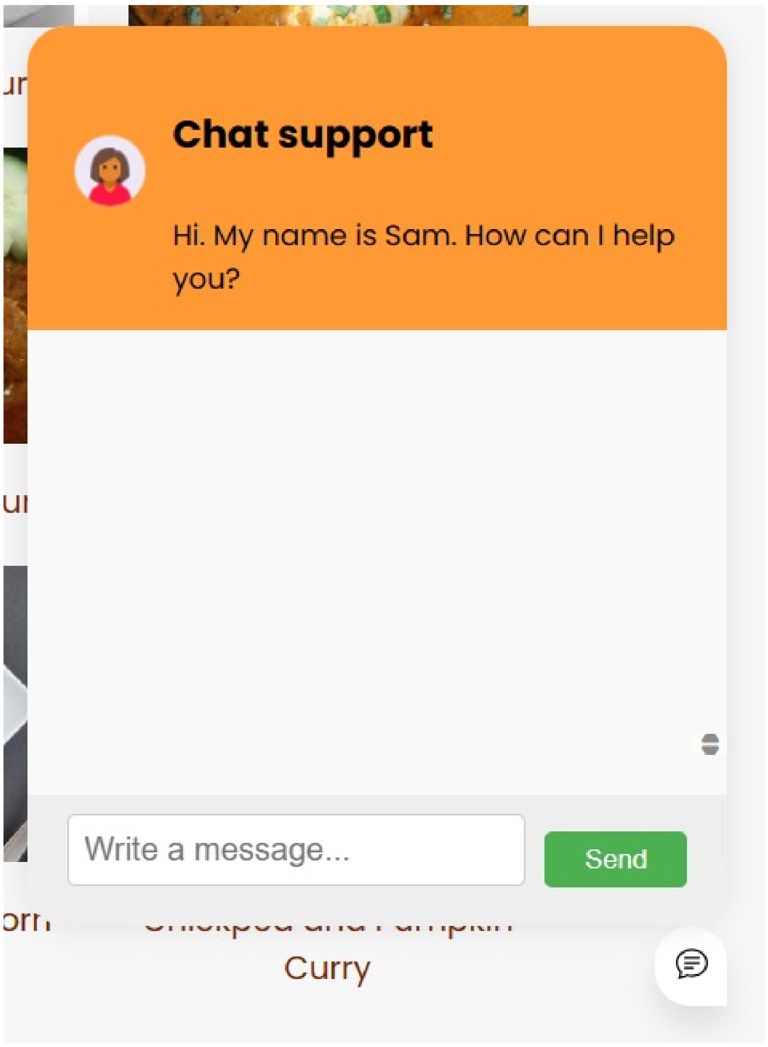
Substitutes: Gluten-free grains like rice, quinoa, and buckwheat, gluten-free flours (almond flour, coconut flour), and gluten-free products.

Dairy Allergy/Lactose Intolerance

Figure 8.9: Information on different allergies

Cooking tasty recipes for you
This might take a few minutes :D

Figure 8.10: Loading Page


Chat support

Hi. My name is Sam. How can I help you?

Figure 8.11: AI Chatbot that answers food-related queries

Chapter 9

Risks and Challenges

1. Data Accuracy: Ensuring the accuracy and reliability of the recipe data, allergen information, and ingredient databases can be a challenge. Inaccurate or outdated data can lead to incorrect recipe recommendations and potential health risks for users with allergies.

2. Allergen Identification: Identifying and categorizing allergens accurately can be challenging as allergies can vary from person to person. Ensuring that the website covers a wide range of common allergens and provides options for users to specify their specific allergies accurately is important.

3. Recipe Filtering: Implementing an effective recipe filtering mechanism to exclude allergens and specific ingredients can be complex. It requires integrating with reliable recipe APIs that provide comprehensive allergen and ingredient data and ensuring that the filtering process is robust and accurate.

4. User Engagement and Personalization: Encouraging user engagement and providing personalized recipe recommendations requires a well-designed user interface, intuitive user experience, and effective communication. It can be a challenge to keep users actively involved and continuously learn their preferences to provide relevant and appealing recipe suggestions.

5. Integration with APIs: Integrating and relying on external APIs, such as the recipe API, may introduce challenges such as rate limits, data inconsistencies, and potential API changes or deprecations. Regularly monitoring and adapting to changes in the APIs is essential to maintain the website's functionality.

6. Privacy and Data Security: Handling user data, such as allergy information and user preferences, requires careful consideration of privacy and data security measures.

7. Accessibility and Cross-Browser Compatibility: Ensuring the website is accessible to users with different abilities and compatible with various web browsers and devices can be challenging. It requires adhering to web accessibility standards and thoroughly testing the website across different platforms to provide a consistent user experience.

8. Performance and Scalability: As the website grows and handles increasing traffic and user interactions, performance and scalability become important considerations. Optimizing the website's performance, implementing caching strategies, and scaling the infrastructure appropriately are necessary to provide a smooth user experience.

9. Regulatory Compliance: Depending on the region or jurisdiction, there may be specific regulations and guidelines related to handling allergen information, user data, and food-related websites. Ensuring compliance with applicable regulations, such as data protection laws and food safety regulations, is crucial.

Chapter 10

Conclusion

In conclusion, the recipe search and allergen exclusion web application provides a valuable resource for users to discover a wide range of recipes based on their preferences and dietary restrictions. The implementation of the Spoonacular API allows for efficient recipe retrieval, while the allergen exclusion feature ensures that users can find recipes that align with their specific dietary needs.

The user-friendly interface, modern design, and responsive layout enhance the overall user experience, making it easy for users to explore and access recipes on various devices. The inclusion of the chatbox feature further enhances user engagement, providing personalized support and assistance.

10.1 Future Scope

While the current implementation of the recipe search and allergen exclusion application is already quite functional and user-friendly, there are several exciting future scopes to enhance and expand the project further:

1. User Accounts and Personalization: Introduce user accounts to allow users to save favorite recipes, create meal plans, and receive personalized recipe recommendations based on their preferences and dietary restrictions.

2. Advanced Filtering and Nutrition Information: Implement advanced filtering options, such as dietary preferences and cooking time, and integrate a nutrition API to provide detailed nutritional information for each recipe.

3. Meal Planning and Shopping Integration: Develop a meal planning feature that suggests recipes for weekly meal plans and enables users to add recipe ingredients directly to their shopping carts for easy grocery shopping.

4. Mobile App Development: Create a mobile app version of the web application to reach a broader audience and provide a seamless cooking experience on mobile devices.

5. Social Sharing and Recipe Reviews: Add social media sharing buttons to allow users to share their favorite recipes with friends and family, and incorporate a rating and review system to provide helpful feedback and social proof.

6. Community Forum and Video Tutorials: Establish a community forum where users can interact, share cooking tips, and seek advice. Additionally, include video tutorials for more complex recipes to guide users through the cooking process.

7. Allergy Alerts and Integration with Allergy Databases: Implement allergy alerts to notify users if a new recipe contains any of their allergens, and consider integrating with allergy databases to expand the allergen exclusion feature.

8. Internationalization and Multilingual Support: Provide support for multiple languages and international cuisines to cater to a global audience.

References

[1] G. Shani and A. Gunawardana, “Evaluating Recommendation Systems,” in Recommender Systems Handbook,Boston, MA: Springer US, 2011, (pp. 16–38).

[2] ”A Brief Analysis of Collaborative and Content Based Filtering Algorithms used in Recommender Systems”, Sri Hari Nallamala, Usha Rani Bajjuri, Sarvani Anandarao, Dr. D. Durga Prasad and Dr. Pragnaban Mishra.

[3] ”Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System” Poonam B. Thorat, R. M. Goudar and Sunita Barve in International Journal of Computer Applications (0975 – 8887) Volume 110 – No. 4, January 2015 (pp. 32-35)

[4] ”AlleRT: Food Recommender Web Application with Allergy Filtration”, Mary Jane Samonte, Zowee Gabrielle De Vera, Carlos Jaime Ruiz Jr., Clarisse Francesca Sunga, Danielle Samonte in 5th European International Conference on Industrial Engineering and Operations Management Rome, Italy, July 26-28, 2022

[5] Ho, C.-S. and Chang, Y.-M. (2018, December 22-23)*. ”Design and Implementation of Intelligent Personalized Dietary Meal Recommendation System.” In Proceedings of the International Conference on CCME (pp. 137-140).

[6] Food Recommender Systems: Important Contributions, Challenges and Future Research Directions Christoph Trattner, David Elsweiler

[7] Toledo, R.Y., Alzahrani, A.A. and Martinez, L., 2019. A food recommender system considering nutritional information and user preferences. IEEE Access, 7, pp.96695-96711, 2019. (pp. 4-5)

COURSE OUTCOMES:

After completion of the course the student will be able to

SL. NO	**DESCRIPTION**	**Blooms' Taxonomy Level**
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
CO2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
CO3	3	3	3	3	3	2	2	3	2	2	2	3			2
CO4	2	3	2	2	2			3	3	3	2	3	2	2	2
CO5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
100003/CS622T.1-PO1	**HIGH**	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.1-PO2	**HIGH**	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
100003/CS622T.1-PO3	**HIGH**	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
100003/CS622T.1-PO4	**HIGH**	Identify technically and economically feasible problems by analysis and interpretation of data.
100003/CS622T.1-PO6	**MEDIUM**	Responsibilities relevant to the professional engineering practice by identifying the problem.
100003/CS622T.1-PO7	**MEDIUM**	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
100003/CS622T.1-PO8	**HIGH**	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
100003/CS622T.1-PO9	**MEDIUM**	Identify technically and economically feasible problems by working as a team.
100003/CS622T.1-PO10	**MEDIUM**	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
100003/CS622T.1-P011	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
100003/CS622T.1-PO12	**HIGH**	Identify technically and economically feasible problems for long term learning.
100003/CS622T.1-PSO1	**MEDIUM**	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
100003/CS622T.1-PSO2	**MEDIUM**	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
100003/CS622T.1-PSO3	**MEDIUM**	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
100003/CS622T.2-PO1	**HIGH**	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

100003/CS622T.2-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
100003/CS622T.2-PO3	**HIGH**	Design solutions for complex engineering problems and design based on the relevant literature.
100003/CS622T.2-PO4	**HIGH**	Use research-based knowledge including design of experiments based on relevant literature.
100003/CS622T.2-PO5	**HIGH**	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
100003/CS622T.2-PO6	**MEDIUM**	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
100003/CS622T.2-PO8	**HIGH**	Apply ethical principles and commit to professional ethics based on the relevant literature.
100003/CS622T.2-PO9	**MEDIUM**	Identify and survey the relevant literature as a team.
100003/CS622T.2-PO10	**HIGH**	Identify and survey the relevant literature for a good communication to the engineering fraternity.
100003/CS622T.2-PO11	**MEDIUM**	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
100003/CS622T.2-PO12	**HIGH**	Identify and survey the relevant literature for independent and lifelong learning.
100003/CS622T.2-PSO1	**MEDIUM**	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
100003/CS622T.2-PSO2	**MEDIUM**	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
100003/CS622T.2-PSO3	**MEDIUM**	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
100003/CS622T.3-PO1	**HIGH**	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
100003/CS622T.3-PO2	**HIGH**	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

100003/CS622T.3-PO3	**HIGH**	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO4	**HIGH**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.3-PO5	**HIGH**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
100003/CS622T.3-PO6	**MEDIUM**	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
100003/CS622T.3-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PO8	**HIGH**	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
100003/CS622T.3-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.3-PO10	**MEDIUM**	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
100003/CS622T.3-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
100003/CS622T.3-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
100003/CS622T.3-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
100003/CS622T.4-PO1	**MEDIUM**	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.4-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

100003/CS622T.4-PO3	**MEDIUM**	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
100003/CS622T.4-PO4	**MEDIUM**	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
100003/CS622T.4-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
100003/CS622T.4-PO8	**HIGH**	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/CS622T.4-PO9	**HIGH**	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
100003/CS622T.4-PO10	**HIGH**	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
100003/CS622T.4-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work by prepare technical report and deliver presentation.
100003/CS622T.4-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
100003/CS622T.4-PSO1	**MEDIUM**	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
100003/CS622T.4-PSO2	**MEDIUM**	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
100003/CS622T.4-PSO3	**MEDIUM**	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
100003/CS622T.5-PO1	**HIGH**	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/CS622T.5-PO2	**HIGH**	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PO3	**HIGH**	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
100003/CS622T.5-PO4	**MEDIUM**	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/CS622T.5-PO5	**MEDIUM**	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO6	**MEDIUM**	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO7	**MEDIUM**	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO8	**HIGH**	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO9	**MEDIUM**	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO11	**MEDIUM**	Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PO12	**HIGH**	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO1	**MEDIUM**	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

100003/CS622T.5-PSO2	**MEDIUM**	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
100003/CS622T.5-PSO3	**MEDIUM**	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.