Industry Oriented Mini Project Report

AI-AUTOMATED RECIPE GENERATOR

A dissertation submitted in partial fulfillment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

D. Laxmi Prasanna- 22B81A1284

Yasa Nikitha Reddy- 22B81A1293

T. Sri Varsha- 22B81A12B7

Under the esteemed guidance of

C.V.S Satya Murty

Associate Professor, IT Department

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH,

Accredited by NBA, and NAAC)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Ranga Reddy (Dist.) - 501510, Telangana State.

2024-2025

Cherabuddi Education Society's
CVR COLLEGE OF ENGINEERING
(An Autonomous Institution)
ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE
(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)
Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510
Web : http://cvr.ac.in, email : info@cvr.ac.in
Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Project Report entitled **"AI-Automated Recipe Generator"** is a bonafide work done and submitted by **D. Laxmi Prasanna(22B81A1284)**, **Yasa Nikitha Reddy(22B81A1293),T. Sri Varsha (22B81A12B7)** during the academic year 2024-2025, in partial fulfillment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to the best of my knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE

C.V.S Satya Murty

Associate Professor, IT Department

HEAD OF THE DEPARTMENT

Dr. Bipin Bihari Jayasingh

Professor, IT Department

PROJECT COORDINATOR

Ms. S. Suhasini

Assistant Professor, IT Department

EXTERNAL EXAMINER

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We hereby declare that the project report entitled **"AI-Automated Recipe Generator"** is an original work done and submitted to the IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfillment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **C.V.S Satya Murty**, Associate Professor, **Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

D. Laxmi Prasanna – 22B81A1284

Yasa Nikitha Reddy - 22B81A1293

T. Sri Varsha - 22B81A12B7

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, and Vice Principal for availing infrastructural facilities to complete the major project in time.

We offer our sincere gratitude to our internal guide **Mr. C.V.S Satya Murty,** Associate Professor , IT Department, CVR College of Engineering for his immense support, timely cooperation, and valuable advice throughout our project work.

We would like to thank of Project In-Charge **Mr. K. Veeranjaneyulu** Sr. Assistant Professor, IT Department, CVR College of Engineering for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to Project Coordinator**, Ms. S. Suhasini**, Assistant Professor, IT Department, CVR College of Engineering for her supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We also thank the **Project Review Committee Members** for their valuable suggestions.

D. Laxmi Prasanna(22B81A1284)

Yasa Nikitha Reddy (22B81A1293)

T. Sri Varsha (22B81A12B7)

ABSTRACT

The AI Automated Recipe Generator is a web-based platform that simplifies recipe discovery using artificial intelligence. Users can input available ingredients, and the AI suggests personalized recipes. The platform features an intuitive interface where users can select ingredients through clickable images or a search bar. Once the ingredients are entered, the AI processes the data to provide a list of recipe suggestions with detailed instructions, estimated cooking times, and nutritional information. Users can filter recipes by dietary preferences, cuisines, or cooking time. Additional features include saving favourite recipes, sharing options, and a feedback system for reviews. The responsive design ensures the platform works seamlessly on mobile and desktop devices. This project is designed to enhance creativity in the kitchen and save time for home cooks. By utilizing AI, it makes meal planning more efficient and convenient. The platform aims to offer a fun and easy way for food enthusiasts to explore new recipes based on what they have at home.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

The **AI Automated Recipe Generator** is an innovative web-based platform designed to simplify meal planning by intelligently recommending recipes based on user-provided ingredients and preferences. By integrating artificial intelligence, it offers a personalized cooking experience, helping users discover dishes they can make with what they already have at home. The system caters to a wide range of users, including busy professionals, home cooks, and those with specific dietary needs.

Users can input ingredients through a search bar or select them from a gallery of images. The backend AI then analyzes this input along with user preferences—such as dietary restrictions, allergies, and favorite cuisines—to generate a list of suitable recipes with preparation steps, cooking times, and nutritional information.

The platform is developed using Flask for the backend, MongoDB Atlas for cloud-based storage, and HTML, CSS, and JavaScript for the frontend. It incorporates a large-scale recipe dataset with over two million entries to provide variety in recommendations. Additional features like recipe saving, sharing, and user feedback enhance the overall experience, making cooking smarter, faster, and more enjoyable.

Advantages:

AI-Automated Recipe Generator offers several advantages to users:

- Personalized Recommendations – Recipes are tailored based on user preferences, ingredients, allergies, and dietary restrictions.
- Time-Saving – Quickly generates recipe ideas using available ingredients, reducing time spent planning meals.
- Reduces Food Waste – Encourages users to use leftover or available ingredients efficiently.
- User-Friendly Interface – Simple, intuitive design with ingredient selection through images or search input.

- Wide Recipe Variety – Access to a large dataset of over 2 million recipes ensures rich and diverse suggestions.
- Cuisine Filtering – Recipes can be filtered by cuisine types or meal categories (e.g., breakfast, lunch, dinner).
- Responsive Design – Fully functional across desktop, tablet, and mobile devices.
- Favorite & Feedback Options – Users can save favorite recipes, rate them, and share feedback to improve suggestions.
- User Profiles – Stores personal preferences, allergies, and dietary needs for better personalization.
- Allergy Removal Feature – Filters out recipes that contain allergens specified by the user, ensuring safe suggestions.
- AI-Driven Intelligence – Uses artificial intelligence to process data and deliver smart, relevant recipe results.

Applications:

- Smart Meal Planning – Assists users in planning daily or weekly meals based on the ingredients they already have, saving time and effort.
- Grocery Optimization – Helps reduce food waste and avoid overbuying by suggesting recipes that utilize leftover or expiring ingredients.
- Diet Management – Supports users following specific diets such as vegetarian, vegan, keto, gluten-free, or diabetic-friendly plans by filtering recipes accordingly.
- Allergy-Safe Cooking – Automatically removes recipes containing allergens like nuts, dairy, or gluten, ensuring health and safety for sensitive users.
- Mobile Cooking Companion – The responsive design allows users to access recipe suggestions on their phones or tablets while in the kitchen or at the store.
- Educational Tool for Schools and Colleges – Can be used to demonstrate how AI and machine learning can be applied in real-life scenarios such as food tech, data filtering, and personalization.
- Support for Beginners – Offers step-by-step cooking instructions, estimated time, and nutritional details, making it ideal for those new to cooking.
- Family Meal Planning – Provides recipes suitable for all age groups, supporting families in creating nutritious and enjoyable meals together.

- Cultural Exploration – Encourages users to explore recipes from different regions and cultures, broadening culinary knowledge and experiences.
- AI Integration Showpiece – Demonstrates the practical use of AI in filtering, recommending, and adapting content to user behavior in a web-based environment.
- Recipe Discovery Platform – Acts as a creative source for discovering new dishes users might not have tried otherwise.
- Social Sharing and Engagement – Users can share recipes with friends or post them online, promoting food communities and engagement.
- Health and Nutrition Guidance – Supplies basic nutritional information with each recipe, helping users make healthier choices.
- Modular AI Project – Serves as a base for future integrations like calorie tracking, meal scheduling, voice commands, or IoT-based smart kitchen tools.

1.2 MOTIVATION

AI Automated Recipe Generator applications are motivated by the desire to offer users a smart and efficient solution to everyday cooking challenges. By enabling personalized recipe suggestions through ingredient input and user preferences, these applications prioritize convenience, creativity, and healthier eating habits. With features such as AI-based recommendations, allergy filtering, and user-friendly interfaces, they simplify the process of meal planning and promote better food utilization. Additionally, the time-saving benefits, reduced food waste, and enhanced user satisfaction drive the ongoing development and adoption of such platforms in modern kitchens.

Challenges:

- Data Accuracy and Quality: Ensuring that the massive recipe dataset is clean, accurate, and consistent is critical for delivering relevant and reliable suggestions. Incomplete or incorrect data can lead to poor user experience.
- AI Recommendation Accuracy: Developing an AI system that consistently recommends meaningful and context-aware recipes based on limited ingredients and user preferences is complex and requires continuous tuning.

- User Preference Handling: Accurately interpreting and incorporating individual user preferences, such as dietary restrictions, allergies, and cuisine choices, remains a challenging aspect of personalization.
- Ingredient Recognition: Effectively recognizing and matching user-inputted or selected ingredients with the dataset can be difficult, especially with varied naming conventions, misspellings, or uncommon items.
- Responsive Design and Cross-Device Compatibility: Creating a seamless user interface that works smoothly across desktops, tablets, and mobile devices can be challenging, especially when handling interactive elements like image-based selection.
- Performance and Speed: Processing large-scale data and generating real-time recipe suggestions without noticeable delays requires optimized backend systems and efficient data handling strategies.
- Security and Privacy: As the application handles user profiles, preferences, and possibly sensitive dietary information, implementing strong data protection and privacy measures is essential.
- Scalability: As the user base grows, maintaining consistent performance and responsiveness while scaling backend services and databases becomes increasingly challenging.
- Feedback Integration: Collecting and effectively utilizing user feedback to improve the recommendation engine and user experience requires thoughtful design and regular updates.

Overcoming these challenges requires continuous improvements in AI accuracy, data quality, and user preference handling. Enhancing ingredient recognition, ensuring strong data security, and optimizing performance for real-time results are key.

1.3 PROBLEM STATEMENT

- The platform relies heavily on internet connectivity, which may hinder accessibility in areas with poor or no network coverage.
- Ingredient recognition may be affected by inconsistent naming conventions, typos, or uncommon items, impacting the accuracy of recipe suggestions.
- The AI model's effectiveness depends on the quality and completeness of the dataset, which may contain outdated or irrelevant recipes.

- Users with limited technological literacy may find it challenging to interact with features such as image-based selection or preference settings.
- Personalized recommendations may not always align perfectly with user expectations due to the complexity of dietary needs and preferences.
- The current system may struggle to recommend recipes for very limited or highly specific ingredient combinations.
- Frequent updates and ongoing model training are required to maintain the relevance and accuracy of recipe suggestions.
- Some users may have concerns regarding data privacy, especially when storing dietary habits, allergies, or preferences.
- Limited language support may restrict usability for non-English-speaking users.
- Users without access to modern devices may experience reduced performance or limited functionality on older browsers or phones.

1.4 PROJECT OBJECTIVES

- Personalized Recipe Recommendations: Provide users with recipe suggestions tailored to their available ingredients, dietary preferences, allergies, and cooking habits using advanced AI algorithms.
- Efficient Ingredient Utilization: Help users make the most of the ingredients they have at home, thereby reducing food waste and encouraging sustainable cooking practices.
- Enhanced User Experience: Deliver an intuitive and interactive user interface that allows for easy ingredient selection through images or search bars, ensuring accessibility for users of all tech backgrounds.
- Dietary and Allergy Awareness: Integrate filters that accommodate various dietary needs and exclude ingredients linked to allergies to promote safe and healthy eating.
- Real-Time AI Processing: Implement a fast and responsive AI engine that can process user inputs and return accurate recipe suggestions instantly.
- Comprehensive Recipe Database Access: Maintain and utilize a large, diverse dataset of recipes to ensure variety, uniqueness, and global cuisine representation in suggestions.
- Cross-Platform Compatibility: Ensure that the platform operates smoothly across all devices—desktop, tablet, and mobile—for user convenience.

- User Profile Management: Allow users to save preferences, favorite recipes, and recently viewed items to personalize their cooking experience and increase engagement.
- Data Security and Privacy: Implement secure data handling practices to protect user information, especially dietary preferences and personal data.
- Feedback and Improvement: Collect user feedback and interaction data to continuously refine the AI model, improve suggestions, and enhance overall service quality.

1.5 PROJECT REPORT ORGANIZATION

Organizing a project report effectively is crucial for communicating its objectives, progress, findings, and recommendations clearly to stakeholders. Structure of organizing a project report:

- **Title Page**: Contains the project title, names of the team members, institution or organization name, and submission date.
- **Table of Contents**: Lists all main sections and subsections with page numbers for easy navigation.
- **Executive Summary**: A concise overview of the entire project including objectives, methods, key findings, conclusions, and recommendations.
- **Introduction**: Introduces the project's purpose, background, scope, and goals, explaining why the project was undertaken.
- **Methodology**: Describes the tools, techniques, and data used, including how information was collected and analyzed.
- **Findings**: Presents the outcomes of the project, supported by visuals like charts or tables when applicable.
- **Analysis**: Explains the meaning behind the findings, discussing trends, challenges, or key insights.
- **Recommendations**: Offers practical suggestions based on the analysis to improve or build upon the project.
- **Conclusion**: Summarizes results and reiterates the project's objectives, highlighting its impact or value.
- **References**: Lists all sources used, following a standard citation style.
- **Appendices**: Includes additional supporting materials such as raw data or extra visuals not covered in the main sections.

CHAPTER-2

LITERATURE SURVEY

2.1 OVERVIEW

This section presents a comprehensive review of the existing literature on AI-driven recipe recommendation systems, with a focus on personalization, ingredient-based filtering, and user-centric design. It highlights the foundations on which the current project builds and identifies opportunities for improvement.

Previous Research and Projects:

This subsection examines previous work done in the field of automated recipe recommendation systems. It includes an overview of AI and machine learning approaches used in similar platforms, such as collaborative filtering, content-based filtering, and hybrid recommendation systems.

Technologies and Methodologies:

This part reviews the tools, platforms, and algorithms commonly adopted in the development of recipe generators. Technologies like natural language processing (NLP), computer vision for image recognition, and large-scale dataset utilization are covered.

Gaps in Existing Solutions:

While several systems offer recipe suggestions, many lack a holistic view of user preferences and real-time ingredient availability. This section highlights the limitations in customization, user feedback integration, and accessibility.

2.2 PREVIOUS RESEARCH AND PROJECTS

Previous research in AI-based recipe recommendation systems has explored the use of **collaborative filtering**, **content-based filtering**, and **hybrid models** for personalizing meal suggestions. Platforms like **Yummly** and **Chef Watson** have utilized machine learning to match recipes with user preferences or ingredients.

While these systems offer a foundation, many lack real-time ingredient filtering, allergy detection, and adaptive personalization. This project aims to bridge those gaps by building a smarter, more responsive recipe generator.

2.3 TECHNOLOGIES AND METHODOLOGIES

The project integrates several technologies to enhance recipe recommendations. **Machine learning algorithms** such as content-based filtering are used to match user preferences with suitable recipes. **Natural Language Processing (NLP)** helps process user inputs and analyze ingredient lists, while **computer vision** may be used for identifying ingredients from images.

The system also uses **MongoDB** for flexible data storage, and **Flask** as the backend framework to manage API calls and user interaction. User preferences and allergies are collected through a responsive web interface and stored to personalize recipe results.

This combination of AI, database management, and web technologies ensures a dynamic and scalable platform for recipe generation.

2.4 GAPS IN EXISTING SOLUTIONS

While many existing recipe recommendation platforms offer search and filtering options, they often lack deep personalization based on real-time ingredient availability, allergy preferences, and dynamic user behavior. Most systems do not integrate ingredient-image selection, nor do they efficiently combine user profiles with AI-generated suggestions.

Additionally, limited support for dietary restrictions, poor handling of multilingual data, and lack of feedback-driven improvement mechanisms leave a gap in user experience. These limitations highlight the need for a more intuitive, adaptive, and user-aware platform—addressed by the AI-Automated Recipe Generator.

2.5 PREVIOUS RESEARCH AND PROJECTS

The development of AI-powered culinary applications has been driven by both academic research and industry innovations. This section outlines notable projects, methodologies, and findings that form the foundation for AI-based recipe generation systems.

2.5.1 IBM Chef Watson

IBM's Chef Watson is one of the earliest and most innovative AI systems designed for recipe generation.

- Objective: To inspire novel culinary creations by combining ingredients in ways that human chefs might not consider.
- Key Technologies:
    - Watson Cognitive Computing Platform
    - Natural Language Processing (NLP) for understanding recipe semantics.
    - Flavor Compound Data and Psychographic Models for recommending combinations based on taste, smell, and texture.
- Impact:
    - Demonstrated AI's potential in creativity and food science.
    - Resulted in the publication: *"Cognitive Cooking with Chef Watson: Recipes for Innovation from IBM & the Institute of Culinary Education"*.

2.5.2 Recipe1M & Recipe1M+ (MIT CSAIL & Facebook AI)

Recipe1M+ is the most widely used open-source dataset for training AI models on food data.

- **Details:**
    - Contains over 1 million recipes, 13 million food images.
    - Recipes include title, ingredients, instructions, and associated images.
- **Use Cases:**
    - Training deep learning models for food classification, multimodal retrieval, and generation.

- **Publications:**
    - *"Learning Cross-Modal Embeddings for Cooking Recipes and Food Images"* (Salvador et al., CVPR 2017)
- **Impact:**
    - Set the benchmark for multimodal food AI research.

2.5.3 Im2Recipe (MIT & Facebook AI)

Im2Recipe explored the retrieval of recipes from food images using joint embeddings of text and images.

- **Architecture:**
    - CNNs for image feature extraction.
    - LSTM/GRU or Transformer-based RNNs for recipe text encoding.
    - Triplet loss to align embeddings between visual and textual space.
- **Outcome:**
    - Enabled reverse lookup: users upload an image and retrieve a recipe.
    - Foundation for food image recognition systems.

2.5.4 Neural Recipe Generation (Microsoft Research)

Kiddon et al. (2016) introduced **Neural Checklist Models** for generating procedural text like cooking instructions.

- **Key Ideas:**
    - Generating coherent, step-by-step instructions by tracking ingredient use.
    - Prevents ingredient repetition or omission.
- **Applications:**
    - Useful in generating human-like and logically ordered recipe instructions.

- **Publication:** *"Globally Coherent Text Generation with Neural Checklist Models"*

2.5.5 FlavorGraph (UIUC & ZJU)

FlavorGraph is a knowledge graph that connects ingredients based on **flavor molecules** and **contextual co-occurrence** in recipes.

- **Highlights:**
    - Contains over 16,000 ingredients and 1 million flavor-compound relationships.
    - Enables recommendation of ingredient substitutes and creative pairings.
- **Applications:**
    - Used for recipe completion and novel recipe suggestion.
    - Supports semantic and flavor-based reasoning.

2.5.6 FoodKG (Fraunhofer IAIS)

FoodKG is a **semantically enriched food knowledge graph**, built to support personalized diet planning and food recommendation.

- **Key Features:**
    - Ontologies of food items, nutrition facts, allergies, and cooking processes.
    - Links to external sources like DBpedia, Wikidata, and USDA.
- **Use Cases:**
    - Reasoning about allergies or dietary restrictions.
    - AI-driven diet planners and food recommendation engines.

2.5.7 RecipeGPT & GPT-based Cooking Assistants

With the rise of large language models like GPT-2 and GPT-3, generative recipe systems have improved dramatically.

- **Projects:**
    - **RecipeGPT (2020)**: Fine-tuned GPT-2 on recipe datasets to generate cooking instructions based on ingredient lists or dish names.
    - **Custom GPT Bots**: Trained to answer food-related questions, recommend meals, or generate entire recipes from scratch.
- **Advantages:**
    - Human-like, context-aware recipe writing.
    - Integration with voice assistants and chatbots for interactive cooking guidance.

2.5.8 Generative Adversarial Networks (GANs) for Food

Some experimental systems use GANs to generate **recipe images**, **plated food visuals**, or even **entire menus**.

- **Example:**
    - **RecipeGAN**: Attempts to create synthetic recipe texts and visuals that mimic human-written content.
- **Application:**
    - Enhancing recipe databases, virtual cookbooks, or presentation design for chefs.

These research projects and systems have laid a strong foundation for developing intelligent, personalized, and creative recipe generators. The progression from rule-based systems to neural models, and now to large multimodal generative models, indicates a growing capacity for AI to assist in both everyday cooking and culinary innovation.

CHAPTER-3

SOFTWARE REQUIREMENTS AND SPECIFICATIONS

3.1 OVERVIEW

The AI-Automated Recipe Generator uses a client-server architecture consisting of a user-friendly frontend, a Flask-based backend, and a connected MongoDB database. The frontend allows users to input preferences and select ingredients. The backend processes these inputs using AI models to generate personalized recipe recommendations. The database stores user profiles, ingredients, and recipe data, enabling efficient and dynamic responses. This modular design ensures scalability, fast performance, and easy integration of new features.

3.2 PROPOSED ARCHITECTURE

The architecture of the AI-Automated Recipe Generator is designed to offer a smooth and personalized user experience by integrating multiple functional layers. These include the frontend, backend, database, AI recommendation engine, and content delivery systems. Each layer plays a distinct role in ensuring the efficiency and responsiveness of the application.

The **frontend layer**, built using HTML, CSS, and JavaScript, provides an intuitive user interface with pages such as the homepage, ingredients selection, recipe listings, login/profile, and individual recipe views. Users can search for recipes by selecting ingredients or browsing through recommended options, with interactive elements enhancing the overall usability.

The **backend layer**, developed with Flask, handles the logic behind user requests, manages routing, and connects the frontend with the database and AI modules. It processes ingredient inputs, handles user authentication, and fetches recipe data based on user interactions.

The **database layer** uses MongoDB to store user profiles, recipe data, ingredients, and preferences. Its flexible structure supports quick querying and updates, essential for delivering personalized content in real-time.

3.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the overall qualities or attributes that the Service Buddy platform must possess. These requirements ensure that the system is efficient, reliable, and user-friendly. Non-functional requirements are crucial for the success of the platform, as they impact the user experience, performance, and maintainability of the system

3.3.1 Performance

- The system should return personalized recipe recommendations within a few seconds of user input.
- Ingredient-based search and filtering must be optimized to handle large datasets without noticeable delays.
- The AI model should process and respond to user preferences and dietary restrictions quickly and accurately.
- The application must support concurrent users and maintain responsiveness under peak traffic.
- All database queries and backend processes should be optimized to minimize load times and improve user experience.
- Image loading and content rendering on the frontend should be seamless to ensure a smooth browsing experience.
- Performance metrics should be continuously monitored to identify bottlenecks and areas of improvement**.**

3.3.2 Reliability

- The system should operate consistently without failures, even during high demand or data-intensive operations.
- User data such as preferences, allergies, and saved recipes must be securely stored and reliably retrieved across sessions.
- AI-generated recommendations should maintain consistent accuracy and relevance across different use cases.
- Regular data backups should be scheduled to prevent loss of critical information in case of system failures.

- The system should implement error handling and fallback mechanisms to ensure minimal disruption in case of unexpected behavior.
- Scheduled maintenance or updates should not impact the availability of core services for users.
- The system should be tested extensively to ensure stable performance under various operating conditions.

3.3.3 Scalability

- The system should be capable of handling an increasing number of users without performance degradation.
- It must support the expansion of the recipe dataset as new recipes are added or updated regularly.
- The AI model should be designed to process larger and more complex inputs efficiently, such as combinations of rare ingredients or detailed user preferences.
- The backend infrastructure should be adaptable to support distributed computing for faster data processing and model inference.
- The architecture should allow seamless integration of new features, such as multilingual support or voice-based search, without requiring major redesign.
- The database should be optimized to manage growing amounts of structured and unstructured data, including user feedback and behavioral data.
- The platform should scale across devices (mobile, desktop, tablet) and regions to serve a global user base effectively.

3.3.4 Usability

- The user interface should be intuitive and easy to navigate, allowing users to quickly search for and generate recipes based on selected ingredients or preferences.
- The system should provide clear visual feedback for actions like ingredient selection, recipe suggestions, and saving favorites.
- Accessibility features should be included to support users with disabilities, such as screen reader compatibility and high-contrast modes.

- Users should be able to filter results based on dietary restrictions, cuisine type, and preparation time with minimal effort.
- Tooltips, hints, and guided walkthroughs can be incorporated to help new users understand how to use the platform effectively.
- Mobile responsiveness should be prioritized to ensure a seamless experience across different devices and screen sizes.
- The application should provide meaningful error messages and easy recovery options in case of invalid input or system issues.
- The design should focus on reducing user cognitive load by minimizing unnecessary steps and simplifying interactions.

3.3.5 Maintainability

- The system should be developed using modular and well-documented code to simplify future updates, bug fixes, and feature enhancements.
- A clear separation between frontend, backend, and database components should be maintained to allow independent modifications.
- Configuration files, environment variables, and external integrations should be structured to support easy changes without affecting core functionality.
- Version control (e.g., Git) must be used for tracking changes and enabling collaboration among developers.
- Logs and monitoring tools should be implemented to quickly identify and resolve issues that may arise during operation.
- Dependency management should be handled properly to ensure compatibility with updates and minimize conflicts.
- Automated testing and regular code reviews should be conducted to ensure stability and quality of the codebase.
- Documentation (both user-facing and developer-level) must be kept up-to-date to support onboarding and maintenance efforts.

3.3.6 Availability

- The system should be accessible to users at all times, ensuring minimal downtime and consistent availability of services like recipe recommendations, ingredient filtering, and user profile access.
- Cloud-based infrastructure or reliable hosting services should be used to guarantee high uptime and fault tolerance.
- Load balancing techniques can be employed to distribute traffic evenly and prevent service interruptions during peak usage.
- Scheduled maintenance should be planned during off-peak hours and communicated clearly to users.
- Redundancy mechanisms such as database replication and failover servers should be implemented to maintain continuous service in the event of hardware or software failures.
- Monitoring tools should be used to track system health and respond to availability issues in real time.

3.4 FUNCTIONAL REQUIREMENTS

The functional requirements define the core features and behaviors that the AI-Automated Recipe Generator must exhibit to meet user needs and system objectives. These requirements ensure that users can interact with the system to receive personalized, ingredient-based recipe suggestions efficiently.

3.4.1 User Registration and Login

- The system must provide a user-friendly registration form to collect essential user details (e.g., name, email, password).
- Users should be able to set preferences such as dietary restrictions and allergies during registration.
- A secure login mechanism should be implemented with proper encryption for password protection.
- The system must validate user credentials during login and grant access to personalized features.

- After logging in, users should be able to:
    - Save and manage favorite recipes.
    - View recently generated or viewed recipes.
    - Receive AI-based recipe recommendations tailored to their profile.
- Option to reset forgotten passwords securely.
- Support for session management and user authentication tokens to maintain security.

3.4.2 Ingredient Selection

- Users should be able to view and select ingredients from a visually interactive interface (e.g., clickable ingredient images).
- The platform must support both manual text-based input and image-based ingredient selection.
- Selected ingredients should be visually indicated (e.g., with a tick mark overlay or highlight effect).
- Users should be able to deselect ingredients easily if needed.
- The system should allow multi-ingredient selection to support complex recipe searches.
- Ingredient categories (e.g., vegetables, proteins, spices) should be provided for easier browsing.
- Real-time update of selected ingredients should be displayed in a summary section.
- User selections should be temporarily stored for session-based recipe recommendations.

3.4.3 AI-Powered Recipe Recommendations

- The system should generate personalized recipe suggestions based on selected ingredients and user profile data (e.g., preferences, allergies).
- AI algorithms such as content-based filtering or hybrid recommendation models should be used to enhance suggestion accuracy.
- Recommendations must consider dietary restrictions and preferred cuisines.
- The platform should allow real-time updates to recipe suggestions as users modify ingredient selections.
- Recipes should be ranked based on relevance, popularity, or user history.

- The system should support natural language processing (NLP) for interpreting ingredient combinations and user input.
- A feedback mechanism should be integrated to refine future recommendations based on user interactions.
- Each recommended recipe should include preparation time, nutritional info, ingredients list, and step-by-step instructions.

3.4.4 Recipe Detail View

- The system should display comprehensive information for each selected recipe.
- Recipe details must include the recipe name, high-quality image, preparation and cooking time.
- A clear list of ingredients with exact quantities should be provided.
- Step-by-step cooking instructions should be shown in an easy-to-follow format.
- Nutritional information (e.g., calories, proteins, fats, carbohydrates) should be available for health-conscious users.
- Users should be able to save or bookmark recipes for future reference.
- An interactive section for user ratings, reviews, and feedback should be included.
- Option to share the recipe via social media or download it as a PDF.
- Display allergy warnings based on user preferences.

3.4.5 Feedback and Ratings

- Users should be able to provide feedback on recipes they have tried.
- A star-based rating system (e.g., 1 to 5 stars) should be available for each recipe.
- Users can write detailed reviews or comments to share their cooking experience.
- The system should display average ratings and highlight top-rated recipes.
- Feedback should be filterable (e.g., most recent, highest rated, most helpful).
- Only registered users should be allowed to submit feedback or ratings.

- Admins or moderators should have the ability to monitor and manage inappropriate content.
- User-generated feedback can be used to improve future recipe recommendations.

3.4.6 Search Functionality

- Users should be able to search for recipes using keywords such as recipe name, ingredient, or cuisine.
- The search bar should support autocomplete suggestions based on popular or recent searches.
- Filters should be available to narrow results by dietary preferences, cooking time, difficulty level, etc.
- The search should be optimized to handle typos and return the most relevant results.
- Results should be displayed in a user-friendly layout with thumbnails, titles, and brief descriptions.
- Ingredient-based search should suggest recipes based on selected or entered ingredients.
- Advanced search options should be available for experienced users to define detailed criteria.

3.4.7 View Recently Accessed Recipes

- Users should be able to view a list of recipes they have recently accessed.
- The system should automatically track and store recently viewed recipes for logged-in users.
- Recently accessed recipes should be displayed in reverse chronological order (most recent first).
- Each recipe entry should include a thumbnail image, title, and brief summary.
- Users should be able to click on any recent recipe to quickly revisit the detailed view.
- The system should allow users to clear or manage their recently viewed list.
- This feature should enhance personalization and improve the overall user experience.
- The recently accessed data should persist across sessions for logged-in users.

3.4.8 Save Favourite Recipes

- Users should have the ability to mark recipes as favourites for easy access later.
- A "Save to Favourites" button should be available on each recipe detail page.
- Favourite recipes should be stored under the user's profile in the database.
- Users should be able to view all their saved recipes in a dedicated "Favourites" section.
- The system should allow users to remove recipes from their favourites list.
- Favourite recipes should display key details like image, title, and preparation time.
- This feature should enhance user engagement and promote recipe discovery.
- Favourite recipes should persist across sessions and be retrievable upon login.

3.4.9Admin Controls

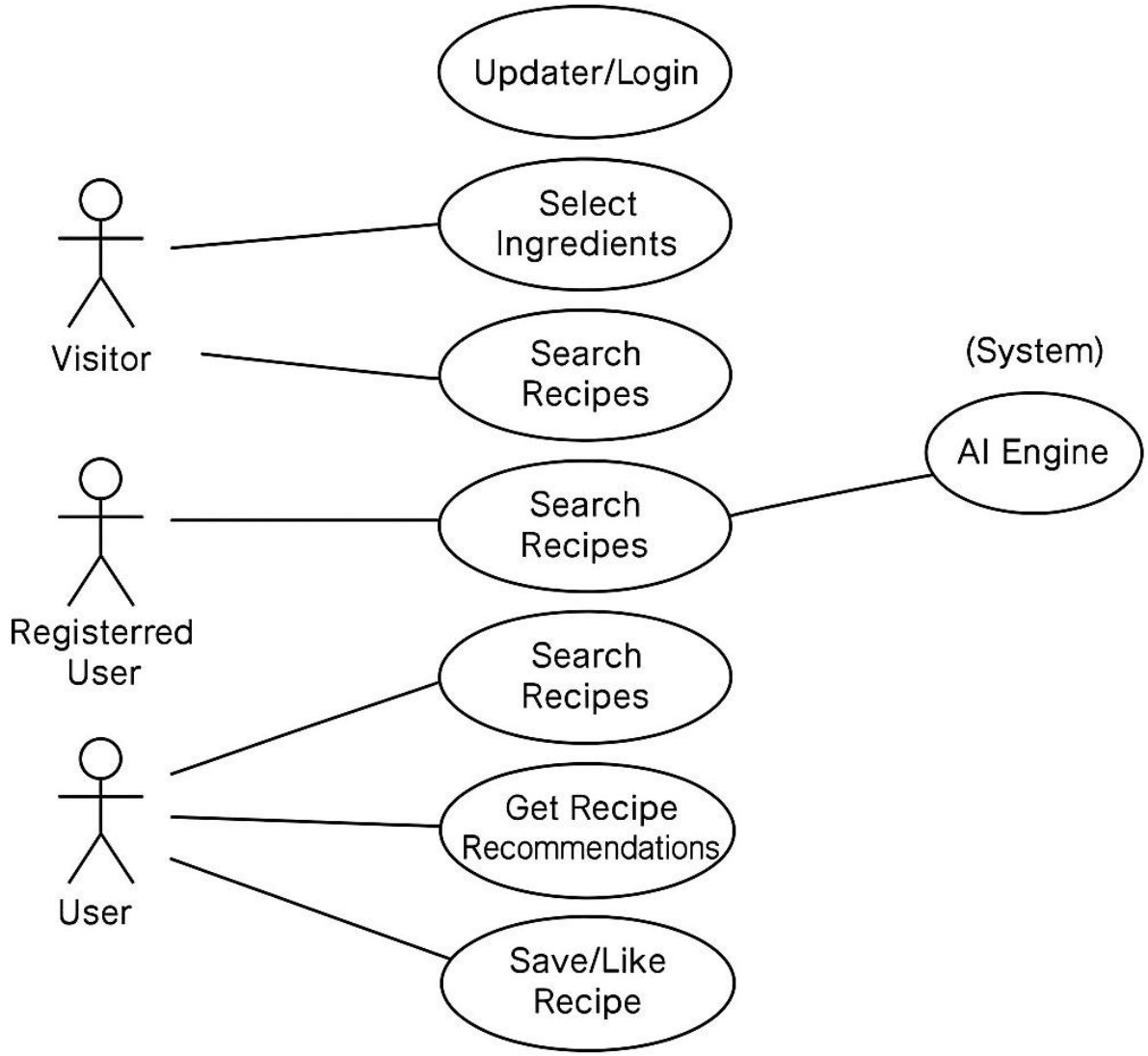
- Admins should have secure login access to a dedicated admin dashboard.
- The dashboard should allow admins to manage user accounts (view, edit, deactivate).
- Admins should be able to add, update, or delete recipe entries manually.
- Admins should monitor and moderate user-submitted feedback and ratings.
- Admins should have control over ingredient data, including uploading or modifying ingredient lists.
- The system should enable admins to view system usage analytics and performance logs.
- Admins should be able to manage content such as featured recipes, cuisine categories, and homepage banners.
- Access to admin functionalities should be restricted and protected with role-based authentication.

CHAPTER-4

DESIGN

4.1 USECASE UML DIAGRAM

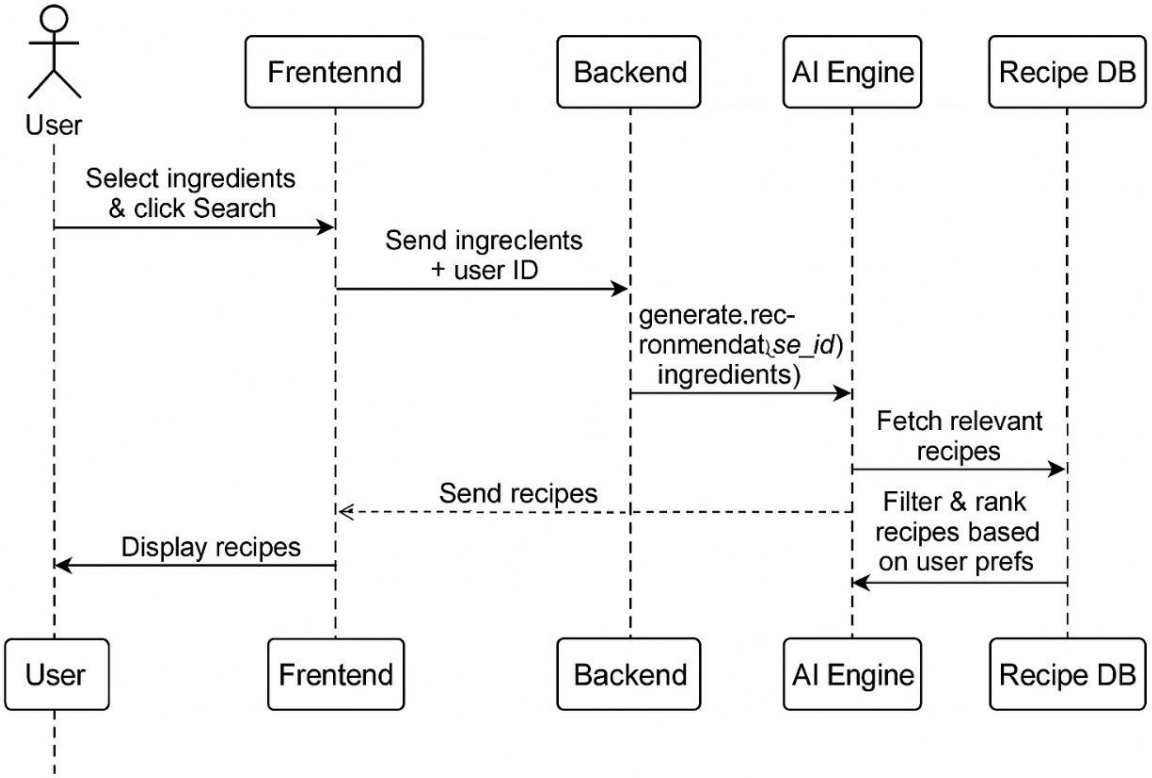
A Use Case Diagram is a simple UML (Unified Modeling Language) diagram that shows the interactions between users (called actors) and the system's functionalities (called use cases).



4.1 Use Case UML Diagram

4.2 SEQUENCE UML DIAGRAM

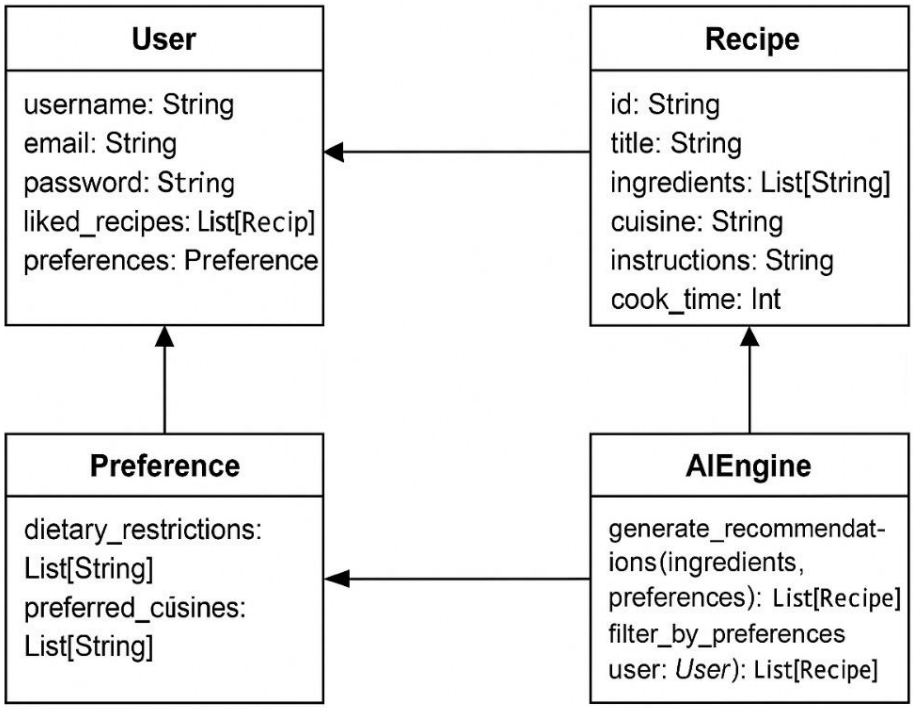
A Sequence Diagram is a type of UML (Unified Modeling Language) diagram that shows how objects interact in a particular scenario over time. It represents the sequence of messages exchanged between actors and system components to carry out a specific function or process.



4.2 Sequence UML Diagram

4.3 CLASS UML DIAGRAM

A Class UML Diagram is a type of static structure diagram in the Unified Modeling Language (UML). It is used to represent the classes, attributes, methods, and the relationships among objects in a software system.



4.3 Class UML Diagram

CHAPTER-5

IMPLEMENTATION

5.1 EXPERIMENTAL DESIGN

The experimental design for the AI-Automated Recipe Generator is structured to evaluate the system's performance, accuracy, and user satisfaction through controlled testing and real-world simulation. The design includes preparing datasets, defining test cases, and selecting evaluation metrics aligned with the project objectives.

A representative dataset of over 2.2 million recipes, including ingredients, cuisine types, user reviews, and nutritional data, is utilized to train and test the AI model. Controlled experiments are conducted using different subsets of this data to evaluate how well the model performs under varying input conditions (e.g., based on limited ingredients or user preferences).

User testing is carried out by allowing a group of participants to interact with the system, perform ingredient selections, and receive personalized recipe recommendations. Feedback is collected regarding the relevance, clarity, and usability of the suggested recipes.

Key evaluation metrics include recommendation accuracy, response time, user satisfaction scores, and system reliability under load. This experimental framework ensures a robust assessment of the system's functionality, scalability, and user-centered effectiveness.

5.2 FLASK APPLICATION CODE (app.py)

```
from flask import Flask
from flask_cors import CORS
from dotenv import load_dotenv
import os
from mongo_connection import db
from routes import recipe_routes, auth_routes
from login import login_blueprint  # Import login Blueprint
```

```
# Initialize Flask App
app = Flask(__name__)
CORS(app)

# Load environment variables
load_dotenv()
OPENAI_API_KEY = os.getenv("OPENAI_API_KEY")

#  Set secret key for sessions
app.secret_key = os.getenv("SECRET_KEY", "supersecretdevkey")

app.config["SECRET_KEY"] = os.getenv("SECRET_KEY")  # Add Secret Key for JWT

# Register Blueprints (Organizing Routes)
app.register_blueprint(recipe_routes)
app.register_blueprint(auth_routes)
app.register_blueprint(login_blueprint, url_prefix="/auth")  # Register Login Routes

# Run Flask Server
if __name__ == "__main__":
    app.run(debug=True)
```

5.3 INTERFACES, CLASSES AND METHODS

Interfaces:

- User Interface (UI): Designed using HTML, CSS, and JavaScript, the UI allows users to register/login, select ingredients, view recipe recommendations, and access recipe details.

- API Interface: RESTful APIs handle communication between the frontend and backend, enabling data exchange for ingredient input, user preferences, and recipe results.

Key Backend Classes:

- User: Manages user information, including registration, login credentials, preferences, allergies, and saved recipes.
- Recipe: Handles recipe data such as title, ingredients, instructions, cuisine, ratings, and nutrition.
- RecommendationEngine: Core class responsible for filtering and recommending recipes based on selected ingredients and user profile data.
- DatabaseHandler: Manages connections to MongoDB for CRUD operations on user profiles, ingredients, and recipe data.

Important Methods:

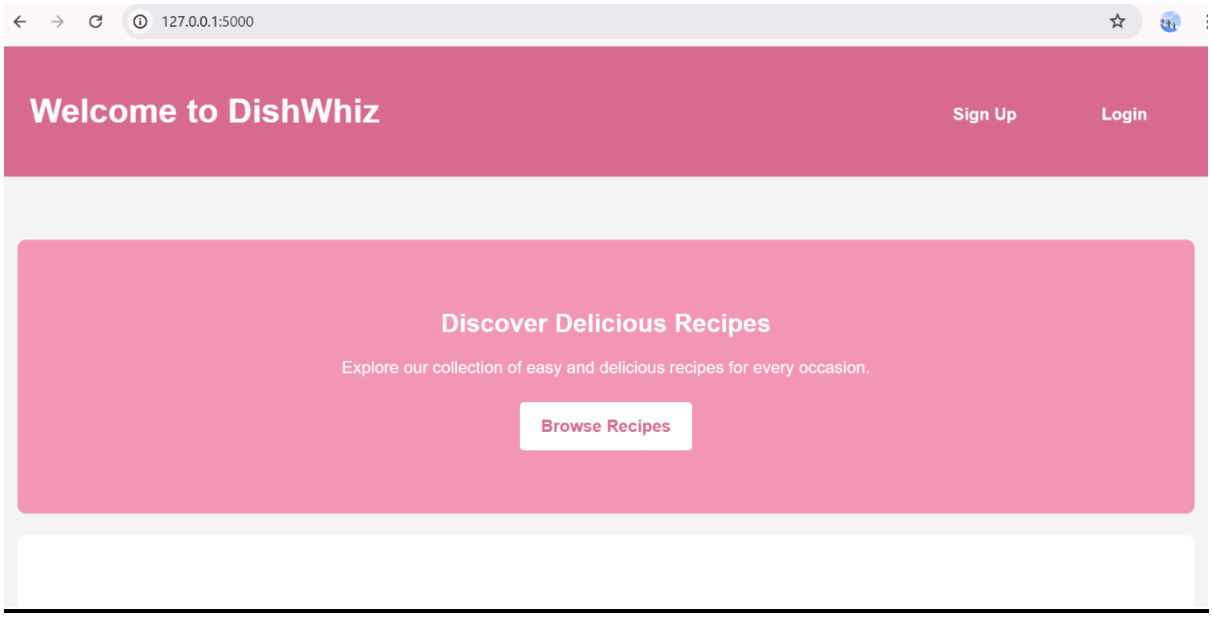
- register_user(data): Registers a new user in the database.
- login_user(credentials): Authenticates user login and returns session/token data.
- get_recipes_by_ingredients(ingredients): Returns a list of recipes matching the provided ingredients.
- recommend_recipes(user_id): Generates personalized recipe recommendations based on profile and preferences.
- save_favorite(recipe_id, user_id): Saves a recipe to the user's favorite list.
- get_recently_viewed(user_id): Fetches the list of recipes the user has recently interacted with.
- submit_feedback(user_id, recipe_id, rating, comment): Stores user feedback on a given recipe.

CHAPTER-6

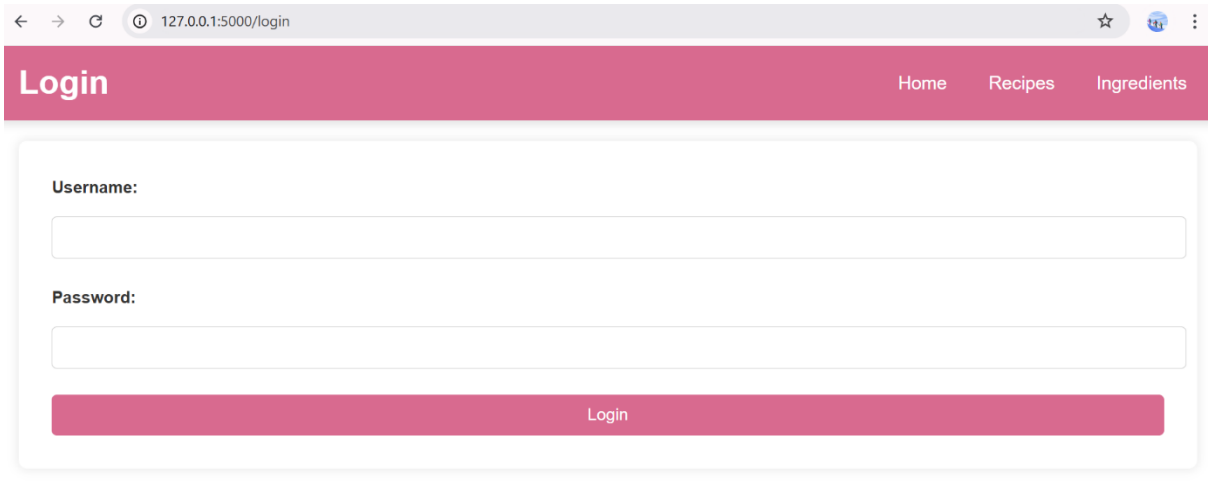
TESTING

6.1 Results

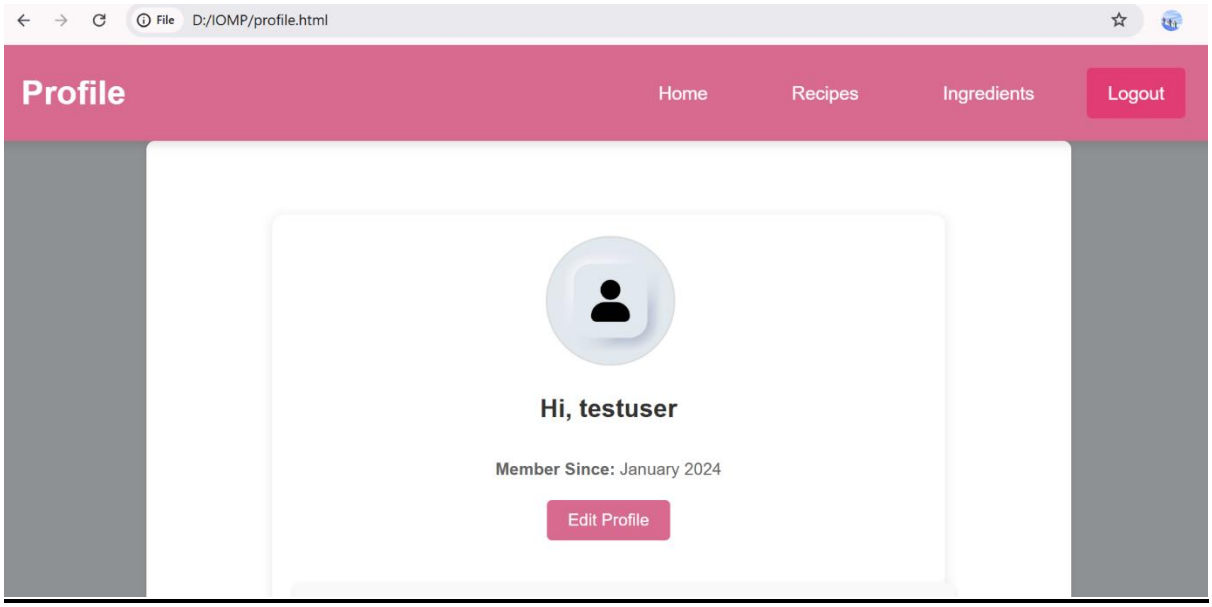
6.1.1 Home page



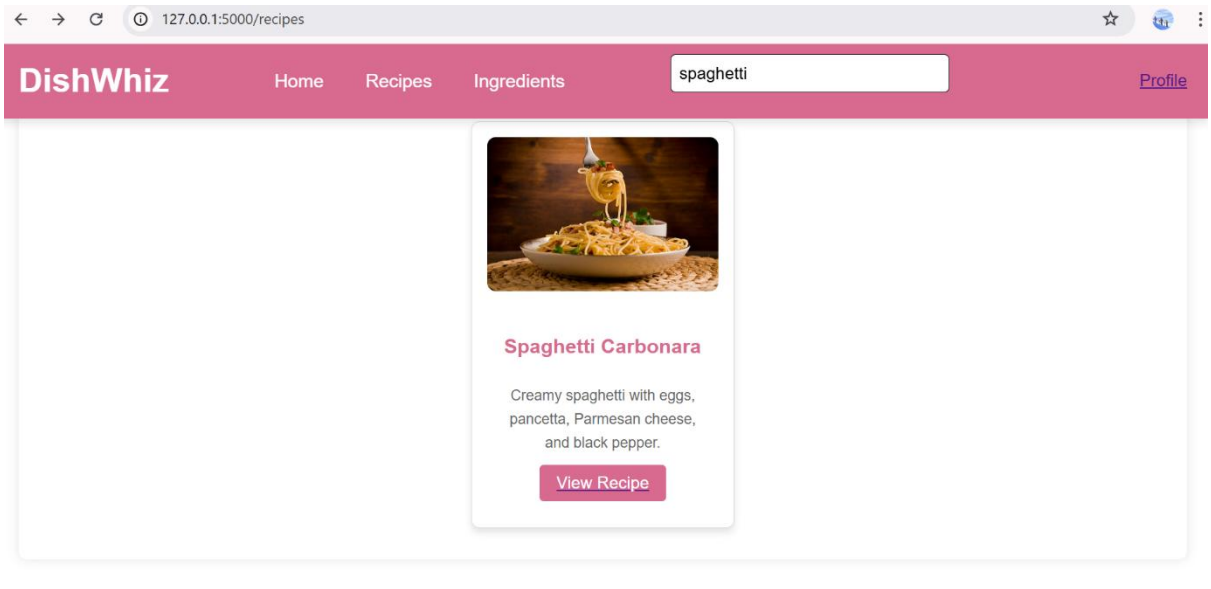
6.1.2 Login Page



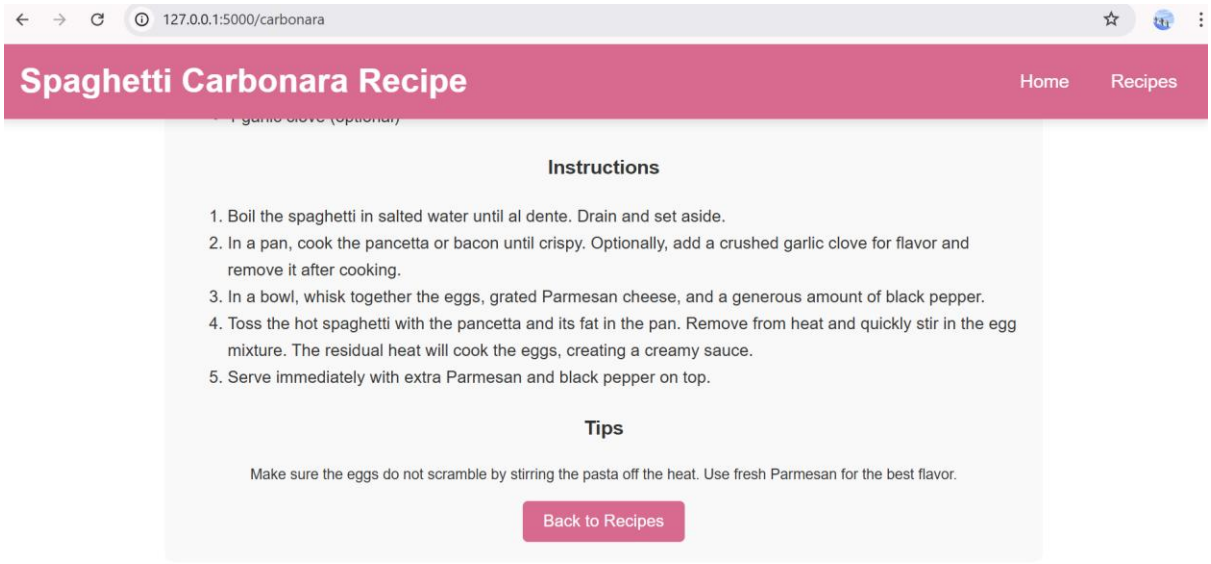
6.1.3 Profile Page



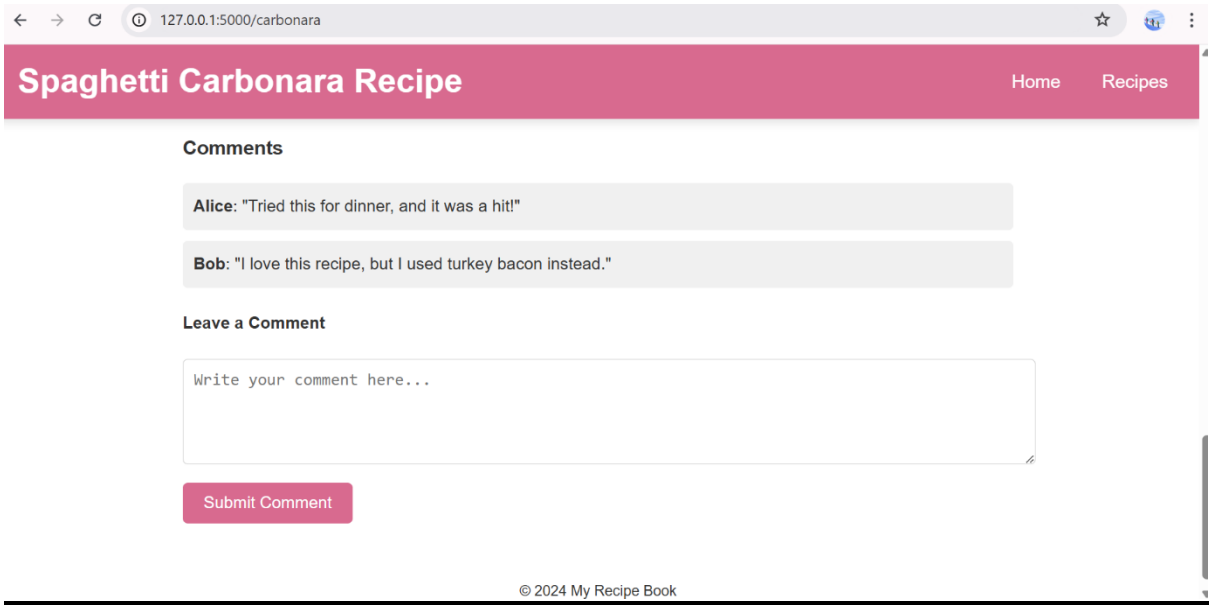
6.1.4 Search Page



6.1.5 Instructions Page



6.1.6 Feedback Page



CONCLUSION

The AI-Automated Recipe Generator project presents a comprehensive approach to modernizing the way users interact with food-related digital platforms. By blending artificial intelligence with intuitive user interfaces and real-time ingredient selection, the system not only improves meal planning efficiency but also introduces a highly personalized culinary experience. The platform addresses the growing demand for customized recipe recommendations by integrating user preferences, allergies, and dietary needs, ensuring that suggestions are both relevant and practical.

Throughout the project, a range of functionalities were implemented to enhance user engagement—such as ingredient-based searching, AI-powered recommendations, recipe detail views, saving favorites, and tracking recently viewed recipes. Each of these features plays a vital role in creating a seamless user journey, making the system highly interactive and user-centric. Additionally, robust backend support ensures secure authentication, efficient data handling, and real-time processing of user inputs.

The non-functional requirements like scalability, reliability, usability, maintainability, and performance were also carefully considered to ensure that the application remains responsive and efficient even with increasing users or data load. The experimental design validated the architecture's effectiveness, and the implementation of clearly structured classes and interfaces guarantees future maintainability and extension.

In conclusion, the AI-Automated Recipe Generator serves as a smart assistant for individuals seeking quick, healthy, and personalized meal suggestions. Its modular design allows for future upgrades, including multilingual support, nutrition tracking, social sharing of recipes, voice-enabled interactions, and integration with smart kitchen appliances. With rising interest in health, technology, and convenience, this platform positions itself as a forward-thinking solution in the growing ecosystem of AI-driven food applications.

FUTURE ENHANCEMENTS

While the AI-Automated Recipe Generator already offers a personalized and interactive experience, several future enhancements can further improve its functionality, user engagement, and scalability.

- **Voice Assistant Integration**: Integrate voice-enabled features using technologies like Google Assistant or Amazon Alexa to allow users to search and navigate recipes hands-free, especially useful while cooking.
- **Nutritional Analysis and Tracking**: Implement a feature that calculates the nutritional value of each recipe, including calories, macronutrients, and vitamins, to assist users with dietary goals such as weight loss, muscle gain, or managing health conditions.
- **Multilingual Support**: Add support for multiple languages to make the platform accessible to a broader global audience, catering to cultural diversity and regional cuisines.
- **Smart Kitchen Integration**: Integrate with smart kitchen devices (like smart fridges or ovens) to automate ingredient tracking, inventory updates, and even cooking instructions directly to appliances.
- **Social Sharing and Community Features**: Allow users to share their own recipes, review others', and create a food-loving community within the platform. This could include forums, following favorite users, and saving community-rated recipes.
- **Meal Planning and Shopping List Generation**: Introduce weekly or monthly meal planning based on user preferences and generate auto-updated shopping lists that sync with local grocery APIs for delivery or pickup.
- **Allergy Detection with Image Recognition**: Use AI and computer vision to scan product labels or uploaded food images to detect allergens, providing a safer experience for users with food sensitivities.
- **Gamification and Rewards System**: Add features like cooking challenges, badges, and reward points to keep users engaged and motivated to try new recipes.
- **Offline Access**: Enable recipe downloads and offline viewing to support users in areas with poor internet connectivity.

REFERENCES

1. IBM Developer. Build a Recipe Recommendation App Using Python and Flask

   https://developer.ibm.com/articles/build-a-recipe-recommendation-app-using-python-and-flask/

2. Medium. Creating a Food Recommendation System Using Machine Learning

   https://medium.com/analytics-vidhya/creating-a-food-recommendation-system-using-machine-learning-cb3d7a3e5a8a

3. Kaggle. Recipes and Ingredients Dataset

   https://www.kaggle.com/kaggle/recipe-ingredients-dataset

4. Medium. A Beginner's Guide to Building a Recommendation System in Python

   https://medium.com/swlh/a-beginners-guide-to-building-a-recommendation-system-in-python-539c1f8c66c5

5. DataCamp. Introduction to Content-Based Filtering in Python

   https://www.datacamp.com/tutorial/content-based-recommendation-system-python

6. Real Python. How to Use Scikit-Learn for Machine Learning

   https://realpython.com/python-machine-learning-scikit-learn/

7. Towards Data Science. Building a Recipe Generator with GPT-3

   https://towardsdatascience.com/building-a-recipe-generator-with-gpt-3-and-python-d10ef186385d