**A Project Report**

**on**

***“FOOD RECOMMENDATION SYSTEM”***

Submitted in partial fulfillment of the requirements

For the award of the degree of Bachelor of Technology in Computer Science & Engineering 

AKS UNIVERSITY, SATNA

**B. Tech (CSE) 5th Semester**

#### Submitted by

##### AYUSH CHAURASIYA (B2255R10106007)

##### HIMANSHU GAUTAM (B2255R10106110)

**Under The Guidance of Approved by**

Mr. Vinay Kumar Dwivedi Prof. (Dr.) Akhilesh A. Waoo

(Assistant Professor) (Associate Dean & Head CS/IT)

**Department of Computer Science & Engineering**

**AKS University, Satna (M.P.)**

## CERTIFICATE

This certify that the project report entitled “**FOOD RECOMMENDATION SYSTEM**” submitted by partial fulfilment of the requirement for the degree of Bachelor of Engineering in Technology in **2024-2025** AKS University, Satna is a bonafide project work carried out by **AYUSH CHAURASIYA (B2255R10106007),** under my supervision. The subject of the project report has been approved by supervisor. This is also to certify that it is his/her original work and no part of this project is report has been submitted for any other degree/diploma.

All the assistance the and help received during the course of the investigation has been duly acknowledged.

1. I am satisfied that the report presented by **AYUSH CHAURASIYA (B2255R10106007)** is worthy of consideration for award of the degree.
2. I certify:
   1. That he/she pursued the prescribed course for project.
   2. That he/she bears good moral character.

**Place: AKSU, Satna ………………………………**

**Date:…/…./……… External’s Signature**

……………………………. ………………………………

**Supervisor Head of Department**

**(Mr.** Vinay Kumar Dwivedi**)** **Prof. (Dr.) Akhilesh A. Waoo**

(Assistant Professor) (Asso. Dean, Professor, CS/IT)

## CERTIFICATE

This certify that the project report entitled “**FOOD RECOMMENDATION SYSTEM**” submitted by partial fulfilment of the requirement for the degree of Bachelor of Engineering in Technology in **2024-2025** AKS University, Satna is a bonafide project work carried out by **HIMANSHU GAUTAM (B2255R10106110),** under my supervision. The subject of the project report has been approved by supervisor. This is also to certify that it is his/her original work and no part of this project is report has been submitted for any other degree/diploma.

All the assistance the and help received during the course of the investigation has been duly acknowledged.

1. I am satisfied that the report presented by **HIMANSHU GAUTAM (B2255R10106110)** is worthy of consideration for award of the degree.
2. I certify:
   1. That he/she pursued the prescribed course for project.
   2. That he/she bears good moral character.

**Place: AKSU, Satna ………………………………**

**Date:…/…./……… External’s Signature**

……………………………. ………………………………

**Supervisor Head of Department**

**(Mr.** Vinay Kumar Dwivedi**)** **Prof. (Dr.) Akhilesh A. Waoo**

(Assistant Professor) (Asso. Dean, Professor, CS/IT)

**CERTIFICATE BY THE CANDIDATE**

I certify that the project report entitled “**FOOD RECOMMENDATION SYSTEM**” is my own work conducted under the supervision of **Mr.** Vinay Kumar Dwivedi (Supervisor), Department of Computer Science, AKS University, Satna (M.P.) for partial fulfilment of the requirement for the degree in Bachelor of Engineering in Technology July-Dec 2024-25**.**

I further certify that to the best of my knowledge and belief the project report does not contain any part of this work which has been submitted for the award of any degree either in this university or in any other University/ Deemed University/ Institutes.

* AYUSH CHAURASIYA (B2255R10106007) - B.Tech (CSE) 5th Semester

**Place:** AKSU, Satna

**Date:**…./…../………

…………………………..

Signature of Candidate

Ayush Chaurasiya

**CERTIFICATE BY THE CANDIDATE**

I certify that the project report entitled “**FOOD RECOMMENDATION SYSTEM**” is my own work conducted under the supervision of **Mr.** Vinay Kumar Dwivedi (Supervisor), Department of Computer Science, AKS University, Satna (M.P.) for partial fulfilment of the requirement for the degree in Bachelor of Engineering in Technology July-Dec 2024-25**.**

I further certify that to the best of my knowledge and belief the project report does not contain any part of this work which has been submitted for the award of any degree either in this university or in any other University/ Deemed University/ Institutes.

* HIMANSHU GAUTAM (B2255R10106110) - B.Tech (CSE) 5th Semester

**Place:** AKSU, Satna

**Date:**…./…../………

…………………………..

Signature of Candidate

Himanshu Gautam

## SELF DECLARATION

I hereby declare that the work presented in this project entitled **“FOOD RECOMMENDATION SYSTEM "** towards the partial fulfilment of the requirement for the award of **Degree in B. Tech** in Department of Computer Science, **AKS University, Satna (M.P.)** is an authentic record of my own work.

I have not submitted the matter embodied in the project for the award of any other degree or diploma to any other institute or university.

…………………………..

Signature of Candidate

Ayush Chaurasiya

## 

## SELF DECLARATION

I hereby declare that the work presented in this project entitled **“FOOD RECOMMENDATION SYSTEM "** towards the partial fulfilment of the requirement for the award of **Degree in B. Tech** in Department of Computer Science, **AKS University, Satna (M.P.)** is an authentic record of my own work.

I have not submitted the matter embodied in the project for the award of any other degree or diploma to any other institute or university.

…………………………..

Signature of Candidate

Himanshu Gautam

## ACKNOWLEDGEMENT

It is a great for me in taking this opportunity to express my sincere thanks and ineptness to **Prof. (Dr.) Akhilesh A. Waoo**, Head of the Department of CSE, AKS University, Satna (M.P.)

I consider myself lucky enough to have such a great project. This project would add as an asset to my profile.

At this moment of accomplishment, first of all I pay homage to my guide, **Mr.** Vinay Kumar Dwivedi from AKS University Satna (M.P.). This work would not have been possible without his guidance, support and encouragement. Under his guidance I successfully overcame many difficulties and learned a lot.

I am deeply and forever indebted to my parents for their love, support and encouragement throughout my entire life.

.

**ABSTRACT**

The Indian Food Recommendation System is a web-based application designed to assist users in identifying Indian dishes they can prepare using the ingredients they already have. The system bridges the gap between ingredient availability and culinary possibilities, offering a practical solution for everyday cooking challenges.

Users input a list of ingredients through a simple and intuitive interface. The frontend (developed using HTML, CSS, and JavaScript) communicates with the backend (powered by PHP) to process these inputs. The backend queries a SQL-based recipe database to identify matching dishes and their corresponding recipes. The results are then displayed to the user, including detailed instructions for preparing the dishes.

This project provides a personalized experience, encouraging efficient ingredient usage and reducing food wastage. By leveraging a lightweight tech stack and a scalable database, the system ensures quick, accurate responses. Its application is particularly beneficial for individuals exploring Indian cuisine or seeking cooking inspiration with minimal resources.

## TABLE OF CONTENT

* Introduction……...………………………………………..…**10**
* Objective of The Project…………….……………………... **12**
* Design and development process…..………...…………..… **13**
* Benefit and application..………………………………...…. **17**
* Scope of the Project……………………………………………………**25**
* Definition of Problem………………………………………**29**
* Solution Outline…………………………….………………**34**
* Main Module………………………….…………………… **35**
* Algorithm …………………………………………………. **38**
* Error handling…..….…………………………….……...….**41**
* Technical overview………………………………….……...**43**
* Non-functional requirement ……... ………………….…….**47**
* Code………………………….....……….….…….……...... **51**
* Code Explanation…………….....……….….…….……...... **56**
* Logic……………………………………………………………………….**61**
* Output Screen …..…………………………………………. **64**
* DFD diagram …..……………………………………...……**65**
* Project Explanation…………………………………………………….**66**
* Software Testing …………………………………...……….**68**
* Conclusion……………………………………………….… **71**

**Introduction to Food Recommendation system**

The Indian Food Recommendation System is a user-centric web application aimed at simplifying the cooking experience by offering personalized recipe suggestions based on the ingredients available to the user. Indian cuisine, known for its diversity and richness, offers countless dishes that can be prepared with various combinations of ingredients. This system bridges the gap between what users have in their kitchen and the culinary possibilities they can explore.

Users interact with the system through a straightforward interface where they input the ingredients they possess. The application processes this input and queries a structured database of Indian recipes using backend technologies. The system then identifies dishes that match the provided ingredients and presents the user with a list of recommendations along with step-by-step preparation instructions.

The project leverages a simple tech stack, including HTML, CSS, JavaScript, PHP, and SQL, ensuring easy deployment and functionality. It also encourages optimal utilization of ingredients, reducing food wastage and making meal planning more efficient. This application is particularly useful for individuals looking for cooking inspiration, beginners experimenting with Indian dishes, or anyone seeking to make the most of limited ingredients.

The Indian Food Recommendation System combines functionality with cultural appreciation, making Indian cuisine accessible and enjoyable for all.

**The Need for a User-Friendly Food Recommendation System**

Cooking often begins with the question, “What can I make with what I have?” Many individuals struggle to decide on a dish when faced with a limited set of ingredients in their kitchen. The Indian Food Recommendation System addresses this challenge by providing a practical, efficient solution that bridges the gap between ingredient availability and recipe selection.

Indian cuisine, celebrated for its variety and richness, often requires specific combinations of ingredients. However, most people are unaware of the multitude of dishes that can be prepared with commonly available ingredients. This system caters to such users, helping them explore and prepare Indian dishes effortlessly, even with minimal resources. It reduces dependency on extensive recipe books or online searches, saving time and effort.

The project also promotes sustainable cooking practices by encouraging users to utilize ingredients they already have, thereby reducing food wastage. This is especially valuable in today’s fast-paced world, where resource optimization is crucial. Additionally, for beginners or individuals unfamiliar with Indian cuisine, the system serves as a guide to discover and prepare authentic dishes.

In essence, this project combines technology and culinary creativity to make cooking an accessible, enjoyable, and waste-free experience for users.

**Objective of the Food Recommendation Project**

The primary objective of the Indian Food Recommendation System is to provide an intuitive and efficient platform that enables users to discover Indian dishes they can prepare with the ingredients available at hand. This system aims to simplify the decision-making process in cooking by bridging the gap between ingredient availability and recipe selection, making cooking an accessible and enjoyable experience for users.

One of the key objectives is to promote the optimal utilization of ingredients, thereby reducing food wastage. By encouraging users to make use of what they already have, the system fosters sustainable cooking practices. Additionally, it provides users with easy access to a diverse range of Indian recipes, encouraging exploration of Indian cuisine, even for beginners or those with limited cooking knowledge.

The system also aims to enhance user convenience by eliminating the need for extensive online searches or reliance on recipe books. Through a straightforward and user-friendly interface, it delivers quick and accurate results based on user inputs, saving time and effort.

Furthermore, the project strives to demonstrate the potential of a lightweight and scalable tech stack (HTML, CSS, JavaScript, PHP, and SQL) in creating practical and impactful web-based solutions, making it a valuable tool for home cooks and food enthusiasts alike.

**Design and Development Process**

The Indian Food Recommendation System is developed as a user-friendly web application that integrates frontend, backend, and database components to deliver personalized recipe suggestions based on user-provided ingredients. Below is a detailed account of the design and development process, broken into logical stages.

1. Requirement Analysis

The first step was identifying the key requirements of the project:

User Interaction:

A simple interface where users can input a list of ingredients.

Display of relevant Indian recipes based on these ingredients.

Backend Logic:

Processing user input and querying the database to find matching recipes.

Database Design:

A database storing recipes, ingredients, and preparation instructions.

Technology Stack:

Frontend: HTML, CSS, and JavaScript for the user interface.

Backend: PHP to handle user requests and communicate with the database.

Database: MySQL for storing and retrieving recipe data.

Deployment:

Hosting the backend on a server that supports PHP and SQL.

Hosting the frontend on GitHub Pages for easy accessibility.

2. System Design

2.1 Architectural Overview

The system architecture is based on a client-server model:

Frontend:

Accepts user inputs (ingredients) and displays results.

Communicates with the backend via AJAX requests.

Backend:

Processes user inputs, interacts with the database, and returns recipes.

Database:

Contains structured tables storing recipe details, ingredients, and preparation steps.

2.2 Frontend Design

The frontend design prioritizes simplicity and usability:

Homepage:

A text box where users can input ingredients.

A submit button to send the inputs to the backend.

Result Page:

Displays matching recipes in a clean, readable format.

Provides a detailed recipe with steps for preparation.

Tools Used:

HTML for structure.

CSS for styling.

JavaScript for interactivity.

2.3 Backend Design

The backend is implemented using PHP:

Input Validation:

Validates user inputs to prevent SQL injection and ensure meaningful queries.

Query Processing:

Extracts matching recipes from the database based on the ingredients provided.

Response Formatting:

Sends recipe results to the frontend in JSON format for easy parsing.

2.4 Database Design

The database is designed to handle recipe storage efficiently:

Table: Recipes

Stores recipe ID, name, and preparation instructions.

Table: Ingredients

Stores ingredient ID, name, and associated recipe IDs.

Relationships:

A many-to-many relationship between recipes and ingredients to accommodate shared ingredients across recipes.

#### **3.1 Frontend Development**

* **HTML**:
  + A form for user input with a textbox for ingredients.
* **CSS**:
  + Styling for the form and result display to ensure a clean layout.
* **JavaScript**:
  + Handles user interactions and communicates with the backend .

**Recipe table-**

| Recipe id | Recipe name | Preparation steps |
| --- | --- | --- |
| 1 | Aloo paratha | Instructions….. |
| 2 | Paneer butter masala | Instructions….. |

**Ingredient table-**

| Ingredient id | Ingredient name |
| --- | --- |
| potato | paneer |

**Recipe table-**

| Recipe id | Ingredient id |
| --- | --- |
| 1 | 1 |
| 2 | 2 |

**Benefits and Applications**

**Benefits**

* Efficient Meal Planning:
* The Indian Food Recommendation System simplifies meal planning by suggesting recipes based on ingredients the user already has. This reduces the effort and time spent deciding what to cook.
* Minimizing Food Waste:
* By providing tailored recipes that utilize available ingredients, the system helps users make the most of their kitchen inventory, thereby reducing food wastage.
* Convenience:
* The system eliminates the need for users to search through extensive recipe books or online platforms. With just a few clicks, they can find relevant dishes.
* Exploration of Indian Cuisine:
* Users unfamiliar with Indian dishes can explore a wide variety of recipes, encouraging cultural appreciation and culinary experimentation.
* Beginner-Friendly:
* This platform is particularly helpful for beginner cooks, providing them with simple and clear recipes that match their ingredient availability.
* Cost-Effective:
* By utilizing only available ingredients, users can save money that might otherwise be spent on additional grocery shopping.

**Benefits for Developers**

1. Skill Enhancement and Technical Growth

Web Development:

The project involves the use of a variety of web development technologies including HTML, CSS, JavaScript, PHP, SQL, and MySQL. Developers can deepen their understanding of front-end and back-end development while working with databases. Building this system allows developers to get hands-on experience with full-stack development, making them more proficient in managing both client-side and server-side operations.

Database Management:

Working with SQL and designing the database schema for the recipe and ingredients system helps developers practice efficient database design, query optimization, and data manipulation. They gain valuable experience in relational database management and learn how to structure complex data for effective querying.

Algorithm Development:

One of the most important technical aspects is developing the algorithm that matches available ingredients with appropriate recipes. This requires logic and problem-solving skills to handle multiple variables (ingredients), ensuring that the search results are both accurate and relevant. This challenge improves a developer's algorithmic thinking and logical problem-solving skills.

2. Practical Experience in Building a Real-World Application

Project Management:

Developing a working food recommendation system requires careful planning, task prioritization, and time management. Developers experience the full cycle of project development, from requirements gathering to system deployment. This provides practical experience in project management and gives them a portfolio-worthy product to showcase.

User-Centric Design:

The system requires user-friendly interfaces and intuitive navigation to ensure accessibility, which is important for both experienced and beginner cooks. Developers must focus on user experience (UX) and user interface (UI) design, making this an opportunity to learn about creating aesthetically pleasing and functional web applications.

Real-World Impact:

Developers can see how their work solves a real problem faced by users: meal planning and food wastage. This gives them a sense of accomplishment as they create a practical solution that benefits people, and they get to see the tangible impact of their work on the everyday lives of users.

3. Exposure to Full-Stack Development

Front-End and Back-End Integration:

Developing the system requires integration of front-end (user interface) and back-end (server and database) components. Developers gain a solid understanding of how these layers work together, learning how to send user inputs from the front-end, process the data on the back-end, and retrieve relevant information from the database to display as results.

Handling User Input and Data Validation:

Developers will be responsible for input validation, ensuring that users provide valid ingredients and ensuring that results match those inputs correctly. This will help them learn best practices for handling user data and maintaining application integrity.

**Benefits for Other Users**

The Indian Food Recommendation System offers numerous advantages to a wide range of users who seek to make the most of their ingredients, explore new cuisines, and simplify their cooking experiences. Below are the primary benefits for other users of the system:

1. Time-Saving and Efficient Meal Planning

Benefit:

The system provides users with quick and personalized recipe recommendations based on the ingredients they already have, eliminating the need to search through various websites, cookbooks, or other resources to find recipes. This feature saves a significant amount of time for individuals who are busy, have a limited culinary background, or just want a more efficient and convenient way to plan meals.

Impact:

With this system, users can easily find meal ideas in seconds, saving time during meal preparation and helping them focus on enjoying their food. It's especially helpful for professionals, students, and families who want to put nutritious meals on the table without spending too much time cooking.

2. Reduces Food Waste

Benefit:

The system helps users efficiently utilize ingredients they already have at home by suggesting recipes based on what’s available. This reduces the amount of food waste by providing a practical way to transform ingredients into tasty meals.

Impact:

Users no longer have to worry about food going to waste because of unused ingredients, which leads to both financial savings and a more environmentally friendly approach to cooking. This feature is crucial in promoting sustainable food consumption practices.

3. Cost-Effective Cooking

Benefit:

By using the Indian Food Recommendation System, users can plan their meals more cost-effectively by only buying the ingredients that are actually needed for the recipes. The system provides recipes that maximize the utilization of available ingredients, so users don’t need to purchase unnecessary items. This translates into more efficient grocery shopping and significant savings on food costs.

Impact:

This feature is particularly valuable for households with limited budgets, students, young professionals, or anyone who wants to make more informed and economical decisions about food purchases.

4. Improved Cooking Skills

Benefit:

The Indian Food Recommendation System provides step-by-step recipes with clear instructions, even for those who are new to cooking or unfamiliar with Indian cuisine. The system helps users build confidence in the kitchen by guiding them through the cooking process, from selecting ingredients to preparing the dish.

Impact:

With access to clear, detailed recipes, more people can successfully cook authentic Indian food at home, making cooking more enjoyable and less intimidating. Users who previously felt discouraged by the complexity of Indian dishes will find it much easier to replicate recipes and get delicious results.

**Applications**

* Household Cooking:
* Ideal for families and individuals looking for quick meal ideas without extensive preparation or shopping.
* Educational Tool:
* The platform can be used in culinary training to help students learn how to create dishes based on limited ingredients.
* Restaurants and Catering:
* Useful for small-scale chefs or catering services to quickly decide on dishes based on inventory.
* Travel and Camping:
* Helps travelers or campers plan meals with limited resources available in unfamiliar locations.
* Community and Food Banks:
* This system can assist in creating meal ideas for large groups using donated or surplus food items.

**Scope of the Project**

The Indian Food Recommendation System is a versatile web application that caters to a wide range of users, from individuals seeking quick meal ideas to organizations looking to optimize ingredient usage. The primary scope of this project lies in providing an intuitive and efficient platform that matches user-provided ingredients with a database of Indian recipes, delivering relevant and personalized suggestions.

The system is particularly beneficial for households, helping individuals and families make the most of available ingredients while reducing food wastage. For culinary beginners, it serves as a practical guide, introducing them to authentic Indian dishes with step-by-step preparation instructions.

Beyond individual use, the system has broader applications in industries like small-scale catering, restaurants, and food banks. It can assist chefs and managers in utilizing surplus inventory creatively. Moreover, the platform has educational value, serving as a learning tool for culinary students to experiment with ingredient combinations.

The project also has the potential for future expansion. Features like advanced search filters (e.g., dietary preferences or cooking time), multilingual support for regional Indian languages, and integration with smart kitchen devices could significantly enhance its utility. A mobile application version could extend its reach further, making it accessible to a wider audience.

In summary, the scope of the project is vast, spanning personal, educational, and professional use cases. Its potential for scalability and innovation ensures its relevance in modern cooking and sustainable food practices.

**Functional Scope**

The Indian Food Recommendation System offers a well-defined functional scope to deliver an efficient and user-friendly experience. Below are the key functionalities it encompasses:

1. Ingredient-Based Recipe Search

Input:

Users provide a list of available ingredients through a simple text input interface.

Processing:

The backend queries the database to identify recipes matching the provided ingredients.

Output:

Displays a list of Indian recipes along with preparation instructions.

2. Recipe Details

Functionality:

For each recipe, the system provides step-by-step cooking instructions, including ingredient quantities and preparation techniques.

Purpose:

Ensures that users can prepare the dishes confidently, regardless of their cooking experience.

3. User-Friendly Interface

Features:

A simple input field for ingredients.

Clean, readable results displayed in a structured format.

Purpose:

Enhances user experience and ensures accessibility for people with varying levels of technical proficiency.

4. Efficient Database Interaction

Backend Process:

Validates inputs to ensure meaningful searches.

Uses optimized SQL queries to handle ingredient-recipe mapping efficiently.

Scalability:

Can handle large recipe databases for future expansion.

5. Error Handling

Scenarios Handled:

No matching recipes found for the given ingredients.

Invalid or incomplete user inputs.

User Feedback:

Displays clear error messages and suggestions for re-trying with different inputs.

6. Multi-Ingredient Matching

Feature:

Matches recipes even if only some of the provided ingredients are available, ranking results by relevance.

7. Future Extendability

Potential Additions:

Filter by dietary preferences (e.g., vegetarian, vegan, gluten-free).

Add user login for saving favorite recipes.

Support for advanced search options like cooking time or difficulty level.

**Out of Scope**

While the Indian Food Recommendation System is designed to provide a streamlined and efficient experience for users, there are certain features and functionalities that are intentionally excluded from the current implementation to keep the project focused and simple. Below are the aspects that are out of scope:

1. Real-Time Ingredient Availability Tracking

The system does not track or update the inventory of ingredients in the user's kitchen automatically. Users must manually input the list of available ingredients.

2. Advanced Filters

Features such as filtering recipes by dietary preferences (e.g., vegan, gluten-free), cooking time, difficulty level, or calorie count are not included.

3. Dynamic Recipe Customization

The system does not allow users to customize or modify recipes (e.g., scaling quantities, substituting ingredients).

4. Multiple Cuisines

The focus is solely on Indian cuisine. Recipes from other cuisines are not part of the current implementation.

5. User Accounts and Personalization

The system does not include features like user registration, login, saving favorite recipes, or personalizing recommendations based on user history or preferences.

6. Voice or AI-Assisted Features

The application does not support voice inputs, AI chatbots, or advanced natural language processing for ingredient recognition.

7. Mobile Application

The current implementation is a web-based system. A dedicated mobile app for Android or iOS is out of scope.

8. Multi-Language Support

The application supports only English for input and output; regional Indian languages are not included.

**Definition of Problem: Food recommendation system**

In today's fast-paced world, people often struggle with meal planning, especially when they have a limited set of ingredients in their kitchen. This challenge becomes more prominent when individuals wish to explore diverse cuisines, such as Indian food, which often requires specific combinations of ingredients and spices. As a result, many people either end up wasting ingredients or making repetitive meals due to a lack of inspiration or time to search for new recipes.

The problem can be defined as follows:

1. **Limited Ingredient Usage**:  
   People often have a variety of ingredients at home but are unsure how to use them to create diverse, delicious meals. They may not have the knowledge of what dishes can be made with the ingredients they currently possess.
2. **Time and Effort in Recipe Searching**:  
   Searching for recipes online or in cookbooks can be time-consuming and inefficient, especially when users don’t know what they can cook with the specific ingredients available to them. This results in frustration and sometimes leads to wasted food or repeated meals.
3. **Difficulty in Exploring New Cuisines**:  
   For individuals unfamiliar with Indian cuisine, discovering and cooking traditional Indian dishes can be intimidating. The complexity of Indian recipes, which often require a combination of unique spices and ingredients, can discourage beginners from experimenting.
4. **Food Wastage**:  
   Due to improper planning or lack of recipe knowledge, unused ingredients often go to waste, which not only impacts household budgets but also contributes to environmental issues related to food waste.

**Problem Statement**

In today's fast-paced lifestyle, individuals often face challenges when it comes to meal planning, especially when they have limited ingredients available in their kitchen. This problem becomes more complex when it comes to cooking Indian cuisine, which requires a specific combination of spices and ingredients. Many people struggle to identify which dishes they can prepare with the ingredients they currently have, leading to wasted food, repetitive meals, or unnecessary grocery shopping.

The problem is compounded by:

Inefficient Recipe Searching:

Searching for recipes online or in cookbooks is time-consuming and often unhelpful when users are unsure of what dishes can be made from the ingredients available to them.

Lack of Culinary Knowledge:

People, especially beginners, may lack knowledge about which ingredients are essential for cooking Indian dishes and how they can be combined to create diverse meals.

Food Waste:

Unused ingredients often go to waste, which not only affects household budgets but also contributes to environmental issues associated with food wastage.

Difficulty in Exploring New Cuisines:

For individuals unfamiliar with Indian cuisine, finding recipes and learning how to use specific spices and ingredients can be overwhelming.

This Indian Food Recommendation System aims to solve these problems by providing users with personalized recipe suggestions based on the ingredients they already have, reducing food waste, making meal planning more efficient, and helping users explore and enjoy Indian cuisine with ease.

**Problem Analysis**

The **Indian Food Recommendation System** addresses several challenges related to cooking, ingredient management, and the exploration of Indian cuisine. A detailed analysis of the problem highlights the pain points that users commonly experience and the reasons behind these issues.

#### **1. Limited Ingredient Usage**

* **Current Situation**:  
  Many individuals have an assortment of ingredients in their kitchens but lack the knowledge of which dishes they can create with those ingredients. For example, someone may have potatoes, onions, and tomatoes but may not know that they can prepare a variety of Indian dishes such as "Aloo Gobi" or "Aloo Tamatar ki Sabzi" using these basic ingredients.
* **Impact**:  
  Users end up purchasing more ingredients or resort to making repetitive meals that don’t utilize their existing stock efficiently. This not only wastes money but also contributes to food wastage.
* **Cause**:  
  There is no easy-to-access, quick tool that shows what can be cooked with the ingredients a person already has. People rely on their memory or try searching online, which is often time-consuming and not always helpful.

#### **2. Time-Consuming Recipe Searching**

* **Current Situation**:  
  Searching for recipes on the internet or flipping through cookbooks can be a tedious and unproductive process. Users often end up opening multiple websites or books without knowing whether they have the ingredients required for the recipes.
* **Impact**:  
  This leads to frustration, time wastage, and unnecessary grocery shopping. Many
* individuals, especially beginners or busy professionals, find it difficult to make meal decisions when they don't have time to browse through numerous resources.
* **Cause**:  
  Recipe search platforms do not offer personalized suggestions based on available ingredients, and users must often input a long list of ingredients in hopes of finding a matching recipe.

#### **3. Lack of Knowledge about Indian Cuisine**

* **Current Situation**:  
  For someone new to Indian cuisine, understanding which ingredients are essential for different dishes can be overwhelming. For example, unfamiliarity with spices like garam masala, turmeric, cumin, and coriander may make it difficult for users to prepare even basic Indian dishes.
* **Impact**:  
  This discourages users from attempting to cook Indian food, even if they have the necessary ingredients. They might opt for simpler or more familiar cuisines instead of experimenting with Indian dishes.
* **Cause**:  
  There are limited resources available that guide users through the specific combinations of spices and ingredients needed to make Indian dishes, especially for beginners.

#### **4. Food Waste**

* **Current Situation**:  
  Often, people buy ingredients that they don’t use, or they forget about ingredients that are sitting unused in their fridge or pantry. Without a clear recipe plan, some of these ingredients go bad and are thrown away.
* **Impact**:  
  Food waste contributes to environmental pollution and financial loss. It also results in a missed opportunity to create nutritious meals that could have been prepared with the available ingredients.
* **Cause**:  
  Without an easy way to find recipes that match available ingredients, people tend to buy new groceries unnecessarily, leading to unused ingredients.

#### **5. Difficulty in Exploring Indian Cuisine**

* **Current Situation**:  
  Indian cuisine is known for its complexity and diversity, requiring specific spices and cooking techniques. For those unfamiliar with it, the thought of cooking Indian dishes from scratch can be daunting.
* **Impact**:  
  This discourages people from trying new Indian recipes and prevents them from enjoying the variety of flavors and dishes Indian cuisine has to offer. It may also limit their exposure to the cultural richness of Indian food.
* **Cause**:  
  There is often a lack of guidance on how to combine various spices and ingredients to create a well-balanced Indian dish, making it difficult for beginners or non-experts to confidently cook Indian food.

### **Solution Outline**

The **Indian Food Recommendation System** solves these problems by:

1. **Simplifying Recipe Search**:  
   Users can input the ingredients they have, and the system will return a list of recipes they can make with those ingredients. This eliminates the need to search multiple sources.
2. **Ingredient Optimization**:  
   By suggesting recipes that use up available ingredients, the system helps minimize food waste and encourages users to utilize their kitchen inventory efficiently.
3. **Guidance for Beginners**:  
   The system provides clear, step-by-step recipes that help users understand how to prepare Indian dishes, including the correct use of spices and cooking methods.
4. **Cultural Exploration**:  
   The system introduces users to a variety of Indian dishes, allowing them to explore and experiment with new cuisines in a convenient and accessible manner.

**Main Module:**

1. User Interface (UI) Module

The User Interface (UI) module is responsible for interacting with the end-user. It includes forms, buttons, and other elements that allow users to input ingredients, view recipe results, and navigate through the website.

Features:

Ingredient Input Form:

The user can enter the ingredients they have at home through an input field (e.g., a list of text fields, checkboxes, or a search bar). The UI allows users to easily enter their ingredients and submit them to the system.

Recipe Display Area:

After the user submits their ingredients, the system will return a list of recipes that can be prepared with those ingredients. This list will be displayed in an easy-to-read format with names, images, and brief descriptions.

Search Filters:

Users can filter recipes based on dietary preferences such as vegetarian, vegan, gluten-free, etc., if such features are implemented.

Interactive Features:

Buttons and links that allow users to view more details for each recipe (e.g., preparation time, ingredients required, step-by-step instructions, etc.).

2. Ingredient Processing Module

This module handles the core logic for processing the ingredients entered by the user and matching them with the available recipes in the database.

Features:

Input Validation:

Ensures that the user input is valid (e.g., checking for empty fields or invalid characters).

Ingredient Parsing:

Converts the ingredients entered by the user into a format that can be used to query the database. For example, the ingredients could be split into an array for efficient searching.

Matching Logic:

After parsing the ingredients, the system will compare them against the list of ingredients for each recipe in the database. Recipes are selected based on the number of matching ingredients. More complex algorithms can be used to rank or filter recipes based on user preferences or ingredient availability.

Ranking of Recipes:

Recipes can be ranked according to the number of ingredients the user has. The more ingredients that match, the higher the ranking of the recipe.

3. Database Module

The database module stores all the relevant data, including the recipes, ingredients, and their relationships. It plays a crucial role in managing and querying the recipe data based on user inputs.

Features:

Recipes Table:

This table stores information about each recipe, including:

Recipe ID

Recipe name

Recipe description

Preparation time

Cooking time

Steps to prepare the dish

Required ingredients (linked to the Ingredients Table)

Ingredients Table:

This table stores information about all possible ingredients and their details. Each ingredient can be associated with multiple recipes.

Ingredient-Recipe Mapping Table:

This table links the ingredients to the recipes that require them. It helps in querying recipes based on the ingredients the user has.

Example Schema:

recipes (id, name, description, prep\_time, cook\_time, steps)

ingredients (id, name)

recipe\_ingredients (recipe\_id, ingredient\_id)

The SQL database queries will retrieve the relevant recipes based on the ingredients entered by the user.

4. Recipe Suggestion Module

This is the part of the system that uses the data from the database to generate suggestions for the user.

Features:

Search Functionality:

The system queries the database to find recipes that match the ingredients entered by the user. It can return exact matches or recipes with partial ingredient matches, depending on how the system is designed.

Recipe Ranking:

Once the potential recipes are found, they are ranked according to how closely they match the user's available ingredients. Recipes with more matching ingredients will appear higher in the search results.

Recipe Details:

Once a recipe is selected, users can click to view detailed information, including the full list of ingredients, cooking steps, and any other relevant details like nutritional value or variations of the recipe.

**Algorithm for the Indian Food Recommendation System**

The algorithm of the Indian Food Recommendation System involves several key steps that work together to provide users with personalized recipe recommendations based on the ingredients they already have. The process includes input validation, ingredient matching, recipe ranking, and displaying the results. Below is a step-by-step algorithm outlining the core logic of the system:

Step 1: Input Ingredients from the User

Input: The user enters a list of ingredients they currently have (e.g., using text boxes, checkboxes, or a search bar).

Action: The system receives this input, which can be a list of ingredients such as "onion, tomatoes, potatoes, turmeric," etc.

Step 2: Validate User Input

Input: The list of ingredients entered by the user.

Action:

Check if the input is not empty.

Check if each ingredient entered by the user exists in the database.

Clean the input (e.g., remove extra spaces or invalid characters).

Validation Algorithm:

For each ingredient entered by the user, verify if it exists in the ingredients table of the database.

If any invalid or non-existent ingredient is detected, return an error message asking the user to input valid ingredients.

Step 3: Process Ingredients for Matching

Input: The validated list of ingredients entered by the user.

Action:

Convert the ingredients into a suitable format (e.g., array or list).

Break down the input into individual ingredients and prepare them for querying the database.

Processing Algorithm:

Convert the string input into an array (e.g., ['onion', 'tomatoes', 'potatoes', 'turmeric']).

Query the database for matching recipes that use these ingredients.

Step 4: Query the Database for Matching Recipes

Input: The list of ingredients the user has entered.

Action:

Search the recipe\_ingredients mapping table to identify recipes that use the ingredients provided by the user.

Retrieve recipes that contain at least one or more of the ingredients from the recipes table.

Matching Algorithm:

For each ingredient in the user's input list, check for matches in the recipe\_ingredients table.

Collect all recipe IDs that have matching ingredients.

If partial matches (e.g., fewer ingredients) are allowed, rank the recipes based on how many ingredients match.

Step 5: Rank the Recipes Based on Ingredient Matches

Input: The list of recipes that match the ingredients.

Action:

Rank the recipes based on the number of matching ingredients.

If multiple recipes match the same number of ingredients, consider other factors such as recipe popularity or cooking time to rank them.

Ranking Algorithm:

For each recipe, count the number of ingredients that match the user’s list.

Sort the recipes by the number of matching ingredients in descending order.

Optionally, if two or more recipes have the same number of matching ingredients, apply a secondary ranking criteria (e.g., preparation time, rating).

Step 6: Display the Recipe Suggestions

Input: The ranked list of recipes.

Action:

Display the recipes on the webpage with essential details (name, description, preparation time, steps).

Include links to view more detailed instructions for each recipe.

Display Algorithm:

For each recipe in the ranked list:

Display the recipe name.

Show a short description.

Display preparation time and cooking time.

Optionally, show the number of ingredients the user already has for that recipe.

Provide a link to view full recipe details (e.g., ingredients list, step-by-step cooking instructions).

Step 7: Handle User Interaction (Optional)

Input: User feedback or interaction.

Action:

Allow users to rate recipes or save their favorite recipes.

Optionally, allow the user to share recipes or add custom ingredients.

Interaction Algorithm:

If the user rates a recipe:

Store the rating in the database (e.g., in a ratings table).

Use the ratings to influence recipe rankings in future queries (higher-rated recipes could be ranked higher).

If the user saves a recipe:

Add the recipe to their list of favorites in the database.

Step 8: Repeat or Refine the Search (Optional)

Input: If no satisfactory recipes are found, or if the user wants to refine the search.

Action:

Offer suggestions to enter more ingredients or refine the query by allowing them to specify preferences like vegetarian, vegan, etc.

**Error Handling**

1. Invalid or Empty Input

Scenario:

The user submits the ingredient input form without entering any ingredients or provides invalid input (e.g., special characters or numbers).

Handling Strategy:

Validation:

Check for empty fields before processing.

Validate each ingredient against a predefined list or database.

Strip unwanted characters (e.g., symbols, excessive spaces).

Error Message:

Display a user-friendly message like:

"Please enter valid ingredients to search for recipes."

2. No Matching Recipes Found

Scenario:

The system cannot find any recipes matching the entered ingredients.

Handling Strategy:

Fallback Suggestions:

Provide suggestions for commonly used ingredients to expand the search.

Offer generic recipes that require minimal ingredients.

Error Message:

Display a message like:

"No recipes found with the given ingredients. Try adding more common ingredients like onions or tomatoes."

3. Database Connection Error

Scenario:

The system fails to connect to the SQL database due to incorrect credentials, server issues, or other connection problems.

Handling Strategy:

Display a Friendly Error:

Show a generic error message without revealing sensitive details.

"We are currently experiencing issues. Please try again later."

Log the Error:

Write detailed error logs on the server for debugging.

Retry Mechanism:

Attempt to reconnect a few times before showing the error.

4. Invalid Database Query

Scenario:

A query fails due to missing data, incorrect SQL syntax, or database corruption.

Handling Strategy:

Validate Queries:

Use parameterized queries to prevent SQL injection and syntax errors.

Error Message:

Log the query error and display a user-friendly message:

"Unable to retrieve recipes at the moment."

Fallback:

Return an empty result set instead of crashing the application.

**Technical Overview :**

Technical Overview of the Indian Food Recommendation System

The Indian Food Recommendation System is a web-based application that uses a combination of front-end and back-end technologies to provide users with personalized recipe suggestions based on the ingredients they input. This section details the technical architecture, components, and implementation strategies of the system.

1. System Architecture

The system follows a three-tier architecture:

Presentation Layer (Front-End):

Handles user interaction and displays recipe recommendations.

Technologies: HTML, CSS, JavaScript.

Logic Layer (Back-End):

Processes user input, queries the database, and ranks recipes based on ingredient matches.

Technologies: PHP.

Data Layer (Database):

Stores recipes, ingredients, and their relationships.

Technologies: MySQL.

2. Front-End Overview

The front-end is responsible for providing a user-friendly interface that allows users to:

Input ingredients.

View recipe suggestions.

Interact with the system via search, filters, and navigation.

Key Technologies:

HTML: Structures the web pages (input forms, recipe lists, etc.).

CSS: Styles the application to ensure it is visually appealing and responsive.

JavaScript: Adds interactivity, such as dynamic search and form validation.

Features:

Ingredient Input Form: Accepts user input (e.g., text boxes or checkboxes).

Recipe Display: Shows ranked recipes with relevant details (e.g., preparation time, ingredients).

Dynamic Updates: Uses JavaScript to validate inputs and update the interface without full-page reloads.

3. Back-End Overview

The back-end processes user requests and interacts with the database to retrieve matching recipes. PHP serves as the main programming language for server-side logic.

Key Technologies:

PHP: Manages form submissions, processes inputs, and executes SQL queries.

Apache Server: Hosts the application and handles HTTP requests.

Features:

Input Processing:

Validates and cleans user input before querying the database.

Database Queries:

Retrieves recipes based on matching ingredients.

Ranks recipes based on the number of matching ingredients.

Error Handling:

Manages errors like invalid input, no matches found, or database connection issues.

4. Database Overview

The database stores all recipes, ingredients, and their relationships. MySQL is used for efficient storage and retrieval of data.

Database Schema:

Recipes Table:

id (Primary Key)

name (Recipe name)

description (Short description)

prep\_time (Preparation time)

cook\_time (Cooking time)

steps (Cooking instructions)

Ingredients Table:

id (Primary Key)

name (Ingredient name)

Recipe\_Ingredients Table (Mapping table):

recipe\_id (Foreign Key to Recipes)

ingredient\_id (Foreign Key to Ingredients)

Features:

Scalability:

New recipes and ingredients can be added easily.

Query Optimization:

Indexing and foreign keys improve query performance.

5. Algorithm Overview

The system uses a structured algorithm to recommend recipes:

Input Validation: Checks user input for validity.

Ingredient Matching: Queries the database to find recipes containing the user's ingredients.

Recipe Ranking: Sorts recipes by the number of matching ingredients.

Result Display: Shows recipes with options for detailed views.

6. Key Functionalities

Ingredient-Based Search:

Allows users to input available ingredients.

Matches recipes using the ingredients.

Recipe Ranking:

Ranks recipes based on how many ingredients match the user's input.

Recipe Display:

Displays recipe details, including preparation steps and cooking times.

Error Handling:

Manages issues like invalid input, database errors, or no results found.

7. Security Measures

Input Sanitization:

Prevents SQL injection and XSS attacks by sanitizing user input.

Secure Queries:

Uses parameterized queries in SQL to avoid injection vulnerabilities.

Data Validation:

Ensures only valid data is processed by the system.

8. Deployment

The system can be hosted using:

Local Server: Using XAMPP for development and testing.

**Non-Functional Requirements:**

Non-functional requirements (NFRs) define the quality attributes, performance standards, and constraints that the system must meet to ensure usability, reliability, and scalability. Below is a detailed list of non-functional requirements for the Indian Food Recommendation System:

1. Usability

Ease of Use:

The system must be user-friendly with a simple and intuitive interface.

Users should be able to search for recipes with minimal training or guidance.

Responsiveness:

The system should provide feedback to user actions (e.g., form submissions, button clicks) within 1 second.

Accessibility:

The system should follow accessibility standards to accommodate users with disabilities, such as proper color contrast, alt-text for images, and keyboard navigation support.

2. Performance

Speed:

The system should retrieve and display recipe recommendations within 2-3 seconds for typical queries.

Scalability:

The system must handle up to 500 concurrent users without degradation in performance.

Capacity:

The database should efficiently manage and retrieve data for up to 10,000 recipes and 5,000 unique ingredients.

3. Reliability

Availability:

The system should have at least 99.5% uptime, ensuring it is accessible to users most of the time.

Error Recovery:

In the event of a failure (e.g., database disconnection), the system should display a friendly error message and attempt automatic recovery within 1 minute.

Data Consistency:

The system should ensure that the database remains consistent even during high traffic or crashes.

4. Security

Data Protection:

User data and interactions should be secure, with no sensitive information exposed.

Input Validation:

All user inputs must be sanitized to prevent SQL injection, XSS, and other malicious attacks.

Authentication:

If the system includes user accounts, access to saved recipes or preferences must require authentication.

5. Maintainability

Code Readability:

The codebase should be well-documented, with comments explaining complex logic.

Modularity:

The system should be modular to allow easy updates (e.g., adding new recipes or improving the search algorithm).

Error Logging:

All errors should be logged for developers to debug efficiently.

| **Operating System** | Windows 11, Windows XP, Windows7 etc. |
| --- | --- |
| **Language** | HTML, CSS, Javascript |
| **Database** | PHP & SQL |
| **Browser** | Any of Mozilla, Opera, Chrome etc |
| **Additional Software** | VS Code, XAMPP |

**Requirements:**

| Processor | AMD Ryzen, Pentium III 500MHz or more |
| --- | --- |
| RAM | 255-512 MB |
| Hard disk | 105.1 GB |
| Monitor | 32-Bit Color Monitor |
| Keyboard | PS2 or USB |
| Mouse | PS2 or USB |

**Code :**

**1.Html code :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Indian Food Recommendation</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h1>Indian Food Recommendation</h1>

<form id="ingredientForm">

<label for="ingredients">Enter ingredients (comma-separated):</label>

<input type="text" id="ingredients" name="ingredients" placeholder="e.g., onion, tomato, garlic" required>

<button type="submit">Find Recipes</button>

</form>

<div id="results"></div>

</div>

<script src="script.js"></script>

</body>

</html>

**2.Css code :**

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f8f8f8;

color: #333;

}

.container {

max-width: 600px;

margin: 50px auto;

padding: 20px;

background: white;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

h1 {

text-align: center;

color: #e67e22;

}

form {

margin-bottom: 20px;

}

label {

font-weight: bold;

margin-bottom: 5px;

display: block;

}

input {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

background-color: #e67e22;

color: white;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #d35400;

}

#results {

margin-top: 20px;

}

.recipe {

margin-bottom: 20px;

padding: 10px;

border: 1px solid #ddd;

border-radius: 4px;

background-color: #fafafa;

}

.recipe h2 {

margin: 0 0 10px;

color: #e74c3c;

}

**3.Javascript code :**

document.getElementById("ingredientForm").addEventListener("submit", async function (e) {

e.preventDefault();

const ingredients = document.getElementById("ingredients").value.trim();

if (!ingredients) return;

const response = await fetch("process.php", {

method: "POST",

headers: {

"Content-Type": "application/json",

},

body: JSON.stringify({ ingredients }),

});

const results = await response.json();

const resultsDiv = document.getElementById("results");

resultsDiv.innerHTML = "";

if (results.length === 0) {

resultsDiv.innerHTML = "<p>No recipes found.</p>";

} else {

results.forEach((recipe) => {

const recipeDiv = document.createElement("div");

recipeDiv.classList.add("recipe");

recipeDiv.innerHTML = <h2>${recipe.name}</h2><p><strong>Ingredients:</strong> ${recipe.ingredients}</p><p><strong>Recipe:</strong> ${recipe.recipe}</p>;

resultsDiv.appendChild(recipeDiv);

});

}});

**Sql (structured query language) :**

CREATE DATABASE indian\_foods;

-- Use the database

USE indian\_foods;

CREATE TABLE recipes (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

ingredients TEXT NOT NULL,

recipe TEXT NOT NULL

);

-- Insert Indian food recipes

INSERT INTO recipes (name, ingredients, recipe) VALUES

('Dal Tadka', 'lentils, onion, garlic, tomato, ghee, cumin, turmeric', '1. Cook lentils. 2. Prepare tadka with onion, garlic, tomato, ghee, and spices. 3. Mix together.'),

('Aloo Paratha', 'potato, wheat flour, green chili, cumin, butter', '1. Prepare potato filling. 2. Make dough. 3. Stuff dough with filling and roll into parathas. 4. Cook on a skillet.'),

('Paneer Butter Masala', 'paneer, butter, cream, tomato, cashews, spices', '1. Prepare tomato-cashew paste. 2. Cook paneer in butter with paste and spices. 3. Add cream.'),

('Chole', 'chickpeas, onion, tomato, ginger, garlic, spices', '1. Soak and cook chickpeas. 2. Prepare onion-tomato gravy. 3. Add chickpeas and simmer.'),

('Biryani', 'rice, chicken, yogurt, onion, spices, saffron', '1. Marinate chicken. 2. Layer rice and chicken. 3. Cook with spices.'),

('Poha', 'poha, onion, mustard seeds, turmeric, curry leaves', '1. Rinse poha. 2. Sauté onion, mustard seeds, and spices. 3. Mix poha and cook.'),

('Masala Dosa', 'rice, urad dal, potato, onion, mustard seeds', '1. Prepare dosa batter. 2. Make potato filling. 3. Cook dosa and stuff with filling.'),

('Butter Chicken', 'chicken, butter, cream, tomato, spices', '1. Marinate and cook chicken. 2. Prepare creamy tomato gravy. 3. Add chicken and simmer.'),

('Upma', 'semolina, onion, mustard seeds, curry leaves', '1. Roast semolina. 2. Sauté onion and spices. 3. Add water and semolina.'),

('Gulab Jamun', 'khoya, sugar, cardamom, rose water', '1. Make dough with khoya. 2. Fry balls. 3. Soak in sugar syrup.');

**Code Explanation**

1. HTML Code:

This is the structure of the webpage where the user interacts with the system.

Key Elements:

<html>: The root element that defines the HTML document.

<head>: Contains meta-information about the document, such as character encoding, viewport settings, and external resources like stylesheets and scripts.

<meta charset="UTF-8">: Specifies the character encoding to be used for the page (UTF-8).

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Ensures the page is responsive on mobile devices by setting the viewport width to match the device's width.

<link rel="stylesheet" href="styles.css">: Links to the external CSS file (styles.css) for styling the page.

<title>: Specifies the title of the webpage (shown in the browser tab).

<body>: Contains the content of the page that will be visible to the user.

<div class="container">: A container element used to center the content and give it some styling.

<h1>: The main heading of the page ("Indian Food Recommendation").

<form>: A form where the user can input ingredients. It contains:

<label>: A label for the ingredients input field.

<input>: A text input where users enter a list of ingredients (comma-separated).

<button>: A button to submit the form and find recipes.

<div id="results">: A container to display the results after the user submits the form.

<script src="script.js"></script>: Links to an external JavaScript file (script.js) that will handle the form submission and display of recipes.

2. CSS Code:

This CSS file is responsible for styling the HTML content.

Key Styles:

body: Sets the font family, background color, and text color for the page.

background-color: #f8f8f8; sets a light grey background for the entire page.

.container: Centers the form and result container on the page with padding and a shadow effect.

The max-width: 600px; ensures that the content doesn't stretch too wide.

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); adds a subtle shadow around the container to make it look raised.

h1: Styles the heading with a center alignment and a bright orange color (#e67e22).

form: Adds spacing below the form elements.

input: Styles the text input field, including width, padding, and border radius.

width: calc(100% - 20px); ensures the input field takes up the full width of the container (minus padding).

button: Styles the submit button with a background color, padding, and rounded corners.

The button changes color when hovered: background-color: #d35400;.

#results: Adds some margin at the top to space out the results section.

.recipe: Styles individual recipe results with a border, background color, and padding.

margin-bottom: 20px; separates the recipe cards.

recipe h2: Styles the title of each recipe with a red color (#e74c3c).

3. JavaScript Code:

The JavaScript file (script.js) handles the dynamic functionality of the food recommendation system.

Key Concepts:

Form Submission:

The script listens for the form submission using addEventListener("submit", ...).

When the form is submitted, the default behavior is prevented using e.preventDefault(), which stops the page from reloading.

Getting User Input:

The ingredients entered by the user are retrieved using document.getElementById("ingredients").value.trim().

The .trim() method removes leading and trailing spaces.

Fetching Data:

The fetch() function is used to send a POST request to a server-side script (process.php). The ingredients are sent as a JSON object.

The headers specify that the data sent is in JSON format.

Handling Response:

Once the response is received, the script checks if the response contains any recipes:

If no recipes are found, a message "No recipes found" is displayed.

If recipes are found, a div element is created for each recipe and appended to the results section.

Each recipe div includes the recipe name, ingredients, and instructions.

Dynamic HTML Creation:

A new div for each recipe is created using document.createElement("div").

The content is populated with the recipe name, ingredients, and recipe steps, and then added to the results div.

4. SQL Code:

This SQL code sets up a database to store Indian food recipes.

Key SQL Commands:

Create Database:

sql

Copy code

CREATE DATABASE indian\_foods;

This creates a database named indian\_foods.

Use Database:

sql

Copy code

USE indian\_foods;

This selects the indian\_foods database for the subsequent operations.

Create Table:

sql

Copy code

CREATE TABLE recipes (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

ingredients TEXT NOT NULL,

recipe TEXT NOT NULL

);

This creates a table named recipes with the following fields:

id: A unique identifier for each recipe (auto-incremented).

name: The name of the recipe.

ingredients: A text field containing the list of ingredients for the recipe.

recipe: A text field containing the steps to prepare the dish.

Insert Data:

sql

Copy code

INSERT INTO recipes (name, ingredients, recipe) VALUES

('Dal Tadka', 'lentils, onion, garlic, tomato, ghee, cumin, turmeric', '1. Cook lentils. 2. Prepare tadka with onion, garlic, tomato, ghee, and spices. 3. Mix together.'),

...

This inserts sample recipes into the recipes table, each with a name, ingredients, and recipe instructions.

**LOGIC:**

1. User Interaction (HTML + CSS)

The user visits the webpage where they see an input field and a button.

They are prompted to enter ingredients (comma-separated).

Once they enter ingredients and click the "Find Recipes" button, the form is submitted.

Key UI Elements:

Input Field: Where users enter ingredients.

Submit Button: When clicked, triggers the process of searching for recipes.

Results Section: Displays the recipes fetched from the server.

The HTML and CSS define how the page looks and how the user interacts with the system.

2. Handle Form Submission (JavaScript)

Step-by-Step JavaScript Logic:

Form Submission Event Listener:

When the form is submitted, JavaScript listens for this event via addEventListener("submit", async function (e) {...}).

e.preventDefault() is called to prevent the page from reloading. Normally, form submission would reload the page, but in this case, we want to process the data without reloading.

Extract User Input:

The ingredients entered by the user are retrieved via:

javascript

Copy code

const ingredients = document.getElementById("ingredients").value.trim();

The .trim() method is used to remove any leading or trailing spaces from the input.

Check If Ingredients Are Provided:

The code checks whether the user has entered any ingredients.

javascript

Copy code

if (!ingredients) return;

If the input is empty (i.e., no ingredients are provided), the function simply exits.

Send Ingredients to Server (Fetch API):

The ingredients are sent to the server-side script (process.php) using the fetch API. This is done with a POST request:

javascript

Copy code

const response = await fetch("process.php", {

method: "POST",

headers: {

"Content-Type": "application/json",

},

body: JSON.stringify({ ingredients })

});

The body of the request contains the user input in JSON format ({ ingredients }).

The fetch() function is asynchronous, meaning it will wait for the response from the server.

Handle Server Response:

Once the server responds, the result is processed:

javascript

Copy code

const results = await response.json();

The response is expected to be in JSON format (an array of recipes). If the response doesn't contain any recipes, the user sees a message: "No recipes found."

javascript

Copy code

if (results.length === 0) {

resultsDiv.innerHTML = "<p>No recipes found.</p>";

}

Display Recipes:

If recipes are found, they are dynamically displayed in the results section. The code loops through each recipe and creates HTML elements to show the recipe name, ingredients, and instructions.

javascript

Copy code

results.forEach((recipe) => {

const recipeDiv = document.createElement("div");

recipeDiv.classList.add("recipe");

recipeDiv.innerHTML = `

<h2>${recipe.name}</h2>

<p><strong>Ingredients:</strong> ${recipe.ingredients}</p>

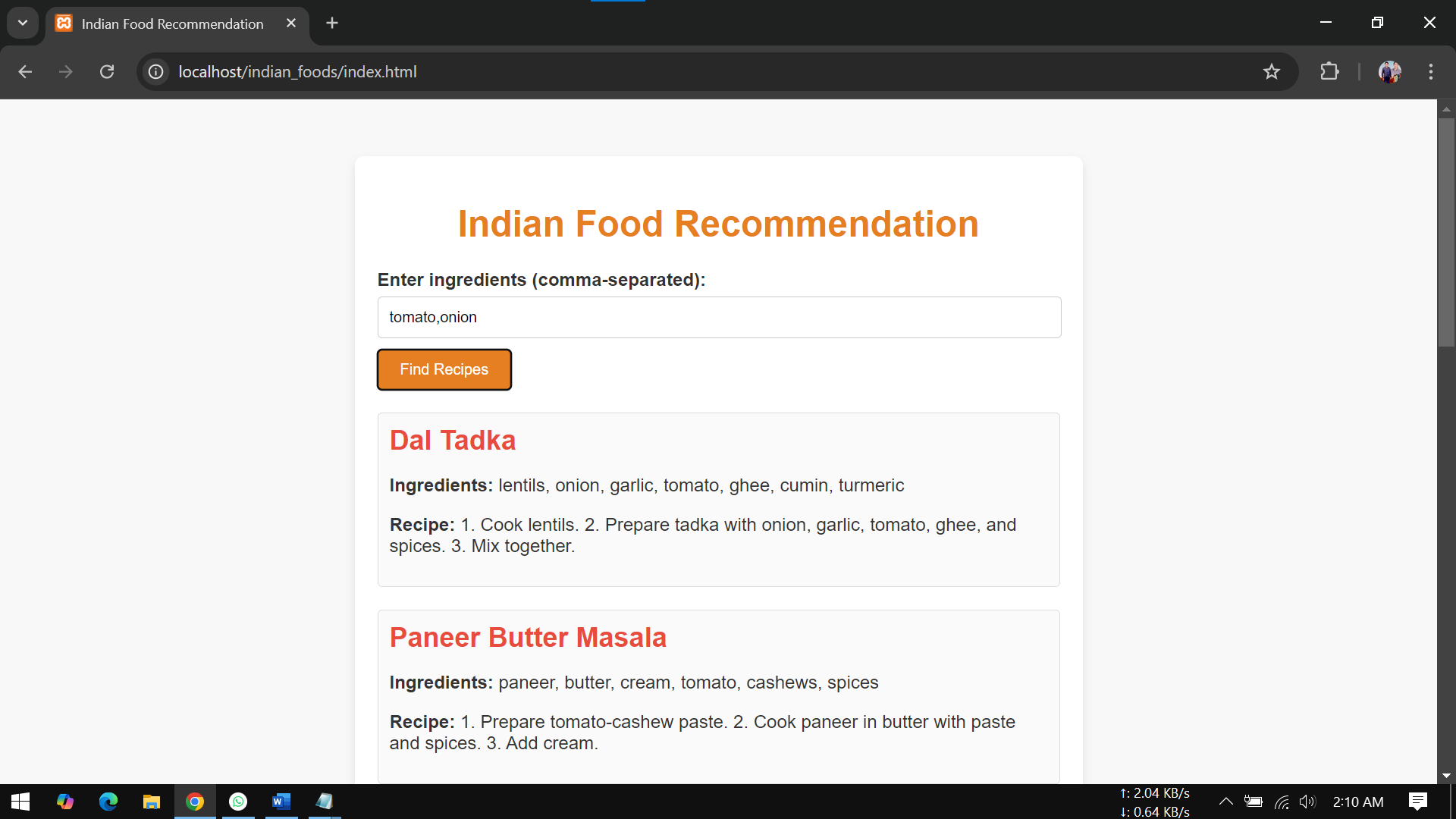
<p><strong>Recipe:</strong> ${recipe.recipe}</p>

`;

resultsDiv.appendChild(recipeDiv);

});

**Output screen :**



**DFD DIAGRAM :**







**Project Explanation**

**Key Components of the System**

User Input: Users enter the ingredients they have available (e.g., onions, tomatoes, garlic).

The system processes this input to identify suitable recipes.

Database: Contains a collection of Indian recipes, each with a list of ingredients, preparation steps, and cooking instructions.

The database also maps relationships between recipes and ingredients for efficient searching.

Recipe Matching:The system matches the user-provided ingredients with the stored recipes in the database.

Recipes are ranked based on the number of matching ingredients, prioritizing those with the highest matches.

Recipe Display: The matched recipes are displayed in a user-friendly format, including preparation and cooking times, required ingredients, and detailed steps.

How the System Works

Input Collection: The user inputs the ingredients into a simple and intuitive form on the website.

Processing:

The system validates the input to ensure accuracy.

Using SQL queries, it searches the database for recipes that contain the entered ingredients.

Ranking and Sorting:

Recipes are sorted by relevance based on the number of matching ingredients.

If no exact match is found, the system suggests recipes with similar ingredients.

Result Display:

The matched recipes are presented to the user, with the option to view detailed cooking instructions.

Additional suggestions, such as complementary ingredients or common substitutions, are provided to improve user experience.

Features

Ingredient-Based Search:

Users can search for recipes using the ingredients they have on hand.

Recipe Ranking:

Recipes are prioritized based on the relevance of the input ingredients.

Detailed Instructions:

Each recipe includes step-by-step instructions, making it easy for users to follow.

Error Handling:

Displays appropriate error messages for invalid inputs or if no recipes are found.

Suggests popular ingredients to refine the search.

Responsive Design:

The website is mobile-friendly and works seamlessly on various devices.

**Software Testing**

Software Testing for Indian Food Recommendation System

Software testing is a critical phase in the development process to ensure the Indian Food Recommendation System meets functional and non-functional requirements. The goal is to identify and fix bugs, verify the correctness of the system, and ensure a smooth user experience. This document outlines the testing methodology, types of tests performed, and the test cases used.

1. Testing Methodology

The system employs a combination of manual and automated testing approaches to validate its functionality and performance.

Manual Testing:

Used for testing the user interface, forms, and error messages.

Verifies user input handling, recipe display, and responsiveness.

Automated Testing:

Employed for testing the back-end logic, database queries, and performance.

Ensures that the system behaves as expected under various scenarios.

2. Types of Testing

A. Functional Testing

Validates that the system's features work as intended.

Input Validation: Ensures valid ingredients are processed, and invalid inputs are handled gracefully.

Recipe Matching: Tests the correctness of recipes displayed based on user-provided ingredients.

Error Handling: Ensures proper error messages are displayed for missing inputs or no matching recipes.

B. Non-Functional Testing

Focuses on the system's quality attributes.

Performance Testing: Measures how the system performs under different loads.

Usability Testing: Evaluates the user interface for intuitiveness and ease of use.

Compatibility Testing: Tests the system across different browsers (Chrome, Firefox, Safari, Edge) and devices (mobile, tablet, desktop).

C. Integration Testing

Validates the interaction between different modules (front-end, back-end, and database).

Ensures data flow between the user interface and the database is seamless.

Confirms that the ranking algorithm produces correct results.

D. Security Testing

Identifies vulnerabilities and ensures data protection.

Tests for SQL injection, cross-site scripting (XSS), and other security threats.

Validates input sanitization and secure query handling.

E. User Acceptance Testing (UAT)

Confirms the system meets user expectations.

Involves feedback from end-users on functionality, speed, and usability

Tools for Testing

Manual Testing:

Browser developer tools for debugging and responsiveness testing.

Tools like Postman for API testing.

Automated Testing:

Selenium for UI and browser compatibility testing.

PHPUnit for back-end testing.

3. Error Tracking and Resolution

Errors encountered during testing are logged in a bug tracker with details such as:

Error description.

Steps to reproduce the error.

Severity level (High, Medium, Low).

Resolution status (Open, In Progress, Fixed).

4. Results and Reports

After testing, a detailed report is generated to summarize:

Total test cases executed.

Number of passed/failed cases.

Areas requiring improvement.

**Conclusion**

The Indian Food Recommendation System is a robust, user-centric application designed to help individuals prepare meals using ingredients they already have. By leveraging modern web development tools such as HTML, CSS, JavaScript, PHP, and SQL, the system offers an efficient and intuitive solution for recipe discovery and food management.

The project highlights the importance of a structured development process, including requirements gathering, design, implementation, and thorough testing. Rigorous functional and non-functional testing ensures that the system is reliable, secure, and user-friendly across various devices and browsers.

This application addresses common challenges faced in meal preparation, such as reducing food wastage, saving time, and promoting the use of available ingredients. It serves both cooking enthusiasts and those new to cooking by providing easy-to-follow recipes tailored to their needs.

Through the implementation of advanced error handling, database integration, and algorithmic recipe matching, the system achieves its goal of delivering a seamless user experience. The project’s modular design and scalability make it adaptable for future enhancements, such as AI-based recipe suggestions and multilingual support.

In conclusion, the Indian Food Recommendation System demonstrates how technology can simplify daily tasks, improve efficiency, and contribute to sustainable living practices, marking it as a practical and impactful solution for everyday users.

**Bibliography and Reference**

HTML5 Documentation

Mozilla Developer Network. "HTML5." MDN Web Docs.

This source provides a comprehensive guide to HTML5 elements and attributes used in creating the web interface of the Indian Food Recommendation application.

CSS3 Styling

W3C. "CSS3: Cascading Style Sheets." World Wide Web Consortium (W3C).

This document outlines CSS3 specifications that help in the visual styling of web pages, which is used in this project for designing the front-end user interface.

JavaScript Guide

Mozilla Developer Network. "JavaScript Guide." MDN Web Docs.

This resource provides a detailed guide to JavaScript, including event handling, functions, and asynchronous operations, which are utilized in the JavaScript code for fetching and displaying recipes.

PHP Manual

The PHP Group. "PHP: Hypertext Preprocessor." PHP Manual.

This manual provides the core documentation for PHP, including how to work with databases and process user input, which is used for handling the backend logic of the food recommendation system.

Food Data Sources

Indian Food Recipes. "Popular Indian Recipes." Allrecipes.

This source provides a large collection of Indian food recipes, serving as a reference for creating a database of dishes in the project.

SQL Injection Prevention in PHP

OWASP. "SQL Injection." Open Web Application Security Project (OWASP).

This article describes how to secure SQL queries in PHP applications, which was crucial in preventing SQL injection attacks in the backend script.