iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XI **Month of publication:** November 2024

DOI: https://doi.org/10.22214/ijraset.2024.65222

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XI Nov 2024- Available at www.ijraset.com

Survey Paper on Recipe Finder Website

Azlaan Khan[1], Prof. Rajeshwari Dandage[2]
Computer Department, Savitribai Phule University

Abstract: ***This paper focuses on the development of a Recipe Finder Website that will help the user to type in the foods that he/she has and get a list of foods that can be prepared with such foods. The system includes a number of filters that help use define a list of restaurants by the type of cuisine, diet, or time needed for cooking. From various recipes carefully collected, a database guarantees strictly unique relevant recommendations. It also recommends Indian recipes cooked on a daily basis in Indian households.***

Keywords: Web Development, Recipe Finder, Ingredient-based Search, Database Management, MySQL, HTML, CSS, JavaScript, User Interface Design, Dietary Filters.

I. INTRODUCTION

With the way of life of today being so fast-paced and technologically driven, many people end up with very less time to plan or prepare meals. As digital solutions are increasing everywhere from how you plan your day to what you eat, meal planning now also has its own set of online platforms and mobile applications providing recipe suggestions. But a downside that many users find is that it offers less relevancy when it comes to suggesting a recipe to make that uses ingredients already at home. Often due to wasted food, inefficient meal preparation, and a generally bad user experience, it's difficult to find practical, ingredient-based recipes which is easy to prepare. Meals are prepared with more thought as food prices go up, and the desire to not waste food and to save time, the need for a smarter and that is more responsive to them, has become evident. As a solution to this problem, Recipe Finder Website is introduced. It is an innovative web platform to search recipes using the ingredients you have in store for a personalized and efficient meal plan. Users can enter one or more ingredients on the website, then filter according to cuisine type, dietary preferences, cooking time, etc. and then get personalized recipe suggestions that best fit their needs. Along with making it convenient to cook this way, it helps users optimize how they use the available ingredients that they have and it also helps to save time, specially for people who follow certain diets or have little time to cook. It also recommends Indian meals cooked in households on a daily basis like dal chawal, different bhajis (sabjis) etc. It provides a flexible but user friendly technical system that helps reduce food waste by making use of ingredients we already have on hand. Leftovers in many households are often spoilt or thrown away because people don't know how to integrate them into meals. By allowing ingredient type based search, this platform nudges users to try out their pantry/kitchen ingredients to create more sustainable cooking and reduce food waste. Users can quickly eliminate irrelevant options with filtering and immediately find recipes fitting their own needs, without wasting time browsing.

II. METHODOLOGY

The Recipe Finder Website has a structured development approach, providing a user friendly platform able to suggest recipes where you already have available ingredients. The methodology is divided into several phases: requirement gathering, system design/ architecture, algorithm design, flowchart, implementation, testing.

A. Requirement Gathering

This initial phase is about understanding what the Recipe Finder Website requires as its core functionalities. Surveys are carried out among potential users of the system and analyzed how the Recipe Finder Website can fill the gaps in already existing recipe websites (recommending Indian recipes, etc.). A user-centered approach is adopted, with the primary requirements defined as:

1) *Ingredient-Based Search:* The users should be able to put in available ingredients and receive corresponding recipe suggestions.
2) *Filter Options:* This platform should allow users to filter the recipes based on cuisine type, cooking time and diet (vegan, gluten free, and more).
3) *User-Friendly Interface:* The interface should be accessible across devices (desktop, tablet, and mobile) and intuitive at the same time.
4) *Recipe Management:* Recipe details like ingredients, preparation steps, cuisine and cooking time should be stored in a database.

©IJRASET: All Rights are Reserved

B. System Architecture

The Recipe Finder Website has three major components: frontend, backend, and database. The system architecture is a client server model, with the user interface of the front end, pushing the requests to the backend server and the server making requests to the database. A data flow is established between the 3 components so that when a user enters a inputs, they will get real time recipe suggestions.

1) *Frontend:* built using HTML (layout), CSS (styling), JavaScript (dynamic responsive user interactions). Users enter ingredients, apply filters and see matching recipes. For a modern and consistent layout, Bootstrap is used, supporting different screen sizes, devices.
2) *Backend:* developed using Python Flask framework as it nicely handles frontend requests. It was chosen for the ease of use and integrating with MySQL database. All user inputs (ingredients and filters) are read by the backend, matched recipes fetched from the database by the backend, and returned to the frontend. RESTful API fetches recipes not present in the database and links frontend and backend if needed.
3) *Database:* stores recipe information in a MySQL relational database. It efficiently allows recipe querying when you give them some of the ingredients out of which a recipe can be made. The database schema has tables for Recipes (stores the details of any recipe - name, ingredients, cuisine, diet type, cooking time, etc.) and Ingredients (stores cookbook data containing a link to the recipes).

C. Algorithm

An important part of the system's functionality is the algorithm for matching recipes. The following steps outline how the system processes user inputs and retrieves relevant recipes:

1) *Start:* Recipe finder System initialization.
2) *User Inputs Ingredients:* Ask the user to type in ingredients and store in array ingredients[].
3) *Set Optional Filters:* Ask the user to choose the filters, if any, for example cuisine type, diet preference and cooking time and store the filter (cuisine, diet, and cooking time) chosen in variables.
4) *Search Recipes In Database*: Initialize an empty list matchingRecipes[], search for recipes based on the ingredients entered by the user using a dynamic SQL query. For each recipe check if the recipe is subset of the list of the ingredients the user inputted.
5) *Apply Filters to Matching Recipes if set:* If any of the optional filters are selected then filter down the matchingRecipes[] on the searched cuisine type, diet preference, and cooking time.
6) *Display the Matching Recipes:* Show all contents from matchingRecipes[], including name, cooking time, ingredients and preparation steps, return this data to the frontend.
7) *End:* Stop the process when the recipes are displayed to the user.

D. Flowchart

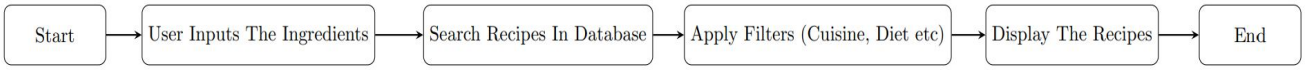


Fig. 1 Flowchart

As shown in Fig. 1, the Recipe Finder system has a flowchart depicting a process flow of successive operations from the input by the user to the final display of recipe results.

1) *Start:* In this step, the Recipe Finder system is initialized, all parts of the system (UI, database, predefined optional filters (cuisine types, dietary restriction)) are loaded. System is set up and once we start sending in inputs, this system will respond. The recipe data can be processed further, so this makes sure the system's backend is connected and is ready to make use of the recipe data.
2) *User Inputs the Ingredients:* The system asks user for the ingredients through a user friendly interface (ingredient search form, user types or picks from a list of ingredients). Then, these ingredients are put in ingredients[] array, which is the central array by which recipes are searched from the database. The user's ingredients inputted are important as they are what first filters the data. First, the system is made to capture, store, and act on each ingredient appropriately.

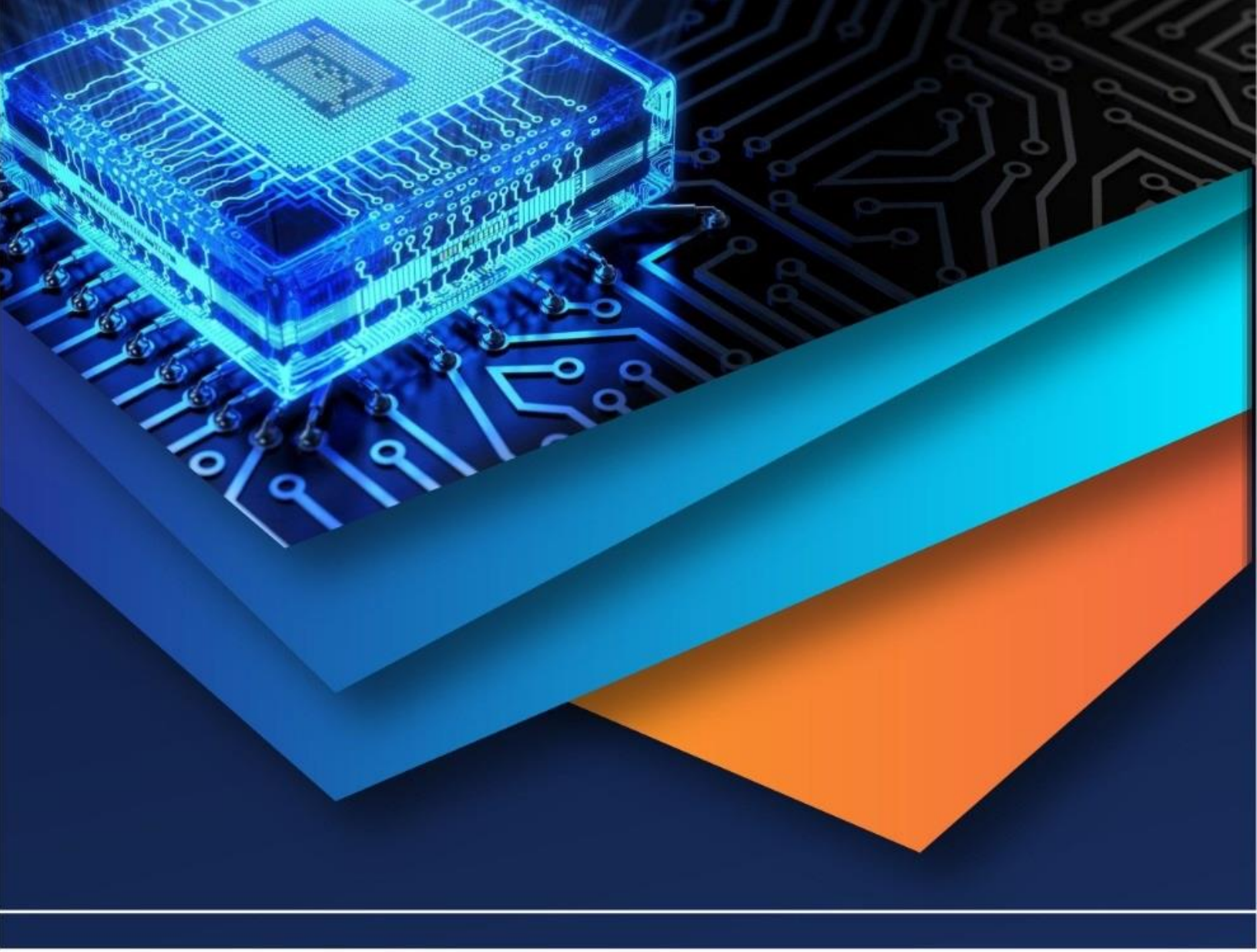
3) *Search Recipes in Database:* When user enters ingredients, the system queries our database in real time to find entries where required ingredients in the recipe match or are a subset of the ingredients from the user by SQL SELECT statement (uses LIKE to find matching records). Recipes not matching the user's inputted ingredients are excluded to reduce a large list of possible recipes down to something the user sees.
4) *Apply Filters:* Once the user's ingredients have been processed into a recipe list, the system returns that recipe list and checks for any optional filter applied. Cuisine type (Italian, Indian, Mexican), dietary preferences (vegan, vegetarian, gluten free) and cooking time (quick, 30 minutes) are examples of these filters. The system applies any of these filters if the user has set one, to the list of matching recipes. Each recipe in the matchingRecipes[] array is checked to see if the recipe fits the user filter criteria. The array is filtered out to only include recipes that match the specified filters and rest are deleted from the array.
5) *Display the Recipes:* The final list of recipes is compiled and shown to the user. Recipes are neatly presented to the user as a list or grid, with each recipe's name, cooking time, required ingredients, and step by step preparation instructions and additional information like user ratings, difficulty level etc. Users can quickly review and select a recipe according to their preferences.
6) End: The user can now choose a recipe and view its details or simply refresh to start a new search with different ingredients again or change filter. This is the final step after a complete search session, but the user could always want to make another search if it's not finalized.

III. CONCLUSION

The Recipe Finder system represents an innovation in the use of already existing ingredients to make more informative meals with reducing waste. It gives personalized recommendations for recipes based on the ingredients entered by the user due to its intuitive interface and adaptable filtering options and an effective algorithm for searching the system. Integration of dietary and cuisine-based filters enhances the user experience as it leads directly to satisfying personal preferences and health needs. An integration of frontend and backend technologies based on database administration and server-side functionality ensures that the platform is completely seamless and dependable for the users. In fact, the Recipe Finder acts as a resource that serves towards meal planning and sustainability by encouraging resource use efficiency within food-related settings.

REFERENCES

[1] Parth P. Sharma and Ankit Gupta, Hardik Parmar, "Food Recipe Recommendation System", IRJET, 2000
[2] Chetaneshwari Vyas, "Food Hub: The Food Recipe", IRJMETS, 2023
[3] Dr. C K Srinivasa, Harshitha S and Mounisha G, "Recipe Finder Web Application Using Convolutional Neural Network", IJRPR, 2024
[4] Dev Pratiksha Pramod and Mrs. Shraddha S. Chafale, "RECIPE FINDER WEB APPLICATION", IRJMETS, 2024
[5] Abhijith V , Athira Sunil and Sanjay Sajeevan, "SMART CUISINE RECIPE FINDER", IJCRT, 2020



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)