

PERSONALISED RECIPE WEBSITE

MAJOR PROJECT

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

**BACHELOR OF SCIENCE
IN
INFORMATION TECHNOLOGY**

Submitted by

**KAMALI M P
22BIT020**

Under the Guidance of

**Dr. K. S. JEEN MARSELINE
Head and Dean, Department of IT and Cognitive Systems**



**SRI KRISHNA ARTS AND SCIENCE COLLEGE
An Autonomous Institution,
Accredited by NAAC with 'A' Grade
Coimbatore – 641008**



MARCH 2025



SRI KRISHNA ARTS AND SCIENCE COLLEGE

Accredited by NAAC with 'A' Grade,

Kuniamuthur, Coimbatore – 641008

DECLARATION

I hereby declare that the Major Project report entitled **“PERSONALISED RECIPE WEBSITE”** submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Science in Information Technology** is an original work submitted and it has not been previously formed the basis for the award of any other Degree, Diploma, Associate ship, Fellowship or similar titles to any other university or body during the period of my study.

Place: Coimbatore

Date:

Signature of the Candidate

KAMALI M P (22BIT020)



SRI KRISHNA ARTS AND SCIENCE COLLEGE

Accredited by NAAC with 'A' Grade,

Kuniamuthur, Coimbatore – 641008

CERTIFICATE

This is to certify that the Major Project report entitled **“PERSONALISED RECIPE WEBSITE”** in partial fulfilment of requirements for the award of the degree of Bachelor of Science in Information Technology is a record of bonafide work carried out by **KAMALI M P (22BIT020)** and that no part of this has been submitted for the award of any other degree or diploma and the work has not been published in popular journal or magazine.

GUIDE

HOD & DEAN

This Major Project is submitted for the viva voce conducted on _____ at Sri Krishna Arts and Science College.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I am ineffably indebted to **Dr. K. Sundararaman, M.Com., M.Phil., Ph.D., CEO,** Sri Krishna Institutions , Sri Krishna Arts and Science College, Coimbatore.

I convey my profound gratitude to **Dr. R. Jagajeevan, MBA., Ph.D., Principal,** Sri Krishna Arts and Science College for giving me this opportunity to undergo this IET.

It is my prime to solemnly express my sense of gratitude to **Dr. K. S. Jeen Marseline, M.C.A., M.Phil., Ph.D., Head and Dean,** Department of IT & Cognitive Systems, Sri Krishna Arts and Science College.

I would like to extend my thanks and unbound sense for the timely help and assistance given by **Dr. K. S. Jeen Marseline, M.C.A., M.Phil., Ph.D., Head and Dean,** Department of IT & Cognitive Systems, Sri Krishna Arts and Science College in completing the Major Project Report. Her remarkable guidance at every stage of my work was coupled with suggestion and motivation.


I owe my gratitude and thanks to the Director and HR Manager, Altitudes Pvt Ltd for permitting me to carry out the project.

I take this opportunity to thank my parents and friends for their constant support and encouragement throughout this training.

KAMALI M P

22BIT020

PROJECT COMPLETION CERTIFICATE




Net Tel Solutions India Pvt Ltd
Excellence in Services

30.05.2024
Coimbatore

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. KAMALI M P (Reg No : 22BIT020)**, of
III B.Sc Information Technology student from **Sri Krishna Arts and Science College** has successfully
completed her Industrial exposure training and Mini Project Entitled **"GOOGLE BOOKS
APPLICATION USING FLUTTER"** in our concern from **25.04.2024 to 25.05.2024**.
During this period, She was sincere and regular in attending all phases of IET program.

For Net Tel Solutions (NTS),



Authorized Signatory

#205, The Spring Building, Block-B, Nehru Street, Gandhipuram, Coimbatore - 641 009.
Mob : 98944 84950 info@nettelsolution.in www.nettelsolution.in

TABLE OF CONTENTS

CHAPTER NO.	CHAPTER TITLE	PAGE NO.
	ORGANIZATION PROFILE	
1	ABSTRACT	8
2	INTRODUCTION	9
3	SYSTEM STUDY 3.1 EXISTING SYSTEM 3.2 PROPOSED SYSTEM 3.3 SOFTWARE AND HARDWARE REQUIREMENTS	11
4	4.1 DESIGN 4.1.1 DATA FLOW DIAGRAM 4.1.2 ER DIAGRAM 4.1.3 INPUT AND OUTPUT DESIGN 4.2 MODULES & DESCRIPTION	29
5	TESTING 5.1 UNIT TESTING 5.2 INTEGRATION TESTING 5.3 FUNCTIONAL TESTING 5.4 PERFORMANCE TESTING 5.5 SECURITY TESTING 5.6 REGRESSION TESTING	37
	CONCLUSION & FUTURE SCOPE	40
	APPENDIX (i) SAMPLE CODE (ii) SCREENSHOTS (iii) BIBLIOGRAPHY	42 46 49

ORAGANIZATION PROFILE

ORGANIZATION NAME : AltiTudeS PVT LTD.

LOCATION : RAM NAGAR, COIMBATORE.

AltiTudeS (ATS) is a dynamic software solutions company based in Coimbatore, Tamil Nadu, specializing in delivering cutting-edge software development services and solutions. With a strong focus on innovation and industry-relevant expertise, ATS offers a range of services, including software development, web development, mobile applications, and data analytics. Through its comprehensive internship programs and practical exposure to real-world projects, the company nurtures talent and bridges the gap between education and industry requirements. ATS is committed to providing top-tier services to its clients while empowering individuals with the skills and knowledge needed to excel in the ever-evolving tech landscape.

Chapter 1

ABSTRACT

This project involves creating a dynamic food recipe website using PHP, integrating cURL and TheMealDB API to provide users with an extensive collection of recipes. The platform enables users to search for recipes based on ingredients, meal types, or categories, offering an interactive and user-friendly experience for cooking enthusiasts. By leveraging TheMealDB API, the website fetches real-time recipe data, including meal instructions, ingredients, preparation time, and images.

Key features include a search functionality that allows users to filter recipes by ingredients or meal categories, and detailed recipe pages with step-by-step cooking instructions. The platform also incorporates cURL for efficient communication with external APIs, fetching live data from TheMealDB and presenting it in a user-friendly format.

To ensure seamless navigation, the website employs responsive design principles using HTML, CSS, and JavaScript, making it accessible across different devices. The combination of PHP, cURL, and TheMealDB API provides a robust, scalable, and interactive platform for users to explore and discover new recipes, transforming their cooking experience.

Chapter 2

INTRODUCTION

2.1 OVERVIEW OF THE PROJECT

In today's fast-paced world, cooking at home has become increasingly popular as people seek healthier and more affordable meal options. With the rise of digital technology, culinary enthusiasts are turning to the internet for inspiration and guidance in their cooking endeavors. This trend has paved the way for innovative platforms that provide access to a wealth of recipes and cooking resources.

The dynamic food recipe website proposed in this project aims to meet the growing demand for an interactive cooking experience by leveraging modern web technologies. By integrating PHP with cURL and TheMealDB API, the platform will offer users a diverse collection of recipes that can be easily searched and explored. This initiative not only facilitates the discovery of new culinary ideas but also empowers users to experiment with various ingredients and cooking techniques.

By incorporating a responsive design, the website ensures accessibility across multiple devices, making it convenient for users to access recipes anytime, anywhere. The project embodies the essence of modern cooking culture, emphasizing convenience, accessibility, and user engagement, ultimately transforming the cooking experience for home chefs and food enthusiasts alike.

2.2 OBJECTIVES

Develop a User-Friendly Interface: Create an intuitive and aesthetically pleasing website design that enhances user experience, allowing users to easily navigate through recipes and related features.

Integrate TheMealDB API: Utilize TheMealDB API to fetch real-time recipe data, ensuring that users have access to a vast and updated collection of recipes, including meal instructions, ingredients, preparation time, and images.

Implement Search Functionality: Enable users to search for recipes based on various criteria such as ingredients, meal types, or categories, facilitating a tailored cooking experience that caters to individual preferences.

Provide Detailed Recipe Pages: Design detailed recipe pages that offer step-by-step cooking instructions, ingredient lists, and tips, helping users follow the recipes with ease.

Enhance Responsiveness: Ensure the website is fully responsive and accessible across different devices, including desktops, tablets, and smartphones, to provide a seamless experience for all users.

Facilitate User Interaction: Implement features that allow users to save, edit, and delete their favorite recipes, enhancing personalization and engagement with the platform.

Utilize cURL for API Communication: Employ cURL to efficiently communicate with TheMealDB API, optimizing data retrieval processes and enhancing the website's performance.

Support Community Engagement: Create opportunities for users to share their cooking experiences, tips, and feedback, fostering a community of cooking enthusiasts that promotes interaction and knowledge exchange.

Ensure Data Security: Implement necessary security measures to protect user data and ensure a safe browsing experience on the website.

Continuous Improvement and Updates: Establish a framework for ongoing updates and improvements to the website, ensuring that it remains relevant and continues to meet the needs of its users over time.

Chapter 3

SYSTEM STUDY

3.1 EXISTING SYSTEM

Existing recipe websites often have static collections of recipes with limited filtering options. Here's a breakdown of their common characteristics:

1. Static Recipe Collections:

- Recipes are manually uploaded by users or administrators.
- Updates to the recipe database are infrequent, with no real-time integration.

2. Basic Search Features:

- Most platforms allow users to search for recipes based on the title or category, but not necessarily by specific ingredients.
- Search functionality is often limited, making it harder for users to find exactly what they want.

3. No API Integration:

Existing recipe websites generally do not utilize external APIs for real-time data. All recipe data is stored in local databases, leading to less dynamic content.

4. No User Interaction with External Data:

- Users cannot see real-time recipe updates or recommendations based on new ingredients or categories.
- Interaction is limited to viewing recipes already present on the website.

5. Non-Responsive or Outdated Design:

- Many older recipe websites are not mobile-friendly.
- Limited interactivity, with basic HTML/CSS implementations that do not provide a modern user experience.

6. Limited Recipe Details:

Recipes may provide only basic information like ingredients and steps but lack detailed meal preparation time, servings, or step-by-step instructions.

3.2 PROPOSED SYSTEM

The proposed system aims to create a dynamic and scalable recipe platform using PHP and TheMealDB API. This system provides real-time access to a wide variety of recipes, offers a more interactive user experience, and resolves the limitations of existing static recipe websites. Here's how the proposed system works:

1. Integration with TheMealDB API:

Real-time Recipe Data: Using cURL and TheMealDB API, the website fetches live recipe data, including meal categories, ingredients, preparation instructions, images, and other meal details. This eliminates the need to manually store recipes on the local server as the API fetches a wide array of recipes dynamically.

2. User Authentication:

Secure User Registration and Login: Implement user authentication using PHP sessions to allow users to register, log in, and maintain their session. This ensures that each user can manage their own recipes securely.

Role-Based Access Control: Only authenticated users can create, view, update, or delete their recipes. This helps to prevent unauthorized access to recipe data.

3. Advanced Search and Filtering:

Custom Search Functionality: Users can search for recipes based on specific ingredients, meal types (e.g., breakfast, dinner), or categories (e.g., vegetarian, seafood). This enhances user experience by making it easy to find relevant recipes.

4. Interactive Recipe Pages:

Comprehensive Recipe Information: Detailed recipe pages display comprehensive information such as ingredients, measurements, cooking instructions, preparation time, and images of the meals. Step-by-step instructions ensure clarity for users, and rich visuals offer an immersive experience.

5. Efficient Data Retrieval with cURL:

Fast API Communication: The platform uses cURL in PHP to communicate efficiently with TheMealDB API, ensuring smooth and fast data retrieval. By fetching and parsing live data from the API, the system ensures that users always have access to the latest and most accurate recipe information.

6. CRUD Operations with MongoDB:

Inserting Recipes: Authenticated users can add their own recipes, which will be stored in MongoDB. This includes the recipe name, ingredients, instructions, and an optional image.

Viewing User Recipes: Users can view all recipes they have added to the platform, with the ability to see details such as ingredients, preparation time, and cooking instructions.

Updating Recipes: Users can edit their recipes, making it easy to correct mistakes or update information as needed.

Deleting Recipes: Users can remove their recipes from the platform if they choose to do so, ensuring they have full control over their content.

7. Responsive Design for Seamless Navigation:

Cross-Device Compatibility: HTML, CSS, and JavaScript are used to ensure that the website has a responsive design. The platform will work seamlessly on desktops, tablets, and mobile devices, providing a consistent user experience across all platforms.

8. Scalability and Robustness:

Dynamic and Scalable Architecture: Leveraging PHP as the server-side language and TheMealDB API ensures the platform can scale as the user base grows without performance bottlenecks. The dynamic nature of the API reduces the need for storing large amounts of data locally, making the platform more efficient.

9. User-Friendly Interface:

Intuitive Design: The interface is designed to be intuitive, focusing on simplicity and usability. Users can easily browse recipes, filter based on their preferences, and access detailed instructions for meal preparation.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS:

Hardware requirements:

Development Machine:

- **Processor:** Dual-core processor (Intel or AMD) or better.
- **RAM:** Minimum 8 GB (16 GB recommended for smooth performance).
- **Storage:** At least 50 GB of free disk space for server, database, and project files.
- **Operating System:** Windows, macOS, or a Linux distribution.

Software Requirements:

1. Web Server:

- **XAMPP / WAMP / LAMP:**
- XAMPP: Apache + PHP + MySQL (Cross-platform)
- WAMP: Windows + Apache + MySQL + PHP
- LAMP: Linux + Apache + MySQL + PHP

2. Database:

MongoDB:

- MongoDB Community Server for database management. [MongoDB Download](#)
- MongoDB Compass (optional, for GUI management): [MongoDB Compass](#)

3. Programming Languages:

- **PHP:** Version 7.0 or higher. [PHP Official Website](#)
- **JavaScript:** For front-end functionality.

4. Libraries and Frameworks:

- **Bootstrap:** For responsive design. [Bootstrap Documentation](#)
- **cURL:** Ensure PHP has the cURL extension enabled for API requests.

5. Development Environment:

Code Editor or IDE:

- Visual Studio Code: [Download VS Code](#)
- PHPStorm: [Download PHPStorm](#)

6. Version Control (optional but recommended):

Git: For version control. [Git Download](#)

7. API Access:

TheMealDB API: No additional software needed, just an internet connection to access the API. TheMealDB API

SOFTWARE DESCRIPTION

XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It stands for X (cross-platform), Apache (the web server), MySQL (the database), PHP (the programming language), and Perl. Designed for ease of use, XAMPP allows developers to create a local web server environment to test and deploy their web applications without the need for a full production server.

Key Features:

1. Cross-Platform Compatibility:

XAMPP is compatible with various operating systems, including Windows, Linux, and macOS, making it accessible to a wide range of developers.

2. Easy Installation:

It offers a straightforward installation process with a user-friendly interface, allowing developers to set up a web server in minutes without complicated configurations.

3. Integrated Components:

XAMPP bundles Apache, MySQL (or MariaDB), PHP, and Perl, providing all the necessary tools for developing web applications in a single package.

4. Database Management:

It includes phpMyAdmin, a web-based interface that simplifies database management, allowing users to easily create, modify, and manage databases.

5. Development Flexibility:

Developers can run applications in a local environment, which helps in testing and debugging before deploying to a live server.

6. Community Support:

XAMPP has a large community of users and extensive documentation, providing a wealth of resources for troubleshooting and learning.

7. Modular Architecture:

Users can add additional components or modules as needed, such as support for different programming languages or databases, enhancing its functionality.

XAMPP is particularly popular among beginners and those who want a quick way to set up a development environment for PHP and web application development. It serves as an essential tool for web developers to build, test, and deploy their applications efficiently.

HTML

HyperText Markup Language (HTML) is the standard markup language used for creating web pages and web applications. It serves as the backbone of web content, providing structure and organization to text, images, and other media on the Internet. HTML consists of a series of elements, or tags, that define various parts of the content, such as headings, paragraphs, links, images, and lists.

Key Features of HTML:

- 1. Structure and Semantics:** HTML uses a hierarchy of elements to structure content meaningfully. This includes headings (<h1> to <h6>), paragraphs (<p>), lists (,), and semantic tags (<header>, <footer>, <article>, <section>), which enhance both accessibility and search engine optimization (SEO).
- 2. Hyperlinks:** HTML allows the creation of hyperlinks using the <a> tag, enabling users to navigate between different web pages or websites, facilitating easy access to information.
- 3. Media Integration:** HTML supports various multimedia elements, including images (), audio (<audio>), and video (<video>), allowing developers to enrich user experiences with dynamic content.
- 4. Forms and Input:** HTML provides the framework for creating interactive forms (<form>), enabling data collection from users through various input types such as text fields, checkboxes, and dropdown menus.
- 5. Cross-Platform Compatibility:** HTML is universally supported by all modern web browsers, ensuring that web content can be accessed and displayed consistently across different devices and operating systems.

CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS controls the layout, colors, fonts, and overall aesthetic of web pages, enabling developers and designers to create visually appealing and user-friendly interfaces. By separating content from design, CSS enhances the flexibility and maintainability of web development.

Key Features of CSS:

- 1. Styling and Layout:** CSS allows for comprehensive styling options, including colors, fonts, spacing, and positioning of elements. It enables developers to create layouts that are visually structured and appealing.

- 2. Selectors and Properties:** CSS uses selectors to target HTML elements for styling. Properties define the visual styles applied to those elements, such as color, font-size, margin, padding, and background-color.
- 3. Responsive Design:** With the introduction of media queries, CSS enables responsive design practices. This allows web pages to adapt their layout and styling based on the screen size and device type, ensuring a consistent experience across desktops, tablets, and smartphones.
- 4. Cascading and Inheritance:** CSS follows the cascading principle, meaning that styles can be applied in a hierarchy, allowing for a more organized and efficient approach to styling. Inheritance allows child elements to inherit styles from their parent elements, reducing redundancy in code.
- 5. Animations and Transitions:** CSS provides capabilities for creating animations and transitions, enhancing user interactions with smooth visual effects. This can improve user engagement and provide feedback on actions like hovering or clicking.
- 6. Frameworks and Preprocessors:** Various CSS frameworks, such as Bootstrap and Tailwind CSS, offer pre-built styles and components that speed up the development process. Additionally, CSS preprocessors like SASS and LESS extend CSS with features such as variables and nested rules, allowing for more organized and maintainable stylesheets.

BOOTSTRAP

Bootstrap is an open-source front-end framework designed to facilitate the development of responsive and mobile-first websites. Created by Twitter, it provides a comprehensive set of tools for building web applications quickly and efficiently. Bootstrap combines HTML, CSS, and JavaScript components to create a cohesive and user-friendly development environment.

Key Features of Bootstrap:

- 1. Responsive Grid System:** Bootstrap includes a powerful grid system that allows developers to create flexible and responsive layouts. The grid system is based on a 12-column layout, enabling the design of complex layouts that adapt to various screen sizes,

from mobile devices to large desktop screens.

- 2. Predefined Components:** The framework offers a wide range of prebuilt components such as navigation bars, modals, buttons, forms, cards, alerts, and more. These components come with default styles and functionality, which can be easily customized to fit the design needs of any project.
- 3. Customizable Styles:** Bootstrap provides a comprehensive set of CSS classes that simplify styling tasks. Developers can quickly apply styles for typography, spacing, colors, and more, helping maintain consistency across the application. Additionally, Bootstrap can be customized using SASS variables to suit the branding and design requirements of a project.
- 4. JavaScript Plugins:** Bootstrap includes several built-in JavaScript plugins that enhance interactivity and functionality. These plugins are designed for common tasks such as modals, tooltips, carousels, and collapsible elements. They can be easily integrated into web applications to improve user experience.
- 5. Mobile-First Approach:** Bootstrap adopts a mobile-first design philosophy, meaning that it prioritizes the development of applications for smaller screens before enhancing them for larger devices. This approach ensures that websites are accessible and functional on a wide range of devices.
- 6. Cross-Browser Compatibility:** Bootstrap is designed to work seamlessly across all modern browsers, ensuring that websites maintain a consistent look and feel regardless of the platform used. This feature reduces the time and effort required for cross-browser testing.
- 7. Community and Support:** Being one of the most popular front-end frameworks, Bootstrap has a large community of developers and extensive documentation. This community support offers developers access to a wealth of resources, tutorials, and examples that can help troubleshoot issues and improve their skills.

JAVASCRIPT

JavaScript is a high-level, dynamic, and interpreted programming language widely used for building interactive and dynamic web applications. As one of the core technologies of the World Wide Web, alongside HTML and CSS, JavaScript enables developers to create rich

user experiences and enhance the functionality of web pages. Initially developed by Netscape as a client-side scripting language, JavaScript has evolved into a versatile language that can be used for both front-end and back-end development.

Key Features of JavaScript:

- 1. Interactivity:** JavaScript allows developers to add interactivity to web pages by responding to user events such as clicks, key presses, and mouse movements. This capability enables the creation of dynamic content that can change without requiring a full page reload, leading to a more engaging user experience.
- 2. Event-Driven Programming:** JavaScript uses an event-driven model, where code execution is triggered by specific events. This makes it ideal for developing applications that require real-time updates, such as chat applications or notifications.
- 3. Client-Side Execution:** JavaScript code is executed on the user's browser, reducing the load on the server and allowing for faster response times. This client-side execution enhances performance by enabling immediate feedback for user interactions.
- 4. Rich APIs:** JavaScript provides a variety of built-in APIs (Application Programming Interfaces) that allow developers to interact with the browser and perform complex tasks. These APIs include DOM manipulation for dynamic content updates, AJAX for asynchronous data retrieval, and Web APIs for accessing device features such as geolocation and local storage.
- 5. Compatibility:** JavaScript is supported by all modern web browsers, making it a universal language for web development. Its widespread compatibility ensures that applications can run seamlessly across different platforms and devices.
- 6. Frameworks and Libraries:** JavaScript has a rich ecosystem of frameworks and libraries, such as React, Angular, and Vue.js, which simplify the development of complex applications. These tools provide pre-built components and architecture that enhance productivity and maintainability.
- 7. Server-Side Development:** With the introduction of environments like Node.js, JavaScript can also be used for server-side development. This allows developers to write both client-side and server-side code in the same language, streamlining the development process and facilitating code reuse.

- 8. Asynchronous Programming:** JavaScript supports asynchronous programming through callbacks, promises, and async/await syntax. This feature enables developers to handle long-running operations, such as data fetching or file uploads, without blocking the main execution thread, resulting in smoother user experiences.

PHP

PHP (Hypertext Preprocessor) is a popular open-source, server-side scripting language designed primarily for web development but also used as a general-purpose programming language. It is widely used for building dynamic and interactive websites and applications. PHP is embedded within HTML and interacts with databases to handle complex server-side operations, such as form submissions, authentication, content management, and data manipulation.

Key Features of PHP:

1. Server-Side Scripting:

- PHP runs on the server, generating dynamic content that is sent to the client's browser. It processes requests, interacts with databases, and delivers HTML output.
- This allows developers to build dynamic web pages that change based on user input or other variables.

2. Integration with HTML and Databases:

PHP can be easily embedded within HTML, making it ideal for web development. It interacts smoothly with databases like MySQL, PostgreSQL, and MongoDB to fetch, insert, update, or delete data.

3. Platform Independence:

- PHP is cross-platform and runs on a variety of operating systems, including Windows, Linux, macOS, and Unix. It can be deployed on different servers, such as Apache, Nginx, or IIS.
- Its wide compatibility makes it a reliable choice for web development across different environments.

4. Dynamic Content Creation:

- PHP can create dynamic content based on various conditions or user interactions, such as displaying personalized messages, processing form submissions, or loading content from a database.
- It is commonly used in Content Management Systems (CMS) like WordPress, Joomla, and Drupal, where the content is fetched and displayed dynamically.

5. Session Management:

PHP supports session handling, enabling user-specific data to persist across multiple page requests. This is crucial for functionalities like user login, shopping carts, and maintaining user preferences.

6. Error Handling:

PHP has built-in support for handling errors and exceptions. Developers can control how errors are displayed and logged, helping with debugging and improving the overall stability of applications.

7. Object-Oriented Programming (OOP):

PHP supports both procedural and object-oriented programming. Developers can create classes, objects, and methods, enabling the development of modular, reusable code that is easier to maintain and extend.

8. Built-In Security Features:

- PHP offers several built-in functions to help secure web applications, such as protecting against SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) attacks.
- Proper input validation and sanitization techniques can be implemented easily to ensure a more secure application.

9. Large Community and Extensive Libraries:

PHP has a massive community of developers, offering a wide range of open-source libraries, frameworks, and tools to extend its capabilities. Popular frameworks like Laravel, Symfony, and CodeIgniter make development faster and more organized.

10. Interfacing with APIs:

PHP can interact with external services and APIs, allowing it to fetch, send, and process data from third-party platforms like REST APIs, SOAP, and other web services. This is particularly useful in applications like social media integrations, payment gateways, and data exchange between services.

Advantages of PHP:

- **Ease of Learning:**

PHP is easy to learn and has a relatively low barrier to entry, making it accessible to beginner developers. Its syntax is straightforward, and it is well-documented, with many learning resources available.

- **Cost-Effective:**

As an open-source language, PHP is free to use, which helps reduce development costs. It also works well with free, open-source database systems like MySQL.

- **Wide Hosting Support:**

PHP is widely supported by most web hosting providers. Its popularity ensures that it is available in most server environments, making deployment convenient and cost-effective.

- **Fast Performance:**

PHP executes scripts quickly, especially when used with modern versions that come with performance improvements and memory management optimizations.

- **Scalability:**

PHP scales well with the growth of websites and applications. Large platforms like Facebook and Wikipedia use PHP, showcasing its ability to handle high traffic and complex operations.

Disadvantages of PHP:

- **Inconsistent Naming Conventions:**

PHP's naming conventions for functions and parameters are inconsistent, making the language somewhat harder to master for developers transitioning from other programming languages.

- **Security Vulnerabilities:**

PHP can be prone to security vulnerabilities if not properly handled, especially in terms of input validation and handling user-generated content. However, these risks can be mitigated with best practices.

- **Interpretation-Based:**

PHP is an interpreted language, which generally makes it slower compared to compiled languages like C or Java. However, modern caching techniques (such as Opcode caching) help improve performance.

Role in Web Development:

PHP plays a pivotal role in web development, powering over 75% of websites worldwide. It excels in creating dynamic content, making it one of the most preferred languages for building web applications, content management systems, e-commerce platforms, and social media sites. Its integration with databases, ability to handle server-side tasks, and widespread hosting support make PHP a reliable and versatile choice for web developers.

With the growing adoption of frameworks like Laravel, PHP development has become more structured and efficient, helping developers build scalable and maintainable applications with ease. The ease of learning, coupled with its extensive community support, ensures that PHP will continue to be a dominant force in the world of web development.

Curl

cURL (Client URL) is a command-line tool and library used for transferring data with URLs. It supports a wide range of protocols, including HTTP, HTTPS, FTP, FTPS, and many others, making it an essential utility for web developers, system administrators, and anyone who

needs to interact with web services or APIs. Originally developed by Daniel Stenberg in 1997, cURL has become a popular tool for performing data transfers due to its versatility and ease of use.

Key Features of cURL:

- 1. Protocol Support:** cURL supports numerous protocols, including HTTP, HTTPS, FTP, FTPS, SCP, SFTP, and LDAP, allowing users to interact with various types of web services and servers. This broad protocol support makes cURL a go-to tool for many different tasks.
- 2. Data Transfer:** cURL excels in performing data transfers between clients and servers. It allows users to upload and download files, send data to APIs, and retrieve web pages, making it highly valuable for tasks like API testing, web scraping, and automating file transfers.
- 3. Command-Line Interface:** cURL operates through a command-line interface, providing a straightforward way to execute various data transfer tasks without the need for a graphical user interface (GUI). This makes it ideal for scripting and automation in both local and remote environments.
- 4. Robustness:** cURL is known for its reliability and error-handling capabilities. It can handle a wide range of HTTP errors and issues during data transfers, allowing users to troubleshoot problems effectively.
- 5. SSL/TLS Support:** cURL provides built-in support for SSL/TLS, enabling secure data transfers over HTTPS. This feature is essential for maintaining data privacy and security when communicating with remote servers.
- 6. Customizable Requests:** With cURL, users can easily customize their requests. This includes setting HTTP headers, handling cookies, managing redirects, and specifying request methods (GET, POST, PUT, DELETE, etc.). This level of customization is particularly useful when interacting with RESTful APIs.
- 7. Integration with Programming Languages:** cURL is available in multiple programming languages, including PHP, Python, Ruby, and JavaScript, through libraries and extensions. This allows developers to easily integrate cURL functionality into their applications, enabling seamless data transfers and API interactions.

8. **Support for Form Submission:** cURL can simulate form submissions by sending data in various formats, such as URL-encoded or multipart/form-data. This capability makes it useful for testing web forms and APIs that accept data submissions.

Use Cases:

- **API Integration:** cURL is widely used for interacting with APIs, allowing developers to send requests, retrieve data, and handle responses efficiently.
- **File Transfers:** It is commonly used for uploading and downloading files to and from servers, especially in automated scripts or cron jobs.
- **Web Scraping:** cURL can be used to fetch HTML pages or data from websites, making it a valuable tool for web scraping projects.
- **Testing and Debugging:** Developers use cURL to test and debug API endpoints, verifying the correctness of requests and responses.

MONGODB

MongoDB is a popular, open-source, NoSQL database management system that is designed for high performance, scalability, and flexibility. Developed by MongoDB Inc. in 2007, it is built to handle unstructured data and provides a document-oriented data model, allowing developers to work with data in a more intuitive way compared to traditional relational databases.

Key Features of MongoDB:

1. **Document-Oriented Storage:** Unlike relational databases that store data in tables with fixed schemas, MongoDB uses a flexible schema-less model where data is stored in JSON-like documents (BSON format). This allows developers to represent complex data structures easily and adapt the database schema without downtime.
2. **Scalability:** MongoDB is designed to scale out easily by adding more servers to a cluster. It supports sharding, which partitions data across multiple servers, ensuring high availability and improved performance for large-scale applications.

3. **High Performance:** MongoDB provides high performance for read and write operations. It uses indexes to improve query performance and supports in-memory processing for fast data access. Its ability to handle large volumes of data makes it suitable for real-time analytics and applications requiring quick data retrieval.
4. **Rich Query Language:** MongoDB offers a powerful query language that supports a variety of query types, including filtering, sorting, and aggregating data. It allows developers to perform complex queries and data manipulations easily, enhancing the ability to analyze and extract insights from data.
5. **Aggregation Framework:** The aggregation framework in MongoDB allows for the efficient processing of data records and returning computed results. It supports operations like grouping, filtering, and transforming data, enabling powerful data analysis and reporting.
6. **Automatic Failover and Replication:** MongoDB provides automatic replication and failover capabilities through replica sets. This ensures high availability and data redundancy, as one or more copies of the data are maintained across different servers.
7. **Support for Geospatial Data:** MongoDB includes built-in support for geospatial data, allowing developers to create location-based applications. It provides geospatial queries to store, index, and query geographical data efficiently.
8. **Flexible Data Model:** The flexible data model allows developers to evolve their applications and data structures without worrying about predefined schemas. This adaptability is particularly useful in agile development environments where requirements can change rapidly.
9. **Extensive Ecosystem:** MongoDB has a rich ecosystem of tools and libraries that enhance its capabilities, such as MongoDB Atlas (cloud database), MongoDB Compass (GUI for data exploration), and various client drivers for different programming languages.

Use Cases:

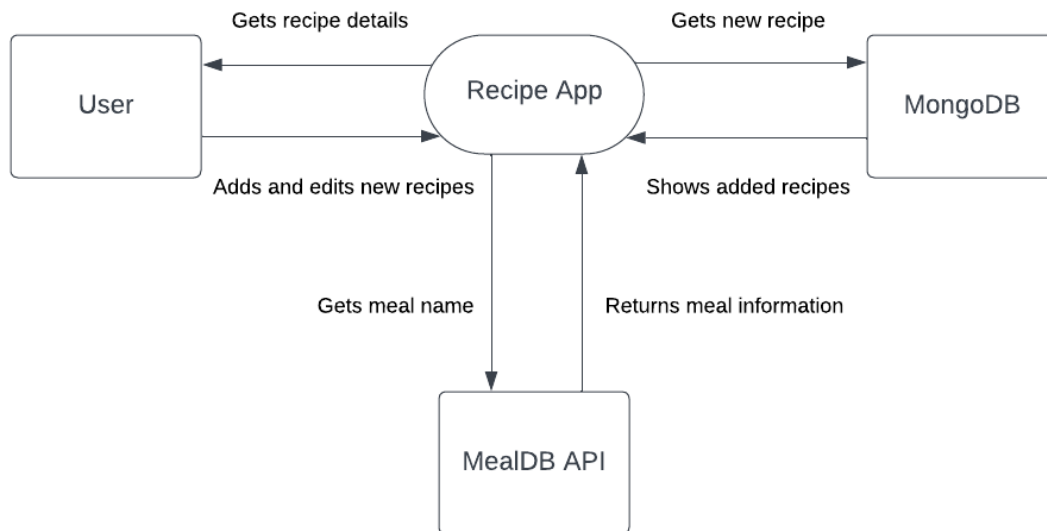
- **Content Management Systems:** MongoDB is ideal for content management systems where data structures can vary and evolve over time.

- **Real-Time Analytics:** Its high performance and ability to process large volumes of data make MongoDB suitable for real-time analytics and big data applications.
- **IoT Applications:** MongoDB's ability to handle massive amounts of unstructured data from various sources makes it a popular choice for Internet of Things (IoT) applications.
- **E-commerce Platforms:** Its flexible data model is advantageous for managing product catalogs, user profiles, and transaction histories in e-commerce applications.

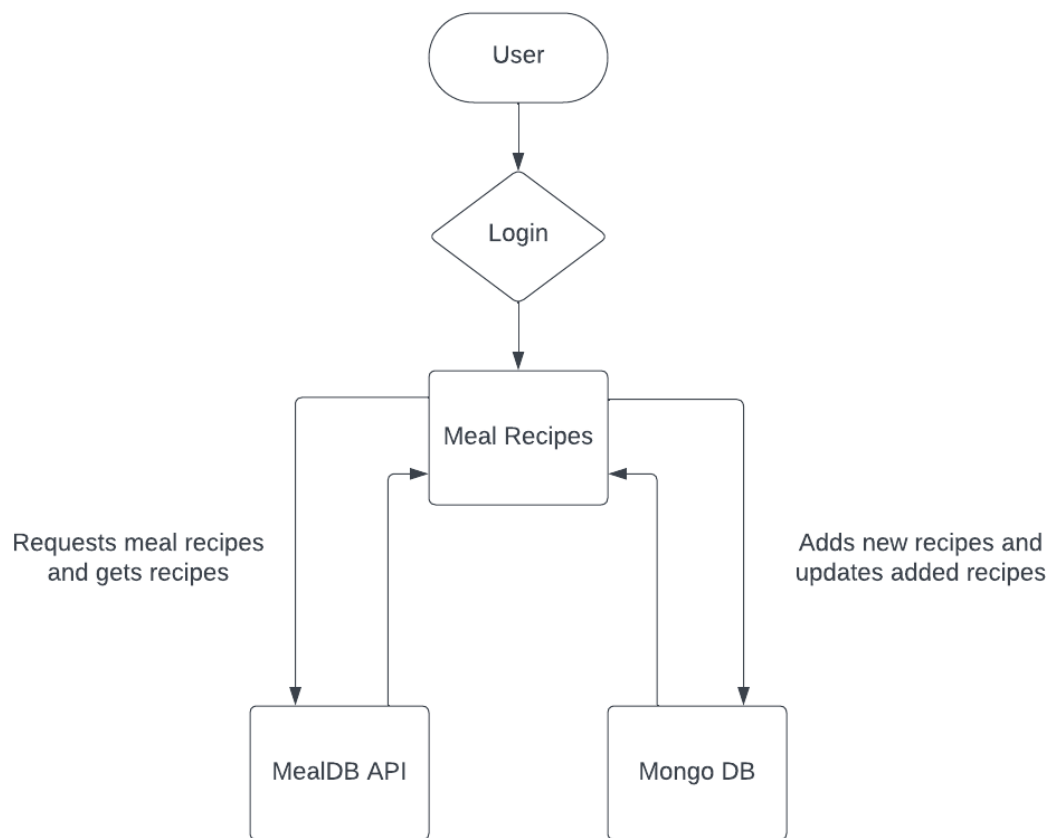
Chapter 4

4.1 DESIGN:

4.1.1 DATA FLOW DIAGRAM:



4.1.2 ER DIAGRAM :



s

4.1.3 INPUT AND OUTPUT DESIGN

The major inputs and outputs and major functions of the system are following:

Input Design:

The Food Recipe website will receive user input through various methods on the web interface. Below is a breakdown of potential user inputs:

1. Search Bar for Recipes:

Input Type: Textbox (Single-line input)

Description: Users can enter specific ingredients or keywords (e.g., "chicken," "pasta," "vegetarian," etc.) to search for recipes that match their query.

Validation:

- The input must be non-empty.
- Only alphanumeric characters and commas are allowed (if multiple ingredients are entered).

2. Filters (Meal Type and Category:

Input Type: Dropdown Menus

Meal Type Filter:

- Options: "Breakfast," "Lunch," "Dinner," "Snack," etc.
- Description: Users can select the type of meal they are looking for.
- Validation: The user must select one of the predefined meal types.

Recipe Category Filter:

- Options: "Vegan," "Vegetarian," "Non-Vegetarian," "Dessert," etc.
- Description: Users can filter recipes by specific categories like dietary preferences.
- Validation: The user must select one of the predefined categories.

3. Submit/Search Button:

Input Type: Button (Search)

Action: After filling out the search bar and/or selecting filters, the user clicks the search button to initiate the search.

Validation: The user must enter at least one input (ingredient, meal type, or category) before submitting the search.

Output Design:

The output of the website will display the search results, recipe details, and any relevant information retrieved from the external API (TheMealDB). Below are examples of potential outputs:

1. Search Results Page:

Output Type: List of Recipes

- **Displayed Elements:**
 - **Recipe Name:** The title of each recipe.
 - **Image:** A thumbnail image of the recipe.
 - **Meal Type:** Category of the recipe, such as "Breakfast," "Dinner," etc.
 - **Preparation Time:** Time required to make the recipe.
 - **Recipe Link:** A clickable link or button leading to the detailed recipe page.
- **Search Results Behavior:**
 - **No Results Found:** If no results match the search criteria, the system will display a message like "No recipes found. Please try different ingredients or categories."
 - **Matching Results:** The system will display a list of recipes matching the user's search criteria.

2. Detailed Recipe Page:

Output Type: Recipe Information

Displayed Elements:

- **Recipe Name:** The title of the recipe.
- **Ingredients:** A list of ingredients required for the recipe, along with quantities.
- **Step-by-Step Instructions:** Detailed instructions for preparing the meal.
- **Meal Type:** The type of meal (e.g., "Lunch," "Dinner").
- **Category:** Recipe category (e.g., "Vegan," "Non-Vegetarian").
- **Preparation Time:** Time needed for preparation.
- **Image:** A high-quality image of the final dish.
- **Recipe Video** (if available): A video tutorial or guide for making the recipe.

3. System Output - API Responses (TheMealDB):

The system will send requests to TheMealDB API and receive responses in JSON format, which the website will parse and present to the user.

Displayed Elements from API Response:

- Recipe Name
- Recipe Image
- Meal Type
- Meal Instructions
- Ingredients
- Preparation Time

4. Recommendations (Optional):

Based on user history or behavior (e.g., frequent search queries), the website may also display a "Recommended for You" section with recipes the user may be interested in. This could include:

- Recommended Recipe Name

- Recommended Image
- Link to Recipe Page

4.2MODULES & DESCRIPTION:

1. User Authentication Module

- **Purpose:** To manage user access and secure personal data.
- **Key Features:**
 - **Registration:** Allows new users to create accounts by providing their email, password, and other necessary information.
 - **Login/Logout:** Enables users to log in to their accounts and securely log out.
 - **Session Management:** Uses PHP sessions to maintain user authentication states throughout their interaction with the website.
 - **Password Management:** Includes password hashing for security, along with features for password recovery and reset.

2. Recipe Data Fetching Module

- **Purpose:** To fetch real-time recipe data from TheMealDB API.
- **Key Features:**
 - **cURL Integration:** Utilizes cURL to send requests to TheMealDB API and retrieve data, including meal names, categories, ingredients, preparation instructions, and images.
 - **Dynamic Data Display:** Presents fetched recipes dynamically on the website, ensuring users always have access to the latest recipe information without the need for local data storage.

3. Recipe Search and Filtering Module

- **Purpose:** To enhance user experience by allowing them to find recipes easily.
- **Key Features:**

- **Ingredient Search:** Users can input specific ingredients to find matching recipes.
- **Meal Type and Category Filters:** Allows users to filter recipes based on categories such as breakfast, lunch, dinner, vegetarian, etc.
- **Keyword Search:** Supports searching by keywords related to meal names or descriptions.

4. User Recipe Management Module

- **Purpose:** To allow users to create, view, update, and delete their own recipes.
- **Key Features:**
 - **Add Recipe:** Users can submit their recipes, which include name, ingredients, instructions, and an optional image. This data is stored in MongoDB.
 - **View Recipes:** Displays all recipes submitted by the logged-in user, allowing them to access and view their own content easily.
 - **Edit Recipe:** Provides functionality for users to update existing recipes, enabling them to modify any details as needed.
 - **Delete Recipe:** Allows users to remove their recipes from the database, giving them complete control over their submissions.

5. Recipe Details Module

- **Purpose:** To provide detailed views of each recipe for users.
- **Key Features:**
 - **Comprehensive Information Display:** Shows recipe name, image, ingredients, preparation instructions, and cooking time.
 - **Step-by-Step Instructions:** Clearly outlines each step in the cooking process, enhancing usability for users.
 - **Responsive Design:** Ensures that recipe details are presented in a user-friendly format across all devices (desktops, tablets, and smartphones).

6. Navigation and User Interface Module

- **Purpose:** To provide a seamless and intuitive navigation experience for users.

- **Key Features:**

- **Responsive Layout:** Utilizes HTML, CSS, and JavaScript to ensure the website adapts to various screen sizes.
- **Intuitive Menu Structure:** Features a clear navigation menu that allows users to access different sections of the website easily.
- **User-Friendly Design:** Emphasizes simplicity and clarity in layout, enhancing overall user engagement.

7. Error Handling and Validation Module

- **Purpose:** To ensure robust application performance and user experience.
- **Key Features:**
 - **Input Validation:** Validates user input during registration, login, recipe submission, and search queries to prevent errors and enhance security.
 - **Error Messages:** Provides user-friendly error messages and alerts to guide users when something goes wrong (e.g., invalid input, failed API requests).

8. Database Module (MongoDB)

- **Purpose:** To handle data storage and retrieval for user-generated recipes.
- **Key Features:**
 - **User Data Storage:** Stores user authentication details and session information securely.
 - **Recipe Storage:** Manages user-submitted recipes, allowing for efficient CRUD operations.
 - **Scalability:** Leverages MongoDB's scalability to support growing data requirements as more users engage with the platform.

Chapter 5

TESTING

System testing is a stage of implementation which is aimed at ensuring that the system works accurately and efficiently. The primary goal of the test process is to make sure that the application performs as discussed in the requirement and specifications.

Testing begins at the module level and works towards the integration of the entire computer based system.

Testing methodologies

- Unit Testing
- Integration Testing
- Functional Testing
- Performance Testing
- Security Testing
- Regression Testing

5.1 Unit Testing

Unit testing focuses on validating individual components or modules of the application to ensure that each part functions correctly in isolation. In the context of the recipe website, unit tests can be written for various functions, such as:

- **Recipe Fetching:** Testing functions that retrieve recipes from TheMealDB API to ensure they return the expected data structure.
- **User Authentication:** Validating the logic for user registration and login processes to confirm that credentials are correctly authenticated.
- **Data Manipulation:** Testing functions responsible for inserting, updating, and deleting recipes in MongoDB.

5.2 Integration Testing

Integration testing evaluates how different modules of the application work together. This type of testing ensures that the interaction between various components, such as the front-end interface and the back-end server, operates smoothly. For the recipe website, integration testing can include:

- **API Interaction:** Testing the communication between the PHP backend and TheMealDB API to verify that data is correctly fetched and displayed.
- **Database Operations:** Validating that data inserted or modified in MongoDB reflects accurately in the user interface.
- **User Flows:** Ensuring that user actions, such as adding or viewing recipes, trigger the expected sequences of events across different modules.

5.3 Functional Testing

Functional testing assesses the application against its specified requirements, focusing on user interactions and expected outcomes. This type of testing can be carried out through:

- **User Interface Testing:** Validating that all UI components function as intended, including buttons, forms, and navigation links.
- **Feature Testing:** Ensuring that all core features, such as searching for recipes, filtering by ingredients, and viewing details, work correctly from a user's perspective.
- **User Acceptance Testing (UAT):** Engaging real users to test the application and provide feedback on its functionality, usability, and overall experience.

5.4 Performance Testing

Performance testing evaluates the responsiveness, speed, and stability of the application under various conditions. This includes:

- **Load Testing:** Simulating multiple users accessing the website simultaneously to assess how well the application handles high traffic.

- **Stress Testing:** Pushing the application beyond its limits to determine its breaking point and how it recovers from failures.
- **Response Time Testing:** Measuring the time taken to fetch data from TheMealDB API and MongoDB, ensuring it meets acceptable thresholds for a smooth user experience.

5.5 Security Testing

Security testing aims to identify vulnerabilities in the application that could be exploited by malicious users. Key aspects of security testing for the recipe website include:

- **Authentication Testing:** Ensuring that the login and registration processes are secure and protect user credentials.
- **Input Validation:** Testing forms to ensure that user inputs are properly validated and sanitized to prevent SQL injection and cross-site scripting (XSS) attacks.
- **Data Protection:** Verifying that sensitive user information, such as passwords and email addresses, is encrypted and stored securely.

5.6 Regression Testing

Regression testing is performed after changes are made to the application, such as bug fixes or feature enhancements, to ensure that existing functionality remains unaffected. This involves:

- **Re-running Test Cases:** Executing previously conducted tests to confirm that new code does not introduce any new defects.
- **Automated Testing:** Utilizing testing frameworks to automate the execution of regression tests, improving efficiency and coverage.

Chapter 6

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

The dynamic food recipe website, built using PHP and integrated with TheMealDB API, represents a significant advancement in how users can explore, create, and manage recipes online. By combining real-time data access, interactive features, and a user-friendly interface, this platform offers an engaging experience for cooking enthusiasts.

The implementation of user authentication ensures secure access, allowing individuals to create and manage their own recipes while benefiting from a wealth of culinary resources. The use of cURL to fetch live data from TheMealDB API guarantees that users always have access to the most current and diverse recipes, alleviating the limitations of traditional static recipe sites.

Moreover, the advanced search and filtering capabilities enhance user experience, making it easier to discover new recipes based on personal preferences. The detailed recipe pages provide comprehensive information, including ingredients and step-by-step cooking instructions, which foster confidence and creativity in the kitchen.

The responsive design ensures accessibility across various devices, catering to the needs of modern users who engage with content on desktops, tablets, and smartphones. As a scalable and robust platform, it is well-equipped to accommodate future growth, supporting an expanding user base without compromising performance.

In summary, this project not only transforms the way users interact with recipes but also encourages culinary exploration and creativity. By leveraging modern web technologies and APIs, the platform stands as a comprehensive resource for anyone looking to enhance their cooking experience, making it a valuable addition to the digital culinary landscape.

FUTURE SCOPE:

The Food Recipe Website has a solid foundation with its current features, such as recipe

search, filtering by ingredients and categories, integration with TheMealDB API, and user-friendly design. However, there are several areas in which the project can be expanded and improved in the future to offer more functionality, better user engagement, and a more dynamic experience. The Food Recipe Website has significant potential for future expansion. By adding user-centric features such as personalized recommendations, social sharing, advanced search filters, and integrating AI and machine learning, the platform can become a powerful cooking tool for food enthusiasts globally. As the website grows, it can transition from a simple recipe aggregator to a comprehensive cooking platform that empowers users to explore, create, and share culinary experiences.

APPENDIX

SAMPLE CODE

```
<?php
session_start();

if (isset($_SESSION["active"]) && $_SESSION["active"] == 1) {

    ?>

    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

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

        <title>MealDB - Search Recipe</title>

        <!-- Bootstrap CSS -->

        <link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

    </head>

    <body>

        <nav class="nav nav-tabs flex-row justify-content-end">

            <a class="nav-link active" href="index.php">Home</a>

            <a class="nav-link" href="add_recipe.php">Add recipe</a>

            <a class="nav-link" href="view_recipe.php">View recipe</a>

            <a class="nav-link link-danger" href="logout.php">Log Out</a>
```

```
</nav>
```

```
<div class="container mt-5">
```

```
<h1 class="text-center mb-4">Search for a Recipe</h1>
```

```
<form action="index.php" method="GET" class="d-flex justify-content-center mb-4">
```

```
<input type="text" name="search" class="form-control me-2" style="width: 300px;"
```

```
placeholder="Enter meal name" required>
```

```
<button type="submit" class="btn btn-primary">Search</button>
```

```
</form>
```

```
<?php
```

```
if (isset($_GET['search'])) {
```

```
    $mealName = $_GET['search'];
```

```
    $url = "https://www.themealdb.com/api/json/v1/1/search.php?s=" .  
urlencode($mealName);
```

```
    $curl = curl_init($url);
```

```
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
```

```
    if (!$response = curl_exec($curl)) {
```

```
        echo "<div class='alert alert-danger'>cURL Error: " . curl_error($curl) . "</div>";
```

```
    } else {
```

```

$data = json_decode($response, true);

if (!empty($data['meals'])) {

    echo "<h2 class='text-center'>Results for '" . htmlspecialchars($mealName) .
    "'</h2>";

    echo "<div class='row'>";

    foreach ($data['meals'] as $meal) {

        echo "<div class='col-md-4 mb-4'>";

        echo "<div class='card'>";

            echo "<img src='" . $meal['strMealThumb'] . "' class='card-img-top'
alt='Meal Image'>";

        echo "<div class='card-body'>";

        echo "<h5 class='card-title'>" . $meal['strMeal'] . "</h5>";

        echo "<a href='meal.php?id=" . $meal['idMeal'] . "' class='btn btn-
primary'>View Details</a>";

        echo "</div></div></div>";

    }

    echo "</div>";

} else {

    echo "<div class='alert alert-warning'>No meals found. Try another
search.</div>";

}

}

curl_close($curl);

```

```

    }

    ?>

</div>

<!-- Bootstrap JS and Popper.js -->

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.7/dist/umd/popper.min.js"></script>

    <script      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.min.js"></script>

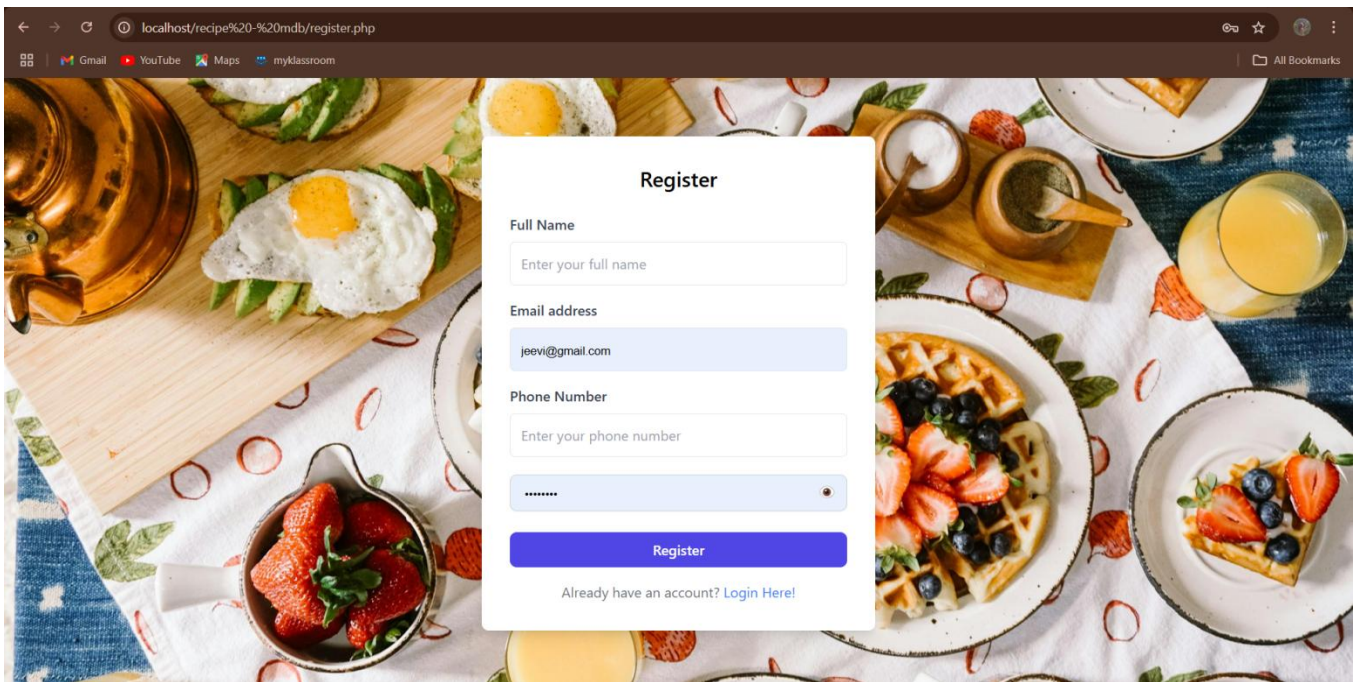
</body>

</html>

<?php
} else {
    header("Location:login.php");
}

```

1. SCREENSHOTS



localhost/recipe%20-%20mdb/register.php

[Gmail](#) [YouTube](#) [Maps](#) [myclassroom](#) [All Bookmarks](#)

Register

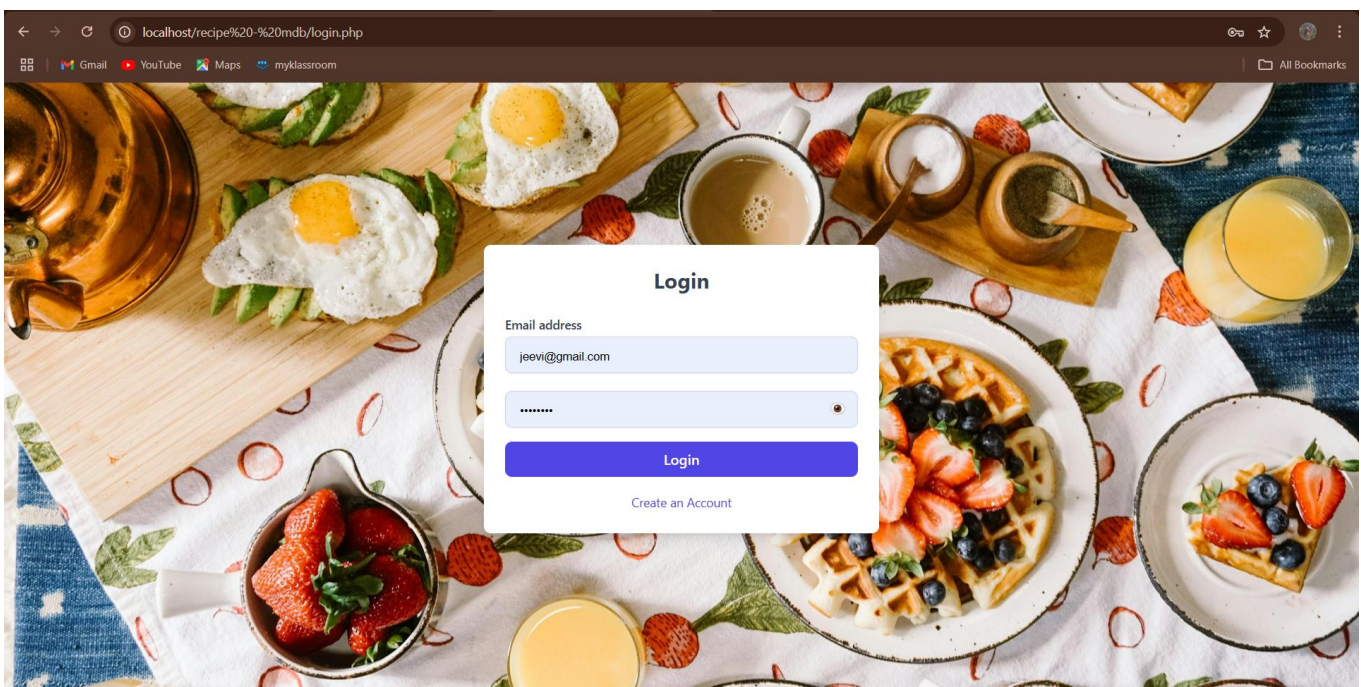
Full Name

Email address

Phone Number

[Register](#)

[Already have an account? Login Here!](#)



localhost/recipe%20-%20mdb/login.php

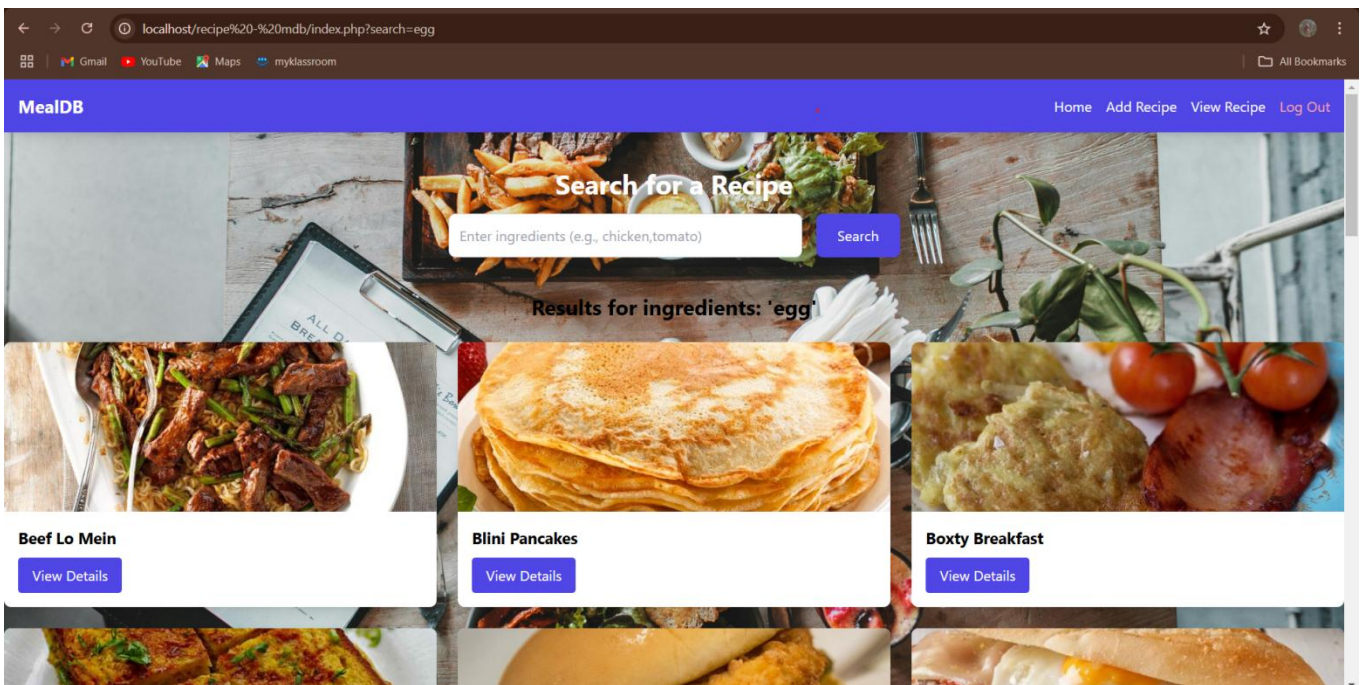
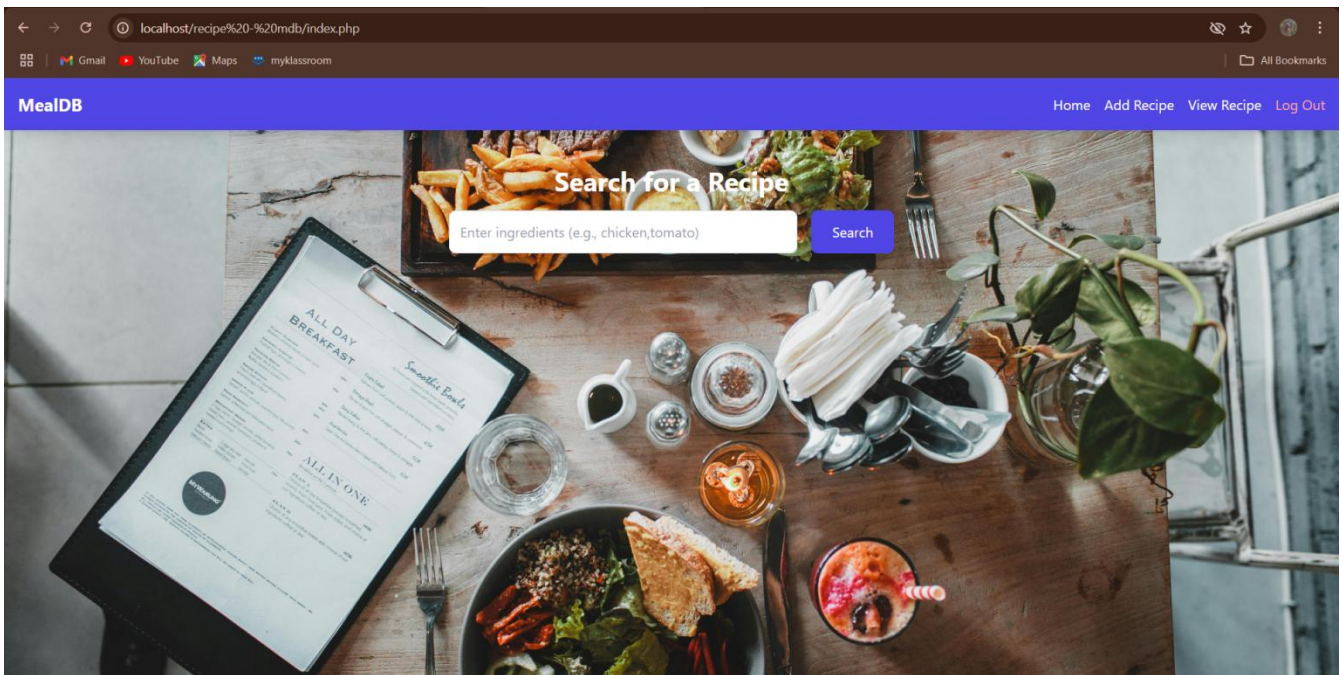
[Gmail](#) [YouTube](#) [Maps](#) [myclassroom](#) [All Bookmarks](#)

Login

Email address

[Login](#)


[Create an Account](#)



localhost/recipe%20-%20mdb/meal.php?id=53080

Gmail YouTube Maps myclassroom All Bookmarks

Blini Pancakes



Instructions

In a large bowl, whisk together 1/2 cup buckwheat flour, 2/3 cup all-purpose flour, 1/2 teaspoon salt, and 1 teaspoon yeast.

Make a well in the center and pour in 1 cup warm milk, whisking until the batter is smooth.

Cover the bowl and let the batter rise until doubled, about 1 hour.

Enrich and Rest the Batter

Stir 2 tablespoons melted butter and 1 egg yolk into the batter.

In a separate bowl, whisk 1 egg white until stiff, but not dry.

Fold the whisked egg white into the batter.

Cover the bowl and let the batter stand 20 minutes.

Pan-Fry the Blini

Heat butter in a large nonstick skillet over medium heat.

Drop quarter-sized dollops of batter into the pan, being careful not to overcrowd the pan. Cook for about 1 minute or until bubbles form.

Turn and cook for about 30 additional seconds.

localhost/recipe%20-%20mdb/add_recipe.php

Gmail YouTube Maps myclassroom All Bookmarks

MealDB Home [Add Recipe](#) [View Recipe](#) [Log Out](#)

Add Your Recipe

Recipe Name

Upload Picture

 No file chosen

BIBLIOGRAPHY

1. Books and Articles on Web Development:

- Welling, Luke, and Laura Thomson. *PHP and MySQL Web Development*. 4th ed. Boston: Addison-Wesley, 2016.
- Duckett, Jon. *HTML and CSS: Design and Build Websites*. Chichester: Wiley, 2011.
- Freeman, Eric, and Bert Bates. *Head First Java*. 2nd ed. O'Reilly Media, 2009.

2. PHP and cURL:

- Gosselin, Paul. *PHP in Action: Objects, Design, Agility*. 2nd ed. Greenwich: Manning Publications, 2014.
- W3Schools. "PHP cURL." W3Schools - PHP cURL (Accessed October 2024).

3. MongoDB:

- Chodorow, Kristina. *MongoDB: The Definitive Guide*. 2nd ed. Sebastopol: O'Reilly Media, 2013.
- MongoDB Documentation. "MongoDB Manual." [MongoDB Documentation](#) (Accessed October 2024).

4. APIs and RESTful Services:

- Fielding, Roy Thomas. "Architectural Styles and the Design of Network-based Software Architectures." PhD Dissertation, University of California, Irvine, 2000. [Link](#)
- Kittlaus, M., & Fröhlich, M. (2018). "Building APIs with Node.js." *Apress*.

5. TheMealDB API:

TheMealDB. "TheMealDB API Documentation." TheMealDB API (Accessed October 2024).

6. Web Design:

- Marcotte, Ethan. *Responsive Web Design*. A Book Apart, 2011.
- Keith, Jeremy. *HTML5 for Web Designers*. A Book Apart, 2010.

7. User Experience and Interface Design:

- Norman, Don. *The Design of Everyday Things: Basic Principles of Human-Centered Design*. New York: Basic Books, 2013.
- Garrett, Jesse James. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. 2nd ed. New Riders, 2010.

8. Cooking and Culinary Arts:

- Ruhlman, Michael. *The Making of a Chef: Mastering Heat at the Culinary Institute of America*. New York: Penguin Books, 1997.
- Child, Julia, et al. *Mastering the Art of French Cooking*. 2nd ed. New York: Knopf, 2001.

Online Resources:

- PHP: [PHP Official Documentation](#) (Accessed October 2024).
- Bootstrap: Bootstrap Documentation (Accessed October 2024).
- JavaScript: [MDN Web Docs on JavaScript](#) (Accessed October 2024).

