Project Report

On

Recipe Haven

Course - Web Development with Open Source Frameworks

ETVAWD009

Submitted by:

Student Name (Roll No.)

Shikha: 2401010063

Saieena Negi: 2401010081

Ananya Sharma: 2401010056

Diksha Kumari:2401010055

Department of

Computer Science and Engineering

School of Engineering and Technology

K. R. Mangalam University, Gurugram- 122003

Index

Abstract

Recipe Haven is a web-based recipe sharing platform designed to provide users with an interactive and engaging experience for creating, sharing, and discovering recipes. The platform allows users to upload their own recipes with detailed ingredients and cooking steps, browse through recipes shared by others, and explore a variety of cuisines through a user-friendly interface. Built using HTML, CSS, and JavaScript for the front-end, and Node.js with Express.js for the back-end, the application ensures seamless interaction and efficient data management. The integration of SQL enables secure storage and retrieval of user and recipe information, while GitHub is utilized for version control and collaborative development. Recipe Haven aims to build a community-driven environment that promotes creativity, collaboration, and ease of access to culinary knowledge.

1. Introduction

Cooking and sharing recipes have always been an integral part of cultural exchange and creativity. In today's digital era, people seek convenient platforms to explore new dishes, share their culinary ideas, and connect with others who share the same passion for cooking. Recipe Haven is developed with the goal of fulfilling this need by providing an online space where users can easily create, share, and discover recipes.

The platform allows users to contribute their own recipes by adding ingredients, preparation steps, and images, while also enabling others to browse and try these recipes. The website's interactive and user-friendly design ensures a smooth experience for both beginners and experienced users.

Technically, Recipe Haven is built using HTML, CSS, and JavaScript for front-end design and interactivity, while Node.js and Express.js power the server-side functionalities. Data management is handled efficiently using SQL, ensuring secure storage and quick retrieval of recipes and user information. The use of GitHub for version control facilitated organized and collaborative development.

Overall, Recipe Haven serves as a dynamic and accessible platform that encourages users to learn, share, and connect through food, fostering a vibrant online cooking community.

1.1 Background

With the growing popularity of online platforms, people now prefer digital spaces to explore and share cooking ideas. Many existing recipe websites lack simplicity and user engagement, making it difficult for users to easily share or find recipes.

Recipe Haven was developed to overcome these challenges by offering an interactive and easy-to-use platform for recipe sharing. Built using HTML, CSS, JavaScript, Node.js, Express.js, and SQL, it provides users with a seamless experience to create, share, and discover recipes within a vibrant online community.

1.2 Project Objectives

The main objective of Recipe Haven is to create an interactive and user-friendly web platform where users can seamlessly create, share, and discover recipes from different cuisines. The project aims to promote a collaborative online community that encourages the exchange of culinary ideas and experiences.

The specific objectives of this project are:

- To design a simple and intuitive interface that allows users to easily navigate and interact with the platform.
- To enable users to add, view, edit, and manage their own recipes.
- To provide a feature for users to browse and discover recipes shared by others based on categories or ingredients.
- To ensure secure and efficient data management using SQL for storing recipe and user information.
- To implement Node.js and Express.js for handling backend operations and server-side logic.
- To utilize GitHub for version control and collaborative development.

Through these objectives, Recipe Haven aims to create a digital community that simplifies recipe sharing while inspiring creativity and connection among food enthusiasts.

1.3 Scope

The scope of Recipe Haven includes the design and development of a web-based platform that allows users to create, share, and explore recipes with ease. The project focuses on providing a smooth user experience through a clean interface and efficient data management.

The system covers:

- User interaction through adding, viewing, and managing recipes.
- Categorization of recipes for easier search and discovery.
- Secure storage of recipe and user details using SQL.
- Backend management and data handling using Node.js and Express.js.
- Front-end development with HTML, CSS, and JavaScript for an interactive design.

The project currently focuses on basic recipe sharing features but can be extended in the future to include user authentication, ratings, comments, and personalized recommendations to enhance user engagement.

2.Technology Overview

Recipe Haven is developed using a combination of front-end and back-end web technologies to ensure an interactive, efficient, and scalable platform. The technologies used are:

1. Front-End Technologies

- **HTML (Hyper Text Markup Language):**
  Used for structuring the web pages, defining elements such as forms, recipe lists, and navigation menus. It serves as the backbone of the website's layout.
- **CSS (Cascading Style Sheets):**
  Responsible for the visual design and appearance of the platform. It enhances the user interface by providing consistent styling, colors, and responsive layouts across different devices.
- **JavaScript:**
  Adds interactivity and dynamic behavior to the website. It enables features such as form validation, interactive recipe browsing, and real-time user interaction without reloading the page.

2. Back-End Technologies

- **Node.js:**
  Acts as the server-side runtime environment that allows JavaScript to run outside the browser. It handles user requests, manages data communication between the front-end and the database, and ensures fast processing.
- **Express.js:**
  A lightweight and flexible web framework built on top of Node.js. It simplifies the creation of APIs and routes, manages HTTP requests, and handles server-side logic effectively.

3. Database

- **SQL (Structured Query Language):**
  Used to store and manage application data such as user details, recipe information, and categories. SQL ensures data consistency, integrity, and secure access during operations like insertion, updating, and retrieval.

4. Version Control and Deployment

- **GitHub:**
  Utilized for version control and collaborative development. It helps in tracking code changes, managing multiple versions of the project, and maintaining backups.

Together, these technologies form a robust full-stack architecture that ensures Recipe Haven is fast, reliable, and easy to maintain.

3. Project Implementation

The implementation of Recipe Haven involves the integration of front-end, back-end, and database components to create a fully functional recipe sharing platform. Each phase of implementation focuses on achieving specific objectives, ensuring smooth functionality and a user-friendly experience.

1. Front-End Implementation

• The front-end of the application was developed using HTML, CSS, and JavaScript.

• HTML was used to design the structure of web pages such as the home page, recipe creation page, and recipe view page.

• CSS provided styling and layout design to make the platform visually appealing and responsive across different devices.

• JavaScript added interactivity to features like form validation, dynamic recipe display, and navigation menus.
The focus was on creating an intuitive and attractive interface that encourages users to easily explore and share recipes.

2. Back-End Implementation

• The server-side of the project was implemented using Node.js and Express.js.

• Node.js handled the execution of JavaScript code on the server and managed client requests efficiently.

• Express.js was used to define routes, handle HTTP requests, and connect the application with the database.
The back-end ensures smooth communication between the

user interface and the database, enabling secure storage and retrieval of recipes.

3. Database Implementation

• The database was created using SQL, which stores user information, recipe details, ingredients, and categories.

• Tables were designed with proper relationships and constraints to maintain data integrity.

• CRUD (Create, Read, Update, Delete) operations were implemented to manage recipe and user data.
This ensures that all data is securely stored and efficiently managed.

4. Version Control and Testing

• GitHub was used for version control to track changes and manage code updates. The system was tested at each stage to identify and fix bugs.

• Unit testing was performed for individual modules.

• Integration testing ensured that all components worked together properly.

• User testing was conducted to verify ease of navigation and usability.

5. Deployment

After successful testing, the application was prepared for deployment on a local or cloud-based server. The deployment phase ensured that the website runs smoothly and is accessible to users.

4. Key Features Developed

• **User-Friendly Interface**: Simple and responsive design built with HTML, CSS, and JavaScript for smooth navigation.

• **Recipe Management**: Users can add, edit, and delete their own recipes with details like ingredients and steps.

• **Recipe Discovery**: Allows users to browse and search recipes shared by others easily.

• **Database Integration**: All data is securely stored and managed using SQL with full CRUD functionality.

• **Backend Functionality**: Node.js and Express.js handle server-side logic and data communication efficiently.

• **Version Control**: GitHub is used for code management and collaboration.

• **Responsive Design**: Works smoothly across different devices and screen sizes.

5. Screenshots

Fig. 1: Home Page

This page serves as the main landing screen of *Recipe Haven*, welcoming users and providing easy navigation options such as Home, About, Services, and Testimonials.

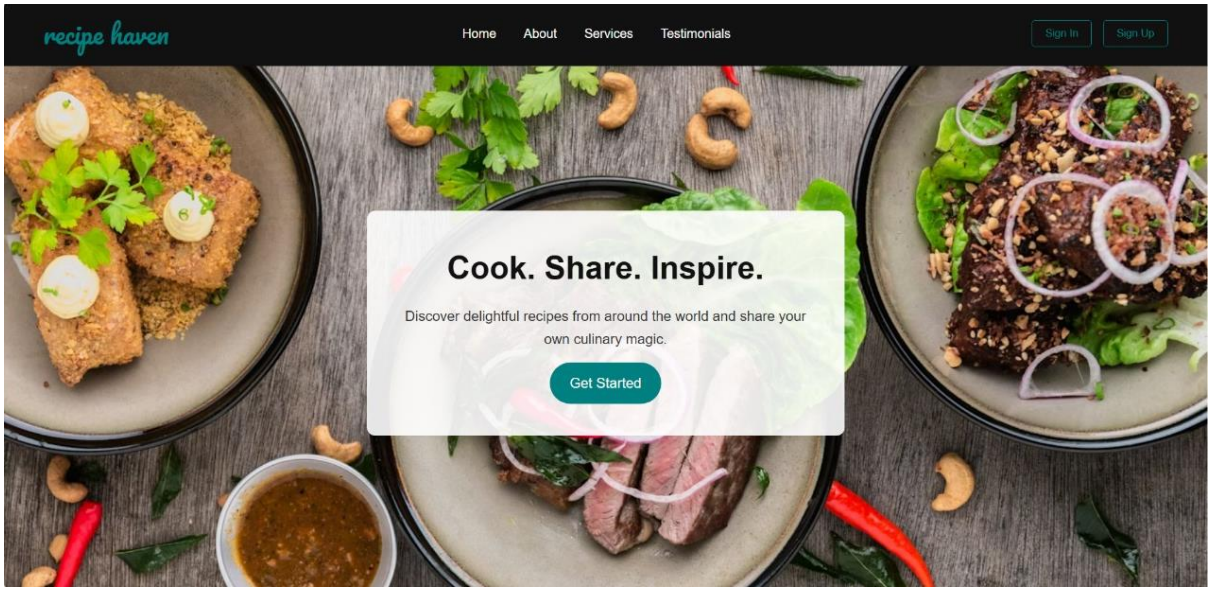


Fig. 2: About Page

This section explains the purpose of *Recipe Haven*, allowing users to understand how they can share, explore, and connect through recipes.

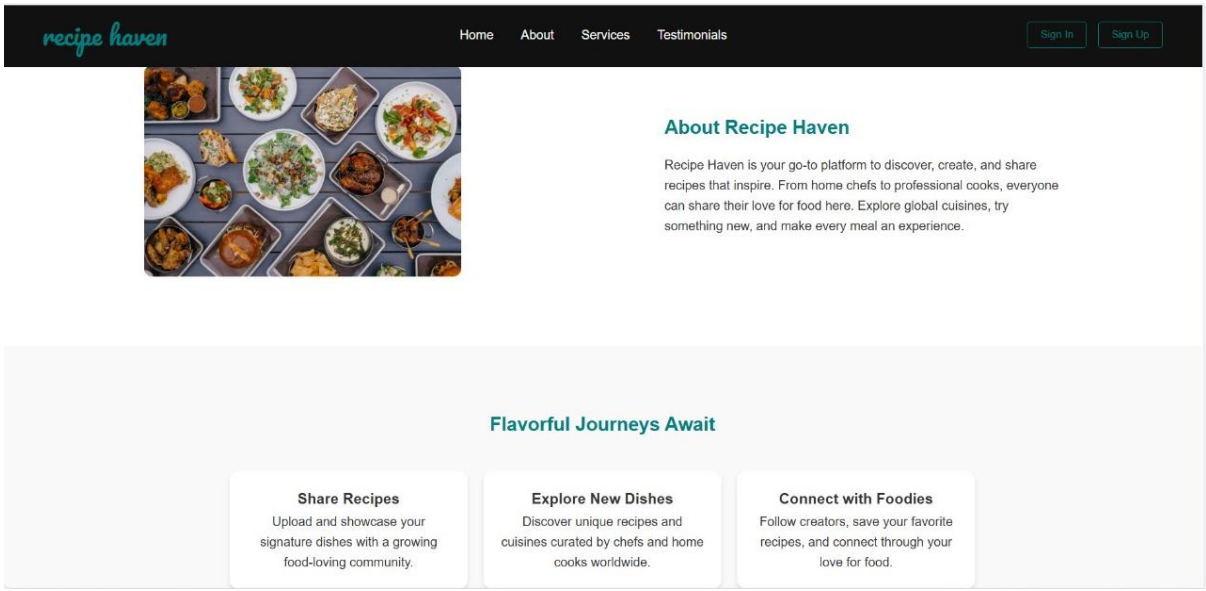
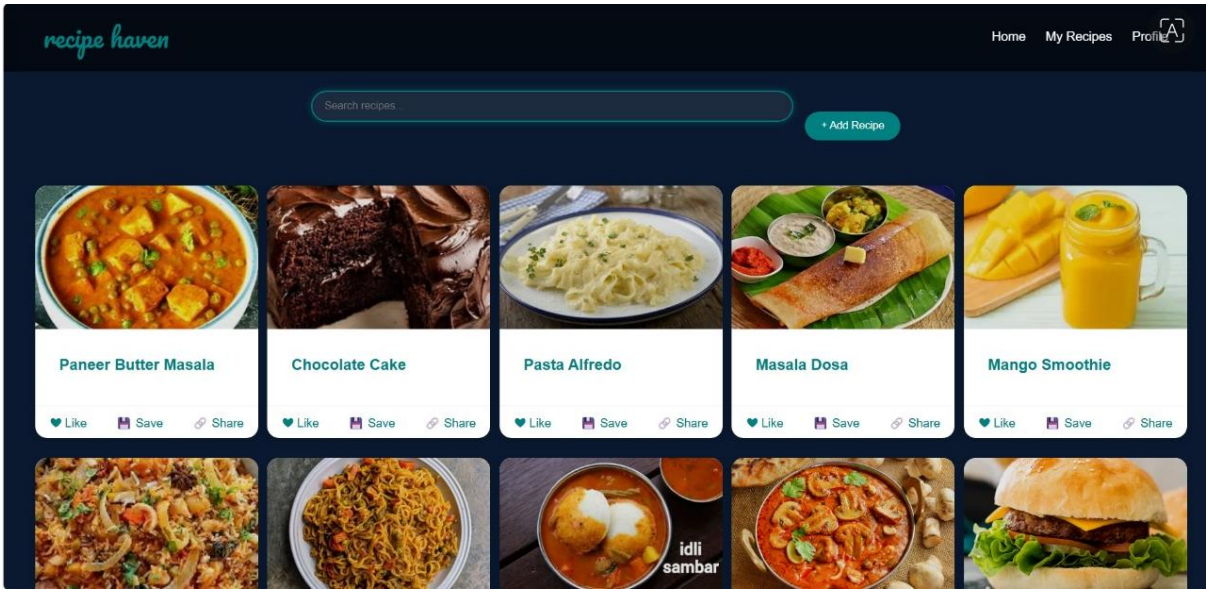


Fig. 3: Recipe Dashboard

This page displays the collection of recipes shared by users. It includes options to search, like, and add new recipes, creating an interactive and engaging experience.



6. Conclusion And Future Work

Conclusion

The Recipe Haven project successfully achieved its goal of developing a web-based platform that allows users to create, share, and explore recipes in an interactive and user-friendly manner. By integrating **HTML, CSS, JavaScript, Node.js, Express.js, and SQL,** the system provides a smooth experience with efficient data handling and secure storage.

The project demonstrates the effective use of full-stack web development technologies to build a functional and engaging application. It also promotes community interaction by enabling users to exchange culinary ideas easily. Overall, Recipe Haven serves as a practical and scalable solution for recipe sharing in the digital era.

Future Work

Although the project meets its current objectives, several enhancements can be added in future versions to improve functionality and user engagement:

- Implement **user authentication and login system** for personalized experiences.
- Add **recipe ratings, comments, and likes** to increase user interaction.
- Introduce **image uploads for recipes** to make them more attractive.
- Include **search filters and recommendation features** based on ingredients or popularity.
- Enable **API integration** for importing or exporting recipes from other platforms.

7. References

1. **W3Schools - HTML, CSS, and JavaScript Tutorials**

2. **MDN Web Docs - Node.js and Express.js Documentation**

3. **GeeksforGeeks - Web Development and SQL Tutorials**

4. **TutorialsPoint - Full Stack Development Guide**

5. **GitHub Documentation - Version Control and Repository Management**