Practical Activity Report

UI & UX SPECIALIST-(UCS542)

END-Semester Lab Evaluation

Submitted to

Asst. Prof. Kanupriya

BE Third Year

Submitted by

Gaganbir Singh (102317046)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

TIET, Patiala

August-December 2025

INDEX

INTRODUCTION

In the contemporary digital era, the intersection of culinary arts and technology has opened new avenues for home cooking. However, despite the abundance of online resources, many home cooks struggle with a common, daily dilemma: "What can I cook with the ingredients I currently have?" This disconnect between available pantry stock and recipe discovery often leads to household food waste, unnecessary grocery expenditure, and decision fatigue.

FlavorForge is a robust, web-based application designed to address these challenges. It serves as an intelligent culinary assistant that empowers users to discover recipes tailored specifically to their inventory. By leveraging modern web technologies, FlavorForge provides a seamless, intuitive, and visually engaging platform where users can explore thousands of recipes, find ingredient substitutes, and manage their culinary preferences.

Scope and Utility

The scope of FlavorForge extends beyond simple recipe retrieval. It functions as a comprehensive kitchen companion.

- **For the novice cook:** It provides step-by-step instructions and "kitchen savior" tools like ingredient substitution.
- **For the eco-conscious user:** It helps reduce food waste by prioritizing recipes that use up existing ingredients.
- **For the health-conscious:** It offers detailed nutritional breakdowns, including caloric and macronutrient data.

By integrating with the Spoonacular API, FlavorForge ensures that its data is always current, diverse, and reliable. The project encapsulates the full software development lifecycle, demonstrating how vanilla web technologies (HTML, CSS, JavaScript) can be orchestrated to build a production-grade application without the overhead of complex frameworks.

PROBLEM STATEMENT

The Core Issues

While the internet is flooded with recipe websites, the user experience is often suboptimal. Through our research and analysis of the current market, we identified four critical problems that FlavorForge aims to solve:

1. The "Pantry Paradox" and Food Waste One of the most significant issues facing modern households is food waste. A substantial amount of produce and pantry staples are discarded simply because consumers lack the immediate knowledge of how to utilize them. Traditional recipe sites work backwards: they ask, "What do you want to eat?" requiring the user to then go buy specific ingredients. There is a distinct lack of tools that ask, "What do you *have*?" and provide solutions based on those constraints.

2. Decision Fatigue The paradox of choice is prevalent in meal planning. When faced with thousands of uncurated options, users often feel overwhelmed and revert to unhealthy takeout or repetitive meals. Users need a system that filters and curates options based on specific, immediate inputs—making the decision process faster and less stressful.

3. Dietary and Ingredient inflexibility Many users face dietary restrictions or suddenly realize they are missing a key ingredient mid-cooking. Standard recipe blogs rarely offer dynamic solutions for these moments. A static recipe page cannot tell you that "yogurt" is a viable substitute for "sour cream" in a specific context. This lack of flexibility often leads to abandoned cooking attempts.

4. Poor User Experience (UX) A technical analysis of competitor sites reveals a prevalence of poor UX patterns: slow loading times due to heavy ad scripts, non-responsive layouts that break on mobile devices, and cluttered interfaces that bury the actual recipe instructions. Users need a "Clean Web" approach—a fast, dark-mode-enabled, and focused interface that respects their attention.

The Proposed Solution

FlavorForge addresses these problems directly. By reversing the search logic to focus on *ingredients first*, it tackles food waste. By providing a clean, ad-free interface with instant search capabilities, it eliminates decision fatigue. The inclusion of a dedicated substitution tool and comprehensive nutritional data ensures that the platform adapts to the user's needs, not the other way around.

SPECIFIC REQUIREMENTS

A. Functional Requirements

1. Ingredient-Based Search Engine The core feature of the system is the "Pantry Search." Users must be able to input a comma-separated list of ingredients (e.g., "chicken, rice, broccoli"). The system must process this string, sanitize the input, and query the API to return recipes that strictly or loosely match these items. The results must visually indicate which user-ingredients are used and which are missing from the recipe.

2. Natural Language Recipe Discovery Users should have the ability to search for recipes using natural language queries, such as "Gluten-free Pasta" or "High protein breakfast." The system must interpret these queries and apply appropriate filters for cuisine type, diet (Vegan, Keto, etc.), and intolerances via the API parameters.

3. Dynamic Recipe Detailing Upon selecting a recipe, the application must fetch and display a comprehensive detail view. This includes:

- **Visuals:** High-resolution hero image of the dish.
- **Metadata:** Preparation time in minutes and number of servings.
- **Instructions:** A clear, step-by-step ordered list of cooking directions.
- **Ingredients:** An interactive checklist of required items.
- **Nutrition:** A data visualization or summary of calories, protein, fats, and carbohydrates

.

4. "Kitchen Savior" Substitution Tool A dedicated module must be available for ingredient substitutions. When a user inputs an ingredient name, the system must asynchronously fetch valid substitutes and display them in a clear, card-based layout.

5. Local Persistence (Favorites & History) The system must allow users to "Favorite" recipes. This data must be stored in the browser's `localStorage` to ensure it persists even if the browser is closed or the page is refreshed, without requiring a server-side user account.

b) Non-Functional Requirements

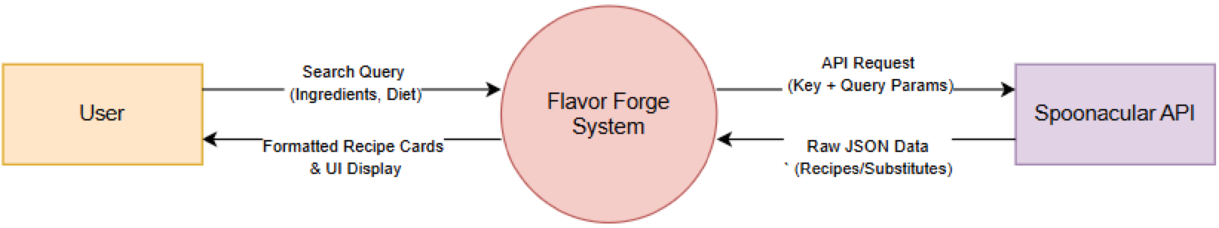
1. Performance and Caching The system must load under 2 seconds on 4G networks. To achieve this, it must implement client-side caching. If a user repeats a search within a specific timeframe (e.g., 1 hour), the data should be retrieved from local storage rather than making a redundant API call, saving bandwidth and API quota.

2. Accessibility (a11y) The UI must adhere to WCAG 2.1 AA standards. This includes high-contrast color ratios (especially for text), semantic HTML tags (header, nav, main, article), and proper ARIA labels for screen readers. The site must be fully navigable via keyboard.

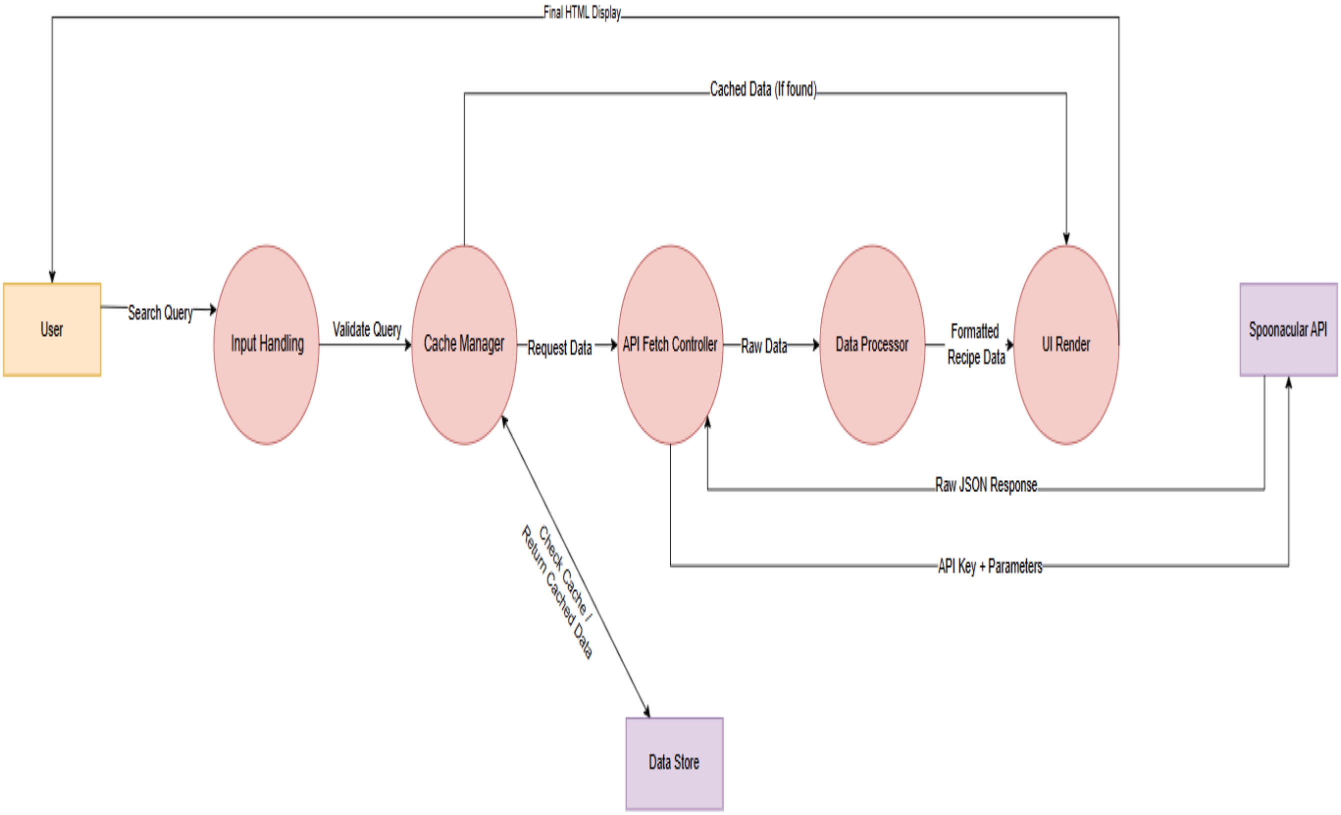
3. Responsive Design The interface must be "Mobile-First." It should render correctly on devices ranging from small mobile screens (320px width) to large desktop monitors (1920px+). The layout should utilize CSS Grid and Flexbox to adapt content dynamically

CONTEXT LEVEL AND DATA FLOW DIAGRAM

Context Level Diagram -(Level 0)



Data Flow Diagram -(Level 1)



SYSTEM SPECIFICATION

Hardware Specifications

FlavorForge is designed as a lightweight, client-side application. It does not require powerful hardware and is optimized to run efficiently on low-end devices.

Minimum Requirements (To Run the App):

- **Processor:** Any basic processor (Intel Celeron, Pentium, or mobile equivalent).
- **RAM:** 512 MB or higher.
- **Storage:** 5 MB of free space (for caching images and favorites).
- **Network:** Basic 3G or broadband connection (for fetching API data).
-

Recommended Requirements (For Best Experience):

- **Processor:** Intel Core i3 / AMD Ryzen 3 or better.
- **RAM:** 4 GB (standard for modern web browsing).
- **Display:** 1280x720 (HD) resolution or higher.

Software Specifications

The application is platform-agnostic and relies only on a standard web browser environment.

- **Operating System:** Any OS capable of running a modern web browser (Windows 7/10/11, macOS, Linux, Android 6.0+, iOS 12+).
- **Web Browser:**
  - Google Chrome (v60+)
  - Mozilla Firefox (v60+)
  - Safari (v12+)
  - Microsoft Edge (Chromium)
- **Dependencies:** None. The user does not need to install Node.js, Python, or any runtime environment to use the application. It is a static website.

TOOLS USED

1. HyperText Markup Language (HTML5)

HTML5 serves as the backbone of the application. We utilized semantic HTML5 tags (such as `<header>`, `<main>`, `<section>`, `<article>`, and `<footer>`) rather than generic `<div>` tags. This approach significantly improves the Accessibility (a11y) of the site for screen readers and enhances Search Engine Optimization (SEO). It defines the structure of the recipe cards, the search forms, and the layout grids.

2. Cascading Style Sheets (CSS3)

CSS3 was used for all visual styling. Crucially, we avoided heavy frameworks like Bootstrap to maintain complete control over the design and keep the file size low.

- **CSS Variables (Custom Properties):** Used to define the color palette (Terracotta, Cream, Saffron) globally. This made implementing the "Dark Mode" feature a simple matter of swapping variable values.
- **Flexbox & Grid:** Used for the responsive layouts. CSS Grid handles the two-dimensional recipe results, while Flexbox manages the navigation bar and alignment within cards.

3. Vanilla JavaScript (ES6+)

The logic of the application is written in pure, modern JavaScript (ECMAScript 2015+).

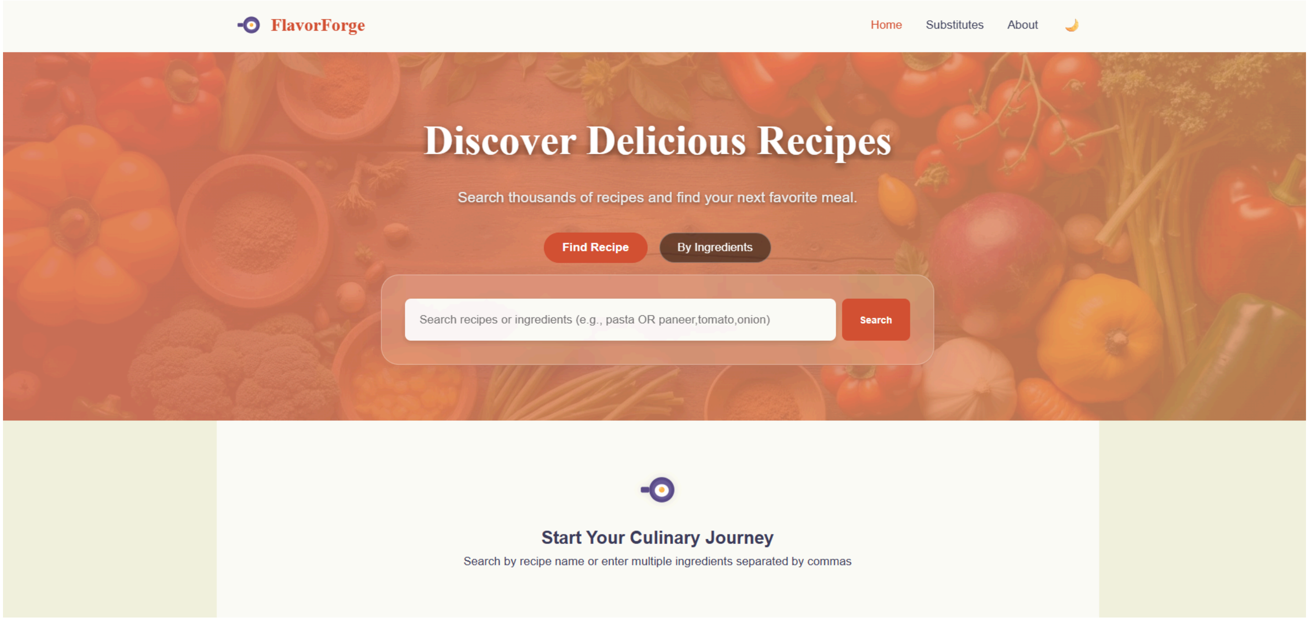
- **Fetch API:** Used to handle asynchronous network requests to the Spoonacular API.
- **Async/Await:** Implemented to write cleaner, more readable code for handling promises and API responses.
- **DOM Manipulation:** Used to dynamically generate HTML content based on API data without needing a page reload (Single Page Application feel).
- **Modules:** The code is split into modular files (`main.js`, `search.js`, `recipe.js`) to ensure maintainability and separation of concerns.

4. Spoonacular API

Spoonacular is the data engine of the project. It is a powerful food intelligence API that allows for complex queries. We utilized endpoints such as `findByIngredients`, `complexSearch`, and `getRecipeInformation`. It was chosen for its high accuracy in parsing natural language and its extensive database of over 360,000 recipes.

SAMPLE SCREENSHOTS

1. Home page



Discover Delicious Recipes

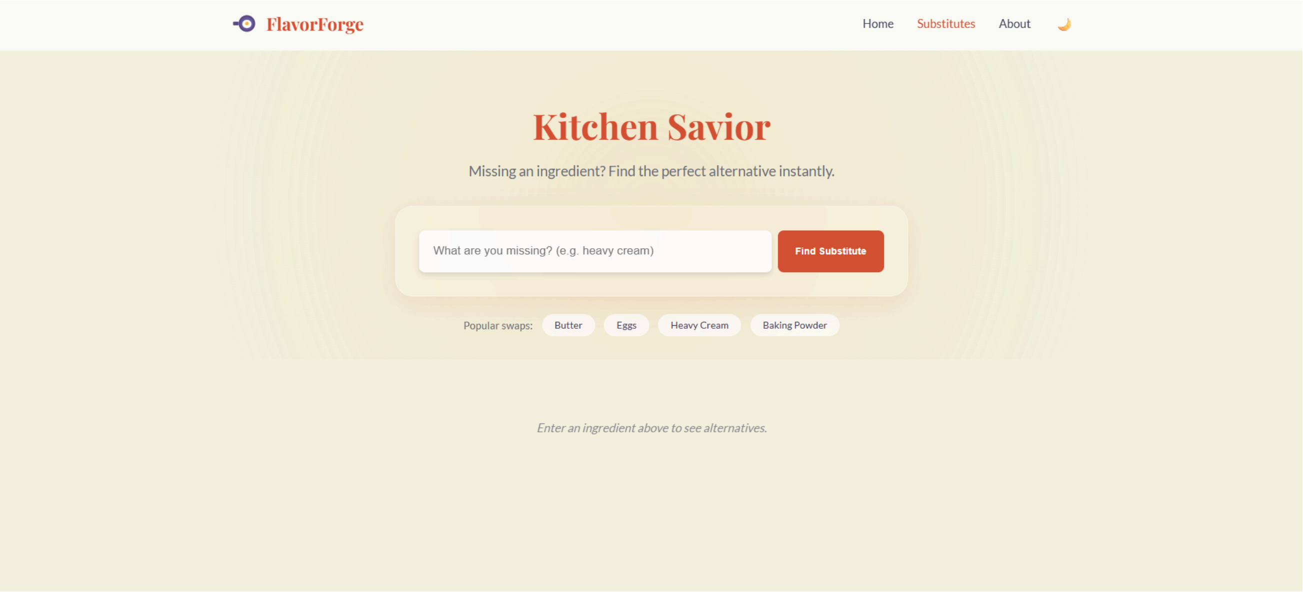
Search thousands of recipes and find your next favorite meal.

Search recipes or ingredients (e.g., pasta OR paneer,tomato,onion)

Start Your Culinary Journey

Search by recipe name or enter multiple ingredients separated by commas

2. Substitute Page



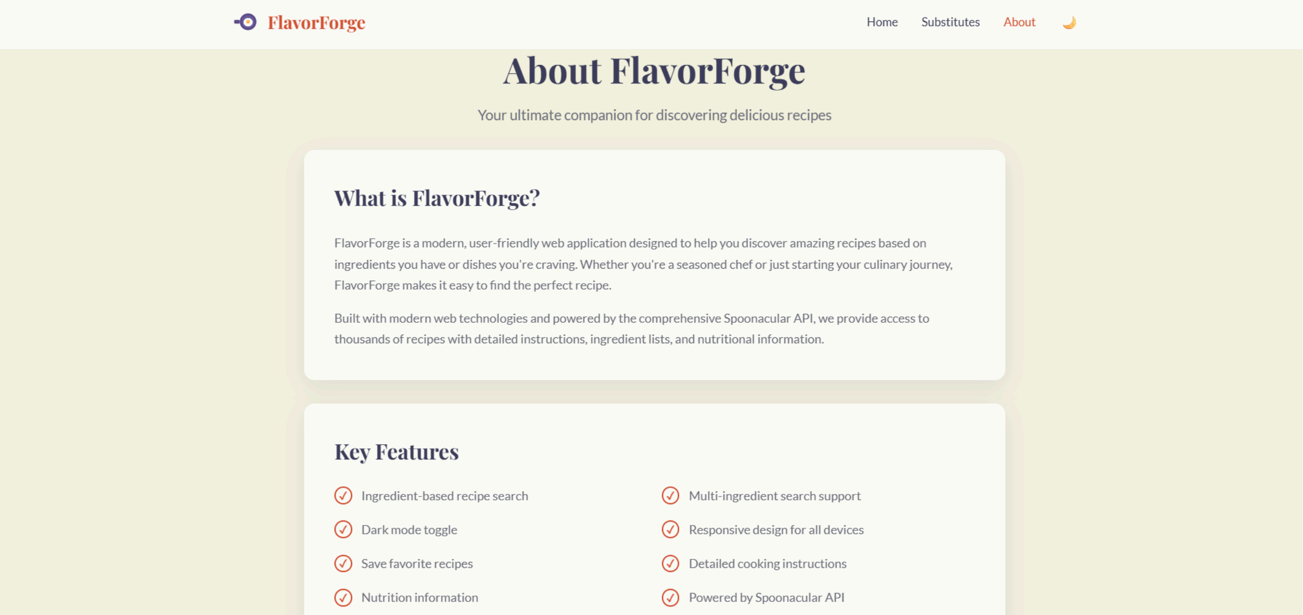
Kitchen Savior

Missing an ingredient? Find the perfect alternative instantly.

What are you missing? (e.g. heavy cream)

Enter an ingredient above to see alternatives.

3. About page



About FlavorForge

Your ultimate companion for discovering delicious recipes

What is FlavorForge?

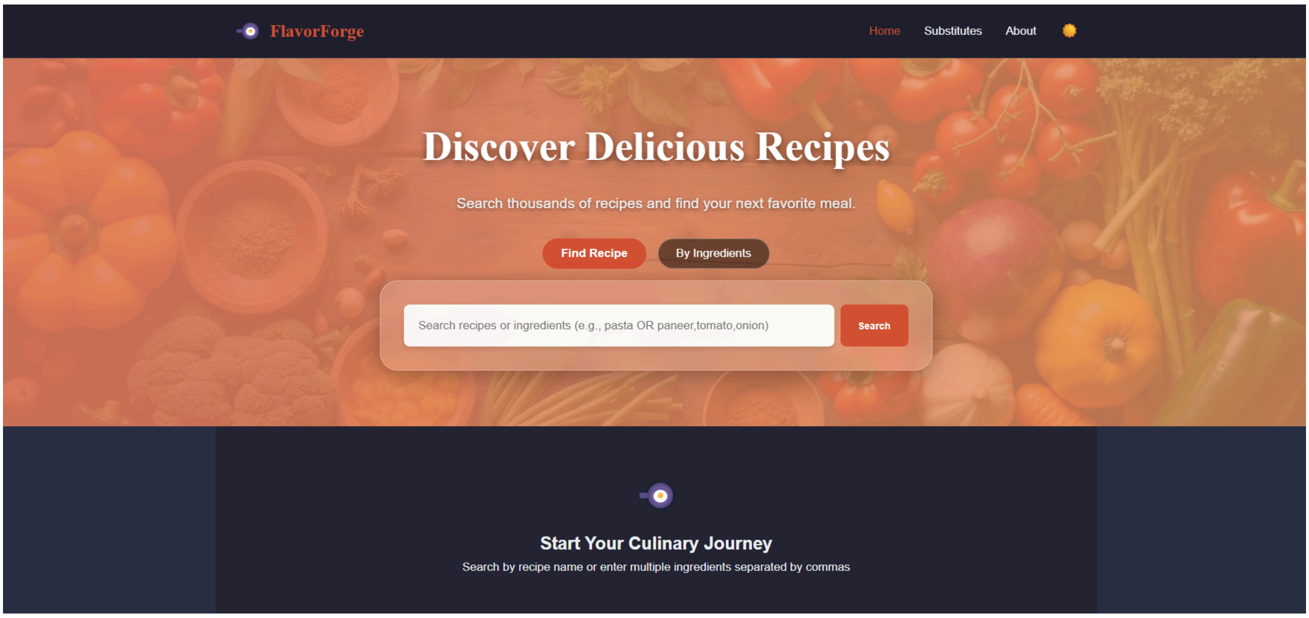
FlavorForge is a modern, user-friendly web application designed to help you discover amazing recipes based on ingredients you have or dishes you're craving. Whether you're a seasoned chef or just starting your culinary journey, FlavorForge makes it easy to find the perfect recipe.

Built with modern web technologies and powered by the comprehensive Spoonacular API, we provide access to thousands of recipes with detailed instructions, ingredient lists, and nutritional information.

Key Features

- ✓ Ingredient-based recipe search
- ✓ Dark mode toggle
- ✓ Save favorite recipes
- ✓ Nutrition information
- ✓ Multi-ingredient search support
- ✓ Responsive design for all devices
- ✓ Detailed cooking instructions
- ✓ Powered by Spoonacular API

4. Dark Mode


FlavorForge
Home
Substitutes
About
Find Recipe
By Ingredients
Search recipes or ingredients (e.g., pasta OR paneer,tomato,onion)
Search

Discover Delicious Recipes

Search thousands of recipes and find your next favorite meal.

Start Your Culinary Journey

Search by recipe name or enter multiple ingredients separated by commas

OUTPUT REPORTS

Testing Methodology

To ensure FlavorForge is production-ready, a rigorous testing strategy was employed, focusing on functional correctness and user experience.

1. API Response Verification We conducted manual testing using Postman and the browser console to verify the integrity of the data returned by Spoonacular.

- **Test Case 1 (Ingredient Search):** Input "Apples, Flour, Sugar". *Result:* System correctly returned apple pie and crumble recipes.
- **Test Case 2 (Error Handling):** Input random gibberish string. *Result:* System displayed a friendly "No results found" message instead of a code error.
- **Test Case 3 (Quota Limit):** Simulated API limit reach. *Result:* System handled the 402 error code and displayed "Daily Limit Reached" advice.

2. Cross-Browser and Responsive Testing The application was tested across Chrome, Firefox, and Microsoft Edge.

- **Desktop (1920x1080):** Confirmed 3-column grid layout.
- **Tablet (768x1024):** Confirmed automatic shift to 2-column grid.
- **Mobile (375x667):** Confirmed single-column layout and hamburger menu functionality.

CONCLUSION

The development of **FlavorForge** stands as a testament to the capabilities of modern standard web technologies. By deliberately choosing a "Vanilla" tech stack over complex frameworks, we successfully created an application that is lightweight, incredibly fast, and easy to maintain, while still delivering a "world-class" user experience.

The project successfully solves the identified problems of food waste and decision fatigue. It transforms the vague anxiety of "what's for dinner?" into a structured, enjoyable discovery process. The integration of the Spoonacular API proved to be robust, providing high-quality data that makes the application genuinely useful for daily cooking. Visually, the warm, modern design language successfully establishes a brand identity that is both professional and inviting.

Future Scope

While the current version of FlavorForge is a complete, deployable product, the modular codebase allows for significant future expansion:

1. **AI-Powered Meal Planning:** Integrating OpenAI or similar AI models to generate weekly meal plans based on the user's past favorites and specific caloric goals
2. **User Authentication & Cloud Sync:** Currently, favorites are stored locally. Future updates could implement Firebase Authentication to allow users to log in and sync their favorites across their phone and laptop.
3. **"Pantry Tracker":** A feature allowing users to digitally "stock" their pantry in the app, with the system automatically suggesting recipes before items expire.
4. **Social Sharing:** Adding functionality to share discovered recipes directly to social media platforms or generate a shopping list to share via WhatsApp.
5. **Voice Navigation:** Implementing the Web Speech API to allow users to navigate recipes via voice commands (e.g., "Next step") while their hands are messy from cooking