Project Report for the Subject

TEST AUTOMATION
(UCS662)

Submitted By:

RAVNEET KAUR	(102203202)
HARDETYA GILL	(102203213)
SARA AGNIHOTRI	(102203248)

BE COE- 3C21

FORK & FEAST
Online Recipe Book & Food Blog

Course Instructor:
Dr. JAVED IMRAN

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)
COMPUTER SCIENCE DEPARTMENT,
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,
PATIALA

Session: Jan - May 2025
Date of Submission: 6 May 2025

CONTENTS

INTRODUCTION

In today's fast-paced digital era, the demand for intelligent web applications that cater to everyday user needs is growing rapidly. "Fork & Feast" is one such web application designed to revolutionize the way users interact with online recipe platforms. Unlike traditional recipe websites, Fork & Feast provides a user-centric interface that allows food enthusiasts to search for recipes based on ingredients they have at hand, making meal planning more efficient and personalized.

The project is designed to simplify the culinary decision-making process by offering ingredient-based search functionalities, user login features, and an aesthetically pleasing UI. It is particularly targeted at individuals who may not have the time or resources to plan elaborate meals but still wish to cook something delicious and easy.

This project not only focuses on building a responsive and dynamic web application but also integrates robust testing using automation tools like Selenium and TestNG. The automated testing ensures that the application behaves as expected under various conditions and provides a seamless user experience.

Fork & Feast is built with modern web technologies for the frontend and leverages MongoDB for backend data storage, ensuring scalability and efficiency. It serves as a comprehensive case study in integrating frontend development, backend services, and test automation, making it an ideal project for demonstrating full-stack engineering skills.

PROBLEM STATEMENT

The food industry has seen a tremendous surge in the number of online platforms offering recipes, nutritional advice, and meal planning services. However, most existing platforms assume that users already know what they want to cook or have a recipe in mind. This can be inconvenient for users who want to make meals based solely on available ingredients, avoid food waste, or discover new dishes without intensive searching.

The core problem lies in the lack of personalized ingredient-based search and recommendation systems. Users often find themselves inputting multiple queries, toggling between pages, or abandoning the search altogether due to the inefficiency of existing tools.
Additionally, manual testing of such a web application can be time-consuming and prone to human error. Without automation, maintaining quality across updates and changes is not scalable.

The Fork & Feast project addresses these pain points by:

- Offering a powerful search engine that filters recipes by ingredients.
- Providing a clean and interactive UI to engage users.
- Automating the testing process to ensure reliable functionality and consistent performance.

This project aims to fill the gap between user expectations and actual web functionalities, bringing convenience, speed, and innovation to the modern kitchen experience.

SPECIFIC REQUIREMENTS

3.1 FUNTIONAL REQUIREMENTS

Functional requirements define what the system should do. For the Fork & Feast application, these are described in terms of specific behaviors and tasks that the software must perform:

1. *User Authentication*
   - Users must be able to register and log in using an email and password.
   - Form validation must ensure data integrity before submission.

2. *Ingredient-Based Recipe Search*
   - Users can input one or multiple ingredients.
   - The system filters and displays only those recipes that include the selected ingredients.

3. *Responsive Navigation*
   - Users should be able to navigate between pages (Home, Login, Search, Results) seamlessly.
   - Buttons like "Explore Now" must redirect appropriately.

4. *Data Storage & Management*
   - MongoDB will be used to store user credentials and possibly recipes in future extensions.

5. *Interactive UI Elements*
   - Buttons, text inputs, and dynamic messages must be interactive and intuitive.

6. *Search Result Display*
   - Recipes should be displayed with images, ingredient lists, and cooking steps.
   - A fallback message should appear if no recipes are found.

7. *Logout Option*
   - Users must be able to log out securely, clearing any session data.

8. *Automation Coverage*
   - All major functionalities should be testable using Selenium and TestNG.
   - Tests must cover login, navigation, search inputs, and result validation.

3.2 NON- FUNTIONAL REQUIREMENTS

Non-functional requirements are not about specific behaviors but rather how the system performs. For Fork & Feast, these are crucial in determining the quality and usability of the application.

1. *Performance*
   - The application must load within 2 seconds on a standard broadband connection.
   - Searches should return results within 3 seconds after clicking the search button.

2. *Scalability*
   - The backend should be able to accommodate a growing user base and expanding recipe data.

3. *Usability*
   - The UI must be user-friendly, accessible, and responsive across all major devices (phones, tablets, desktops).

4. *Security*
   - User credentials must be securely stored.
   - Forms must be protected against common threats like SQL Injection and XSS.

5. *Maintainability*
   - The codebase must be modular and well-documented to support future enhancements.

6. *Reliability*
   - The system should function reliably with minimal downtime or failures.

7. *Compatibility*
   - Application must support the latest versions of Chrome, Firefox, and Edge.

8. *Test Automation*
   - Test cases should execute in parallel where applicable, reducing test cycle time.
   - Automation scripts must be modular and follow best practices.

SYSTEM ARCHITECTURE

4.1 CONTEXT LEVEL DIAGRAM

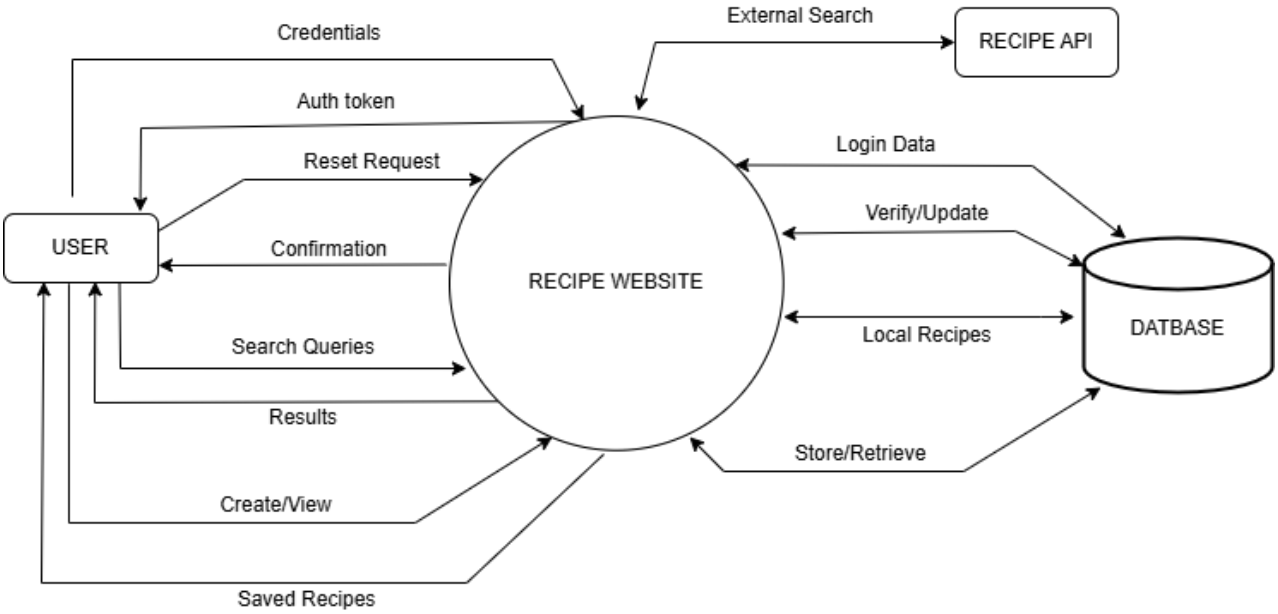


Fig 4.1- Context Level Diagram

4.2 DATA FLOW DIAGRAM

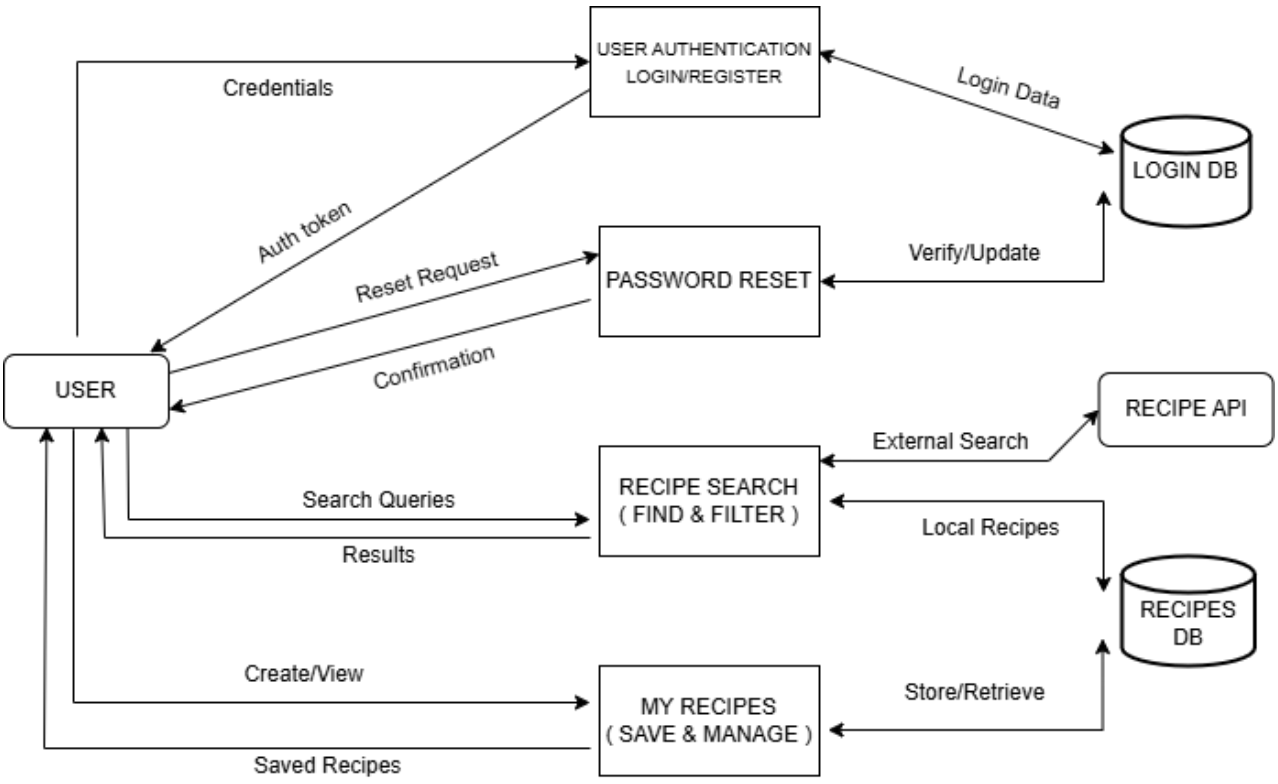


Fig 4.2- Data Flow Diagram

SYSTEM SPECIFICATIONS

5.1 HARDWARE SPECIFICATIONS

The Fork & Feast application, while being lightweight, still demands a reasonable development and test environment. Below are the hardware specifications ideal for running this project efficiently:

- Processor: Intel Core i5 or higher (Quad Core, 7th Gen or later)
- RAM: 8 GB minimum, 16 GB recommended for test automation
- Hard Drive: 500 GB HDD or 256 GB SSD
- Operating System: Windows 10 or Windows 11 (64-bit)
- Display: 1366x768 minimum resolution, preferably Full HD for development
- Graphics Card: Integrated GPU sufficient; dedicated GPU optional
- Internet Connection: Minimum 5 Mbps for basic server and client communication

These hardware components ensure that both the development and automated testing processes run smoothly without interruptions or latency issues.

5.2 SOFTWARE SPECIFICATIONS

Developing and testing Fork & Feast involves multiple software tools and technologies. The chosen stack ensures scalability, modern UI/UX, and robust test automation.

- Frontend:
  - HTML, CSS, JavaScript
- Backend:
  - MongoDB Shell
  - Node.js
- Testing & Automation:
  - Selenium WebDriver
  - TestNG
  - Maven
  - Eclipse IDE
  - ANT
- Browser:
  - Google Chrome (compatible ChromeDriver)

TOOLS USED

6.1 SELENIUM

Selenium is a powerful open-source tool used for automating web browsers. In the Fork & Feast, Selenium ensures the website performs correctly across various user interactions.

Key Roles Selenium Plays:

- Automates login functionality to verify user authentication flow.
- Simulates user behavior like clicking buttons, entering ingredients, and searching for recipes.
- Helps identify issues like broken links, missing elements, or slow responses.

Why Selenium?

- Supports multiple browsers (Chrome, Firefox, Edge).
- Works with various programming languages (Java, Python, etc.).
- Easy integration with TestNG for assertion and report generation.

Implementation in Fork & Feast:

- Used ChromeDriver to automate Chrome-based interactions.
- Scripts simulate user flows like Login → Index Page → Search Page → Results.
- Supports both single-thread and parallel execution using TestNG.

6.2 TestNG

TestNG (Test Next Generation) is a powerful testing framework inspired by JUnit and NUnit, designed to simplify testing in Java environments. It enhances the Selenium framework by adding capabilities such as test grouping, prioritization, parameterization, and report generation.

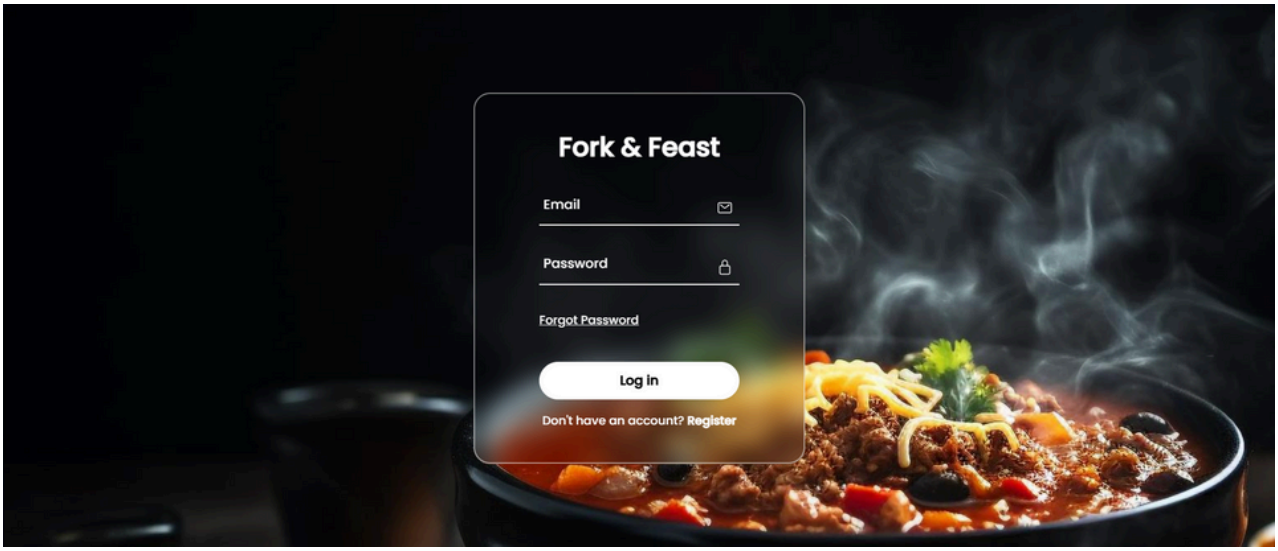
Why TestNG?

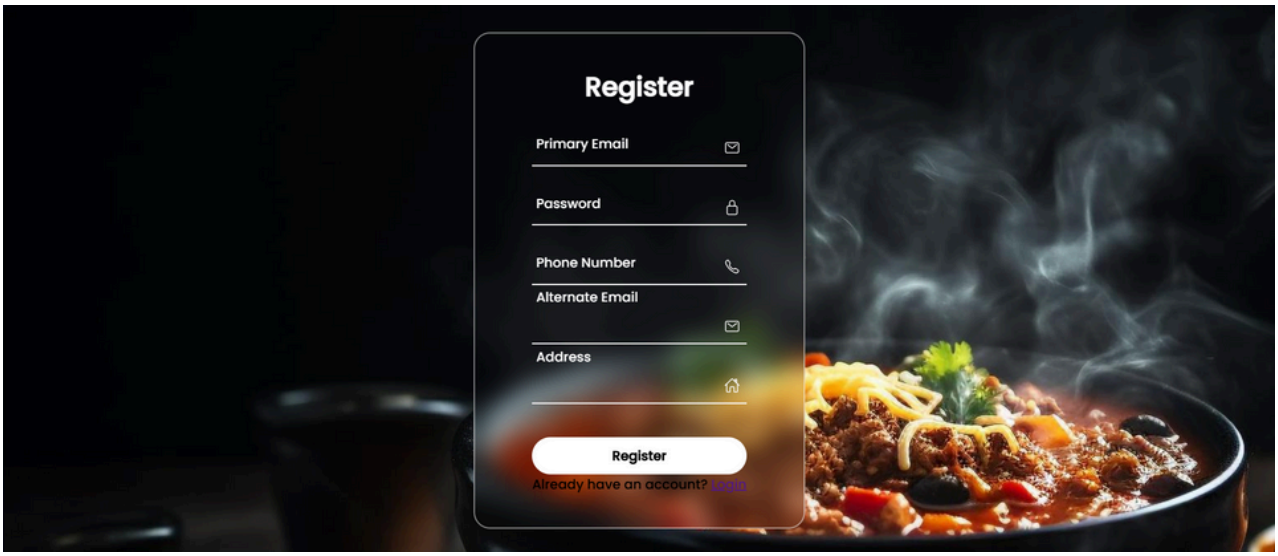
- Supports data-driven testing with @DataProvider.
- Offers detailed HTML and XML reports post-execution.
- Allows test cases to be grouped and executed in parallel for faster feedback.

Usage in Fork & Feast:

- Defined separate test methods for login, search, and navigation.
- Used assertions to verify search results.
- Generated detailed reports showing test case execution, status, and logs.

SAMPLE SCREENSHOTS





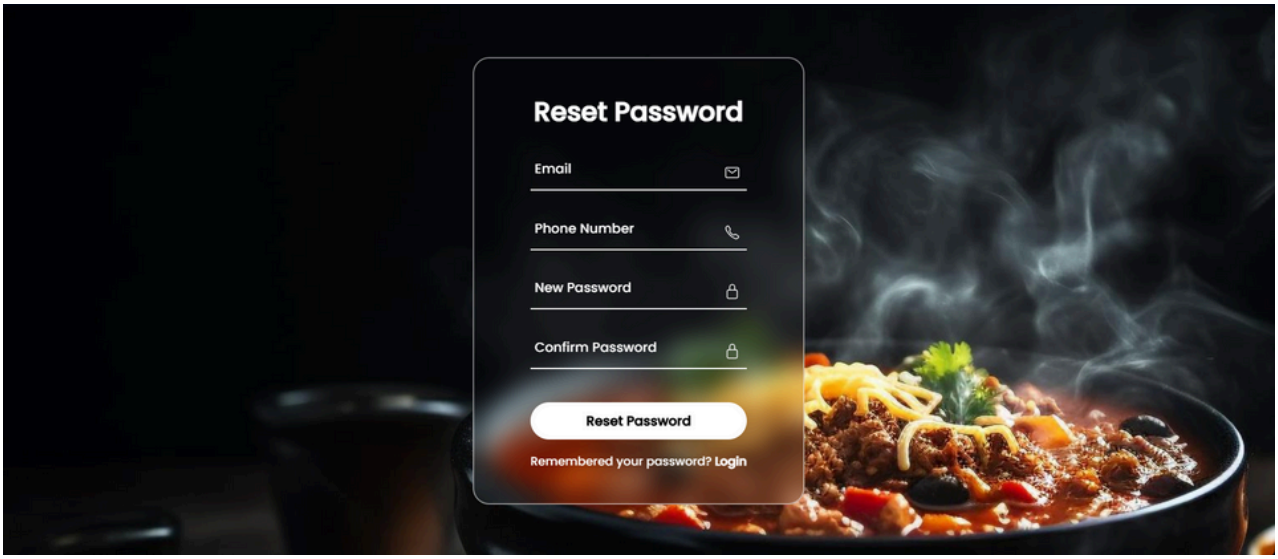


Fig 7.1, 7.2, 7.3- Login Page, Registration Page, Reset/Forgot Password Page

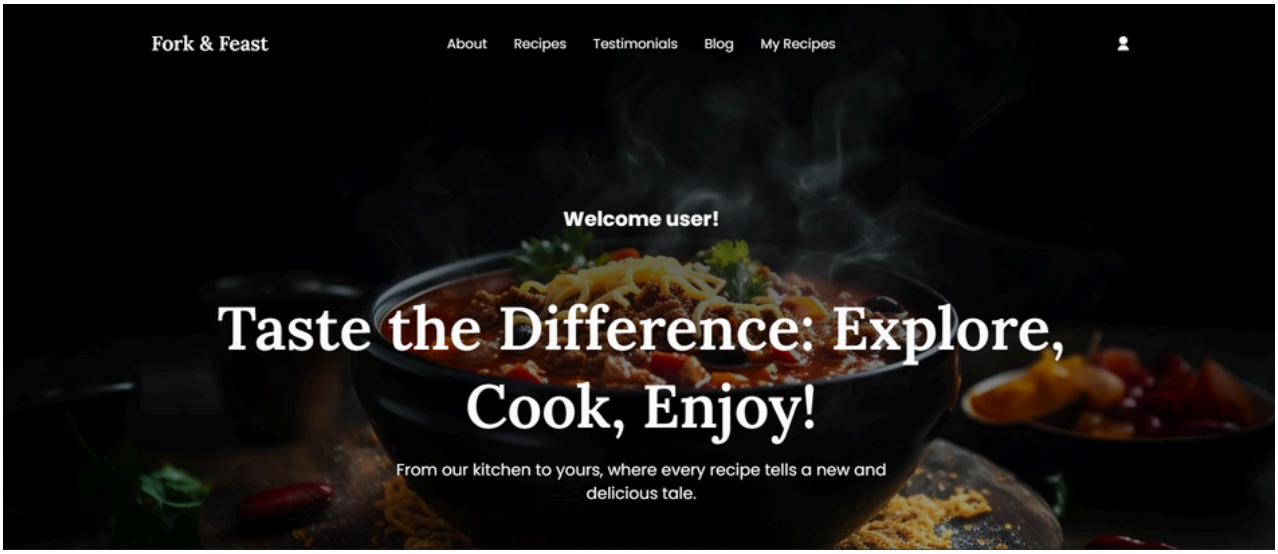


Fig 7.4- Welcome/Home Page

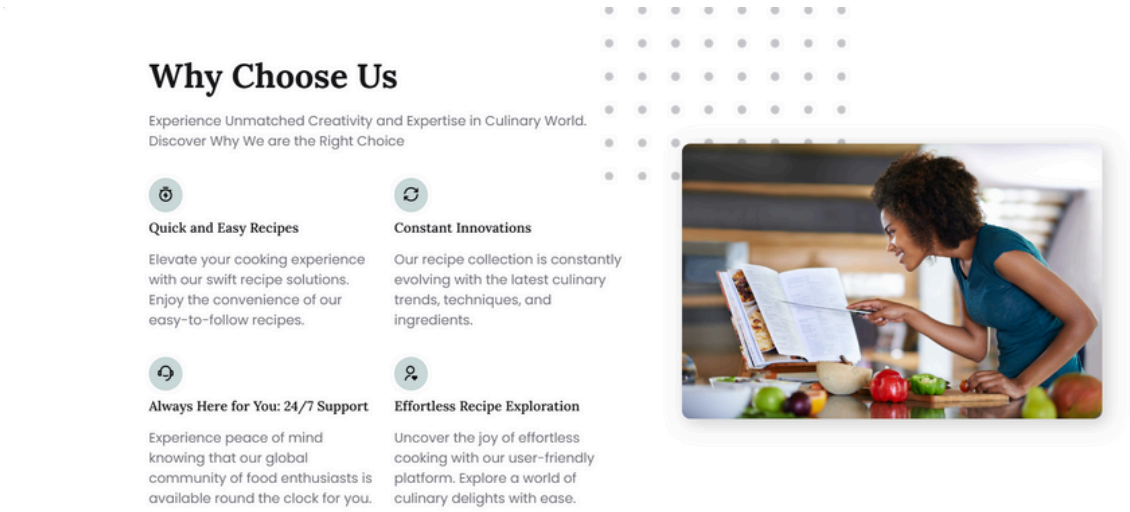


Fig 7.5- About Us Page

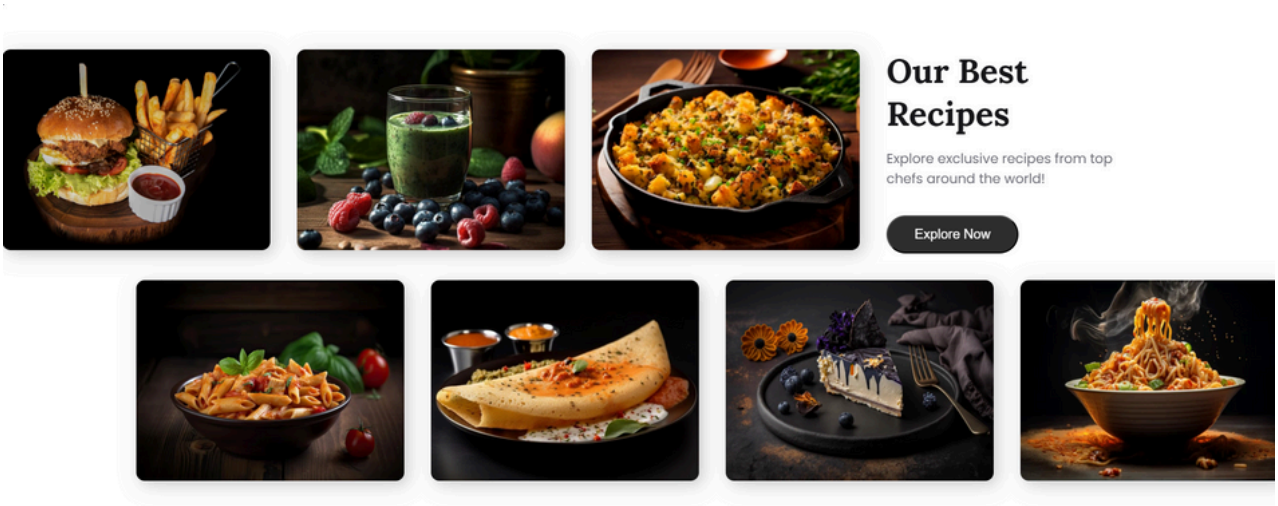


Fig 7.6- Recipes Page

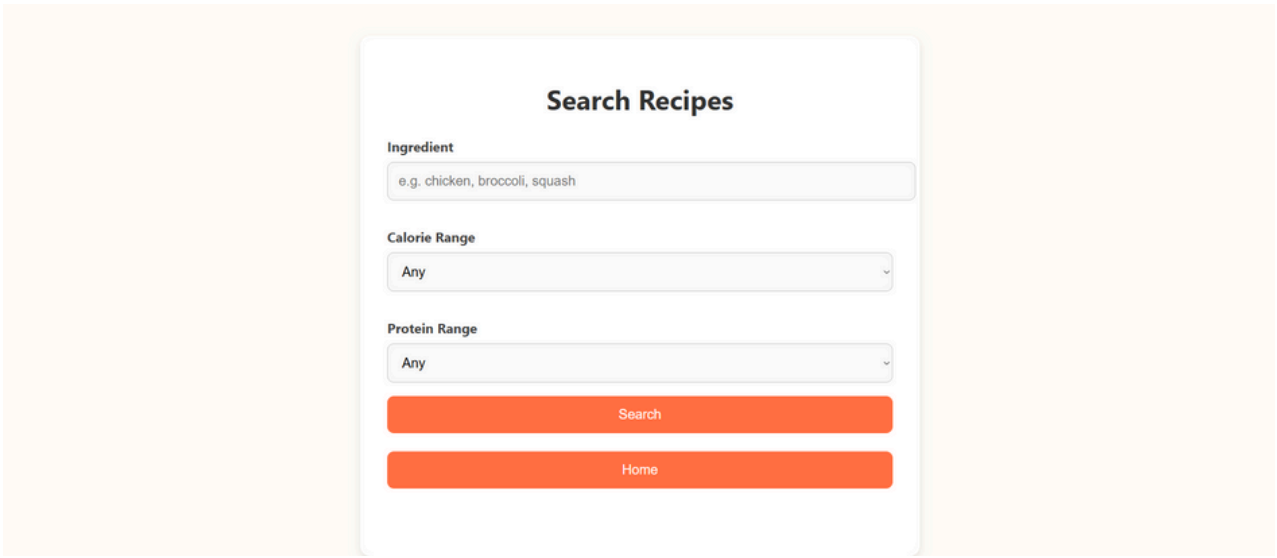


Fig 7.7- Search Recipes Page

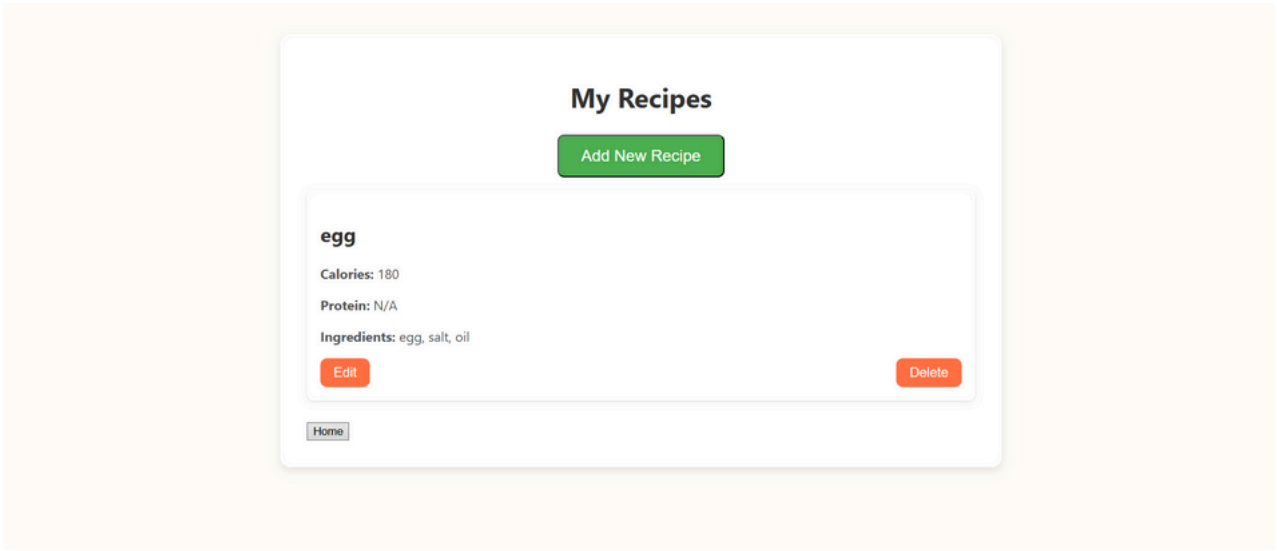


Fig 7.8- My Recipes Page

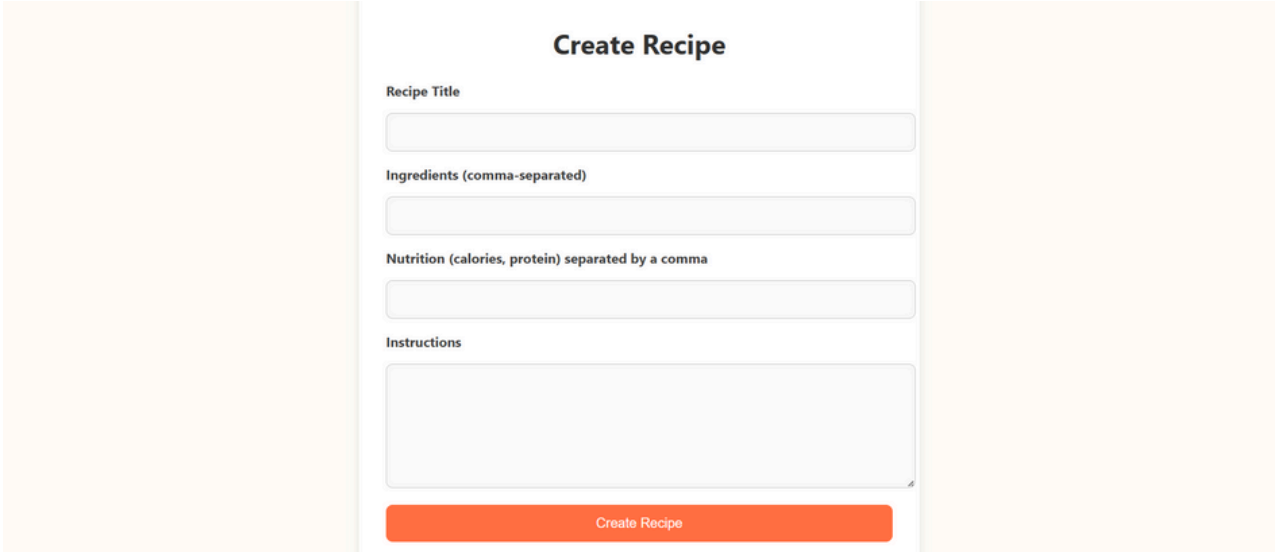


Fig 7.9- Create a Recipe Page

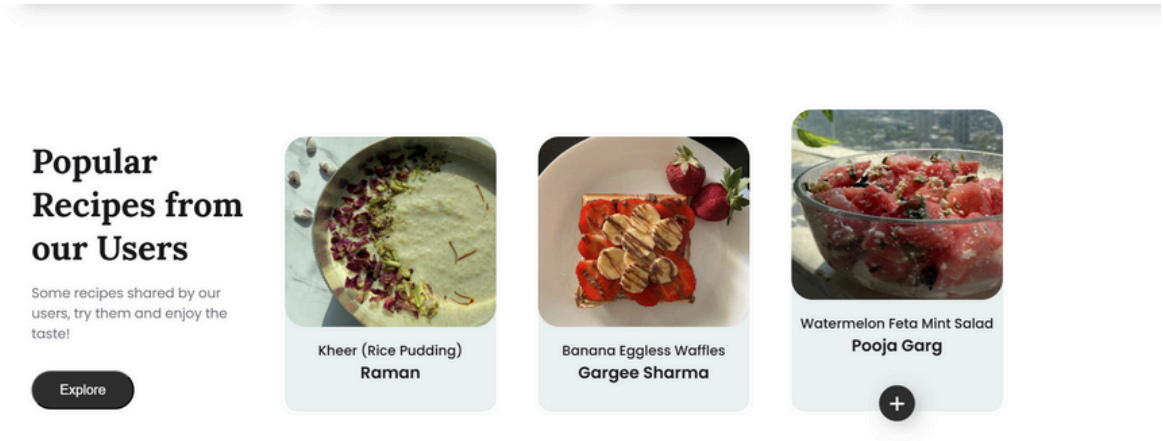


Fig 7.10- Explore Recipes Page

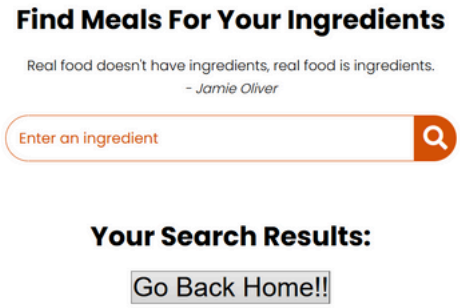


Fig 7.11- Enter Ingredients Page

Fig 7.12- Testimonials Page

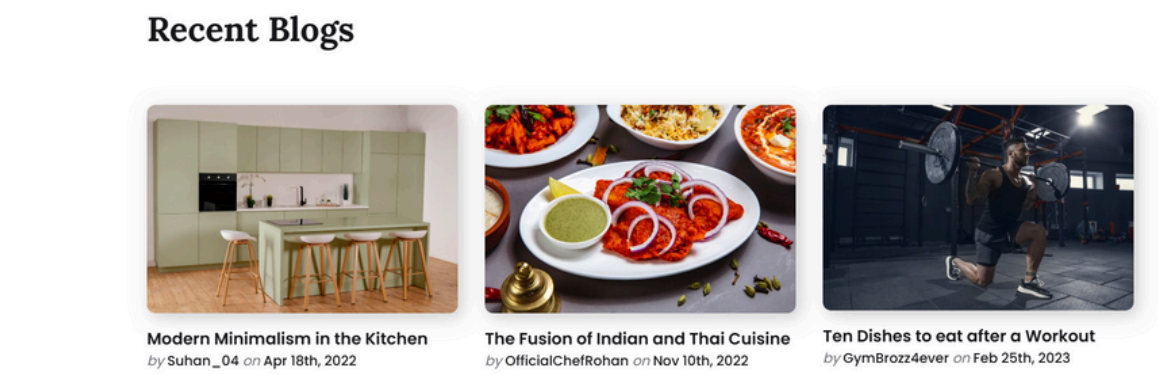


Fig 7.13- Blogs Page

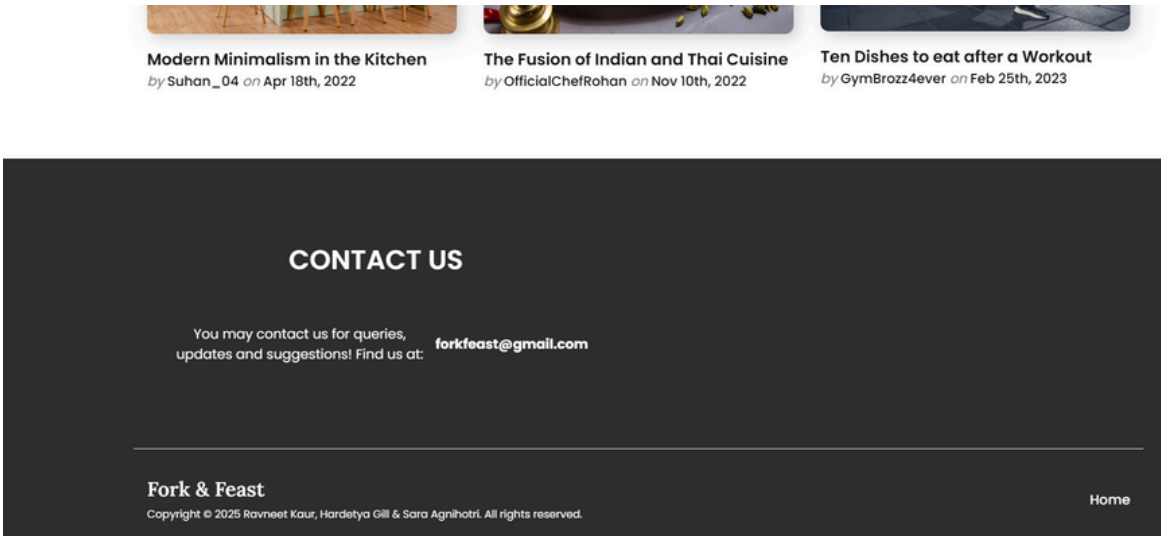
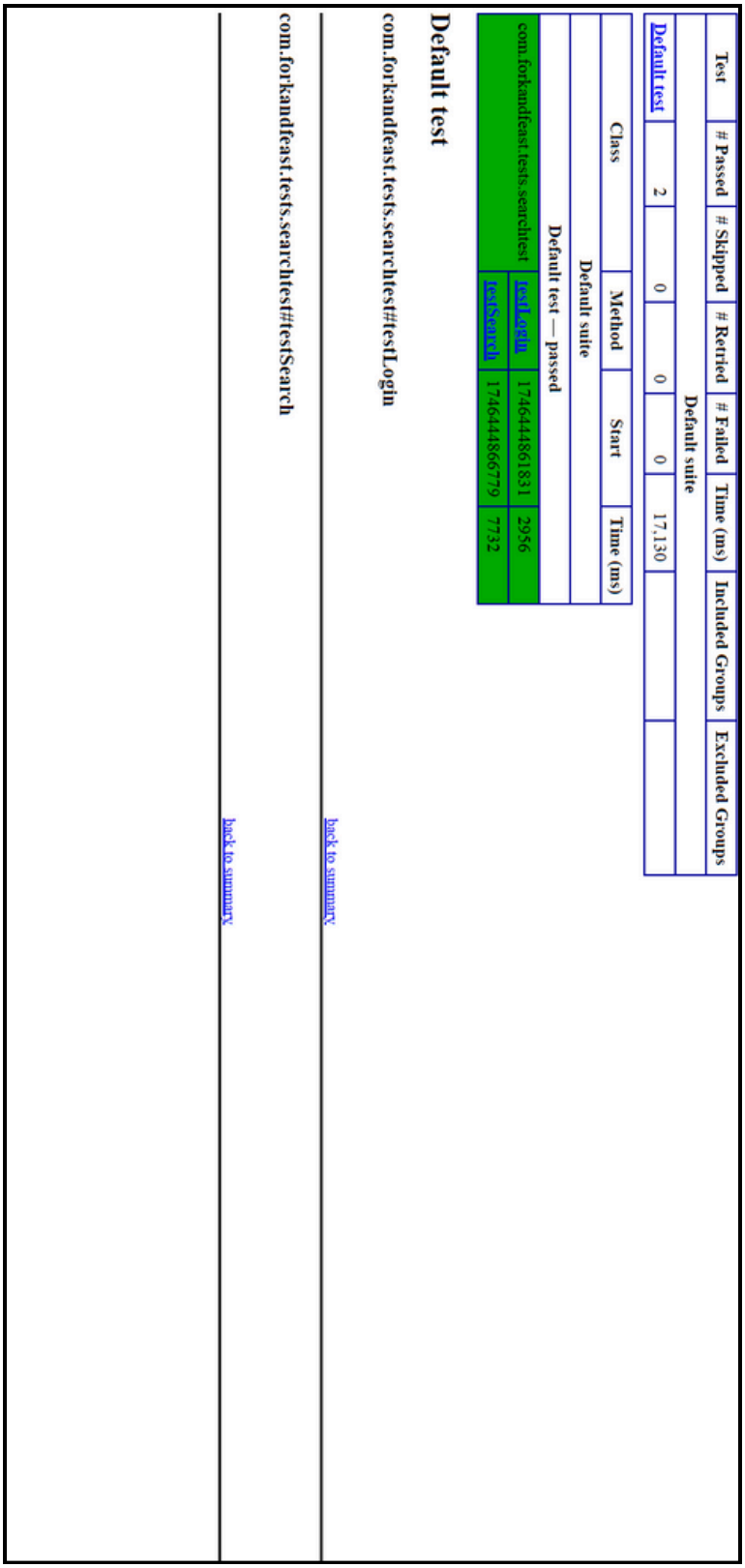


Fig 7.14- Footer Page

OUTPUT REPORTS

8.1 EMAILABLE REPORT



Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	2	0	0	0	17,130		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
com.forkandfeast.tests.searchtest	testLogin	1746444861831	2956
	testSearch	1746444866779	7732

Default test

com.forkandfeast.tests.searchtest#testLogin

back to summary

com.forkandfeast.tests.searchtest#testSearch

back to summary

Fig 8.1- Emailable Report

8.2 INDEX REPORT

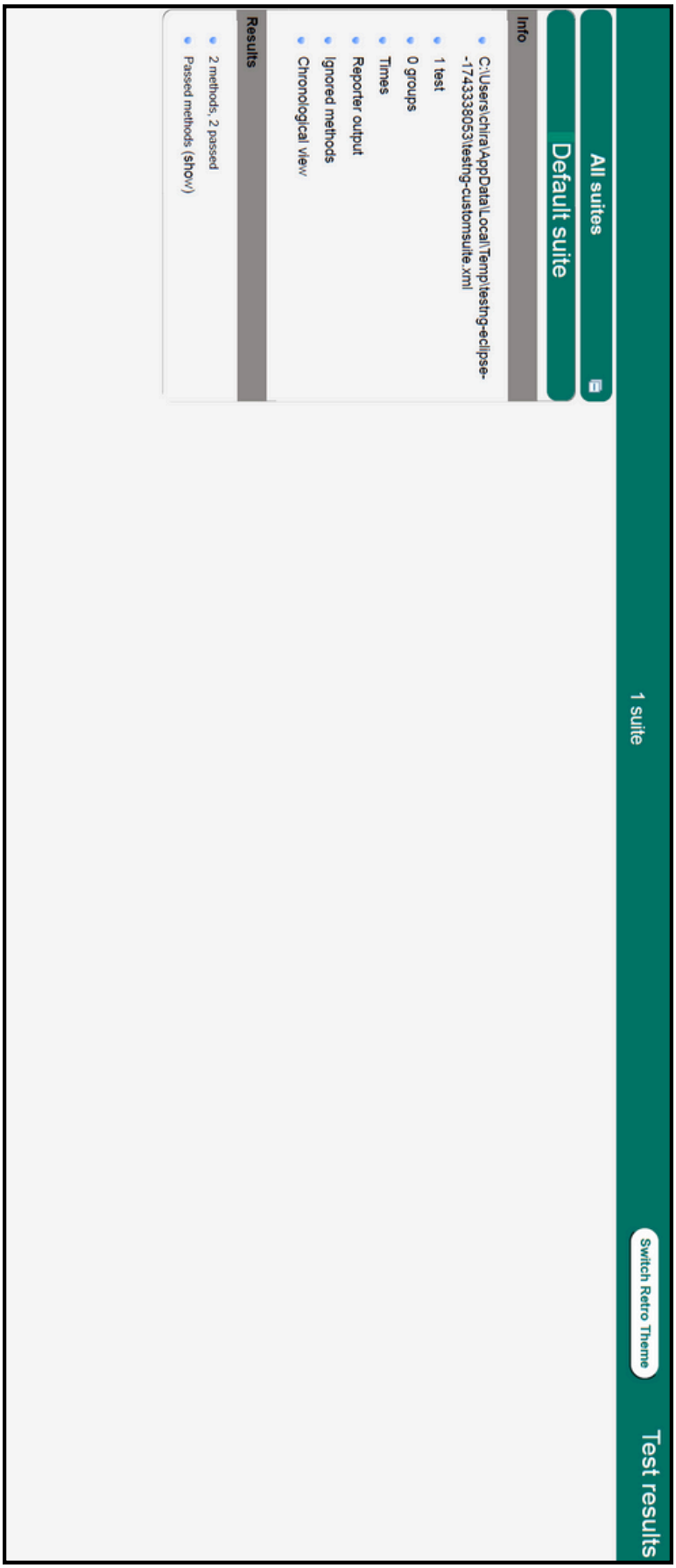


Fig 8.2- Index Report

8.3 XSLT REPORT

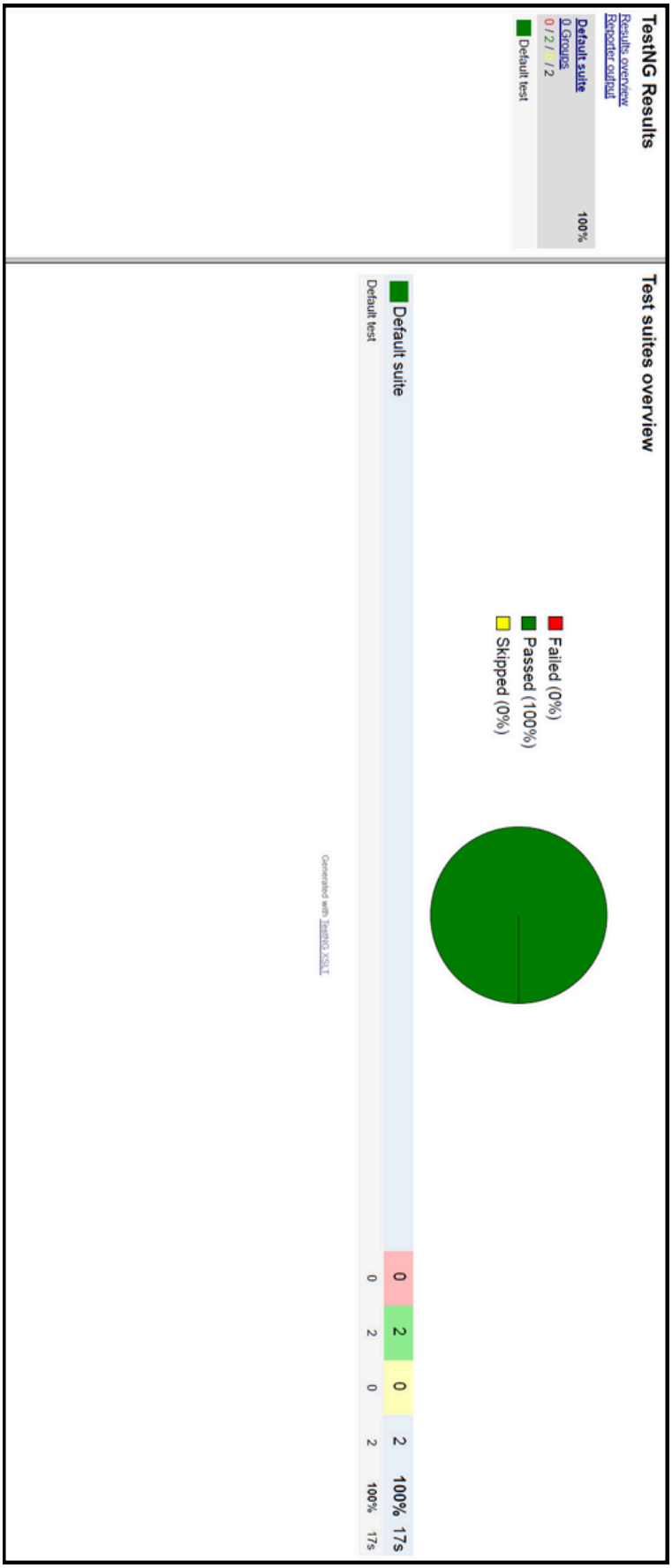


Fig 8.3- XSLT Report

CONCLUSION

The Fork & Feast project is a testament to the capabilities of full-stack web development combined with powerful test automation. It solves a real-world problem by helping users plan meals based on available ingredients, thus promoting food efficiency and user convenience.

By implementing both functional and non-functional requirements thoroughly, the project ensures an engaging and robust user experience. Automation tools like Selenium and TestNG further elevate the project by guaranteeing consistent performance and easing the burden of manual testing.

This project not only highlights technical skills in web development and test engineering but also reflects the importance of user-centric design and automation in modern applications. Fork & Feast is scalable, maintainable, and ready for future extensions like RESTful APIs, AI-based recipe recommendations, and mobile compatibility.

THANK YOU