WEB TECHNOLOGIES AND PLATFORMS

Tanay Jivanbhai Ahir
1984877

Naineesh Reddy Alluri
2198416

Contents

Introduction 2
Project Overview 2
Requirements and Functionality 3
Core Functionalities 3
Functional Requirements 4
Non-Functional Requirements 5
Key Features 6
Selected Web Technologies 7
System Architecture 8
UML Modelling 9
Use Case Diagram 9
Activity Diagram: "Log-in/Register" Use Case 11
Activity Diagram: "Home Page" Use Case 12
Implementation 13
Development Schedule 17
Future Enhancements 18
Conclusion 19
Project Division & Individual Contributions 19
Part 1: Naineesh Reddy Alluri (ID: 2198416) 19
Responsibilities: 19
Part 2: Tanay Jivanbhai Ahir (ID: 1984877) 20
Responsibilities: 20
References 21

Introduction

With the advent of the digital age, online apps have emerged as essential resources for connecting, sharing, and interacting with people across the world. The internet provides an unparalleled means of communication and sharing for companies, individual blogs, and large-scale publication platforms. Chefs may now post recipes, engage with foodies, and create a dynamic, interactive environment thanks to digital platforms in the culinary industry.

The creation of a web service that would connect foodies and chefs is the topic of this article. The website gives cooks a way to create, arrange, and distribute their recipes as well as interact with their followers in real time. It makes it easier to communicate directly, get comments, and find recipes.

The goal, key features, and technologies used in the creation of this platform, as well as its system architecture and security aspects, are all covered in the paper. The deployment strategy, industry trends influencing contemporary web apps, and UI/UX design concerns will also be covered. A prototype will show off the interface and user experience, giving a detailed look at its features.

Project Overview

With the help of the suggested web application, chefs and foodies will be able to interact with, learn from, and exchange culinary material in a community setting. The primary goal is to establish a platform where chefs can oversee, develop, and share their recipes, and foodies can interact, discuss, and learn about new culinary concepts. The website is intended to foster a feeling of community by making foodies and chefs alike feel appreciated, informed, and involved. The website will concentrate on many main goals in order to achieve this:

1. **Recipe Creation and Publishing**: Using user-friendly interfaces, the website will make it simple for chefs to create, format, and publish recipes. Chefs can also update and maintain published recipes appropriately.

2. **Intuitive User Interface**: Both chefs and hobbyists will find it easy to use thanks to the user interface's simplicity and intuitiveness. The interface will be created to make using the platform less difficult, improving the user experience.

3. **Security**: To safeguard user information, personal information, and intellectual property, strong security measures will be implemented. Data privacy protection, secure authentication, and encryption will all be part of this.

4. Search and Filtering: The site's robust search and filtering tools will make it simple for visitors to look for recipes that fit their needs based on categories, ingredients, degree of difficulty, and other factors.

5. **Scalability**: The website will be designed to accommodate an expanding user base and massive amounts of material, guaranteeing seamless operation even during periods of high traffic.

6. **Accessibility**: Users will be able to visit the website at any time and from any location thanks to its comprehensive responsive design, which offers smooth access on PCs, tablets, and smartphones.

Requirements and Functionality

Core Functionalities

1. **User Authentication:**
   Both cooks and foodies will be able to authenticate themselves via the online application.
   Users must create a secure password and supply a working email address to register. To improve security and provide an extra line of defence against unwanted access, multi-factor authentication, or MFA, will be used. Crypt, a very safe cryptographic hashing method, will be used to safely hash and store

passwords. This guarantees that passwords remain safe even if user data is compromised.

2. **Profile Management:**

Profiles may be made and customised by users. In addition to writing personal biographies and adding connections to their social media profiles, they may post profile photographs. In order to interact with their fans and make their recipes public, chefs may create a portfolio. Chefs can change visibility settings to decide whether their recipes are shared with food enthusiast groups or the public to preserve privacy and control over the content.

3. **Recipe Creation and Management:**

To create and modify recipes, the platform would provide chefs access to a powerful yet user-friendly text editor. Users will be able to supplement their content with photographs, videos, and other media using the editor. Because they may change or remove their recipes at any moment, chefs will also be in control of them. This makes it possible to provide information that is up-to-date, pertinent, and in line with emerging culinary trends.

4. **Search and Filtering:**

The website will have a sophisticated search engine that will enable users to look for recipes using a variety of criteria to provide a more customised user experience. Users will have the option to search by publication date, ingredients, popularity, or recipe category. Additionally, AI-powered recommendations will personalise the surfing experience by recommending recipes according to each user's interests and surfing history. The purpose of the feature is to pique the attention of foodies and encourage them to try new foods.

5. **Responsive UI:**

A completely responsive design will guarantee a flawless user experience. The program will be available on a range of screen sizes because it will be optimised for desktop, tablet, and mobile contexts.
Flexbox and CSS Grid will be used to produce adaptable and visually appealing layouts. Additionally, media queries will improve responsiveness, guaranteeing smooth and simple navigation.
The design will prioritise user-friendly interfaces, pages that load quickly, and a logical layout that makes it easy for users to navigate between functions.

6. **Commenting and Interaction:**

There will be a comments system to encourage user engagement and conversation. Recipes are open to comments, queries, and suggestions from foodies. Nonetheless, chefs can react to these exchanges, fostering a feeling of community and ongoing discussion. A moderating mechanism will eliminate offensive content to provide a welcoming and peaceful atmosphere, fostering productive and positive discourse.

7. **Security Implementation:**

The platform places the utmost importance on security. Every user's data will be encrypted both at rest (using AES encryption) and in transit (using SSL/TLS encryption). Only authorised personnel will be allowed to carry out sensitive tasks like altering recipes or doing administrative tasks due to role-based access control, or RBAC.

Second, in order to monitor activities, spot any security threats, and guard against invasions, thorough user activity records will be kept.
Users will be able to interact with the platform in a secure and reliable environment thanks to these security measures.

Functional Requirements

1. User Management

- FR1: Users can register as either clients or chefs with different profile types
- FR2: Users can log in/out and reset passwords via email
- FR3: Chefs can create and manage comprehensive profiles (bio, specialties, rates, availability)

2. Chef Discovery & Booking

- FR4: Clients can search/filter chefs by cuisine, location, availability, and price
- FR5: Clients can view detailed chef profiles with ratings, menu options, and sample recipes
- FR6: Clients can submit booking requests with date/time, guest count, and special requests
- FR7: Chefs can accept/decline booking requests with optional messages

3. Recipe System

- FR8: Chefs can create and manage recipe collections
- FR9: Clients can browse chef recipes with filtering by cuisine/difficulty

- FR10: Recipe pages display ingredients, instructions, and chef notes

4. Communication

- FR11: In-app messaging between clients and chefs
- FR12: Automated notifications for booking confirmations/updates

Non-Functional Requirements

1. Performance

- NFR1: Search results load within 2 seconds for 95% of queries
- NFR2: System supports 1,000 concurrent users during peak hours

2. Usability

- NFR3: First-time users can complete booking in ≤3 steps
- NFR4: Platform supports at least English, Spanish, and French languages
- NFR5: Mobile-responsive design for all screen sizes

3. Reliability

- NFR6: 99.5% uptime excluding scheduled maintenance
- NFR7: Booking system prevents double-booking conflicts

4. Data Management

- NFR8: Chef profiles remain editable by owners only
- NFR9: All user data encrypted in transit and at rest

Key Features

1. **User Dashboard:**

A quick summary of the user's activities will be provided via the customised dashboard. This will contain interaction data, comments, new followers, and analytics on the popularity of the recipes for the chefs.
Foodies may view reading histories, recipes they've saved, and tailored suggestions based on their preferences.

2. **Recipe Creation Interface:**

   Using a WYSIWYG (What You See Is What You Get) editor, which supports basic formatting like bold, italics, lists, and headers, chefs may create material on the recipe creation page. To ensure that the material is shown as intended, chefs may also preview their recipe before posting.

3. **Profile Customization:**

Profile pages will allow users to personalise their looks, upload a profile photo, attach social network accounts, and create their profile's backdrop and colour palettes.
Chefs may share a portfolio of their previous work, and foodies can follow their favourite chefs to receive notifications of new recipes.

4. **Advanced Search & Filters:**

   An Elasticsearch engine will power search, allowing it to process complex queries and produce quick, relevant results.
   Filters will allow consumers to sort recipes by ingredients, category, publish date, and rating, making it easier for foodies to locate their favourites.

5. **Real-Time Notifications:**

   The notification system will employ email and in-app notifications to notify users of any new messages, comments, or recipe modifications. Users will be able to customise notifications, allowing them to choose what they would like to get alerts on.

Selected Web Technologies

- **Front-End Technologies:**

  HTML5: The most recent version of HTML, featuring semantic tags for improved accessibility and organisation as well as support for multimedia elements.

  CSS3: Using more recent styling tools, including grid layouts, gradients, animations, and Flexbox. On a variety of devices, responsive design is made possible with media queries.

  JavaScript: Used to enable DOM modification, provide asynchronous programming via Promises and async/await, and provide dynamic content and interactivity.

  React.js: JavaScript package for creating dynamic, modular user interfaces with reusable parts. It includes hooks for side effects and state management and renders content more efficiently by using a virtual document object model.

- **Back-End Technologies:**

  Node.js: An event-driven, non-blocking JavaScript runtime for server-side programming that creates scalable applications.

  Express.js: Node.js online application framework that makes middleware administration, routing, and API creation easier.

  MySQL: An SQL-based relational database management system that facilitates complicated queries and transactions that adhere to ACID standards.

- **Security Technologies:**
  - JWT Authentication, SSL/TLS Encryption

System Architecture

The platform follows the **Model-View-Controller (MVC)** architecture to separate concerns, making it easier to manage and scale.

- **Model:** Represents the data structure and business logic. It includes classes for users, recipes, and comments.
- **View:** Responsible for rendering the user interface, using React.js.
- **Controller:** Handles incoming HTTP requests and communicates with the model to fetch or manipulate data, returning appropriate responses to the view.

UML Modelling

Use Case Diagram

This diagram shows the high-level functionalities and interactions within the platform, including how chefs, food lovers, and the system interact.

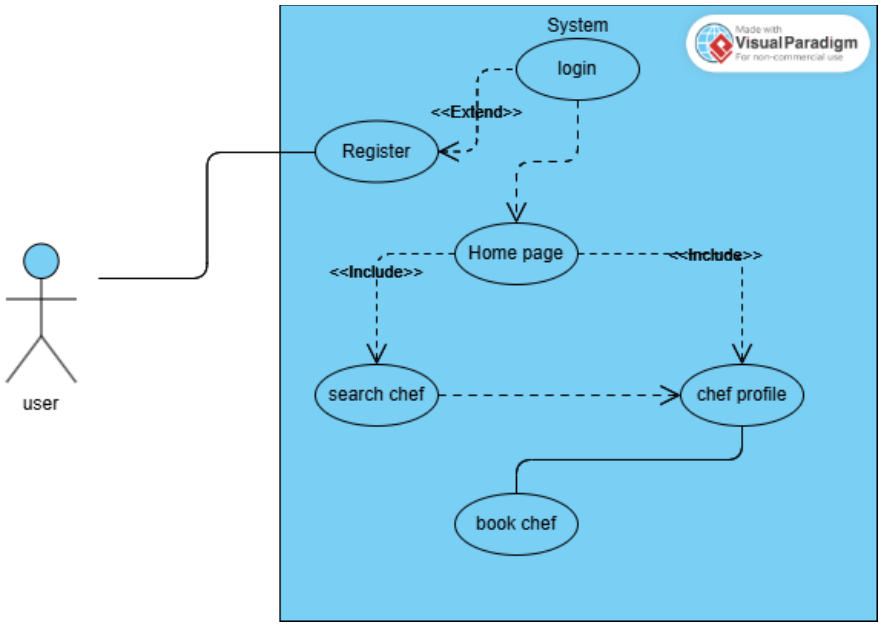
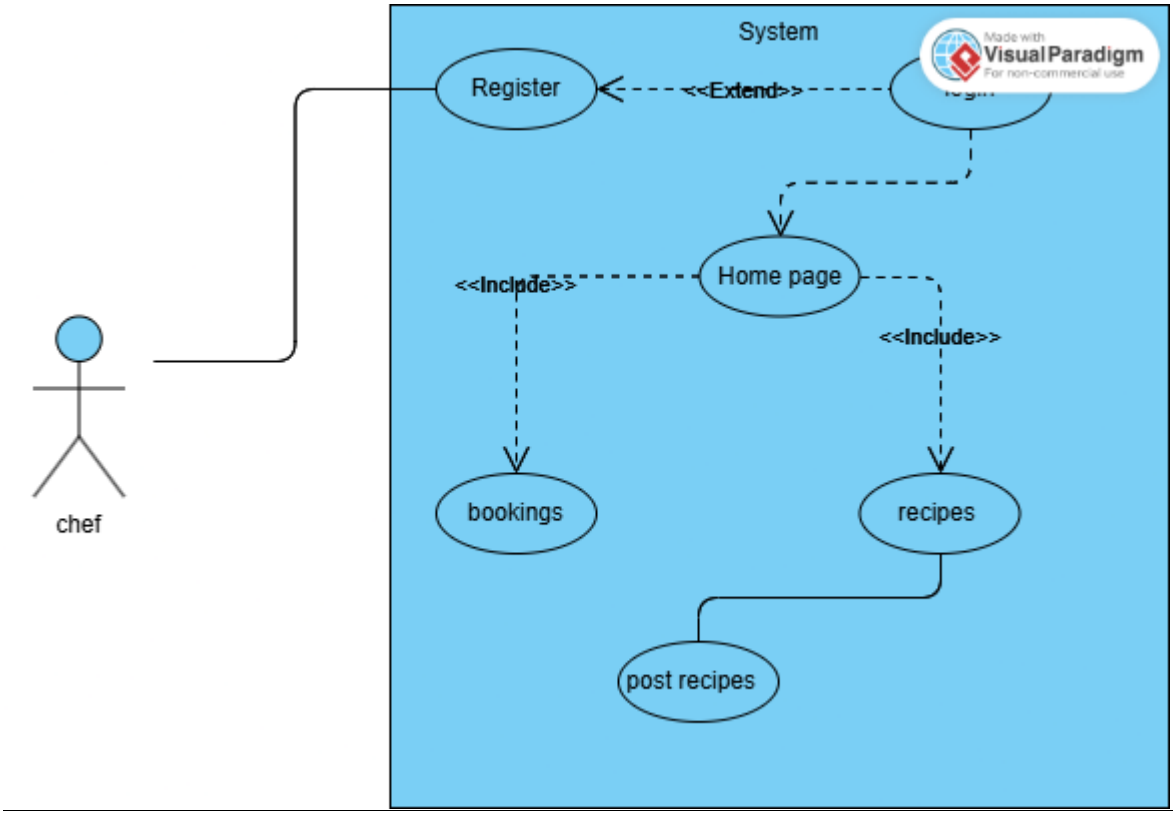


Diagram 1


System
Register
<<Extend>>
Home page
<<Include>>
<<Include>>
bookings
recipes
post recipes
chef
Made with VisualParadigm
For non-commercial use

Activity Diagram: "Log-in/Register" Use Case

This activity diagram outlines the steps involved when a chef creates a new recipe.

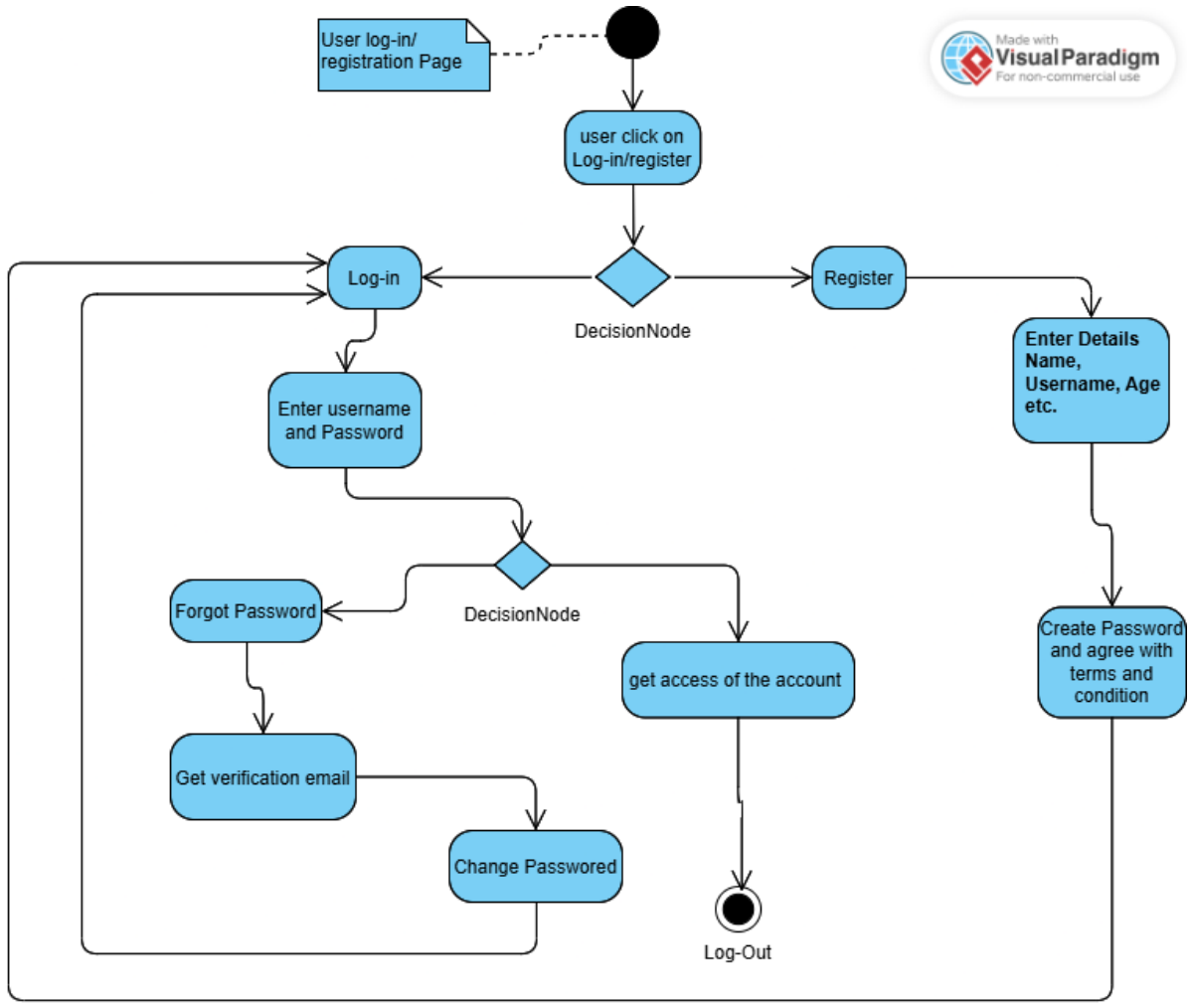


Diagram 2

Activity Diagram: "Home Page" Use Case

This activity diagram shows the flow of actions when a User comments on a recipe.

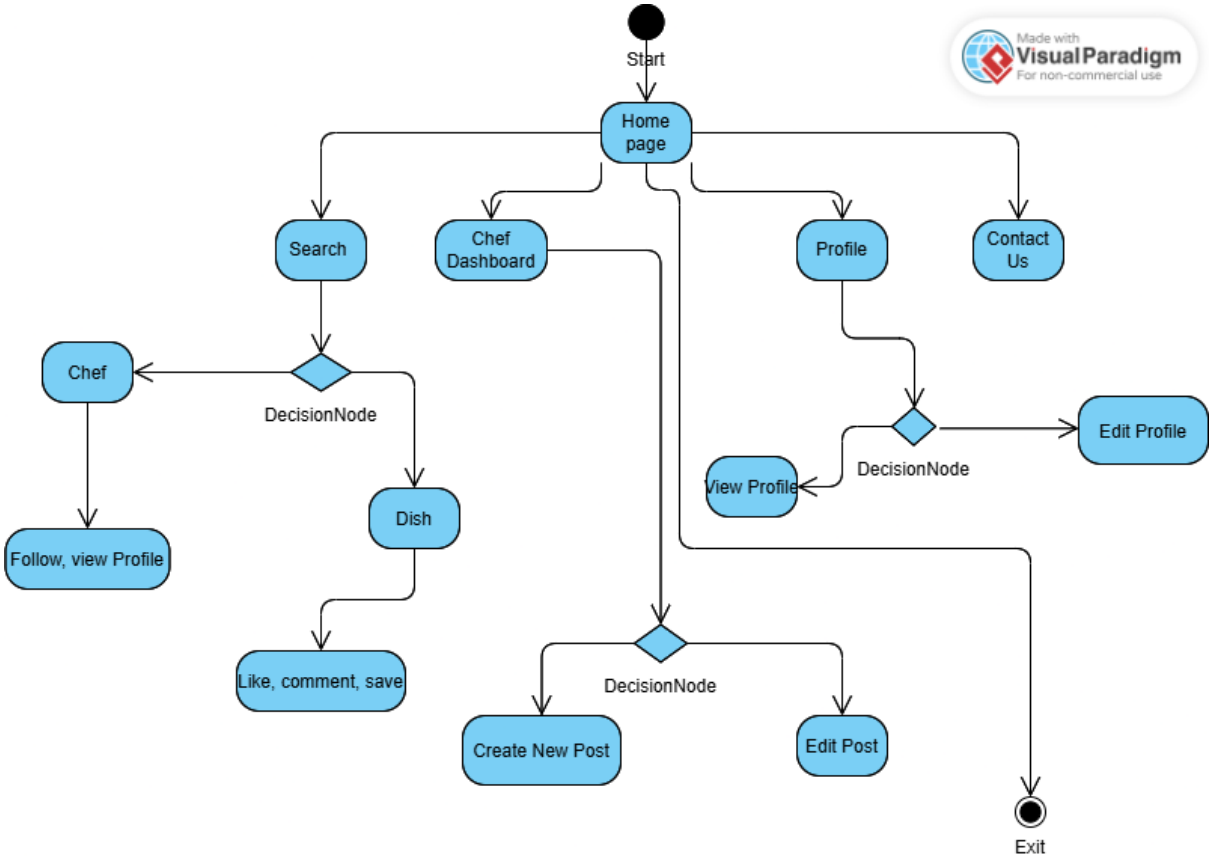


Diagram 3

Implementation

The prototype will be developed in stages. Each component will be fully functional and connected to the backend. Key features include:

1. **Login Page:** Allows users to log in securely

Technologies Used:

- ✓ Frontend: React.js for dynamic form handling and validation.
- ✓ Backend: Node.js with Express.js to handle authentication logic.
- ✓ Database: MySQL for storing user credentials securely (hashed passwords).

Challenges & Solutions:

1. Challenge: Ensuring real-time validation for user inputs (e.g., email format, password strength).
2. Solution: Implemented client-side validation using React hooks and server-side validation via Express middleware.

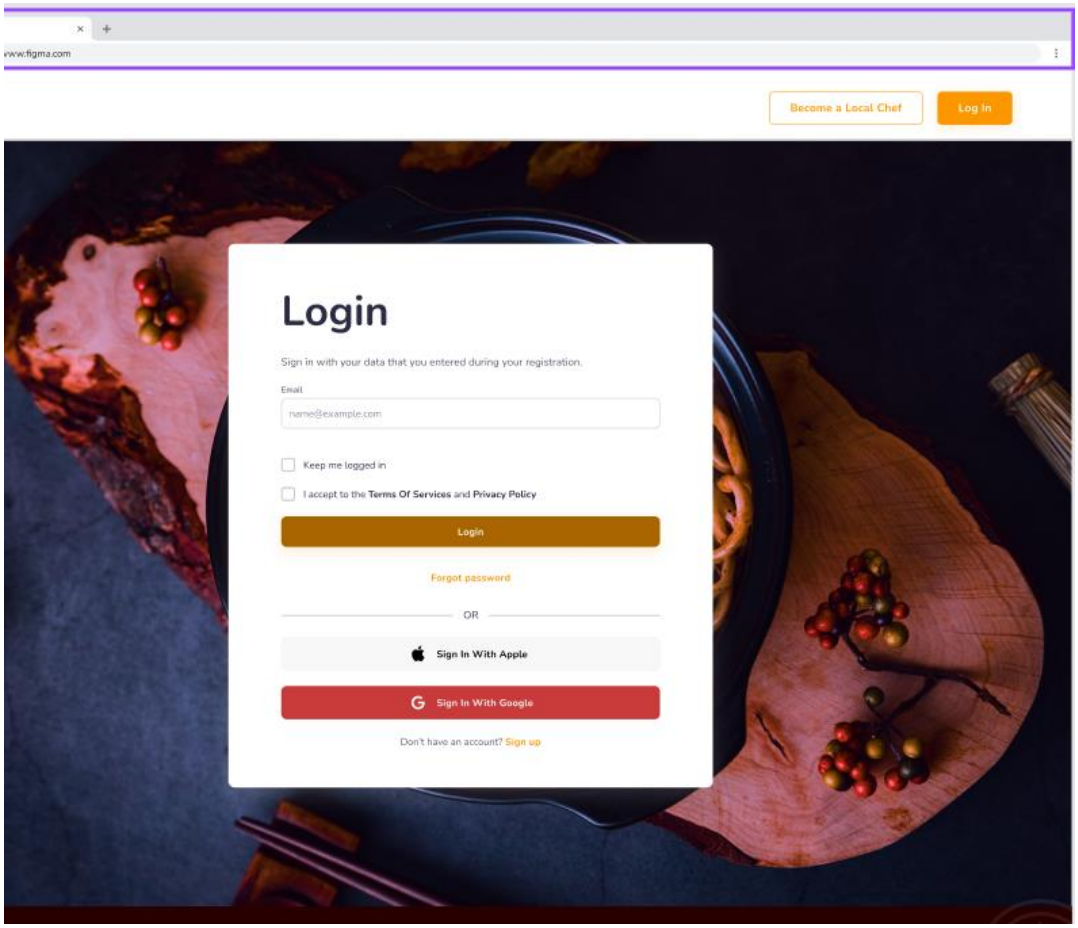


Image 1

2. **<u>Chef Dashboard</u>:** Provides chefs with an intuitive interface to manage their recipes.

Key Features:

- ✓ Recipe Management: Chefs can create, edit, and delete recipes using a WYSIWYG editor.
- ✓ Analytics: Displayed metrics such as recipe views, likes, and comments.
- ✓ Booking Requests: Integrated a calendar view to manage and respond to booking requests.

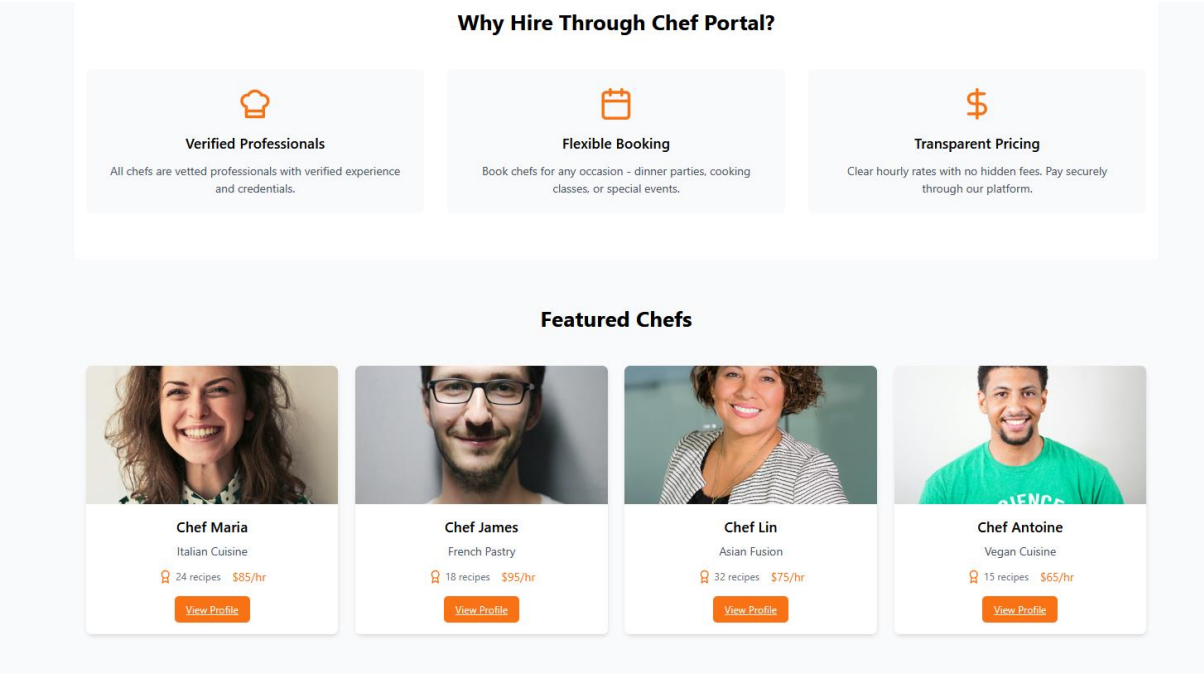
Technologies Used:

- ✓ Frontend: React.js with Chart.js for analytics visualization.
- ✓ Backend: RESTful APIs (Node.js/Express) to fetch and update recipe/booking data.
- ✓ Database: MySQL with ACID-compliant transactions for booking conflicts (NFR7).

Challenges & Solutions:

- ✓ Challenge: Handling concurrent booking requests to prevent double-booking.
- ✓ Solution: Implemented database locks and transaction rollbacks for conflicting requests.

Image 2



3. **Profile Page:** Allows chefs and food lovers to customize their profiles.

Key Features:

- ✓ Profile Customization: Users can upload profile pictures, update bios, and link social media accounts.
- ✓ Portfolio Display: Chefs can showcase their work with images and videos.
- ✓ Privacy Settings: Role-based access control (RBAC) to manage visibility of personal information.

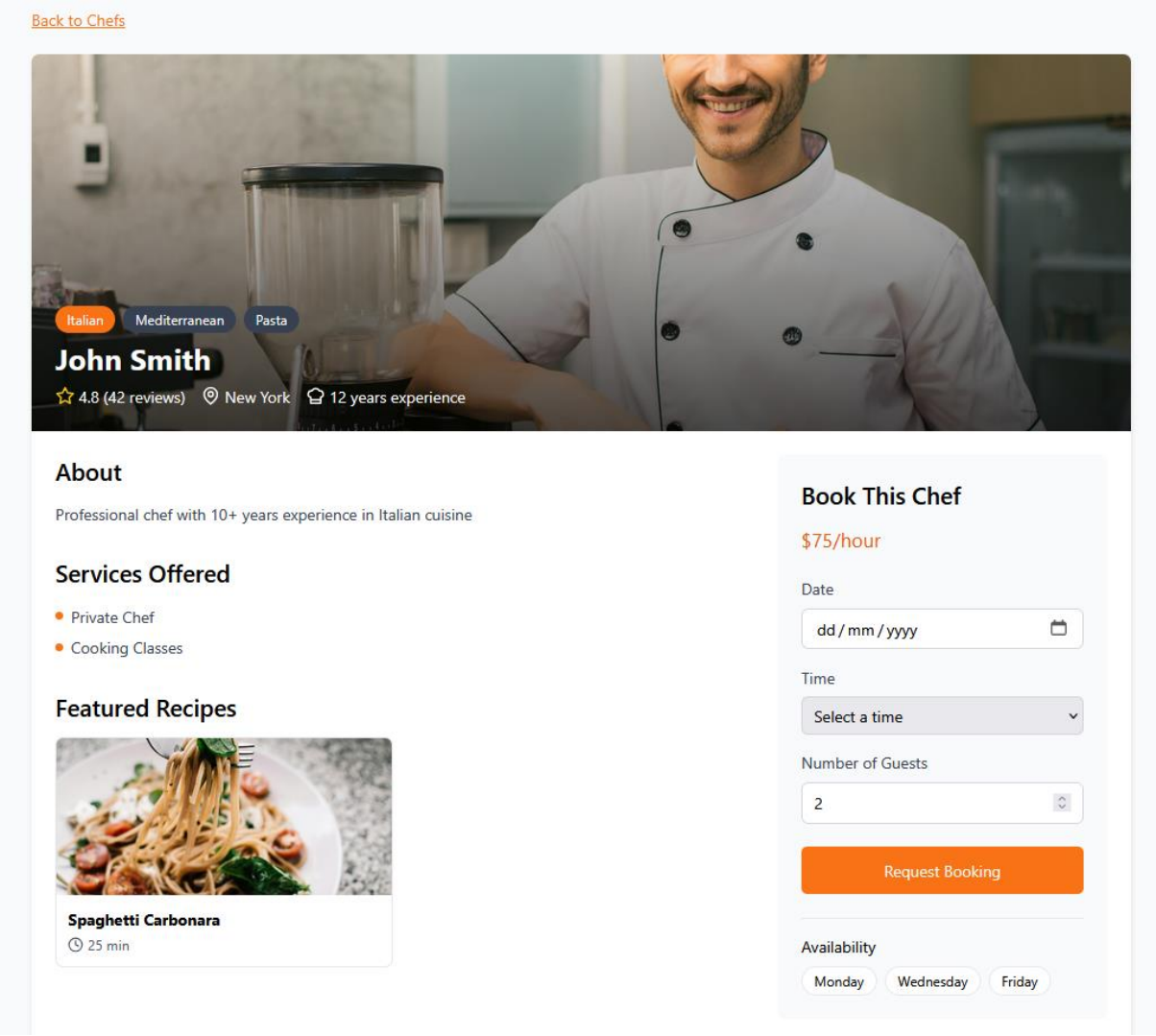
Technologies Used:

- ✓ Frontend: React.js with drag-and-drop file uploads (using React Dropzone).
- ✓ Backend: Node.js for handling file uploads (stored in AWS S3 for scalability).
- ✓ Database: MySQL for storing user metadata.

Challenges & Solutions:

- ✓ Challenge: Optimizing image uploads for performance.
- ✓ Solution: Compressed images client-side before uploading and used CDN for faster delivery.

Image 3


Back to Chefs
Italian
Mediterranean
Pasta
John Smith
4.8 (42 reviews)
New York
12 years experience
About
Professional chef with 10+ years experience in Italian cuisine
Services Offered
Private Chef
Cooking Classes
Featured Recipes
Spaghetti Carbonara
25 min
Book This Chef
$75/hour
Date
dd / mm / yyyy
Time
Select a time
Number of Guests
2
Request Booking
Availability
Monday
Wednesday
Friday

4. **Recipe Page:** A rich-text editor for chefs to create and publish recipes.

Key Features:

- ✓ Rich-Text Editor: Chefs can format recipes with headings, lists, and media embeds (TinyMCE editor).
- ✓ Preview Mode: Allows chefs to preview recipes before publishing.
- ✓ Interactive Elements: Users can like, comment, and save recipes.

Technologies Used:

- ✓ Frontend: React.js with TinyMCE for the WYSIWYG editor.
- ✓ Backend: Node.js to handle recipe CRUD operations.
- ✓ Database: MySQL with full-text search capabilities for Elasticsearch integration.

Challenges & Solutions:

- ✓ Challenge: Supporting real-time updates for comments and likes.
- ✓ Solution: Implemented WebSocket connections for live interactions.

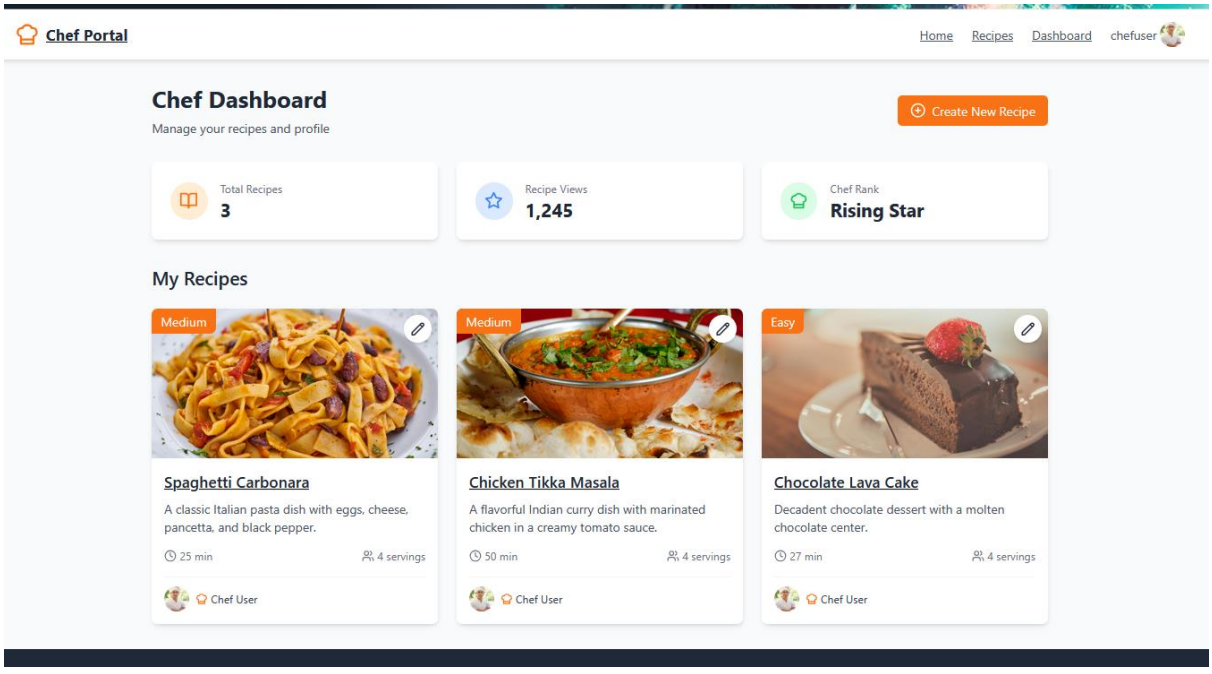


Image 4

5. **Search & Filter Page:** Enables User to search and filter chef and recipes.

Key Features:

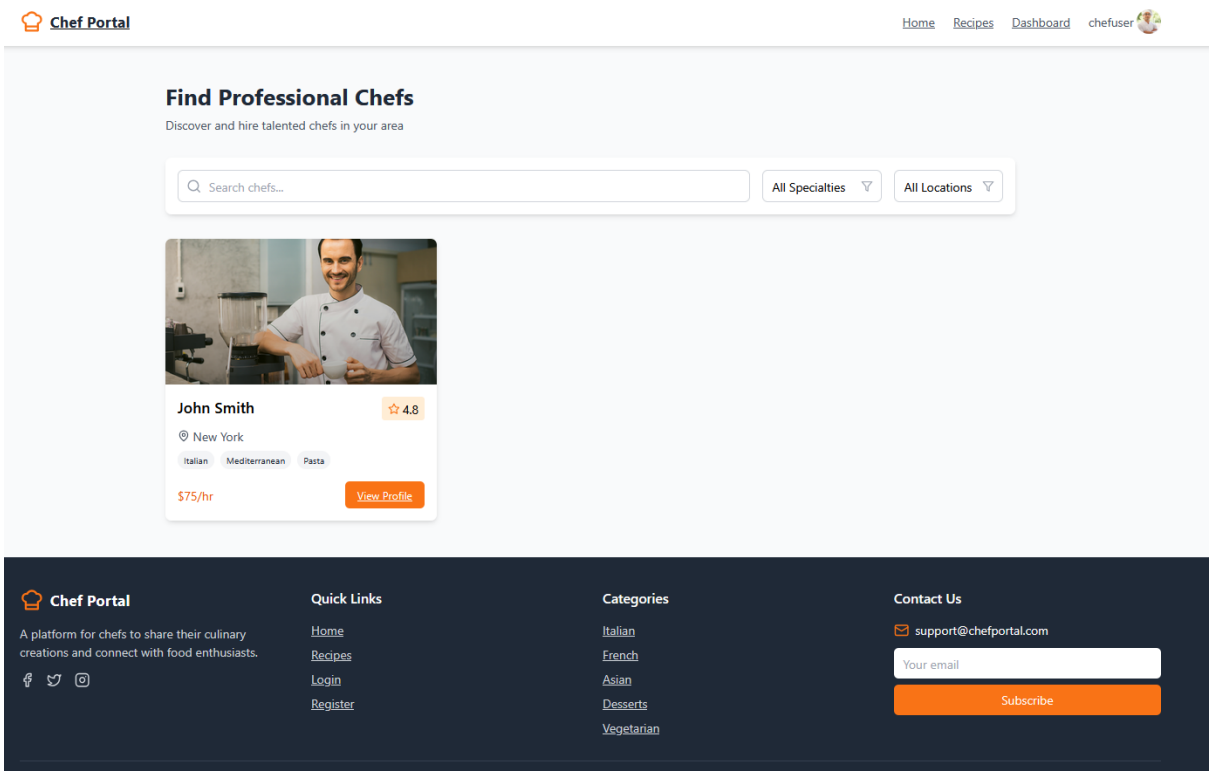
- ✓ Advanced Filters: Users can filter by cuisine, difficulty, ingredients, and ratings.
- ✓ Elasticsearch Integration: Powered fast and relevant search results (NFR1).
- ✓ AI Recommendations: Suggested recipes based on user preferences and browsing history.

Technologies Used:

- ✓ Frontend: React.js with debounced search queries to optimize performance.
- ✓ Backend: Node.js/Express with Elasticsearch for indexing and querying.
- ✓ Database: MySQL for storing recipe metadata.

- ✓ Challenge: Scaling search performance for large datasets.
- ✓ Solution: Implemented Elasticsearch with sharding and replication.

Image 5



Testing

- ✓ API Integration: Connected frontend and backend using Axios for HTTP requests.
- ✓ Unit Testing: Used Jest and React Testing Library for frontend components.
- ✓ End-to-End Testing: Conducted with Cypress to validate user workflows (e.g., booking a chef).

Development Schedule

- **Weeks 1-2:** Foundation & Design Environment configuration (React/Node.js stack setup)
- **UI/UX wireframing for all key interfaces:**
  - Chef discovery portal
  - Profile management system
  - Booking workflow screens
- **Multilingual architecture planning:**
  - Implement i18n framework selection
  - Prepare translation files structure
  - Design language toggle component Core component library development
- **Weeks 3-4:** Core Functionality Implementation
  - Backend API development:
  - Chef profile management endpoints
  - Booking reservation system
  - Search and filtering logic
- **Frontend integration:**
  - Dynamic chef listings with filters
  - Interactive booking calendar
  - User/chef dashboard skeletons
- **Language system implementation:**
  - Integrate translation files
  - Develop language switching mechanism
  - Test UI text rendering

Weeks 5-6: Refinement & Testing

- Responsive design adjustments
- Loading performance improvements
- Accessibility enhancements

Language system completion:

- Finalize all translations
- Implement dynamic content localization
- Test right-to-left language support

User testing and iteration:

- Conduct usability sessions
- Gather feedback on multilingual experience
- Refine booking workflow

Future Enhancements

1. **AI-Powered Content Recommendations**
2. **Multilingual Support**
3. **Mobile Application Development**
4. **Blockchain Integration**

Conclusion

In order to give foodies and cooks alike a fresh and engaging experience, the suggested Recipe Nest web application combines cutting-edge technology, user-centric design, and strong security measures. A highly engaging and immersive experience is offered by the platform's built-in capabilities, which include push alerts, extensive search and filtering, recipe development, and maintenance. Recipe Nest creates an active community by providing cooks with a specialised forum to exchange culinary knowledge and enabling food enthusiasts to find, rank, and comment on recipes. With features like role-based access control, data encryption, and secure authentication, Recipe Nest also prioritises security and privacy. Recipe Nest may grow even further with potential future connections, such as blockchain integration to provide open royalties administration and AI-driven content suggestion to customise the user experience. It will continue to be a useful platform for the production, consumption, and community involvement of culinary content, making chefs reachable by a wider, global audience.

Project Division & Individual Contributions

Part 1: Naineesh Reddy Alluri (ID: 2198416)

Responsibilities:

- ✓ Frontend Development & UI/UX Implementation
- ✓ Designed and developed React.js components:

- ✓ Chef discovery portal (ChefCard, SearchFilters)
- ✓ Booking workflow (CalendarPicker, BookingForm)
- ✓ Implemented responsive layouts using CSS3 Flexbox/Grid
- ✓ Integrated i18n for multilingual support (English/Spanish/French)

System Architecture & Prototyping

- ✓ Created Figma wireframes for all key interfaces
- ✓ Implemented MVC architecture:
- ✓ View Layer: React.js components with state management (Context API)
- ✓ Controller Layer: Express.js route handlers

Documentation:

- ✓ Wrote Sections:
- ✓ Introduction
- ✓ Project Overview
- ✓ Selected Web Technologies (Frontend: HTML5/CSS3/React.js)
- ✓ Prototype (Login Page, Chef Dashboard)

Key Deliverables:

- ✓ Functional chef search interface with filters (FR4-FR5)
- ✓ Booking request form with validation (FR6-FR7)
- ✓ Mobile-responsive UI (NFR5)

Part 2: Tanay Jivanbhai Ahir (ID: 1984877)

Responsibilities:

Backend Development & Database

- ✓ Built RESTful APIs using Node.js/Express.js:
- ✓ /api/chefs (search/filter)
- ✓ /api/bookings (CRUD operations)
- ✓ Designed MySQL schema:
- ✓ Tables: users, chefs, bookings, recipes
- ✓ ACID-compliant transactions for bookings (NFR7)

Security & Performance

- ✓ Implemented JWT authentication (FR1-FR3)
- ✓ Optimized SQL queries for <2s search results (NFR1)
- ✓ Configured rate limiting (1,000 concurrent users - NFR2)

Documentation:

- ✓ Wrote Sections:
- ✓ Requirements and Functionality (Core Features, FR/NFR)
- ✓ System Architecture (MVC, MySQL)
- ✓ UML Diagrams (Use Case, Activity Diagrams)

Key Deliverables:

- ✓ Secure user authentication (NFR9)
- ✓ Booking conflict prevention system (NFR7)
- ✓ API documentation (Swagger)

References

MDN Web Docs – Comprehensive documentation on HTML, CSS, JavaScript, and web technologies.

https://developer.mozilla.org/

W3Schools – Tutorials and references for modern web development, including React and ASP.NET Core.

https://www.w3schools.com/

Microsoft Learn – Official documentation for ASP.NET Core, C#, and Entity Framework.

https://learn.microsoft.com/

React Documentation – Official guide for building user interfaces with React.

https://react.dev

Smashing Magazine – Guides on UX/UI design, responsive layouts, and modern web technologies.

https://www.smashingmagazine.com/

Figma – Collaborative interface design tool for creating prototypes and user interfaces.

https://www.figma.com/community

Visual Paradigm – Software for creating UML diagrams and system modeling.

https://online.visual-paradigm.com/