Front-End UI/UX Mini Project

Project 1:FITNESS TRACKER DASHBOARD

Project 2:RECIPE BOOK

Submitted By:

• SARAH MARIAM RAJESH (2460445)

Gmail: sarah.mariam@btech.christuniversity.in

• ANN MARIA ABY (2460331) Gmail: ann.mariaaby@btech.christuniversity.in

• BELCITA BIJU (2460345)

Gmail: belcita.biju@btech.christuniversity.in

Course: Front-End UI/UX

Instructor name: Dr.Nagaveena

Institution: Christ University

Date of Submission: 26/09/2025

Project 1:FITNESS TRACKER DASHBOARD

2. Abstract

This project aims to design and develop a Fitness Tracker Dashboard using HTML, CSS, JavaScript, and Bootstrap. The application allows users to log workouts, track calories burned, set fitness goals, and visualize progress through an interactive chart. The focus was on creating a user-friendly, responsive interface with real-time updates and dynamic feedback.

3. Objectives

- Create an intuitive and visually appealing fitness tracking interface
- Enable users to add, view, and delete workout entries
- Display workout progress using a dynamic bar chart
- Allow users to set and track calorie goals
- Ensure the application is fully responsive across devices

4. Scope of the Project

- Front-end only (no backend or database)
- Uses JavaScript for dynamic interactions
- Responsive design using Bootstrap
- Data stored locally in the browser (resets on refresh)
- Includes form validation and interactive feedback

5. Tools & Technologies Used

Tool/Technology	**Purpose**
HTML5	Structure and layout
CSS3	Custom styling and animations
Bootstrap 4.6	Responsive UI components
JavaScript (jQuery)	DOM manipulation and interactivity
Chart.js	Data visualization

6. HTML Structure Overview

- Used semantic Bootstrap components: card, form, table
- Organized into sections:
    - Workout entry form
    - Goal setting panel
    - Progress chart
    - Workout history table
- Clean and modular structure for easy maintenance

7. CSS Styling Strategy

- External CSS file (styles.css)
- Gradient background with card-based layout
- Hover effects and transitions for interactivity
- Custom styles for badges, buttons, and alerts
- Fixed canvas dimensions for Chart.js responsiveness

8. Key Features

Feature	Description
Dynamic Workout Log	Add and delete workouts with real-time table updates
Goal Tracking	Set weekly calorie goals with visual progress alerts
Interactive Chart	Bar chart showing calories burned per workout session
Responsive Design	Works on mobile, tablet, and desktop
Visual Feedback	Color-coded alerts for goal achievement status

9. Challenges Faced & Solutions

Challenge	Solution
Chart responsiveness	Fixed canvas dimensions and used maintainAspectRatio: false
Dynamic data updates	Used Chart.js destroy() and re-render on data change
Goal progress calculation	Implemented percentage-based alerts with color-coded messages
Local data persistence	Used in-memory array (can be extended with localStorage)

10. Outcome

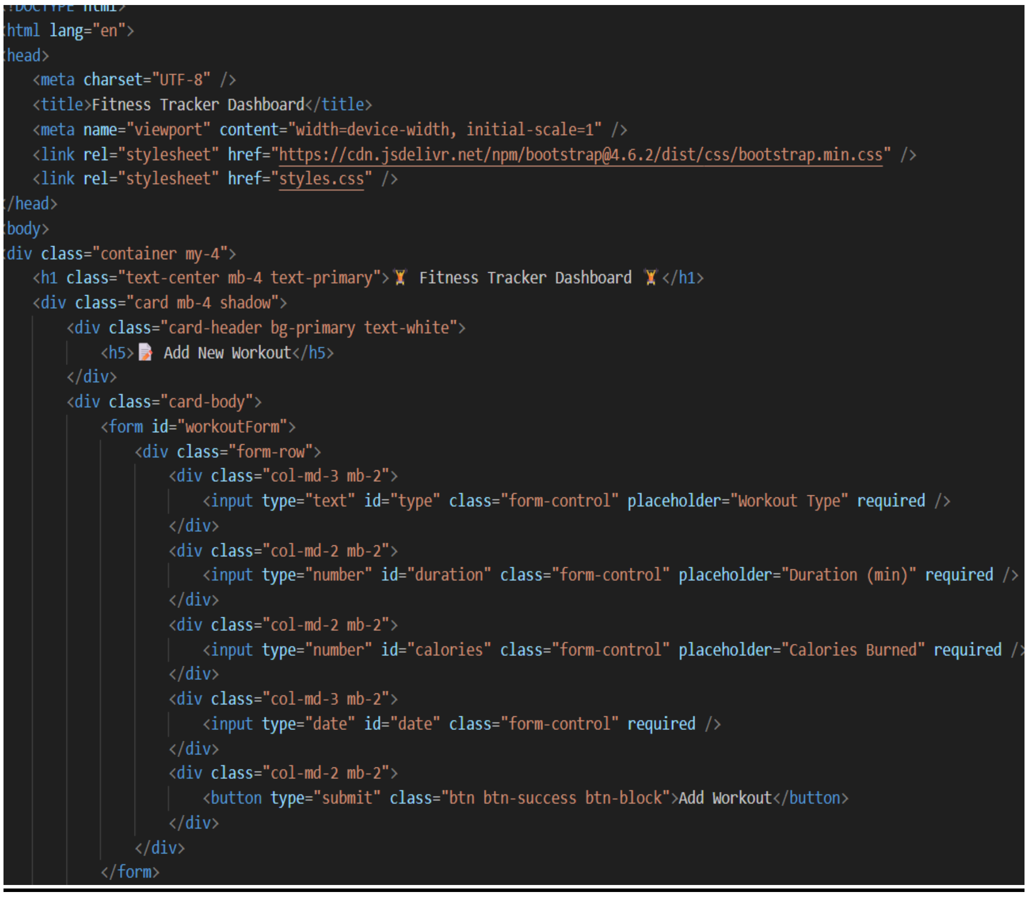
- Fully functional fitness tracking dashboard
- Clean and modern UI with smooth interactions
- Responsive across all screen sizes
- Effective use of third-party libraries (Bootstrap, Chart.js)

11. Future Enhancements

- Add user authentication and data persistence (e.g., localStorage)
- Include workout categories and filtering
- Add more chart types (e.g., line chart for trends)
- Integrate with fitness APIs for automatic data sync
- Export data as CSV or PDF

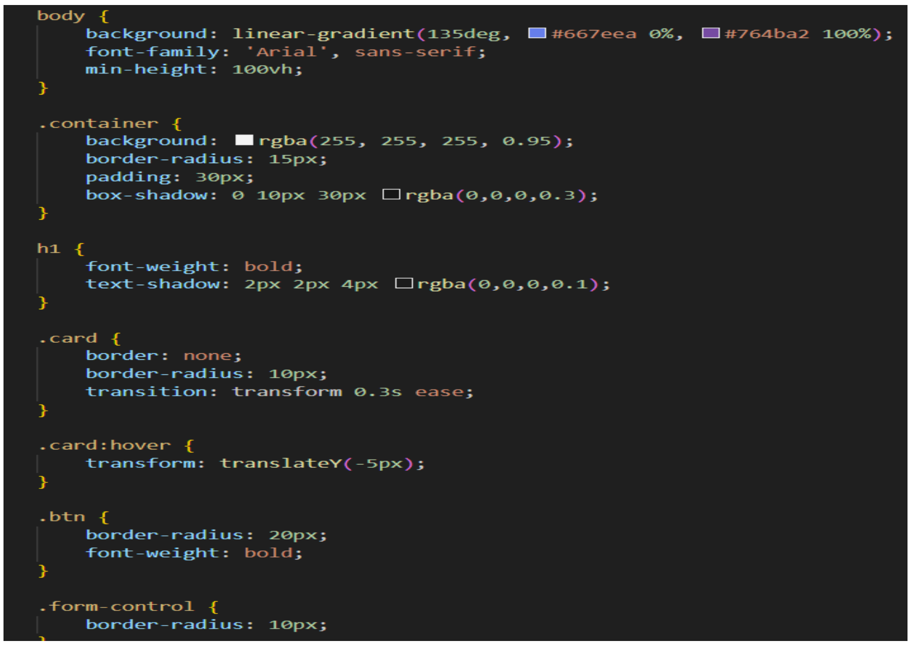
12. Sample Code

index.html



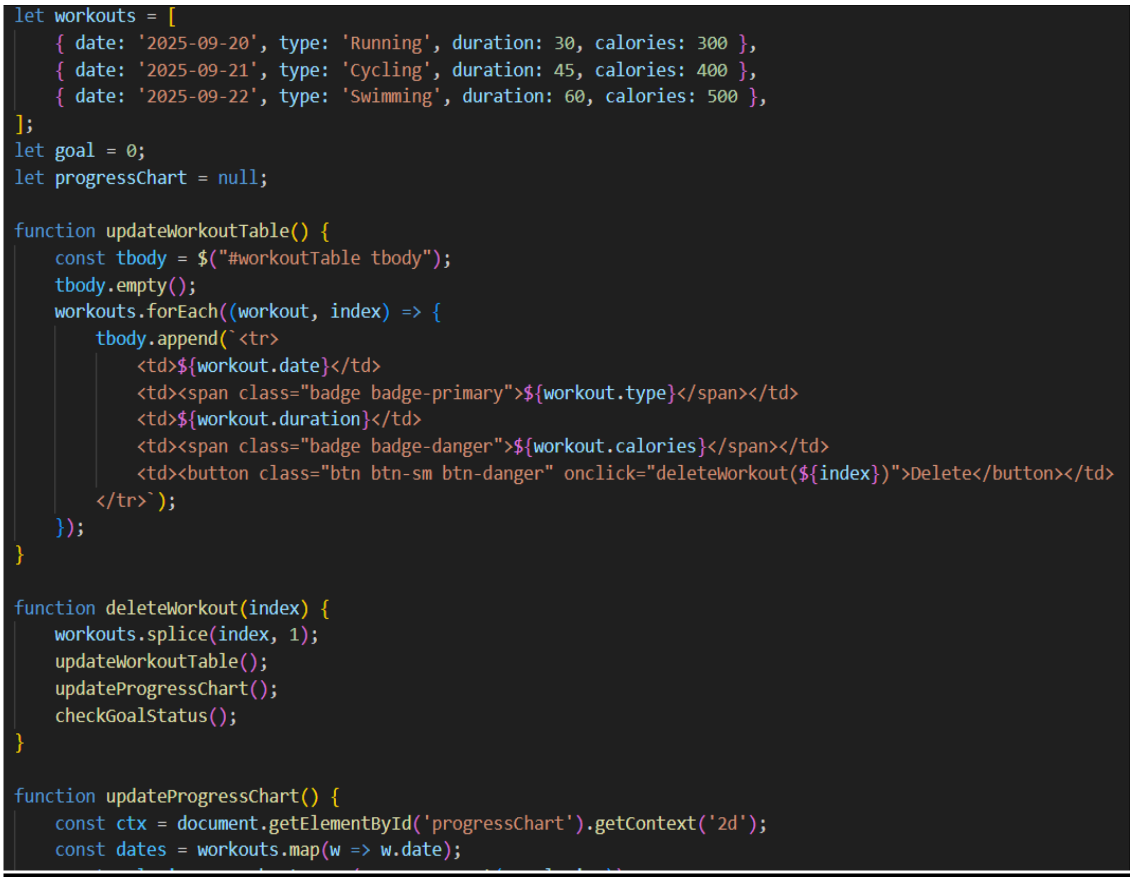
```html
!DOCTYPE html>
html lang="en">
head>
    <meta charset="UTF-8" />
    <title>Fitness Tracker Dashboard</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="styles.css" />
/head>
body>
div class="container my-4">
    <h1 class="text-center mb-4 text-primary">🏋️ Fitness Tracker Dashboard 🏋️</h1>
    <div class="card mb-4 shadow">
        <div class="card-header bg-primary text-white">
            <h5>📝 Add New Workout</h5>
        </div>
        <div class="card-body">
            <form id="workoutForm">
                <div class="form-row">
                    <div class="col-md-3 mb-2">
                        <input type="text" id="type" class="form-control" placeholder="Workout Type" required />
                    </div>
                    <div class="col-md-2 mb-2">
                        <input type="number" id="duration" class="form-control" placeholder="Duration (min)" required />
                    </div>
                    <div class="col-md-2 mb-2">
                        <input type="number" id="calories" class="form-control" placeholder="Calories Burned" required /
                    </div>
                    <div class="col-md-3 mb-2">
                        <input type="date" id="date" class="form-control" required />
                    </div>
                    <div class="col-md-2 mb-2">
                        <button type="submit" class="btn btn-success btn-block">Add Workout</button>
                    </div>
                </div>
            </form>
```

style.css



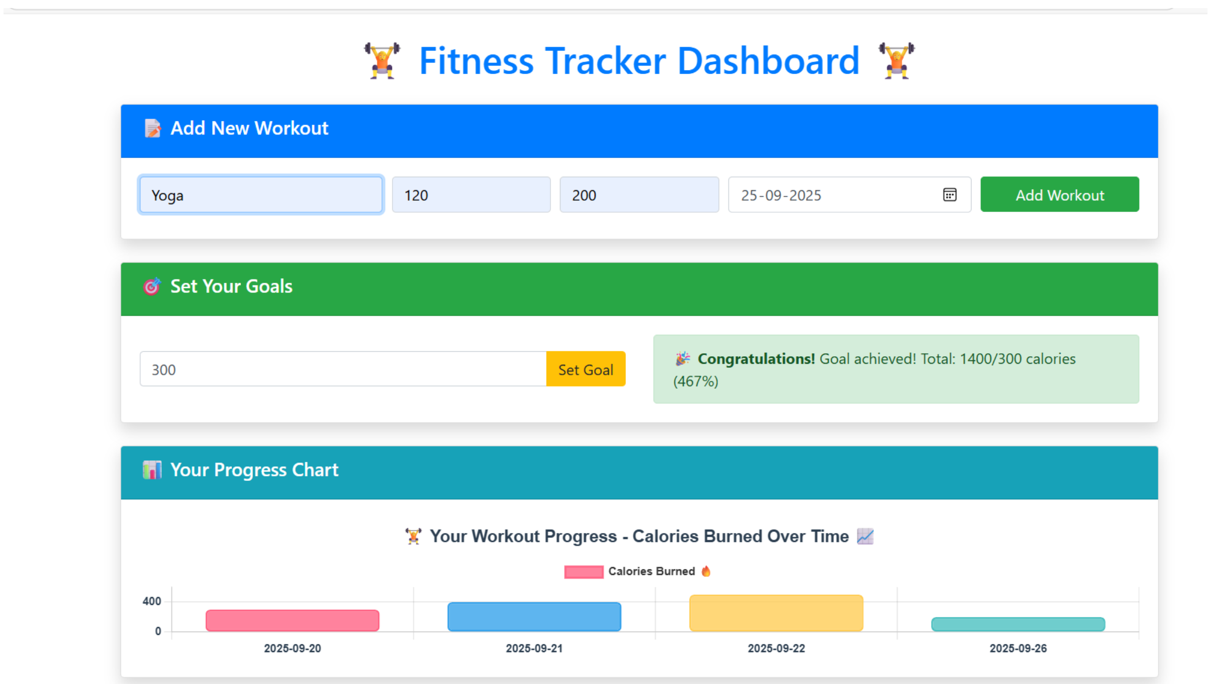
```css
body {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    font-family: 'Arial', sans-serif;
    min-height: 100vh;
}

.container {
    background: rgba(255, 255, 255, 0.95);
    border-radius: 15px;
    padding: 30px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.3);
}

h1 {
    font-weight: bold;
    text-shadow: 2px 2px 4px rgba(0,0,0,0.1);
}

.card {
    border: none;
    border-radius: 10px;
    transition: transform 0.3s ease;
}

.card:hover {
    transform: translateY(-5px);
}

.btn {
    border-radius: 20px;
    font-weight: bold;
}

.form-control {
    border-radius: 10px;
```

app.js



```javascript
let workouts = [
    { date: '2025-09-20', type: 'Running', duration: 30, calories: 300 },
    { date: '2025-09-21', type: 'Cycling', duration: 45, calories: 400 },
    { date: '2025-09-22', type: 'Swimming', duration: 60, calories: 500 },
];
let goal = 0;
let progressChart = null;

function updateWorkoutTable() {
    const tbody = $("#workoutTable tbody");
    tbody.empty();
    workouts.forEach((workout, index) => {
        tbody.append(`<tr>
            <td>${workout.date}</td>
            <td><span class="badge badge-primary">${workout.type}</span></td>
            <td>${workout.duration}</td>
            <td><span class="badge badge-danger">${workout.calories}</span></td>
            <td><button class="btn btn-sm btn-danger" onclick="deleteWorkout(${index})">Delete</button></td>
        </tr>`);
    });
}

function deleteWorkout(index) {
    workouts.splice(index, 1);
    updateWorkoutTable();
    updateProgressChart();
    checkGoalStatus();
}

function updateProgressChart() {
    const ctx = document.getElementById('progressChart').getContext('2d');
    const dates = workouts.map(w => w.date);
```

OUTPUT



🏋️ Fitness Tracker Dashboard 🏋️

📝 Add New Workout

Yoga | 120 | 200 | 25-09-2025 | Add Workout

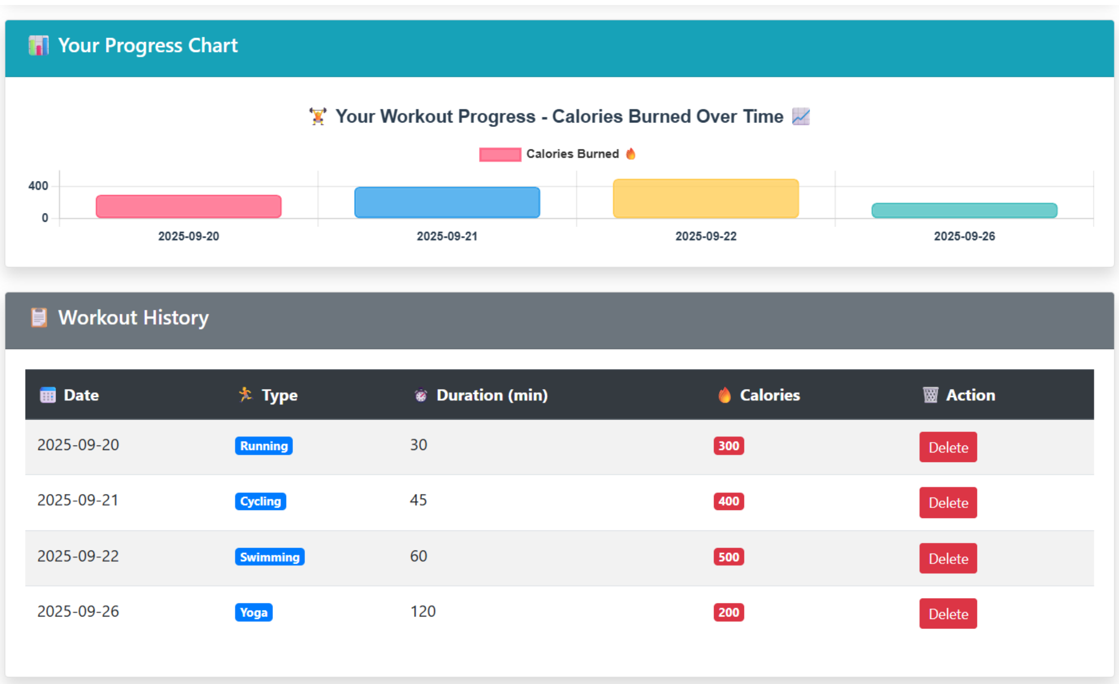
🎯 Set Your Goals

300 | Set Goal

🎉 **Congratulations!** Goal achieved! Total: 1400/300 calories (467%)

📊 Your Progress Chart

🏋️ Your Workout Progress - Calories Burned Over Time 📈



📊 Your Progress Chart

🏋️ Your Workout Progress - Calories Burned Over Time 📈

📋 Workout History

📅 Date	🏃 Type	⏱ Duration (min)	🔥 Calories	🗑 Action
2025-09-20	Running	30	300	Delete
2025-09-21	Cycling	45	400	Delete
2025-09-22	Swimming	60	500	Delete
2025-09-26	Yoga	120	200	Delete

13. Conclusion

This Fitness Tracker Dashboard successfully demonstrates how to build an interactive web application using front-end technologies. It provides users with a clear overview of their fitness progress and goal achievement. Through this project, I strengthened my skills in DOM manipulation, dynamic chart rendering, and responsive design principles.

14. References

- **Bootstrap Documentation: https://getbootstrap.com**
- **Chart.js Documentation: https://www.chartjs.org**
- **jQuery API: https://api.jquery.com**

Project 2:RECIPE BOOK

1. Abstract

This project aims to design and develop an interactive recipe book web application that allows users to browse, search, and filter recipes from various cuisines. The application features a responsive design with comprehensive recipe details including ingredients, preparation steps, and cooking information. Built using HTML5, CSS3, JavaScript, Bootstrap, and jQuery, the application provides an intuitive user experience with dynamic content filtering and modal-based recipe viewing.

2. Objectives

Clearly list the project goals:

- Create an interactive recipe browsing experience
- Implement search and filter functionality for recipes
- Ensure full responsiveness across all devices (mobile, tablet, desktop)
- Use proper HTML5 semantic structure
- Add interactive features using JavaScript and jQuery
- Create a visually appealing food-themed design
- Implement modal windows for recipe details

3. Scope of the Project

Explain what the project includes and any boundaries:

Included Features:

- Front-end design with interactive functionality
- Three cuisine categories (Italian, Mexican, Indian) with two recipes each
- Search by recipe name and description
- Filter by cuisine type and difficulty level
- Responsive design using Bootstrap framework

- Modal popups for detailed recipe viewing
- Works on desktop, tablet, and mobile devices

Boundaries:
- Static recipe data (no database integration)
- Front-end only (no backend services)
- No user authentication system
- No recipe submission functionality

4. Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Website structure and semantic markup
CSS3	Custom styling and responsive design
JavaScript	Interactive functionality and DOM manipulation
Bootstrap	Responsive framework and UI components
jQuery	Simplified DOM manipulation and event handling

5. HTML Structure Overview

- Used proper semantic tags: \`<header>\`, \`<nav>\`, \`<main>\`, \`<section>\`, \`<footer>\`
- Organized sections: Navigation, Header, Search/Filter, Recipe Grid, Modal
- Bootstrap grid system for responsive layout
- Accessible form elements and modal structure
- Clean separation of content and presentation

6. CSS Styling Strategy

- External CSS file with organized sections
- Bootstrap classes combined with custom styles
- Techniques used:
  - CSS Grid and Flexbox for layouts
  - Hover effects and transitions for interactivity
  - Color scheme focused on food-friendly tones (greens, warm colors)
  - Mobile-first responsive design approach
  - Custom animations for loading and interactions

7. Key Features

Feature	Description
Responsive Recipe Grid	Adaptive card layout that works on all devices
Advanced Filtering	Search by text and filter by multiple criteria
Interactive Modal System	Detailed recipe view without page refresh
Cuisine-based Organization	Recipes categorized by Italian, Mexican, Indian
Difficulty Indicators	Visual cues for recipe complexity
Preparation Information	Complete cooking times and serving sizes
Hover Effects	Interactive feedback on recipe cards
Mobile-Optimized Navigation	Collapsible menu for small screens

8. Challenges Faced & Solutions

Challenge	Solution
Dynamic content filtering	Used jQuery for efficient DOM manipulation
Responsive image handling	Implemented Bootstrap img-fluid classes
Modal content population	Created JavaScript function to dynamically load recipe data
Cross-browser compatibility	Used Bootstrap for consistent rendering

Mobile touch interactions	Added appropriate event handlers for touch devices
Performance optimization	Minimized DOM queries and used event delegation

9. Outcome

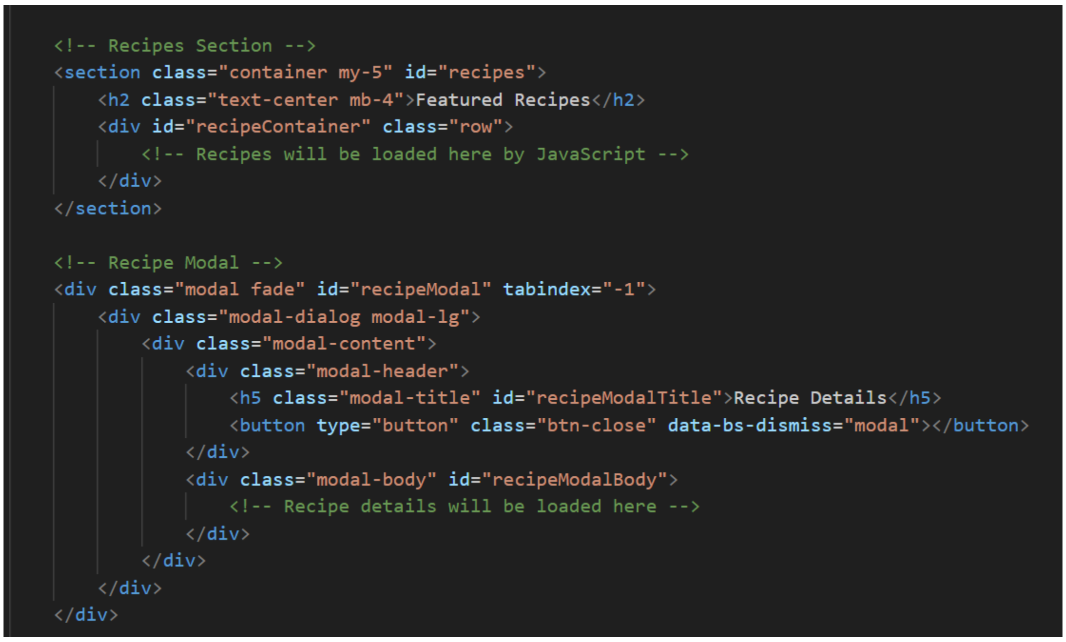
- Created a fully functional recipe book web application
- Successfully implemented interactive features using JavaScript and jQuery
- Achieved perfect responsiveness across all device sizes
- Delivered professional-looking food-themed design
- Demonstrated proficiency in multiple front-end technologies
- Created reusable code structure for future enhancements

10. Future Enhancements

- Connect to recipe API for dynamic content
- Implement user authentication and personal recipe collections
- Add recipe rating and review system
- Include nutrition information and calorie counts
- Implement shopping list functionality
- Add social sharing features
- Include video cooking tutorials
- Develop Progressive Web App (PWA) capabilities

11. Sample Code

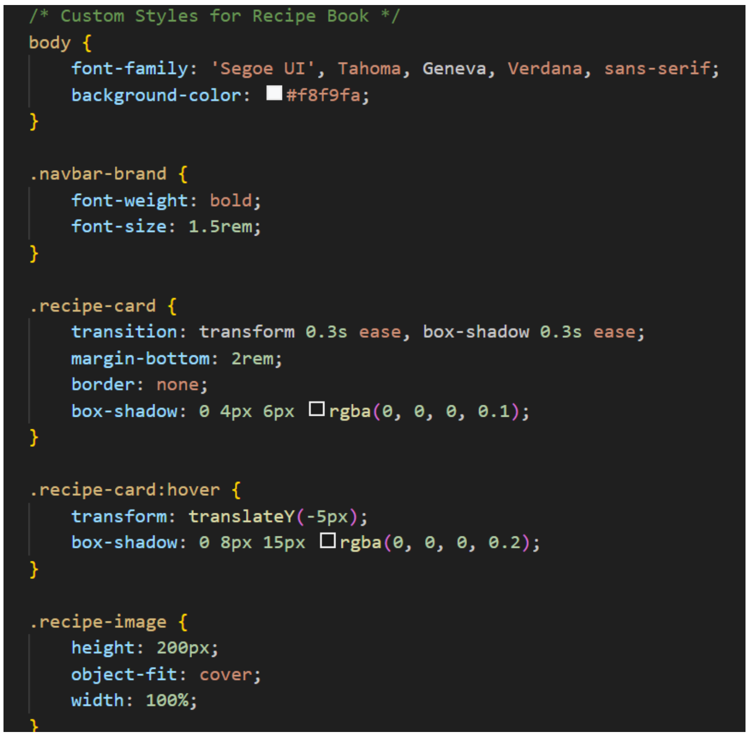
- **Index.html**



```html
<!-- Recipes Section -->
<section class="container my-5" id="recipes">
    <h2 class="text-center mb-4">Featured Recipes</h2>
    <div id="recipeContainer" class="row">
        <!-- Recipes will be loaded here by JavaScript -->
    </div>
</section>

<!-- Recipe Modal -->
<div class="modal fade" id="recipeModal" tabindex="-1">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="recipeModalTitle">Recipe Details</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
            </div>
            <div class="modal-body" id="recipeModalBody">
                <!-- Recipe details will be loaded here -->
            </div>
        </div>
    </div>
</div>
```

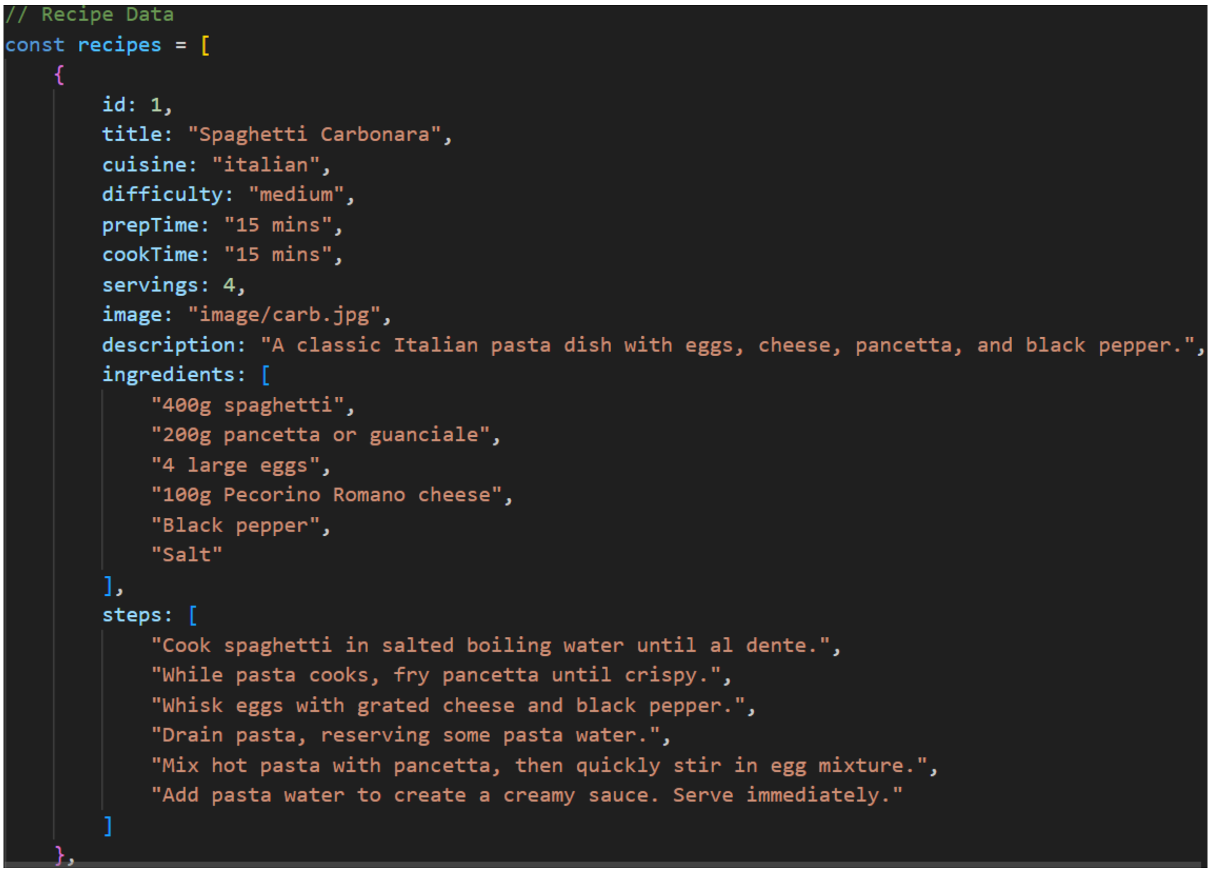
```html
<!-- Recipes Section -->
<section class="container my-5" id="recipes">
    <h2 class="text-center mb-4">Featured Recipes</h2>
    <div id="recipeContainer" class="row">
        <!-- Recipes will be loaded here by JavaScript -->
    </div>
</section>

<!-- Recipe Modal -->
<div class="modal fade" id="recipeModal" tabindex="-1">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="recipeModalTitle">Recipe Details</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
            </div>
            <div class="modal-body" id="recipeModalBody">
                <!-- Recipe details will be loaded here -->
            </div>
        </div>
    </div>
</div>
```

- **Styles.css**



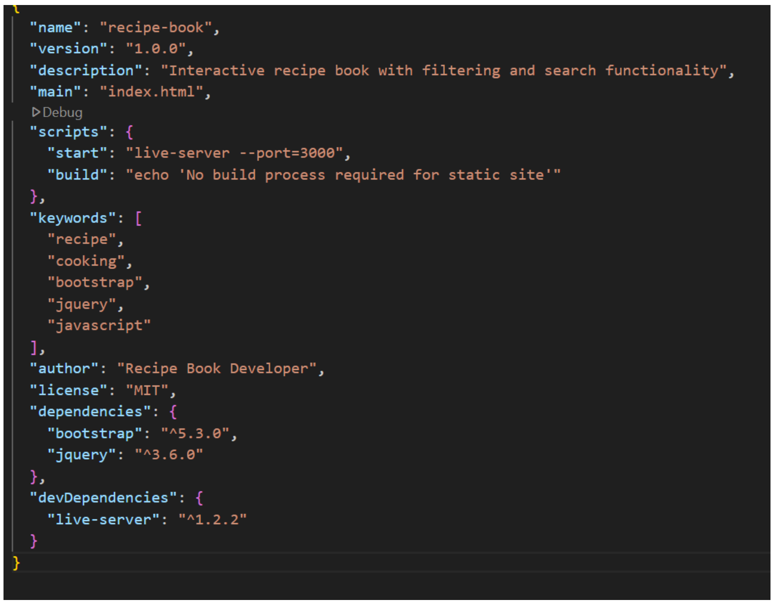
```css
/* Custom Styles for Recipe Book */
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #f8f9fa;
}

.navbar-brand {
    font-weight: bold;
    font-size: 1.5rem;
}

.recipe-card {
    transition: transform 0.3s ease, box-shadow 0.3s ease;
    margin-bottom: 2rem;
    border: none;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

.recipe-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 8px 15px rgba(0, 0, 0, 0.2);
}

.recipe-image {
    height: 200px;
    object-fit: cover;
    width: 100%;
}
```

- **Script.js**



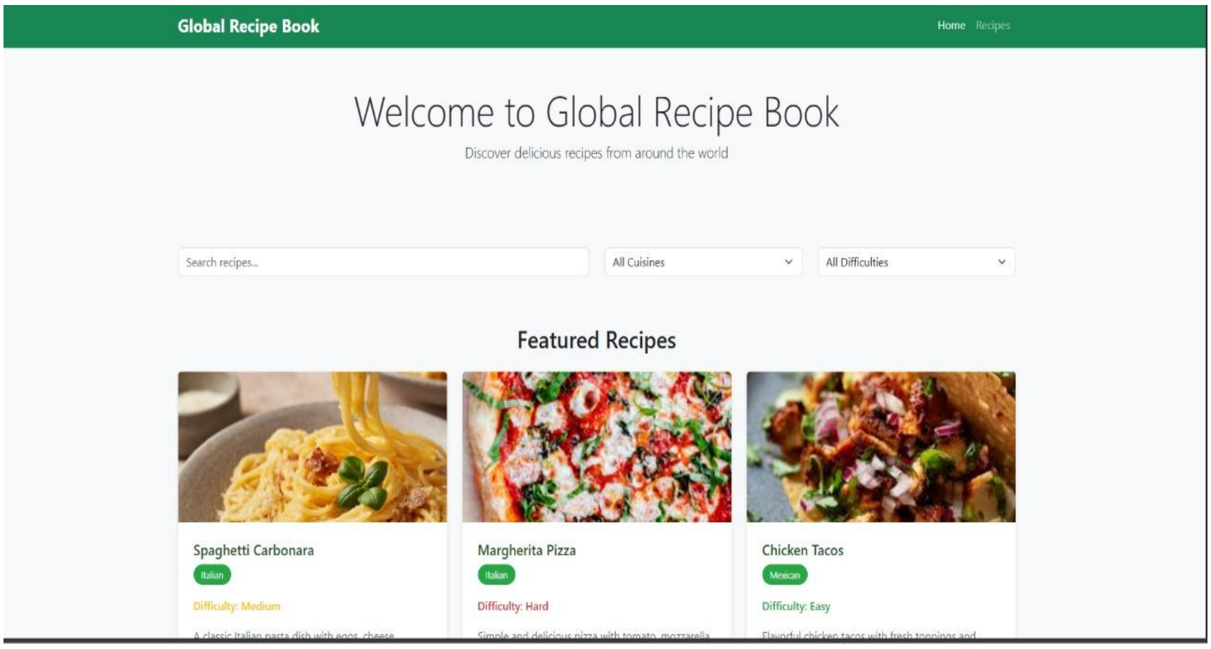
```javascript
// Recipe Data
const recipes = [
    {
        id: 1,
        title: "Spaghetti Carbonara",
        cuisine: "italian",
        difficulty: "medium",
        prepTime: "15 mins",
        cookTime: "15 mins",
        servings: 4,
        image: "image/carb.jpg",
        description: "A classic Italian pasta dish with eggs, cheese, pancetta, and black pepper.",
        ingredients: [
            "400g spaghetti",
            "200g pancetta or guanciale",
            "4 large eggs",
            "100g Pecorino Romano cheese",
            "Black pepper",
            "Salt"
        ],
        steps: [
            "Cook spaghetti in salted boiling water until al dente.",
            "While pasta cooks, fry pancetta until crispy.",
            "Whisk eggs with grated cheese and black pepper.",
            "Drain pasta, reserving some pasta water.",
            "Mix hot pasta with pancetta, then quickly stir in egg mixture.",
            "Add pasta water to create a creamy sauce. Serve immediately."
        ]
    },
```

- **Package.js**



```
{
  "name": "recipe-book",
  "version": "1.0.0",
  "description": "Interactive recipe book with filtering and search functionality",
  "main": "index.html",
  "scripts": {
    "start": "live-server --port=3000",
    "build": "echo 'No build process required for static site'"
  },
  "keywords": [
    "recipe",
    "cooking",
    "bootstrap",
    "jquery",
    "javascript"
  ],
  "author": "Recipe Book Developer",
  "license": "MIT",
  "dependencies": {
    "bootstrap": "^5.3.0",
    "jquery": "^3.6.0"
  },
  "devDependencies": {
    "live-server": "^1.2.2"
  }
}
```

OUTPUT



Welcome to Global Recipe Book

Discover delicious recipes from around the world

Featured Recipes

Spaghetti Carbonara

Difficulty: Medium

A classic Italian pasta dish with eggs, cheese

Margherita Pizza

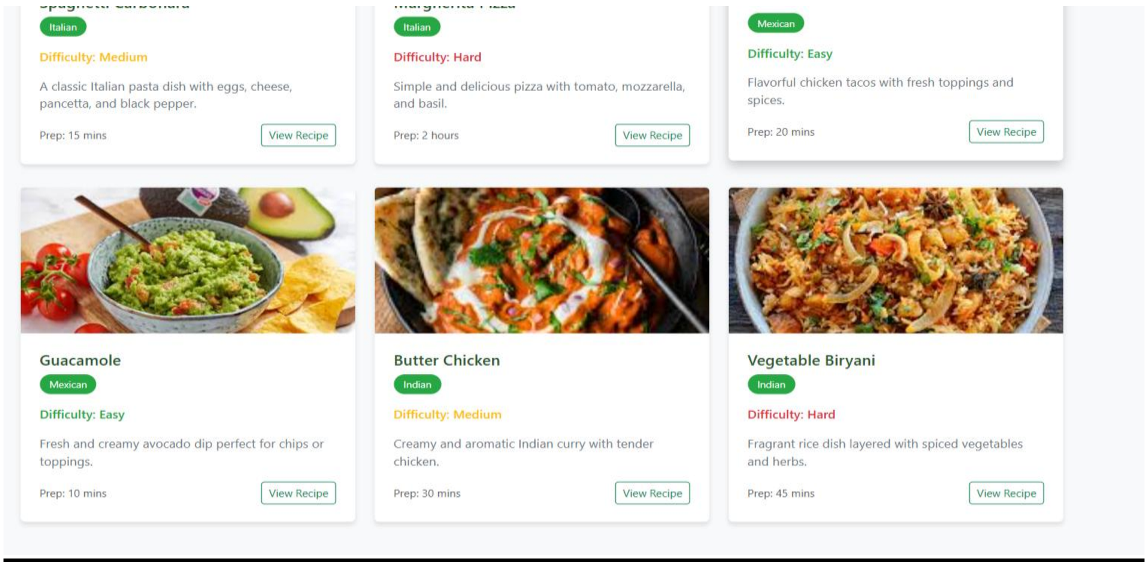
Difficulty: Hard

Simple and delicious pizza with tomato, mozzarella

Chicken Tacos

Difficulty: Easy

Flavorful chicken tacos with fresh toppings and



Italian

Difficulty: Medium

A classic Italian pasta dish with eggs, cheese, pancetta, and black pepper.

Prep: 15 mins

Italian

Difficulty: Hard

Simple and delicious pizza with tomato, mozzarella, and basil.

Prep: 2 hours

Mexican

Difficulty: Easy

Flavorful chicken tacos with fresh toppings and spices.

Prep: 20 mins

Guacamole

Mexican

Difficulty: Easy

Fresh and creamy avocado dip perfect for chips or toppings.

Prep: 10 mins

Butter Chicken

Indian

Difficulty: Medium

Creamy and aromatic Indian curry with tender chicken.

Prep: 30 mins

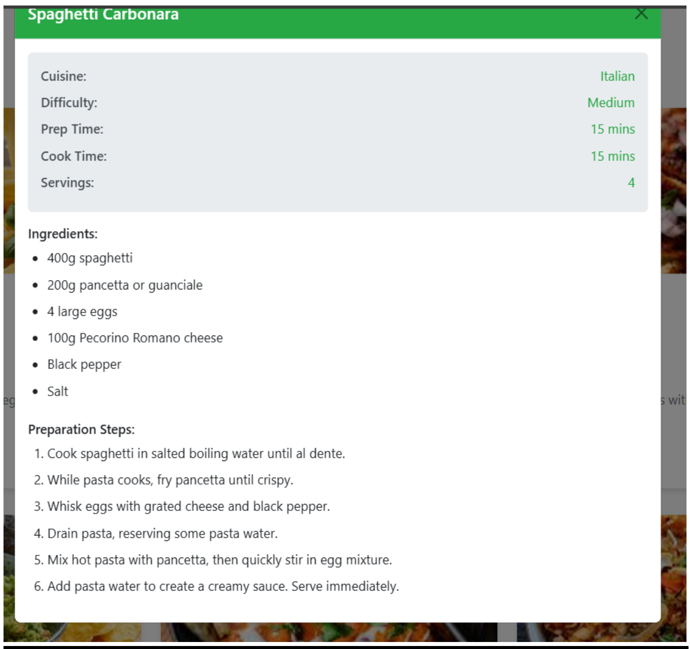
Vegetable Biryani

Indian

Difficulty: Hard

Fragrant rice dish layered with spiced vegetables and herbs.

Prep: 45 mins



Spaghetti Carbonara

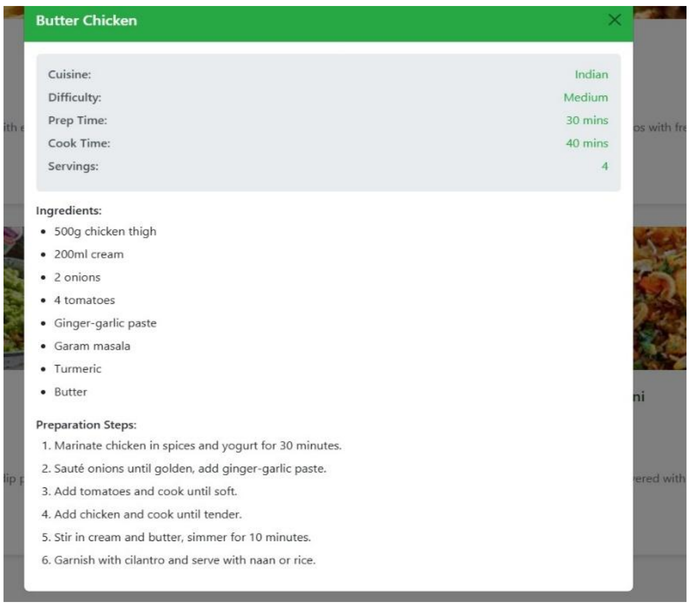
Cuisine:	Italian
Difficulty:	Medium
Prep Time:	15 mins
Cook Time:	15 mins
Servings:	4

Ingredients:

- 400g spaghetti
- 200g pancetta or guanciale
- 4 large eggs
- 100g Pecorino Romano cheese
- Black pepper
- Salt

Preparation Steps:

1. Cook spaghetti in salted boiling water until al dente.
2. While pasta cooks, fry pancetta until crispy.
3. Whisk eggs with grated cheese and black pepper.
4. Drain pasta, reserving some pasta water.
5. Mix hot pasta with pancetta, then quickly stir in egg mixture.
6. Add pasta water to create a creamy sauce. Serve immediately.



Butter Chicken

Cuisine:	Indian
Difficulty:	Medium
Prep Time:	30 mins
Cook Time:	40 mins
Servings:	4

Ingredients:

- 500g chicken thigh
- 200ml cream
- 2 onions
- 4 tomatoes
- Ginger-garlic paste
- Garam masala
- Turmeric
- Butter

Preparation Steps:

1. Marinate chicken in spices and yogurt for 30 minutes.
2. Sauté onions until golden, add ginger-garlic paste.
3. Add tomatoes and cook until soft.
4. Add chicken and cook until tender.
5. Stir in cream and butter, simmer for 10 minutes.
6. Garnish with cilantro and serve with naan or rice.

14. Conclusion

The Global Recipe Book project successfully demonstrates the integration of multiple front-end technologies to create an interactive and user-friendly web application. This project represents a significant advancement from previous static websites by incorporating dynamic functionality through JavaScript and jQuery.

Key learning outcomes include:

- Mastering JavaScript for interactive web applications
- Implementing complex filtering and search functionality
- Integrating Bootstrap framework for responsive design
- Using jQuery to simplify DOM manipulation
- Creating modal-based user interfaces
- Handling dynamic content rendering
- Implementing mobile-first responsive strategies

This project showcases the ability to combine HTML5, CSS3, JavaScript, Bootstrap, and jQuery to create professional-grade web applications that provide excellent user experiences across all devices.

15. References

- Bootstrap Documentation: https://getbootstrap.com
- jQuery Documentation: https://api.jquery.com
- L&T LMS: https://learn.lntedutech.com/Landing/MyCourse