



**Visvesvaraya Technological University**

**BELAGAVI, KARNATAKA**

**ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ**  
**ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ**

**Full Stack Development Project report on**

**“Nutrition Management System”**

**Submitted by**

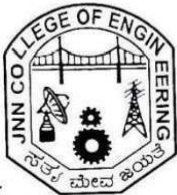
- |                       |                   |
|-----------------------|-------------------|
| <b>1. Tarun D R</b>   | <b>4JN22IS183</b> |
| <b>2. Veeresh G J</b> | <b>4JN22IS179</b> |

**Under the guidance of**

**Mr. Arun Kumar** B.E, MTech

Associate Professor, Dept. of IS&E,

JNNCE, Shivamogga.



**Department of Information Science & Engineering**

**JNN College of Engineering**

**Shivamogga-577204**

**2024-25**

## TABLE OF CONTENTS

SL NO	TITLES	PAGE NO
01	Introduction	02
02	Problem Statement	03
03	Objective of the Project	03
04	Tools & Technologies Used	04
05	System Requirements	05
06	System Design	05
07	Screenshots of Web Page	06
08	Project Code	09
09	Conclusion	18

## **1. Introduction**

In the modern world, where sedentary lifestyles and processed food have become the norm, managing personal health and nutrition is more important than ever. Nutritional imbalances often lead to long-term health issues such as obesity, diabetes, hypertension, and vitamin deficiencies. While awareness around healthy eating is growing, many individuals lack the tools to accurately monitor and maintain their dietary habits. This has led to a rising demand for digital platforms that support personalized nutrition tracking and guidance.

The Nutrition Management System is a user-friendly web-based application developed using React.js, designed to help users track their daily food intake, calculate the nutritional value of meals, and monitor progress toward dietary goals. It allows users to register, log meals, and view daily/weekly nutritional summaries. The system emphasizes ease of use, real-time feedback, and insightful analytics on macro- and micronutrient consumption.

### **1.1 Overview of the Banking System**

This project simulates a basic yet functional nutrition tracking platform that can be used by individuals, dieticians, and fitness enthusiasts. The application is built with modern web technologies that enable responsive interfaces and interactive data handling. It includes components for user registration, meal entry, nutrition calculations, and dashboard visualization. The system stores data locally, allowing users to revisit their entries and gain long-term insights into their eating habits. In the future, such a system could be expanded with machine learning for personalized meal recommendations, integration with wearable health devices, and backend support for multi-user management or data syncing across devices.

### **1.2 Motivation (Why this System is Useful in Real Life)**

The motivation for developing this system stems from real-world health challenges and the lack of simple digital tools for everyday nutrition tracking. While advanced apps exist, they are often cluttered with features or locked behind pay walls.

This system focuses on simplicity, usability, and educational value. It is especially beneficial for:

- Students learning about health tech and frontend development.
- Individuals aiming to manage weight, build muscle, or control dietary conditions.
- Trainers or dieticians who want to demonstrate nutritional planning.

The project bridges health awareness and digital literacy, making it both socially relevant and technically enriching.

## **2. Problem Statement**

In today's fast-paced world, many individuals struggle to maintain a balanced and nutritious diet. People often rely on memory or informal, inconsistent methods—such as jotting notes on paper, recalling meals mentally, or loosely following diet plans—to track what they eat. These approaches are prone to inaccuracy and make it difficult to understand the true nutritional intake over time. As a result, poor dietary habits go unchecked, contributing to a rise in preventable health issues such as obesity, fatigue, malnutrition, and lifestyle diseases.

Despite the availability of advanced nutrition tracking apps, many are overly complex, require subscriptions, or lack user-friendly interfaces, especially for beginners.

## **3. Objective of the Project**

- To create a simple Nutrition Management System using React.js with intuitive navigation.
- To allow users to register and log in securely.
- To enable users to log daily meals and get a breakdown of nutritional values.
- To educate students on frontend development using modern tools and concepts.

## **4. Tools & Technologies Used**

### **1. Frontend: React.js (JavaScript):**

React.js is a popular JavaScript library for building user interfaces using reusable components. It helps in creating a dynamic, single-page web application efficiently. Its virtual DOM improves performance and responsiveness of the UI. React is ideal for developing modern web apps with interactive features.

### **2. Routing: React Router DOM**

React Router DOM is used to handle navigation between different pages in a React app. It allows the app to switch views without reloading the entire page. Routes help define user paths like registration, login, and dashboard. This enables a smooth user experience in single-page applications.

### **3. Data Storage: Local Storage**

Local Storage is a browser feature used to store data locally on the user's machine. In this project, it stores user account details and balance without a backend. Data remains even after the browser is refreshed or closed. It's simple to use for small-scale applications like prototypes or demos.

### **4. IDE: Visual Studio Code**

VS Code is a lightweight and powerful code editor for web development. It supports JavaScript and React with features like IntelliSense and debugging. Extensions like ESLint, Prettier, and React snippets improve productivity. Its terminal integration and Git support make it developer-friendly.

### **5. Browser: Google Chrome / Mozilla Firefox**

Modern browsers like Chrome and Firefox are used to test and run the web application. They support advanced JavaScript features and developer tools. Built-in consoles help

debug and monitor app behavior easily. They ensure the app works smoothly across different environments.

## 5. System Requirements

To run this Bank Account Management System, certain software requirements must be met. The system requires **Node.js** and **npm (Node Package Manager)**, which are essential for running and managing the React.js environment. Additionally, a modern web browser like Google Chrome or Mozilla Firefox is needed to access and interact with the application smoothly.

On the hardware side, the system should run on a machine with a minimum of **4 GB RAM** to ensure smooth performance. A **processor of Intel i3** or above is recommended to handle the development and execution of the web application efficiently. These basic hardware specifications are sufficient for small-scale web projects built using React.js.

## 6. System Design

The Nutrition Management System is designed using a client-side architecture developed entirely with React.js. The system adopts a modular, component-based structure, ensuring clarity, reusability, and scalability. All user interactions and data processing occur within the browser, and local storage is utilized to temporarily hold user inputs such as personal information, BMI, daily calorie requirements, and nutritional recommendations. This design makes the application lightweight and responsive, ideal for small-scale use cases or academic demonstrations without requiring backend integration or database support.

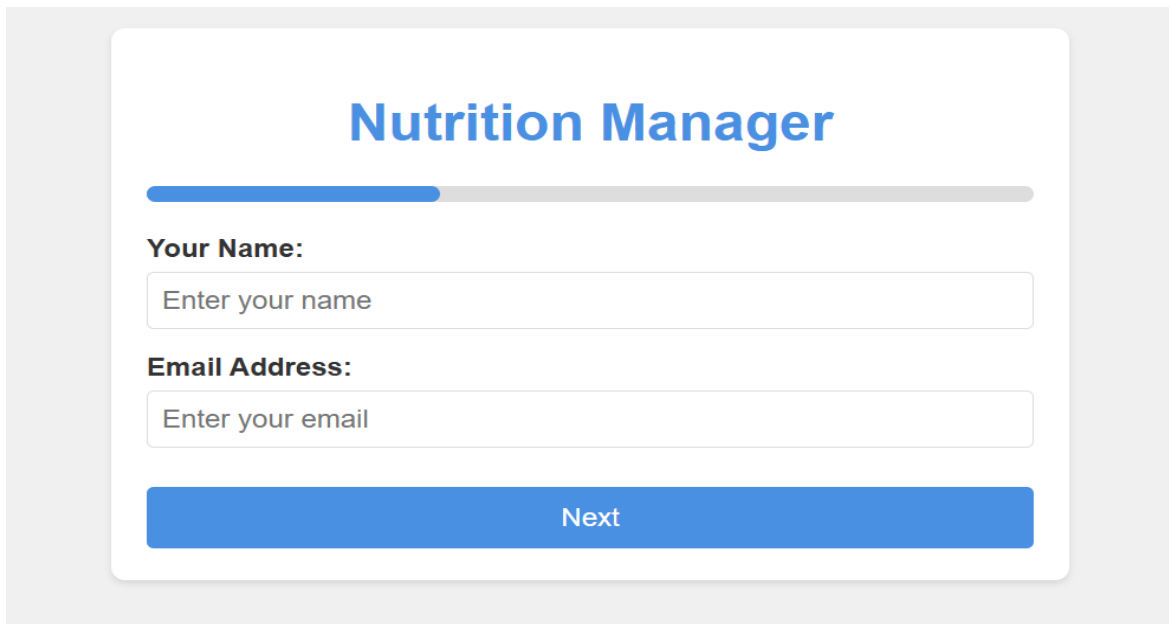
The system begins with an introductory page where the user is prompted to enter their name and email address. Once this basic information is captured, the user is guided through a step-by-step form to input details such as age, gender, height, weight, physical activity level, and their health goal—such as gaining, losing, or maintaining weight. Based on this information, the application calculates the user's Body Mass Index (BMI) and daily calorie target using standard nutritional formulas. It also determines a macronutrient breakdown in terms of proteins, carbohydrates, and fats based on the selected fitness goal.

After the calculations are complete, the user is presented with a dynamic dashboard. This dashboard summarizes the user's personal details, daily calorie target, macronutrient distribution, and customized recommendations. The recommendations section provides actionable advice tailored to the user's goal—for example, suggesting frequent high-calorie meals for weight gain or calorie control tips for weight loss. These suggestions include food choices like nuts, dairy, fatty fish, or meal timing strategies such as incorporating smoothness between meals.

The system also offers a "Start Over" option, allowing the user to reset the session and re-enter data, facilitating experimentation and re-evaluation without backend dependencies. The entire application is split into logical components—such as Intro, User Details, Nutrition Plan, and Dashboard—each responsible for a specific function within the system. This structure not only simplifies the development and debugging process but also makes the project easily extendable in the future.

## 7. Screenshots of Web Page

### 7.1 Home page of nutrition Management System



**Nutrition Manager**

Progress bar: [Blue segment] [Grey segment]

**Your Name:**

**Email Address:**

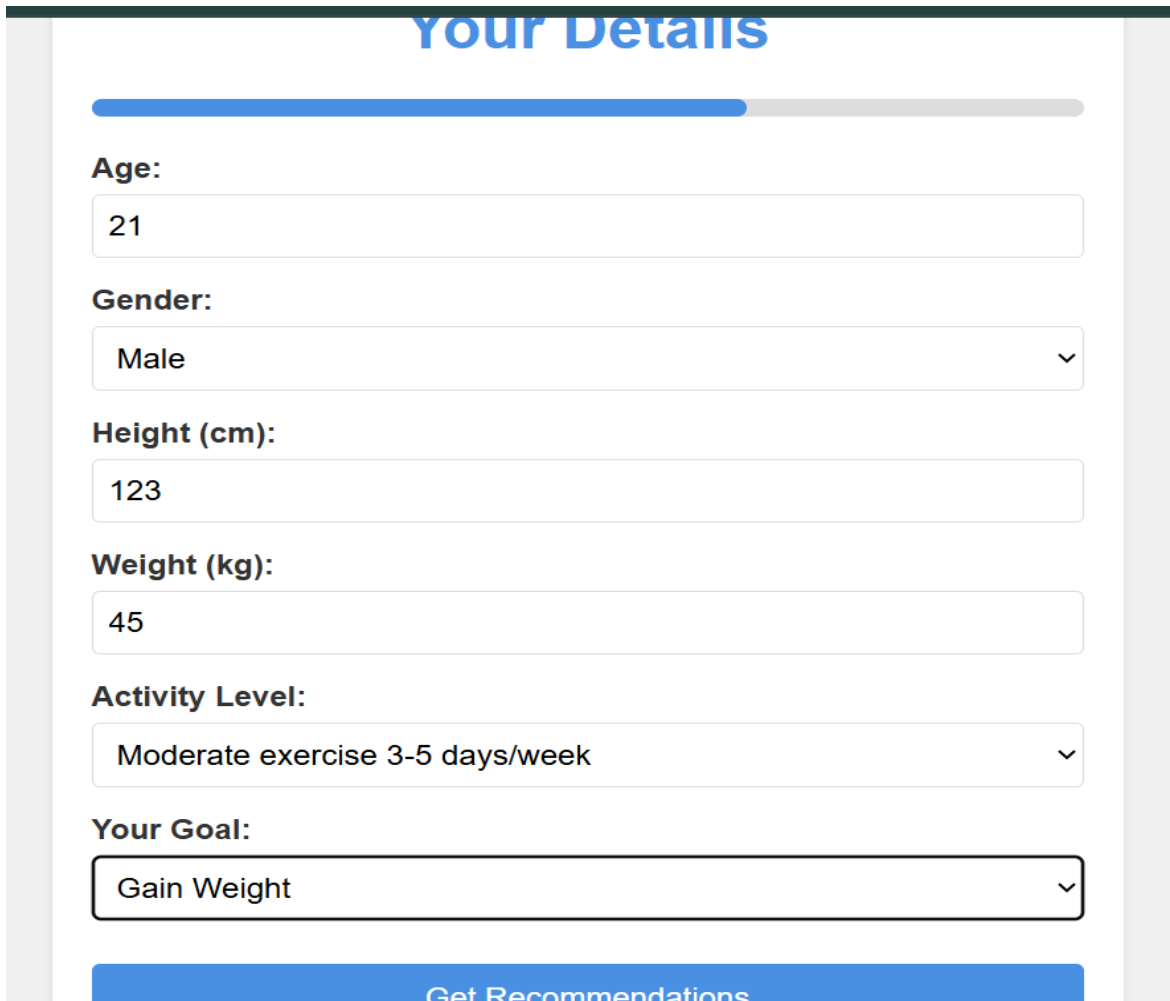
**Next**

**Figure 1: Home page of Nutrition Management System**

The Home page provides navigation links for account registration and user login. It serves as the entry point to the application.

### 7.1 Account Registration Form

This form allows the bank manager to input customer details and generate a unique account number. It collects all necessary information for account creation.



The screenshot shows a web form titled "Your Details" in blue text. Below the title is a progress bar with a blue segment on the left and a grey segment on the right. The form contains several input fields and dropdown menus, each with a label above it: "Age:" with a text input containing "21"; "Gender:" with a dropdown menu showing "Male" and a downward arrow; "Height (cm):" with a text input containing "123"; "Weight (kg):" with a text input containing "45"; "Activity Level:" with a dropdown menu showing "Moderate exercise 3-5 days/week" and a downward arrow; and "Your Goal:" with a dropdown menu showing "Gain Weight" and a downward arrow. At the bottom of the form is a blue button with the text "Get Recommendations" in white.

**Figure 2: Account Registration Form**

### 7.3 Login Page:

Users enter their account number and password here to access their account securely. It validates credentials before granting access.



## 7.4 Dashboard:

The dashboard shows the user's account information including the current balance. It also provides options for crediting and debiting the account.

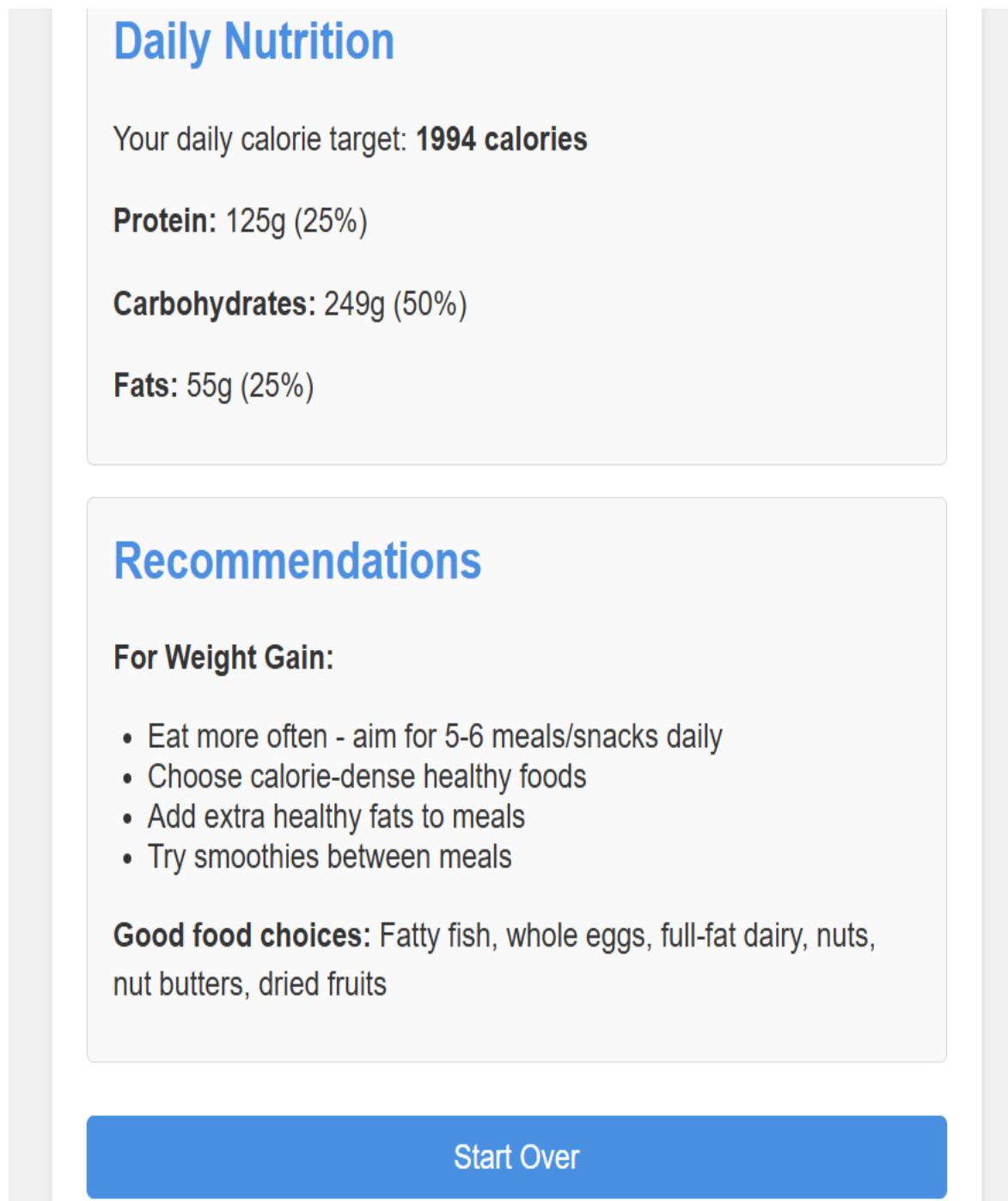


Figure 3: User Dash Board

## 8. Project Code:

### 8.1 Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Nutrition Manager</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <!-- Login Page -->
    <div id="login-page" class="page active">
      <h1>Nutrition Manager</h1>
      <div class="progress-bar">
        <div class="progress" style="width: 33%"></div>
      </div>

      <div class="form-group">
        <label for="name">Your Name:</label>
        <input type="text" id="name" placeholder="Enter your name">
      </div>

      <div class="form-group">
        <label for="email">Email Address:</label>
        <input type="email" id="email" placeholder="Enter your
email">
      </div>

      <button id="login-btn">Next</button>
    </div>

    <!-- Profile Page -->
    <div id="profile-page" class="page">
      <h1>Your Details</h1>
      <div class="progress-bar">
        <div class="progress" style="width: 66%"></div>
      </div>

      <div class="form-group">
        <label for="age">Age:</label>

```

```
        <input type="number" id="age" placeholder="Your age in
years">
    </div>

    <div class="form-group">
        <label for="gender">Gender:</label>
        <select id="gender">
            <option value="male">Male</option>
            <option value="female">Female</option>
            <option value="other">Other</option>
        </select>
    </div>

    <div class="form-group">
        <label for="height">Height (cm):</label>
        <input type="number" id="height" placeholder="Your height in
centimeters">
    </div>

    <div class="form-group">
        <label for="weight">Weight (kg):</label>
        <input type="number" id="weight" placeholder="Your weight in
kilograms">
    </div>

    <div class="form-group">
        <label for="activity">Activity Level:</label>
        <select id="activity">
            <option value="">Select your activity level</option>
            <option value="sedentary">Mostly inactive or
sitting</option>
            <option value="light">Light exercise 1-3
days/week</option>
            <option value="moderate">Moderate exercise 3-5
days/week</option>
            <option value="active">Hard exercise 6-7
days/week</option>
            <option value="very_active">Very hard daily exercise or
physical job</option>
        </select>
    </div>

    <div class="form-group">
        <label for="goal">Your Goal:</label>
        <select id="goal">
            <option value="">Select your goal</option>
```

```
        <option value="weight_loss">Lose Weight</option>
        <option value="maintenance">Maintain Weight</option>
        <option value="weight_gain">Gain Weight</option>
        <option value="muscle_gain">Build Muscle</option>
        <option value="health">Improve Overall Health</option>
    </select>
</div>

    <button id="profile-btn">Get Recommendations</button>
</div>

<!-- Results Page -->
<div id="results-page" class="page">
    <h1>Your Nutrition Plan</h1>
    <div class="progress-bar">
        <div class="progress" style="width: 100%"></div>
    </div>

    <div class="result-box">
        <h2>Your Details</h2>
        <div id="profile-summary"></div>
    </div>

    <div class="result-box">
        <h2>Daily Nutrition</h2>
        <div id="calories-result"></div>
        <div id="macros-result"></div>
    </div>

    <div class="result-box">
        <h2>Recommendations</h2>
        <div id="recommendations"></div>
    </div>

    <button id="start-over-btn">Start Over</button>
</div>

</div>

<script src="script.js"></script>
</body>
</html>
```

## 8.2 Script.js code

```
// Navigation between pages
document.getElementById('login-btn').addEventListener('click', function() {
    const name = document.getElementById('name').value;
    const email = document.getElementById('email').value;

    if (name && email) {
        document.getElementById('login-page').classList.remove('active');
        document.getElementById('profile-page').classList.add('active');
    } else {
        alert('Please enter your name and email');
    }
});

document.getElementById('profile-btn').addEventListener('click', function()
{
    const age = document.getElementById('age').value;
    const gender = document.getElementById('gender').value;
    const height = document.getElementById('height').value;
    const weight = document.getElementById('weight').value;
    const activity = document.getElementById('activity').value;
    const goal = document.getElementById('goal').value;

    if (age && gender && height && weight && activity && goal) {
        calculateRecommendations();
        document.getElementById('profile-page').classList.remove('active');
        document.getElementById('results-page').classList.add('active');
    } else {
        alert('Please fill in all the fields');
    }
});

document.getElementById('start-over-btn').addEventListener('click',
function() {
    document.getElementById('results-page').classList.remove('active');
    document.getElementById('login-page').classList.add('active');

    // Clear form fields
    document.getElementById('name').value = '';
    document.getElementById('email').value = '';
    document.getElementById('age').value = '';
    document.getElementById('gender').value = '';
    document.getElementById('height').value = '';
    document.getElementById('weight').value = '';
    document.getElementById('activity').value = '';
```

```
document.getElementById('goal').value = '';
});

// Calculate nutrition recommendations
function calculateRecommendations() {
  const name = document.getElementById('name').value;
  const age = parseInt(document.getElementById('age').value);
  const gender = document.getElementById('gender').value;
  const height = parseInt(document.getElementById('height').value);
  const weight = parseInt(document.getElementById('weight').value);
  const activity = document.getElementById('activity').value;
  const goal = document.getElementById('goal').value;

  // Display user profile summary
  const profileSummary = document.getElementById('profile-summary');
  profileSummary.innerHTML = `
    <p><strong>Name:</strong> ${name}</p>
    <p><strong>Age:</strong> ${age} years</p>
    <p><strong>Gender:</strong> ${gender.charAt(0).toUpperCase() +
gender.slice(1)}</p>
    <p><strong>Height:</strong> ${height} cm</p>
    <p><strong>Weight:</strong> ${weight} kg</p>
    <p><strong>BMI:</strong> ${calculateBMI(height,
weight).toFixed(1)}</p>
  `;

  // Calculate BMR (Basal Metabolic Rate)
  let bmr;
  if (gender === 'male') {
    bmr = 10 * weight + 6.25 * height - 5 * age + 5;
  } else {
    bmr = 10 * weight + 6.25 * height - 5 * age - 161;
  }

  // Calculate TDEE (Total Daily Energy Expenditure)
  const activityFactors = {
    'sedentary': 1.2,
    'light': 1.375,
    'moderate': 1.55,
    'active': 1.725,
    'very_active': 1.9
  };

  const tdee = Math.round(bmr * activityFactors[activity]);

  // Adjust calories based on goal
```

```
let calorieGoal;
let proteinRatio, carbRatio, fatRatio;

switch(goal) {
  case 'weight_loss':
    calorieGoal = Math.round(tdee * 0.8);
    proteinRatio = 0.3;
    carbRatio = 0.4;
    fatRatio = 0.3;
    break;
  case 'maintenance':
    calorieGoal = tdee;
    proteinRatio = 0.25;
    carbRatio = 0.5;
    fatRatio = 0.25;
    break;
  case 'weight_gain':
    calorieGoal = Math.round(tdee * 1.15);
    proteinRatio = 0.25;
    carbRatio = 0.5;
    fatRatio = 0.25;
    break;
  case 'muscle_gain':
    calorieGoal = Math.round(tdee * 1.1);
    proteinRatio = 0.35;
    carbRatio = 0.45;
    fatRatio = 0.2;
    break;
  case 'health':
    calorieGoal = tdee;
    proteinRatio = 0.25;
    carbRatio = 0.45;
    fatRatio = 0.3;
    break;
  default:
    calorieGoal = tdee;
    proteinRatio = 0.25;
    carbRatio = 0.5;
    fatRatio = 0.25;
}

// Calculate macronutrients in grams
const proteinGrams = Math.round((calorieGoal * proteinRatio) / 4);
const carbGrams = Math.round((calorieGoal * carbRatio) / 4);
const fatGrams = Math.round((calorieGoal * fatRatio) / 9);
```

```
// Display calorie and macro results
document.getElementById('calories-result').innerHTML = `
    <p>Your daily calorie target: <strong>${calorieGoal}
calories</strong></p>
`;

document.getElementById('macros-result').innerHTML = `
    <p><strong>Protein:</strong> ${proteinGrams}g
(${Math.round(proteinRatio * 100)}%)</p>
    <p><strong>Carbohydrates:</strong> ${carbGrams}g
(${Math.round(carbRatio * 100)}%)</p>
    <p><strong>Fats:</strong> ${fatGrams}g (${Math.round(fatRatio *
100)}%)</p>
`;

// Generate recommendations based on goal
const recommendations = document.getElementById('recommendations');
let recommendationText = '';

switch(goal) {
    case 'weight_loss':
        recommendationText = `
            <p><strong>For Weight Loss:</strong></p>
            <ul>
                <li>Eat more protein to stay full longer</li>
                <li>Choose vegetables, beans, and whole grains</li>
                <li>Include small amounts of healthy fats</li>
                <li>Drink plenty of water</li>
            </ul>
            <p><strong>Good food choices:</strong> Chicken, fish, eggs,
yogurt, vegetables, fruits, nuts</p>
        `;
        break;

    case 'maintenance':
        recommendationText = `
            <p><strong>For Maintaining Weight:</strong></p>
            <ul>
                <li>Eat balanced meals with protein, carbs, and healthy
fats</li>
                <li>Focus on whole foods</li>
                <li>Watch portion sizes</li>
                <li>Stay active</li>
            </ul>
            <p><strong>Good food choices:</strong> Variety of meats,
fish, eggs, dairy, fruits, vegetables, whole grains</p>
        `;
        break;
}
```



```
`;  
break;  
  
case 'weight_gain':  
    recommendationText = `  
        <p><strong>For Weight Gain:</strong></p>  
        <ul>  
            <li>Eat more often - aim for 5-6 meals/snacks daily</li>  
            <li>Choose calorie-dense healthy foods</li>  
            <li>Add extra healthy fats to meals</li>  
            <li>Try smoothies between meals</li>  
        </ul>  
        <p><strong>Good food choices:</strong> Fatty fish, whole  
eggs, full-fat dairy, nuts, nut butters, dried fruits</p>  
    `;  
    break;  
  
case 'muscle_gain':  
    recommendationText = `  
        <p><strong>For Muscle Gain:</strong></p>  
        <ul>  
            <li>Eat more protein throughout the day</li>  
            <li>Eat carbs before and after workouts</li>  
            <li>Have protein within 30-60 minutes after  
exercise</li>  
            <li>Drink plenty of water</li>  
        </ul>  
        <p><strong>Good food choices:</strong> Chicken, lean beef,  
eggs, fish, rice, oats, sweet potatoes</p>  
    `;  
    break;  
  
case 'health':  
    recommendationText = `  
        <p><strong>For Better Health:</strong></p>  
        <ul>  
            <li>Eat mostly whole, unprocessed foods</li>  
            <li>Aim for 5+ servings of vegetables and fruits  
daily</li>  
            <li>Choose a variety of colorful foods</li>  
            <li>Limit added sugar and processed foods</li>  
        </ul>  
        <p><strong>Good food choices:</strong> Vegetables, fruits,  
fish, eggs, nuts, beans, whole grains</p>  
    `;  
    break;
```

```
    default:
      recommendationText = `
        <p><strong>General Tips:</strong></p>
        <ul>
          <li>Eat whole, nutrient-rich foods</li>
          <li>Drink plenty of water</li>
          <li>Include protein, carbs, and healthy fats in
meals</li>
          <li>Be consistent with your eating habits</li>
        </ul>
        <p><strong>Good food choices:</strong> Lean meats, fish,
eggs, dairy, vegetables, fruits, whole grains, nuts</p>
      `;
    }

    recommendations.innerHTML = recommendationText;
  }

function calculateBMI(height, weight) {
  // Convert height from cm to m
  const heightInMeters = height / 100;
  // BMI formula: weight (kg) / height2 (m2)
  return weight / (heightInMeters * heightInMeters);
}
```

## 9. Conclusion

This project presents a basic yet practical Nutrition Management System developed using React.js. It allows users to enter personal details such as age, gender, height, weight, and fitness goals to receive personalized dietary recommendations. The system calculates important metrics like BMI and daily calorie needs, and provides a macronutrient breakdown tailored to the user's objective—whether it's weight gain, weight loss, or maintenance. React Router is used for smooth navigation between steps, and local storage is implemented for storing user inputs temporarily, enabling a seamless and responsive experience without requiring a backend.

Overall, this project offers students hands-on exposure to real-world health tech applications using modern web technologies. It bridges the gap between technology and wellness, and serves as an educational tool in both web development and basic nutrition awareness.