MEAL PLANNER
AND NUTRITION TRACKER

A

Mini Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOG

By

Amruta Kajuluri 1602-23-737-006

Dharanikota Sanjana 1602-23-737-044

Department of Information Technology

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE) Ibrahimbagh, Hyderabad-31

2023

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and

Approved by AICTE) Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We, *Amruta Kajuluri*, and *D.Sanjana,* bearing hall ticket numbers, 1602-23-737-006 and 1602-23-737-044, hereby declare that the project report entitled *Meal Planner and Nutrition Tracker* is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Amruta Kajuluri

1602-23-737-006

Dharanikota Sanjana

1602-23-737-044

(Faculty In-Charge) (Head,Dept of IT)

ACKNOWLEDGMENT

We extend our heartfelt gratitude to **Dr. S. V. Ramana**, Principal, Vasavi College of Engineering, for his continued support and encouragement throughout the duration of this project.

We would like to express our sincere thanks to **SATYADEVI,** Department of Information Technology, Vasavi College of Engineering, for introducing the Mini Project module into the curriculum, which gave us the opportunity to apply our academic learning in a practical and meaningful way.

Our special thanks go to our mini project coordinators, **SATYADEVI** and for their unwavering guidance, valuable suggestions, and continuous motivation throughout every phase of the project development.

We would also like to sincerely thank the project reviewers for their insightful feedback and constructive criticism, which significantly improved the quality and scope of our work.

ABSTRACT

In today's fast-paced world, maintaining a balanced diet and managing nutrition have become increasingly important. To address this need, **Meal Planner & Nutrition Tracker** has been developed as a full-stack web application aimed at simplifying meal planning and promoting healthier eating habits.

The application utilizes layered architecture based on modern web technologies. The front end is built using HTML, CSS, JavaScript, and Bootstrap, providing users with an intuitive and responsive interface for creating meal plans, tracking nutrient intake, managing shopping lists, and monitoring dietary goals. The backend is implemented using Node.js and MongoDB, ensuring efficient handling of user data, secure authentication, and seamless real-time updates.

Key functionalities include automated food analysis through external APIs, personalized meal recommendations, nutrient tracking charts, and a smart shopping list generator. The system also incorporates user authentication, data validation, and role-based access control to enhance security and usability.

This project demonstrates the application of full-stack development principles in building scalable, secure, and user-centric health management systems. Through this work, practical experience was gained in software architecture design, RESTful API development, database management, and frontend-backend integration.

TABLE OF CONTENTS

Introduction

The growing awareness of healthy living and balanced nutrition has led to a demand for digital tools that can help individuals manage their dietary habits effectively. Meal planning and nutrient tracking applications have become an essential part of modern health management, enabling users to plan their meals, monitor nutritional intake, and make informed decisions about their diet. In this context, the **Meal Planner & Nutrition Tracker** project aims to design and implement a full-stack web application that supports meal planning, nutrition analysis, shopping list generation, and user profile management through a seamless and interactive experience.

This project was developed as part of the Mini Project module under the Department of Information Technology, Vasavi College of Engineering. It integrates core concepts from frontend development, backend engineering, database management, and API integration, providing an opportunity to build a comprehensive, user-centered solution.

The front end of the application has been developed using HTML, CSS, JavaScript, and Bootstrap to create a responsive and intuitive user interface. It features pages for meal planning, nutrient tracking, shopping list management, user profiles, contact information, and more. The backend architecture has been built using Node.js with Express and MongoDB, following a layered design to ensure modularity, maintainability, and scalability of the application.

A key highlight of the Meal Planner & Nutrition Tracker is its ability to analyze user-entered meal data using third-party APIs to calculate calorie intake, protein, fats, and carbohydrates. The application also enables users to dynamically manage their meals and shopping lists, view nutrient charts, and track their dietary progress over time. Secure authentication mechanisms and form validations are implemented to ensure a safe and reliable user experience.

This project not only shows our technical expertise but also reflects our ability to solve real-world problems through software development. Through the creation of the Meal Planner & Nutrition Tracker, we have gained hands-on experience with full-stack technologies, real-time data management, and user experience design, preparing us for more advanced project development in professional environments.

TECHNOLOGY

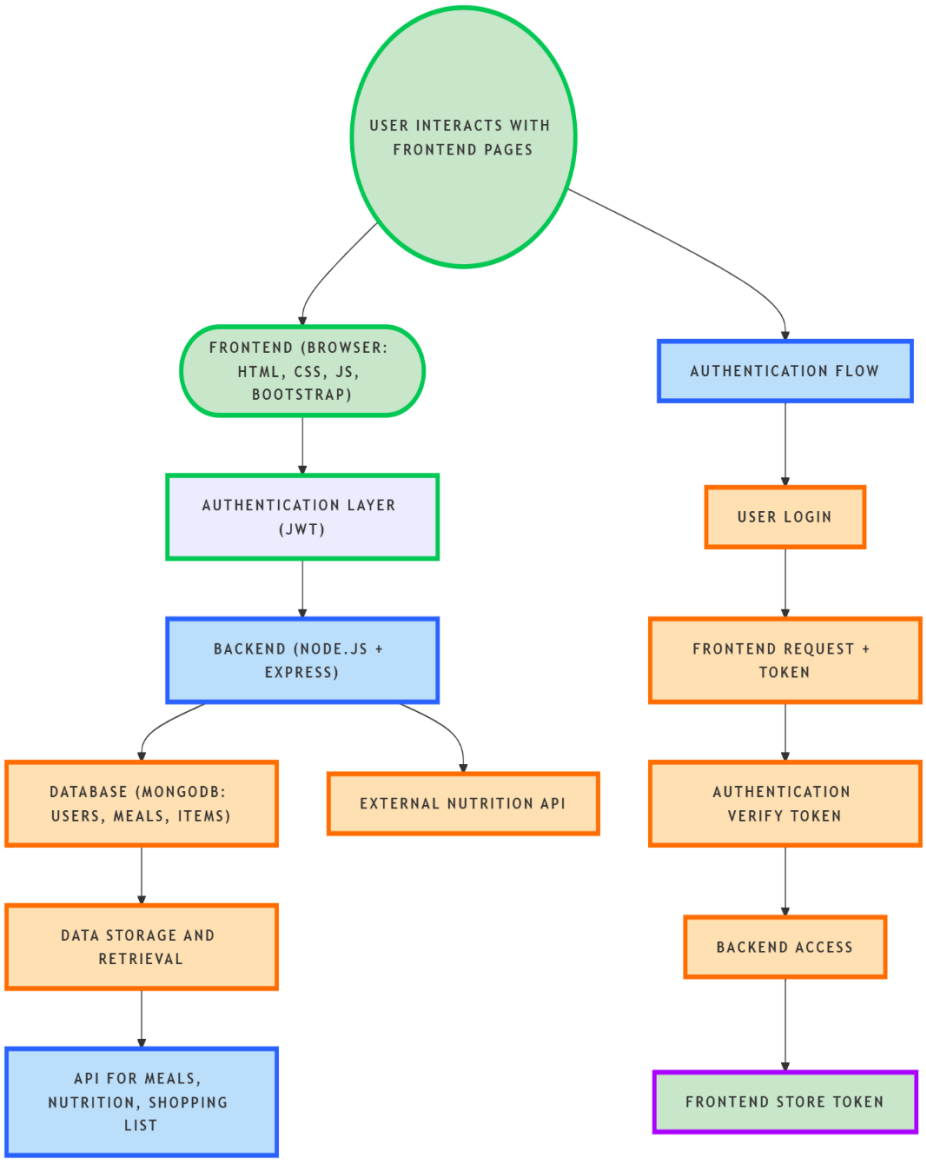
a. Software Requirements:

Component	Details
Frontend	HTML5, CSS3, JavaScript, Bootstrap
Backend	Node.js with Express.js (JavaScript)
Database	MongoDB (NoSQL Database)
Package Manager	npm (for Node.js)
IDE/Code Editor	Visual Studio Code
Version Control	Git
Browser	Google Chrome / Mozilla Firefox (for testing)
Other Tools	Postman (API testing), MongoDB Compass (GUI for MongoDB)

b. Hardware Requirements

Component	Minimum Specification
Processor	Intel Core i3 or equivalent (i5/i7 recommended)
RAM	4 GB (8 GB or more recommended for smoother dev)
Storage	At least 1 GB free space for project files
Display	13" screen or larger, 1366×768 resolution minimum
Internet	Required for npm installs, testing APIs, etc.

Architecture diagram



Implementation

BACKEND:

DatabaseConnection :

```
const mongoose = require('mongoose');

// MongoDB Connection

// MongoDB Connection with better error handling

mongoose.connect(process.env.MONGODB_URI || 'mongodb://localhost:27017/nutrient-tracker', {

   useNewUrlParser: true,

   useUnifiedTopology: true

})

.then(() => {

   console.log('✓ MongoDB Connected successfully');

      console.log('✓ Database URL:', process.env.MONGODB_URI || 'mongodb://localhost:27017/nutrient-tracker');

})

.catch(err => {

   console.error('✗ MongoDB Connection Error:');

   console.error(err);

   process.exit(1);

});

// Log when database connection is lost

mongoose.connection.on('error', err => {

   console.error('MongoDB connection error:', err);

});

mongoose.connection.on('disconnected', () => {

   console.log('MongoDB disconnected');

});
```

Models:

```
//Meal.js
const mongoose = require('mongoose');

const mealSchema = new mongoose.Schema({
    userId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'User',
        required: true
    },
    name: {
        type: String,
        required: true
    },
    date: {
        type: Date,
        default: Date.now
    },
    type: {
        type: String,
        enum: ['breakfast', 'lunch', 'dinner', 'snack'],
        required: true
    },
    nutrients: {
        calories: Number,
        protein: Number,
        carbs: Number,
        fats: Number
    }
});

module.exports = mongoose.model('Meal', mealSchema);
```

//ShoppingItem.js

```
const mongoose = require('mongoose');

const mealSchema = new mongoose. Schema({
    userId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'User',
        required: true
    },
    name: {
        type: String,
        required: true
    },
    date: {
        type: Date,
        default: Date.now
    },
    type: {
        type: String,
        enum: ['breakfast', 'lunch', 'dinner', 'snack'],
        required: true
    },
    nutrients: {
        calories: Number,
        protein: Number,
        carbs: Number,
        fats: Number
    }
});

module.exports = mongoose.model('Meal', mealSchema);
```

//Users.js:

```
const mongoose = require('mongoose');


const userSchema = new mongoose.Schema({
    username: {
        type: String,
        required: true,
        unique: true,
        trim: true
    },
    email: {
        type: String,
        required: true,
        unique: true,
        trim: true
    },
    password: {
        type: String,
        required: true
    },
    name: {
        type: String,
        trim: true
    },
    diet: {
        type: String,
        enum: ['balanced', 'low-carb', 'high-protein', 'vegetarian', 'vegan'],
        default: 'balanced'
    },
    goal: {
        type: String,
        enum: ['lose', 'maintain', 'gain'],
        default: 'maintain'
    },
    createdAt: {
        type: Date,
        default: Date.now
```

```
    }
});

module.exports = mongoose.model('User', userSchema);
```

Routes:

//meals.js

```
const express = require('express');
const router = express.Router();
const auth = require('../middleware/auth');
const Meal = require('../models/Meal');
const axios = require('axios');

// Get all meals for a user
router.get('/', auth, async (req, res) => {
    try {
        const meals = await Meal.find({ userId: req.user.id }).sort({ date: -1 });
        res.json(meals);
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});

// Add new meal
router.post('/', auth, async (req, res) => {
    try {
        const newMeal = new Meal({
            userId: req.user.id,
            ...req.body
        });

        const meal = await newMeal.save();
        res.json(meal);
    } catch (err) {
```

```
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});

// Update meal
router.put('/:id', async (req, res) => {
    try {
        let meal = await Meal.findById(req.params.id);
        if (!meal) return res.status(404).json({ message: 'Meal not found' });

        // Make sure user owns meal
        if (meal.userId.toString() !== req.user.id) {
            return res.status(401).json({ message: 'Not authorized' });
        }

        meal = await Meal.findByIdAndUpdate(
            req.params.id,
            { $set: req.body },
            { new: true }
        );

        res.json(meal);
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});

// Delete meal
router.delete('/:id', async (req, res) => {
    try {
        let meal = await Meal.findById(req.params.id);
        if (!meal) return res.status(404).json({ message: 'Meal not found' });
```

```
        // Make sure user owns meal
        if (meal.userId.toString() !== req.user.id) {
            return res.status(401).json({ message: 'Not authorized' });
        }

        await Meal.findByIdAndRemove(req.params.id);
        res.json({ message: 'Meal removed' });
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});

// Analyze food using Nutritionix API
router.post('/analyze-food', auth, async (req, res) => {
    try {
        const { description } = req.body;
        if (!description) {
            return res.status(400).json({ message: 'Please provide a food description' });
        }

        // Call Nutritionix API
        const response = await axios.post('https://trackapi.nutritionix.com/v2/natural/nutrients',
            { query: description },
            {
                headers: {
                    'x-app-id': process.env.NUTRITIONIX_APP_ID,
                    'x-app-key': process.env.NUTRITIONIX_APP_KEY,
                    'Content-Type': 'application/json'
                }
            }
        );
```

```
        // Extract relevant nutrition information
        const foods = response.data.foods;
        const totalNutrients = foods.reduce((acc, food) => {
            acc.calories += food.nf_calories || 0;
            acc.protein += food.nf_protein || 0;
            acc.carbs += food.nf_total_carbohydrate || 0;
            acc.fats += food.nf_total_fat || 0;
            return acc;
        }, { calories: 0, protein: 0, carbs: 0, fats: 0 });

        res.json(totalNutrients);
    } catch (err) {
        console.error('Error analyzing food:', err.message);
        res.status(500).json({ message: 'Failed to analyze food' });
    }
});

module.exports = router;
```

//shopping-list.js

```
const express = require('express');
const router = express.Router();
const auth = require('../middleware/auth');
const ShoppingItem = require('../models/ShoppingItem');

// Get all shopping items for a user
router.get('/', auth, async (req, res) => {
    try {
        const items = await ShoppingItem.find({ userId: req.user.id }).sort({ createdAt: -1 });
        res.json(items);
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});
```

```
// Add a new shopping item
router.post('/', auth, async (req, res) => {
    try {
        const newItem = new ShoppingItem({
            userId: req.user.id,
            name: req.body.name,
            category: req.body.category,
            quantity: req.body.quantity,
            completed: req.body.completed || false
        });

        const item = await newItem.save();
        res.json(item);
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});

// Update a shopping item
router.patch('/:id', auth, async (req, res) => {
    try {
        let item = await ShoppingItem.findById(req.params.id);

        if (!item) {
            return res.status(404).json({ msg: 'Item not found' });
        }

        // Make sure user owns the item
        if (item.userId.toString() !== req.user.id) {
            return res.status(401).json({ msg: 'User not authorized' });
        }
```

```
        item = await ShoppingItem.findByIdAndUpdate(
            req.params.id,
            { $set: req.body },
            { new: true }
        );

        res.json(item);
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});

// Delete a shopping item
router.delete('/:id', auth, async (req, res) => {
    try {
        const item = await ShoppingItem.findById(req.params.id);

        if (!item) {
            return res.status(404).json({ msg: 'Item not found' });
        }

        // Make sure user owns the item
        if (item.userId.toString() !== req.user.id) {
            return res.status(401).json({ msg: 'User not authorized' });
        }

        await item.remove();
        res.json({ msg: 'Item removed' });
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server Error');
    }
});
```

```
module.exports = router;
//users.js
const express = require('express');
const router = express.Router();
const bcrypt = require('bcryptjs');
const jwt = require('jsonwebtoken');
const auth = require('../middleware/auth');
const User = require('../models/User');
const Meal = require('../models/Meal');
const ShoppingItem = require('../models/ShoppingItem');

// Register User
router.post('/register', async (req, res) => {
  try {
    const { username, email, password, name } = req.body;

    // Check if user exists
    let user = await User.findOne({ email });
    if (user) {
      return res.status(400).json({ message: 'User already exists' });
    }

    // Create new user
    user = new User({
      username,
      email,
      password,
      name: name || username
    });

    // Hash password
    const salt = await bcrypt.genSalt(10);
    user.password = await bcrypt.hash(password, salt);

    await user.save();
```

```
        // Create JWT token
        const payload = {
            user: {
                id: user.id
            }
        };

        jwt.sign(
            payload,
            process.env.JWT_SECRET || 'your-secret-key',
            { expiresIn: '1h' },
            (err, token) => {
                if (err) throw err;
                res.json({ token });
            }
        );
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server error');
    }
});

// Login User
router.post('/login', async (req, res) => {
    try {
        const { email, password } = req.body;

        // Check if user exists
        let user = await User.findOne({ email });
        if (!user) {
            return res.status(400).json({ message: 'Invalid credentials' });
        }

        // Verify password
        const isMatch = await bcrypt.compare(password, user.password);
```

```
        if (!isMatch) {
            return res.status(400).json({ message: 'Invalid credentials' });
        }

        // Create JWT token
        const payload = {
            user: {
                id: user.id
            }
        };

        jwt.sign(
            payload,
            process.env.JWT_SECRET || 'your-secret-key',
            { expiresIn: '1h' },
            (err, token) => {
                if (err) throw err;
                res.json({ token });
            }
        );
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server error');
    }
});

// Get user profile
router.get('/profile', auth, async (req, res) => {
    try {
        const user = await User.findById(req.user.id).select('-password');
        res.json(user);
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server error');
    }
});
```

```
// Update user profile
router.put('/profile', auth, async (req, res) => {
    try {
        const { name, diet, goal } = req.body;
        const user = await User.findById(req.user.id);

        if (!user) {
            return res.status(404).json({ message: 'User not found' });
        }

        user.name = name || user.name;
        user.diet = diet || user.diet;
        user.goal = goal || user.goal;

        await user.save();
        res.json(user);
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server error');
    }
});

// Get user stats
router.get('/stats', auth, async (req, res) => {
    try {
        const meals = await Meal.find({ userId: req.user.id });
        const shoppingLists = await ShoppingItem.find({ userId: req.user.id });

        // Calculate stats
        const mealCount = meals.length;
        const avgCalories = mealCount > 0
            ? Math.round(meals.reduce((acc, meal) => acc + (meal.nutrients.calories || 0), 0) / mealCount)
            : 0;
        const listCount = shoppingLists.length;
```

```
        res.json({
            mealCount,
            avgCalories,
            listCount
        });
    } catch (err) {
        console.error(err.message);
        res.status(500).send('Server error');
    }
});

module.exports = router;
```

Middleware:

//auth.js

```
const jwt = require('jsonwebtoken');

module.exports = function(req, res, next) {
    // Get token from header
    const token = req.header('x-auth-token');

    // Check if no token
    if (!token) {
        return res.status(401).json({ message: 'No token, authorization denied' });
    }

    try {
        // Verify token
        const decoded = jwt.verify(token, process.env.JWT_SECRET || 'your-secret-key');
        req.user = decoded.user;
        next();
    } catch (err) {
        res.status(401).json({ message: 'Token is not valid' });
    }
};
```

//Checkauth.js

```
function checkAuth(req, res, next) {
    // Get token from local storage
    const token = localStorage.getItem('token');

    // Check if no token
    if (!token) {
        window.location.href = '/login.html';
        return;
    }

    try {
        // Add token to all API requests
        req.headers['x-auth-token'] = token;
        next();
    } catch (err) {
        window.location.href = '/login.html';
    }
}
```

// Server.js

```
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const dotenv = require('dotenv');

dotenv.config();

const app = express();

// Middleware
app.use(cors({
    origin: '*',
    methods: ['GET', 'POST', 'PUT', 'DELETE'],
    allowedHeaders: ['Content-Type', 'x-auth-token']
```

```
}));
app.use(express.json());
app.use(express.static('../'));

// MongoDB Connection
// MongoDB Connection with better error handling
mongoose.connect(process.env.MONGODB_URI || 'mongodb://localhost:27017/nutrient-tracker', {
    useNewUrlParser: true,
    useUnifiedTopology: true
})
.then(() => {
    console.log('✓ MongoDB Connected successfully');
    console.log('✓ Database URL:', process.env.MONGODB_URI || 'mongodb://localhost:27017/nutrient-tracker');
})
.catch(err => {
    console.error('✗ MongoDB Connection Error:');
    console.error(err);
    process.exit(1);
});

// Log when database connection is lost
mongoose.connection.on('error', err => {
    console.error('MongoDB connection error:', err);
});

mongoose.connection.on('disconnected', () => {
    console.log('MongoDB disconnected');
});

// Define Routes
app.use('/api/users', require('./routes/users'));
app.use('/api/meals', require('./routes/meals'));
app.use('/api/shopping-list', require('./routes/shopping-list'));
```

```
// Root route redirect to login
app.get('/', (req, res) => {
    res.redirect('/login.html');
});

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

FrontEnd:

Dashboard:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard - Meal Planner</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      background: url('images/image2.jpg') no-repeat center center/cover;
      min-height: 100vh;
    }
    .navbar {
      background-color: #343a40 !important;
    }
    .navbar-brand, .nav-link {
      color: #ffffff !important;
    }
    .main-container {
      padding: 20px;
      min-height: calc(100vh - 100px);
    }
    .welcome-message {
      font-size: 2rem;
      font-weight: bold;
      color: #ffc107;
      text-shadow: 2px 2px 4px #000000;
      text-align: center;
      margin-bottom: 20px;
    }
    .card {
      border-radius: 8px;
      box-shadow: 0px 0px 10px #000000;
    }
    #nutritionChart {
      width: 100%;
      height: 300px !important; /* Reduced height */
    }
    footer {
      background-color: #2c3e50;
      color: #ecf0f1;
      text-align: center;
      padding: 15px;
      position: fixed;
      bottom: 0;
      width: 100%;
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark">
    <div class="container">
      <a class="navbar-brand" href="base.html">
        <i class="fas fa-utensils me-2"></i>Meal Planner
      </a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
```

```
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav me-auto" id="authNav">
          <!-- Dynamic links will be inserted here via JS -->
        </ul>
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="dashboard.html">🧭 Dashboard</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="meal_plan.html">🍽️ My Meals</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="shopping-list.html">🛒 Shopping List</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="contact.html">📞 Contact</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="profile.html">👤 Profile</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="about.html">ℹ About</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>



  <div class="main-container">
    <h1 class="welcome-message">Welcome, User!</h1>

    <div class="row g-4">
      <div class="col-md-3">
        <div class="card text-center">
          <div class="card-body">
            <i class="fas fa-calendar-check fa-2x text-primary mb-3"></i>
            <h5>Meals Planned</h5>
            <p id="mealsPlanned">0</p>
          </div>
        </div>
      </div>
      <div class="col-md-3">
        <div class="card text-center">
          <div class="card-body">
            <i class="fas fa-fire fa-2x text-danger mb-3"></i>
            <h5>Calories Today</h5>
            <p id="caloriesToday">0</p>
          </div>
        </div>
      </div>
      <div class="col-md-3">
        <div class="card text-center">
          <div class="card-body">
```

```
                <i class="fas fa-weight fa-2x text-success mb-3"></i>
                <h5>Protein (g)</h5>
                <p id="proteinToday">0</p>
            </div>
        </div>
    </div>
    <div class="col-md-3">
        <div class="card text-center">
            <div class="card-body">
                <i class="fas fa-shopping-cart fa-2x text-warning mb-3"></i>
                <h5>Shopping Items</h5>
                <p id="shoppingItems">0</p>
            </div>
        </div>
    </div>
</div>

<div class="row mt-4">
    <div class="col-md-8">
        <div class="card">
            <div class="card-header">
                <h5>Nutrition Overview</h5>
            </div>
            <div class="card-body">
                <canvas id="nutritionChart"></canvas>
            </div>
        </div>
    </div>
    <div class="col-md-4">
        <div class="card">
            <div class="card-header">
                <h5>Today's Meals</h5>
            </div>
            <div class="card-body">
                <div id="upcomingMeals">No meals planned for today</div>
            </div>
        </div>
    </div>
</div>
</div>

<footer>
    <p>© 2025 Meal Planner & Nutrition Tracker. All Rights Reserved.</p>
</footer>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script>
    // Check authentication
    document.addEventListener('DOMContentLoaded', function () {
        const token = localStorage.getItem('token');
        if (!token) {
            window.location.href = '/login.html';
            return;
        }
        fetchUserData();
        fetchMeals();
        fetchShoppingItems(); // Added call to fetch shopping items
        setupChart();
    });

    async function fetchUserData() {
        try {
```

```
        const response = await fetch('http://localhost:5000/api/users/me', {
            headers: {
                'x-auth-token': localStorage.getItem('token')
            }
        });
        const user = await response.json();
        document.querySelector('.welcome-message').textContent = `Welcome, ${user.username}!`;
    } catch (error) {
        console.error('Error fetching user data:', error);
    }
}

async function fetchMeals() {
    try {
        const response = await fetch('http://localhost:5000/api/meals', {
            headers: {
                'x-auth-token': localStorage.getItem('token')
            }
        });
        const meals = await response.json();

        // Update stats
        document.getElementById('mealsPlanned').textContent = meals.length;

        // Calculate today's nutrition
        const today = new Date().toISOString().split('T')[0];
        const todayMeals = meals.filter(meal => meal.date.startsWith(today));

        let totalCalories = 0;
        let totalProtein = 0;

        todayMeals.forEach(meal => {
            totalCalories += meal.nutrients?.calories || 0;
            totalProtein += meal.nutrients?.protein || 0;
        });

        document.getElementById('caloriesToday').textContent = Math.round(totalCalories);
        document.getElementById('proteinToday').textContent = Math.round(totalProtein);

        // Update meals list
        const upcomingMealsDiv = document.getElementById('upcomingMeals');
        if (todayMeals.length > 0) {
            upcomingMealsDiv.innerHTML = todayMeals.map(meal => `
                <div class="meal-item mb-2">
                    <strong>${meal.name}</strong> (${meal.type})<br>
                    <small>Calories: ${meal.nutrients?.calories || 0} | Protein: ${meal.nutrients?.protein || 0}g</small>
                </div>
            `).join('');
        }
    } catch (error) {
        console.error('Error fetching meals:', error);
    }
}

async function fetchShoppingItems() {
    try {
        const response = await fetch('http://localhost:5000/api/shopping-list', {
            headers: {
                'x-auth-token': localStorage.getItem('token')
            }
        });
        const items = await response.json();
```

```
        // Update shopping items count
        document.getElementById('shoppingItems').textContent = items.length;
    } catch (error) {
        console.error('Error fetching shopping items:', error);
    }
}

async function setupChart() {
    try {
        const response = await fetch('http://localhost:5000/api/meals', {
            headers: {
                'x-auth-token': localStorage.getItem('token')
            }
        });
        const meals = await response.json();

        // Get last 7 days
        const dates = [];
        const calories = [];

        for (let i = 6; i >= 0; i--) {
            const date = new Date();
            date.setDate(date.getDate() - i);
            const dateStr = date.toISOString().split('T')[0];
            dates.push(date.toLocaleDateString('en-US', { weekday: 'short' }));

            const dayMeals = meals.filter(meal => meal.date.startsWith(dateStr));
            const dayCalories = dayMeals.reduce((sum, meal) => sum + (meal.nutrients?.calories || 0), 0);
            calories.push(dayCalories);
        }

        const ctx = document.getElementById('nutritionChart').getContext('2d');
        new Chart(ctx, {
            type: 'line',
            data: {
                labels: dates,
                datasets: [{
                    label: 'Calories',
                    data: calories,
                    borderColor: 'rgb(255, 99, 132)',
                    fill: false,
                    tension: 0.1
                }]
            },
            options: {
                responsive: true,
                plugins: {
                    legend: {
                        position: 'top',
                    }
                },
                scales: {
                    y: {
                        beginAtZero: true
                    }
                }
            }
        });
    } catch (error) {
        console.error('Error setting up chart:', error);
    }
```

```
    }
  </script>
</body>
</html>
```

Basepage

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Meal Planner & Nutrition Tracker</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;

    }

    body {
      font-family: Arial, sans-serif;
      background: url('images/image5.jpg') no-repeat center center/cover;
      min-height: 100vh;
      color: #ffffff;
    }
    .navbar {
      background-color: rgba(0, 0, 0, 0.8) !important;
      padding: 3px 10px !important;
      height: 40px;
    }
    .navbar-brand, .nav-link {
      font-size: 0.85rem;
      padding-top: 2px;
      padding-bottom: 2px;
    }
    .navbar-dark .navbar-nav .nav-link {
      color: #ffffff !important;
    }
    .navbar-dark .navbar-nav .nav-link:hover {
      color: #ffc107 !important;
    }
    .main-container {
      min-height: calc(100vh - 100px);
      display: flex;
      justify-content: center;
      align-items: center;
    }
    .welcome-box {
      background: rgba(0, 0, 0, 0.7);
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0px 0px 20px #000000;
      text-align: center;
    }
    .welcome-box h1 {
      font-size: 2rem;
      margin-bottom: 10px;
    }
```

```
        .welcome-box p {
            font-size: 1.1rem;
            margin-bottom: 20px;
        }
        .btn-custom {
            background-color: #ffc107;
            color: #000000;
            font-weight: bold;
            border-radius: 20px;
            padding: 10px 20px;
            transition: 0.3s;
        }
        .btn-custom:hover {
            background-color: #ffda44;
            color: #000;
        }
        footer {
            background-color: #2c3e50;
            color: #ecf0f1;
            text-align: center;
            padding: 15px;
            position: fixed;
            bottom: 0;
            width: 100%;
        }
    </style>
</head>
<body>

    <nav class="navbar navbar-expand-lg navbar-dark">

        <div class="container">
            <a class="navbar-brand" href="base.html">
                <i class="fas fa-utensils me-2"></i>Meal Planner
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav me-auto" id="authNav"></ul>
            </div>

        </div>
    </nav>

    <div class="main-container">
        <div class="welcome-box">
            <h1>Welcome to Meal Planner & Nutrition Tracker</h1>
            <p>Your companion for a healthier lifestyle!</p>


            <button class="btn btn-custom" onclick="redirectToLogin()">Get Started</button>

            <script>
                // Function for Get Started Button
                function redirectToLogin() {
                    window.location.href = "login.html"; // Redirects to login.html
                }
            </script>

        </div>
    </div>
```

```
  <footer>
    <p>© 2025 Meal Planner & Nutrition Tracker. All Rights Reserved.</p>
  </footer>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
  <script>
    var isLoggedIn = false;

    function renderNavigation() {
      const authNav = document.getElementById("authNav");
      authNav.innerHTML = "";

      if (isLoggedIn) {
        authNav.innerHTML = `
          <li class="nav-item"><a class="nav-link" href="#" onclick="showContent('dashboard')"><i class="fas fa-
tachometer-alt me-1"></i>Dashboard</a></li>
          <li class="nav-item"><a class="nav-link" href="#" onclick="showContent('meal_plan')"><i class="fas fa-
calendar-alt me-1"></i>Meal Plan</a></li>
          <li class="nav-item"><a class="nav-link" href="#" onclick="showContent('nutrition_tracker')"><i class="fas
fa-chart-line me-1"></i>Nutrition</a></li>
          <li class="nav-item"><a class="nav-link" href="#" onclick="logout()"><i class="fas fa-sign-out-alt me-
1"></i>Logout</a></li>
        `;
      } else {
        authNav.innerHTML = `
          <li class="nav-item"><a class="nav-link" href="login.html" onclick="showContent('login')"><i class="fas fa-
sign-in-alt me-1"></i>Login</a></li>
          <li class="nav-item"><a class="nav-link" href="register.html" onclick="showContent('register')"><i class="fas
fa-user-plus me-1"></i>Register</a></li>
        `;
      }
    }

    function navigatetopage(page) {
      const content = {
        dashboard: "<h1>Dashboard</h1><p>Track your meals and nutrition.</p>",
        meal_plan: "<h1>Meal Plan</h1><p>Create and customize your meal plan.</p>",
        nutrition_tracker: "<h1>Nutrition Tracker</h1><p>Track your daily nutrition intake.</p>",
        login: "<h1>Login</h1><p>Please login to continue.</p>",
        register: "<h1>Register</h1><p>Create a new account.</p>"
      };
      document.querySelector(".welcome-box").innerHTML = content[page] || "<h1>Welcome!</h1>";
    }

    function login() {
      isLoggedIn = true;
      renderNavigation();
      alert("Successfully logged in!");
    }

    function logout() {
      isLoggedIn = false;
      renderNavigation();
      alert("Logged out successfully.");
    }

    renderNavigation();
  </script>
</body>
</html>
```

Homepage

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Meal Planner & Nutrition Tracker</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <style>
    body {
      background: url('images/home-bg.jpg') no-repeat center center/cover;
      font-family: 'Segoe UI', sans-serif;
      color: white;
      min-height: 100vh;
    }

    .overlay {
      background-color: rgba(0, 0, 0, 0.7);
      padding: 50px 20px;
      min-height: 100vh;
    }

    h1, h2 {
      font-weight: 700;
    }

    .feature-box {
      background: rgba(255, 255, 255, 0.1);
      border-radius: 15px;
      padding: 20px;
      margin-bottom: 20px;
      transition: all 0.3s ease;
    }

    .feature-box:hover {
      background: rgba(255, 255, 255, 0.2);
    }

    .nav-link {
      color: #fff !important;
    }

    .nav-link:hover {
      color: #ffc107 !important;
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-black">
    <div class="container-fluid">
      <a class="navbar-brand fw-bold" href="#">Meal Planner 🥗</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item"><a class="nav-link" href="base.html">Home</a></li>
          <li class="nav-item"><a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a></li>
          <li class="nav-item"><a class="nav-link" href="shopping-list.html">🛒 Shopping List</a></li>
```

```
        <li class="nav-item"><a class="nav-link" href="contact.html">📞 Contact</a></li>
        <li class="nav-item"><a class="nav-link" href="profile.html">👤 Profile</a></li>
        <li class="nav-item"><a class="nav-link" href="about.html">ℹ About</a></li>
      </ul>
    </div>
  </div>
 </nav>

 <div class="overlay text-center">
  <div class="container">
   <h1 class="display-4 mb-4">Welcome to Meal Planner & Nutrition Tracker</h1>
   <p class="lead mb-5">Your personalized companion for planning balanced meals, tracking nutrients, and living a
healthier lifestyle.</p>

    <div class="row text-start">
     <div class="col-md-4">
      <div class="feature-box">
       <h4>📅 Meal Planning</h4>
       <p>Plan your daily meals by adding what you eat and tracking your slots – Breakfast, Lunch, Dinner, and
Snacks.</p>
      </div>
     </div>
     <div class="col-md-4">
      <div class="feature-box">
       <h4>📈 Nutrient Tracker</h4>
       <p>Log calories, proteins, carbs, and fats. Keep track of your daily intake and nutritional goals with ease.</p>
      </div>
     </div>
     <div class="col-md-4">
      <div class="feature-box">
       <h4>🍽 Smart Meal Recommendations</h4>
       <p>Get AI-based meal suggestions based on your preferences like calorie range, meal type, and diet.</p>
      </div>
     </div>
     <div class="col-md-4">
      <div class="feature-box">
       <h4>🛒 Shopping List</h4>
       <p>Create and manage a shopping list based on your meal plan to never miss an ingredient.</p>
      </div>
     </div>
     <div class="col-md-4">
      <div class="feature-box">
       <h4>📊 Weekly Nutrition Chart</h4>
       <p>Visualize your weekly progress with a nutritional chart to stay motivated and consistent.</p>
      </div>
     </div>
     <div class="col-md-4">
      <div class="feature-box">
       <h4>💾 Save & Export</h4>
       <p>Save your data and export your meal history or nutrition summary as a PDF for future reference.</p>
      </div>
     </div>
    </div>
   </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Register

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Register - Meal Planner</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"/>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"/>
  <style>
   * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
   }
   body {
    font-family: Arial, sans-serif;
    background: url('images/image3.jpg') no-repeat center center/cover;
    min-height: 100vh;
    color: #ffffff;
   }
   .navbar {
    background-color: rgba(0, 0, 0, 0.8) !important;
    padding: 3px 10px !important;
    height: 40px;
   }
   .navbar-brand, .nav-link {
    font-size: 0.85rem;
    padding-top: 2px;
    padding-bottom: 2px;
   }
   .navbar-dark .navbar-nav .nav-link {
    color: #ffffff !important;
   }
   .navbar-dark .navbar-nav .nav-link:hover {
    color: #ffc107 !important;
   }
   .main-container {
    min-height: calc(100vh - 100px);
    display: flex;
    justify-content: center;
    align-items: center;
   }
   .register-box {
    background: rgba(0, 0, 0, 0.7);
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0px 0px 20px #000000;
    text-align: center;
    width: 100%;
    max-width: 400px;
   }
   .register-box h3 {
    font-size: 1.8rem;
    margin-bottom: 20px;
   }
   .form-label {
    color: #ffffff;
   }
   .input-group-text {
    background-color: #ffc107;
```

```
      color: #000000;
    }
    .btn-custom {
      background-color: #ffc107;
      color: #000000;
      font-weight: bold;
      border-radius: 20px;
      padding: 10px 20px;
      transition: 0.3s;
    }
    .btn-custom:hover {
      background-color: #ffda44;
      color: #000;
    }
    .text-primary {
      color: #ffc107 !important;
    }
    footer {
      background-color: #2c3e50;
      color: #ecf0f1;
      text-align: center;
      padding: 15px;
      position: fixed;
      bottom: 0;
      width: 100%;
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark">
    <div class="container">
      <a class="navbar-brand" href="login.html">
        <i class="fas fa-utensils me-2"></i>Meal Planner
      </a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav me-auto">
          <li class="nav-item">
            <a class="nav-link" href="login.html"><i class="fas fa-sign-in-alt me-1"></i>Login</a>
          </li>
          <li class="nav-item">
            <a class="nav-link active" href="register.html"><i class="fas fa-user-plus me-1"></i>Register</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

  <div class="main-container">
    <div class="register-box">
      <h3><i class="fas fa-user-plus me-2"></i>Register</h3>
      <form id="registerForm">
        <div class="mb-3">
          <label for="fullName" class="form-label">Full Name</label>
          <div class="input-group">
            <span class="input-group-text"><i class="fas fa-user"></i></span>
            <input type="text" class="form-control" id="fullName" name="fullName" required>
          </div>
        </div>
        <div class="mb-3">
```

```
      <label for="email" class="form-label">Email address</label>
      <div class="input-group">
        <span class="input-group-text"><i class="fas fa-envelope"></i></span>
        <input type="email" class="form-control" id="email" name="email" required>
      </div>
    </div>
    <div class="mb-3">
      <label for="password" class="form-label">Password</label>
      <div class="input-group">
        <span class="input-group-text"><i class="fas fa-lock"></i></span>
        <input type="password" class="form-control" id="password" name="password" required>
      </div>
    </div>
    <button type="submit" class="btn btn-custom btn-lg w-100">
      <i class="fas fa-user-plus me-2"></i>Register
    </button>
  </form>
  <div class="text-center mt-4">
    <p class="mb-0">Already have an account? <a href="login.html" class="text-primary">Login here</a></p>
  </div>
 </div>
</div>

<footer>
 <p>© 2025 Meal Planner & Nutrition Tracker. All Rights Reserved.</p>
</footer>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
<script>
 document.getElementById("registerForm").addEventListener("submit", async function (e) {
  e.preventDefault();

  const username = document.getElementById("fullName").value.trim();
  const email = document.getElementById("email").value.trim();
  const password = document.getElementById("password").value;

  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  let errorMessage = "";

  if (username.length < 3) {
   errorMessage += "Full name must be at least 3 characters long. ";
  }
  if (!emailRegex.test(email)) {
   errorMessage += "Please enter a valid email address. ";
  }
  if (password.length < 6) {
   errorMessage += "Password must be at least 6 characters long.";
  }

  if (errorMessage) {
   alert(errorMessage);
   return;
  }

  try {
   const response = await fetch('http://localhost:5000/api/users/register', {
    method: 'POST',
    headers: {
     'Content-Type': 'application/json'
    },
    body: JSON.stringify({
     username,
```

```
          email,
          password,
          name: username
        })
      });

      const data = await response.json();

      if (response.ok) {
        // Store the token
        localStorage.setItem('token', data.token);
        alert('Registration successful!');
        window.location.href = 'dashboard.html';
      } else {
        alert(data.message || 'Registration failed. Please try again.');
        document.getElementById('registerForm').reset();
      }
    } catch (error) {
      console.error('Error:', error);
      alert('Server error. Please try again later.');
    }
  });
 </script>
</body>
</html>
```

userProfile

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Profile - Meal Planner</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            background: url('images/image2.jpg') no-repeat center center/cover;
            min-height: 100vh;
        }
        .navbar {
            background-color: #343a40 !important;
        }
        .navbar-brand, .nav-link {
            color: #ffffff !important;
        }
        .container-box {
            background: rgba(255, 255, 255, 0.9);
            border-radius: 15px;
            padding: 30px;
            margin-top: 30px;
            box-shadow: 0 0 15px rgba(0,0,0,0.1);
        }
        .profile-section {
            margin-bottom: 30px;
        }
        .btn-custom {
            background-color: #28a745;
            color: white;
            border: none;
```

```
            padding: 10px 20px;
            border-radius: 5px;
            cursor: pointer;
        }
        .btn-custom:hover {
            background-color: #218838;
        }
        .stats-card {
            background: white;
            border-radius: 10px;
            padding: 20px;
            margin-bottom: 20px;
            box-shadow: 0 2px 5px rgba(0,0,0,0.1);
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark">
        <div class="container">
            <a class="navbar-brand" href="dashboard.html">
                <i class="fas fa-utensils me-2"></i>Meal Planner
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="dashboard.html">🧭 Dashboard</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="meal_plan.html">🍽️ My Meals</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="shopping-list.html">🛒 Shopping List</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="contact.html">📞 Contact</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="profile.html">👤 Profile</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="about.html">ℹ️ About</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <div class="container-box">
            <div class="row">
                <div class="col-md-8">
                    <div class="profile-section">
                        <h2 class="mb-4">👤 Your Profile</h2>
                        <form id="profileForm">
                            <div class="mb-3">
```

```
                    <label for="name" class="form-label">Name</label>
                    <input type="text" class="form-control" id="name" required>
                  </div>
                  <div class="mb-3">
                    <label for="email" class="form-label">Email</label>
                    <input type="email" class="form-control" id="email" required readonly>
                  </div>
                  <div class="mb-3">
                    <label for="diet" class="form-label">Preferred Diet</label>
                    <select class="form-control" id="diet">
                      <option value="balanced">Balanced</option>
                      <option value="low-carb">Low Carb</option>
                      <option value="high-protein">High Protein</option>
                      <option value="vegetarian">Vegetarian</option>
                      <option value="vegan">Vegan</option>
                    </select>
                  </div>
                  <div class="mb-3">
                    <label for="goal" class="form-label">Goal</label>
                    <select class="form-control" id="goal">
                      <option value="lose">Lose Weight</option>
                      <option value="maintain">Maintain Weight</option>
                      <option value="gain">Gain Weight</option>
                    </select>
                  </div>
                  <button type="submit" class="btn btn-custom">Save Changes</button>
                </form>
              </div>
            </div>
            <div class="col-md-4">
              <div class="stats-section">
                <h3 class="mb-4">Your Stats</h3>
                <div class="stats-card">
                  <h5>Meal Plans Created</h5>
                  <p class="h2 mb-0" id="mealCount">0</p>
                </div>
                <div class="stats-card">
                  <h5>Average Daily Calories</h5>
                  <p class="h2 mb-0" id="avgCalories">0</p>
                </div>
                <div class="stats-card">
                  <h5>Shopping Lists</h5>
                  <p class="h2 mb-0" id="listCount">0</p>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>

      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
      <script>
        // Check authentication
        document.addEventListener('DOMContentLoaded', async function() {
          const token = localStorage.getItem('token');
          if (!token) {
            window.location.href = '/login.html';
            return;
          }

          try {
            // Fetch user profile
```

```
            const response = await fetch('http://localhost:5000/api/users/profile', {
                headers: {
                    'x-auth-token': token
                }
            });
            const user = await response.json();

            // Populate form
            document.getElementById('name').value = user.name;
            document.getElementById('email').value = user.email;
            document.getElementById('diet').value = user.diet || 'balanced';
            document.getElementById('goal').value = user.goal || 'maintain';

            // Fetch stats
            const statsResponse = await fetch('http://localhost:5000/api/users/stats', {
                headers: {
                    'x-auth-token': token
                }
            });
            const stats = await statsResponse.json();

            // Update stats
            document.getElementById('mealCount').textContent = stats.mealCount || 0;
            document.getElementById('avgCalories').textContent = stats.avgCalories || 0;
            document.getElementById('listCount').textContent = stats.listCount || 0;
        } catch (error) {
            console.error('Error:', error);
        }
    });

    // Handle form submission
    document.getElementById('profileForm').addEventListener('submit', async function(e) {
        e.preventDefault();
        const token = localStorage.getItem('token');

        try {
            const response = await fetch('http://localhost:5000/api/users/profile', {
                method: 'PUT',
                headers: {
                    'Content-Type': 'application/json',
                    'x-auth-token': token
                },
                body: JSON.stringify({
                    name: document.getElementById('name').value,
                    diet: document.getElementById('diet').value,
                    goal: document.getElementById('goal').value
                })
            });

            if (response.ok) {
                alert('Profile updated successfully!');
            } else {
                alert('Failed to update profile');
            }
        } catch (error) {
            console.error('Error:', error);
            alert('Error updating profile');
        }
    });
  </script>
</body>
</html>
```

Shopping-list

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Shopping List</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            background: url('images/image2.jpg') no-repeat center center/cover;
            min-height: 100vh;
        }
        .navbar {
            background-color: #343a40 !important;
        }
        .navbar-brand, .nav-link {
            color: #ffffff !important;
        }
        .container-box {
            background: rgba(255, 255, 255, 0.9);
            border-radius: 15px;
            padding: 20px;
            margin-top: 30px;
            box-shadow: 0 0 15px rgba(0,0,0,0.1);
        }
        .shopping-item {
            display: flex;
            align-items: center;
            padding: 10px;
            border-bottom: 1px solid #eee;
            transition: all 0.3s ease;
        }
        .shopping-item:last-child {
            border-bottom: none;
        }
        .shopping-item.completed {
            text-decoration: line-through;
            color: #888;
            background-color: rgba(0, 0, 0, 0.05);
        }
        .category-badge {
            font-size: 0.8em;
            padding: 3px 8px;
            border-radius: 10px;
            margin-left: 10px;
        }
        .btn-custom {
            background-color: #ffc107;
            color: white;
            border: none;
            padding: 8px 16px;
            border-radius: 5px;
            cursor: pointer;
            transition: background-color 0.3s;
        }
        .btn-custom:hover {
            background-color: #ffc107;
```

```
        }
        .category-header {
            background-color: #f8f9fa;
            padding: 10px;
            border-radius: 5px;
            margin-bottom: 10px;
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark">
        <div class="container">
            <a class="navbar-brand" href="dashboard.html">
                <i class="fas fa-utensils me-2"></i>Meal Planner
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="dashboard.html">🧭 Dashboard</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="meal_plan.html">🍽️ My Meals</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="shopping-list.html">🛒 Shopping List</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="contact.html">📞 Contact</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="profile.html">👤 Profile</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="about.html">ℹ️ About</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <div class="container-box">
            <div class="d-flex justify-content-between align-items-center mb-4">
                <h2>🛒 Shopping List</h2>
                <button class="btn btn-custom" onclick="downloadList()">
                    <i class="fas fa-download me-2"></i>Download List
                </button>
            </div>

            <!-- Add Item Form -->
            <form id="addItemForm" class="mb-4">
                <div class="row g-3">
                    <div class="col-md-5">
                        <input type="text" class="form-control" id="itemName" placeholder="Item name" required>
                    </div>
```

```
            <div class="col-md-3">
              <select class="form-select" id="itemCategory" required>
                <option value="">Category...</option>
                <option value="produce">🥦 Produce</option>
                <option value="dairy">🥛 Dairy</option>
                <option value="meat">🥩 Meat</option>
                <option value="pantry">🥫 Pantry</option>
                <option value="frozen">❄️ Frozen</option>
                <option value="other">📦 Other</option>
              </select>
            </div>
            <div class="col-md-2">
              <input type="number" class="form-control" id="itemQuantity" placeholder="Qty" min="1" required>
            </div>
            <div class="col-md-2">
              <button type="submit" class="btn btn-custom w-100">
                <i class="fas fa-plus me-1"></i> Add
              </button>
            </div>
          </div>
        </form>

        <!-- Shopping List -->
        <div id="shoppingList">
          <!-- Items will be added here -->
        </div>
      </div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
    <script>
      // Check authentication
      document.addEventListener('DOMContentLoaded', function() {
        const token = localStorage.getItem('token');
        if (!token) {
          window.location.href = '/login.html';
          return;
        }
        loadShoppingList();
      });

      // Category colors and icons
      const categoryConfig = {
        produce: { color: 'bg-success', icon: '🥦' },
        dairy: { color: 'bg-info', icon: '🥛' },
        meat: { color: 'bg-danger', icon: '🥩' },
        pantry: { color: 'bg-warning', icon: '🥫' },
        frozen: { color: 'bg-primary', icon: '❄️' },
        other: { color: 'bg-secondary', icon: '📦' }
      };

      // Load shopping list
      async function loadShoppingList() {
        try {
          const response = await fetch('http://localhost:5000/api/shopping-list', {
            headers: {
              'x-auth-token': localStorage.getItem('token')
            }
          });
          const items = await response.json();
```

```
        displayItems(items);
    } catch (error) {
        console.error('Error loading shopping list:', error);
    }
}

// Display items
function displayItems(items) {
    const shoppingList = document.getElementById('shoppingList');
    const groupedItems = groupByCategory(items);

    shoppingList.innerHTML = Object.entries(groupedItems)
        .sort(([a], [b]) => a.localeCompare(b))
        .map(([category, items]) => `
            <div class="mb-4">
                <div class="category-header">
                    <h5 class="mb-0">
                        ${categoryConfig[category].icon}
                        ${category.charAt(0).toUpperCase() + category.slice(1)}
                        <span class="badge bg-secondary ms-2">${items.length}</span>
                    </h5>
                </div>
                ${items.map(item => `
                    <div class="shopping-item ${item.completed ? 'completed' : ''}">
                        <div class="form-check mb-0">
                            <input type="checkbox" class="form-check-input"
                                ${item.completed ? 'checked' : ''}
                                onchange="toggleItem('${item._id}', this.checked)">
                            <label class="form-check-label">
                                ${item.name} (${item.quantity})
                            </label>
                        </div>
                        <span class="category-badge ${categoryConfig[item.category].color} text-white ms-auto">
                            ${categoryConfig[item.category].icon}
                        </span>
                        <button class="btn btn-sm btn-danger ms-2"
                                onclick="deleteItem('${item._id}')">
                            <i class="fas fa-trash"></i>
                        </button>
                    </div>
                `).join('')}
            </div>
        `).join('');
}

// Group items by category
function groupByCategory(items) {
    return items.reduce((groups, item) => {
        const category = item.category || 'other';
        if (!groups[category]) {
            groups[category] = [];
        }
        groups[category].push(item);
        return groups;
    }, {});
}

// Add new item
document.getElementById('addItemForm').addEventListener('submit', async function(e) {
    e.preventDefault();

    const itemData = {
```

```
        name: document.getElementById('itemName').value,
        category: document.getElementById('itemCategory').value,
        quantity: parseInt(document.getElementById('itemQuantity').value),
        completed: false
    };

    try {
        const response = await fetch('http://localhost:5000/api/shopping-list', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'x-auth-token': localStorage.getItem('token')
            },
            body: JSON.stringify(itemData)
        });

        if (response.ok) {
            document.getElementById('addItemForm').reset();
            loadShoppingList();
        }
    } catch (error) {
        console.error('Error adding item:', error);
    }
});

// Toggle item completion
async function toggleItem(itemId, completed) {
    try {
        const response = await fetch(`http://localhost:5000/api/shopping-list/${itemId}`, {
            method: 'PATCH',
            headers: {
                'Content-Type': 'application/json',
                'x-auth-token': localStorage.getItem('token')
            },
            body: JSON.stringify({ completed })
        });

        if (response.ok) {
            loadShoppingList();
        }
    } catch (error) {
        console.error('Error updating item:', error);
    }
}

// Delete item
async function deleteItem(itemId) {
    if (!confirm('Are you sure you want to delete this item?')) return;

    try {
        const response = await fetch(`http://localhost:5000/api/shopping-list/${itemId}`, {
            method: 'DELETE',
            headers: {
                'x-auth-token': localStorage.getItem('token')
            }
        });

        if (response.ok) {
            loadShoppingList();
        }
    } catch (error) {
        console.error('Error deleting item:', error);
```

```
        }
    }

    // Download shopping list
    async function downloadList() {
        try {
            const response = await fetch('http://localhost:5000/api/shopping-list', {
                headers: {
                    'x-auth-token': localStorage.getItem('token')
                }
            });
            const items = await response.json();

            // Group items by category
            const groupedItems = groupByCategory(items);

            // Create text content
            let content = 'SHOPPING LIST\n\n';

            Object.entries(groupedItems).forEach(([category, items]) => {
                content += `${category.toUpperCase()}:\n`;
                items.forEach(item => {
                    content += `${item.completed ? '✓' : '☐'} ${item.name} (${item.quantity})\n`;
                });
                content += '\n';
            });

            // Create blob and download
            const blob = new Blob([content], { type: 'text/plain' });
            const url = window.URL.createObjectURL(blob);
            const a = document.createElement('a');
            a.href = url;
            a.download = 'shopping-list.txt';
            document.body.appendChild(a);
            a.click();
            window.URL.revokeObjectURL(url);
            document.body.removeChild(a);
        } catch (error) {
            console.error('Error downloading list:', error);
            alert('Failed to download shopping list');
        }
    }
  </script>
</body>
</html>
```

Nutrient tracker

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Nutrient Tracker</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      background: url('images/image2.jpg') no-repeat center center/cover;
      min-height: 100vh;
```

```
}
.navbar {
    background-color: #343a40 !important;
}
.navbar-brand, .nav-link {
    color: #ffffff !important;
}
.navbar-dark .navbar-nav .nav-link:hover,
.navbar-dark .navbar-nav .nav-link.active {
    color: #ffc107 !important;
}
.container-box {
    background: rgba(255, 255, 255, 0.9);
    border-radius: 15px;
    padding: 20px;
    margin-top: 30px;
    box-shadow: 0 0 15px rgba(0,0,0,0.1);
}
.progress {
    height: 25px;
    margin-bottom: 10px;
}
.nutrient-card {
    background: white;
    border-radius: 10px;
    padding: 15px;
    margin-bottom: 15px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
.table th, .table td {
    vertical-align: middle;
}
.footer {
    margin-top: 60px;
    color: #fff;
    text-align: center;
    font-size: 14px;
}
.btn-custom {
    background-color: #ffc107;
    border: none;
    color: white;
    font-weight: bold;
}
.btn-custom:hover {
    background-color: #e0a800;
}
.command-palette {
    position: absolute;
    top: 100%;
    left: 0;
    right: 0;
    background: white;
    border-radius: 5px;
    box-shadow: 0 2px 10px rgba(0,0,0,0.2);
    max-height: 200px;
    overflow-y: auto;
    z-index: 1000;
}
.command-palette .list-group-item {
    padding: 10px;
    cursor: pointer;
```

```
            border: none;
        }
        .command-palette .list-group-item:hover {
            background-color: #f8f9fa;
        }
        .command-palette .list-group-item small {
            display: block;
            color: #6c757d;
        }
        .command-palette .list-group-item.selected {
            background-color: #e9ecef;
        }
        .loading-spinner {
            display: none;
            text-align: center;
            padding: 10px;
            color: #6c757d;
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark">
        <div class="container">
            <a class="navbar-brand" href="dashboard.html">
                <i class="fas fa-utensils me-2"></i>Meal Planner
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="dashboard.html">🧭 Dashboard</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="meal_plan.html">🍽My Meals</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="shopping-list.html">🛒 Shopping List</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="contact.html">📞 Contact</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="profile.html">👤 Profile</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="about.html">ℹ About</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <div class="container-box">
            <div class="d-flex justify-content-between align-items-center mb-4">
                <h2>Daily Nutrient Tracker</h2>
```

```
        <button class="btn btn-warning" data-bs-toggle="modal" data-bs-target="#analyzeModal">
          <i class="fas fa-calculator me-2"></i>Analyze Food
        </button>
      </div>

      <!-- Date Selector -->
      <div class="row mb-4">
        <div class="col-md-6 mx-auto">
          <input type="date" id="dateSelector" class="form-control" value="">
        </div>
      </div>

      <!-- Nutrient Progress -->
      <div class="row">
        <div class="col-md-6">
          <div class="nutrient-card">
            <h5>Calories</h5>
            <div class="progress">
              <div id="caloriesProgress" class="progress-bar bg-primary" role="progressbar"></div>
            </div>
            <small><span id="caloriesValue">0</span> / <span id="caloriesGoal">2000</span> kcal</small>
          </div>

          <div class="nutrient-card">
            <h5>Protein</h5>
            <div class="progress">
              <div id="proteinProgress" class="progress-bar bg-success" role="progressbar"></div>
            </div>
            <small><span id="proteinValue">0</span> / <span id="proteinGoal">50</span> g</small>
          </div>

          <div class="nutrient-card">
            <h5>Carbohydrates</h5>
            <div class="progress">
              <div id="carbsProgress" class="progress-bar bg-warning" role="progressbar"></div>
            </div>
            <small><span id="carbsValue">0</span> / <span id="carbsGoal">275</span> g</small>
          </div>

          <div class="nutrient-card">
            <h5>Fats</h5>
            <div class="progress">
              <div id="fatsProgress" class="progress-bar bg-danger" role="progressbar"></div>
            </div>
            <small><span id="fatsValue">0</span> / <span id="fatsGoal">55</span> g</small>
          </div>
        </div>

        <div class="col-md-6">
          <div class="nutrient-card">
            <h5>Today's Meals</h5>
            <div id="mealsList" class="list-group">
              <!-- Meals will be listed here -->
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <!-- Analyze Food Modal -->
  <div class="modal fade" id="analyzeModal" tabindex="-1">
```

```
    <div class="modal-dialog modal-lg">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title">Nutrition Analyzer</h5>
          <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
        </div>
        <div class="modal-body">
          <form id="analyzeForm">
            <div class="mb-3 position-relative">
              <label class="form-label">Food Description</label>
              <textarea class="form-control" id="foodDescription" rows="3"
                placeholder="Enter food items with quantities (e.g., 1 cup rice, 100g chicken breast)"></textarea>
              <!-- Command Palette -->
              <div id="commandPalette" class="command-palette" style="display: none;">
                <ul id="suggestionList" class="list-group">
                  <li class="loading-spinner"><i class="fas fa-spinner fa-spin"></i> Loading...</li>
                </ul>
              </div>
            </div>
            <button type="submit" class="btn btn-warning w-100">
              <i class="fas fa-calculator me-2"></i>Analyze
            </button>
          </form>
          <div id="analysisResults" class="mt-4" style="display: none;">
            <h5>Analysis Results</h5>
            <div class="nutrient-card">
              <div class="row">
                <div class="col-md-6">
                  <p><strong>Calories:</strong> <span id="analyzeCalories">0</span> kcal</p>
                  <p><strong>Protein:</strong> <span id="analyzeProtein">0</span> g</p>
                </div>
                <div class="col-md-6">
                  <p><strong>Carbs:</strong> <span id="analyzeCarbs">0</span> g</p>
                  <p><strong>Fats:</strong> <span id="analyzeFats">0</span> g</p>
                </div>
              </div><!--
              <button class="btn btn-primary w-100 mt-3" id="addToMealBtn">
                <i class="fas fa-plus me-2"></i>Add to Today's Meals
              </button>-->
            </div>
          </div>
        </div>
      </div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
    <script>
      // Nutritionix API configuration
      const NUTRITIONIX_API = {
        APP_ID: '86775e6a', // Replace with your Nutritionix App ID
        API_KEY: '8bf58ac739d4946a7e7ff4c15005813a', // Replace with your Nutritionix API Key
        SEARCH_ENDPOINT: 'https://trackapi.nutritionix.com/v2/search/instant',
        NUTRIENTS_ENDPOINT: 'https://trackapi.nutritionix.com/v2/natural/nutrients'
      };

      // Initialize
      document.addEventListener('DOMContentLoaded', function() {
        const token = localStorage.getItem('token');
        if (!token) {
          window.location.href = '/login.html';
          return;
```

```
        }

        // Set today's date as default
        const today = new Date().toISOString().split('T')[0];
        document.getElementById('dateSelector').value = today;

        // Load initial data
        loadNutrients(today);

        // Add date change listener
        document.getElementById('dateSelector').addEventListener('change', function(e) {
            loadNutrients(e.target.value);
        });

        // Command palette logic with debouncing
        const foodDescriptionInput = document.getElementById('foodDescription');
        const commandPalette = document.getElementById('commandPalette');
        const suggestionList = document.getElementById('suggestionList');
        let debounceTimeout;

        foodDescriptionInput.addEventListener('input', function() {
            clearTimeout(debounceTimeout);
            debounceTimeout = setTimeout(async () => {
                const query = foodDescriptionInput.value.trim().toLowerCase();
                if (query.length < 2) {
                    commandPalette.style.display = 'none';
                    return;
                }

                // Show loading spinner
                suggestionList.innerHTML = '<li class="loading-spinner"><i class="fas fa-spinner fa-spin"></i>
Loading...</li>';
                commandPalette.style.display = 'block';

                try {
                    const response = await fetch(NUTRITIONIX_API.SEARCH_ENDPOINT, {
                        method: 'POST',
                        headers: {
                            'Content-Type': 'application/json',
                            'x-app-id': NUTRITIONIX_API.APP_ID,
                            'x-app-key': NUTRITIONIX_API.API_KEY
                        },
                        body: JSON.stringify({ query })
                    });

                    if (!response.ok) {
                        throw new Error(`HTTP ${response.status}: ${response.statusText}`);
                    }

                    const data = await response.json();
                    const filteredSuggestions = (data.common || []).map(item => ({
                        name: item.food_name,
                        description: `${item.serving_qty} ${item.serving_unit}`,
                        nutrients: {
                            calories: item.nf_calories || 0,
                            protein: item.nf_protein || 0,
                            carbs: item.nf_total_carbohydrate || 0,
                            fats: item.nf_total_fat || 0
                        }
                    }));

                    if (filteredSuggestions.length === 0) {
```

```
            suggestionList.innerHTML = '<li class="list-group-item">No suggestions found</li>';
            commandPalette.style.display = 'block';
            return;
        }

        // Populate suggestion list
        suggestionList.innerHTML = filteredSuggestions.map(food => `
            <li class="list-group-item" data-food='${JSON.stringify(food)}'>
                <strong>${food.name}</strong>
                <small>${food.description}</small>
            </li>
        `).join('');
        commandPalette.style.display = 'block';

        // Handle suggestion click
        suggestionList.querySelectorAll('.list-group-item').forEach(item => {
            item.addEventListener('click', function() {
                const food = JSON.parse(this.dataset.food);
                foodDescriptionInput.value = food.description;
                commandPalette.style.display = 'none';

                // Auto-populate analysis results
                document.getElementById('analysisResults').style.display = 'block';
                document.getElementById('analyzeCalories').textContent = Math.round(food.nutrients.calories);
                document.getElementById('analyzeProtein').textContent = Math.round(food.nutrients.protein);
                document.getElementById('analyzeCarbs').textContent = Math.round(food.nutrients.carbs);
                document.getElementById('analyzeFats').textContent = Math.round(food.nutrients.fats);
                window.lastAnalysis = food.nutrients;
            });
        });
    } catch (error) {
        console.error('Error fetching suggestions:', error);
        suggestionList.innerHTML = '<li class="list-group-item text-danger">Error loading suggestions</li>';
        commandPalette.style.display = 'block';
    }
}, 300);
});

// Keyboard navigation for command palette
foodDescriptionInput.addEventListener('keydown', function(e) {
    const items = suggestionList.querySelectorAll('.list-group-item');
    if (!items.length) return;

    let selectedIndex = Array.from(items).findIndex(item => item.classList.contains('selected'));

    if (e.key === 'ArrowDown') {
        e.preventDefault();
        if (selectedIndex < items.length - 1) {
            items[selectedIndex]?.classList.remove('selected');
            selectedIndex++;
            items[selectedIndex].classList.add('selected');
            items[selectedIndex].scrollIntoView({ block: 'nearest' });
        }
    } else if (e.key === 'ArrowUp') {
        e.preventDefault();
        if (selectedIndex > 0) {
            items[selectedIndex]?.classList.remove('selected');
            selectedIndex--;
            items[selectedIndex].classList.add('selected');
            items[selectedIndex].scrollIntoView({ block: 'nearest' });
        }
    } else if (e.key === 'Enter' && selectedIndex >= 0) {
```

```
            e.preventDefault();
            items[selectedIndex].click();
        }
    });

    // Hide command palette when clicking outside
    document.addEventListener('click', function(e) {
        if (!commandPalette.contains(e.target) && e.target !== foodDescriptionInput) {
            commandPalette.style.display = 'none';
        }
    });
});

async function loadNutrients(date) {
    try {
        const response = await fetch('http://localhost:5000/api/meals', {
            headers: {
                'x-auth-token': localStorage.getItem('token')
            }
        });
        if (!response.ok) {
            throw new Error('Failed to load meals');
        }
        const meals = await response.json();

        // Filter meals for selected date
        const dayMeals = meals.filter(meal => meal.date.startsWith(date));

        // Calculate total nutrients
        let totals = {
            calories: 0,
            protein: 0,
            carbs: 0,
            fats: 0
        };

        dayMeals.forEach(meal => {
            totals.calories += meal.nutrients?.calories || 0;
            totals.protein += meal.nutrients?.protein || 0;
            totals.carbs += meal.nutrients?.carbs || 0;
            totals.fats += meal.nutrients?.fats || 0;
        });

        // Update progress bars
        updateProgress('calories', totals.calories, 2000);
        updateProgress('protein', totals.protein, 50);
        updateProgress('carbs', totals.carbs, 275);
        updateProgress('fats', totals.fats, 55);

        // Update meals list
        const mealsList = document.getElementById('mealsList');
        mealsList.innerHTML = dayMeals.map(meal => `
            <div class="list-group-item">
                <h6>${meal.name}</h6>
                <small>
                    🔥 ${meal.nutrients?.calories || 0} kcal |
                    🥩 ${meal.nutrients?.protein || 0}g |
                    🍚 ${meal.nutrients?.carbs || 0}g |
                    🥑 ${meal.nutrients?.fats || 0}g
                </small>
            </div>
```

```
        `).join('') || '<div class="text-center">No meals recorded for this date</div>';

    } catch (error) {
        console.error('Error loading nutrients:', error);
        alert('Failed to load meals');
    }
}

function updateProgress(nutrient, value, goal) {
    const percentage = Math.min((value / goal) * 100, 100);
    document.getElementById(`${nutrient}Progress`).style.width = `${percentage}%`;
    document.getElementById(`${nutrient}Value`).textContent = Math.round(value);
    document.getElementById(`${nutrient}Goal`).textContent = goal;
}

// Handle food analysis
document.getElementById('analyzeForm').addEventListener('submit', async function(e) {
    e.preventDefault();
    const description = document.getElementById('foodDescription').value;
    const commandPalette = document.getElementById('commandPalette');

    try {
        const response = await fetch(NUTRITIONIX_API.NUTRIENTS_ENDPOINT, {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'x-app-id': NUTRITIONIX_API.APP_ID,
                'x-app-key': NUTRITIONIX_API.API_KEY
            },
            body: JSON.stringify({ query: description })
        });

        if (!response.ok) {
            throw new Error(`HTTP ${response.status}: ${response.statusText}`);
        }

        const data = await response.json();
        if (data.foods && data.foods.length > 0) {
            const nutrients = data.foods.reduce((acc, food) => ({
                calories: acc.calories + (food.nf_calories || 0),
                protein: acc.protein + (food.nf_protein || 0),
                carbs: acc.carbs + (food.nf_total_carbohydrate || 0),
                fats: acc.fats + (food.nf_total_fat || 0)
            }), { calories: 0, protein: 0, carbs: 0, fats: 0 });

            // Show results
            document.getElementById('analysisResults').style.display = 'block';
            document.getElementById('analyzeCalories').textContent = Math.round(nutrients.calories);
            document.getElementById('analyzeProtein').textContent = Math.round(nutrients.protein);
            document.getElementById('analyzeCarbs').textContent = Math.round(nutrients.carbs);
            document.getElementById('analyzeFats').textContent = Math.round(nutrients.fats);
            window.lastAnalysis = nutrients;

            // Hide command palette
            commandPalette.style.display = 'none';
        } else {
            alert('No nutritional data found for the provided description');
        }
    } catch (error) {
        console.error('Error analyzing food:', error);
        alert(error.message.includes('429') ? 'API rate limit exceeded. Please try again later.' : 'Failed to analyze food');
    }
```

```
    });

    // Handle adding analyzed food to meals
    document.getElementById('addToMealBtn').addEventListener('click', async function() {
        if (!window.lastAnalysis) {
            alert('Please analyze food first');
            return;
        }

        const description = document.getElementById('foodDescription').value;
        try {
            const response = await fetch('http://localhost:5000/api/meals', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                    'x-auth-token': localStorage.getItem('token')
                },
                body: JSON.stringify({
                    name: description,
                    date: document.getElementById('dateSelector').value,
                    nutrients: window.lastAnalysis
                })
            });

            if (response.ok) {
                // Reload nutrients
                loadNutrients(document.getElementById('dateSelector').value);
                // Clear form and hide results
                document.getElementById('analyzeForm').reset();
                document.getElementById('analysisResults').style.display = 'none';
                // Close modal
                bootstrap.Modal.getInstance(document.getElementById('analyzeModal')).hide();
                delete window.lastAnalysis;
            } else {
                const data = await response.json();
                alert(data.message || 'Failed to add meal');
            }
        } catch (error) {
            console.error('Error adding meal:', error);
            alert('Failed to add meal');
        }
    });
  </script>
</body></html>
```

MealPlan

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Meal Planner & Nutrition Tracker</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
  <style>
    body {
      background: linear-gradient(to right, #1a1a19, #e2d1c3);
      background: url('images/image2.jpg') no-repeat center center/cover;
      font-family: 'Segoe UI', sans-serif;
    }
    .navbar-dark .navbar-nav .nav-link {
      color: #ffffff;
```

```
    }
    .navbar-dark .navbar-nav .nav-link:hover {
      color: #ffc107;
    }
    .container-box {
      background: #ffffff;
      border-radius: 20px;
      box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);
      padding: 30px;
      margin-top: 50px;
    }
    h2 {
      font-weight: 700;
      margin-bottom: 20px;
      color: #333;
    }
    label {
      font-weight: 500;
    }
    .btn-custom {
      background-color: #ffc107;
      border: none;
      color: white;
      font-weight: bold;
    }
    .btn-custom:hover {
      background-color: #e0a800;
    }
    .recommendation-box {
      max-height: 500px;
      overflow-y: auto;
      background: #f9f9f9;
      border-radius: 15px;
      padding: 20px;
    }
    .list-group-item {
      border-radius: 10px;
      margin-bottom: 10px;
      background: #ffffff;
      box-shadow: 0 2px 5px rgba(0,0,0,0.05);
    }
    .list-group-item h5 {
      margin-bottom: 5px;
      color: #333;
    }
    .list-group-item small {
      color: #6c757d;
    }
    .loading-spinner {
      text-align: center;
      padding: 10px;
      color: #6c757d;
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <a class="navbar-brand fw-bold" href="#">Meal Planner 🥗</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
```

```
    <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="dashboard.html">🧭 Dashboard</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="meal_plan.html">🍽️ My Meals</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="shopping-list.html">🛒 Shopping List</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="contact.html">📞 Contact</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="profile.html">👤 Profile</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="about.html">ℹ About</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

<div class="container">
  <div class="row g-4">
    <!-- Meal Entry Form -->
    <div class="col-md-6">
      <div class="container-box">
        <h2>Add Your Meal</h2>
        <form id="mealForm">
          <div class="mb-3">
            <label for="slot">Meal Slot</label>
            <select class="form-select" id="slot" required>
              <option>Breakfast</option>
              <option>Lunch</option>
              <option>Dinner</option>
              <option>Snack</option>
            </select>
          </div>
          <div class="mb-3">
            <label for="mealName">Meal Name</label>
            <input type="text" class="form-control" id="mealName" placeholder="e.g., Grilled Chicken Salad" required>
          </div>
          <div class="mb-3">
            <label for="calories">Calories (kcal)</label>
            <input type="number" class="form-control" id="calories" required>
          </div>
          <div class="mb-3">
            <label for="protein">Protein (g)</label>
            <input type="number" class="form-control" id="protein" required>
          </div>
          <div class="mb-3">
            <label for="carbs">Carbohydrates (g)</label>
            <input type="number" class="form-control" id="carbs" required>
          </div>
          <div class="mb-3">
```

```
          <label for="fats">Fats (g)</label>
          <input type="number" class="form-control" id="fats" required>
        </div>
        <button type="submit" class="btn btn-custom w-100">Add Meal</button>
      </form>
    </div>
  </div>

  <!-- Meal Recommendations -->
  <div class="col-md-6">
    <div class="container-box">
      <h2>Meal Recommendations 🍽️</h2>
      <div class="mb-3">
        <label for="mealType">Meal Type</label>
        <select class="form-select" id="mealType">
          <option value="">Any</option>
          <option value="breakfast">Breakfast</option>
          <option value="lunch">Lunch</option>
          <option value="dinner">Dinner</option>
          <option value="snack">Snack</option>
        </select>
      </div>
      <div class="mb-3">
        <label for="dietType">Diet Type</label>
        <select class="form-select" id="dietType">
          <option value="">Any</option>
          <option value="vegetarian">Vegetarian</option>
          <option value="vegan">Vegan</option>
          <option value="keto">Ketogenic</option>
          <option value="balanced">Balanced</option>
          <option value="gluten-free">Gluten Free</option>
          <option value="low-carb">Low Carb</option>
        </select>
      </div>
      <div class="mb-3">
        <label for="calorieRange">Calorie Range</label>
        <select class="form-select" id="calorieRange">
          <option value="">Any</option>
          <option value="low">Below 300 kcal</option>
          <option value="medium">300-600 kcal</option>
          <option value="high">Above 600 kcal</option>
        </select>
      </div>
      <button class="btn btn-warning w-100 mb-3" onclick="getRecommendations()">Get Recommendations</button>
      <div class="recommendation-box">
        <ul class="list-group" id="recommendationList">
          <!-- Recommendations will appear here -->
        </ul>
        <div class="text-center mt-2">
          <small>Powered by <a href="https://spoonacular.com/food-api" target="_blank">Spoonacular</a></small>
        </div>
      </div>
    </div>
  </div>
 </div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
<script>
 // Spoonacular API configuration
 const SPOONACULAR_API = {
  API_KEY: '325da1d14441419ab0277b594eed728a', // Replace with your Spoonacular API Key
```

```
  SEARCH_ENDPOINT: 'https://api.spoonacular.com/recipes/complexSearch',
  INFO_ENDPOINT: 'https://api.spoonacular.com/recipes'
};

// Check authentication
document.addEventListener('DOMContentLoaded', function() {
  const token = localStorage.getItem('token');
  if (!token) {
    window.location.href = '/login.html';
    return;
  }
  loadMeals();
});

// Load existing meals
async function loadMeals() {
  try {
    const response = await fetch('http://localhost:5000/api/meals', {
      headers: {
        'x-auth-token': localStorage.getItem('token')
      }
    });
    if (!response.ok) {
      throw new Error(`HTTP ${response.status}: ${response.statusText}`);
    }
    const meals = await response.json();
    displayMeals(meals);
  } catch (error) {
    console.error('Error loading meals:', error);
    alert('Failed to load meals');
  }
}

// Display meals in the recommendations section
function displayMeals(meals) {
  const recommendationList = document.getElementById('recommendationList');
  recommendationList.innerHTML = meals.map(meal => `
    <li class="list-group-item">
      <div class="d-flex justify-content-between align-items-center">
        <div>
          <h5>${meal.name}</h5>
          <small>${meal.type} - ${meal.nutrients.calories} kcal</small>
        </div>
        <button class="btn btn-danger btn-sm" onclick="deleteMeal('${meal._id}')">Delete</button>
      </div>
    </li>
  `).join('');
}

// Handle meal form submission
document.getElementById('mealForm').addEventListener('submit', async function(e) {
  e.preventDefault();

  const mealData = {
    name: document.getElementById('mealName').value,
    type: document.getElementById('slot').value.toLowerCase(),
    date: new Date().toISOString(),
    nutrients: {
      calories: Number(document.getElementById('calories').value),
      protein: Number(document.getElementById('protein').value),
      carbs: Number(document.getElementById('carbs').value),
      fats: Number(document.getElementById('fats').value)
```

```
    }
  };

  try {
    const response = await fetch('http://localhost:5000/api/meals', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'x-auth-token': localStorage.getItem('token')
      },
      body: JSON.stringify(mealData)
    });

    if (response.ok) {
      alert('Meal added successfully!');
      this.reset();
      loadMeals();
    } else {
      const data = await response.json();
      alert(data.message || 'Error adding meal');
    }
  } catch (error) {
    console.error('Error:', error);
    alert('Error adding meal');
  }
});

// Delete a meal
async function deleteMeal(mealId) {
  if (!confirm('Are you sure you want to delete this meal?')) return;

  try {
    const response = await fetch(`http://localhost:5000/api/meals/${mealId}`, {
      method: 'DELETE',
      headers: {
        'x-auth-token': localStorage.getItem('token')
      }
    });

    if (response.ok) {
      alert('Meal deleted successfully!');
      loadMeals();
    } else {
      const data = await response.json();
      alert(data.message || 'Error deleting meal');
    }
  } catch (error) {
    console.error('Error:', error);
    alert('Error deleting meal');
  }
}

// Get meal recommendations from Spoonacular API
async function getRecommendations() {
  const type = document.getElementById('mealType').value;
  const diet = document.getElementById('dietType').value;
  const calRange = document.getElementById('calorieRange').value;
  const list = document.getElementById('recommendationList');

  // Show loading spinner
  list.innerHTML = '<li class="loading-spinner"><i class="fas fa-spinner fa-spin"></i> Loading recommendations...</li>';
```

```
    // Build query parameters for complex search
    const searchParams = new URLSearchParams({
      apiKey: SPOONACULAR_API.API_KEY,
      number: '10', // Limit to 10 results to stay within free plan quota
      addRecipeNutrition: 'true', // Include nutrition data directly
      sort: 'random' // Randomize results
    });

    // Add filters
    if (type) {
      searchParams.append('type', type);
    }
    if (diet) {
      searchParams.append('diet', diet);
    }
    if (calRange) {
      if (calRange === 'low') {
        searchParams.append('maxCalories', '300');
      } else if (calRange === 'medium') {
        searchParams.append('minCalories', '300');
        searchParams.append('maxCalories', '600');
      } else if (calRange === 'high') {
        searchParams.append('minCalories', '600');
      }
    }

    try {
      // Step 1: Search for recipes
      const searchResponse = await fetch(`${SPOONACULAR_API.SEARCH_ENDPOINT}?${searchParams.toString()}`,
{
        headers: {
          'Accept': 'application/json'
        }
      });

      if (!searchResponse.ok) {
        throw new Error(`HTTP ${searchResponse.status}: ${searchResponse.statusText}`);
      }

      const searchData = await searchResponse.json();
      const recipes = searchData.results || [];

      // Step 2: Map recipes to required format
      const filteredMeals = recipes.map(recipe => ({
        name: recipe.title,
        type: type || recipe.dishTypes?.find(type => ['breakfast', 'lunch', 'dinner', 'snack'].includes(type)) || 'unknown',
        diet: diet || recipe.diets?.find(d => ['vegetarian', 'vegan', 'ketogenic', 'gluten free', 'low carb',
'balanced'].includes(d.toLowerCase())) || 'balanced',
        nutrients: {
          calories: Math.round(recipe.nutrition?.nutrients.find(n => n.name === 'Calories')?.amount || 0),
          protein: Math.round(recipe.nutrition?.nutrients.find(n => n.name === 'Protein')?.amount || 0),
          carbs: Math.round(recipe.nutrition?.nutrients.find(n => n.name === 'Carbohydrates')?.amount || 0),
          fats: Math.round(recipe.nutrition?.nutrients.find(n => n.name === 'Fat')?.amount || 0)
        }
      }));

      // Display recommendations
      list.innerHTML = '';
      if (filteredMeals.length === 0) {
        list.innerHTML = "<li class='list-group-item text-danger'>No meals found for your selection.</li>";
      } else {
        filteredMeals.forEach(meal => {
```

```
          const li = document.createElement('li');
          li.className = 'list-group-item';
          li.innerHTML = `
            <h5>${meal.name}</h5>
            <small>
              ${meal.type.charAt(0).toUpperCase() + meal.type.slice(1)} | ${meal.diet.charAt(0).toUpperCase() +
meal.diet.slice(1)}<br>
              Calories: ${meal.nutrients.calories} kcal | Protein: ${meal.nutrients.protein}g | Carbs: ${meal.nutrients.carbs}g |
Fats: ${meal.nutrients.fats}g
            </small>
          `;
          list.appendChild(li);
        });
      }
    } catch (error) {
      console.error('Error fetching recommendations:', error);
      list.innerHTML = `<li class='list-group-item text-danger'>Error loading recommendations:
${error.message.includes('402') ? 'API quota exceeded. Please try again tomorrow.' : error.message.includes('429') ? 'API rate
limit exceeded. Please try again later.' : 'Please try again.'}</li>`;
    }
  }
 </script>
</body>
</html>
```

Aboutpage:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>About - Meal Planner</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      background: url('images/image2.jpg') no-repeat center center/cover;
      min-height: 100vh;
    }
    .navbar {
      background-color: #343a40 !important;
    }
    .navbar-brand, .nav-link {
      color: #ffffff !important;
    }
    .container-box {
      background: rgba(255, 255, 255, 0.9);
      border-radius: 15px;
      padding: 30px;
      margin-top: 30px;
      box-shadow: 0 0 15px rgba(0,0,0,0.1);
    }
    .feature-card {
      border: none;
      border-radius: 10px;
      box-shadow: 0 2px 5px rgba(0,0,0,0.1);
      margin-bottom: 20px;
      transition: transform 0.3s;
    }
    .feature-card:hover {
```

```
            transform: translateY(-5px);
        }
        .feature-icon {
            font-size: 2rem;
            margin-bottom: 15px;
            color: #ffc107;
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark">
        <div class="container">
            <a class="navbar-brand" href="dashboard.html">
                <i class="fas fa-utensils me-2"></i>Meal Planner
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <a class="nav-link" href="dashboard.html">🧭 Dashboard</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="meal_plan.html">🍽️ My Meals</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="shopping-list.html">🛒 Shopping List</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="contact.html">📞 Contact</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="profile.html">👤 Profile</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="about.html">ℹ️ About</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="container">
        <div class="container-box">
            <h1 class="text-center mb-5">About Meal Planner & Nutrition Tracker</h1>

            <div class="row mb-5">
                <div class="col-md-8 mx-auto">
                    <p class="lead text-center">
                        Your personal assistant for maintaining a healthy lifestyle through proper meal planning and nutrition
tracking.
                    </p>
                </div>
            </div>

            <div class="row">
                <div class="col-md-4">
```

```
                <div class="card feature-card">
                    <div class="card-body text-center">
                        <i class="fas fa-utensils feature-icon"></i>
                        <h4>Meal Planning</h4>
                        <p>Plan your meals ahead of time with our easy-to-use meal planner. Track your breakfast, lunch, dinner, and snacks.</p>
                    </div>
                </div>
            </div>

            <div class="col-md-4">
                <div class="card feature-card">
                    <div class="card-body text-center">
                        <i class="fas fa-chart-pie feature-icon"></i>
                        <h4>Nutrient Tracking</h4>
                        <p>Monitor your daily intake of calories, protein, carbs, and fats. Stay on top of your nutrition goals.</p>
                    </div>
                </div>
            </div>

            <div class="col-md-4">
                <div class="card feature-card">
                    <div class="card-body text-center">
                        <i class="fas fa-shopping-cart feature-icon"></i>
                        <h4>Shopping Lists</h4>
                        <p>Create and manage your grocery shopping lists. Never forget what you need to buy for your planned meals.</p>
                    </div>
                </div>
            </div>
        </div>

        <div class="row mt-5">
            <div class="col-md-8 mx-auto">
                <h3 class="text-center mb-4">How It Works</h3>
                <div class="card">
                    <div class="card-body">
                        <ol class="mb-0">
                            <li class="mb-3">Create your account and set your nutrition goals</li>
                            <li class="mb-3">Plan your meals using our comprehensive meal planner</li>
                            <li class="mb-3">Track your daily nutrient intake</li>
                            <li class="mb-3">Generate shopping lists for your planned meals</li>
                            <li>Monitor your progress through detailed analytics</li>
                        </ol>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
<script>
    // Check authentication
    document.addEventListener('DOMContentLoaded', function() {
        const token = localStorage.getItem('token');
        if (!token) {
            window.location.href = '/login.html';
            return;
        }
    });
</script>
```

```
</body>
</html>
```

Contact.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contact Us | Meal Planner</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <style>
    body {
      background: url('images/image2.jpg') no-repeat center center/cover;
      font-family: 'Segoe UI', sans-serif;
    }

    .container-box {
      background: #ffffff;
      border-radius: 20px;
      box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);
      padding: 30px;
      margin-top: 50px;
    }

    h2 {
      font-weight: 700;
      color: #333;
    }

    label {
      font-weight: 500;
    }

    .btn-custom {
      background-color: #ffc107;
      border: none;
      color: white;
      font-weight: bold;
    }

    .btn-custom:hover {
      background-color: #e6b800;
    }
  </style>
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-black">
    <div class="container-fluid">
      <a class="navbar-brand fw-bold" href="#">Meal Planner 🥗</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="dashboard.html">🧭 Dashboard</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="meal_plan.html">🍽️ My Meals</a>
```

```
        </li>
        <li class="nav-item">
          <a class="nav-link" href="nutrient-tracker.html">📈 Nutrient Tracker</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="shopping-list.html">🛒 Shopping List</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="contact.html">📞 Contact</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="profile.html">👤 Profile</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="about.html">ℹ About</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

<div class="container">
  <div class="container-box">
    <h2>📞 Get in Touch</h2>
    <form>
      <div class="mb-3">
        <label for="name">Your Name</label>
        <input type="text" class="form-control" id="name" placeholder="Full Name">
      </div>
      <div class="mb-3">
        <label for="email">Your Email</label>
        <input type="email" class="form-control" id="email" placeholder="example@example.com">
      </div>
      <div class="mb-3">
        <label for="message">Your Message</label>
        <textarea class="form-control" id="message" rows="5" placeholder="Type your message here..."></textarea>
      </div>
      <button type="submit" class="btn btn-custom">Send Message</button>
    </form>
  </div>
</div>
</body>
</html>
```

Output:

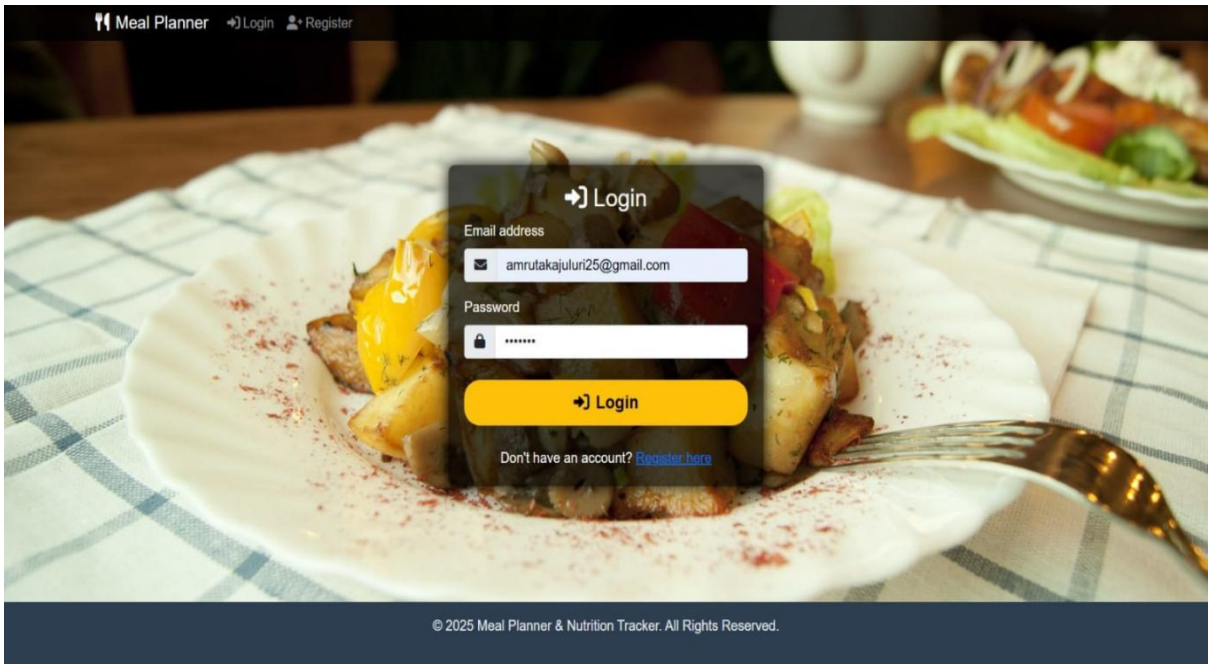


Fig:1. The Login page provides a secure interface for users to access their personalized meal plans, nutrient tracking, and shopping lists. It features user authentication with email and password, ensuring that only authorized users can interact with the application.

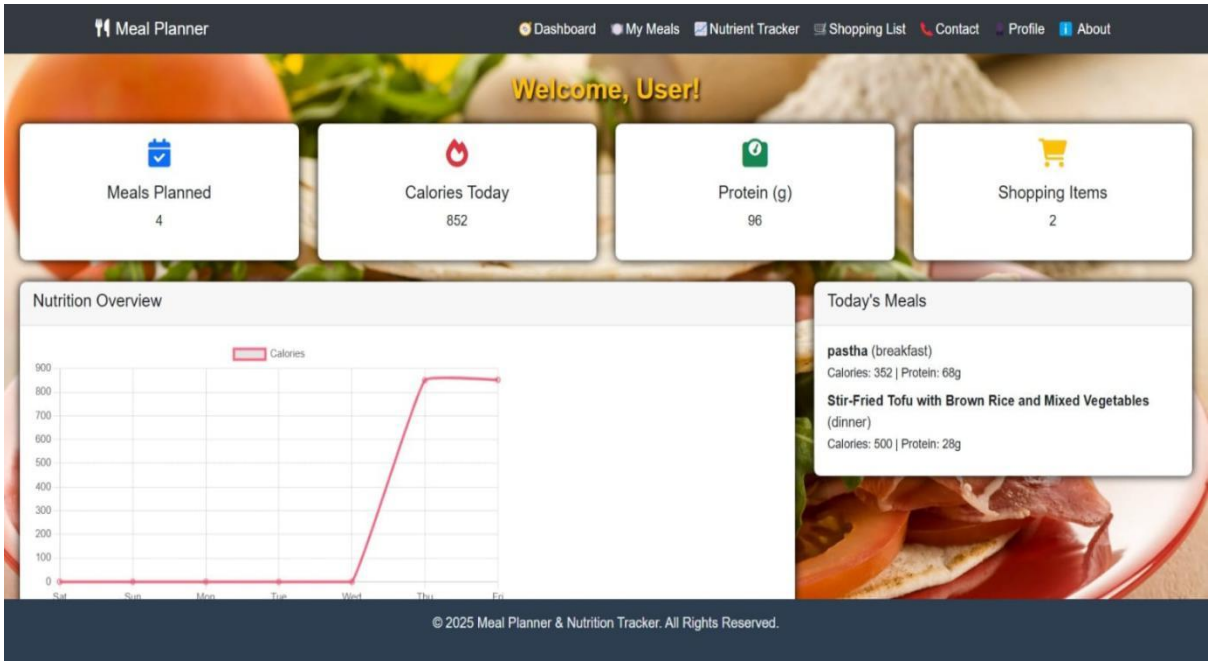


Fig2. The Dashboard serves as the central hub for users to view and manage their meal plans, track nutrient intake, and access shopping lists. It features a nutrition tracking chart that visually displays the breakdown of carbohydrates, proteins, fats, and other key nutrients to help users monitor their daily goals.

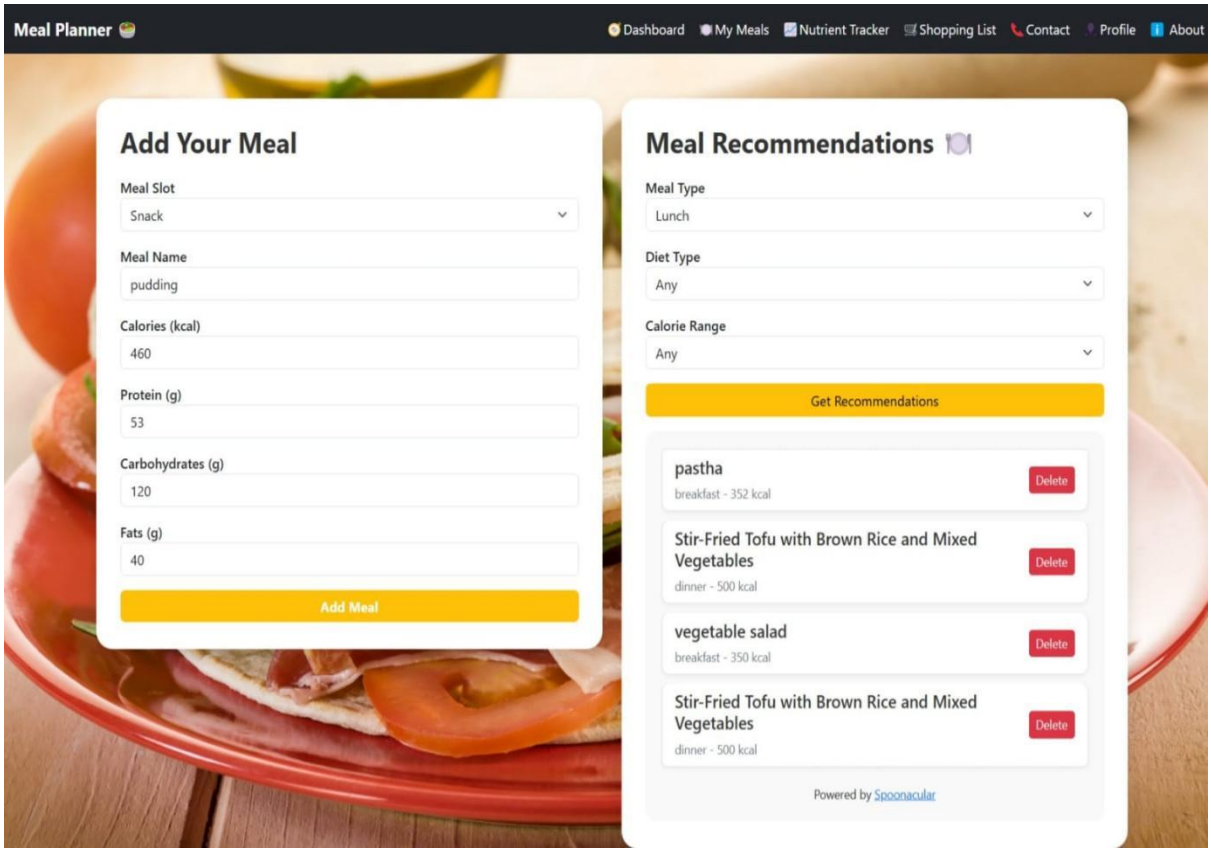


Fig3. The Meal Planner page allows users to create, customize, and organize their daily or weekly meal plans based on nutritional goals and personal preferences. It offers an intuitive interface to add meals, view nutrient breakdowns, and generate corresponding shopping lists.

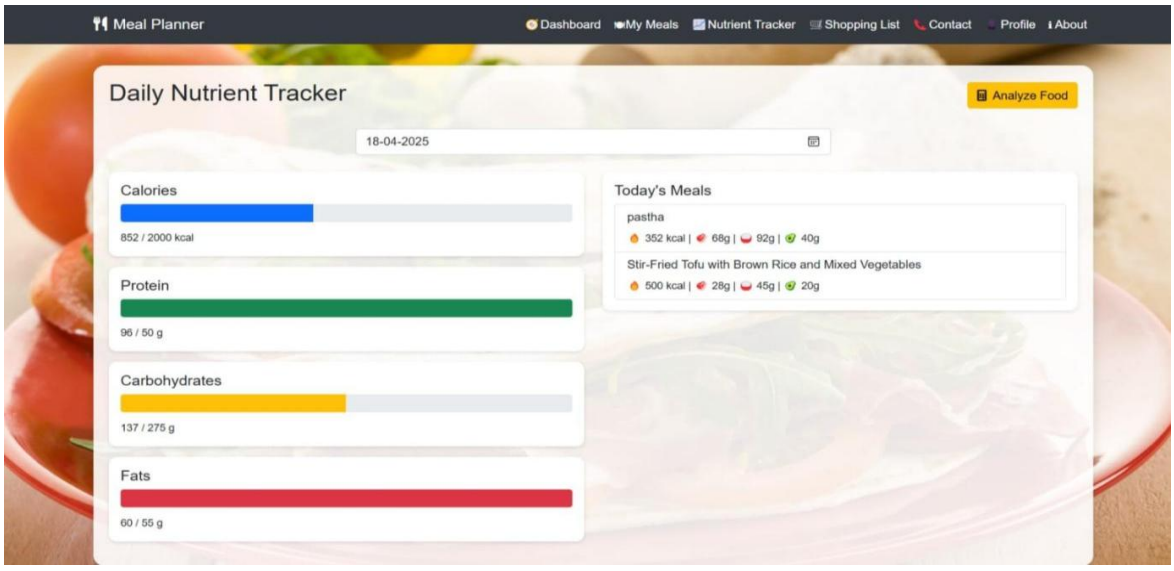


Fig4: The Daily Nutrient Tracker helps users monitor their daily intake of key nutrients such as carbohydrates, proteins, fats, and calories. It provides real-time visual feedback through charts and progress indicators, ensuring users stay aligned with their health and dietary goals.

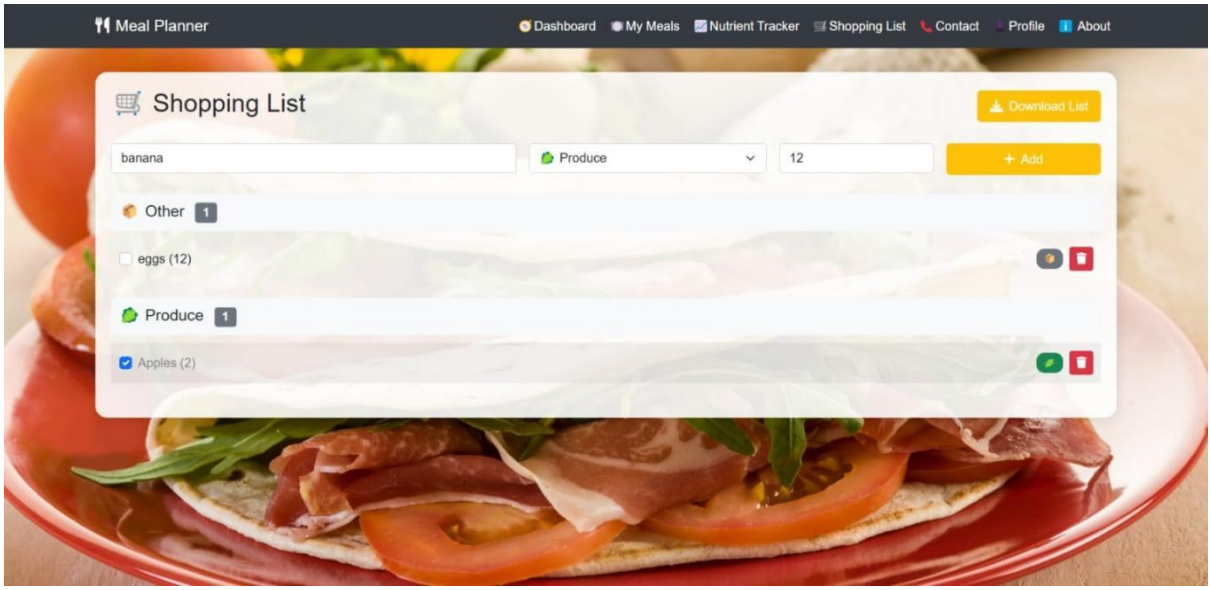


Fig5 . The Shopping List page automatically generates a list of ingredients based on the user's meal plans, helping them organize grocery shopping efficiently. Users can view, add, edit, or check off items to ensure they have all necessary ingredients for their planned meals.

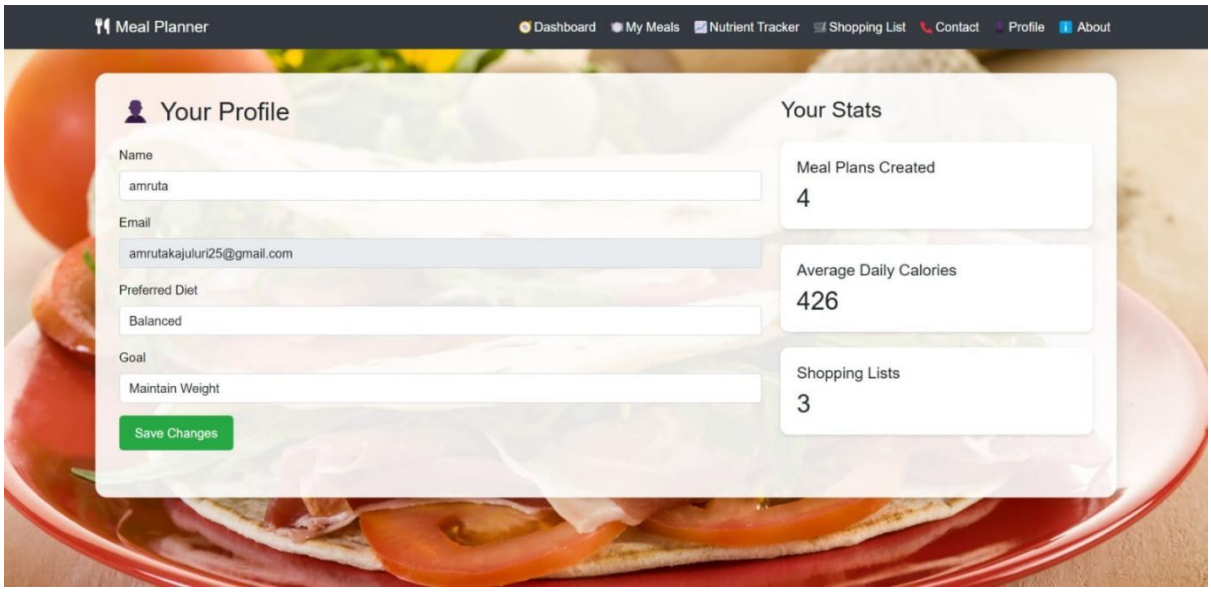


Fig6: The User Profile page allows users to manage their personal information, including name, email, dietary preferences, and goals. It provides options to update account settings and personalize the meal planning experience based on individual needs.

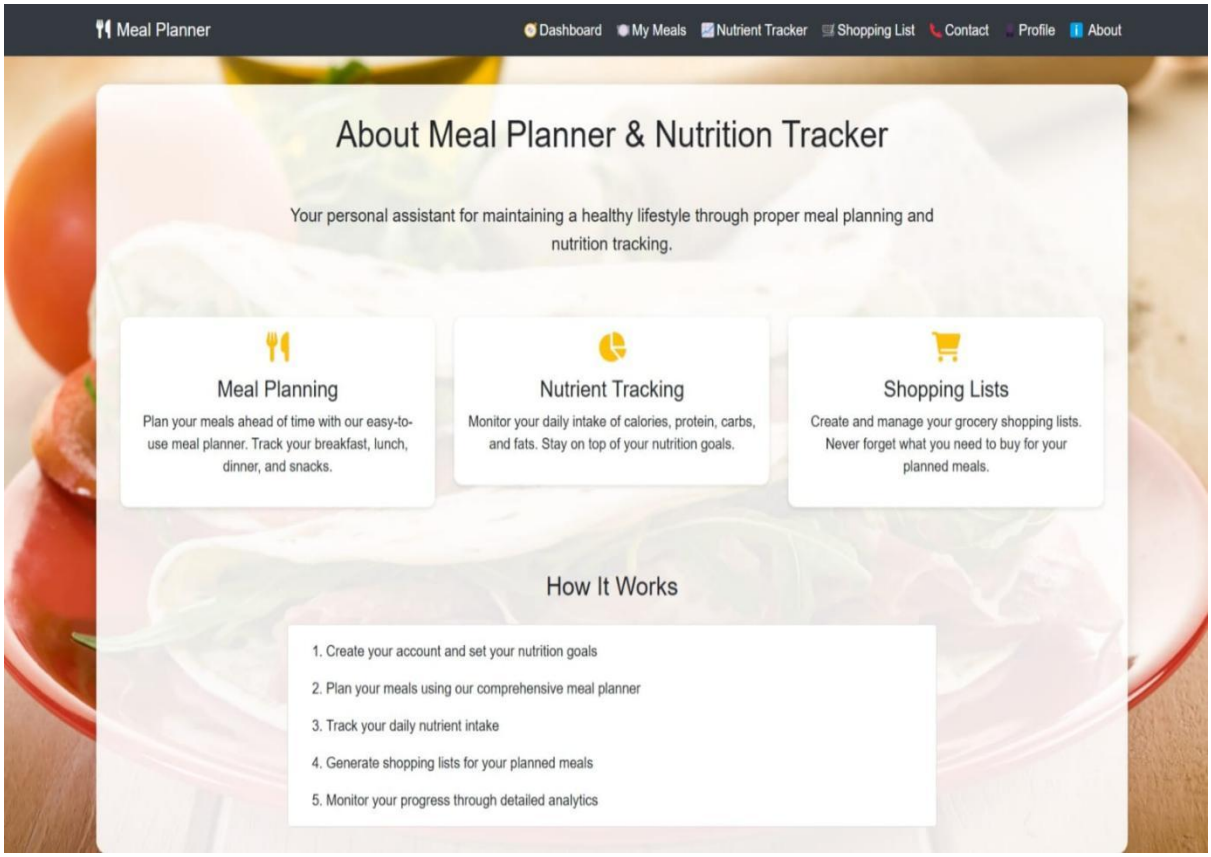
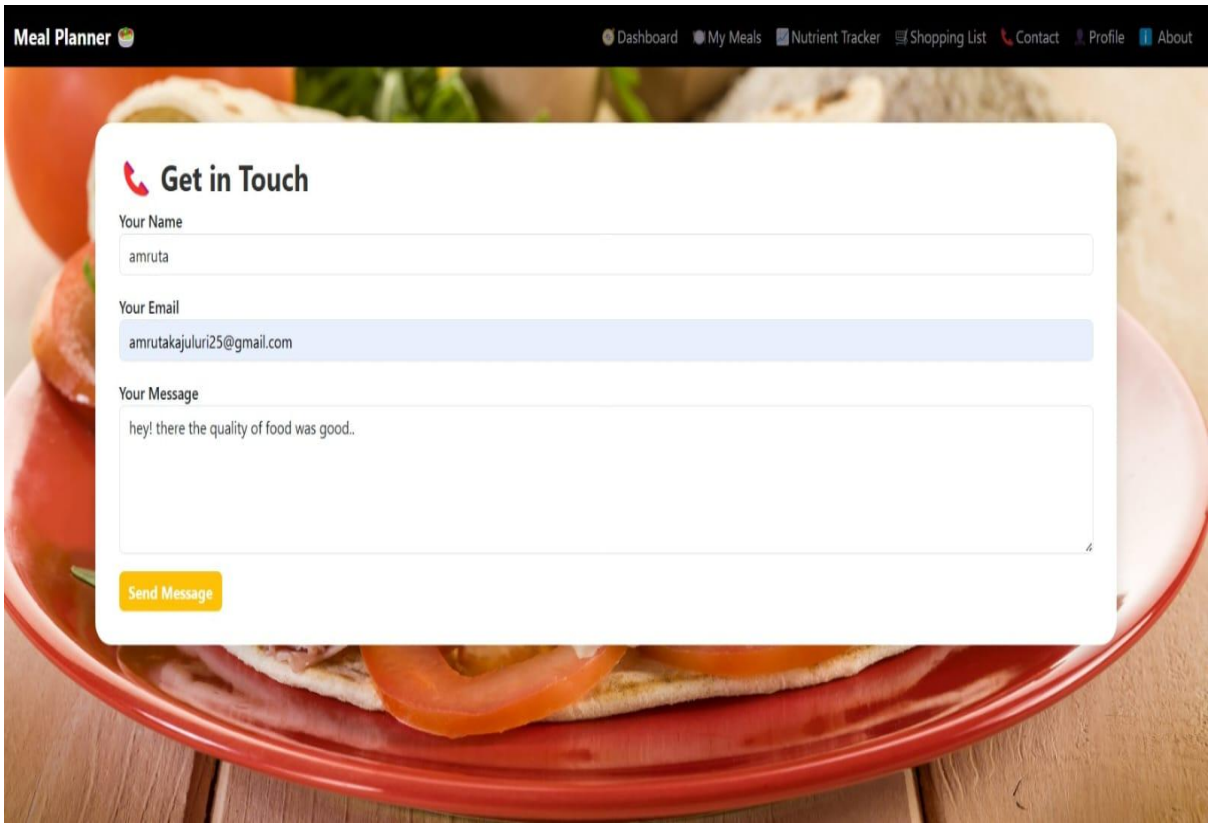


Fig7: The About page provides an overview of the Meal Planner & Nutrition Tracker application, its purpose, and the team behind its development. It highlights the project's mission to promote healthier eating habits through technology-driven meal planning and nutrient tracking.



Database:

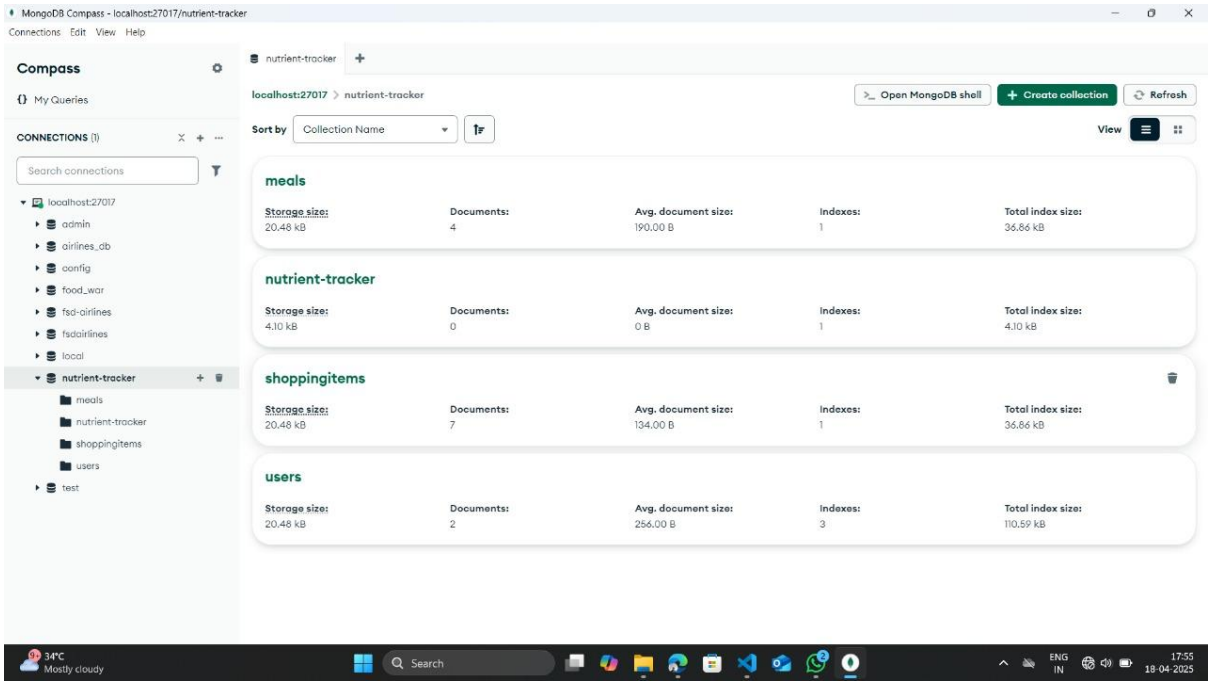


Fig9

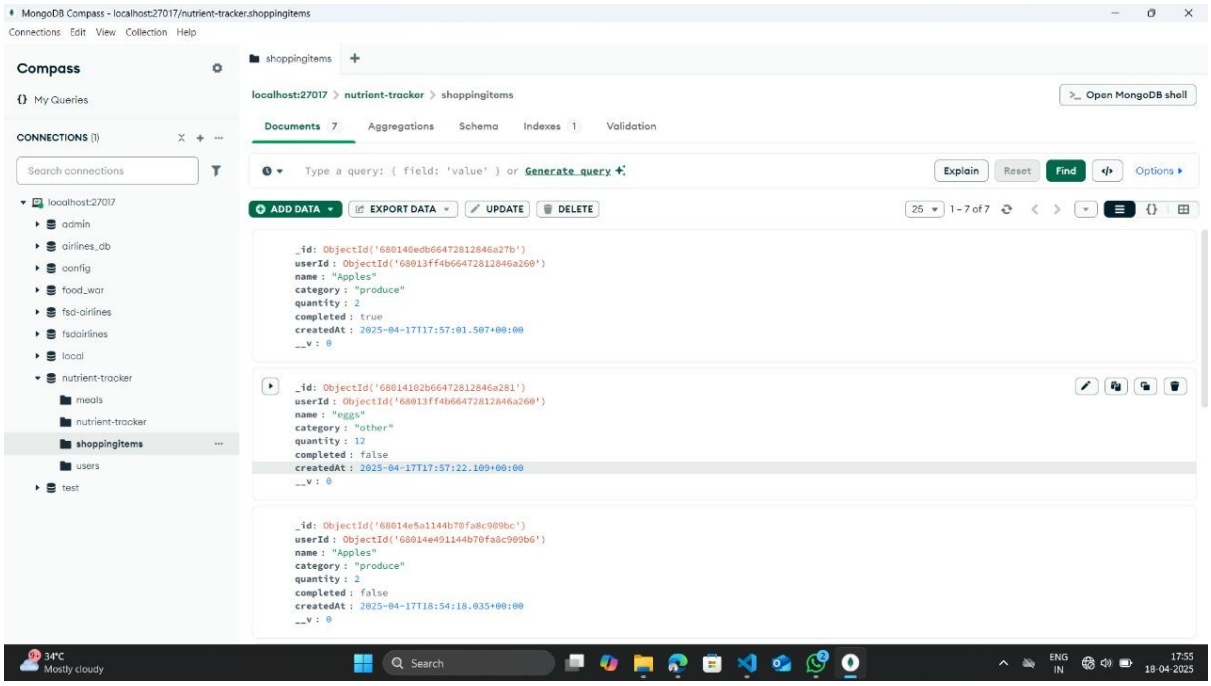


Fig10

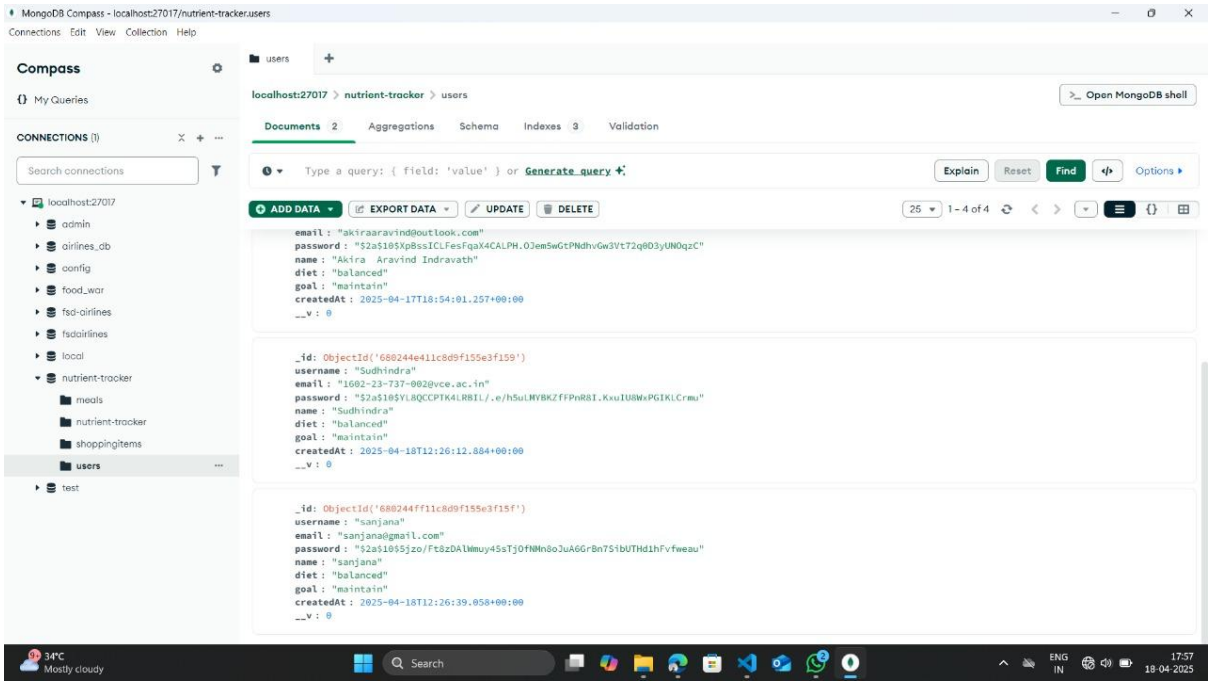


fig11

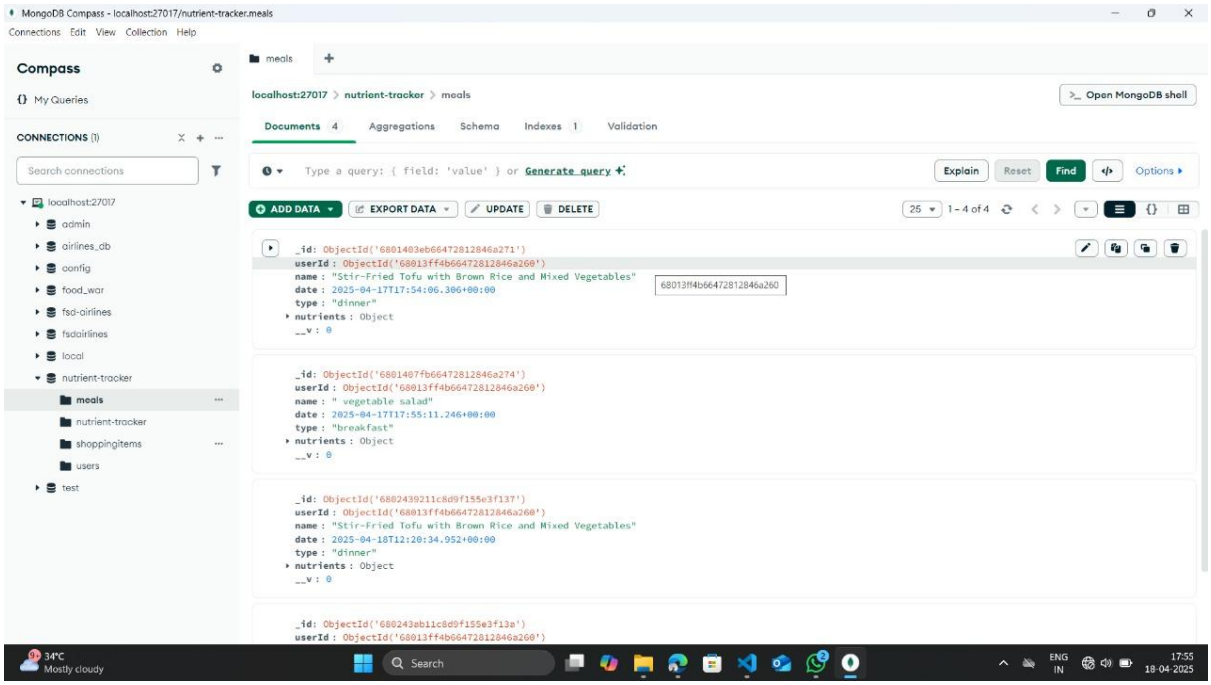


Fig12

The **users collection** stores user account details, including name, email, password, dietary preferences, and personal goals. The **meals collection** manages planned meals for users, including meal names, ingredients, nutritional breakdowns, and associated dates. The **shopping items collection** keeps track of grocery items derived from meal plans, allowing users to organize and manage their shopping lists efficiently. Each collection is designed to support real-time updates and maintain relational consistency between user profiles, meal entries, and shopping data.

Conclusion

The Meal Planner & Nutrition Tracker project was undertaken to design a full-stack web application that addresses real-world needs such as meal planning, nutrient tracking, shopping list management, and user profile handling. Through this project, we applied core software engineering principles and leveraged modern development tools to construct an end-to-end system that is scalable, user-friendly, and rich in functionality.

This project enabled us to gain practical experience working with frontend and backend technologies, understanding layered architectures, integrating external APIs, and managing database interactions using MongoDB. Features such as JWT-based authentication, dynamic meal and nutrient management, and real-time API communication helped us simulate realistic scenarios and build a prototype that can be extended into a fully production-ready solution.

Beyond technical implementation, the project enhanced our understanding of teamwork, API integration, version control with Git, and the iterative nature of full-stack development. Each aspect—from user interface design to server-side logic and database structuring—offered valuable hands-on learning and improved our ability to deliver complete, user-centric applications.

Future Work

Although the current version of the Meal Planner & Nutrition Tracker application successfully implements the core functionalities of meal planning, nutrient tracking, and shopping list management, there remains significant scope for future enhancements and feature extensions. Some of the planned or potential improvements include:

- **Mobile Application Development:** Building a dedicated Android/iOS app using frameworks like Flutter or React Native to allow users to plan meals and track nutrients on the go.
- **Advanced Nutrient Analytics:** Adding detailed charts and insights such as macronutrient distribution, calorie balance tracking, and weekly/monthly health reports.
- **Barcode Scanner Integration:** Allowing users to scan food products for automatic nutrition entry using their mobile devices.
- **Recipe Recommendation Engine:** Suggesting meal recipes based on user goals (e.g., weight loss, muscle gain, balanced diet) and available ingredients.
- **Integration with Fitness Apps:** Connecting with fitness trackers like Fitbit, Google Fit, or Apple Health to sync physical activity and adjust meal plans dynamically.
- **AI-Based Meal Suggestions:** Using machine learning algorithms to provide personalized meal plans based on user preferences, restrictions, and past behavior.
- **Voice Assistance:** Implementing a voice-based assistant that allows users to log meals, request nutritional information, or generate shopping lists using simple voice commands, enhancing accessibility and convenience.
- **Social Features:** Enabling users to share meal plans, favorite recipes, and grocery lists with friends or groups.
- **Multilingual and Accessibility Support:** Expanding the platform to support multiple languages and accessibility standards to reach a broader audience.

REFERENCES

1. **MongoDB Documentation** – https://www.mongodb.com/docs/
2. **Bootstrap Framework** – https://getbootstrap.com/
3. **Node.js Documentation** – https://nodejs.org/docs/latest/api/http.html
4. **Express.js Documentation** – https://expressjs.com/
5. **HTML/CSS/JS Tutorials** – https://www.w3schools.com/