

Diet Management System [.NET + React] [Lightweight]

Objective

The primary goal of this project is to develop a comprehensive platform that allows dietitians to manage their clients and diet plans efficiently. Administrators can manage dietitians, ensuring robust oversight and system integrity.

Description

This diet management system provides a platform where dietitians can create profiles for their clients, manage diet plans, and monitor client progress.

Administrators can manage dietitian profiles. The system will support multiple roles, with specific permissions and capabilities for each.

Expected Outcomes

Upon completion of the project, the system will allow:

- Administrators to manage dietitians, clients and diet plans.
- Dietitians to manage clients and diet plans.
- Diet Plans to be created, updated, and monitored with detailed meal scheduling. Each diet plan includes essential information such as a title, start and end dates, and initial weight, along with the timing and contents of meals throughout the day. Meals are detailed with their own titles, start and end times, and what will be eaten. This structure is designed to help the individual following the diet program to implement it correctly and effectively. Having a detailed plan facilitates users in reaching their health goals while maintaining regular and controlled food intake.

Description

Deliverables

- **Frontend:** A user interface built with Next.js or another React-based framework, providing pages for client management, diet plan creation, and administrator controls.
- **Backend:** A set of APIs designed with .NET, following RESTful standards, secured with .NET Identity for authentication and role-based authorization, and documented using Swagger or OpenAPI.
- **Infrastructure:** Deployment using Docker, managed via CI/CD pipelines in GitHub Actions, and hosted on AWS (AWS is optional, hosting is preferred, local development ready docker compose required).

Description

Roles and Permissions

Admin:

- Can perform CRUD operations on dietitians.
- Can perform CRUD operations on clients.
- Can perform CRUD operations on diet plans.

Dietitian:

- Can perform CRUD operations on clients.
- Can perform CRUD operations on diet plans.

System Requirements

Frontend

- A clean UI.
- Role based interfaces for admins and dietitians.

- Client-side validation.

Backend

- RESTful API.
- Authentication with .NET Identity Framework.
- Role-based authorization.
- Adherence to DRY principles.
- Fluent validation for data integrity.

Bonus Enhancements

- Implementation of CI/CD pipelines via GitHub Actions.
- Comprehensive unit and integration tests.
- Use AWS for Infrastructure. 🌟
- Use Terraform for building infrastructure as a code. 🌟

Documentation

A detailed [README](#) with instructions on setting up, testing, and deploying the system.