Report

Student number: 33782985

Outline

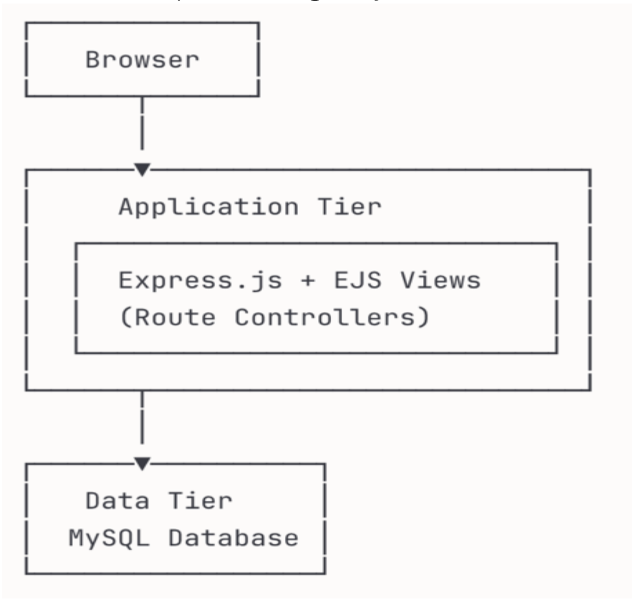
YakTime is a web-based healthcare management application for tracking supplements and monitoring weight. The application provides an interface for recording health data, setting goals, and accessing nutritional information through the USDA FoodData Central API. Built with Node.js, Express, MySQL, and EJS, YakTime employs MVC architecture and uses bcrypt-based password hashing with custom salting.

Architecture

YakTime follows the Model-View-Controller pattern.

Application Tier: Node.js runtime with Express.js framework handles HTTP requests through route controllers (auth.js, dashboard.js, weight.js, supplements.js, nutrition.js, search.js). EJS templating engine renders server-side views. Two middleware components (requireAuth, redirectIfAuthenticated) manage authentication. Utility modules (password.js, usda.js) provide cryptographic operations and API integration.

Data Tier: MySQL relational database stores data across five tables (users, weight_records, goals, supplements, favorite_foods) with foreign key constraints ensuring referential integrity.



Data model

The database schema supports multi user operation with relational integrity. The users table stores usernames, email addresses, password hashes, and cryptographic salts, with uniqueness constraints on username and email fields.

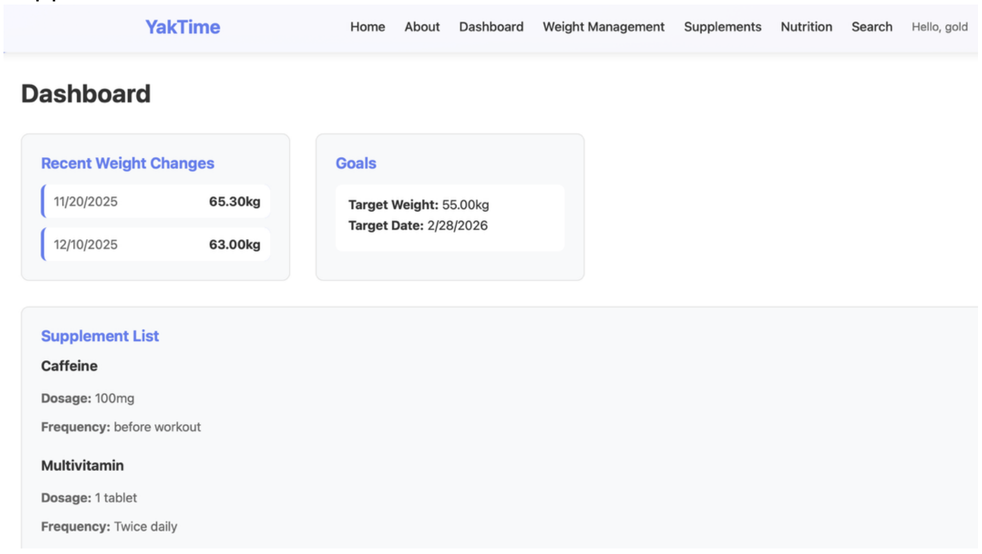
The weight_records table maintains time series weight measurements with a composite unique constraint on user_id and record_date, enabling upsert operations. The goals table stores weight targets with optional dates. The supplements table includes optional fields for dosage, frequency and notes.

User Functionality

Registration implements password validation requiring minimum eight characters with at least one lowercase letter, uppercase letter, number, and special character. The system checks for duplicate usernames and emails before account creation.

Authentication uses bcrypt with custom salting. The system generates a 16 byte salt using crypto.randomBytes, concatenates it with the password, and applies bcrypt hashing. Sessions persist user identity across requests.

The dashboard displays the seven most recent weight records, the current goal, and all supplements with action buttons for common tasks.



Dashboard

Recent Weight Changes

11/20/2025	65.30kg
12/10/2025	63.00kg

Goals

Target Weight: 55.00kg
Target Date: 2/28/2026

Supplement List

Caffeine

Dosage: 100mg

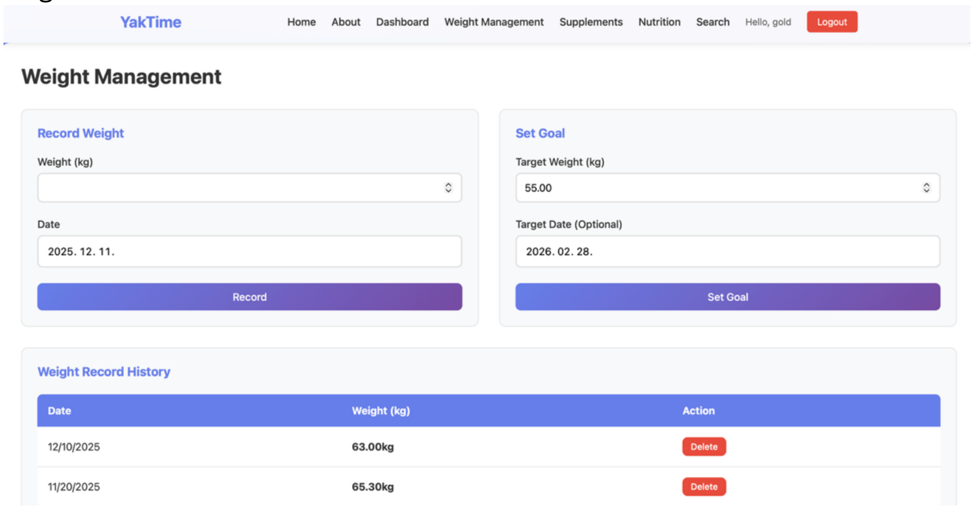
Frequency: before workout

Multivitamin

Dosage: 1 tablet

Frequency: Twice daily

Weight management allows users to record weight with dates. The system updates existing records when users log weight for the same day twice and enables goal setting with optional target dates.



Weight Management

Record Weight

Weight (kg):

Date: 2025. 12. 11.

Set Goal

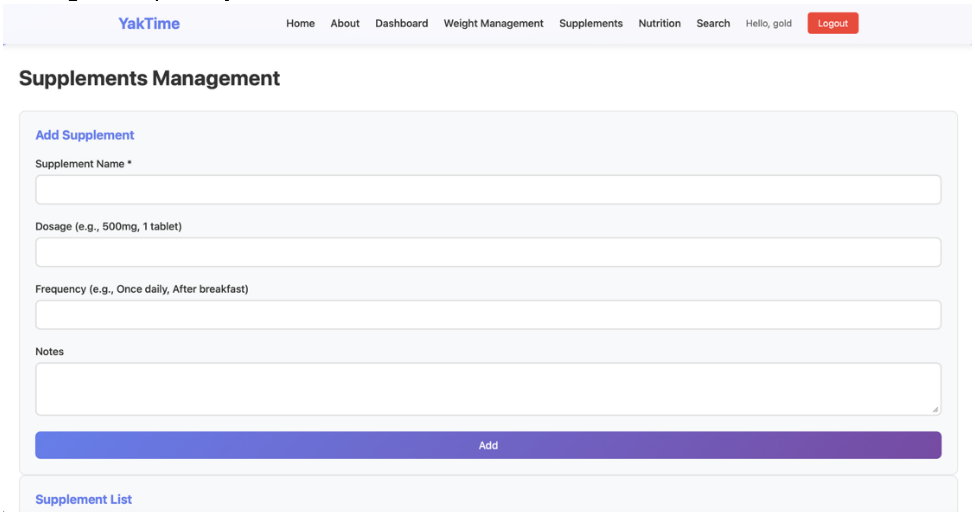
Target Weight (kg): 55.00

Target Date (Optional): 2026. 02. 28.

Weight Record History

Date	Weight (kg)	Action
12/10/2025	63.00kg	Delete
11/20/2025	65.30kg	Delete

Supplement management provides CRUD operations with required name field and optional dosage, frequency, and notes fields.



YakTime
Home About Dashboard Weight Management Supplements Nutrition Search Hello, gold Logout

Supplements Management

Add Supplement

Supplement Name *

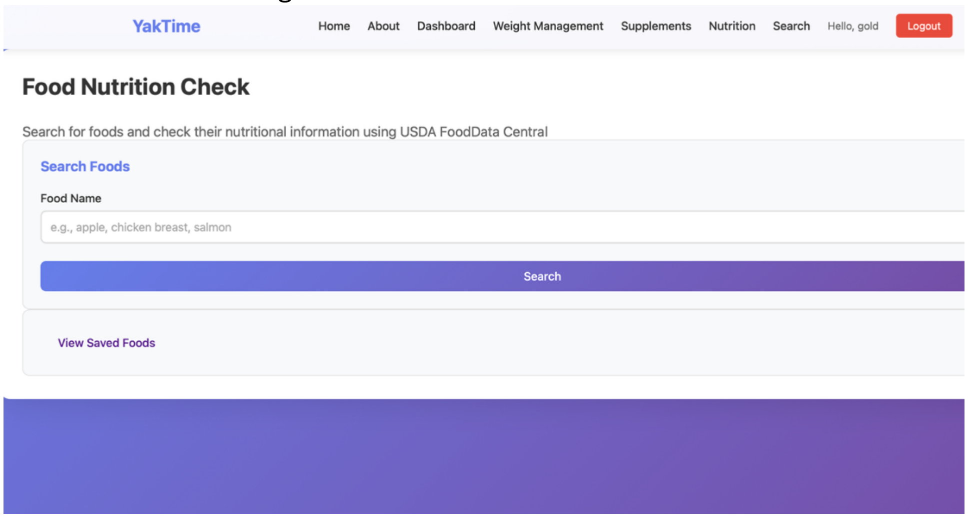
Dosage (e.g., 500mg, 1 tablet)

Frequency (e.g., Once daily, After breakfast)

Notes

Supplement List

Nutrition search uses the USDA FoodData Central API for nutritional data. Users can save foods to favorites and manage their saved list.



YakTime
Home About Dashboard Weight Management Supplements Nutrition Search Hello, gold Logout

Food Nutrition Check

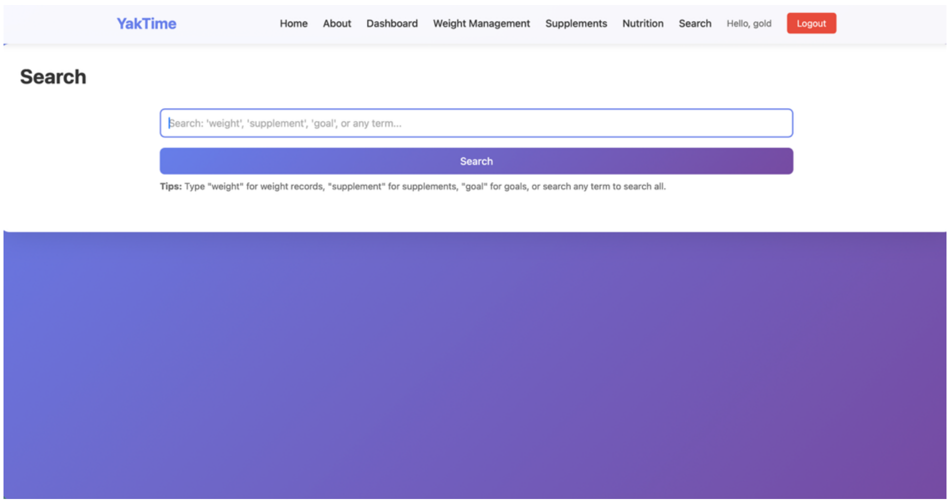
Search for foods and check their nutritional information using USDA FoodData Central

Search Foods

Food Name

e.g., apple, chicken breast, salmon

Search functionality provides queries across all user data with category filtering using keywords like "weight", "supplement", or "goal".



YakTime
Home About Dashboard Weight Management Supplements Nutrition Search Hello, gold Logout

Search

Search: 'weight', 'supplement', 'goal', or any term...

Tips: Type "weight" for weight records, "supplement" for supplements, "goal" for goals, or search any term to search all.

Advanced Techniques

The security architecture combines bcrypt's adaptive hashing with custom salting. YakTime generates a salt using crypto.randomBytes, concatenates it with the password before hashing, and stores it separately. This approach increases resistance against rainbow table attacks.
In password.js:

```
function generateSalt() {.
   return crypto.randomBytes(16).toString('hex');
}

async function hashPassword(password, salt) {
   return await bcrypt.hash(password + salt, 10);
}
```

All database operations use parameterized queries in auth.js:

```
await pool.execute(
   'INSERT INTO users (username, email, password_hash, salt) VALUES.
 (?, ?, ?, ?)',
   [username, email, passwordHash, salt]
);
```

Database operations employ ON DUPLICATE KEY UPDATE in weight.js:

```
await pool.execute(
   'INSERT INTO weight_records (user_id, weight, record_date) VALUES
(?, ?, ?) ON DUPLICATE KEY UPDATE weight = ?',
   [userId, weight, record_date, weight]
);
```

The USDA API integration in usda.js wraps calls in try-catch blocks and transforms responses through formatNutritionInfo and getCommonNutrients functions.

The USDA API integration wraps API calls in try-catch blocks and transforms JSON responses into simplified structures. The formatNutritionInfo function extracts nutrients, while getCommonNutrients filters to commonly tracked values.

Error handling uses try-catch blocks with messages propagated through query parameters for feedback.

AI Declaration

During the development of this website, I utilized AI tools for specific tasks. ChatGPT was used for translating content from my native language to English when needed. Claude was employed to standardize code comments, using prompts such as: "Translate these comments to English and unify the style."

For debugging purposes, I used Cursor on December 8th with prompts including: "I keep getting 404 errors when uploading to the VM, and I can't figure out the cause." These AI tools assisted with language barriers and troubleshooting, but all architectural decisions, feature implementations, and core programming logic were developed independently.