Health and Fitness App Report

OUTLINE

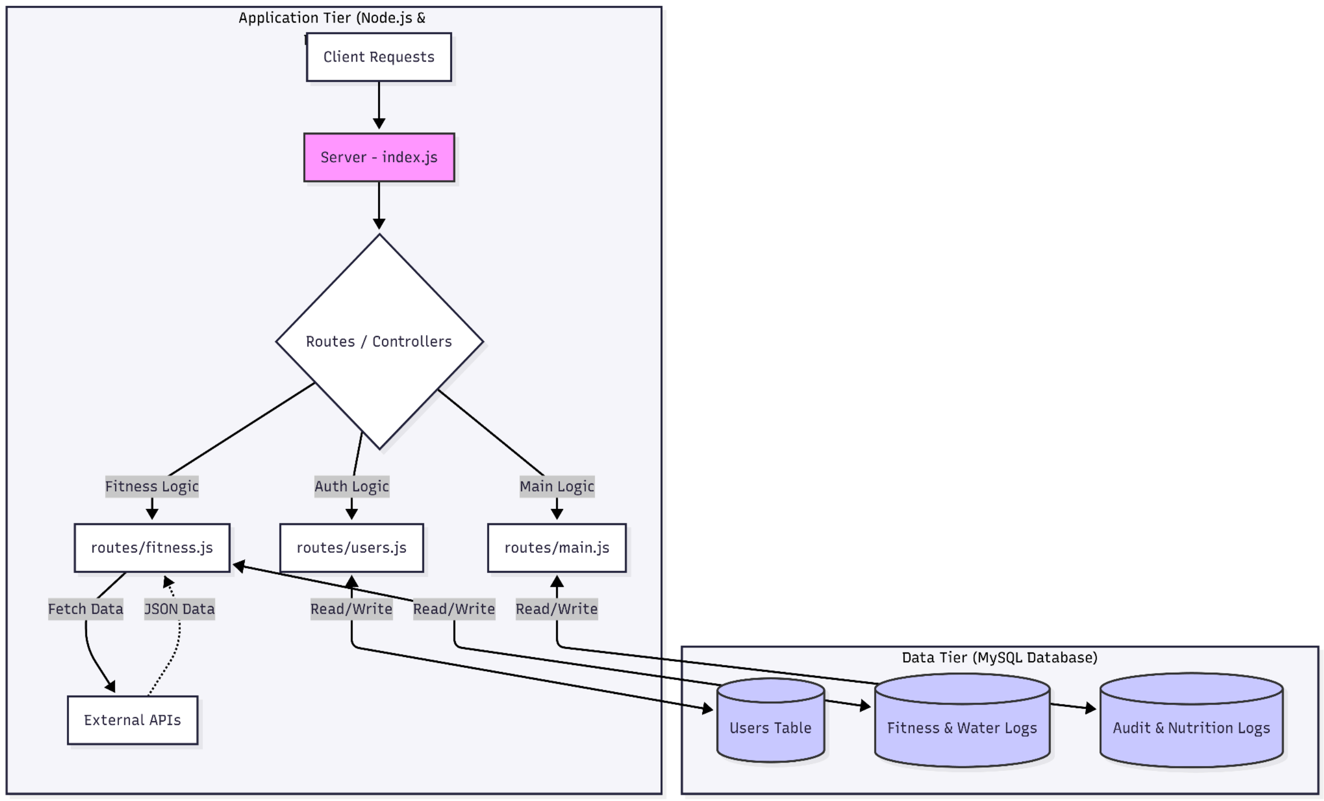
Bitality is a comprehensive web-based health and fitness application designed to help individuals track their physical activity and monitor their health metrics. The application allows users to securely log and view their daily workouts, including details such as activity type, duration, calories burned, and intensity. Beyond simple logging, Bitality provides valuable health tools including BMI (Body Mass Index), BMR (Basal Metabolic Rate), and Macronutrient calculators to better inform user health decisions. Integrated charting and dashboard widgets offer visual feedback on progress. The platform prioritizes user privacy with secure authentication and session management, while a responsive, high-contrast "Dark Mode" design ensures accessibility and a premium user experience across devices. The goal is to provide an intuitive, all-in-one platform for personal health management.

ARCHITECTURE

The application utilizes a classic Three-Tier Architecture implemented via the Model-View-Controller (MVC) pattern:

1. **Presentation Tier (View):** EJS templates render dynamic HTML pages on the server, served to the client browser.
2. **Application Tier (Controller):** Node.js and Express handle routing, business logic, authentication, and request processing.
3. **Data Tier (Model):** A MySQL database securely stores relational data for users, fitness logs, and audit trails.

High-Level Architecture Diagram:

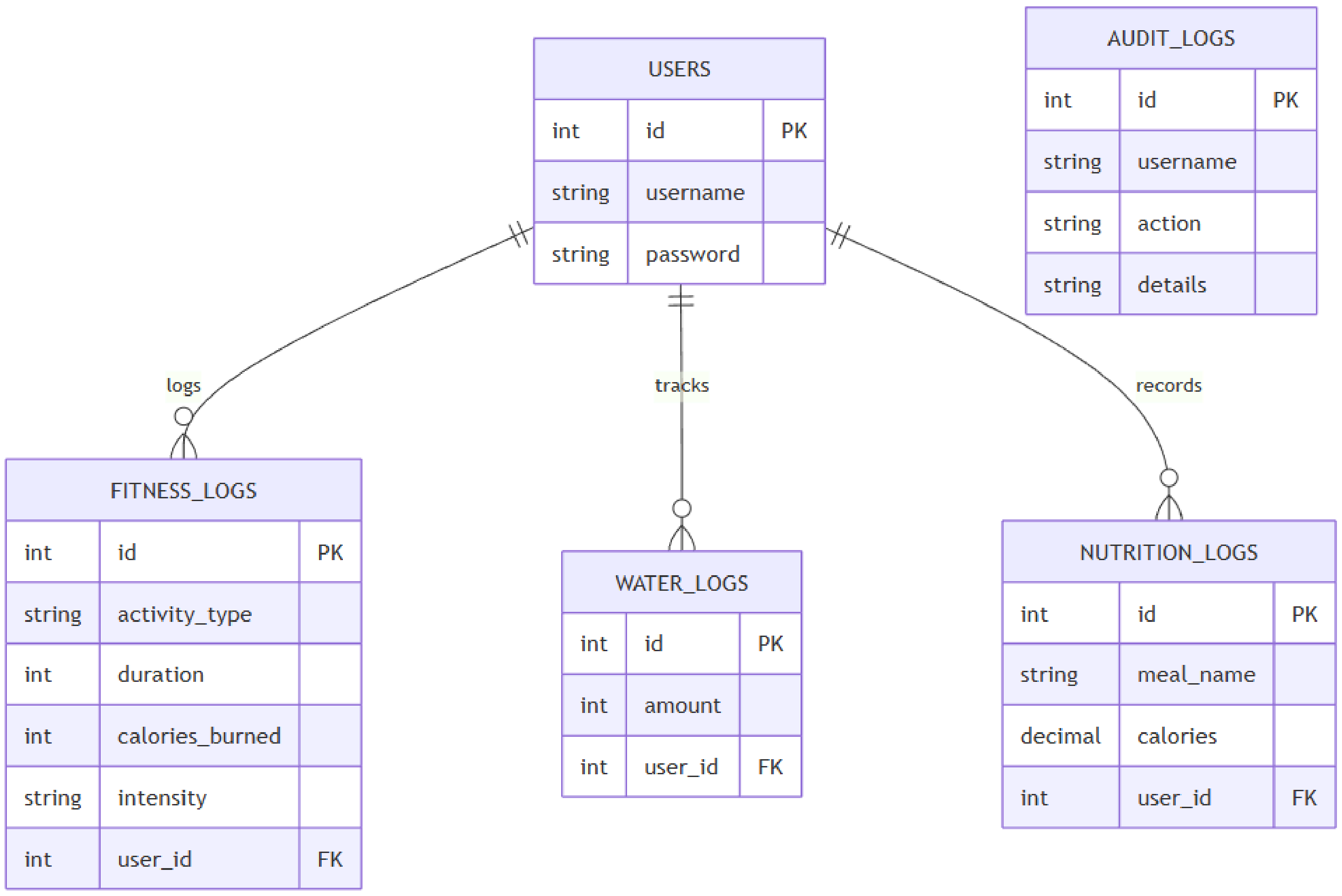


DATA MODEL

The database schema (health) is normalized and consists of three primary tables:

1. **users:** Manages authentication (ID, unique username, password).
2. **fitness_logs:** Stores workout activities. It has a Foreign Key (user_id) relationship with users (One-to-Many).
3. **audit_logs:** Tracks system events for security and debugging.
4. **water_logs:** Stores daily water intake records for persistence.
5. **nutrition_logs:** Stores meal data (name, calories, protein, carb, fat) for nutrition tracking.

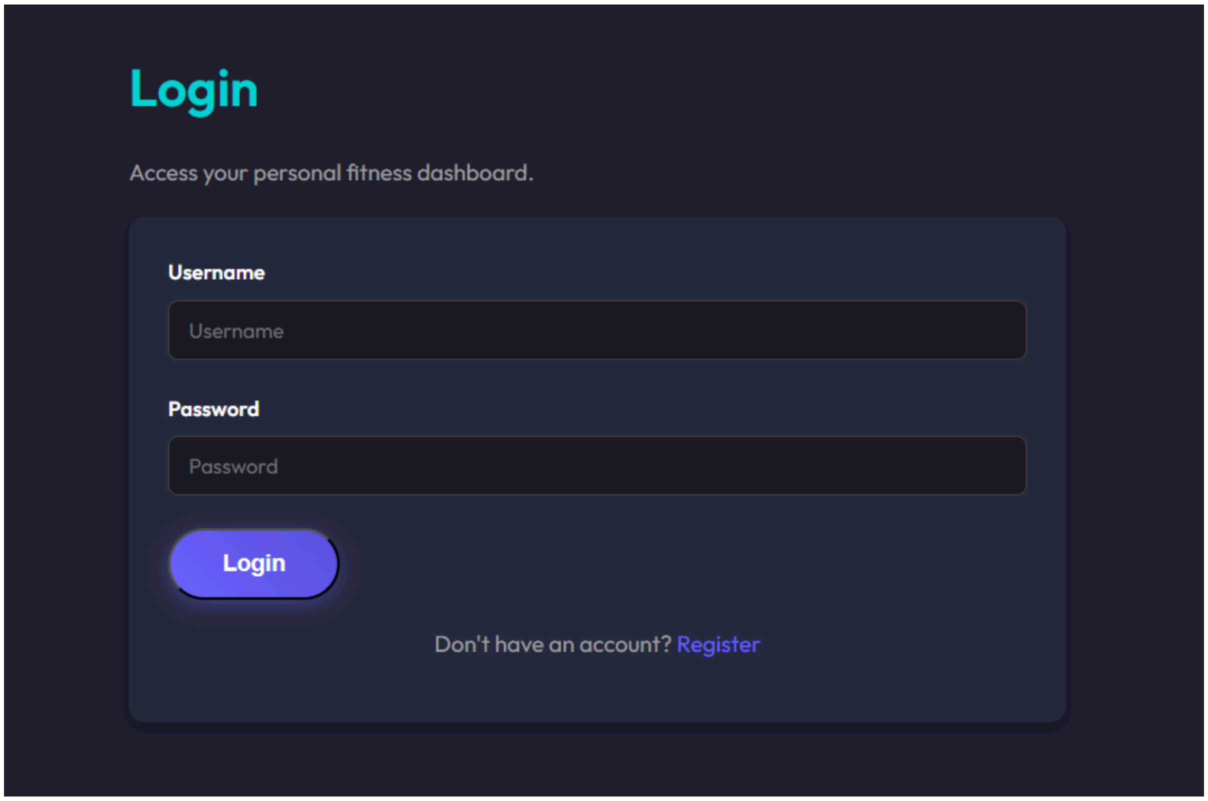
Data Model Diagram:



USER FUNCTIONALITY

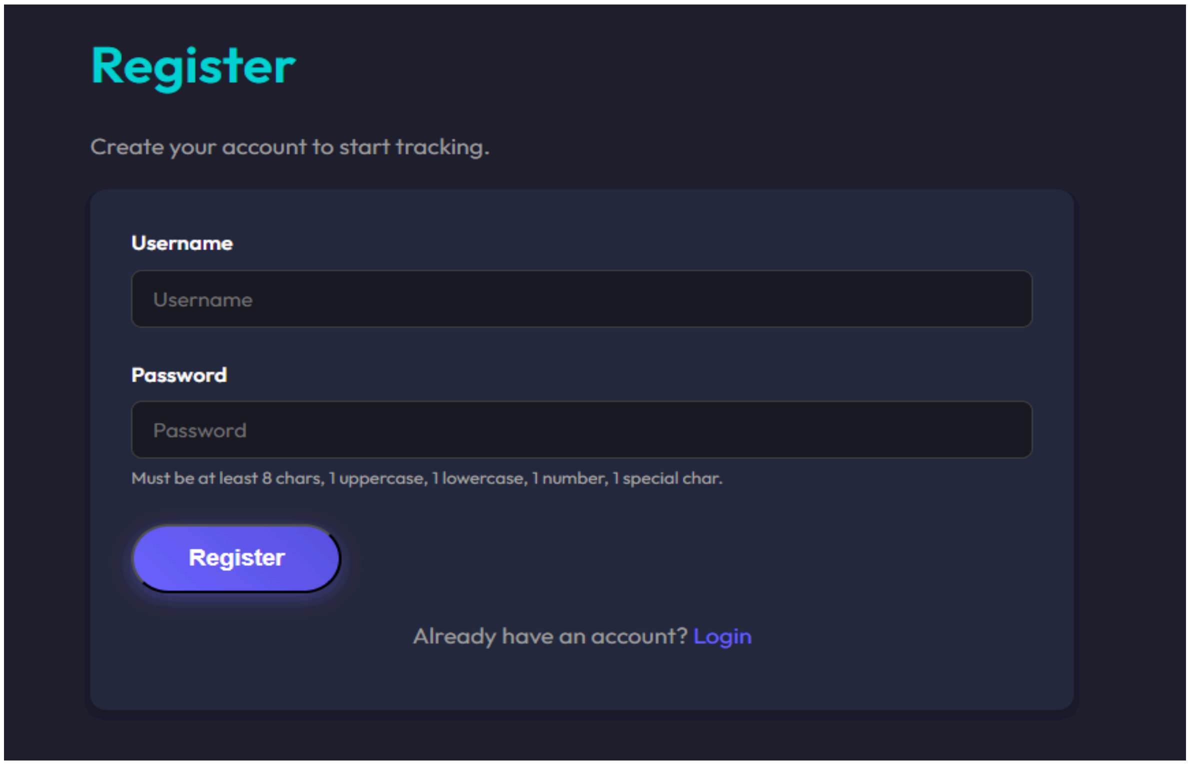
Authentication & Profiles

Users must register and login to access personal data. The system enforces strong passwords for security.



Login

Access your personal fitness dashboard.



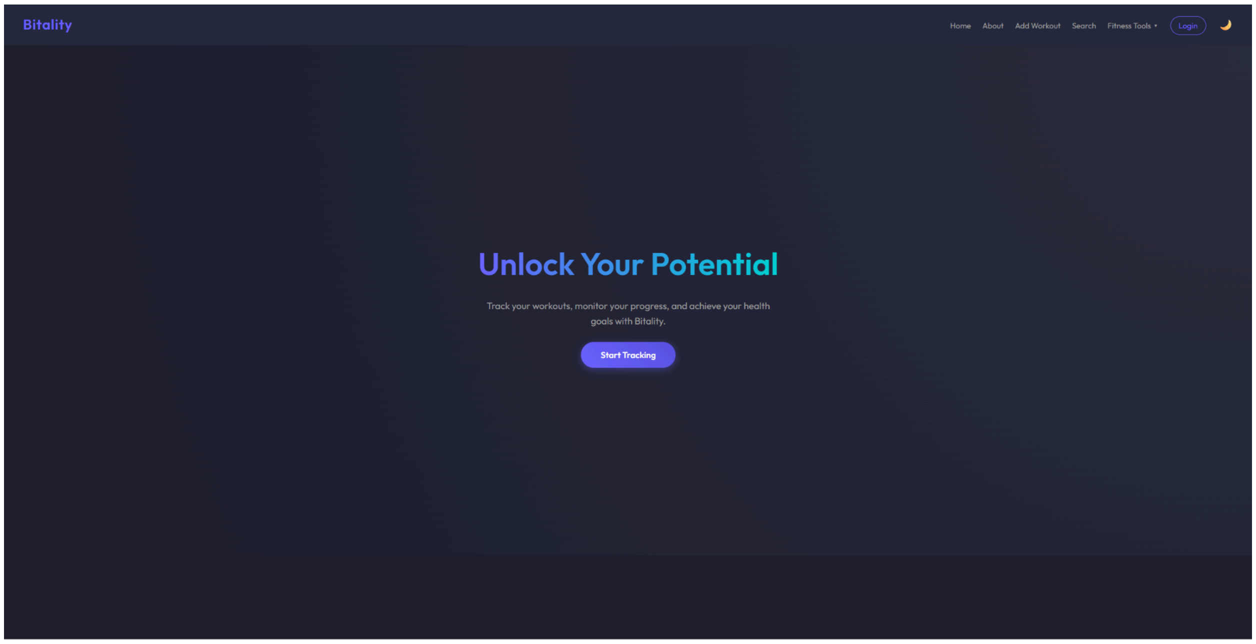
Register

Create your account to start tracking.

Must be at least 8 chars, 1 uppercase, 1 lowercase, 1 number, 1 special char.

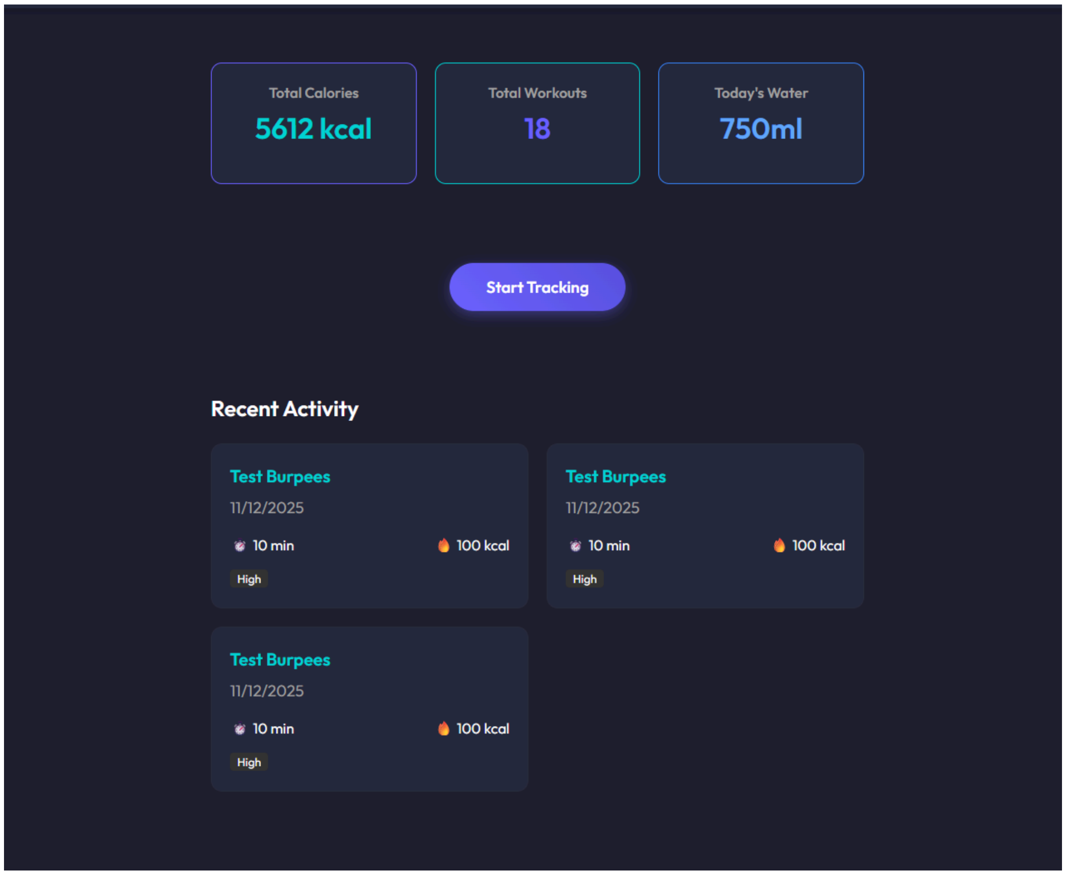
Dashboard & User Experience

Upon login, the Home Dashboard greets users with "Quick Stats" widgets displaying their total calories burned, total workouts, and daily water intake at a glance, followed by their recent activity.



Unlock Your Potential

Track your workouts, monitor your progress, and achieve your health goals with Bitality.



Total Calories	Total Workouts	Today's Water
5612 kcal	18	750ml

Recent Activity

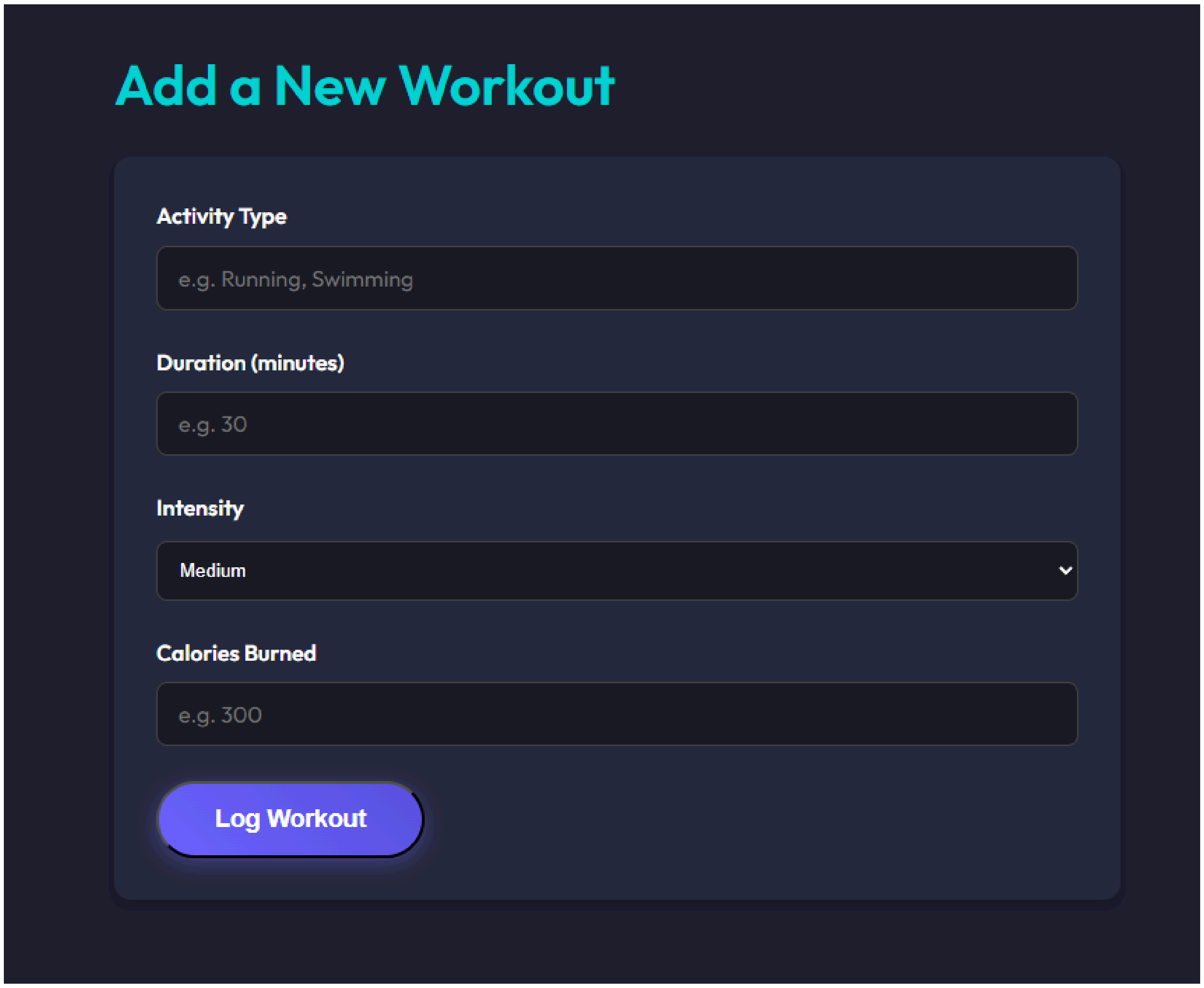
Test Burpees
11/12/2025
10 min 100 kcal
High

Test Burpees
11/12/2025
10 min 100 kcal
High

Test Burpees
11/12/2025
10 min 100 kcal
High

Workout Logging

The Add Workout feature allows users to log exercises with specific metrics like duration, calories, and intensity. This data is validated server-side to ensure accuracy (e.g., positive integers).



Add a New Workout

Activity Type

e.g. Running, Swimming

Duration (minutes)

e.g. 30

Intensity

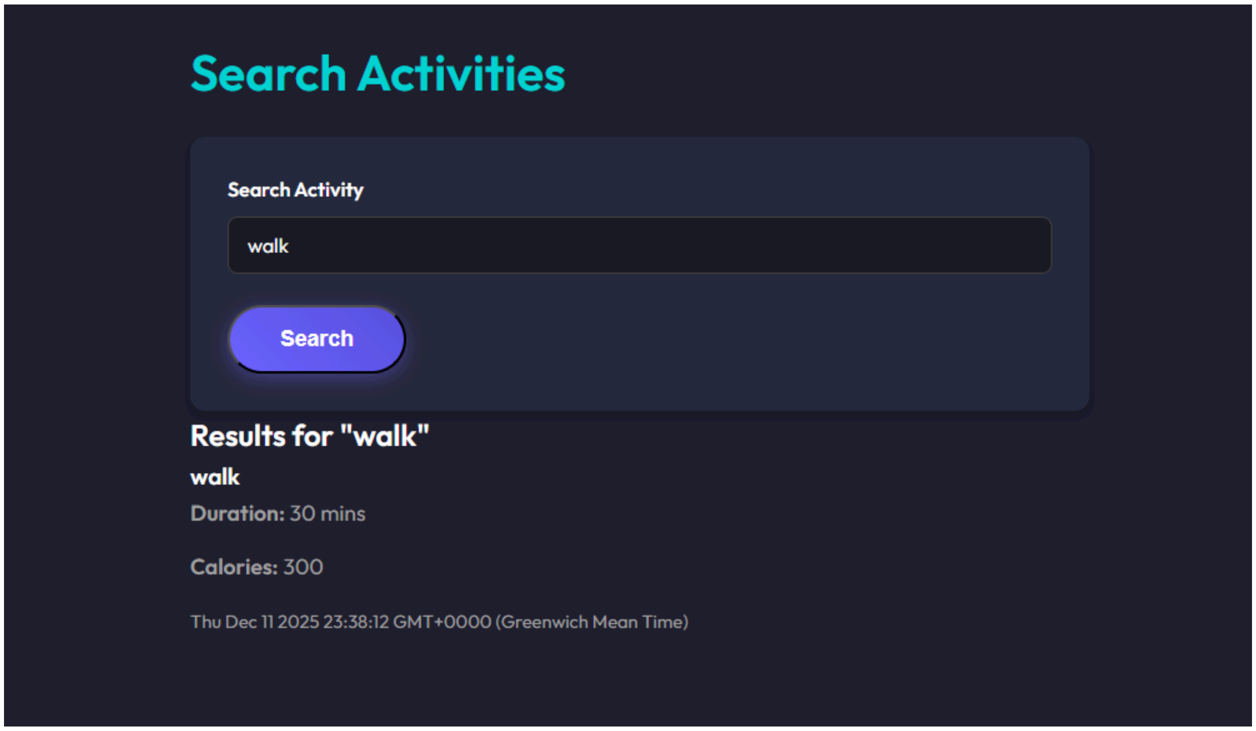
Medium

Calories Burned

e.g. 300

Search & History

A robust Search function filters workout history by activity type, allowing users to quickly find past runs, swims, or cycles.



Search Activities

Search Activity

walk

Results for "walk"

walk

Duration: 30 mins

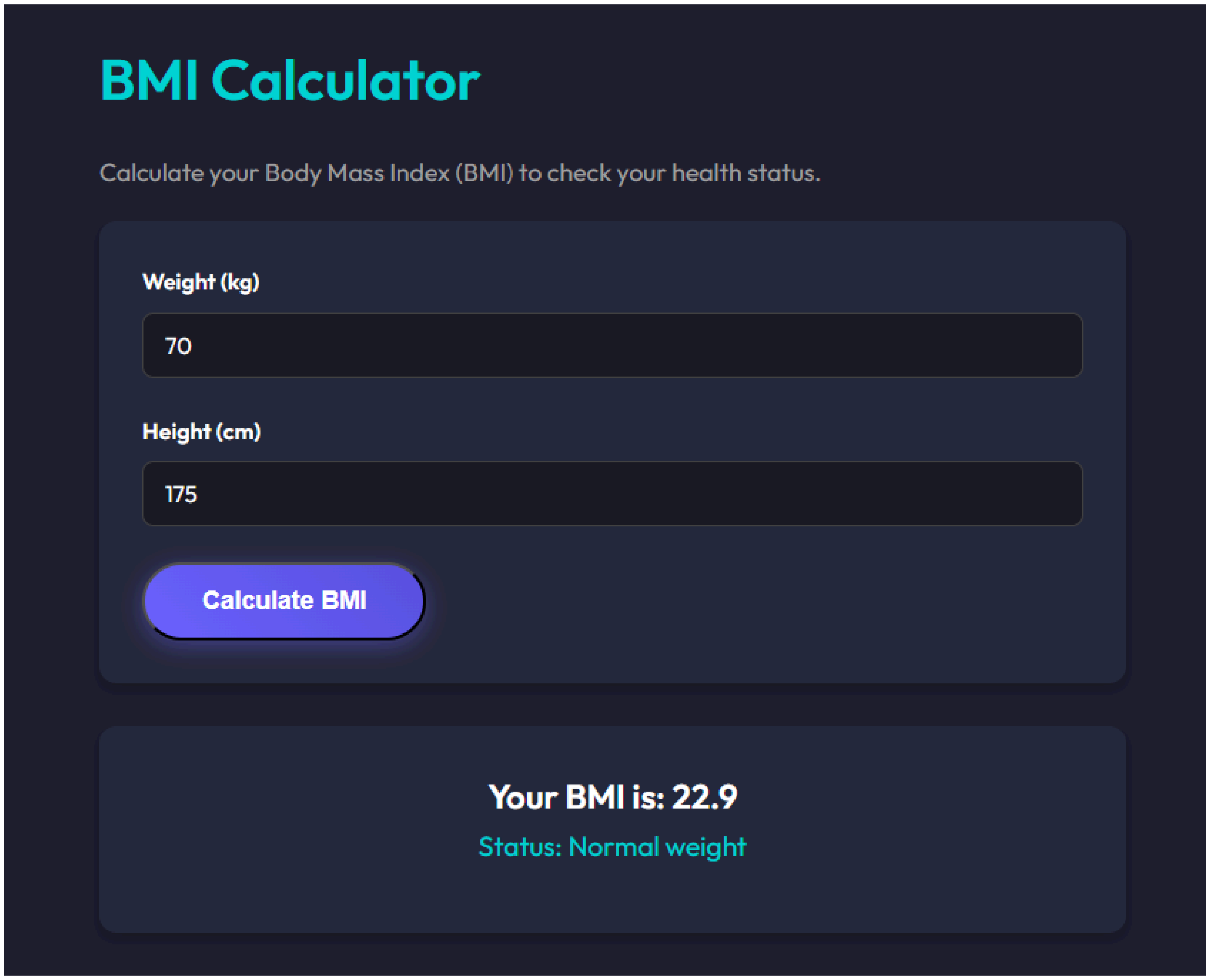
Calories: 300

Thu Dec 11 2025 23:38:12 GMT+0000 (Greenwich Mean Time)

Health Tools

Bitality includes a suite of calculators and tools:

- **BMI Calculator:** Assesses weight category.



BMI Calculator

Calculate your Body Mass Index (BMI) to check your health status.

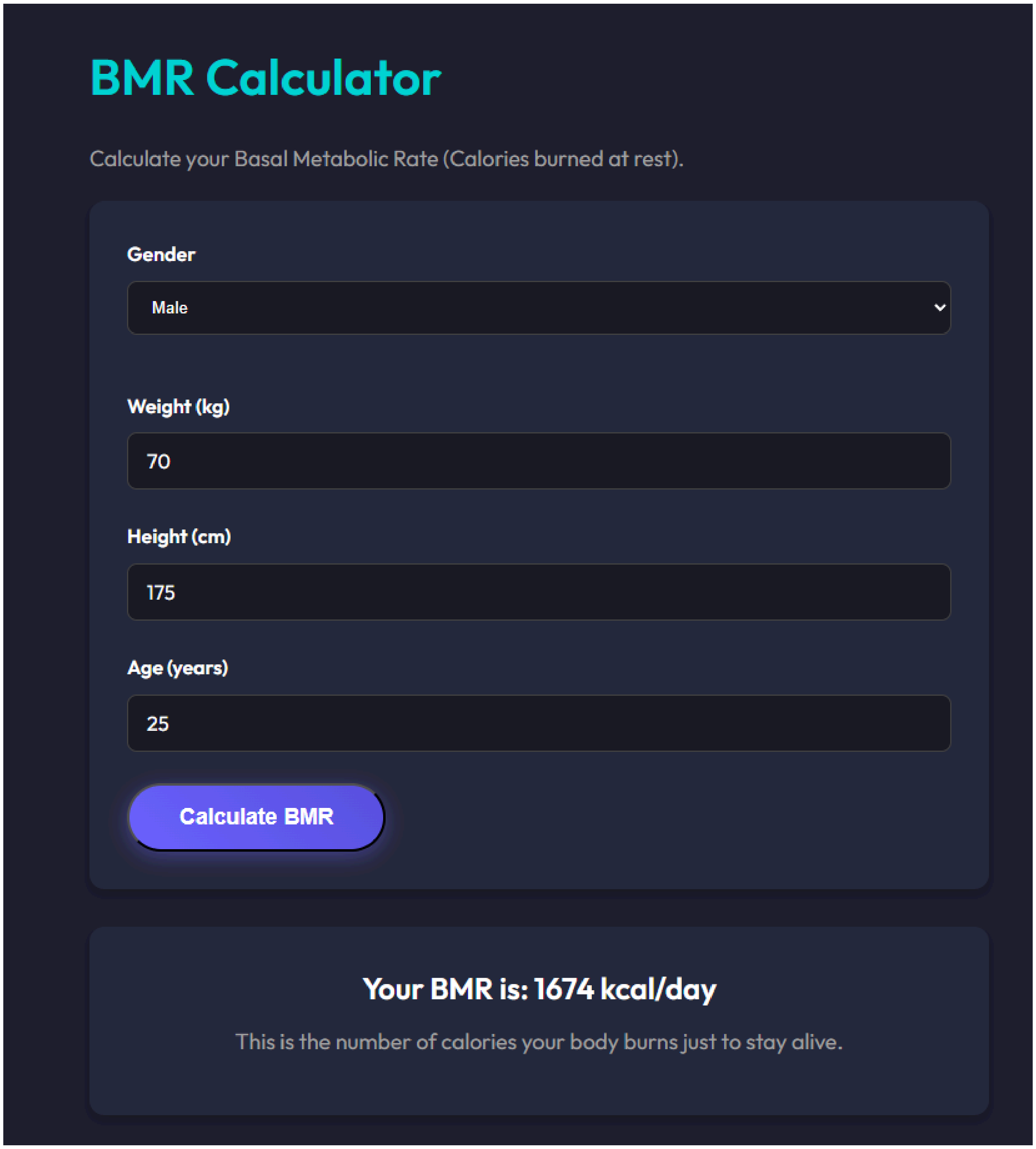
Weight (kg): 70

Height (cm): 175

Your BMI is: 22.9

Status: Normal weight

- **BMR Calculator:** Estimates daily caloric needs.



BMR Calculator

Calculate your Basal Metabolic Rate (Calories burned at rest).

Gender

Male

Weight (kg)

70

Height (cm)

175

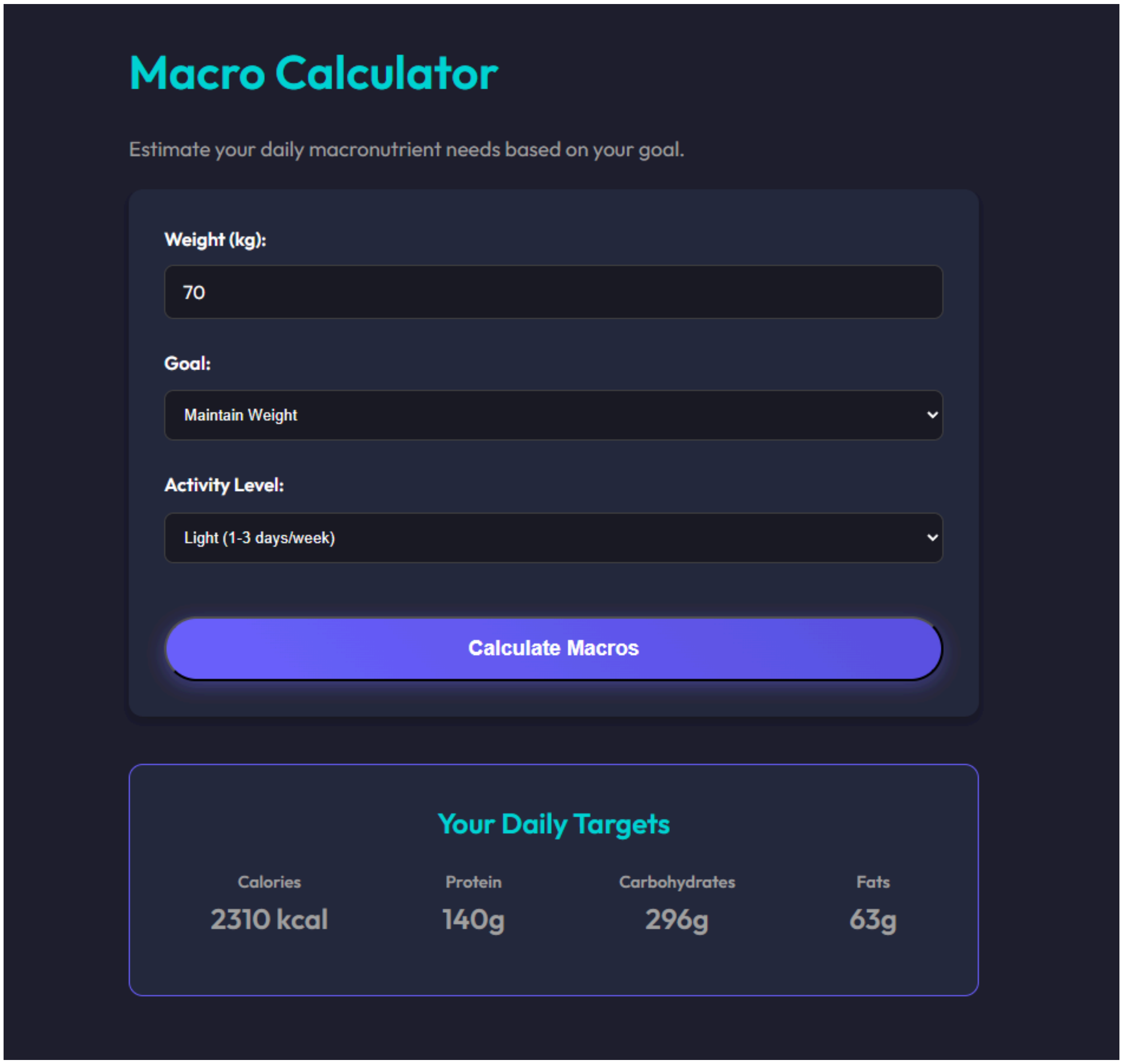
Age (years)

25

Your BMR is: 1674 kcal/day

This is the number of calories your body burns just to stay alive.

- **Macro Calculator:** Suggests nutrient splits based on fitness goals (lose/gain weight)



Macro Calculator

Estimate your daily macronutrient needs based on your goal.

Weight (kg): 70

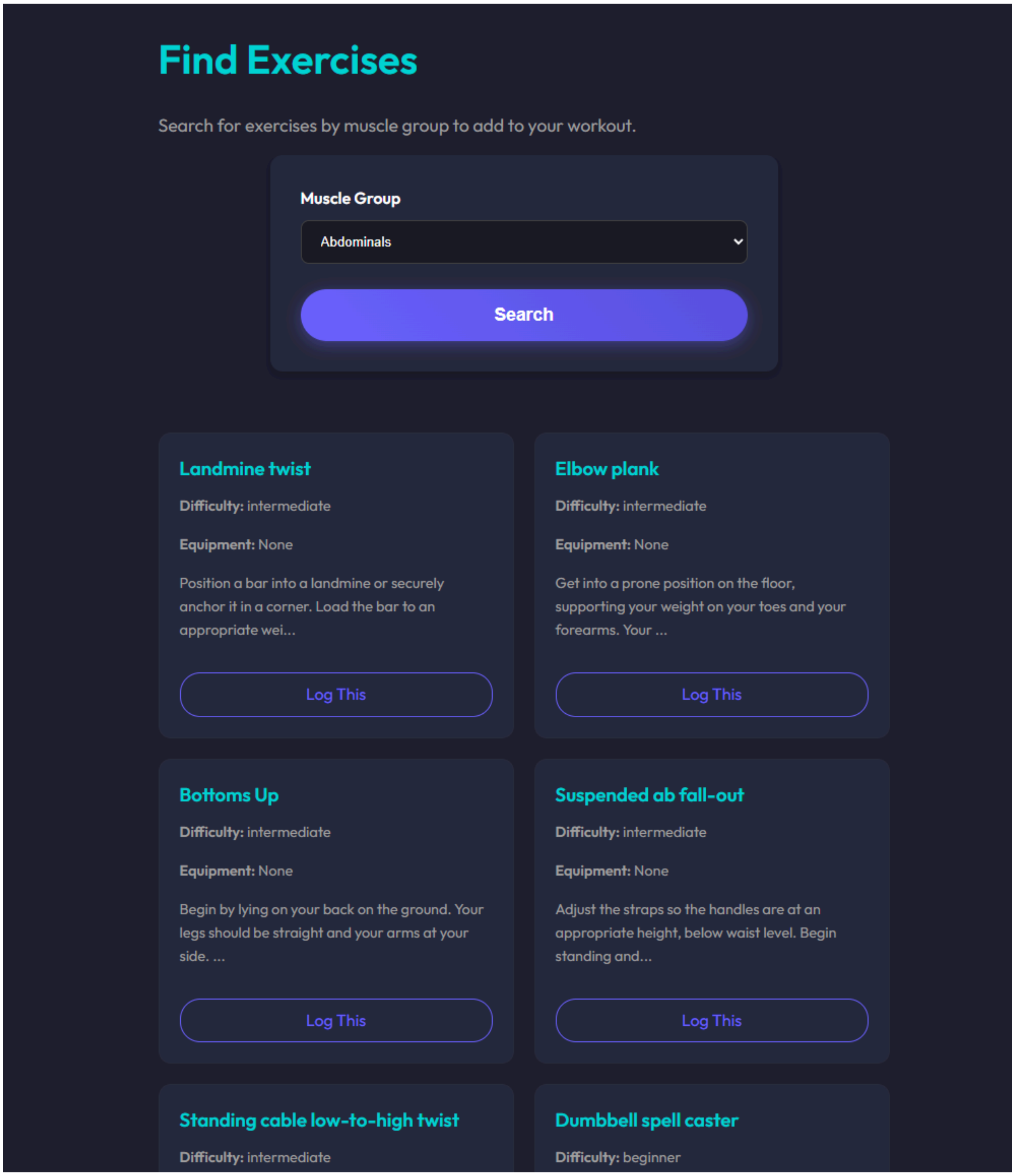
Goal: Maintain Weight

Activity Level: Light (1-3 days/week)

Your Daily Targets

Calories	Protein	Carbohydrates	Fats
2310 kcal	140g	296g	63g

- **Exercise Finder:** Uses the API Ninjas Exercises API to let users discover new workouts by muscle group and seamlessly log them.



Find Exercises

Search for exercises by muscle group to add to your workout.

Muscle Group

Abdominals

Search

Landmine twist

Difficulty: intermediate

Equipment: None

Position a bar into a landmine or securely anchor it in a corner. Load the bar to an appropriate wei...

Log This

Elbow plank

Difficulty: intermediate

Equipment: None

Get into a prone position on the floor, supporting your weight on your toes and your forearms. Your ...

Log This

Bottoms Up

Difficulty: intermediate

Equipment: None

Begin by lying on your back on the ground. Your legs should be straight and your arms at your side. ...

Log This

Suspended ab fall-out

Difficulty: intermediate

Equipment: None

Adjust the straps so the handles are at an appropriate height, below waist level. Begin standing and...

Log This

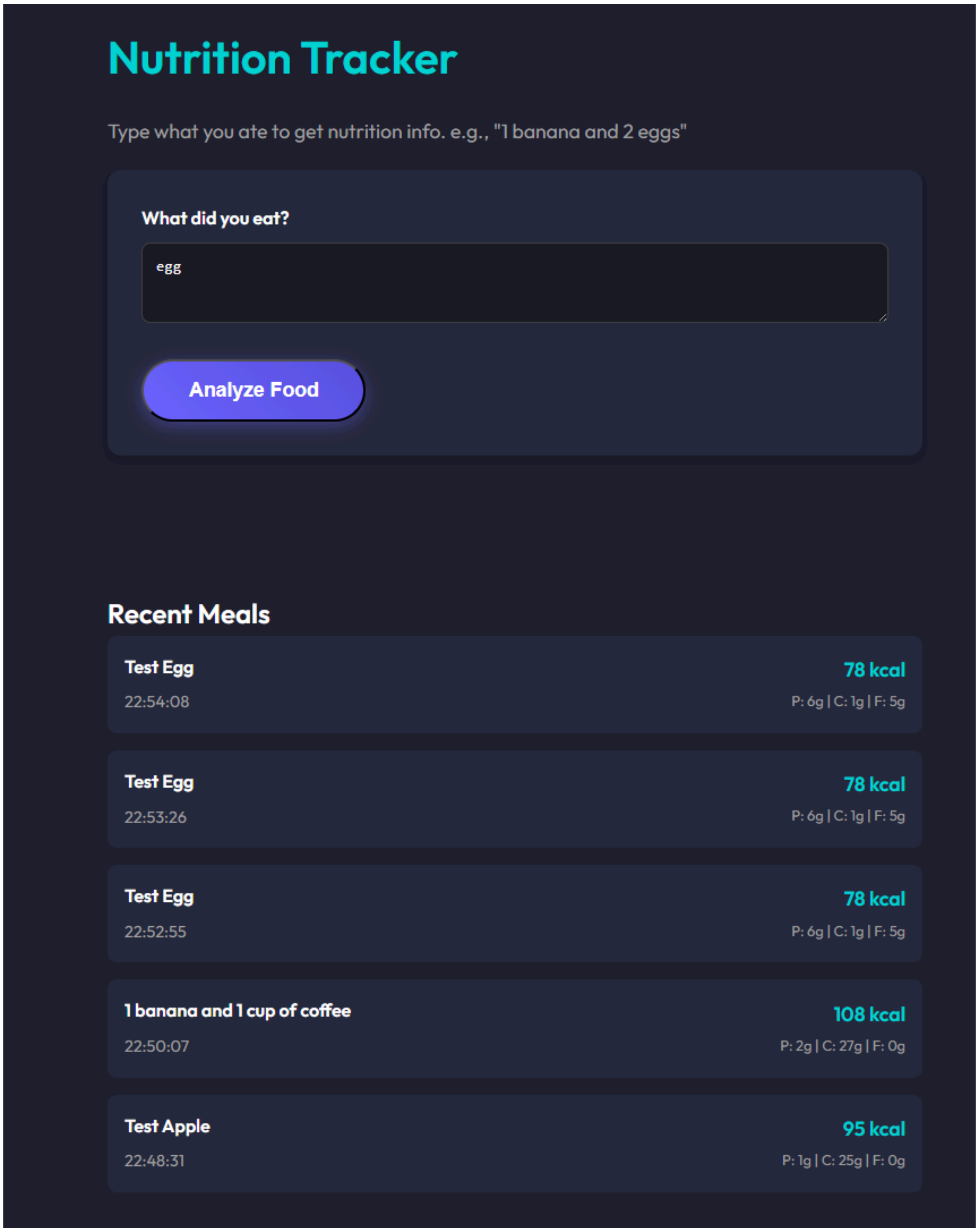
Standing cable low-to-high twist

Difficulty: intermediate

Dumbbell spell caster

Difficulty: beginner

- **Nutrition Tracker:** Users can type natural language meal descriptions (e.g., "1 avocado and 2 eggs"). The app uses the CalorieNinjas API to analyze the text and save a detailed nutrition breakdown (Calories, Macros).



Nutrition Tracker

Type what you ate to get nutrition info. e.g., "1 banana and 2 eggs"

What did you eat?

egg

Recent Meals

Test Egg
22:54:08
78 kcal
P: 6g | C: 1g | F: 5g

Test Egg
22:53:26
78 kcal
P: 6g | C: 1g | F: 5g

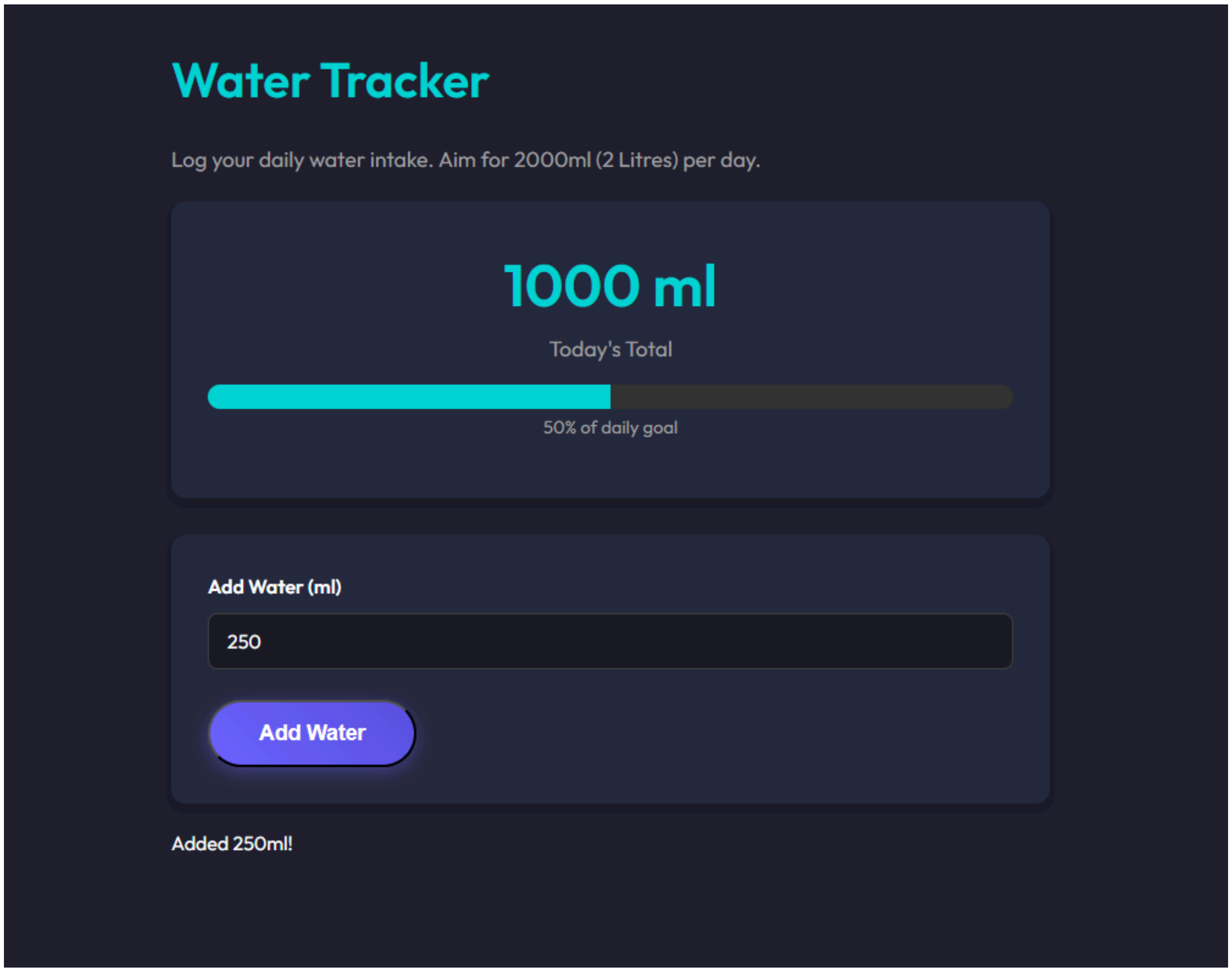
Test Egg
22:52:55
78 kcal
P: 6g | C: 1g | F: 5g

1 banana and 1 cup of coffee
22:50:07
108 kcal
P: 2g | C: 27g | F: 0g

Test Apple
22:48:31
95 kcal
P: 1g | C: 25g | F: 0g

Utilities

- **Water Tracker:** A persistent daily counter for hydration goals, stored in the database so progress isn't lost on logout.



Water Tracker

Log your daily water intake. Aim for 2000ml (2 Litres) per day.

1000 ml

Today's Total

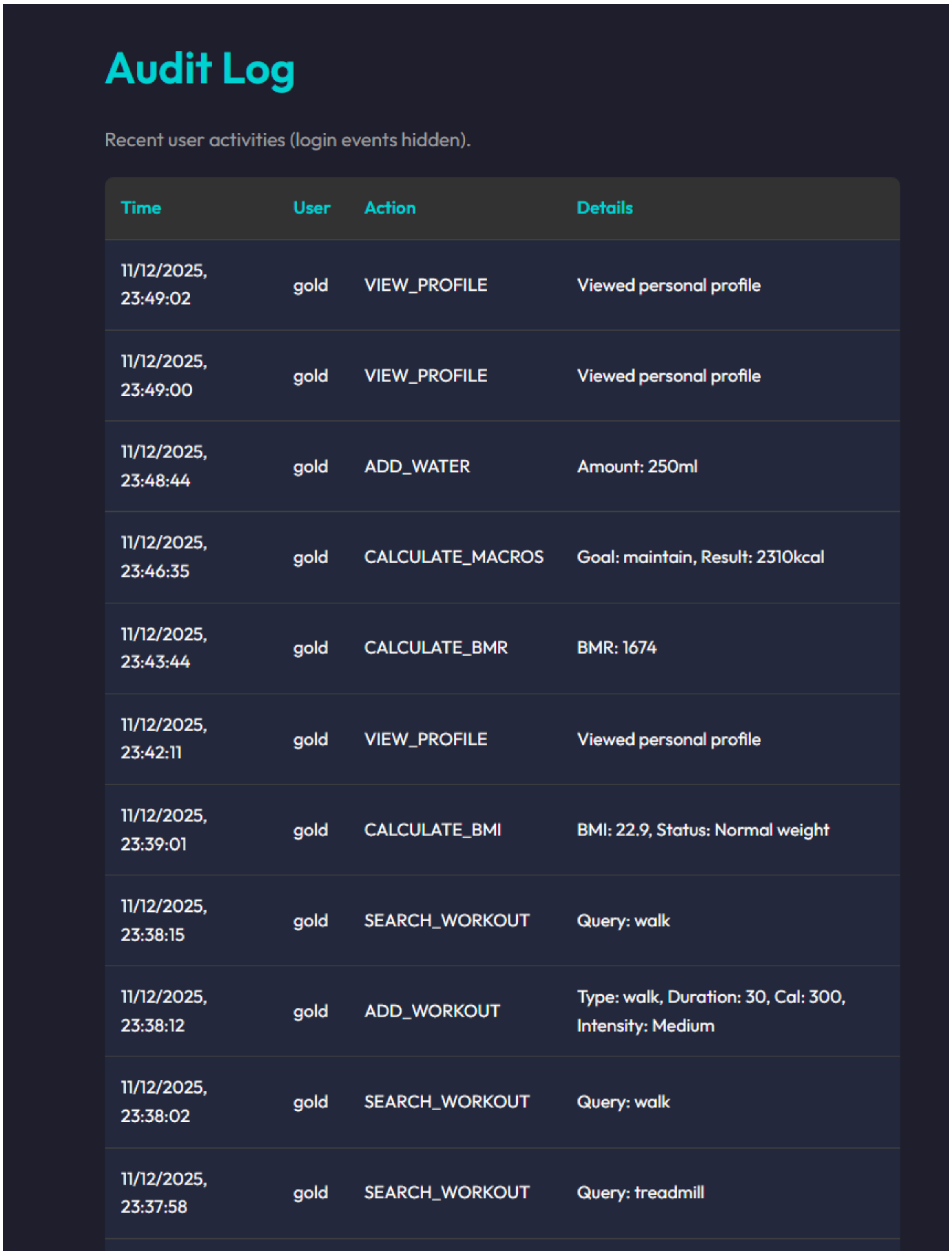
50% of daily goal

Add Water (ml)

250

Added 250ml!

- **Audit Log:** A transparency feature allowing users to view recorded system actions associated with their account.



Audit Log

Recent user activities (login events hidden).

Time	User	Action	Details
11/12/2025, 23:49:02	gold	VIEW_PROFILE	Viewed personal profile
11/12/2025, 23:49:00	gold	VIEW_PROFILE	Viewed personal profile
11/12/2025, 23:48:44	gold	ADD_WATER	Amount: 250ml
11/12/2025, 23:46:35	gold	CALCULATE_MACROS	Goal: maintain, Result: 2310kcal
11/12/2025, 23:43:44	gold	CALCULATE_BMR	BMR: 1674
11/12/2025, 23:42:11	gold	VIEW_PROFILE	Viewed personal profile
11/12/2025, 23:39:01	gold	CALCULATE_BMI	BMI: 22.9, Status: Normal weight
11/12/2025, 23:38:15	gold	SEARCH_WORKOUT	Query: walk
11/12/2025, 23:38:12	gold	ADD_WORKOUT	Type: walk, Duration: 30, Cal: 300, Intensity: Medium
11/12/2025, 23:38:02	gold	SEARCH_WORKOUT	Query: walk
11/12/2025, 23:37:58	gold	SEARCH_WORKOUT	Query: treadmill

ADVANCED TECHNIQUES

1. Model-View-Controller (MVC) Pattern

The codebase is structured to strictly separate concerns. Routes (routes/fitness.js) handle logic, Views (views/*.ejs) handle display, and the Database handles storage. This improves maintainability compared to monolithic scripts.

2. Custom Middleware for Security

I implemented custom middleware requireLogin to protect sensitive routes. This ensures that unauthenticated users cannot access or modify fitness data using forced redirection.

Reference: routes/fitness.js

```
const requireLogin = (req, res, next) => {
    if (req.session.loggedin) {
        next();
    } else {
        res.redirect("/users/login");
    }
};
router.use(requireLogin);
```

3. Server-Side Validation & Sanitization

Using express-validator, input is rigorously checked and sanitized before touching the database. This prevents SQL injection and bad data entry (e.g., negative calories).

Reference: routes/fitness.js

```
check("duration").isInt({ min: 1 }).withMessage("Duration must be a
positive number"),
// ...
const errors = validationResult(req);
```

4. Design System with CSS Variables

A custom "Dark Mode" theme was built using CSS Custom Properties (--bg-color, --accent). This allows for dynamic client-side theming without page reloads, toggled via a simple JS script that updates the body class.

5. Complex SQL Aggregation

The dashboard utilizes complex SQL with subqueries to efficiently fetch summary statistics (Calories, Workouts, Water) in a single database round-trip, optimizing performance and reducing server load.

Reference: routes/main.js

```
const statsQuery = `
    SELECT
        COALESCE((SELECT SUM(calories_burned) FROM fitness_logs...), 0) as
totalCalories,
        COALESCE((SELECT COUNT(*) FROM fitness_logs...), 0) as
totalWorkouts,
        ...
`;
```

6. External API Integration

The application integrates with the API Ninjas Exercises API using Node.js fetch. This feature demonstrates asynchronous data handling (async/await) and third-party service consumption to enrich the user experience with real-world fitness data.

Reference: routes/fitness.js

```
const response = await
fetch(`https://api.api-ninjas.com/v1/exercises?muscle=${muscle}`, {
    headers: { "X-Api-Key": process.env.API_NINJAS_KEY },
});
```

AI DECLARATION

I used Google's Gemini AI to help debug problems and aid during development.

- **Debugging:** The AI helped identify and fix a missing database route for the Audit Log feature and corrected init_db.js configuration credential mismatches.
- **Image Generation:** The AI helped me generate the high level images for the report.
- **Database Queries:** The AI helped me with SQL commands to fix databases which were repeating or duplicated
- **Commit Messages:** The AI was used to properly name some commit messages for neater presentation.