IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

FitZen – Personal Fitness and Wellness Tracker

Ankita Mahendra Nikam[1], Prof. M. V. Nikum[2]

Research Scholar, MCA, Shri Jaykumar Rawal Institute of Technology, Dondaicha, India[1]

HOD MCA & Assistant Professor, Shri Jaykumar Rawal Institute of Technology, Dondaicha, India[2]

Abstract: Maintaining personal fitness requires consistent attention to body measurements, exercise, food intake, and activity history. When these activities are managed through separate applications or manual records, users may find it difficult to maintain an organized view of their progress. FitZen – Personal Fitness and Wellness Tracker is a web-based application designed to bring these everyday fitness functions together within a single platform. The system provides user registration and profile management, a Body Mass Index (BMI) calculator, goal-oriented workout suggestions, general diet recommendations, daily calorie logging, and fitness progress records. The project uses Python Flask for backend routing and application logic, HTML5, CSS3, JavaScript and Bootstrap for the responsive user interface, and Supabase PostgreSQL for relational data storage. Database Row Level Security (RLS), password hashing, session checks, and environment-variable based credential management are incorporated to support user data privacy. The project report describes unit, integration, system, and security testing of the main workflows. FitZen is intended as a beginner-friendly wellness support tool; its BMI output and recommendations are informational and are not substitutes for professional medical advice.

Keywords: FitZen, Fitness Tracker, BMI Calculator, Calorie Tracking, Workout Planning, Flask, Supabase, Row Level Security.

I. INTRODUCTION

Digital technologies have changed the way individuals record and review information about their daily activities. However, personal fitness management often involves multiple independent tasks: understanding basic body measurements, selecting suitable exercises, planning everyday meals, recording calorie intake, and reviewing activity consistency. When these tasks are handled through separate tools or handwritten records, information becomes fragmented and historical review becomes inconvenient.

FitZen – Personal Fitness and Wellness Tracker is proposed as a unified web-based platform for these routine self-monitoring activities. It combines a BMI calculator, personalized workout suggestions, general diet recommendations, a calorie tracker, and a fitness progress module. A user can maintain relevant profile details, enter height and weight, select a fitness goal, record food intake, and save completed workout activities through an organized interface.

A. Problem Statement

Users may face difficulty maintaining consistent fitness routines because useful tools and records are distributed across different platforms. Choosing workouts that correspond to personal goals can be confusing; basic body measurements may not be easy to interpret; daily meal planning and calorie recording require effort; and activity history may not be available in one convenient place. The project addresses this fragmentation by providing an integrated platform for basic fitness planning and progress recording.

B. Objectives and Scope

- Provide a convenient web interface for everyday fitness tools and relevant user profile information.
- Calculate BMI from user-entered height and weight and display a standard adult screening category.
- Suggest beginner-friendly workouts according to selected goals and exercise preferences.
- Present general diet suggestions and allow users to record daily food items and estimated calories.
- Record completed fitness activities and help users review consistency and progress over time.

The scope covers profile inputs, BMI calculation, workout suggestions, general diet guidance, calorie records, and fitness activity history. The application is intended for general wellness support, not diagnosis. Recommendations depend on the information entered by users, and individual health needs may require professional guidance.

II. LITERATURE REVIEW

The project report identifies a shift from paper-based logbooks and isolated spreadsheets toward web-accessible health and fitness systems. Digital platforms can combine data entry, calculation, storage, and progress review within a consistent interface. The design of FitZen is informed by established software engineering practices for modular application development, relational database design, and web application security [2], [7].

A. Review of Existing Platforms

The project report reviews MyFitnessPal, Strava, Google Fit, Apple Health, and standalone BMI or calorie calculators as examples of existing approaches. MyFitnessPal provides extensive food logging and nutritional information, while Strava emphasizes GPS-based outdoor activity tracking. Google Fit and Apple Health act primarily as health-data aggregators. Standalone calculators provide quick calculations but generally do not offer persistent user

accounts or a combined history of fitness activities. These systems illustrate useful individual capabilities, while FitZen focuses on bringing a selected set of beginner-oriented functions into one lightweight web application.

B. Identified Functional Gap

The project identifies three broad gaps: fragmentation between BMI, workout, diet, and calorie tools; complexity that may discourage beginners; and the need for clear user-level privacy controls when personal fitness information is stored. FitZen addresses these requirements through an integrated workflow, a simple interface, and database-level access restrictions.

C. Technical Foundation

The project uses Flask as the backend web framework [3], Bootstrap for responsive interface components [6], and PostgreSQL with Row Level Security policies for database access control [4], [5]. These technologies support the project's emphasis on modular routing, relational records, responsive presentation, and user-level data isolation.

III. METHODOLOGY

FitZen follows a modular three-tier web application structure. The presentation layer accepts user inputs and displays results; the application layer validates requests, performs calculations, manages sessions, and coordinates database operations; and the data layer stores user profiles, food logs, workout records, and progress information. The development approach follows the main software development activities of requirements identification, system design, implementation, and validation [2].

A. System Architecture

The frontend is developed using HTML5, CSS3, JavaScript, and Bootstrap 5. The backend is implemented using Python Flask, with Jinja2 templates for rendering pages and Flask routes for handling requests. Supabase PostgreSQL provides cloud-based relational storage. The python-dotenv package is used to load database endpoints, secret keys, and other sensitive configuration values from environment variables rather than placing them in frontend code.

B. Functional Modules

- User Authentication and Profile: supports registration, login, logout, password hashing, session checks, and storage of baseline profile details such as age, height, weight, and fitness goal.
- BMI Calculator: accepts height and weight, validates the inputs, computes BMI, and displays an adult category with a screening disclaimer.
- Personalized Workout Plans: filters predefined beginner-friendly routines according to goals such as General Fitness, Muscle Strength, and Weight Management.
- Diet Recommendations: presents general balanced meal ideas and healthy-eating guidance aligned with the selected goal; it does not provide clinical prescriptions.
- Calorie Tracker: stores food items with estimated calorie values and calculates daily calorie totals for review.
- Fitness Progress Tracking: records completed workouts with timestamps and presents activity history and consistency summaries.

C. BMI Computation

The BMI module uses the standard calculation shown in (1), where body weight is expressed in kilograms and height is expressed in metres:

BMI = Weight (kg) / [Height (m)]²(1)

The application validates the height and weight inputs before calculation. The resulting value is associated with a standard adult category—Underweight, Normal Weight, Overweight, or Obese—and is presented as a basic screening measure rather than a complete health assessment.

D. Database Design and Security

The relational schema described in the project report contains five principal tables. The users table stores account identifiers, email addresses, password hashes, and creation timestamps. The user_profiles table stores height, weight, fitness goal, and profile update information. The food_logs table records food items, estimated calories, and log dates. The workout_records table stores workout names, durations, and completion timestamps. The fitness_progress table

stores summary values such as BMI, total calories, and recording dates. Primary and foreign key relationships connect user-owned records to the corresponding account.

Row Level Security is enabled on user data tables so that authenticated users can access only records associated with their own user identifier. The project report illustrates this approach with a policy that compares auth.uid() with the user_id field. Password hashing and Flask session validation provide additional application-level safeguards.

E. Application Workflow

The main workflow begins with user registration or login. The user then maintains profile information and enters height and weight for BMI calculation. Based on a selected fitness goal and preferences, the system presents workout and diet suggestions. Food entries are saved in the calorie tracker, while completed workouts are recorded in the progress module. Stored records can subsequently be reviewed through the application dashboard and history views.

IV. RESULT

The project report describes the implementation and validation of the main FitZen workflows. The user interface uses responsive navigation, forms, dashboard cards, and feedback messages. The BMI screen accepts height and weight and displays the calculated value with an adult category. Workout and diet screens filter the available suggestions according to the selected fitness objective. The calorie tracker stores daily food entries and updates the daily total, while the progress module records completed activities and displays historical summaries.

A. Testing and Validation

Four testing phases are described: unit testing of BMI calculation, password hashing, and calorie aggregation; integration testing of Flask routes, environment configuration, database operations, and templates; system testing of the end-to-end registration-to-progress workflow; and security testing of Row Level Security policies to confirm user-level data isolation.

B. Functional Test Outcomes

- Valid registration and login validation were reported as successful; invalid credentials were rejected.
- For a height of 1.75 m and weight of 70 kg, the test case reported a BMI of approximately 22.9 and the Normal Weight category.
- Zero or negative height values were rejected through input validation.
- Selecting the Strength goal displayed the corresponding workout suggestions, while Weight Management displayed matching meal suggestions.
- A food entry such as Oatmeal – 300 kcal was saved and included in the daily calorie total.
- Completed workout entries were stored with timestamps, and the progress history was updated.
- An attempt by one user to query another user's food logs was reported as blocked by the RLS policy.

The project report records PASS for the ten listed functional and security test cases and reports a 100% pass rate across those cases. It also records an average initial page-render time below 280 milliseconds and an average database-query response time of 110 milliseconds under the stated simulated broadband and 4G test conditions. These figures are the measurements documented in the project report and should be interpreted in the context of that test environment.

V. CONCLUSION

FitZen – Personal Fitness and Wellness Tracker brings basic fitness self-monitoring and planning functions together in a single web-based application. The project combines BMI screening, goal-oriented workout suggestions, general diet guidance, daily calorie logging, and fitness progress records. Its modular Flask backend, responsive HTML/CSS/Bootstrap interface, and Supabase PostgreSQL database provide a structured foundation for these features. Password hashing, session validation, environment-variable based secret management, and Row Level Security are included to support the privacy of user records.

The project report describes successful functional testing of the core workflows and user-level database access controls. FitZen is designed as a beginner-friendly wellness support system and does not replace medical diagnosis or individualized professional advice. Future enhancements identified in the project include integration with wearable devices, AI-assisted workout and nutrition suggestions, richer progress analytics, mobile application support, and optional community or gamification features.

ACKNOWLEDGMENT

The author expresses sincere gratitude to the project guide, Prof. M. V. Nikum, for guidance and support throughout the development of this project. The author also acknowledges Shri Jaykumar Rawal Institute of Technology, Dondaicha, for providing the academic environment to undertake this work.

REFERENCES

[1]. World Health Organization, "Global Action Plan on Physical Activity 2018–2030: More Active People for a Healthier World," Geneva, Switzerland, 2018.

[2]. R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York, NY, USA: McGraw-Hill Education, 2020.

[3]. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

[4]. PostgreSQL Global Development Group, "PostgreSQL Documentation: Row Security Policies." [Online]. Available: https://www.postgresql.org/docs/current/ddl-rowsecurity.html

[5]. Supabase, "Supabase Documentation: Architecture, Database Management, and Row Level Security." [Online]. Available: https://supabase.com/docs

[6]. Bootstrap Team, "Bootstrap Documentation." [Online]. Available: https://getbootstrap.com/

[7]. A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed. New York, NY, USA: McGraw-Hill Education, 2020.

[8]. S. H. Patel and A. Kumar, "Digital Health and Wellness: Evaluating User Retention and Data Security in mHealth Web Portals," Journal of Medical Internet Research, vol. 25, no. 3, pp. 115–128, 2022. [As cited in the supplied project report.]