International Research Journal of Education and Technology
Peer Reviewed Journal, **ISSN: 2581-7795**

Fitness and Calorie Tracking Website with Injury Prediction

Mr. G. Jegatheesh kumar, Assistant Professor,

Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

Prasanth N, Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

Abstract

This project presents a comprehensive Food and Workout Tracking Application aimed at helping users effectively monitor and manage their daily nutrition intake and physical activity. With the growing importance of maintaining a healthy lifestyle, the application provides a digital solution that simplifies calorie tracking, macronutrient monitoring, and workout logging through an intuitive and user-friendly interface.The food tracking module allows users to add and manage food items with fully editable quantities, including support for decimal values such as 1.5 or 0.75 servings. Each food entry includes a dedicated edit option, ensuring flexibility and accuracy in daily nutrition tracking without requiring users to delete and re-enter data.Similarly, the workout tracking module enables users to log physical activities with editable parameters such as workout duration (in minutes) and number of sets. Based on these inputs, the application dynamically recalculates the calories burned, providing personalized and precise workout insights. To reduce manual data entry and enhance accuracy, the application integrates a barcode scanning feature

within the food tracker. Users can scan packaged food items using the device camera, and the scanned barcode is automatically processed through a backend API to retrieve corresponding food details from the database. This functionality significantly improves user convenience, minimizes errors, and speeds up the food logging process.For improved usability and visual appeal, food items are displayed along with representative images sourced from Unsplash.The backend of the application is developed using Node.js with the Express framework and MongoDB as the database.

Keywords: fitness tracking; calorie monitoring; injury prediction; wearable health; MERN stack; logistic regression.

1. Introduction

Fitness and nutrition tracking apps are increasingly widespread tools for health management. In 2022, diet and nutrition applications reached approximately 1.4 billion users worldwide, and roughly 45% of these users reported successful weight-loss outcomes by using such apps. Moreover, smartphone-based diet/exercise apps have been shown to produce modest but statistically significant reductions in body weight and BMI over a few months. These mobile tools effectively replicate many components of traditional weight-loss programs (logging, feedback, goal-setting), making healthy behaviors more accessible.

At the same time, advances in wearable sensors and artificial intelligence are transforming how we monitor physical health. Recent reviews note that **AI-enabled wearable devices** are now helping to **prevent injuries and enhance performance** in sports and fitness settings. Smartwatches, fitness bands, and smartphone sensors can continuously collect data (heart rate, motion, sleep) and feed it into algorithms that detect injury precursors before they occur. This AI+wearables approach has significantly changed how people monitor and improve their health: it supports early injury detection and personalized training adjustments.

Despite these trends, most consumer fitness apps focus on retrospective self-monitoring rather than predictive risk assessment. In other words, users can log meals and workouts, but few systems proactively warn about overtraining or injury risk. **Our project aims to bridge this gap.** We design a full-stack web application that combines comprehensive tracking with an ML-driven injury predictor. The main objectives are to provide seamless calorie and workout logging, to compute real-time metrics

(calories in vs. out), and to **integrate an injury-risk model** that alerts users when training loads may be unsafe.

Key Objectives

- Build a secure web app for user profiles, daily diet logs, and exercise records.
- Enable automated calorie calculations (intake vs. burn) and visualize progress over time.
- Integrate a logistic-regression ML module to estimate injury risk from user data.
- Provide an informative dashboard (charts of calories, weight trends, etc.) and risk alerts.
- Use modern full-stack technologies (MERN) and ensure a user-friendly interface.

2. Literature Review

Diet and fitness tracking applications have proven efficacy in weight management. For example, systematic reviews report that smartphone-based diet and exercise apps lead to **significant short-term weight loss and BMI reduction**. Such apps leverage goal-setting, calorie databases, and reminders to help users adhere to diet/exercise plans. The widespread adoption of these tools underscores their impact on health behavior engagement metrics show millions of downloads and daily active usag

Meanwhile, wearable sensor data and AI are rapidly advancing injury prevention research. AI algorithms applied to wearable data (inertial measurements, heart rate variability, GPS tracking) can **identify patterns of fatigue and injury risk before symptoms appear**. In practice, sports teams and rehabilitation programs increasingly use AI-driven analytics – the so-called "Digital Athlete" – to forecast injuries and optimize training loads.

However, existing consumer health apps typically do not leverage this predictive capability. Most focus on logging and feedback after the fact. To our knowledge, few end-user platforms combine nutrition/exercise logging with real-time risk scoring. Therefore, It builds on current research by embedding an injury-prediction pipeline within a conventional fitness-tracking app. This approach addresses a clear opportunity: preventive analytics can encourage safer training, potentially reducing setbacks and improving long-term user outcomes.

3. Proposed System

It is implemented as a *MERN*-stack web application, ensuring a cohesive JavaScript-based environment. The system consists of several interacting modules

- **User Management Module:** Handles registration, authentication, and user profiles. Upon sign-up, each user provides baseline data (age, gender, height, weight, fitness goals). Profile information and credentials are securely stored in MongoDB.
- **Fitness Tracking Module:** Allows users to log workouts and physical activities. Data fields include exercise type (running, cycling, etc.), duration, distance, intensity, and optional sensor inputs (heart rate). This module updates the user's activity history and feeds relevant data to other components.
- **Calorie & Nutrition Module:** Enables logging of meals and nutrition details. Users enter foods and portions (or select from a database), and the system computes the calories and macronutrients consumed. It also maintains cumulative daily totals. This information is used to calculate net energy balance alongside exercise burn.
- **Injury Prediction (ML) Module:** Implements the machine learning pipeline for injury risk. We use a **logistic regression** classifier (built with Python/scikit-learn) that analyzes user data (demographics, recent training history, self-reported fatigue) to estimate a probability of injury. The model outputs risk levels (Low/Medium/High) based on patterns learned from past data. Logistic regression was chosen for its interpretability and proven effectiveness in similar contexts.
- **Analytics & Reporting Module:** Processes stored data to generate user-facing reports and visualizations. The dashboard displays metrics such as current calorie balance (intake vs. burned), charts of weight or activity over time, and the latest injury risk status. It can show trends (a line graph of daily calories) and highlight whether goals are on track.
- **Notification/Alert Module:** Creates real-time notifications for the user. For example, if the injury module flags a high risk, the system alerts the

user: “Your training load is high – consider adding a rest day.” This module also handles reminders (to log today's meals) and motivational messages (. milestones achieved).

These modules communicate through a clear data flow: user inputs are sent via the React frontend to the Express/Node backend, which stores them in MongoDB. The Injury Prediction service (a separate Python process or API endpoint) retrieves relevant user data, computes the risk score, and returns it to the backend. The dashboard then updates to display any new alerts or visualizations. The overall design balances functionality with scalability: adding sensors (wearables) or new analytics would require extending these modules without major redesign.

4.Methodology& Implementation

The It platform is developed using popular open-source technologies. The **frontend** is built in ReactJS, creating a responsive single-page UI for form entries and dashboard views. The **backend API** uses Node.js with the Express framework to handle requests, user sessions, and business logic. All data (user profiles, logs, etc.) are stored in MongoDB, a NoSQL database suited for flexible document schemas. This MERN stack ensures consistency of JavaScript across client and server and simplifies deployment.

During user registration, the system validates inputs (e.g. age or weight must be numeric) and then saves the profile data. Users can then add new entries: meal logs (date, meal type, foods, portion sizes) and workout logs (exercise type, duration, distance, intensity). The server immediately computes calories for these entries meal entries look up a nutrition database (or use user-provided values) to sum intake, while exercise entries estimate calories burned (e.g. using metabolic equivalents). The net calories (intake minus burn) are stored daily.

The core predictive component is the logistic regression model for injury risk. This model is implemented in Python using scikit-learn and is trained on historical sport data (e.g. simulated or sample datasets) that include user attributes and training loads. Features include demographic data (age, weight), recent training volume (hours of exercise in past week), and optional self-reported fatigue/soreness levels. When a user interacts with the system, the backend gathers the latest relevant data and calls the Python service. The model computes a risk

probability and sends it back to Node, which then categorizes it (Low/Med/High). These steps happen in real time when data changes, enabling on-demand risk updates. Studies have shown that logistic regression can achieve high accuracy (around 83%) in similar injury-risk classification tasks, supporting our choice of model.

For testing and quality assurance, we employed standard software testing practices. **Functional testing** verified that core features work as intended: for example, submitting a workout form correctly updated the database and dashboard, and entering a meal log recalculated daily totals. We also tested **user authentication** flows to ensure security(registration forms, password hashing, session handling). The **ML output** was tested using known input cases to confirm that risk levels were reasonable (e.g. unusual training spikes produce higher risk outputs).

5. Results & Analysis

After implementation, we evaluated the system qualitatively through example scenarios. In normal usage, It successfully records entries and updates displays calorie charts and history tables correctly reflect user inputs. For instance, when a user logs a 500 kcal meal and a 300 kcal run, the dashboard shows a net +200 kcal for the day as expected. The injury prediction module also responds to input patterns: e.g. if a user adds three intense workouts in one day, the model's risk output increases compared to a typical routine.

In summary testing, all modules functioned as designed. The **fitness and nutrition tracking** features aligned with user expectations, and visualizations helped illustrate progress. The **injury predictor** produced risk levels that matched our intuitive understanding of load. For example, in our tests, low-intensity weeks corresponded to low predicted risk, whereas erratic spikes (long workouts, high intensity) yielded moderate-to-high risk alerts. We also ensured that alerts and dashboards refreshed promptly when data changed.

No systematic user study was conducted at this stage, but the integrated tests suggest the system meets its goals. Future work should include quantitative evaluation (such as comparing predicted risk against actual injuries in a user study), but our initial analysis confirms that the system operates correctly and provides the intended feedback to the user.

6. Discussion

It demonstrates how combining self-monitoring with predictive analytics can enrich the fitness app experience. By automatically issuing injury-risk warnings, the system helps users adjust training to avoid overuse. This proactive feature is not common in typical fitness apps, making It relatively unique. In context, this aligns with recent research emphasizing AI-powered wearables for injury prevention. Our platform effectively puts some of that technology in the hands of ordinary users without requiring expensive hardware.

There are limitations to note. First, the injury-prediction model relies on the data quality provided. Since we use self-reported exercise logs and optional fatigue ratings, inaccuracies or infrequent logging may reduce prediction accuracy. Also, logistic regression, while interpretable and fast, may not capture very complex interactions. (In prior studies, more advanced models sometimes outperform simpler ones.) As it stands, the model gives **reasonable** risk estimates, but future versions could explore deep learning or ensemble methods for improvement.

Another consideration is data privacy. We store personal health information (weight, workout history, nutrition) in the cloud.

Finally, user engagement is crucial. It's success depends on consistent user logging. Integrating with fitness wearables (e.g. syncing steps or workouts from a smartwatch) could reduce the logging burden. (The current design anticipates such integration as an optional future enhancement.)

7. Future Work & Conclusion

Future enhancements could make It even more powerful and user-friendly

- **Advanced ML Models:** Experiment with richer algorithms. For example, deep neural networks or ensemble methods (random forests, boosting) could capture non-linear patterns in training load. These have outperformed simple models in some activity prediction tasks. Incorporating these may further boost risk prediction accuracy.
- **Wearable Integration:** Directly connect with wearable devices (smartwatches, fitness bands) to automate data collection. Continuous biometric inputs like heart rate

variability and GPS tracking can greatly enhance the injury forecasts. Automatic syncing would also lighten the user's manual logging load.

- **Expanded Data Inputs:** Add factors like sleep quality, stress, or wellness surveys. Emotional and recovery states affect injury risk; our system could include daily questionnaires or integrate sleep tracker data.
- **Mobile and Engagement Features:** Develop a native mobile app with push notifications and gamification. Badges, challenges, and personalized coaching tips could improve user motivation and adherence.In conclusion, It showcases a holistic approach to digital health: users gain both **tracking** and **protective guidance** in one platform. Our implementation on a MERN stack proves that such a system can be built using accessible tools.

Appendices

References

Pujia et al. (2025)[3]; Mahmood (2025)[4]; Cesarelli et al. (2022)[6]; Market.us (2026)[1][2] (stats on nutrition apps).

[1] [2] Diet and Nutrition Apps Statistics and Facts (2026)

https://media.market.us/diet-and-nutrition-apps-statistics/

[3] Journal of Medical Internet Research - The Role of Mobile Apps in Obesity Management: Systematic Review and Meta-Analysis

https://www.jmir.org/2025/1/e66887

[4] [5] The Role of AI and Wearables in Injury Prediction and Performance Optimization - Premier Science

https://premierscience.com/pjai-25-971/

[6] A Logistic Regression Model for Biomechanical Risk Classification in Lifting Tasks - PMC

https://pmc.ncbi.nlm.nih.gov/articles/PMC9689567/