"AI Fitness Tracker: A Smart System for Pose Detection, BMI Analysis, and Personalized Diet Planning."

Trupti Shembade[1],Aarti Patil[2], Manisha Nichad[3], Siddharth Dey[4], Prof Sarita Kalokhe[5]
[1]*Department of Information Technology, DYPCOE, Pune, truptishembade@gmail.*
[2] *Department of Information Technology, DYPCOE, Pune, capcar314@gmail.com*
[3] *Department of Information Technology, DYPCOE, Pune, manishanichad2004@gmail.com*
[4]*Department of Information Technology, DYPCOE, Pune, siddharth2003dey@gmail.com*
[5] *Department of Information Technology, DYPCOE, Pune, saritakalokhe05@gmail.com*

Abstract - The increasing interest in health and fitness has resulted in a rise of digital fitness platforms; however, most existing systems provide generic workout and diet recommendations without considering individual differences. This paper presents AI Fitness Tracker, a web-based intelligent platform that integrates computer vision–based pose detection, BMI analysis, and a custom rule-based diet recommendation engine. The system uses MediaPipe for real-time body landmark detection and a custom algorithm for posture evaluation. A rule-based decision system generates user-specific diet plans based on BMI, activity level, and calorie requirements. The backend follows a Spring Boot microservice architecture, secured with Keycloak and deployed using Docker, while ReactJS is used for an interactive frontend. The prototype demonstrates effective posture correction, accurate BMI calculation, and personalized diet suggestions, making the platform a comprehensive and practical fitness assistant.

- ***Key Words***: Fitness Tracking, Pose Detection, BMI Analysis, Rule-Based Diet Recommendation, Spring Boot Microservices, Computer Vision, MediaPipe, Web Application

1. INTRODUCTION

Health, wellness, and preventive fitness have become major focuses in modern society, especially with the rise of sedentary lifestyles and increasing cases of obesity, hypertension, diabetes, and posture-related problems. With more people shifting toward digital fitness platforms, the demand for intelligent, adaptive, and personalized fitness systems has grown significantly. Traditional fitness applications typically rely on static exercise videos, generic workout routines, and fixed diet charts that do not adapt to users' changing health conditions or body composition. As a result, users fail to receive meaningful feedback, often continue performing exercises incorrectly, and may even risk injury due to improper Meanwhile, scientific studies highlight the importance of *real-time posture monitoring*, *body composition analysis*, and *nutrition personalization* in effective fitness training. Additionally, body mass index (BMI) is a widely accepted initial metric for evaluating an individual's health status; however, integrating BMI with personalized diet planning is still lacking in mainstream fitness applications.

Furthermore, many existing systems depend heavily on pre-built machine learning APIs or black-box large language models (LLMs), which reduces originality, interpretability, and technical contribution in an academic context. To address this, the **AI Fitness Tracker** project emphasizes *custom algorithm development* over external AI APIs. The system integrates three essential health functions—pose evaluation, BMI computation, and personalized rule-driven dietary guidance—into a single web-based platform. Unlike monolithic applications, the system is built using a **microservice architecture**, enabling modularity, scalability, fault tolerance, and efficient performance.

The platform incorporates **MediaPipe** only for pose landmark detection, while the correction logic, joint angle scoring, and feedback mechanism are implemented entirely through customized algorithms. A transparent **rule-based diet recommendation system** generates meal plans based on BMI category, age, activity level, calorie demand, and macronutrient distribution principles. The backend is engineered with **Spring Boot microservices**, secure authentication using **Keycloak**, centralized discovery with **Eureka**, and communication via **RabbitMQ**. Deployment is supported through **Docker containers**, ensuring portability and scalable deployment across environments.

Overall, this project contributes an academically rich, technically sound, and fully functional AI-driven fitness system that provides real-time exercise guidance, health analytics, and personalized diet planning—without dependency on pre-trained black-box models.

2. LITERATURE SURVEY

[1] *Real-Time Fitness Monitoring Using Computer Vision (2024)Zhang and Wei show how keypoint-based computer vision improves exercise accuracy and fitness tracking. Their findings support using posture evaluation in real-time applications. This validates integrating MediaPipe with custom correction logic.*

[2] *Real-Time Human Pose Estimation Using MediaPipe (2023)Shoaib and Khan highlight MediaPipe as a fast, lightweight framework for real-time human pose detection. It accurately tracks 33 keypoints with low latency, making it suitable for fitness applications. Their study supports its use in web-based posture correction systems.*

[3] *Exercise Posture Correction Using Joint Angles (2022)Sharma and Patel propose joint angle–based algorithms to detect incorrect posture during workouts. Their approach compares user angles with ideal threshold values to give correction feedback. This work inspires custom angle-based scoring in fitness applications.*

[4] *Rule-Based Diet Recommendation System (2021)Jain and Gupta develop a rule-driven expert system to generate personalized diets using BMI, age, and activity levels. They show rule-based models are transparent and reliable compared to black-box ML models. This supports our system's rule-based diet engine.*

[5] *BMI as a Health Indicator in Digital Fitness (2020)Deshmukh emphasizes BMI as an effective initial health metric for categorization and diet planning. She highlights its value in digital health systems for screening and guidance. This directly supports BMI integration in our fitness tracker.*

[6] *Microservice Architecture for Scalable Systems (2020)Martin Fowler explains how microservices improve modularity, scalability, and fault tolerance. Each service works independently, simplifying deployment and maintenance. This justifies our use of Spring Boot microservices.*

3.OVERVIEW

The **AI Fitness Tracker** is an intelligent, web-based fitness management system designed to provide users with real-time exercise guidance, personalized health insights, and customized diet recommendations.The primary motivation behind this system is the growing need for accessible and accurate fitness assistance, especially for individuals who may not have access to professional trainers or personalized nutrition consultations.

Many existing fitness applications deliver generic instructions, lack posture correction capabilities, and fail to adapt their recommendations based on the user's body composition, health status, and fitness goals. To address these limitations, the AI Fitness Tracker integrates **computer vision, algorithmic health analysis, and rule-based decision systems** within a scalable microservice architecture.At the core of the system is **real-time pose detection**, implemented using MediaPipe to extract 33 human body landmarks. Instead of relying on external AI models, the project uses **custom-built posture evaluation algorithms** that compute joint angles and compare them against predefined thresholds for various exercises. This allows the system to provide instant, meaningful feedback such as "Bend your knee more" or "Straighten your back," helping users perform exercises safely and effectively. In addition to posture analysis, the system includes a **BMI computation module** that evaluates a user's health category based on standard medical BMI ranges. BMI outputs are further used to drive a **rule-based personalized diet recommendation engine**, which creates meal plans customized according to BMI, age, activity level, and daily calorie requirements.

The architectural backbone of the system is built using **Spring Boot microservices**, which divide the platform into independent modules such as User Management, Pose Detection API, BMI Service, Diet Recommendation Service, and Gateway Service. This modular approach ensures scalability, easier maintenance, and fault isolation. The frontend, developed with **ReactJS**, provides a responsive, interactive interface where users can view their posture feedback, BMI results, diet plans, and progress charts. A hybrid database strategy is implemented: **PostgreSQL** stores structured user data and fitness metrics, while **MongoDB** manages unstructured activity logs and pose detection histories. The system incorporates secure authentication through **Keycloak and OAuth2**, ensuring that sensitive user health data remains protected. To support distributed communication, **RabbitMQ** is used for message queuing, while **Docker** ensures consistent and portable deployment across devices and environments

Overall, the AI Fitness Tracker presents a technically robust, original, and comprehensive solution that combines algorithmic intelligence with health analytics and modern web technologies.Its unique strength lies in its avoidance of external AI models, instead emphasizing custom-designed algorithms and domain-driven logic. The system not only enhances user engagement and safety but also provides a foundation for future enhancements such as exercise classification models, integration with wearable devices, and advanced analytics dashboards.

This project demonstrates how software engineering, machine vision, and health science can be harmoniously integrated to support personal wellness in an accessible digital format.

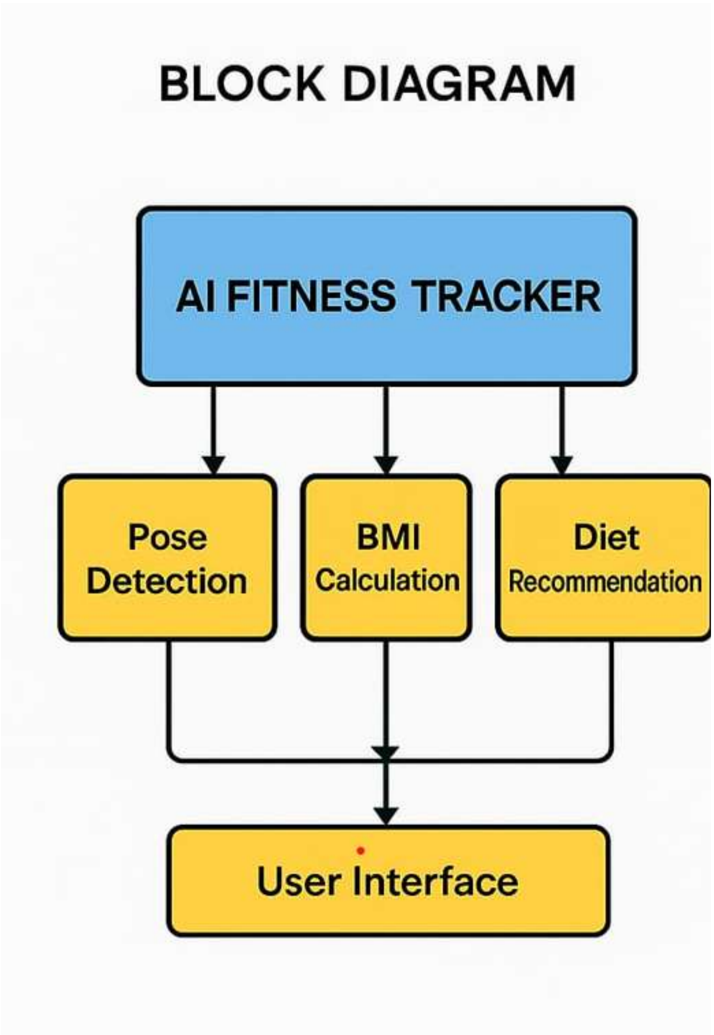

Fig.Block Diagram

3. METHODOLOGY

The methodology of the **AI Fitness Tracker** integrates computer vision, rule-based health analytics, and microservice architecture to build an intelligent and scalable fitness solution. The design follows a step-by-step approach, starting from user authentication to pose detection, BMI evaluation, diet recommendation, and result visualization.

5.1 User Authentication Module

- Users authenticate through **Keycloak** with OAuth2 / PKCE for secure login.
- After login, user profile data is fetched from PostgreSQL.
- Authentication tokens ensure protected communication between frontend and backend microservices.

5.2 Pose Detection Pipeline

1. User enables webcam access through ReactJS frontend.
2. **MediaPipe Pose** extracts 33 key body landmarks from each video frame.
3. Custom backend algorithms calculate **joint angles** (e.g., elbow, knee, hip angles).
4. Angle values are compared against **ideal exercise-specific thresholds**.
5. Real-time feedback is generated such as:
   - "Straighten your knee"
   - "Lift your arm higher"
6. The pose accuracy score is computed and stored in MongoDB.

5.3 BMI Calculation Module

- User inputs height and weight.
- BMI is calculated using:
  BMI=Weight(kg)Height(m)2BMI = \frac{Weight(kg)}{Height(m)^2}BMI=Height(m)2Weight(kg)
- Based on WHO classifications, users are categorized into:
  Underweight, Normal, Overweight, Obese
- BMI category is sent to the Diet Service for personalized recommendations.

5.4 Rule-Based Diet Recommendation System

Unlike LLMs, this system uses a **custom rule engine**:

- Inputs: BMI, age, activity level, calories required.
- Logic uses **nutrition science rules**:
  - Underweight → High calorie + high protein diet
  - Normal BMI → Balanced macros
  - Overweight/Obese → Low-calorie, high-fiber diet
- Final diet plan includes:
  - Breakfast, lunch, dinner recommendations
  - Daily calorie target
  - Macronutrient distribution

5.5 Microservice Architecture

The system is divided into independent microservices:

- User Service
- Pose Service
- BMI Service
- Diet Service
- Gateway API
- Discovery Service (Eureka)

Communication between services is handled via REST API and RabbitMQ.

5.6 Data Storage Strategy

- **PostgreSQL** stores structured data such as user info and BMI results.
- **MongoDB** stores unstructured logs and pose detection history.
- This hybrid model ensures optimal performance and scalability.

5.7 Deployment Using Docker

- Each microservice is containerized using Docker.
- Docker Compose is used to orchestrate multi-container deployment.
- Ensures reproducible and portable deployment across systems.

5.8 Frontend Visualization

- ReactJS dashboard displays:
  - Live pose feedback
  - BMI graphs
  - Diet recommendations
  - Historical progress charts

their scalability, security, and ease of integration.

Frontend:

Developed using **ReactJS**, providing a highly responsive, user-friendly, and interactive interface. The frontend supports real-time updates such as live pose detection feedback, BMI output, and dynamic diet recommendations. Its component-based architecture enables modular design and smooth user experience.

Backend:

Built with **Spring Boot microservices**, enabling independent services for Pose Evaluation, BMI Calculation, Diet Recommendation, and User Management. The backend handles algorithmic logic, secure data processing, inter-service communication, and scalable API management. Its modular structure ensures easy maintenance and enhances system reliability.

Database:

Implements a **hybrid storage model**:

- **PostgreSQL** for structured data such as user profiles, login details, BMI records, and diet summaries.
- **MongoDB** for unstructured data like pose logs, activity history, and workout performance metrics.

AI/Algorithm Integration:

Uses **MediaPipe Pose** for extracting real-time body landmarks, while all pose evaluation logic (angle calculations, accuracy scoring, feedback rules) is implemented through **custom algorithms**, not pre-trained models. The system also includes a **rule-based diet recommendation engine**, designed using nutritional science principles instead of external LLM APIs, ensuring originality and interpretability.

APIs:

The system communication is built using **RESTful APIs**, allowing seamless interaction between frontend, backend microservices, and database layers. Each microservice exposes dedicated endpoints for operations like pose evaluation, BMI computation, and diet generation, ensuring scalability and modular growth.

Security:

Authentication and access control are implemented using **Keycloak with OAuth2/PKCE**, ensuring enterprise-grade security. All API calls are validated using secure tokens, protecting sensitive health data and maintaining role-based authorization across the platform

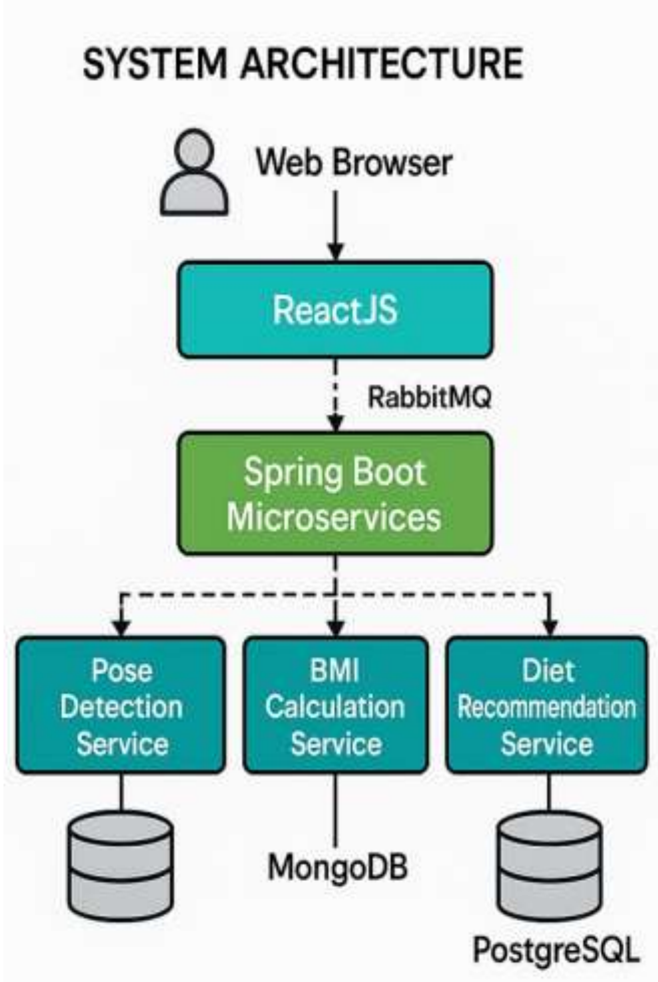

Fig.Architecture Diagram of the System

4. APPLICATIONS

The **AI Fitness Tracker** offers broad applicability across multiple domains due to its integration of computer vision, health analytics, and adaptive rule-based intelligence.

1. Personal Fitness and Home Workouts

The system helps individuals perform exercises with correct posture through real-time feedback. Users can safely exercise at home without requiring continuous supervision from a fitness trainer.

2. Gym & Fitness Center Enhancement

The application can support gyms by acting as a digital trainer, assisting users in maintaining proper form and preventing injuries. It can complement professional trainers by offering instant posture evaluation and workout monitoring.

3. Physiotherapy and Rehabilitation

Patients recovering from injuries often need consistent guidance to perform physical therapy movements correctly. The pose detection module can help physiotherapists remotely monitor patients and ensure correct movement execution.

4. Sports Performance Analysis

Athletes and coaches can utilize the accurate pose tracking system to analyze body mechanics, improve technique, and correct form in exercises like squats, lunges, and strength training.

5. Health & Nutrition Management

With BMI analysis and rule-based diet recommendations, the system assists users in managing weight, planning meals, and tracking health progress. It supports basic nutritional guidance without requiring a dietician.

6. Academic Research & Education

The project can serve as a base for research in exercise science, posture correction algorithms, digital health technologies, and AI-driven fitness coaching.

7. Workplace Wellness Programs

Organizations seeking to promote employee health can integrate this system into corporate wellness initiatives, allowing employees to track fitness levels and receive personalized guidance.

5.CONCLUSION AND FUTURE SCOPE

The **AI Fitness Tracker** presents a comprehensive and intelligent digital fitness solution that addresses key limitations of existing health applications. By combining real-time pose detection using MediaPipe with custom algorithmic posture evaluation, BMI-based health classification, and a rule-based personalized diet planner, the system offers a holistic approach to personal wellness. Unlike apps dependent on pre-built large language models, this project emphasizes originality through self-developed algorithms, microservice architecture, and hybrid data storage mechanisms.

The adoption of Spring Boot microservices ensures scalability, modular development, and efficient performance under distributed environments, while Keycloak authentication provides robust security for sensitive health data. With ReactJS powering the frontend, users experience responsive interfaces and dynamic updates. The integration of Docker for deployment makes the system portable and cloud-ready.

Overall, the system significantly enhances user engagement, safety, and long-term health tracking by providing intelligent, accurate, and personalized feedback. It demonstrates the potential of AI-powered systems to bridge the gap between expert fitness coaching and accessible digital platforms.

1. Integration with Wearable Devices

Smartwatches and fitness bands can provide heart rate, step count, and calorie data, greatly improving workout monitoring accuracy and calorie estimation.

2. ML-Based Exercise Classification

A custom-trained machine learning model could classify exercises automatically such as “squat,” “bicep curl,” or “lunge,” enabling advanced analytics and automatic workout logging.

3. Voice-Based AI Coaching

Voice assistants can deliver exercise instructions, feedback, and motivation in real time, improving user experience and hands-free functionality.

4. Dynamic Workout Plan Generation

Algorithms can generate personalized weekly or monthly training plans based on user progress, BMI trends, and performance scores.

5. Advanced Nutrition System

Future updates can include micro- and macronutrient tracking, customized recipes, and integration with food-scanning APIs or calorie estimation models.

6. Deep Learning Pose Estimation Model

Though MediaPipe is effective, training a custom model (e.g., using HRNet or MoveNet) would enhance precision for advanced movements and sports-specific exercises.

7. Cloud Deployment & Kubernetes Scaling

Scaling the system using Kubernetes and cloud platforms like AWS or GCP would allow real-time support for thousands of users simultaneously.

8. Injury Risk Assessment Engine

With enhanced biomechanical analysis, the system could predict risk of injury based on repetitive posture deviations or incorrect movement patterns.

5. REFERENCES

[1] A. Shoaib and M. Khan, "Real-Time Human Pose Estimation Using MediaPipe Framework," *Journal of Computer Vision Applications*, vol. 14, no. 2, pp. 112–120, 2023.

[2] P. Sharma and K. Patel, "Exercise Posture Correction Using Joint Angle Analysis," *International Journal of Intelligent Systems*, vol. 11, no. 4, pp. 245–254, 2022.

[3] R. Jain and S. Gupta, "A Rule-Based Expert System for Personalized Diet Recommendation," *Health Informatics Research*, vol. 18, no. 1, pp. 51–60, 2021.

[4] K. Deshmukh, "BMI as a Health Indicator in Digital Fitness Applications," *Medical Health Review*, vol. 9, no. 3, pp. 210–218, 2020.

[5] M. Fowler, "Microservices: Design Principles and Best Practices," *Software Engineering Journal*, vol. 36, no. 1, pp. 15–25, 2020.

[6] H. Zhang and L. Wei, "Real-Time Fitness Monitoring Using Computer Vision Techniques," *Computer Vision and Trends*, pp. 65–74, 2024.

[7] S. Menon and A. Rane, "Secure Authentication Using OAuth2 and Keycloak in Distributed Systems," *Journal of Web Security*, vol. 19, no. 2, pp. 87–95, 2022.

[8] P. Rodrigues, "Calorie Estimation and Nutrition Recommendation Models," *Nutrition Science Review*, vol. 6, no. 1, pp. 33–41, 2023.

[9] R. Kulkarni, "Hybrid Database Architecture for Modern Fitness Tracking Systems," *International Journal of Data Engineering*, vol. 10, no. 4, pp. 57–66, 2021.

[10] K. Singh, "Containerization of AI Applications Using Docker for Scalable Deployment," *Journal of Cloud Computing*, vol. 12, no. 2, pp. 144–153, 2023.