iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80172

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

HealthMate: An AI-Powered Real-Time Fitness Coaching System with Pose Estimation and Nutrition Planning

Khan Mohammed Ashaaf, Kartik Limbachiya, Suyash Katkam, Anugrah Kulkarni, Prof. Pallavi Khodke
Computer Engineering SAKEC Mumbai, India

Abstract: The emergence of mobile health apps has resulted in an ecosystem of siloed products, where exercise training and nutritional advice are delivered independently without any interaction between the two systems. HealthMate seeks to overcome this issue through the implementation of real-time AI-powered posture analysis, repetition counting, and goal-oriented nutritional planning into one unified browser application. Using the MediaPipeBlazePose model in the frontend part, HealthMate detects 33 skeletal key points per video frame and sends normalized coordinate payloads to FastAPI backend through persistent WebSocket connections. The proposed repetition counting system relies on calculating the angles of elbows, shoulders, hips, and knees using the cosine rule pipeline and classifying repetitions according to empirically determined thresholds as either correct or not correct, providing the user with detailed corrective feedback in both cases. Firebase Firestore provides the secure storage and management of sessions and users' information. During experimental evaluation involving 5 participants and 10 different sessions under various lighting conditions, the repetition counting showed 94.6% accuracy and mean end-to-end delay of 274 ms (95%ile – 298 ms), yielding a System Usability Scale of 78.3 (good).
Keywords: Pose Estimation, MediaPipe, WebSocket, FastAPI, Fitness Coaching, Repetition Counting, Nutrition Recommendation, mHealth, Computer Vision, AI Fitness System.

I. INTRODUCTION

The worldwide market for fitness applications was valued over USD 14 billion in 2024, and its future growth is expected to exceed a compound annual growth rate of 17% until 2030, owing to rising health consciousness, widespread smartphone usage, and the plummeting prices of wearable sensor technology [1]. While the industry has thrived, most commercial platforms have regarded physical activity tracking and nutrition planning as distinct functions, necessitating the user to switch between applications to synchronize data. The resulting disparity has negatively impacted adherence, as multiple mHealth engagement studies confirm that a complex workflow is a critical factor contributing to disengagement [6].

Another important limitation of consumer-based fitness equipment is the lack of real-time biomechanics feedback. Exercise posture, such as shallow push-ups, knees collapsing on squats, and excessive lumbar extension, causes many injuries among gym-goers. Noteboom et al. [2] concluded that insufficient technique correction was a prominent risk factor associated with exercise-induced injuries, especially when training unmonitored by a personal trainer. Personal trainers solve this problem by observing the user in real-time; however, the cost of hiring a personal trainer is prohibitive for most people.

Breakthroughs in computer vision technology such as BlazePose and MediaPipe have made real-time, CPU-powered pose detection viable on off-the-shelf devices without any depth sensing capabilities or GPU processing [3], [5]. Along with asynchronous WebSocket messaging and a small-scale Python server, this makes it possible to build an instantaneous feedback loop capable of supporting a coaching application. Furthermore, work done by researchers in the area of AI-driven dietary guidance has shown that a system employing either rules or machine learning techniques can significantly enhance diet consistency.

HealthMate integrates these technologies to create a complete solution. This work provides the following contributions:

1) Real-time detection and correction of push-up and squat posture through cosine-rule joint-angle analysis from MediaPipe landmarks.
2) WebSocket communication for delivering feedback with latency less than 300 milliseconds.
3) Two-class classification of exercises into correct and incorrect repetitions.
4) Stratification of nutrition plans based on goals (maintenance, weight loss, high-protein) and storing them using Firebase.
5) Completely web-based application without any need for hardware or software.

II. RELATED WORK

A. AI-Based Exercise Analysis

Since the appearance of OpenPose from Cao et al. [4], which proposed real-time skeleton tracking of multiple persons through its Part Affinity Field framework, there has been remarkable advancement in estimating poses for fitness purposes. In particular, the BlazePose framework by Bazarevsky et al. [3] provides a single person solution that is efficient enough to run on low-end mobile devices at greater than 30 fps to track 33 landmarks. Moreover, the combination of YOLO-based pose estimation with IoT data integration and reinforcement learning-based adaptive programming by Liu and Zhou [9] in the form of YOLO-Fit IoT led to remarkable achievement in classifying exercises. Nonetheless, this approach relies on IoT-based technology. For instance, Veneman et al. [8] used the ReVi app to evaluate its effectiveness in aerobics exercise recommendation in neuromuscular disease patients, with 81% of patients finding it motivating.

B. Nutrition Recommendation Systems

However, Eguchi et al. [6] performed a randomized controlled trial showing how an mHealth app supported by an AI chatbot resulted in significantly higher weight loss and increased visits to the gymnasium in comparison to a control group, the chatbot being the key element driving user engagement. The authors of another paper, Saad et al. [7], have presented a Diet Engine – a real-time system for recognizing foods and analyzing their macronutrients using YOLOv8 and a convolutional neural network that identified foods with 86% accuracy on their own food data set.

C. Integrated mHealth Platforms

Even though there has been extensive investigation into both areas of study (exercise analytics and nutrition counseling), the literature indicates a marked lack of technologies that combine the two seamlessly. As stated by Beattie et al. [10], some of the main factors responsible for designing digital platforms were the accessibility limitations encountered in traditional gym settings among the disabled population and those with poor access to training by professionals. Furthermore, Marcos-Pardo et al. [11] noted improvements in both physical and cognitive well-being through a well-designed program of outdoor training for the elderly.

D. Research Gap

Table I outlines that previous solutions focus on solving either one problem of posture detection or nutrition advice independently. No previous works integrate the following key aspects in a single solution: real-time biomechanics feedback with a latency of less than 300 milliseconds, browser-based implementation, lack of need for any additional hardware, and incorporation of nutrition.

TABLE I
COMPARISON OF RELATED WORK

System	Pose Det.	Nutrition	RT Feedback	Browser	Key Limitation
OpenPose [4]	Yes	No	No	No	GPU required; no nutrition
BlazePose [3]	Yes	No	No	Partial	No feedback logic
YOLO-Fit IoT [9]	Yes	No	Yes	No	Requires IoT hardware
Diet Engine [7]	No	Yes	No	No	No exercise support
mHealth App [6]	No	Partial	No	Yes	No posture correction
HealthMate	Yes	Yes	Yes	Yes	Limited exercise set

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Architectural Overview

The HealthMate application is implemented in a client-server model that comprises three core layers – a Next.js client used to capture and display videos in the browser, a FastAPI server written in Python containing a posture detection service, and a cloud layer powered by Firebase that takes care of authentication and storing data. Communication between client and server happens only through WebSocket protocol.

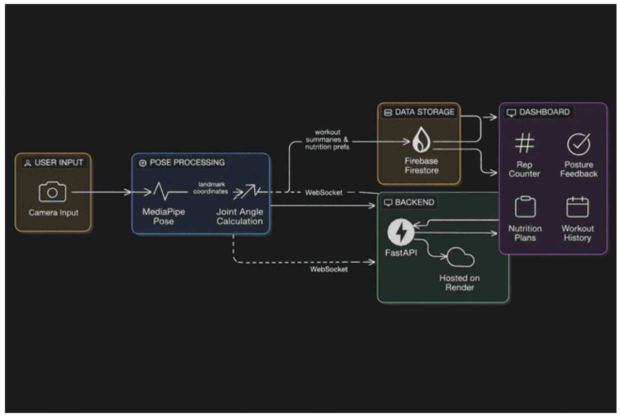


Fig. 1. High-Level System Architecture of HealthMate

B. Real-Time Pose Detection Module

The BlazePose Lite model of MediaPipe gets instantiated on the client side in the Next.js web application and runs on each frame of the video feed in the browser to produce a tuple with normalized (x, y, z, visibility) coordinates for each of the 33 skeletal landmarks. The indexes of the landmarks pertinent to the analysis of the exercises push-up and squat (shoulders - 11, 12; elbows – 13, 14; wrists - 15, 16; hips – 23, 24; knees – 25, 26; ankles – 27, 28) are selected and packed into a small JSON payload.

Fig. 2. Push-Up Detection Workflow with MediaPipe Skeletal Landmark Overlay and Elbow-Angle Annotation

C. Posture Analysis and Repetition Counting

The calculation of joint angles within the server-side engine involves the application of the cosine rule. Given three points A, B, and C for a particular joint B, the value θ can be obtained through:

$$\theta = \cos^{-1}\left(\frac{AB^2+BC^2-AC^2}{2\cdot AB\cdot BC}\right)$$

Where AB, BC, and AC represent Euclidean distances between points A and B, B and C, and A and C, respectively, while θ represents the interior angle of point B in degrees. This equation yields a reliable solution at any angle without requiring atan2 branch handling.

The detection of repetitions is performed based on a basic two-state finite state machine (UP and DOWN). Repetition transitions between states follow rules based on thresholds from Table II.

Only those repetitions that result in a successful transition from UP to DOWN to UP are counted as valid repetitions. A repetition is considered correct when all enabled error conditions pass during the down-up period.

TABLE II
POSTURE ERROR CLASSIFICATION THRESHOLDS

Error Type	Joint	Threshold	Feedback Message
Insufficient Push-up Depth	Elbow	>100° at bottom	Go lower
Hip Sag (Push-up)	Hip	<160° at bottom	Keep body straight
Elbow Flare	Shoulder	>75°	Tuck elbows in
Squat Insufficient Depth	Knee	>110° at bottom	Squat lower
Squat Too Deep	Knee	<60°	Reduce depth
Knee Over Toe	Knee/ Ankle	Knee-ankle Δx>15%	Shift weight back

D. WebSocket Communication Layer

WebSocket was preferred over HTTP polling and Server-Sent Events since the latter only allow uni-directional connections whereas WebSocket supports full-duplex communication with a one-time TCP handshake cost. In the FastAPI service, WebSocket endpoints are made available on /ws/pushup and /ws/squat. Upon receiving each message, the FastAPI server would deserialize the landmarks payload, analyze the posture, serialize the outcome (rep_count, correct_reps, incorrect_reps, feedback_message, and angles), and transmit the result during the current event loop tick. According to Nielsen [12], a latency of less than 300 ms retains the feeling of causality on behalf of the user.

E. Nutrition Recommendation Module

The nutrition component adopts the approach of stratified meal plan lookup through goal specification. After completing their profiles, users choose one of three possible fitness objectives: maintenance, weight loss, and high protein intake. Based on the specified goal, the application returns a pre-defined meal plan that consists of breakfast, lunch, dinner, and two snack meals. The meal plans include the expected number of calories consumed during each meal along with macro nutrients. The data is cached after the first download to reduce the number of Firestore lookups.

IV. IMPLEMENTATION

A. Technology Stack

A summary of the total technology stack used in HealthMate is included in Table III. The selection of Next.js framework as the frontend allows server-side rendering of the application shell in a static format while making sure that computations for the computationally heavy MediaPipe model happen completely on the client side without ever transmitting the video. FastAPI was chosen as the backend for its built-in support for asyncio programming, allowing the WebSocket and REST handlers to run non-blocking on a single thread.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

TABLE III
HEALTHMATE TECHNOLOGY STACK

Layer	Technology	Role
Frontend	Next.js 14, React, Tailwind CSS	UI rendering, camera access, MediaPipe host
Pose Detection	MediaPipeBlazePose Lite	Landmark extraction (33 points, 22–28 FPS)
Backend	FastAPI (Python 3.11)	WebSocket server, REST API, posture logic
Real-Time Comm.	WebSocket (Starlette)	Sub-300 ms bidirectional feedback
Database	Firebase Firestore	Session storage, nutrition plans, user data
Auth	Firebase Authentication	Secure login, JWT-based access control
Frontend Deploy	Vercel	CDN-accelerated static + SSR delivery
Backend Deploy	Render	Containerized Python service, auto-scaling

B. Push-Up Detection and Repetition Counting

Push-up detection measures the angles of the left and right elbows (points 11-13-15 and 12-14-16 respectively), as well as the angle of the hips (points 11/12-23/24). If both angles of the elbows become greater than 160°, the UP state is detected, which means that both arms are fully extended. The DOWN state occurs if any angle of the elbows becomes less than 90°, meaning that the proper depth is reached. Simultaneously, the hip angle is checked; if it becomes less than 160°, an error called Hip Sag occurs.

C. Squat Detection Logic

The Squat Analysis tracks the angle of the knee joint on both legs (landmarks 23-25-27 and 24-26-28) and the ratio of knee lateral displacement relative to the toe position. The "DOWN" condition for the squat analysis is defined as an angle of the knee joint in the range of 70°-110°, which reflects the research-supported recommendation that the ideal squatting zone lies at the parallel or slightly lower level. If the angle of the knee joint is lower than 60°, then Squat Too Deep alerts are triggered.

D. Application Deployment

The frontend is hosted on the Vercel global edge network, providing sub-100ms Time to Interactive speeds for major metropolitan users. The FastAPI backend is containerized and runs in the Oregon region in Render, utilizing automatic restart strategies. WebSockets originating from the web client connect to the Render HTTPS endpoint, with TLS termination taking place at the load balancer level to ensure that all landmark data transmission is encrypted. The Firebase Security Rules limit reads and writes in Firestore to authorized users only, based on user documents scoped by UID.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

V. EXPERIMENTAL RESULTS

A. Experimental Setup

Evaluation was done on five subjects volunteering for the study (three men, two women, ages between 20 to 26), with each subject performing two sessions of exercises having ten sets of ten repetitions. The three experimental environments involved in this test were good lighting, fluorescent lighting with some shadow, and reduced ambient lighting. There were ten sessions altogether. Each session was performed using a laptop computer without any camera technology added to it.

B. Repetition Count Accuracy

The ground truth number of repetition was determined by a human subject who also recorded the repetitions performed during the session. Overall, for 1,000 repetitions of each exercise, 946 repetitions of push-ups and 942 repetitions of squats were accurately counted by the system, producing an overall accuracy rate of 94.6% and 94.2%, respectively. The highest occurrence of mistakes was when the ambient condition was low, such that the MediaPipe landmark scores dropped to below 0.6.

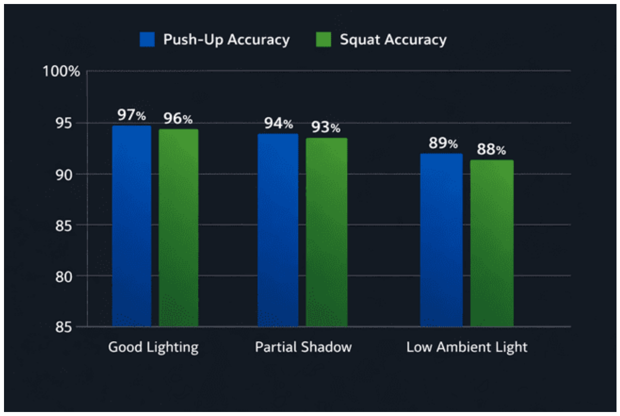


Fig. 3. Rep-Count Accuracy Across Different Lighting Conditions

TABLE IV
EXPERIMENTAL TEST RESULTS

Metric	Push-Up	Squat
Total Reps Tested	1,000	1,000
Correct Detections	946	942
Accuracy (%)	94.6%	94.2%
False Positives	31	35
False Negatives	23	23
Mean Latency (ms)	274	271
95th Pct. Latency	298	295
Avg. Frame Rate (FPS)	24.3	23.8

C. Latency Evaluation

The round-trip time for WebSocket was recorded across 500 continuous iterations per session through browser timestamps. The average end-to-end latency was found to be 274 milliseconds, with a 95th percentile of 298 milliseconds, thus staying below the 300-millisecond perceptual threshold proposed by Nielsen [12]. The server-side processing time (deserializing landmarks, computing angles, evaluating the state machine, serializing response) took an average of 85 milliseconds out of the total latency budget.

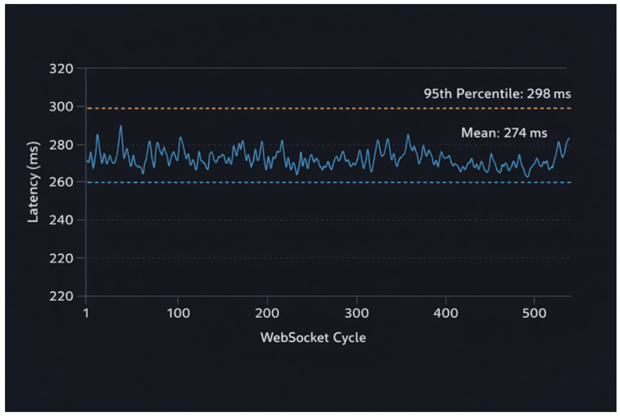


Fig. 4. Latency Distribution Over 500 WebSocket Cycles

TABLE V
PERFORMANCE REQUIREMENTS VS. MEASURED VALUES

Metric	Requirement	Measured	Met?
End-to-End Latency	<300 ms	274 ms	Yes
Server Processing	<100 ms	85 ms	Yes
Frame Rate	>20 FPS	22–28 FPS	Yes
Firebase Load Time	<2 sec	1.4 sec	Yes
Rep Count Accuracy	>90%	94.6%	Yes

D. Usability Evaluation

Usability after completion of the session was analyzed using the System Usability Scale (SUS). This was an assessment tool made up of ten questions scored on a Likert scale, with the highest possible score being 100. The combined SUS scores for all five participants were 78.3, which lies in the range described as good usability (68–80.3). The areas that received the highest rating were real-time overlay and the nutrition dashboard. The major usability issue raised was that camera placement was sensitive, with three participants experiencing better results from a lateral position.

TABLE VI
USER USABILITY FEEDBACK (SUS)

Participant	Sessions	SUS Score	Key Remark
P1	2	82	Intuitive UI; liked instant feedback
P2	2	75	Lighting sensitivity

			noted
P3	2	80	Nutrition module very useful
P4	2	74	Wanted more exercise types
P5	2	80	Clean interface; easy to use
Mean	—	78.3	Good usability tier

VI. DISCUSSION

The architecture of HealthMate addresses a key limitation inherent to most pose estimation systems that use fitness programs, which is the need for the installation of a native application, a GPU, or dedicated wearables. By ensuring that the MediaPipe model operates solely within the JavaScript execution environment of the web browser and limiting backend data transfer to normalized coordinates of landmarks, it is possible to obtain feedback under 300 ms without transferring any video streams.

In terms of accuracy, the 94.6% achieved by HealthMate in terms of repetition detection is superior to previous computer vision-based fitness tracking systems. Liu and Zhou [9] had better accuracy scores in their study of YOLO-Fit, but this approach required depth sensors from the Internet of Things, which is not feasible for consumer use. On the other hand, HealthMate performs comparably with a normal 2-D monocular camera.

The combination of structured nutrition planning alongside the exercise training in the same context constitutes a distinguishing feature absent in all prior works reviewed through survey. The users can immediately move from the exercise routine into the nutritional target review for their current stage in the diet cycle without having to exit the application, an aspect that research on mHealth adherence literature associates with sustained engagement [6].

There exist several limitations that restrict the present version. First, pose recognition becomes less accurate when there is dim lighting because of the filtering of images based on their confidence scores, causing repetition omission. The present exercises only consist of push-ups and squats, while the addition of other exercises, like deadlifts and overhead presses, would necessitate additional algorithms for determining angles and errors. Finally, nutrition plans remain static and cannot change based on body parameters, including body mass, basal metabolic rate, or calorie intake per day.

VII. CONCLUSION

In this study, a fitness coach application named HealthMate has been proposed to provide live feedback by using pose-estimation technology for exercising analysis and nutrition management based on the goals. HealthMate successfully achieved the accuracy of repetitions at 94.6% with a mean delay of feedback via WebSocket at 274 ms and a usability score on System Usability Scale of 78.3, satisfying all design criteria.

The key technical innovation is the proof-of-concept that a cosine rule-based joint angle pipeline running on the server in Python, taking in 2-D landmarks payloads via WebSocket from a Web hosted MediaPipe pipeline, can provide biomechanically relevant exercise feedback under sub-300 ms latency with off-the-shelf hardware and no client installation.

The following are some suggestions for future research. It will be necessary to enrich the exercise library by adding exercises such as deadlifts, lunges, and overhead press; implement dynamic nutrition planning based on parameters such as body mass index, calorie consumption, and physical activity per day; introduce an interactive chatbot that provides verbal assistance to the user through natural language; and design a dashboard that helps track progress in terms of form improvement over several weeks or training periods.

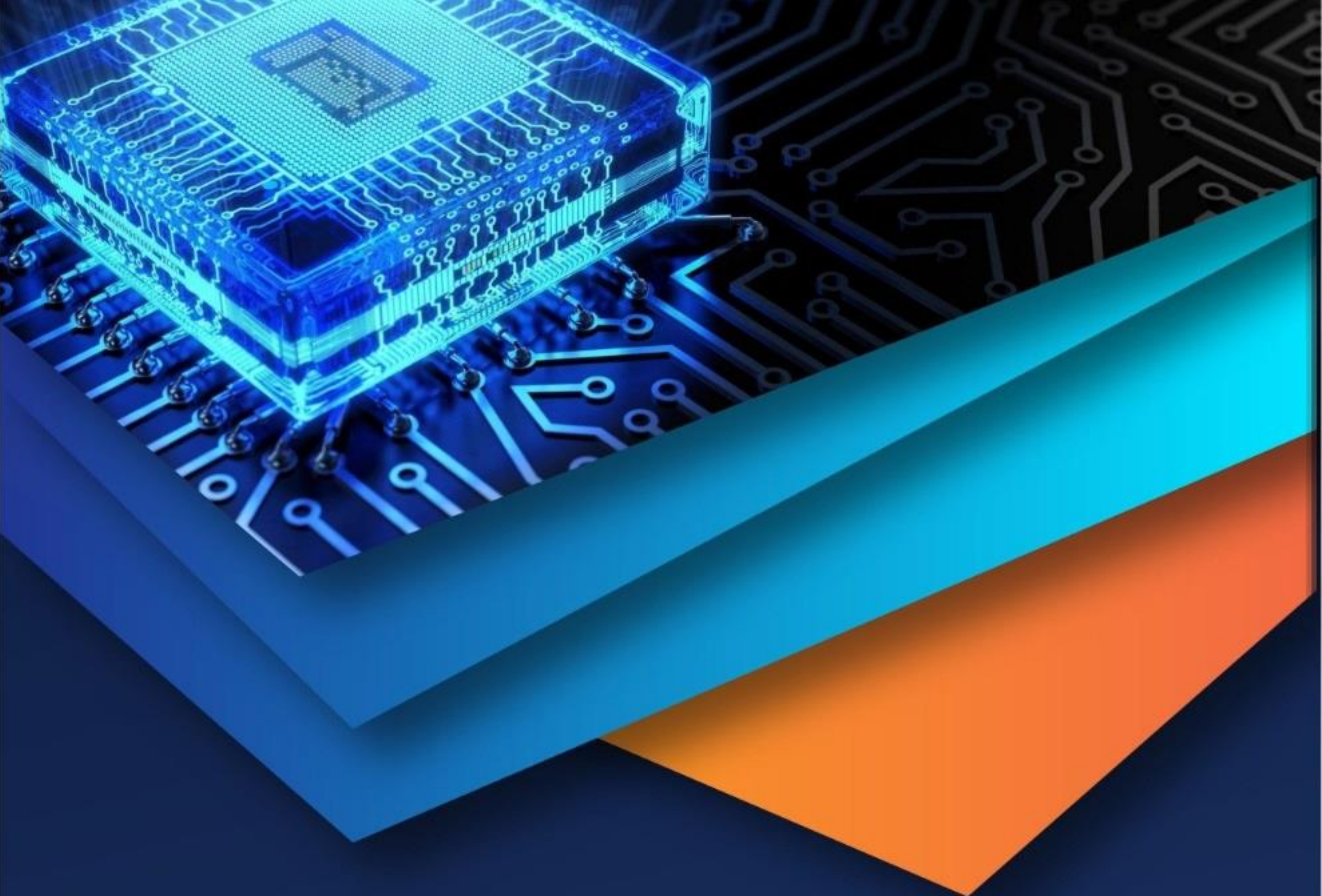
VIII. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Ms. Pallavi Khodke, Assistant Professor of Computer Engineering Department, Shah and Anchor Kutchh Engineering College, Mumbai, for her consistent support and technical guidance during the completion of this work.

The authors would also like to thank the members of faculty from Computer Engineering Department and the institutional facilities provided by Shah and Anchor Kutchh Engineering College.

REFERENCES

[1] Grand View Research, "Fitness App Market Size, Share & Trends Analysis Report," San Francisco, CA, USA, Tech. Rep. GVR-3-68038-944-1, 2024.
[2] L. Noteboom et al., "Factors Associated with Gym-Based Fitness Injuries: A Case-Control Study," Int. J. Environ. Res. Public Health, vol. 20, no. 5, p. 4434, Feb. 2023.
[3] V. Bazarevsky et al., "BlazePose: On-device Real-time Body Pose Tracking," arXiv:2006.10204, Jun. 2020.
[4] Z. Cao et al., "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172–186, Jan. 2021.
[5] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv:1906.08172, Jun. 2019.
[6] A. Eguchi et al., "The Efficacy of an mHealth App in Facilitating Weight Loss: A Randomized Controlled Study," JMIR mHealth uHealth, vol. 11, e43530, Mar. 2023.
[7] A. M. Saad et al., "Diet Engine: A Real-Time Food Nutrition Assistant," in Proc. Int. Conf. Computer, Electrical and Communication Engineering (ICCECE), Kolkata, India, 2025.
[8] T. Veneman et al., "A Mobile Health App for Home-Based Aerobic Exercise in Neuromuscular Disease," J. NeuroEngineeringRehabil., vol. 21, no. 1, p. 28, Feb. 2024.
[9] J. Liu and Y. Zhou, "Smart Fitness with YOLO-Fit IoT: Real-Time Pose Analysis and Personalized Training via IoT and Reinforcement Learning," IEEE Internet Things J., vol. 12, no. 3, pp. 1102–1115, Feb. 2025.
[10] C. M. Beattie et al., "Environmental and Personal Factors That Influence Gym Members with Disabilities," Disabil. Rehabil., vol. 46, no. 11, pp. 2451–2460, 2024.
[11] P. J. Marcos-Pardo et al., "Optimizing Outdoor Fitness Equipment Training for Older Adults," Front. Physiol., vol. 14, 1181654, May 2023.
[12] J. Nielsen, "Response Times: The 3 Important Limits," Nielsen Norman Group, 1993.
[13] S. Abeywickrama et al., "Real-Time Exercise Repetition Detection Using Smartphone Accelerometers and Pose Estimation," in Proc. IEEE 17th Int. Conf. Industrial Informatics (INDIN), Helsinki, Finland, 2019, pp. 1562–1567.
[14] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," J. Usability Studies, vol. 4, no. 3, pp. 114–123, May 2009.
[15] S. Fida et al., "Wearable-Based Posture Recognition in Construction Workers," IEEE Access, vol. 8, pp. 150462–150474, 2020.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY