International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI Gym Assistant: An Intelligent Web-Based Fitness Coach and Real-Time Exercise Pose Analysis System

Mr. Parikshit Barochiya[1] Mr. Aryan Banarase[2] Mr. Soham Bankar[3] Mrs. Vaishnavi Balkawade[4]

Mr. Momin Aman[5]

[1,2,3,4]Student[5]Lecturer

[1,2,3,4,5]Department of Artificial Intelligence Machine Learning (AIML)

[1,2,3,4,5]Zeal Polytechnic, Pune, Maharashtra, India

ABSTRACT

AI Gym Assistant is a comprehensive, web-based intelligent fitness support platform designed to make workout guidance accessible, interactive, and trackable for individuals across all proficiency levels. The system integrates a voice-enabled conversational AI fitness coach, real-time exercise pose estimation, adaptive workout recommendations, Body Mass Index (BMI) tracking, dietary planning with exclusion filters, and visual progress analytics into a unified application. To further expand traditional exercise modalities, this paper introduces a proposed, leveraging suspended aerial gymnastics, decompression movement physics, and specialized spatial keypoint tracking. Developed using Python Flask backend framework and client-side JavaScript execution, core pose estimation and spatial tracking functions execute directly in the web browser via MediaPipe Pose, ensuring low-latency landmark extraction while strictly maintaining user privacy. Joint-angle calculation algorithms coupled with a Finite State Machine (FSM) handle movement phase transitions and accurate repetition counting. Conversational coaching is driven by the Google Gemini API with an automated local keyword fallback mechanism during network dropouts. Session records and physical parameters are stored in SQLite, while Chart.js provides visual analytical reporting. This paper delivers a thorough literature survey, architecture design,, mathematical model, and evaluation plan across 5 structured pages

I. INTRODUCTION

Regular physical activity supports general health and fitness, yet people beginning a workout routine may find it difficult to select suitable exercises, understand correct movement, track consistency, and interpret basic nutrition information Maintaining physical fitness and consistent resistance training is widely recognized as essential for long-term health, cardiovascular endurance, and injury prevention. However, fitness enthusiasts—especially beginners—face numerous hurdles, including lack of access to affordable personal trainers, inability to monitor exercise posture in real time, fragmented information across multiple single-purpose apps, and confusion regarding dietary planning.

Conventional fitness tracking platforms often require users to switch between separate mobile applications for calorie counting, YouTube videos for exercise form demonstration, and standalone web tools for Body Mass Index (BMI) assessment. This fragmented paradigm creates high friction, inconsistent user engagement, and offers zero real-time posture feedback during exercise execution.The result can be a fragmented experience in which guidance and progress records are not organized in one place.

To overcome these challenges, the AI Gym Assistant platform integrates voice-driven conversational coaching, browser-side computer vision posture analysis, adaptive exercise recommendation algorithms, and visual performance tracking into a unified web application.Where pose analysis is enabled, camera frames are processed in the browser rather than uploaded to the application server. This design can reduce server workload and limit transmission of sensitive video, although users should still be informed about browser permissions and data handling.

Furthermore, to address modern innovative movement techniques, joint decompression tracking, and inverted spatial keypoint detection. By conducting vision inference client-side, user video streams remain local, guaranteeing privacy while reducing server infrastructure costs.

II. OBJECTIVES AND SCOPE

The primary engineering and functional objectives of the AI Gym Assistant project include:

• Seamless Web Architecture: Consolidate core fitness guidance utilities into a single, highly accessible web application powered by Python Flask.

• Real-Time Posture and Rep Detection: Utilize MediaPipe Pose to capture 33 3D skeletal landmarks directly inside client browsers, calculating joint vector angles to count reps via a Finite State Machine (FSM).

• Voice-Driven AI Coaching: Connect Google Gemini API with the browser Web Speech API for interactive spoken Q&A, backed by a local keyword engine during offline scenarios.

• Adaptive Recommendations: Develop a rule-based recommendation algorithm ranking a database of 45+ exercises based on user experience level, equipment, and fatigue.

• Health & Analytics Tracking: Compute BMI, generate dietary guidance with allergen filtering, and log workout session statistics into an SQLite database displayed via Chart.js.

• Rank exercises using user goals, experience level, equipment availability, and recent workout history

• To record workout sessions and relevant fitness metrics and display progress using tables or charts.

• To test the application's calculations, recommendations, input validation, AI responses, and error handling

III. EXISTING SYSTEM

The existing fitness-support system often requires users to depend on separate websites, mobile applications, workout videos, BMI calculators, and diet planning tools to obtain fitness information. Users may need to switch between different platforms to understand exercises, calculate their Body Mass Index (BMI), plan workouts & receive answers to fitness-related questions.

Traditional workout video generally demonstrations but may not offer real-time feedback about the user's exercise posture/repetition count Similarly, basic fitness calculators provide individual results without integrating them with workout recommendations or progress tracking. Access to a personal trainer may also be expensive or inconvenient for some users.

Therefore, the existing approach can be fragmented and less convenient for beginners who want to access different fitness-support features in one place.

IV. SURVEY OF RELATED TECHNOLOGIES

A. Human Pose Estimation and Computer Vision

Human pose estimation methods infer anatomical keypoints from images or video. DeepPose demonstrated the use of deep neural networks to regress human joint locations [1], while OpenPose introduced part-affinity fields for associating body parts in multi-person scenes [2]. These approaches helped establish pose estimation as a practical computer-vision task, but their computational requirements and deployment choices vary.

MediaPipe provides reusable perception pipelines designed for real-time applications [3]. MediaPipe Pose can produce a set of body landmarks that are useful for interactive exercise analysis in a browser. In the proposed system, the webcam stream is processed client-side and landmark coordinates are used to estimate angles at joints such as the elbow, knee, and hip. Performance still depends on camera placement, lighting, occlusion, clothing, body orientation, and the exercise being evaluated. Landmark output should therefore be treated as an estimate, not as a clinical measurement.

B. Joint-Angle Analysis and Repetition Counting

A pose model provides coordinates, but a workout assistant must translate these coordinates into meaningful movement events. For three points A, B, and C, with B as the joint, the included angle can be computed from vectors BA and BC:

$$\theta = \cos^{-1}\left((u \cdot v) / (||u||\,||v||)\right)$$

where u and v are the vectors from the joint to the adjacent landmarks. A practical implementation should clamp the cosine input to the valid interval [−1, 1] and reject frames with missing or low-confidence landmarks. A single angle threshold can be unstable because detected landmarks move slightly between frames. A finite state machine addresses this by defining stages such as READY, DOWN, and UP, and allowing transitions only when angle thresholds and minimum timing conditions are satisfied. A repetition is counted only after a valid sequence of movement phases, reducing accidental counts caused by small fluctuations.

Thresholds should be configured for each exercise rather than assumed to work for every body type or camera view. The system should also distinguish an unobserved joint from an incorrect posture; when confidence is low, it is safer to ask the user to adjust camera position than to give a definitive form judgment.

C. Conversational AI and Fallback Handling

Large language models can interpret free-form questions and generate conversational responses. Google provides Gemini API documentation and developer tools for integrating model-based text generation into applications [4]. In AI Gym Assistant, a user's fitness question can be passed to the configured API and the returned response shown in the chat interface. Model responses may be incomplete, generic, or incorrect; prompts, input validation, clear scope limits, and user-facing disclaimers remain important.

A remote API also introduces dependencies on connectivity, credentials, quotas, and service availability. A local keyword-based fallback can handle a limited set of common topics when the API cannot be reached. This fallback is less flexible than a language model, so the interface should not imply that it provides equivalent conversational understanding. Errors and fallback use should be handled explicitly rather than silently returning misleading responses

D. Voice Interaction in the Browser

The Web Speech API exposes browser capabilities for speech recognition and speech synthesis [5]. Recognition can convert supported spoken input into text, while synthesis can read an answer aloud. These capabilities can be useful when a user is exercising and cannot easily type. Availability, language support, and recognition quality differ by browser and device, and microphone access requires permission.

E. Large Language Models and Local Resilience

Generative AI models like Google Gemini supply natural language responses to unstructured user fitness questions. However, reliance on remote APIs introduces vulnerabilities such as rate limits or connectivity loss. Modern production systems employ local keyword-matching engines that gracefully take over during network failures, ensuring consistent system availability

F. Recommendation, BMI, and Progress Tracking

A transparent rule-based recommendation engine is suitable for a student prototype because each recommendation can be traced to the user's stated goal, proficiency, available equipment, exercise category, and recent activity. A curated exercise library can be filtered first by constraints and then ranked by relevance. Recommendations should explain their basis and avoid implying that a simple rule set can fully personalize training or assess medical suitability.

BMI is calculated as body mass in kilograms divided by the square of height in metres. It can provide a basic screening indicator but does not directly measure body composition or determine an individual's health. Diet guidance should remain general, apply user-selected exclusions where supported, and avoid medical claims. SQLite can persist workout and BMI records in a lightweight relational database, while Chart.js supports

browser-based charts for reviewing trends [6], [7]. Stored records should be validated and protected appropriately

V. COMPARATIVE ANALYSIS

Table I summarizes the feature spectrum across existing conventional fitness solutions and the proposed AI Gym Assistant platform.

Feature Metric	Mobile Fitness Apps	Personal Trainer	AI Gym Assistant (Proposed)
Real-time Posture Analysis	None / Video recording	Direct human inspection	Browser-side MediaPipe ML
Aerial Support	Not available	Specialized studios only	Inverted spatial keypoint mapping
Voice AI Interaction	Static automated text	Human conversation	Gemini LLM + Web Speech API
Cost & Data Privacy	Paid / Cloud upload	High recurring expense	Free / Zero video cloud upload

TABLE I.
COMPARATIVE ANALYSIS OF FITNESS PLATFORMS

VI. PROPOSED SYSTEM ARCHITECTURE

AI Gym Assistant is a Flask [20] application with Jinja2 templates. As Fig. 1 shows, the browser handles the interface, pose tracking and speech, while the server handles routing, input validation and the progress, BMI, diet, recommendation and coach modules. Each module is a separate Python file, so the modules can be tested independently. Data are stored in SQLite, and the coach calls the Gemini API through the google-genai Python SDK when an API key is configured.

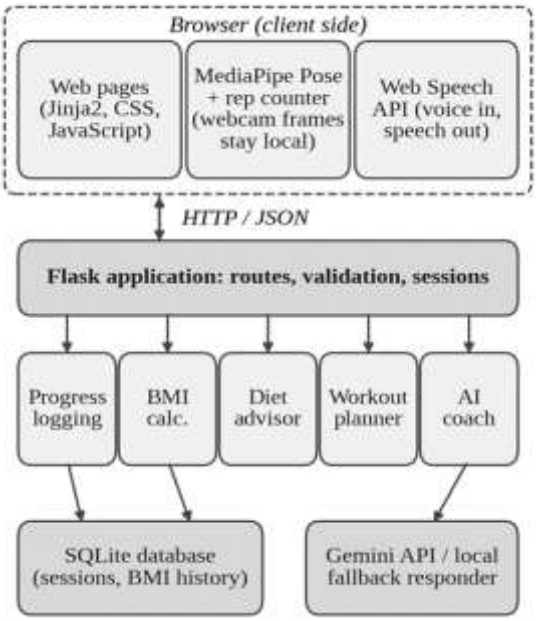


Fig. 1. Architecture of AI Gym Assistant

The proposed architecture has a browser-facing layer, a Flask application layer, and a persistence layer. The browser presents the dashboard and module interfaces. Flask routes requests, validates submitted values, and invokes the relevant server-side logic or external AI service. MediaPipe Pose runs in the client browser for pose analysis where implemented, while SQLite stores persistent workout and BMI history. Chart.js renders stored records as trend visualizations.

A. Dashboard and Navigation

The dashboard serves as the main entry point and links users to the AI coach, pose analysis, workout recommendations, BMI and diet support, and progress pages.Consistent navigation and input validation help reduce confusion and make the website usable for beginners

B. Pose Analysis Pipeline

The browser requests camera permission and obtains frames through standard browser media interfaces. MediaPipe Pose extracts landmarks; the application checks landmark visibility, calculates relevant joint angles, and passes the measurements to the FSM. The FSM tracks exercise phases and updates repetition counts and feedback. Camera frames are intended to remain in browser memory rather than being uploaded to Flask. This is a design choice and should be confirmed in the implementation and tested for unintended data transmission.

C. AI Coach and Recommendation Engine

The coach accepts text or supported speech input. If Gemini is configured and reachable, the application requests a response through the API; otherwise, a local keyword handler responds to supported common queries. Separately, the recommendation module applies user-selected goals, proficiency, equipment constraints, and recent workout information to filter and rank exercises from the curated library.

D. Persistence and Analyticsl

The application stores workout sessions and BMI history in SQLite using validated inputs and parameterized database queries. Progress views summarize the available history in tables and Chart.js graphs. The design should define what is stored, avoid collecting unnecessary information, and provide appropriate safeguards for user records

E. Voice-Enabled Conversational AI

Captures user speech via the browser's Web Speech Recognition API and sends prompts to the Gemini API SDK. When offline, a keyword fallback engine intercepts prompts, delivering relevant pre-verified guidance. Responses are spoken aloud via browser SpeechSynthesis.

VII. METHODOLOGY AND IMPLEMENTATION FLOW

The proposed workflow is organized into the following stages:.

- User input: the user selects a module and enters body measurements, fitness goals, experience level, available equipment, or a question.
- Request handling: the website validates the input and routes it to the relevant Flask handler or browser-side module.
- Feature processing: BMI and diet logic process health-related inputs; the recommendation engine ranks suitable exercises; pose analysis extracts landmarks locally when enabled.
- AI response: the coach sends a query to Gemini when configured and available, or uses the limited local fallback.

- Exercise analysis: joint-angle calculations and the FSM classify movement stages and count completed repetitions.
- Data storage: workout records and BMI history are written to SQLite with input checks.
- Visualization: stored records are summarized in tables and charts for progress review.
- Testing and review: test cases check calculations, valid and invalid inputs, recommendations, route behavior, API failures, and error handling.

This workflow combines formula-based processing, rule-based logic, computer vision, and an optional external AI service. Not every feature requires a machine-learning model. The implementation should keep module boundaries clear so that each function can be tested independently.

VIII. EVALUATION PLAN AND LIMITATIONS

A reliable evaluation should measure behavior rather than rely only on a successful demonstration. Pose analysis can be tested under different lighting, camera distances, clothing contrasts, and viewpoints. Repetition counts should be compared with manually annotated counts across repeated trials, and the error rate should be reported by exercise. Tests should also record cases where landmarks are missing or confidence is low.

For the conversational coach, evaluation can use a set of common fitness questions, check whether responses remain within the intended general-guidance scope, and measure response latency under normal connectivity and API failure. The local fallback should be tested separately to document the topics it can handle. Recommendation tests should verify that goals, equipment constraints, and experience levels affect the ranking as intended. BMI tests should cover valid values, boundary values, invalid input, and unit handling. Database and route tests should check persistence and understandable failure messages.

No measured accuracy, latency, or user-study results are asserted in this survey because those results must come from completed tests on the actual build. Expected limitations include sensitivity of pose estimation to camera conditions, fixed thresholds that may not generalize to all users, possible errors in AI-generated guidance, browser differences in speech support, and the limited personalization of rule-based recommendations. These limitations should be disclosed in the user interface and project documentation.

IX. FUTURE SCOPE

This survey presented the design rationale and technical foundations of AI Gym Assistant, a proposed web-based platform for combining selected fitness-support functions. Pose estimation and joint-angle analysis can support exercise-phase tracking; a finite state machine can make repetition counting more robust than a single threshold; conversational AI and a fallback handler can provide different levels of question-answering support; and database-backed charts can help users review their activity history. The practical value of the project lies in integrating these modules into a simple interface and evaluating them transparently.

Future work may include broader exercise coverage, improved landmark confidence handling, adaptive thresholds, more detailed response validation, secure user profiles, usability testing, and wearable-device integration. Any extension should be implemented and tested before being presented as an available feature. The system should remain a source of general guidance and should direct users to qualified professionals when a question requires medical or individualized clinical advice.

X. CONCLUSION

This survey reviewed the techniques needed for a browser-based fitness assistant and described how they were combined in AI Gym Assistant. Lightweight pose estimation allows form tracking on modest hardware without uploading video; angle thresholds with a state machine give transparent repetition counting; language models make a coach flexible but need safeguards and a fallback; and rule-based scoring keeps recommendations explainable. The prototype is not clinically validated and no accuracy claims are made. Its value lies in showing how these parts can be integrated responsibly, and the evaluation plan sets out how its effectiveness can be measured next.

ACKNOWLEDGMENT

The authors express sincere gratitude to Prof. Momin Aman for project guidance, as well as the Department of Artificial Intelligence and Machine Learning at Zeal Polytechnic, Pune, for technical support and resources.

REFERENCES

[1] Y. J. Oh , J. Zhang, M.-L . Fang , and Y. Fukuoka, “A systematic review of artificial intelligence chatbots for promoting physical activity, healthy diet, and weight loss,” vol. 18,

[2] A. Toshev and C. Szegedy, “DeepPose: Human pose estimation via deep neural networks,” in *Proc. IEEE Conf. Computer. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1653–1660.

[3] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, “OpenPose: Realtime multi-person 2D pose estimation using part affinity fields,” *IEEE Trans. Pattern Anal.Mach. Intell.*, vol. 43.

[4] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang ,and M. Grundmann, “BlazePose : On-device real-time body pose tracking.

[5] D. Smilkov *et al.*, “TensorFlow.js: Machine learning for the web and beyond,”

[6] A . Jaiswal, G . Chauhan, and N. Srivastava, “Using learnable physics for real-time exercise form recommendations,” in *Proc.*

[7] World Wide Web Consortium (W3C), “Web Speech API,” W3C Community Group Report. Available: https://webaudio.github.io/web-speech-api.

[8] Gemini Team, Google, “Gemini: A family of highly capable multimodal models,”

[9] Z. Ji *et al.*, “Survey of hallucination in natural language

generation,"

[10] W3C Community Group, "Web Speech API," Draft Community Group Report. [Online]. Available: https://webaudio.github.io/web-speech-api/.

[11] SQLite Consortium, "About SQLite." [Online]. Available: https://www.sqlite.org/about.html

[12] Chart.js Contributors, "Chart.js documentation. [Online]Available: https://www.chartjs.org/docs/latest/

[13]Pallets, "Flask documentation." [Online]. Available: https://flask.palletsprojects.com/