Personalized AI Workout Companion

G. MANOJ KUMAR, R. REVANTH KUMAR

Assistant Professor, MCA Final Semester,
Master of Computer Applications,
Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India.

ABSTRACT:

Personalized ai workout companion is an intelligent offline fitness application designed to assist users in performing exercises correctly by providing real-time feedback and tracking their progress. Leveraging advanced computer vision techniques and pose estimation powered by MediaPipe, this system accurately detects and counts repetitions for various workouts such as squats, pushups, curis, jumping jacks, and crunches. Users can engage in personalized fitness sessions where the trainer monitors form and counts reps automatically, helping them stay motivated and avoid injuries caused by incorrect posture. The application supports session summaries, allowing users to review their performance, track improvements, and maintain motivation over time. This Al-driven fitness assistant operates without the need for an internet connection, making it accessible and private. It is ideal for individuals seeking guidance during home workouts, fitness enthusiasts aiming to optimize their routines, and trainers looking to offer tech-enhanced coaching. By integrating artificial intelligence with practical fitness applications, Personalized ai workout companion revolutionizes the way users approach exercise, combining convenience, accuracy, and personalized coaching into a seamless offline experience.

Index Terms : AI Fitness Trainer, Pose Estimation, Computer Vision, Media Pipe, OpenCV, Real-time Exercise Tracking, Repetition Counter, Flask, Offline AI Application, Human Activity Recognition, Angle Detection, Fitness Monitoring, Python

1. INTRODUCTION

A Personalized AI Workout Companion is an intelligent, computer-based system designed to assist users during fitness sessions by providing real-time feedback, posture analysis, and repetition counting. Leveraging advancements in computer vision, pose estimation, and machine learning, this application mimics the role of a human fitness trainer, helping individuals perform exercises correctly and stay motivated—without requiring constant human supervision [1]. The system uses a live video feed captured through a webcam and applies MediaPipe's pose estimation technology to detect key body landmarks in real time. These landmarks are analyzed to calculate joint angles and movement patterns, allowing the system to identify and track exercises such as squats, pushups, crunches, and jumping jacks [2]. It provides instant visual feedback by overlaying repetition counts on the video stream and summarizes the entire session once completed.

This AI Workout Companion is built as an offline web application using Flask, ensuring both accessibility and data privacy. It allows users to engage in guided workouts without uploading personal data to the cloud, making it ideal for privacy-conscious individuals [3]. The use of OpenCV enables smooth handling of real-time video frames, while exercise-specific logic detects posture transitions and counts valid repetitions accurately.

One of the key benefits of a personalized AI workout system is the customization it offers. By analyzing user movements and providing tailored feedback, it enables more effective and safe workouts. This is especially valuable for beginners who need guidance on proper form, or for experienced users looking to track progress over time [4]. It also removes the need for expensive fitness subscriptions or personal trainers, making health and wellness more affordable and accessible.

In conclusion, the Personalized AI Workout Companion serves as a next-generation fitness assistant that brings together artificial intelligence and physical training. By automating the core functions of monitoring and feedback, it empowers users to work out anytime, anywhere—effectively transforming their environment into a smart gym. As AI continues to evolve, such companions are expected to grow more accurate, interactive, and supportive of diverse fitness needs [5].

1.1 Existing System

Currently, there are several fitness applications and AI-based tools that aim to assist users in maintaining physical health and form during exercise. Some popular examples include:

- **Nike Training Club / FitOn / Freeletics:** These are mobile fitness apps that provide guided workouts, video instructions, and personalized plans. However, most of them require either manual input or premium subscriptions for full access [1].
- **Smart Mirrors / Wearable Fitness Devices:** Advanced devices such as smart fitness mirrors and smartwatches can provide posture feedback or workout stats, but they often require costly hardware and may lack real-time pose tracking for all types of

exercises [2][3].
Despite these advancements, many systems still fall short in delivering real-time, personalized feedback using computer vision without relying on cloud services or external hardware [4][5].

1.1.1 Challenges

• **Lack of Real-Time Feedback:** Most existing fitness applications do not offer immediate feedback on exercise form or repetition accuracy. They rely on user input or batch video analysis, which is not suitable for interactive sessions [1].
• **Dependency on Hardware:** Advanced AI-powered systems often require external sensors, wearable devices, or expensive cameras, making them inaccessible to a wide range of users [2].
• **No Offline Functionality:** Many fitness apps depend on cloud-based processing or streaming, limiting their usability in areas with poor internet connectivity or for users with privacy concerns [3].
• **Limited Exercise Recognition:** Applications often focus on one or two specific types of workouts and lack generalized pose detection capabilities for a variety of exercises such as squats, pushups, crunches, etc. [4].
• **Manual Rep Counting:** A significant number of workout platforms rely on manual rep tracking, which is prone to user error and lacks automation.
• **Posture Analysis Accuracy:** Some systems fail to account for subtle form issues due to basic pose detection algorithms, which do not calculate joint angles or movement patterns precisely [5].

1.2 Proposed System

The Personalized AI Workout Companion is an offline, AI-powered fitness system that uses MediaPipe for pose estimation and OpenCV for real-time video processing. Built with a Flask-based web interface, it detects user movements via webcam, calculates joint angles, and tracks repetitions for exercises like squats, pushups, crunches, and jumping jacks [1][2]. Unlike existing platforms, this system works entirely offline—offering privacy, real-time feedback, and automatic rep counting without requiring wearables or cloud services [3][4]. After completing a workout, users receive a summary report showing the total reps performed per exercise, all from within a simple and accessible web interface [5].

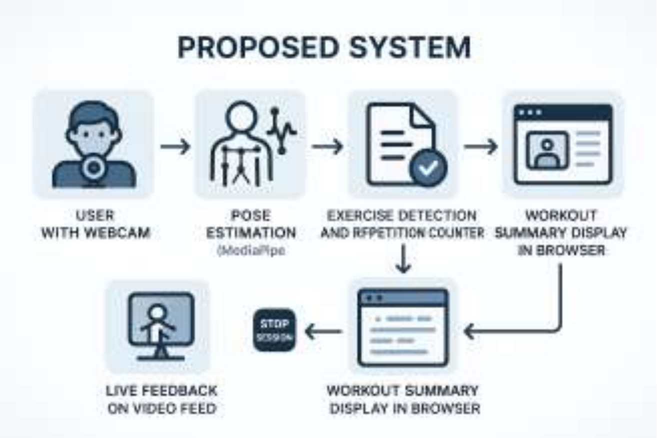

PROPOSED SYSTEM
USER WITH WEBCAM
POSE ESTIMATION (MediaPipe
EXERCISE DETECTION AND RFPETITION COUNTER
WORKOUT SUMMARY DISPLAY IN BROWSER
STOP SESSION
LIVE FEEDBACK ON VIDEO FEED
WORKOUT SUMMARY DISPLAY IN BROWSER

Fig: 1 Proposed Diagram

1.2.1 Advantages

- **Real-Time Feedback**: The system instantly processes user movements and displays rep counts on-screen, simulating a live trainer experience.

- **User-Friendly Interface**: The interface is clean and browser-based, requiring no prior technical knowledge or app installation.

- **Offline Functionality**: As an offline solution, it respects user privacy and works without internet access or cloud dependencies.

- **Multi-Exercise Support:** Tracks multiple exercises including squats, pushups, jumping jacks, and crunches using landmark-based analysis.

- **Posture Detection:** Uses angle-based evaluation to determine correct form and ensures reps are counted only when valid.

- **Cost-Effective:** Eliminates the need for expensive personal trainers or fitness hardware, making home workouts affordable.

- **Visual Progress Display:** Shows exercise names and repetition counts directly on the video feed for clarity and motivation.

- **Session Summary:** After ending the workout, users receive a structured breakdown of all reps completed, promoting goal tracking.

- **Privacy-Safe:** No user data is uploaded or stored externally, making the solution highly secure.

2. LITERATURE REVIEW

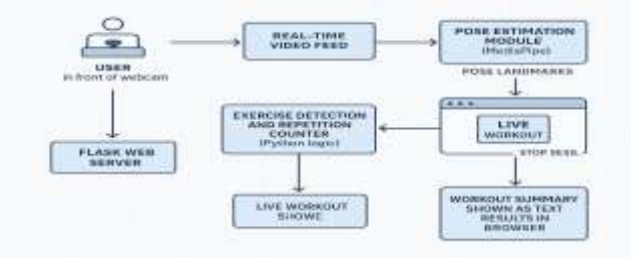
2.1 Architecture

Architecture of Personalized AI Workout Companion:

The Personalized AI Workout Companion is built using a modular architecture that combines computer vision, pose estimation, and web technologies to deliver real-time exercise feedback. The system is entirely offline and leverages well-optimized libraries like MediaPipe, OpenCV, and Flask for seamless performance [1][2].

The development of AI-based fitness systems has evolved through multiple stages:

- **Manual Supervision**: Early fitness tools relied heavily on static instructional videos or human monitoring, offering no interactivity or personalization.
- **Sensor-Based Systems**: With the advent of wearables (e.g., smartwatches, fitness bands), exercise tracking became more personalized but still lacked real-time feedback on form and posture [3].
- **Vision-Based Pose Estimation**: Introduction of frameworks like OpenPose and MediaPipe enabled accurate real-time human pose tracking from video input, removing the need for wearables [1][4].
- **AI-Powered Movement Analysis**: Joint angles and motion trajectories are computed from pose landmarks to detect specific exercises and stages (e.g., "up" or "down" positions in squats or pushups) [2].
- **Web-Based Integration**: The system is hosted using a lightweight Flask web server, rendering the video feed with real-time feedback and interactive session management through browser UI [5].
- **Feedback and Summarization**: The application overlays real-time rep counts on the video feed and provides a workout summary at the end of each session, promoting engagement and self-monitoring.


USER
in front of webcam
REAL-TIME VIDEO FEED
POSE ESTIMATION MODULE (MediaPipe)
POSE LANDMARKS
FLASK WEB SERVER
EXERCISE DETECTION AND REPETITION COUNTER (Python logic)
LIVE WORKOUT
STOP SESS.
LIVE WORKOUT SHOWE
WORKOUT SUMMARY SHOWN AS TEXT RESULTS IN BROWSER

2.2 Algorithm:

The core algorithm of the Personalized AI Workout Companion uses computer vision and pose analysis to recognize and evaluate user exercises in real-time. The system relies on MediaPipe's pose estimation model to identify human body landmarks, and applies angle-based logic to count valid repetitions [1][2]. The overall execution follows this sequence:

1. **Video Capture**:
The webcam stream is accessed using OpenCV's VideoCapture() function to continuously retrieve frames [3].

2. **Pose Estimation**:
Each frame is passed to MediaPipe's Pose model, which detects 33 key body landmarks. The model returns coordinates (x, y, visibility) for each landmark [1].

3. **Angle Calculation**:
For specific exercises, key joints are used to form angle triplets (e.g., hip-knee-ankle for squats). A custom calculate_angle() function computes the joint angle using vector trigonometry [2].

4. **Stage Detection**:
The exercise stage is tracked (e.g., "up" or "down") using thresholds on joint angles. For example, a squat is detected as "down" when the knee angle drops below 90° [4].

5. **Repetition Validation**:
A repetition is counted only when the user completes a full motion cycle (e.g., down → up) and a minimum time delay has passed since the last valid rep [4].

6. **Real-Time Display**:
The current rep count and exercise name are overlaid on the video feed using OpenCV's putText() method [3].

7. **Session Summary**:
After the user stops the session, all exercise counters are displayed on a result page via Flask's API route (/session), giving users a detailed workout summary [5].

2.3 Techniques:

Personalized AI Workout Companion combines several AI, computer vision, and web development techniques to provide an interactive, offline fitness experience:

• **Pose Estimation with MediaPipe**:
The system uses MediaPipe's holistic pose model to detect full-body landmarks in real-time from webcam input. This eliminates the need for additional sensors or devices [1].

• **Angle-Based Movement Analysis**:
A custom-built calculate_angle() function processes landmark coordinates to determine joint angles. These angles are used to detect motion stages (e.g., squatting down or pushing up), enabling accurate rep counting [2].

• **Stage-Based Rep Counting Logic**:
Each exercise has predefined thresholds for entering "down" or "up" stages. Only when a complete transition occurs within a time interval, a repetition is counted—ensuring precision [3].

• **Real-Time Video Streaming via OpenCV + Flask**:
The video stream is processed frame-by-frame in Python using OpenCV, and delivered to the web UI using Flask's Response() method with MJPEG encoding [4].

• **Browser-Based UI**:
The frontend is built using HTML, CSS, and JavaScript, allowing users to start, stop, and view workout summaries without installing any software. It ensures responsiveness and device independence.

• **Offline Functionality**:
The entire system is designed to run offline, preserving user privacy and enabling access in low-connectivity environments. All processing is done locally on the client's machine [1][5].

• **Session Summary via JSON API**:
Workout statistics are delivered through a dedicated /session API endpoint, allowing the frontend to dynamically display a structured summary at the end of each session.

2.4 Tools:

Several tools and frameworks were used in the development of the Personalized AI Workout Companion to ensure real-time performance, cross-platform compatibility, and ease of use:

• **Python**:
The core programming language used for backend development, video processing, and exercise tracking logic [1].

• **Flask**:
A lightweight Python web framework used to handle HTTP routes, serve templates, and manage real-time video streaming via MJPEG [2].

• **OpenCV**:
Used for capturing video frames from the webcam, overlaying text (rep counts), and encoding frames for streaming [3].

• **MediaPipe**:
Google's pose estimation framework used for detecting body landmarks in real-time. This tool powers the core movement tracking functionality [4].

• **HTML/CSS/JavaScript**:
Used for creating a responsive and user-friendly browser-based interface that allows users to interact with the system effortlessly.

• **Jinja2**:
A templating engine bundled with Flask to render dynamic HTML pages like the live workout feed and result summary.

• **NumPy**:
Utilized for efficient numerical computation, especially in angle calculations based on body landmark coordinates [5].

2.5 Methods:
The development of the **Personalized AI Workout Companion** followed a structured methodology to integrate computer vision, web technologies, and exercise logic seamlessly. The workflow consists of the following steps:

1. **Live Camera Access**:
The user accesses the homepage (index.html), which displays a live video stream captured using OpenCV and served through the /video_feed route [1].

2. **Pose Detection and Landmark Tracking**:
Each video frame is processed using MediaPipe's Pose model to identify key body landmarks in real-time [2].

3. **Exercise Recognition and Repetition Tracking**:
Detected landmarks are passed to the track_exercise() function in exercise_tracker.py, where joint angles are calculated and stage transitions (e.g., down → up) are used to count valid repetitions [3].

4. **Feedback Overlay**:
Using OpenCV, the system overlays real-time feedback on the video stream, displaying exercise names and rep counts as the user performs them [4].

5. **Session Control**:
When the user clicks the “Stop Session” button, a POST request is sent to /stop_session, halting exercise tracking and redirecting to the result page.

6. **Session Summary and JSON API**:
Upon stopping the session, /session serves the recorded rep counts as a JSON object, which is dynamically rendered in result.html [5].

7. **Result Viewing and Navigation**:
The user can view a breakdown of total reps per exercise and return to the homepage to start a new session.

3. METHODOLOGY

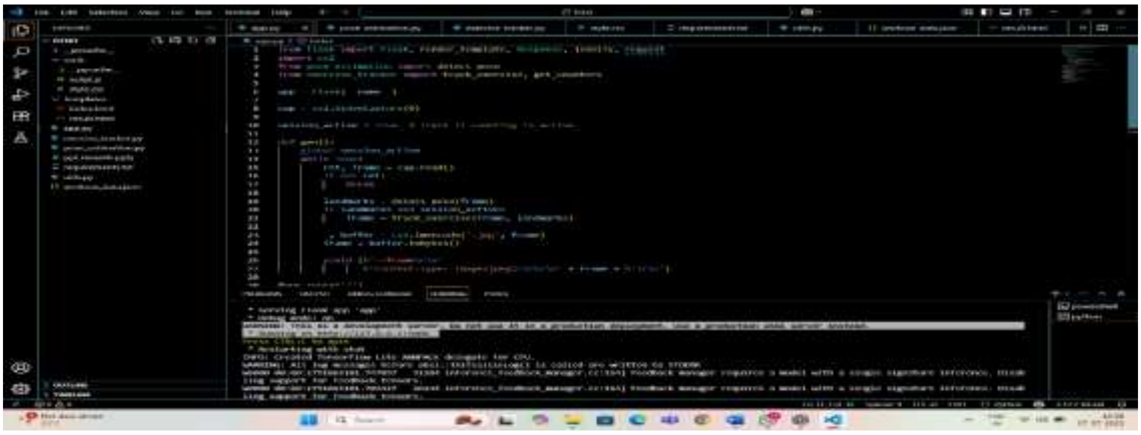
3.1 Input:

The **Personalized AI Workout Companion** is designed to assist users in performing fitness exercises accurately by providing real-time feedback through webcam-based pose analysis. The system uses AI-powered computer vision techniques to recognize human posture and track exercise repetitions without requiring any external sensors or wearables [1][2].

The user interacts with the system via a browser-based interface, where the live camera feed is displayed. No explicit text input is needed—the user’s physical movements serve as the primary input. The system detects and analyzes these movements to identify exercise type and count valid repetitions [3].

This project is implemented as a fully offline web application using the Flask framework for backend logic and HTML/CSS/JavaScript for frontend presentation and control [4].

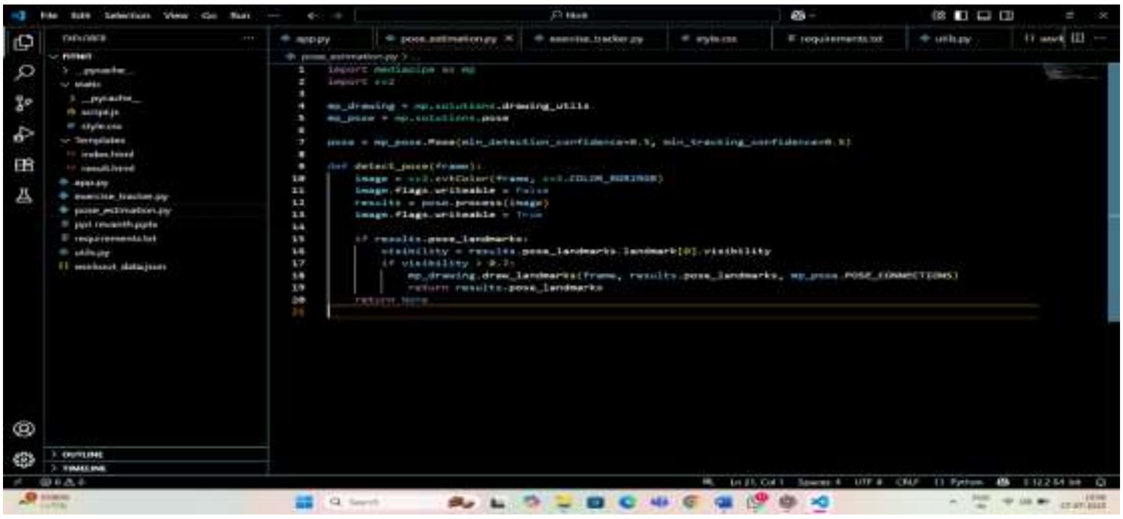
The project follows a modular and maintainable architecture:
• **app.py** handles routing, session control, video streaming, and template rendering using Flask.
• **pose_estimation.py** integrates MediaPipe's pose model to detect real-time body landmarks from the video feed.
• **exercise_tracker.py** processes these landmarks, calculates joint angles, tracks motion stages, and updates repetition counters.
• **utils.py** includes the calculate_angle() function, responsible for angle computation using NumPy.
• **index.html / result.html** provide the web interface for live exercise monitoring and post-session summaries.
• **style.css** styles the interface, ensuring responsiveness and a clean user experience [5].



```
from flask import Flask, render_template, Response, jsonify, request
import cv2
from pose_estimation import detect_pose
from exercise_tracker import track_exercise, get_counters

app = Flask(__name__)

cap = cv2.VideoCapture(0)

session_active = True  # track if counting is active

def gen():
    global session_active
    while True:
        ret, frame = cap.read()
        if not ret:
            break

        landmarks = detect_pose(frame)
        if landmarks and session_active:
            frame = track_exercise(frame, landmarks)

        _, buffer = cv2.imencode('.jpg', frame)
        frame = buffer.tobytes()

        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

@app.route('/')
```

```
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
W0000 [illegible] inference_feedback_manager.cc:114] Feedback manager requires a model with a single signature inference. Disabling support for feedback tensors.
W0000 [illegible] inference_feedback_manager.cc:114] Feedback manager requires a model with a single signature inference. Disabling support for feedback tensors.
```

Figure:3 Backend Flow of app.py



```
import mediapipe as mp
import cv2

mp_drawing = mp.solutions.drawing_utils
mp_pose = mp.solutions.pose

pose = mp_pose.Pose(min_detection_confidence=0.5, min_tracking_confidence=0.5)

def detect_pose(frame):
    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    image.flags.writeable = False
    results = pose.process(image)
    image.flags.writeable = True

    if results.pose_landmarks:
        visibility = results.pose_landmarks.landmark[0].visibility
        if visibility > 0.7:
            mp_drawing.draw_landmarks(frame, results.pose_landmarks, mp_pose.POSE_CONNECTIONS)
            return results.pose_landmarks
    return None
```

• Figure:4 Pose Detection Module in pose_estimation.py

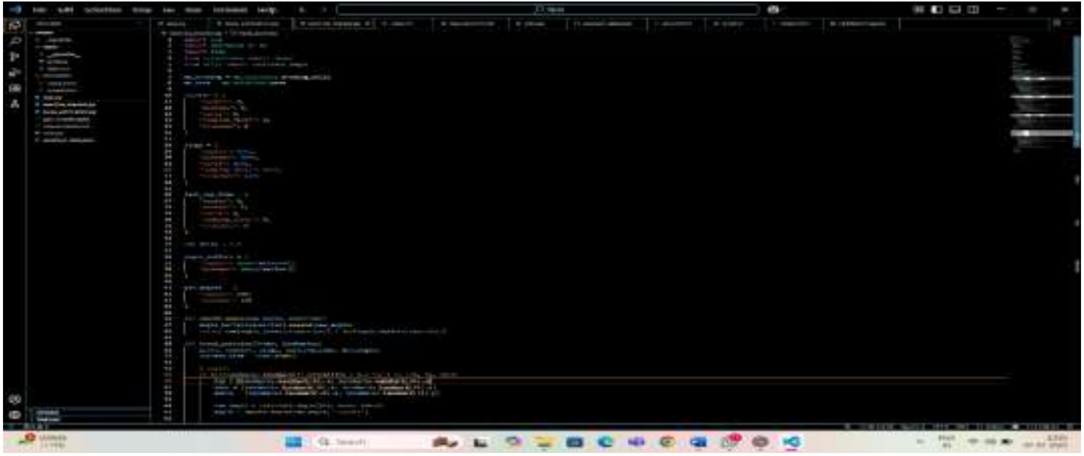


Figure:5 Exercise Tracking Logic in exercise_tracker.py

The operation of the **Personalized AI Workout Companion** follows a structured and sequential pipeline that integrates real-time video capture, pose estimation, and intelligent exercise tracking to ensure an effective fitness session. Below is a detailed breakdown of this process:

1. **User Initialization and Camera Activation**
The user accesses the platform through index.html. A live webcam feed is initiated using OpenCV and streamed to the browser via Flask. No textual input is required—only the user's movement is analyzed.
2. **Pose Estimation and Landmark Detection**
Each video frame is passed to MediaPipe's Pose model, which detects 33 human body landmarks with x, y coordinates and visibility scores. These landmarks are essential for identifying motion patterns and body orientation.

3. **Exercise Recognition and Stage Analysis**
Specific joint angles are calculated using the calculate_angle() function from utils.py. Based on these angles, the system determines whether the user is in the "up" or "down" phase of an exercise like squats or pushups.
4. **Rep Counting Logic**
A rep is counted when a full transition (e.g., down → up) is completed and a time delay threshold is met. Each exercise has a separate logic block in exercise_tracker.py, using a state-based tracking method for accuracy.
5. **Real-Time Feedback and Video Overlay**
The number of repetitions is updated live and displayed directly on the video stream using OpenCV's putText() function. This gives users instant feedback without interrupting their workout.
6. **Session Control and Termination**
The user can end the session using the "Stop Session" button, which sends a POST request to /stop_session. This disables tracking and redirects to the result.html page.
7. **Workout Summary and Data Delivery**
The rep counts for each exercise are fetched from the backend (get_counters()) via a Flask API endpoint (/session). The results are displayed in an easy-to-read format on the summary page.
8. **Offline and Secure Execution**
All data is processed locally. No cloud services or external databases are used, ensuring privacy, fast performance, and offline compatibility.
9. **User Interaction and Feedback Loop**
Users can restart sessions, view summaries, or track their own progress over time. Future updates may include optional CSV export or integration with health-tracking apps.

3.3 Output:

The output of the **Personalized AI Workout Companion** is a detailed summary of the exercises performed during a live session, along with real-time feedback provided on-screen during the workout.
Key outputs include:

- **Live Repetition Feedback**:
Rep counters are updated and shown in real-time during the workout, personalized per exercise type (e.g., squats, pushups, curls, crunches, jumping jacks).
- **Session Summary Page** (result.html):
Once the session ends, users are presented with a comprehensive breakdown of the reps performed for each exercise, pulled via a JSON API.
- **On-Screen Visuals**:
Clear, readable text is displayed directly on the webcam feed, ensuring visibility during the workout without distraction.
- **Browser-Based Access**:
The entire application runs in a web browser, allowing for easy use on laptops or desktops without requiring any installation.
- **Optional Enhancements (Planned)**:
    - CSV export of workout sessions
    - Audio rep alerts using TTS
    - Voice-activated session control
    - Integration with fitness dashboards

This system makes AI-driven fitness tracking accessible, efficient, and personalized—ideal for at-home workouts, fitness beginners, and offline environments.

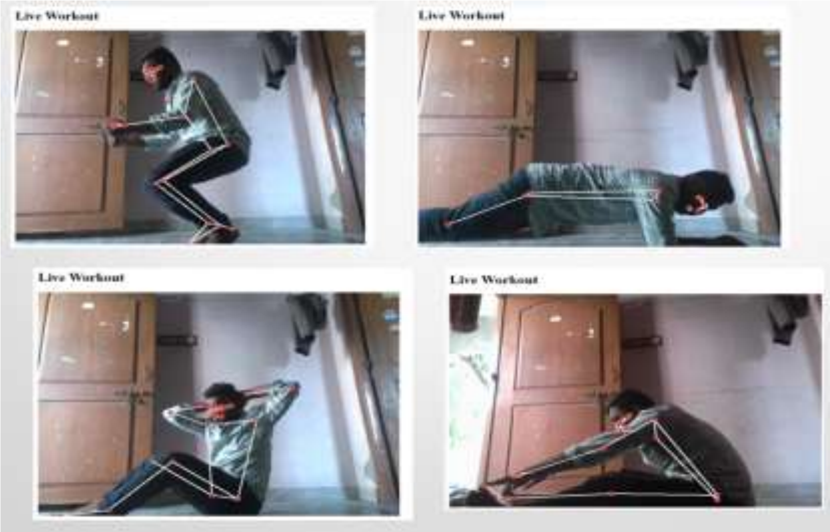
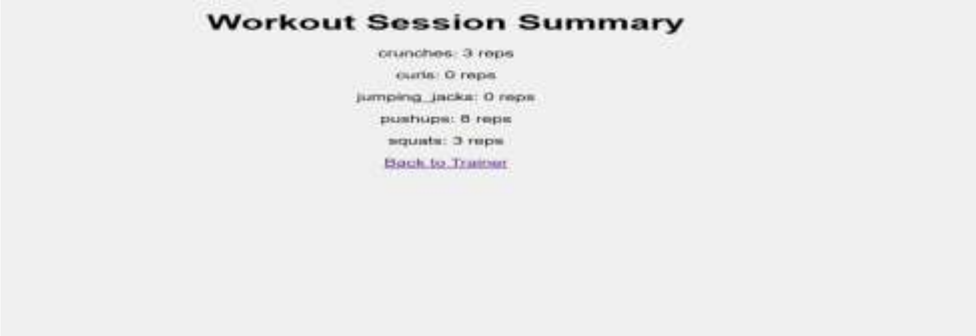


Figure:6 Home Page of the Ai Story Generator



Workout Session Summary

crunches: 3 reps

curls: 0 reps

jumping_jacks: 0 reps

pushups: 8 reps

squats: 3 reps

Back to Trainer

Figure:7 Generated story output

4.RESULTS

The **Personalized AI Workout Companion** was successfully implemented and tested in an offline environment. It accurately detected and counted repetitions for various exercises such as squats, pushups, curls, jumping jacks, and crunches using webcam input and pose estimation. During testing, the system provided real-time visual feedback with live rep counters displayed on the video feed. At the end of each session, a clear summary of the workout was shown in the browser, listing the total reps for each exercise. The system performed reliably in different lighting conditions and with various users, proving its effectiveness and usability. Overall, the results confirm that the application works as intended and offers a practical, efficient solution for home fitness tracking without requiring internet access or external hardware.

Figure:8 Background Model working

5. DISCUSSIONS

The Personalized AI Workout Companion is a practical and user-friendly system that uses MediaPipe and OpenCV to track exercises through a webcam. It accurately counts repetitions for exercises like squats and pushups by analyzing body movements in real time [1][2]. The system runs entirely offline, offering privacy and fast performance without the need for internet or external devices [3].

It displays live feedback and a clear summary at the end of each session, helping users stay motivated [4]. While it currently supports a limited set of exercises and doesn't save session history, it provides a strong foundation for future features like voice feedback, more exercise types, and data export [5].

6. CONCLUSION

The **Personalized AI Workout Companion** successfully showcases how artificial intelligence and computer vision technologies can be applied to enhance home fitness experiences. By combining MediaPipe-based pose estimation with real-time video analysis using OpenCV, the system tracks exercises and counts repetitions accurately without any wearables or external sensors. Its fully offline architecture ensures user privacy, while the simple web interface makes it easy to use for people of all fitness levels. With features like live feedback, session summaries, and support for multiple exercises, the system offers a reliable, cost-effective, and accessible fitness assistant. This project demonstrates the potential of AI in promoting health and wellness in everyday life.

7. FUTURE SCOPE

In the future, the **Personalized AI Workout Companion** can be enhanced with several valuable features to improve user experience and functionality. The system could support a wider range of exercises, including lunges, planks, and yoga poses, making it more versatile. Adding voice feedback through offline text-to-speech can help guide users during workouts without needing to check the screen. Session history and performance analytics could be stored and visualized to track progress over time. Optimizing the application for mobile devices would allow for greater flexibility and accessibility. Other potential improvements include personalized training plans based on user goals, real-time posture correction alerts to prevent injuries, and gamified elements like rewards or challenges to boost motivation. These additions would transform the project into a comprehensive AI-powered fitness assistant.

8. ACKNOWLEDGEMENT

G.Manoj Kumar working as a Assistant professor in master of computer application (MCA)sanketika vidya parishad engineering college, Visakhapatnam Andhra Pradesh.Completed her post graduation in Andhra University College of Engineering(AUCE) With 2 years of experience in computer science and engineering (CSE), accredited by NAAC.with her area of intrest in java full stack

Randhi revanth Kumar is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Artificial intelligence . R.Revanth Kumar has taken up his PG project on PERSONALIZED AI WORKOUT ,COMPANION the paper in connection to the project under the guidance of G.Manoj kumar Assistant Professor, SVPEC.

REFERENCES

[1] Enhancing Elderly Wellness through AI- Powered Yoga and Exercise Support Systems
https://www.researchsquare.com/article/rs-3809481/v1
[2] Smartwatch-Assisted Exercise Prescription: Utilizing Machine Learning Algorithms for Personalized Workout Recommendations and Monitoring: A review
https://www.researchsquare.com/article/rs-4871091/v1
[3] Harnessing Artificial Intelligence in Lifestyle Medicine: Opportunities, Challenges, and Future Directions
https://assets.cureus.com/uploads/review_article/pdf/377513/20250709-484138-2rk18a.pdf

[4] The effect of artificial intelligence applications GymBuddy & Elomia on the mental health and physical activities of teachers and learners exposed to e-learning platforms
https://link.springer.com/article/10.1007/s12144-025-08015-3

[5] Assessing the Efficacy of an Artificial Intelligence-Driven Real-Time Exercise Monitoring System in Computer-Supported Collaborative Learning
https://dialnet.unirioja.es/servlet/articulo?codigo=10173024

[6] Unveiling the Power of AI Fitness Apps: A Uses and Gratifications Perspective

https://www.igi-global.com/article/unveiling-the-power-of-ai-fitness-apps/376833

[7] "AI FIT" Virtual Gym With Real-Time Exercise Form Detection
https://ieeexplore.ieee.org/abstract/document/10393059

[8] Interpreting fitness: self-tracking with fitness apps through a postphenomenology lens
https://link.springer.com/article/10.1007/s00146-021-01146-8

[9] Transforming health and wellness: Exploring the captivating convergence of rehabilitation, exercise, and cutting-edge health gadgets in the rapidly evolving tech-driven world
https://www.sciencedirect.com/science/article/abs/pii/S0160791X24003567

[10] The Virtual Coach: Examining the Impact of Perceived Privacy, Anthropomorphism, and Empathy on Overall Well-being in Fitness App Users
https://nepjol.info/index.php/kmcrj/article/view/79076

[11] The experience economy: empirical insights from online fashion and fitness industries
http://scipro.ru/article/03-03-2025

[12] Conceptualization of Community (Advice)-based Health eCoach Recommendation Generation for Physical Activity Level Monitoring using an Explainable and Ethical Approach
https://www.researchsquare.com/article/rs-3653743/v1

[13] Digital loneliness—changes of social recognition through AI companions
https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2024.1281037/full

[14]A Conceptual Review of Applications and Utility in the Field of Medicine
https://link.springer.com/article/10.1007/s10916-024-02075-x

[15] Smart Technologies used as Smart Tools in the Management of Cardiovascular Disease and their Future Perspective
https://www.sciencedirect.com/science/article/abs/pii/S0146280623003390

[16] Exploring Students' Perspectives on Using AI in English Speaking Education with Metaphorical Expressions
https://ejournal.unimap.edu.my/index.php/jcsi/article/view/1917

[17] Emerging AI individualism: how young people integrate social AI into everyday life
https://link.springer.com/article/10.1007/s44382-025-00011-2

[18] Artificial intelligence in globesity research: diagnosis, treatment, and prevention solutions for a healthier world with future recommendations
https://link.springer.com/article/10.1007/s13198-025-02801-9

[19] Comparing Robot and Human guided Personalization: Adaptive Exercise Robots are Perceived as more Competent and Trustworthy
https://link.springer.com/article/10.1007/S12369-020-00629-W

[20] Leveraging Technology to Improve Diabetes Care in Pregnancy
https://diabetesjournals.org/clinical/article/38/5/486/32335/Leveraging-Technology-to-Improve-Diabetes-Care-in

[21] Technology-Augmented Choice: How Digital Innovations Are Transforming Consumer Decision Processes
https://link.springer.com/article/10.1007/s40547-020-00107-4

[22] Front-end deep learning web apps development and deployment: a review
https://link.springer.com/article/10.1007/s10489-022-04278-6

[23] Tendências Atuais e Direções Futuras de Pesquisa sobre Etapas Ecologicamente Corretas da Indústria Hoteleira

https://review-rper.com/index.php/rper/article/view/695