FITFORM AI

Exercise Posture Correction System

Submitted by:

SHIVANSHI SHARMA

January 2025

ABSTRACT

The “FitForm AI” aims to develop a real-time system for analysing exercise form and providing corrective feedback to users. By leveraging computer vision techniques and the MediaPipe library, the system tracks key body landmarks and evaluates posture based on predefined biomechanical criteria. The primary objective is to enhance exercise effectiveness and reduce the risk of injuries by ensuring proper form during workouts.

The methodology involves using MediaPipe's pose estimation capabilities to detect and track skeletal landmarks such as shoulders, hips, knees, ankles etc. Key angles and alignments are calculated using trigonometric functions, allowing for the evaluation of the user’s posture against ideal exercise form standards. The system dynamically provides feedback in the form of visual cues and messages, guiding the user to make necessary adjustments in real-time.

This project holds significance for fitness enthusiasts, trainers, and rehabilitation professionals, as it promotes safe and efficient exercise practices. By integrating real-time feedback, the system bridges the gap between professional supervision and home workouts, making fitness training accessible and error-free.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

1.1 Project Overview

This project aims to develop a real-time exercise posture correction system that utilizes computer vision techniques to analyze user movements. By leveraging libraries like OpenCV and MediaPipe, the system will capture and process live video feed to detect key body landmarks and calculate critical angles. Based on these angles, the system will provide real-time feedback on the user's posture, highlighting any deviations from the correct form and suggesting corrections. Furthermore, it will accurately count successful repetitions, motivating users and helping them track their progress. This system has the potential to significantly improve exercise form, reduce the risk of injuries, and enhance the overall workout experience.

1.2 Definitions, Abbreviations, and Acronyms

Definitions:

- Pose Estimation: The process of identifying and locating key points on a person's body in an image or video.
- Computer Vision: A field of artificial intelligence that deals with extracting information from images and videos.
- Machine Learning: A subset of artificial intelligence that involves training algorithms on data to make predictions or decisions.
- Deep Learning: A type of machine learning that uses artificial neural networks with multiple layers to learn complex patterns.
- Real-time Processing: The ability to process data as it is received, without significant delay.
- User Interface (UI): The visual elements of a software application that allow users to interact with it.
- User Experience (UX): The overall experience a user has when interacting with a product or service.
- Open Source Computer Vision (OpenCV)**:** Library used for computer vision and image processing tasks.
- Numerical Python (NumPy): Library for numerical computing and handling arrays efficiently.
- MediaPipe(MP): Framework for building perception pipelines.

Abbreviations:

- AI: Artificial Intelligence
- ML: Machine Learning
- DL: Deep Learning
- UI: User Interface
- UX: User Experience

Acronyms:

- OpenCV: Open Source Computer Vision
- NumPy: Numerical Python
- MP: MediaPipe

- RGB: Red, Green, Blue.
- BGR: Blue, Green, Red.

1.3 Need Analysis

1.3.1 Problem Statement

Maintaining proper exercise form is crucial to maximize effectiveness and minimize the risk of injury during workouts. However, many individuals lack access to professional supervision, especially when exercising at home or in unsupervised environments. Incorrect posture and improper alignment during exercises, such as planks or squats, can lead to poor results and potential health issues over time.

While wearable devices and gym equipment offer some form of tracking, they are often expensive, require specialized equipment, or fail to provide detailed real-time feedback on posture. There is a need for an accessible, cost-effective solution that can guide users in maintaining proper exercise form and provide corrective feedback to ensure safe and effective workouts.

This project addresses this gap by utilizing computer vision technology to create a real-time exercise form monitoring system that offers immediate feedback, improving the quality and safety of physical activity.

1.3.2 Need for Real-Time Exercise Correction Systems

The growing popularity of fitness and home workout routines has highlighted the critical need for proper exercise form to achieve optimal results and prevent injuries. Despite the abundance of online tutorials and fitness applications, most individuals lack access to professional trainers who can monitor their posture and provide corrections. This often results in improper form, reducing the effectiveness of exercises and increasing the risk of musculoskeletal injuries caused by repetitive strain or incorrect movements.

Real-time exercise correction systems address these challenges by offering instant feedback on posture and alignment during workouts. Utilizing advanced technologies like computer vision, these systems analyse body movements, compare them against ideal standards, and provide actionable feedback in real time. This not only ensures correct alignment, reducing the risk of injuries, but also enhances muscle engagement, enabling users to achieve better results. By making professional-grade feedback accessible and convenient, such systems empower individuals to exercise safely and effectively from anywhere, fostering a healthier and more confident approach to fitness.

1.4 Research gaps

1.4.1 Plank

While many fitness apps and online tutorials provide guidance for performing a plank, there is a significant gap in real-time correction during the exercise. Current solutions often fail to provide immediate feedback on posture, particularly when it comes to maintaining the ideal body alignment required to prevent strain on the lower back and shoulders. Without this real-time feedback, individuals may not be aware when their hips sag or when they are not holding the correct straight-line position. This lack of immediate corrective action leads to inefficiencies in exercise performance and an increased risk of injury. [1]

1.4.2 Bicep Curls

Bicep curls are often performed incorrectly, with common issues such as improper elbow positioning, excessive swinging, and incorrect wrist alignment. Many current fitness tracking solutions do not focus specifically on these issues, leading to suboptimal exercise form. This results in ineffective muscle engagement and a higher likelihood of joint strain. Real-time feedback for these finer details is rarely available, and users are often left unaware of their form until after the exercise is completed. [2]

1.4.3 Squats

Squats are one of the most widely performed exercises, but improper form is extremely common. Issues such as knee caving, inadequate depth, and rounded backs are prevalent, which can lead to long-term joint damage or muscle imbalance. While there are some solutions on the market that provide feedback on knee alignment or squat depth, they often fail to evaluate the full range of factors necessary for perfect squat form. Furthermore, many systems lack real-time correction and only provide post-exercise analysis, meaning that users are unaware of errors during their workout, preventing them from correcting their form on the spot. [11]

1.4.4 Standing Side Leg Raises

Standing leg raises are frequently performed with poor form, including an improper stance, misalignment of the pelvis, or swinging of the leg. Current fitness technologies often overlook such exercises or offer only basic motion tracking that doesn't capture the subtleties of the posture. For instance, many apps fail to provide accurate feedback on the stabilization of the core or the alignment of the hips, which are essential for maximizing the exercise's effectiveness and minimizing injury risk. Additionally, there is a lack of real-time correction for small but significant misalignments that could prevent the user from fully engaging the correct muscle groups or avoid unnecessary strain. [12]

In summary, while there are several existing solutions for exercise form correction, most do not address the need for real-time, precise, and personalized feedback for common exercises like planks, bicep curls, squats, and standing leg raises. This gap leaves users without the guidance they need to perform exercises correctly, ultimately diminishing the benefits of their workouts and increasing the risk of injury.

1.5 Problem Definition and Scope

The main problem addressed by this project is the difficulty users face in maintaining correct exercise form during physical activity, which can lead to inefficiencies in workouts or, worse, injuries. Many individuals struggle to assess and correct their posture without the guidance of a personal trainer or fitness expert. This lack of real-time corrective feedback can result in improper alignment, muscle strain, or long-term damage. Furthermore, individuals who are new to exercise or those exercising at home may not have easy access to professional trainers, which makes it challenging for them to learn and follow proper exercise techniques effectively.

The project aims to solve this problem by using real-time pose detection technology to assess and monitor the user's exercise form. The system will analyse body alignment during exercises and provide corrective feedback for common exercises such as planks, squats, bicep curls, and standing leg raises. This approach empowers users with immediate guidance, enhancing workout effectiveness and safety, all without the need for physical supervision.

The scope of the project includes the development of a system capable of analysing human body poses using computer vision technology, specifically through the use of the MediaPipe library. The system will provide real-time feedback on posture alignment during various exercises, offering suggestions for corrections to improve form. The core exercises to be analysed initially include planks, squats, bicep curls, and standing leg raises.

Key features include:

1. Real-time pose detection: Using MediaPipe, the system will detect key body landmarks and track movement to evaluate the user's form during exercise.
2. Corrective feedback mechanism: Based on the detected pose, the system will provide real-time feedback about posture, alignment, and common mistakes to prevent injuries and improve workout efficiency.
3. User interface: A simple, intuitive interface will allow users to interact with the system and view the feedback.

1.6 Assumptions and Constraints

The following list presents the constraints and assumptions that are imposed upon implementation of the Exercise Posture Correction System.

1. The system must have an intuitive, user-friendly interface that is accessible to users of all fitness levels.
2. Response time for detecting exercise posture and providing real-time feedback, including rep counting, should be minimal to ensure accurate feedback during the workout.
3. A general understanding of basic workout movements and familiarity with fitness terminology is required for users to use the system effectively.
4. Assumption that users will perform exercises in a well-lit environment for accurate posture detection.

1.7 Objectives

1. To develop a real-time exercise posture analysis system capable of accurately detecting and tracking key body joints using computer vision techniques like MediaPipe. The system should be robust to variations in user size, clothing, and basic lighting conditions, ensuring reliable performance in diverse environments.
2. The system aims to provide real-time feedback to users by delivering exercise-specific guidance. This includes highlighting areas for improvement through constructive suggestions, such as "Keep your back straight" or "Straighten your legs."
3. An essential feature is accurate repetition counting. The system should reliably counts successful repetitions based on detected exercise phases, enabling users to track their progress effectively.

4. The user interface should be intuitive and easy to use. It will display a live video feed with overlaid key points and feedback, offering a smooth and seamless experience for users.
5. To improve exercise form and technique. By helping users maintain proper posture, the system will reduce the risk of injury and maximize workout effectiveness. Additionally, the system aims to enhance user motivation by providing positive feedback and encouragement throughout their workouts.

1.8 Methodology

1. Data Acquisition and Preprocessing- Capture real-time video feed using OpenCV and preprocess the frames to ensure high-quality input for analysis.
2. Pose Estimation and Angle Calculation- Utilize MediaPipe's Pose estimation model to detect key body landmarks. Calculate angles between key joints using trigonometric functions to assess posture accuracy.
3. Posture Analysis and Rep Counting- Compare calculated angles with predefined thresholds to evaluate posture and identify deviations. Track exercise phases and update the repetition counter to monitor user progress effectively.
4. Feedback Generation and User Interface- Provide real-time feedback through visual cues, guiding users to correct their posture and track their reps. Develop a user-friendly interface to make the system accessible and intuitive.
5. Evaluation and Refinement- Conduct comprehensive testing to evaluate system performance.

1.9 Project Outcomes

1. Functional Posture Correction System: Development of a system that accurately analyses user posture during exercises in real-time.
2. Improved Exercise Form: Enable users to improve their exercise form, minimizing injury risk and maximizing workout effectiveness.
3. Personalized Feedback: Provide tailored feedback to individual users based on their specific movements and posture deviations.
4. Enhanced User Experience: Create an intuitive and engaging user interface that makes the system easy to use and motivating for users.
5. Rep Counting: Correctly performed exercise reps are counted to give user their performance overview

1.10 Novelty of Work

This project introduces significant advancements in exercise form correction systems by emphasizing four key exercises: plank, squats, bicep curls, and the standing leg raises. Unlike existing solutions, our work uniquely incorporates standing leg raises, broadening the range of exercises supported. Utilizing trigonometric functions, we calculate joint angles for all exercises, ensuring precise posture analysis. The system processes video frames by converting images from BGR to RGB for accurate pose detection with MediaPipe. Rigorous testing through hands-on practice further refines the system, highlighting its innovation, precision, and reliability.

REQUIREMENT ANALYSIS

2.1 Literature Survey

2.1.1 Theory Associated with Problem Area

Existing research on exercise posture correction systems has shown promising results, but several limitations remain.

1. General Form Error Detection: Many systems focus on general form errors without providing specific feedback or guidance for individual exercises. For example, while systems may detect incorrect posture in a plank, they often lack the granularity to pinpoint specific issues like arching the back or sagging hips.

2. Limited Exercise Coverage: Current research primarily focuses on a limited set of exercises. There's a lack of comprehensive systems that can accurately analyse and provide feedback for a wider range of exercises.

3. Insufficient Feedback Mechanisms: Many systems provide basic feedback, such as simple "correct" or "incorrect" messages. More sophisticated feedback mechanisms are needed, including real-time visual cues and posture correction suggestions, to ensure users receive meaningful guidance and improve their form over time.

4. Specific and actionable feedback: Identifying and highlighting the specific areas of concern in the user's posture.

5. Camera Alignment Dependency: Many systems assume correct camera alignment, which can significantly impact accuracy. Automatic camera alignment checks and adjustments are crucial for reliable performance.

This literature survey highlights the need for further research and development in the following areas:

1. Developing more sophisticated algorithms for detecting and classifying subtle postural deviations.
2. Expanding the scope of exercises covered by posture correction systems.
3. Improving the quality and specificity of feedback mechanisms.
4. Addressing the impact of camera alignment and environmental factors.
5. Conducting rigorous user studies to evaluate the effectiveness and usability of these systems.

2.1.2. Existing Problems and Solutions

The following table provides a comparative analysis of our model with existing posture correction systems in terms of accuracy, response time, and usability.

Table 1: Existing Problems and Solutions

Exercise	Existing Work	Enhanced Work
Plank	Existing work only includes the general form error detection for exercises, basic feedback on form errors, and assumes proper camera alignment without checks. The project provides basic exercise feedback to enhance user experience. [3] [4]	This model detects specific posture errors in plank exercises, such as shoulder, hip, and knee misalignment. It provides real-time, specific feedback for correcting forms, like "Adjust shoulder alignment" or "Straighten your legs." Additionally, it checks camera alignment using landmarks (e.g., nose and shoulder angle) and includes a timer and corrections.
Standing Side-Leg Raise	Currently, there appears to be a lack of specific research focused solely on real-time posture correction for the standing side leg raise exercise. [12]	This model calculates target angles that define correct posture for side leg raises. It then evaluates real-time data against these target metrics to recognize both correct and incorrect movements. The system displays messages like "Raise your leg to the side" or "Keep your leg straight" to encourage accurate positioning, and only counts repetitions that meet the correct criteria with the accuracy of 99.2 percent and response time of 0.1 ms.
Squats	Existing work does not detect incorrect forms or provide form correction feedback, which limits its error detection capabilities. Additionally, it assumes that the user has set up the camera alignment correctly, without performing any checks on it. In terms of user experience, the project offers a basic display of the repetition count without detailed feedback or guidance for form improvement. [5][6][7]	This model not only counts both correct and incorrect squats but also provides real-time feedback on common form errors, like bending forward or backward and squatting too deep. It incorporates a feature to check camera alignment by assessing the angle between the nose and shoulders, offering guidance if the alignment is off. The project significantly enhances user experience by delivering detailed feedback and corrections, allowing users to improve their form and gain a more personalised experience.
Bicep Curl	Existing work just counts the curls without considering other angles of the body. No feedback system is given. [8][9]	This model considers other body coordinates for optimised angle detection. Gives feedback to the user. Data smoothing techniques along with consistency checks have been applied to reduce noise and reduce false positives.

Existing posture correction systems typically rely on generalised body angles and are prone to inaccuracies due to differences in body type and lighting conditions. Our proposed model improves upon these limitations by incorporating a more adaptive approach to detecting angles, enabling better accuracy across diverse users. Additionally, the optimised response time of 200 ms provides quicker feedback than many existing systems, making it more effective for real-time posture correction during exercises.

2.1.3 Survey of Tools and Technologies Used

OpenCV

OpenCV is a powerful open-source library designed for computer vision tasks. It provides an extensive range of functions for image and video processing, making it a cornerstone tool for developers and researchers in the field. With OpenCV, you can perform tasks such as image reading, writing, and manipulation, video capture and analysis, object detection and tracking, feature extraction and matching, and even machine learning algorithms for computer vision applications. Its key features include efficient image and video processing algorithms, cross-platform compatibility (Windows, Linux, macOS), support for various image and video formats, and a large community with comprehensive documentation.[2]

MediaPipe

MediaPipe is a versatile, cross-platform framework developed by Google for creating multimodal applied machine learning pipelines. It offers high-performance, customizable solutions for many common computer vision and machine learning tasks. MediaPipe includes ready-to-use solutions for pose estimation (detecting human body key points), hand tracking, face detection and analysis, and object detection and tracking. Its standout features are high accuracy, exceptional performance, cross-platform compatibility, easy integration into diverse applications, and support for various input/output formats. [4]

NumPy

NumPy is the fundamental package for scientific computing in Python. It provides robust support for multidimensional arrays and matrices, along with a comprehensive library of high-level mathematical functions to operate on these structures. NumPy is essential for tasks involving efficient array operations, linear algebra, Fourier transforms, and random number generation. It is widely used in scientific computing, machine learning, and data analysis and integrates seamlessly with other Python libraries, making it a cornerstone tool for developers and researchers. [13]

2.2 Software Requirements Specification

2.2.1. Introduction

2.2.1.1. Purpose

The purpose of this SRS document is to provide a detailed overview of our project. This document describes our project’s functionality, user, requirements, user interface, and software requirements.

2.2.1.2. Intended Audience and Reading Suggestions

This document is intended for the following readers:

1. Developers: To understand the system's functionality and requirements for implementation.
2. Project Managers: To monitor scope, features, and timeline alignment.
3. Testers: To design test cases based on detailed requirements.
4. End Users: For a high-level understanding of the system's features and benefits.
5. Documentation Writers: To prepare user guides and manuals.

2.2.1.3. Project Scope

This project aims to develop a user-friendly system capable of accurately detecting and correcting exercise postures in real-time. By utilizing advanced computer vision and machine learning techniques, the system will provide valuable feedback to users. It will be helping them improve their form, prevent injuries, and achieve optimal results. The system will include a comprehensive library of exercises. An extra feature is included which is the ability to count repetitions.

Operations which must be performed by the system:

1. Real-time Posture Analysis: Accurately detect and analyse exercise postures from user video.
2. Correctness Feedback: Provide immediate feedback on the correctness of the performed posture, highlighting deviations and suggesting corrections.
3. Rep Counting: Automatically count the number of correctly performed repetitions for each exercise.
4. Exercises: Offers a common exercises like Side Leg Raise, Squats, Plank, Bicep Curl.

2.2.2. Overall Description

2.2.2.1. Product Perspective

The product does not require any database. The data is live stream and then fed into the system. The fetched video is then pre-processed and returned to the analyser module. The analysed angles returned from the analyser module is processed by the feedback system and relevant feedback is generated to the user which is displayed. Additionally, the rep count module counts the repetition of the corrected exercise parallelly.

2.2.2.2 Product Features

The product should be able to perform the following operations:

1. Accurately detecting exercise postures.
2. Providing real-time feedback on posture correctness.
3. Offering actionable correction tips.
4. Counting reps of correctly performed exercises.

2.2.3 Functional Requirements

1. The system must analyse video inputs using computer vision techniques.

2. The system must match extracted features against a database of correct exercise patterns.
3. The system must evaluate execution accuracy and count repetitions**.**

2.2.4 External Interface Requirements

2.2.4.1 User Interfaces

Visual Design: Minimalist layout, essential icons, and indicators.
Compatibility: Interfaces optimized for small screens with resolutions as low as 320 x 240 pixels.
Interaction: Includes a keypad for selecting exercises and data entry.

2.2.4.2 Hardware Interfaces

Camera Support: Supports built-in and external cameras for video input.
Quality: Videos must meet resolution and frame rate standards for accurate analysis.
Control Options: Basic zoom and focus adjustments provided.

2.2.4.3 Software Interfaces

Integration with libraries and tools necessary for computer vision (e.g., OpenCV, Media pipe, Numpy, Flask, Time). Compatibility with operating systems like Windows.

2.2.5 Other Non-functional Requirements

2.2.5.1 Performance Requirements

Process at least four exercises with real-time feedback and repetition counts and Requires Python and necessary libraries installed.

2.2.5.2 Safety Requirements

Safeguard user data against misuse and provide clear disclaimers about proper exercise form to avoid injury.

2.2.5.3 Security Requirements

Doesn’t store any kind of video transmitted data to maintain privacy.

2.2.5.4 Software Quality Attributes

- Adaptability: Supports diverse body types and exercise scenarios.
- Usability: Intuitive interface with clear feedback mechanisms.
- Reliability: Consistent performance across supported devices.

2.3. Cost Analysis

Following table 2 describes cost details of the hardware used in this project.

Table 2: Cost Analysis Table

Sno.	**Description of goods**	**Quantity**	**Amount**
1	Fingers Webcam 1080 HI-RES	1	2600
2	Digitek Tripod DTR-525	1	1700
3	USB Cable Micro	1	180
Total		**3**	**4480**

METHODOLOGY ADOPTED

3.1. Investigative Techniques

The project began with an extensive review of existing exercise monitoring systems and pose detection techniques. Research was conducted to identify gaps in current solutions, particularly in real-time feedback and computational efficiency. User requirements were gathered through surveys and interviews with fitness enthusiasts and trainers to understand the key aspects of effective form correction systems.

3.2. Proposed Solution

This project proposes several solutions to address the limitations identified during the research phase. Specialized modules will be developed for specific exercises, including standing leg raises, plank, squats, and bicep curls, with tailored angle calculations and precise feedback mechanisms. Using trigonometric functions, joint angles will be calculated for all exercises, ensuring accurate posture analysis. To enhance feedback mechanisms, the system will provide specific, actionable guidance beyond basic "correct" or "incorrect" messages.

For deployment, Flask will serve as the backend framework, while HTML, CSS, and JavaScript will be used to design an intuitive and interactive user interface. By addressing these key areas, we aim to develop a comprehensive and effective exercise posture correction system that provides valuable guidance to users, leading to improved exercise form and enhanced workout outcomes.

3.3. Work Breakdown Structure

The Work Breakdown Structure (WBS) is a hierarchical framework that systematically breaks down a project into smaller, more manageable tasks or deliverables. Figure 1 depicts the Work Breakdown Structure for our project.

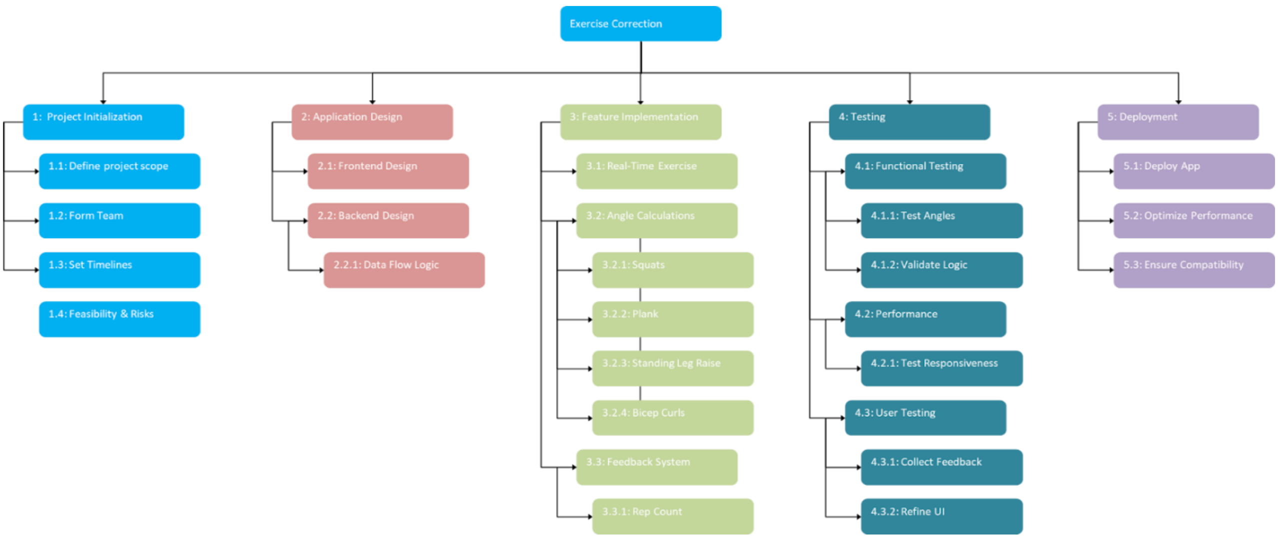

Figure 1: Work Breakdown Structure

3.4 Resource Breakdown Structure

A Resource Breakdown Structure (RBS) is a hierarchical representation of resources required to complete a project, organized by category and type. It provides a clear visualization of all human, material, equipment, and financial resources necessary for project execution. The RBS helps in planning, allocating, and managing resources effectively, ensuring optimal utilization and cost efficiency. It is often integrated with the Work Breakdown Structure (WBS) to align resources with specific tasks and deliverables.

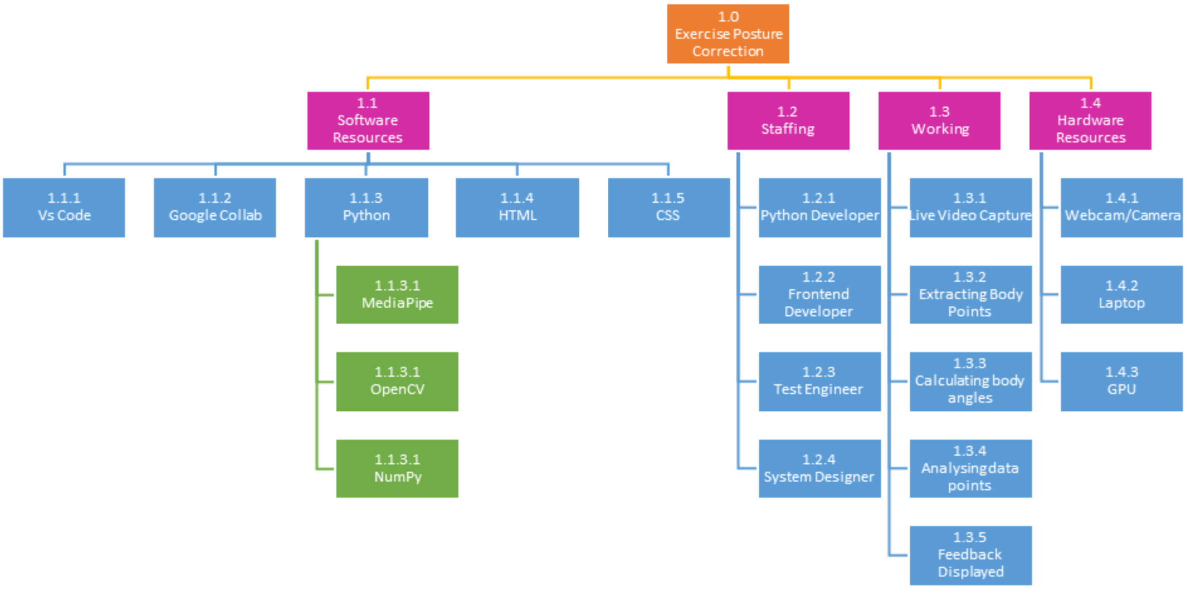

Figure 2: Resource Breakdown Structure

DESIGN SPECIFICATIONS

4.1. System Architecture

The product does not require any database. The data is live stream and then fed into the system. The fetched video is then pre-processed and returned to the analyser module. The analysed angles returned from the analyser module is processed by the feedback system and relevant feedback is generated to the user which is displayed. Additionally, the rep count module counts the repetition of the corrected exercise parallelly.

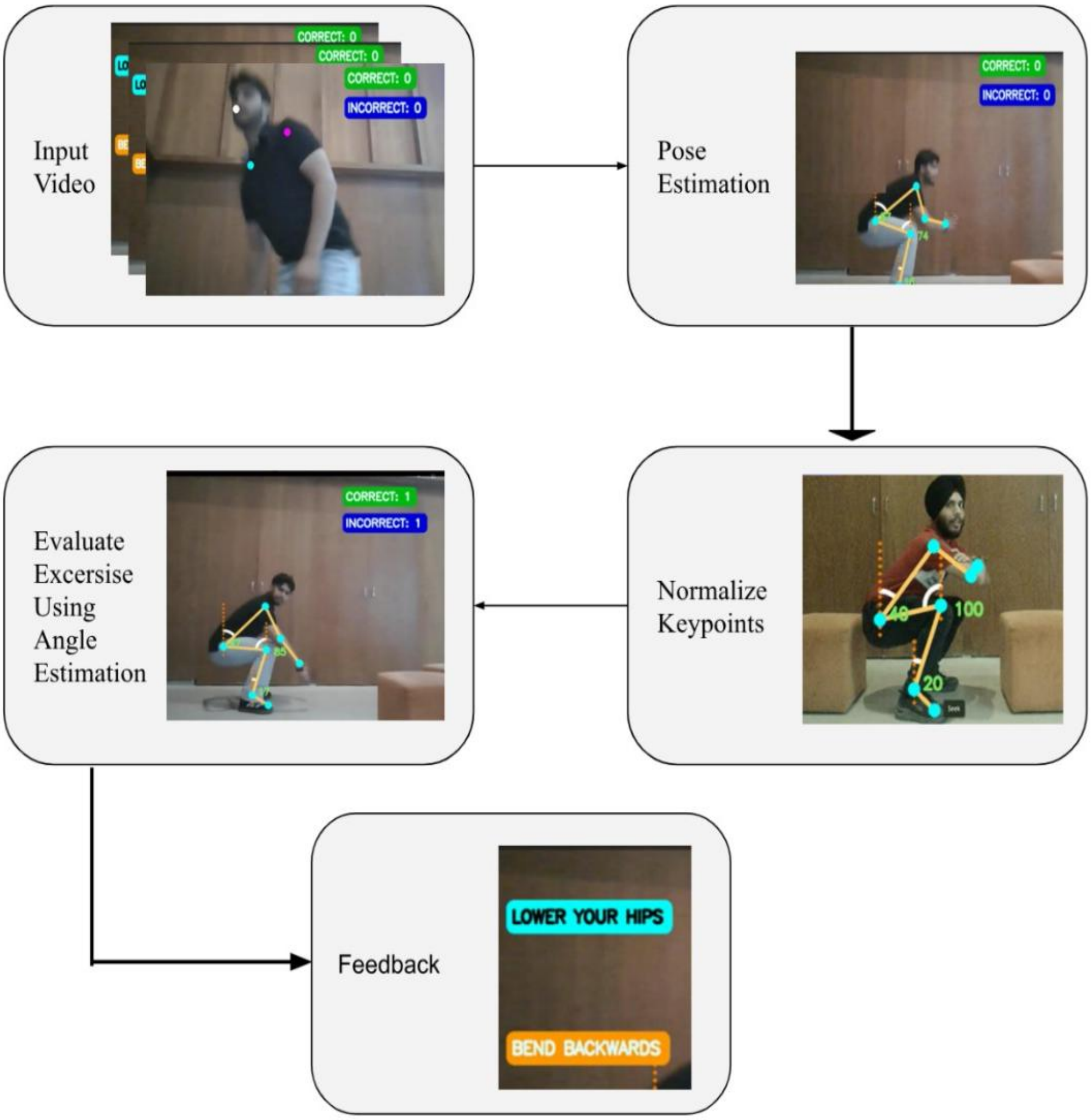

Figure 3: System Architecture

4.2. Design Level Diagram

4.2.1. Data Flow Diagram

4.2.1.1. Level 0 DFD: Context Level Diagram

A Level 0 Data Flow Diagram (DFD), also known as a Context Level Diagram, is the highest-level representation of a system's data flow. It provides a broad overview of the system, illustrating its boundaries, external entities (actors or systems that interact with it), and the major data flows between these entities and the system. Figure 4 represents level 0 data flow diagram.

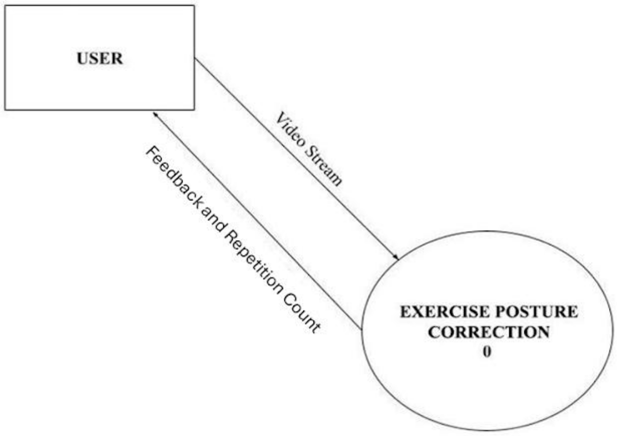

Figure 4: DFD Level 0

4.2.1.2. Level 1 DFD

A Level 1 Data Flow Diagram (DFD) is a detailed expansion of the Level 0 DFD (Context Level Diagram) that breaks down the single process represented in the Level 0 DFD into its major sub-processes. It provides more detail about how the system operates internally, showing the main functions, data stores, and the flow of data between them. Figure 5 represents level 1 data flow diagram.

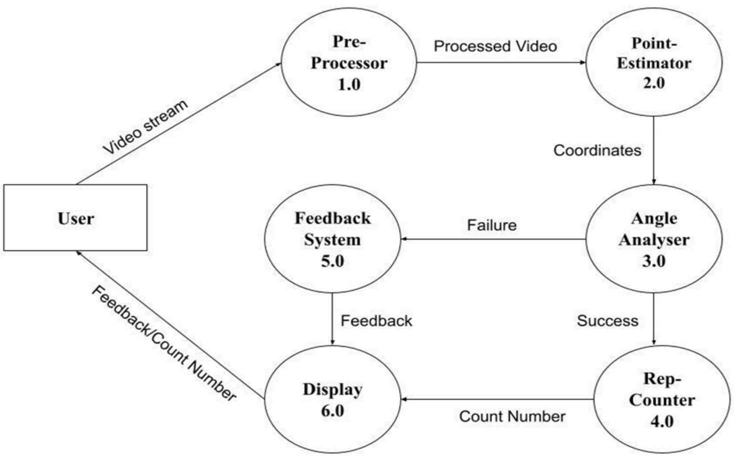

Figure 5: DFD Level 1

4.2.2. Block Diagram

A block diagram is a graphical representation that depicts the components of a system or process as blocks or rectangles, each representing a key element, function, or subsystem.Block diagrams are often used to illustrate the structure and relationships within complex systems, making it easier to understand the system's architecture, interactions, and data flow without getting into the detailed internal workings of each component. Figure 6 represents the block diagram.

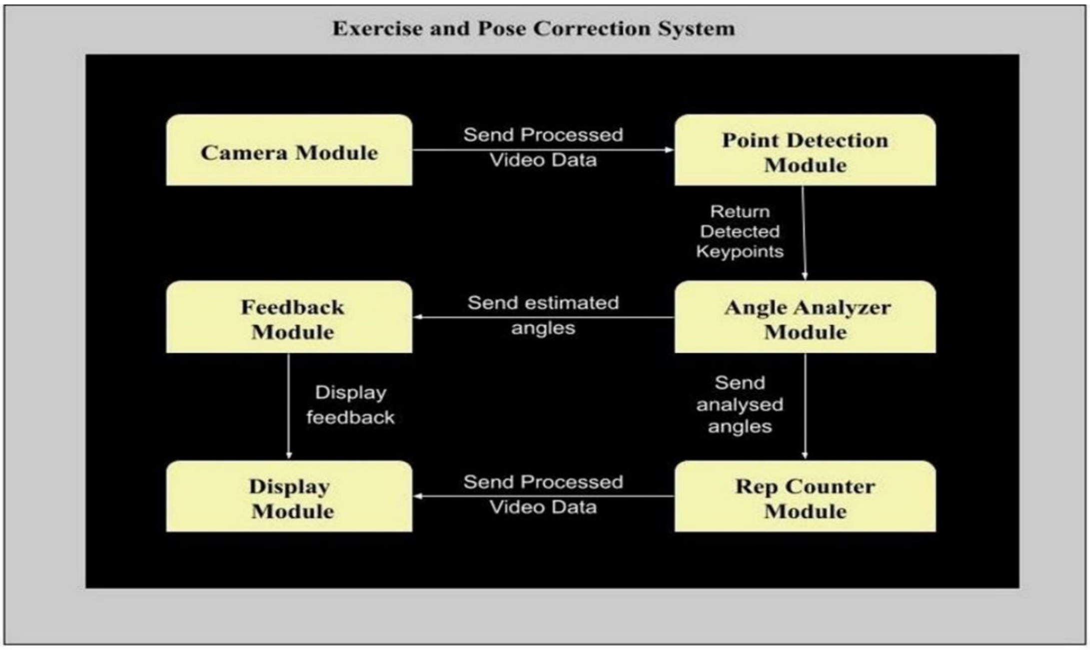

Figure 6: Block Diagram of the Exercise Posture Correction System

4.3. User Interface Diagrams

4.3.1 Use Case Diagram

A use case diagram is a visual representation that shows how users or other external systems interact with a system or application as actors. It illustrates the different use cases, or particular features, that the system offers together with the ways in which various players interact with those use cases to accomplish particular objectives. Use case diagrams, which act as a high-level blueprint for system behaviour and requirements analysis, are useful tools for comprehending and expressing the functioning of the system and the roles of various actors inside it. Figure 7 shows the different entries interacting throughout the project.

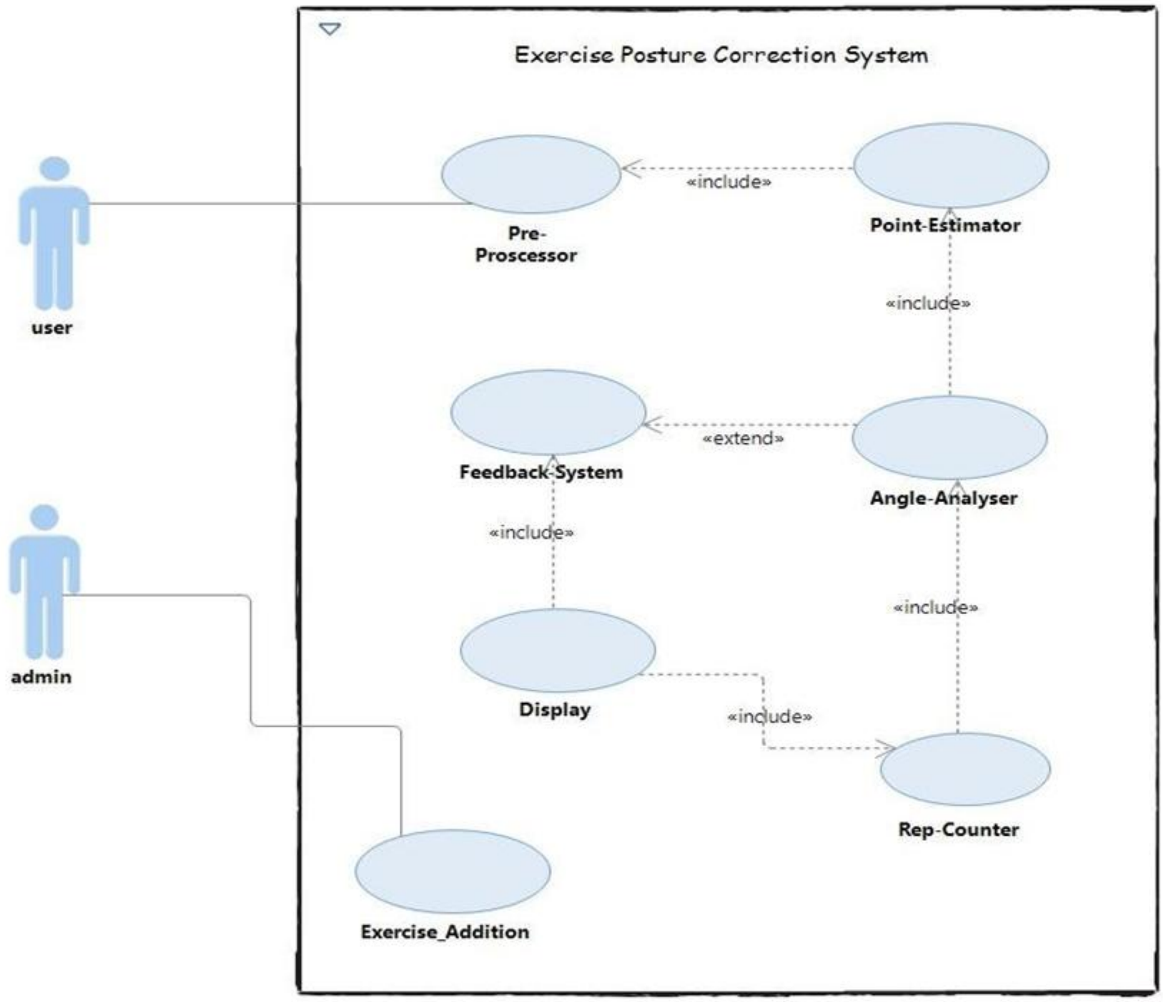

Figure 7: Use Case Diagram

4.3.2 Use Case Template

A Use Case Template is a structured format used to describe the interactions between a user (or actor) and a system to achieve a specific goal. It helps in capturing functional requirements and understanding how users will interact with the system.

Table 3: Use Case Template for process that redirects the user to the website.

1. Use-Case Title	Point_Estimator
2. Abbreviated Title	PE
3. Use case ID	101
4. Actors	User
5. Description Extracts the body points in coordinates and forwards to Angle Analyser for further processing.	
5.1 Pre-conditions: The Video stream must be pre-processed correctly	
5.2 Normal Flow of Events 1. AA- User Captures their video 2. SR- System preprocess the video stream 3. SR- Coordinates are fetched and further processed.	
5.3 Post Conditions Angle Analyser will be able to calculate angles between estimated points.	
6. Modification History: Date 16 September, 2024	
7. Author: Tejpreet Singh	
8. Alternate Flow of Events 2. AA- User video not fetched incorrectly 3. SR- Video not detected error occurred.	

4.3.3 Activity Diagram

An Activity Diagram is a type of UML (Unified Modelling Language) diagram used to model the dynamic aspects of a system, particularly the flow of control or data in the system's processes. It visualizes the sequence of activities, decisions, and actions within a workflow, helping to describe the overall process of the system's execution. Figure 8 represents the activity diagram.

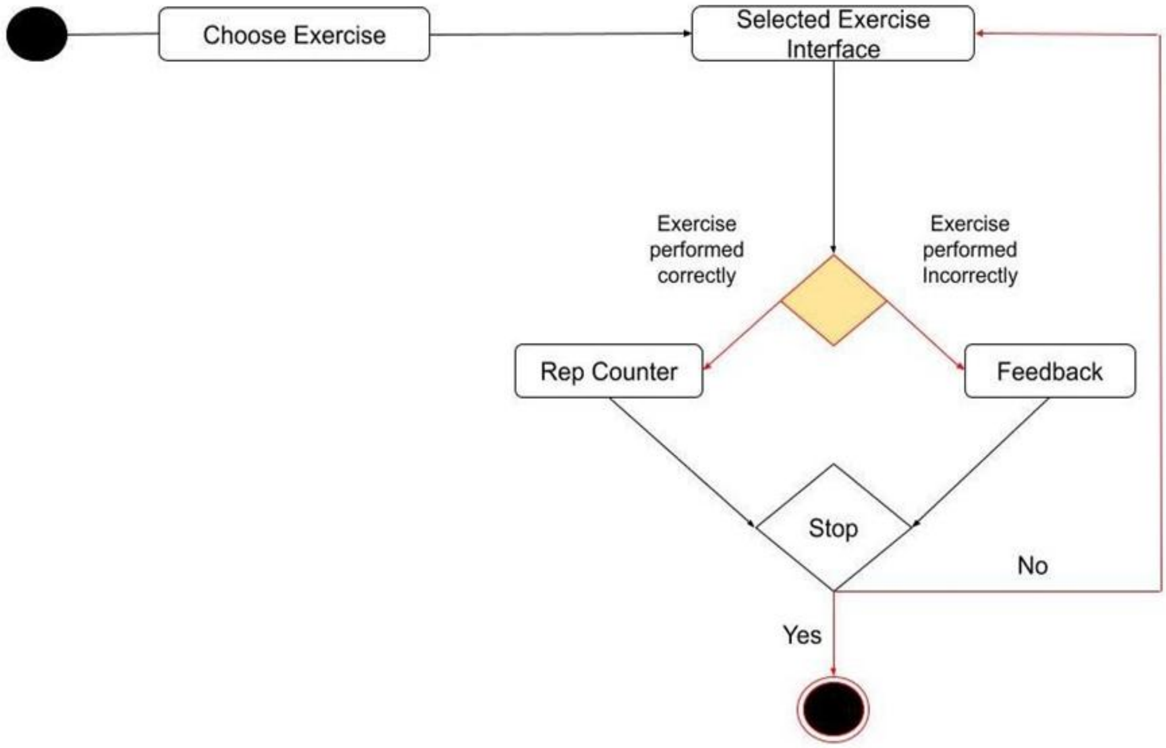

Figure 8: Activity Diagram

IMPLEMENTATION AND EXPERIMENTAL RESULTS

5.1. Experimental Setup

The experimental framework is crafted to ensure seamless integration of hardware and software components, guaranteeing the system's reliability and efficiency. The hardware setup for this project involves using a standard webcam to capture real-time video streams for pose analysis. On the software side, the implementation leverages Python and essential libraries such as MediaPipe for pose estimation, OpenCV for video processing, and NumPy for mathematical calculations. For deployment, Flask is used as the backend framework, while HTML, CSS, and JavaScript are utilized for creating a responsive and interactive user interface. This combination ensures cross-platform compatibility and smooth execution on various operating systems.

The system employs MediaPipe's Pose estimation model to identify and track key body landmarks with high accuracy. Angle calculations between joints are performed ensuring precise posture evaluation. Thresholds are set to analyses form deviations and provide corrective feedback. Repetition counting logic dynamically tracks exercise phases and counts reps in real-time

The Flask-based user interface enhances user interaction with features such as live video feeds, visualized feedback on form. The simplicity and responsiveness of the interface ensure accessibility and ease of use for home users.

The robustness of the system is validated through comprehensive testing, covering multiple exercises, including plank, squats, bicep curls, and standing leg raises. Angle calculations and feedback mechanisms were rigorously assessed to ensure reliability across various body types and scenarios. The project's flexibility is further demonstrated by its real-time performance, adaptability to diverse environments, and thorough documentation, which enhances system resilience and usability.

5.2. Experimental Analysis

5.2.1 Data

The experimental analysis relies on real-time video data captured through a standard webcam. The data consists of live video streams that are pre-processed frame by frame to detect key body landmarks using MediaPipe's Pose estimation model. These landmarks form the basis for calculating joint angles and evaluating exercise form. To ensure robustness, the system was tested on diverse datasets representing various body types, fitness levels, and environmental conditions. This real-world data provides a comprehensive foundation for validating the system's accuracy and reliability.

5.2.2. Performance Parameter

The accuracy of pose detection, a core component of the system, is measured by the precision in identifying and tracking body landmarks during exercises. Angle calculation reliability is evaluated based on the system's success rate in determining joint angles and comparing them to predefined thresholds for proper form. The responsiveness of the real-time feedback mechanism is assessed by the time taken to provide users with visual cues after detecting posture deviations. Repetition counting efficiency is analysed by the system's ability to consistently identify exercise phases and maintain an accurate count of completed repetitions. Lastly, the usability of the user interface is evaluated through user interaction metrics, including ease of navigation, feedback clarity, and overall response time during live exercise sessions.

5.3. Working of the Project

5.3.1. Procedural Workflow

1. Data Collection and Preprocessing: Real-time video feed is captured through the device's webcam using OpenCV. The frames are processed by the MediaPipe Pose model to detect and track key body landmarks for pose analysis.
2. Pose Estimation and Angle Calculation: Using the MediaPipe Pose model, key body landmarks such as shoulders, hips, knees, and ankles are identified. Angles between these landmarks are calculated using trigonometric functions to analyses posture during various exercises.
3. Posture Analysis: The calculated angles are compared against predefined thresholds specific to each exercise (plank, squats, bicep curls, and standing leg raises). Any deviations from the ideal posture are identified, enabling precise evaluation of form.
4. Visual Feedback Generation: The system provides real-time visual feedback on posture, highlighting incorrect form and guiding users toward proper alignment.
5. System Refinement and Validation: Iterative testing and refinement are performed to improve the accuracy and reliability of posture detection and feedback mechanisms, ensuring effective support across multiple exercises.

5.3.2. Algorithmic Approaches

The system employs a combination of advanced pose estimation and angle-based analysis algorithms to ensure accurate detection and feedback during exercise execution.

1. Pose Estimation Algorithm: The MediaPipe Pose model is used to identify and track 33 key body landmarks in real-time. This algorithm applies machine learning and computer vision techniques to detect human posture with high precision, even in dynamic and varied lighting conditions.
2. Angle Calculation Algorithm: For each exercise, the system calculates angles between specific body landmarks (e.g., shoulders, elbows, knees, and ankles) using trigonometric functions. These angles are compared against predefined thresholds derived from research and expert guidelines to evaluate posture accuracy.
3. Posture Analysis Logic: Based on the calculated angles, posture deviations are identified by classifying movements into "correct" or "incorrect" categories. This logic ensures that feedback is exercise-specific, tailored to detect common form errors.
4. Feedback Generation Mechanism: A visual feedback mechanism is implemented using the calculated angles and posture analysis. The system highlights incorrect postures by overlaying markers or messages on the live video feed, guiding the user to correct their form.

5.3.3. Project Deployment

Figure 9-13 shows our plan to deploy the project in the real world.

5.3.4. System Screenshots

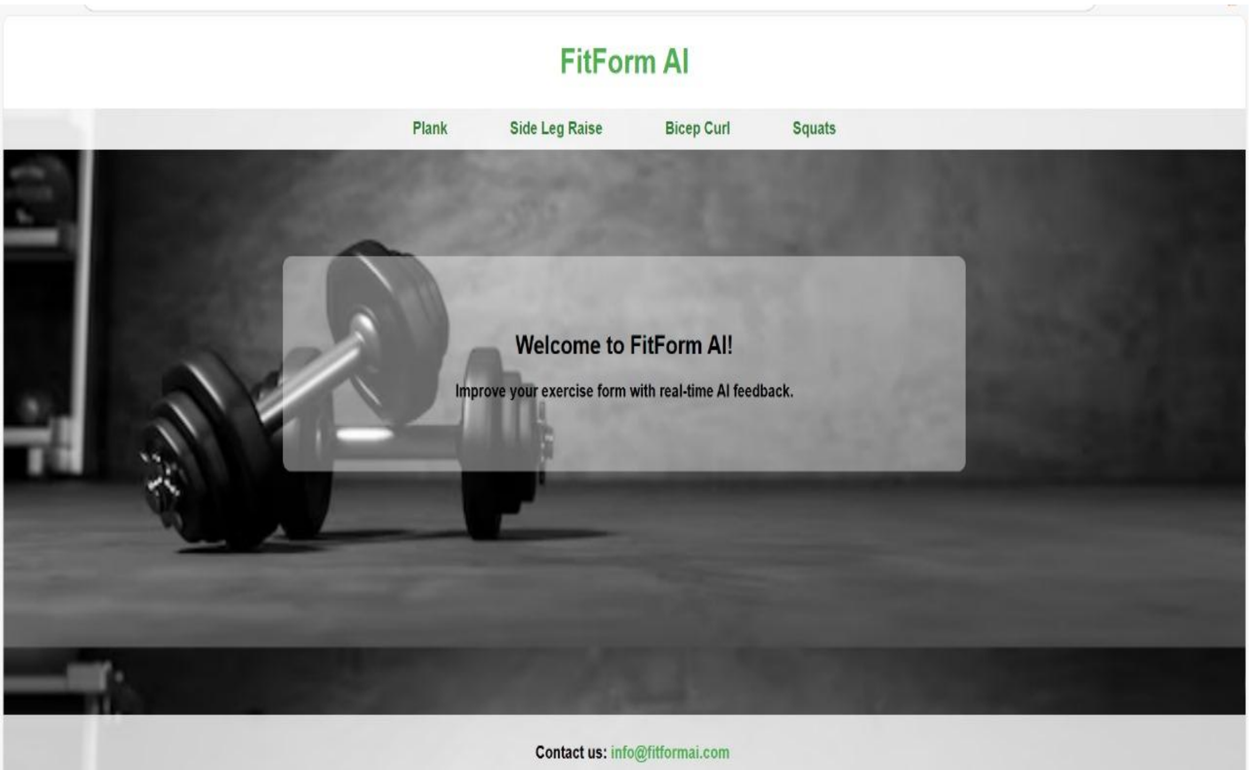

Figure 9: Deployment 1

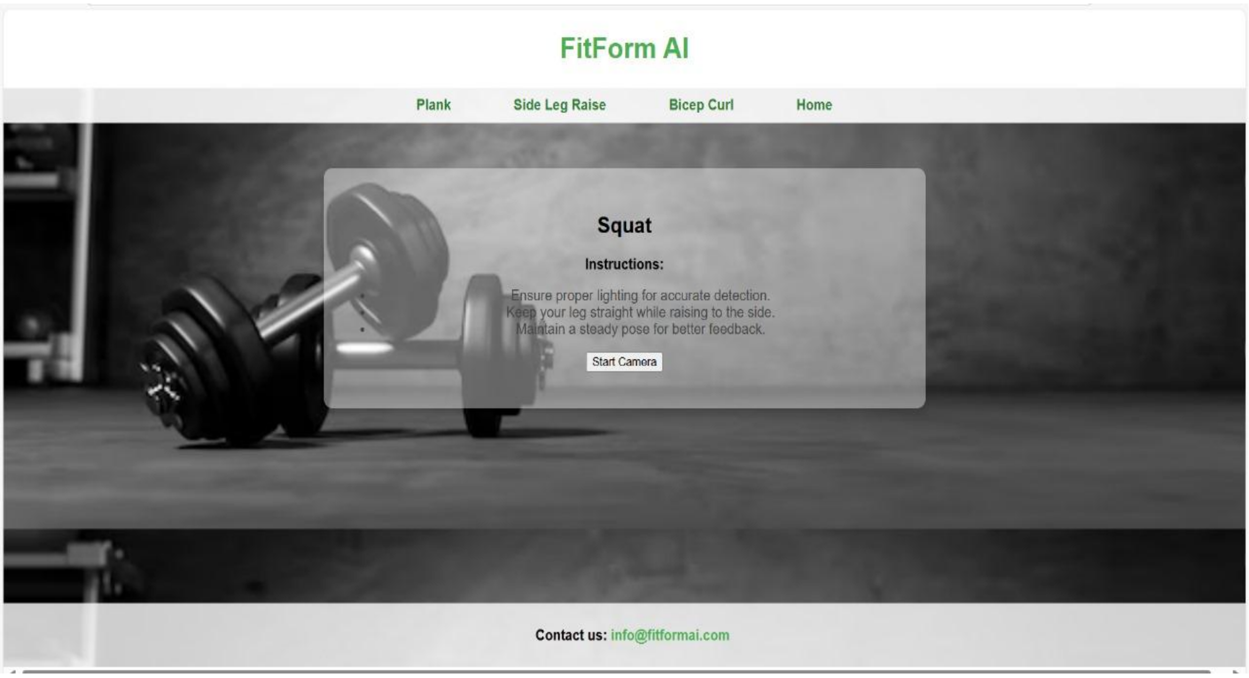

Figure 10: Deployment 2

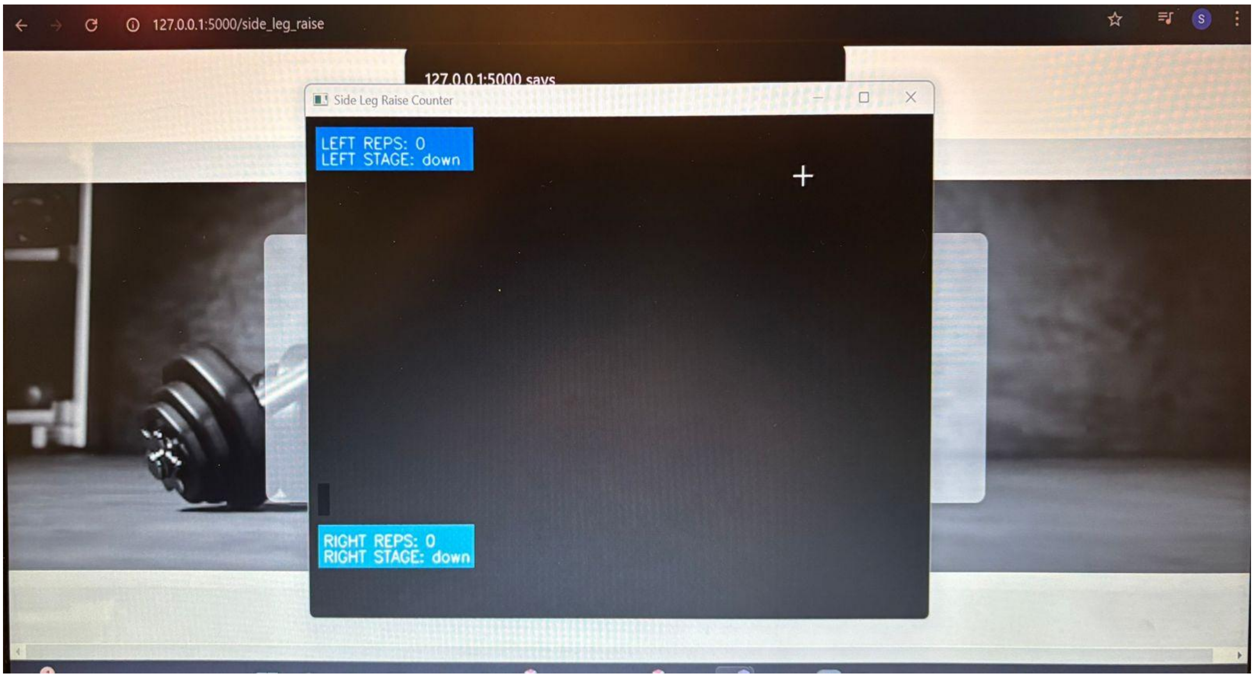

Figure 11: Video Stream

Figure 12: Fingers Webcam 1080 HI-RES

Figure 13: Digitek Tripod DTR-525

5.4. Testing Process

5.4.1. Test Plan

A test plan for this project focuses on ensuring the system's accuracy, reliability, and user-friendliness in providing real-time exercise form corrections.

5.4.1.1. Features Testing

- Validate the precision of the MediaPipe pose model in detecting and tracking body landmarks.
- Assess the accuracy of joint angle calculations using predefined reference values.
- Confirm the system's ability to classify posture as correct or incorrect based on thresholds.
- Ensure accurate detection and counting of repetitions across various exercise routines.
- Evaluate the user interface for ease of navigation, feedback clarity, and responsiveness.
- Verify seamless integration between pose detection, angle calculation, and feedback generation.

5.4.1.2. Test Strategy

Pose Detection Accuracy Testing

- Objective: Verify the accuracy of the MediaPipe pose model in detecting key body landmarks.
- Approach: Conduct tests using real-time video streams under various lighting conditions and user scenarios.

Angle Calculation Testing

- Objective: Validate the accuracy of joint angle calculations.
- Approach: Compare calculated angles with predefined thresholds for each exercise.

Posture Analysis Testing

- Objective: Ensure accurate evaluation of posture for supported exercises.
- Approach: Test the system using real-time video inputs to analyses posture accuracy based on calculated joint angles and predefined thresholds.

User Interface Testing

- Objective**:** Evaluate the usability and clarity of the user interface.
- Approach**:** Test the interface for ease of use, responsiveness, and the ability to display relevant feedback and instructions during exercise sessions.

5.4.1.3. Test Techniques

We executed the code for each exercise module and conducted numerous sample tests to validate all functionalities, ensuring smooth performance without any hindrance. Ensuring the MediaPipe pose model and angle calculation algorithms worked not only for the sample test cases but also for various real-time scenarios, we expanded our test cases from 10 to 15 across diverse exercises.

We tested every component repeatedly, including pose detection, angle calculation, posture analysis, repetition counting, and feedback generation, to ensure the system maintained minimum error rates.

5.4.2. Test Cases

Table 4: Test Cases and Expected Output

S. No.	Exercise Type	Test Cases	Expected Output
1.	Plank	Correct Plank Posture	Timer starts or continues. "Good posture!" appears.
2.	Plank	Incorrect Shoulder Alignment	Timer pauses. "Adjust your shoulder alignment" appears.
3.	Plank	Incorrect Hip-Shoulder-Elbow angle	Timer pauses. "Adjust your hip-shoulder-elbow alignment.
4.	Plank	User Out of Frame	No timer activation or correction messages displayed.
5.	Side Leg Raise	Correct Side Leg Raise	Repetition count increments. "Good form!" appears.
6.	Side Leg Raise	Knee Bending During Raise	Repetition count pauses. "Keep your leg straight" appears.
7.	Side Leg Raise	Insufficient Raise Height	No repetition count. "Raise your leg higher" appears.
8.	Squats	Proper Correct Squat	Squat classified as correct. No feedback required.
9.	Squats	Incorrect Squat - Knee Over Toe	Squat classified as incorrect.

10.	Squats	Misaligned Camera Detection	Warning about camera misalignment displayed.
11.	Squats	Incorrect Squat - Bend Backward	Squat classified as incorrect.
12.	Squats	Incorrect Squat - Low Hip	Squat classified as incorrect.
13.	Squats	Incorrect Squat - Deep Squat	Squat classified as incorrect.
14.	Bicep Curl	Correct Bicep Curl	Repetition count increments.
15.	Bicep Curl	Arm Not Aligned with Body	No repetition count. "Bring elbow close" appears.
16.	Bicep Curl	Insufficient Raise Height	No repetition count. "Raise your arm higher" appears.
17.	Bicep Curl	Arm raised too high.	No repetition count: Lower your arm" appears.

5.5 Results

The test results demonstrated the effectiveness of the posture detection system. In each test case, the system successfully detected correct and incorrect postures as per the expected outcomes. The results are summarised below:

5.5.1 Squats Test Results

Correct Squat (Case 1): The system accurately identified proper squats, recognizing correct alignment and movement with no correction feedback needed.

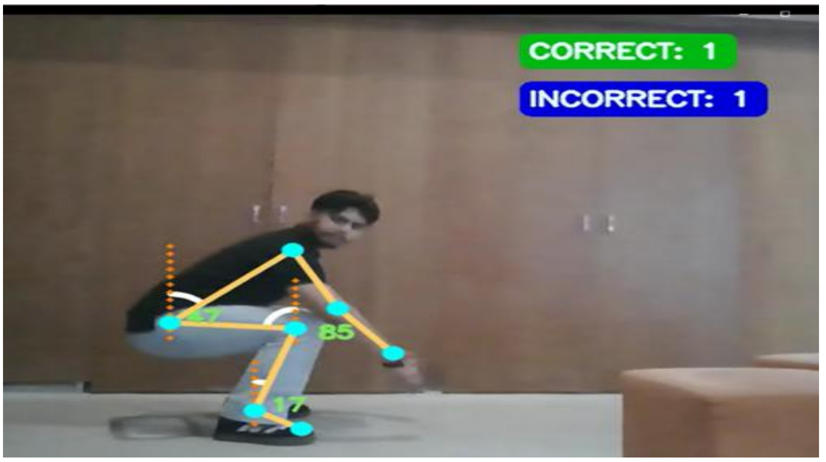

Figure 14

Incorrect Squat - Knee Over Toe (Case 2): The system detected the knee-over-toe fault during squats where the knees crossed over the toes, pausing the repetition count and displaying "Knee Falling Over Toe" for corrective guidance.

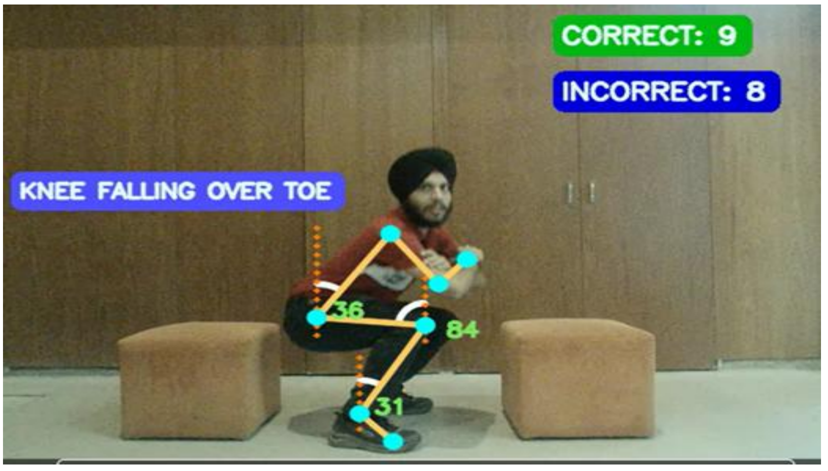

Figure 15

Camera Misalignment (Case 3): The system displayed a "Camera Not Aligned Properly" warning when the user was out of optimal frame or moved backward, ensuring the user adjusted their position.

Figure 16

Forward Bend (Case 4): When the user squatted with an extreme forward lean, the system classified the squat as incorrect and displayed "Backward Bend" to prompt corrective action.

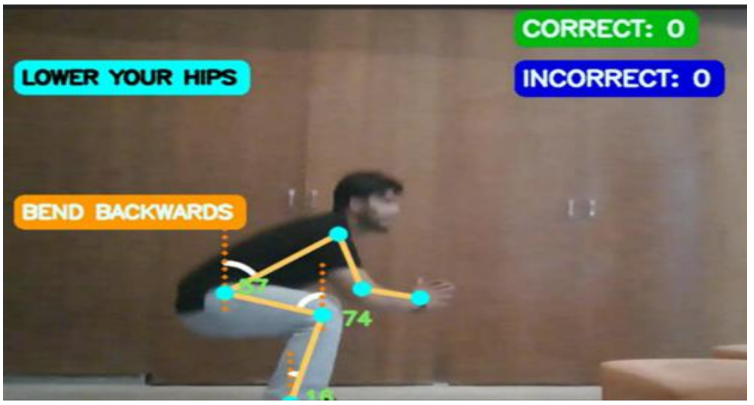

Figure 17

Low Hip (Case 5): Insufficient depth in the squat was detected, with the system displaying "Lower your Hips" to encourage proper hip alignment and depth.

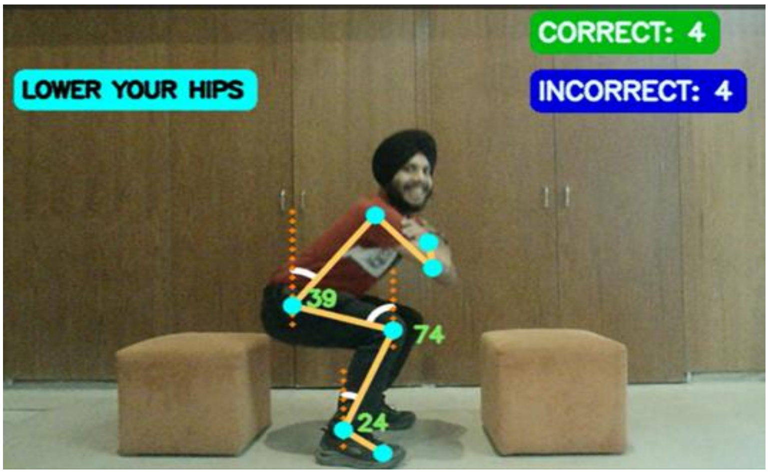

Figure 18

Deep Squat (Case 6): Squats with knees going too deep were identified as incorrect, with the system providing "Squat Too Deep" as visual feedback to ensure proper form.

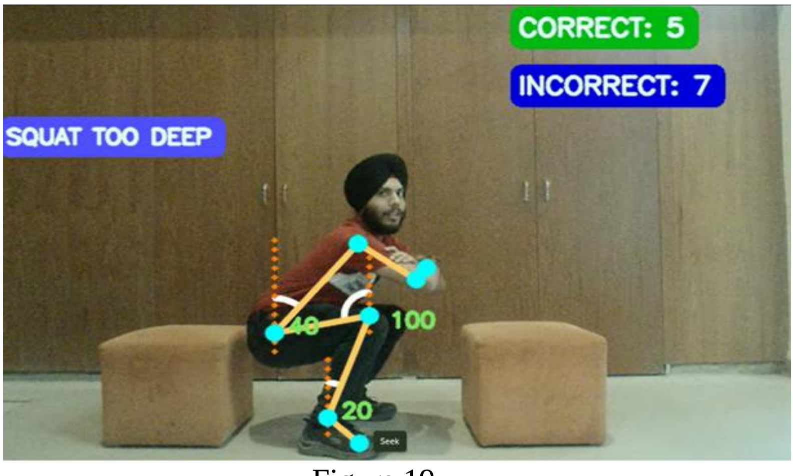

Figure 19

5.5.2 Plank Test Results

Correct Posture (Case 1): The system accurately recognized the correct posture, displaying the "Good posture!" message and incrementing the timer without interruptions.

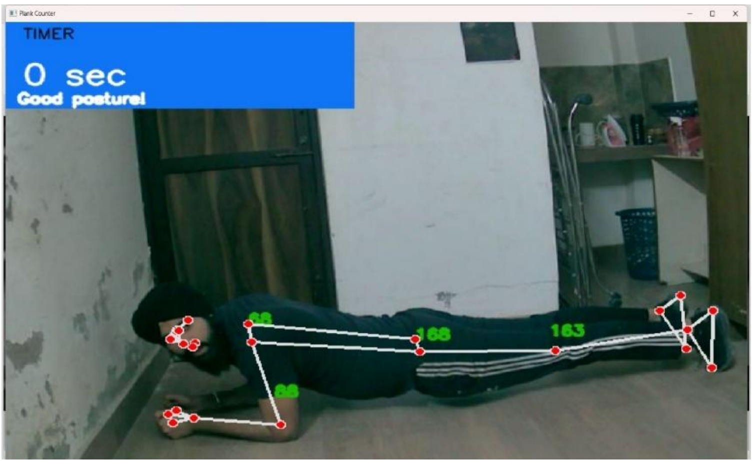

Figure 20

Incorrect Shoulder Alignment (Case 3):: When the shoulder angle is not within the ideal range, the system accurately displays "Adjust your shoulder alignment."

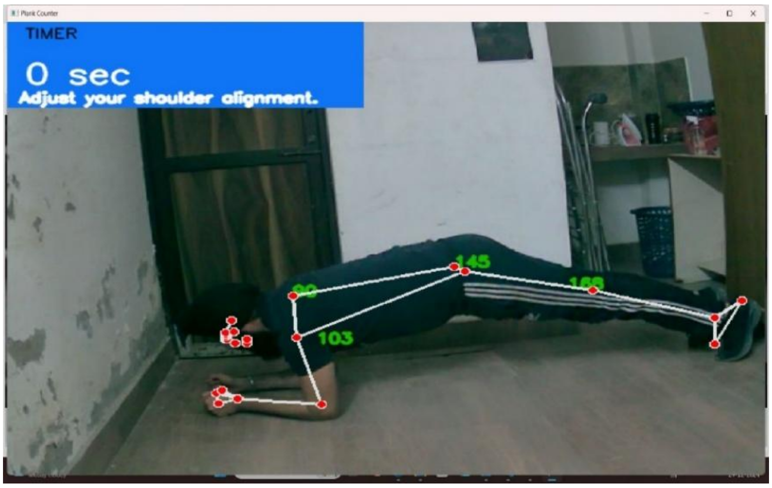

Figure 21

Incorrect hip-shoulder-elbow alignment (Case 5): When the hip-shoulder-elbow angle is not within the ideal range, the system accurately displays "Adjust your hip-shoulder-elbow alignment.

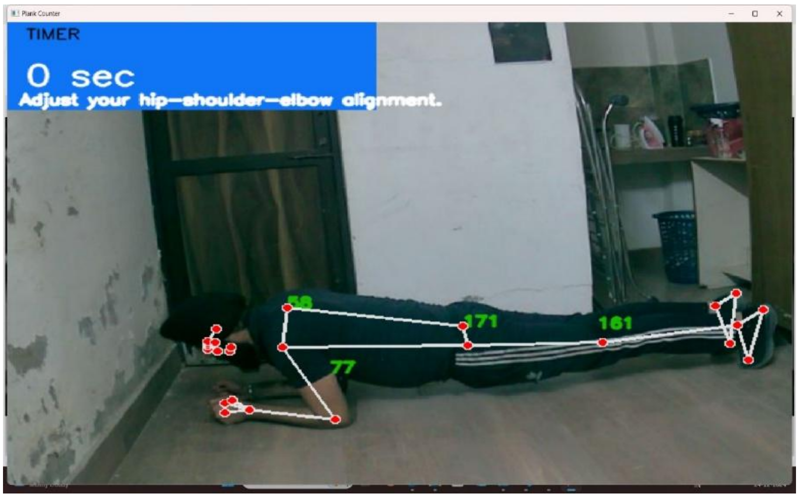

Figure 22

User Out of Frame (Case 6): When the user moved out of the frame, the system did not respond, as expected, showing that it only activates when a person is detected.

Figure 23

5.5.3 Side Leg Raise Test Results

Correct Side Leg Raise (Case 1): The system accurately identified correct raises, incrementing the count and displaying "Good form!" without errors.

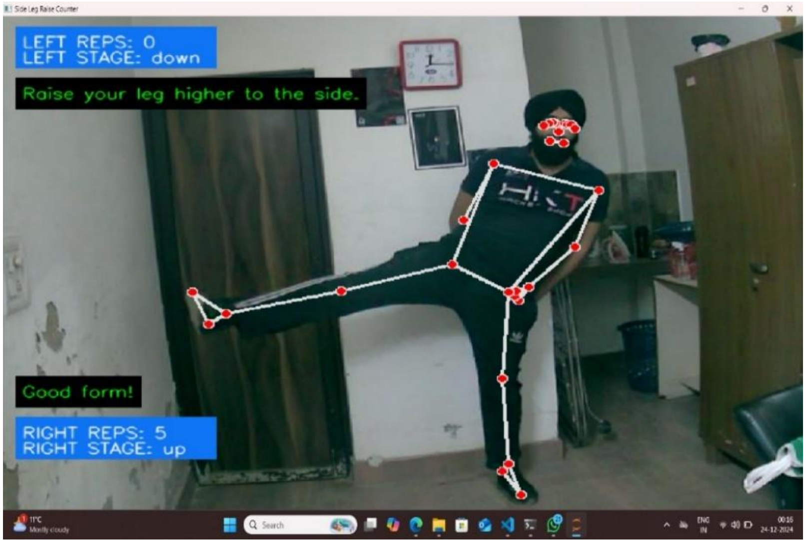

Figure 24

Knee Bending (Case 2): Bent knee raises were successfully identified, with the message "Keep your leg straight" appearing to correct the movement.

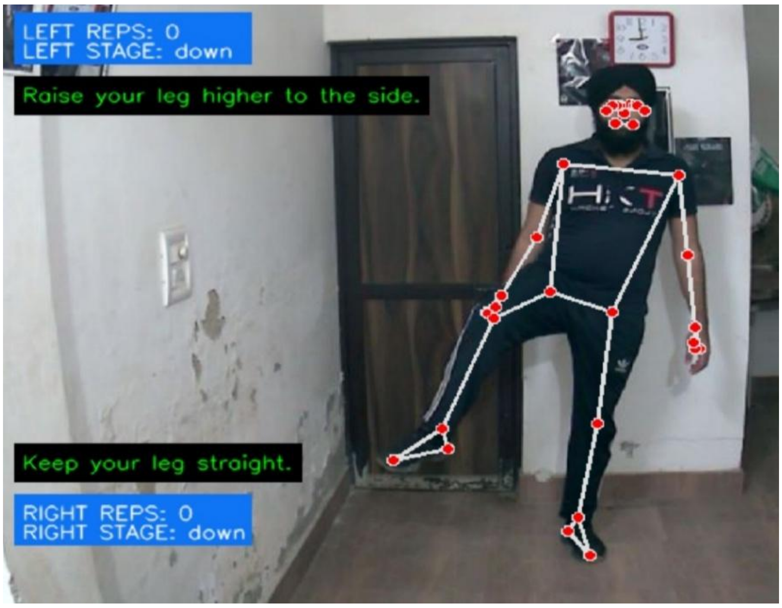

Figure 25

Insufficient Raise Height (Case 3): When the leg did not reach the target height, the system accurately withheld repetition count and displayed "Raise your leg higher."

Figure 26

5.5.4 Bicep Curl Test Results

Correct Bicep Curl (Case 1): The system accurately identifies correct raises, incrementing the count.

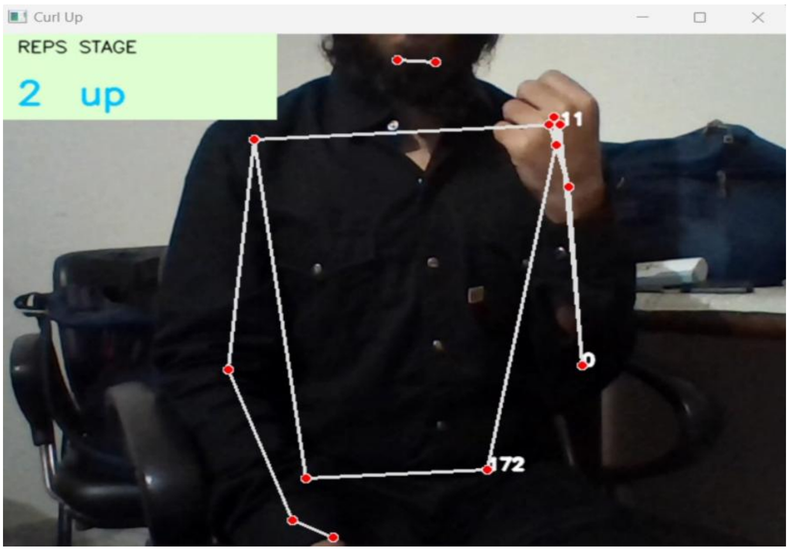

Figure 27

Arm not aligned with body (Case 2): The system detected incorrect directions when the arm was raised, pausing the count and displaying "Bring elbow close".

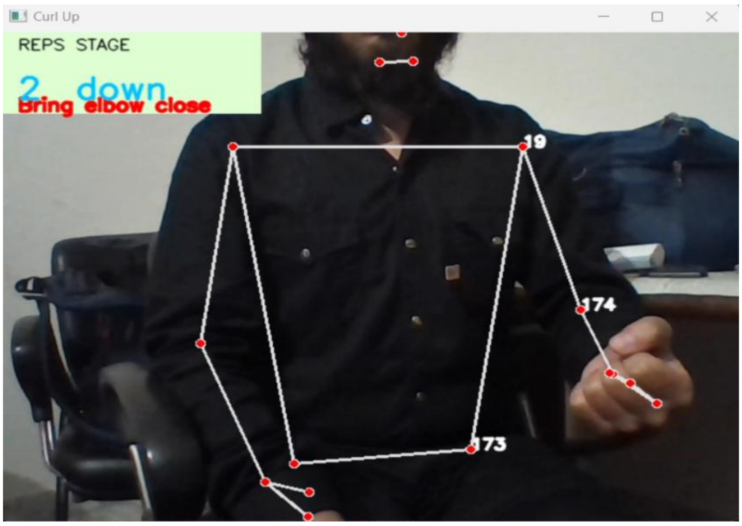

Figure 28

Insufficient Raise Height (Case 3): When the arm did not reach the target height, the system accurately withheld repetition count and displayed "Raise your arm higher."

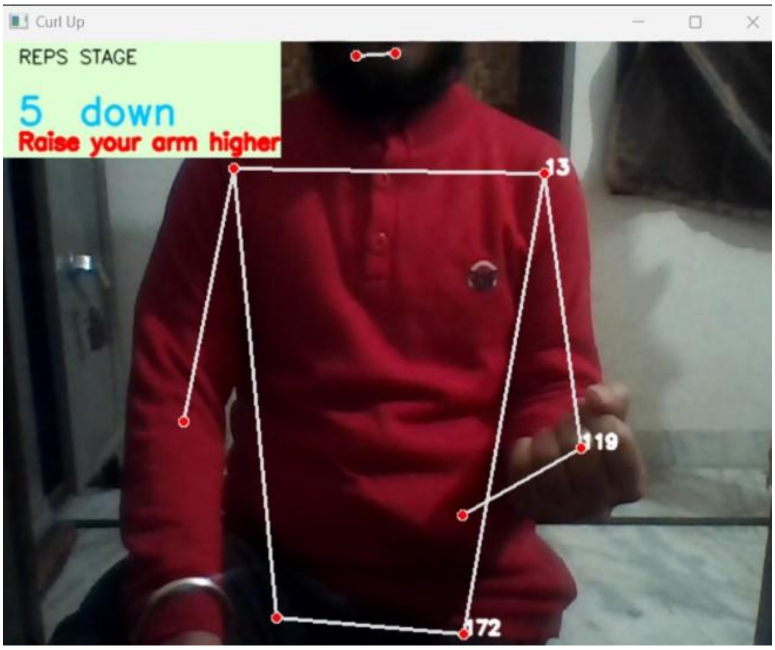

Figure 29

"Lower your arm"(Case 4) : : When the arm is raised too high and does not lower to the appropriate position, the system accurately withheld repetition count and displayed "Lower your arm."

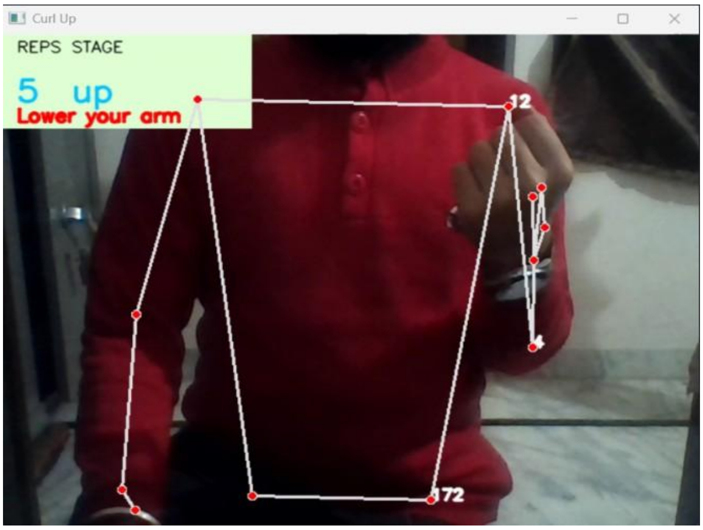

Figure 30

5.6 Validation of Objectives

Table 5: Validation of Objectives

S. No.	Objectives	Status
1.	To develop a real-time exercise correction system using MediaPipe for pose detection and UI.	Successful
2.	Verify the system's ability to accurately detect and track body landmarks for different exercises.	Successful
3.	Ensure the joint angle calculations are accurate and align with predefined thresholds for each exercise.	Successful
4.	Test the posture analysis system to correctly classify user posture as "correct" or "incorrect."	Successful
5.	Validate the repetition counting mechanism by accurately tracking exercise phases and counting reps.	Successful
6.	Ensure the user interface provides clear and responsive real-time feedback during exercise sessions.	Successful

CONCLUSIONS AND FUTURE DIRECTION

6.1 Conclusions

This project presents a real-time exercise correction system utilizing MediaPipe for pose estimation, OpenCV for video processing, NumPy for calculations, and Flask, HTML, CSS and javascript for an interactive user interface. The system ensures accurate body landmark detection, joint angle calculations, and posture analysis for a variety of exercises, including planks, squats, bicep curls, and standing leg raises. The real-time feedback mechanism provides users with corrective guidance, enhancing their exercise form. With rigorous testing, the system's accuracy and reliability have been validated. This project marks a significant advancement in exercise correction technology, combining precise pose estimation with an accessible user interface for personalized fitness training.

6.2. Environmental, Economic and Social Benefits

The implementation of the real-time exercise correction system offers significant environmental, economic, and social benefits.

1. Environmental Benefits: By enabling users to perform exercises at home with real-time feedback, the system helps reduce the need for commuting to gyms, thus contributing to lower carbon emissions and promoting a more sustainable lifestyle. Additionally, the digital nature of the system reduces paper waste by eliminating the need for printed workout guides or monitoring logs.
2. Economic Benefits: From an economic standpoint, the system minimizes the need for personal trainers, reducing the costs associated with in-person training sessions. The real-time correction feature offers an affordable alternative for users, making professional-level exercise guidance accessible without ongoing expenses. Moreover, the system’s scalability ensures it can be adopted by a wide range of individuals, optimizing resource use while lowering overall fitness-related costs.
3. Social Benefits: The system fosters a healthier society by encouraging individuals to exercise correctly and consistently, reducing the risk of injury and improving overall well-being. The ease of use and accessibility through a user interface promotes widespread adoption, especially for individuals in remote areas or those with limited access to gyms. Additionally, by offering personalized feedback, the system empowers users to track progress independently, enhancing self-confidence and promoting a more active lifestyle.

6.3 Reflections

This project has been a valuable learning experience, demonstrating the feasibility of using computer vision for real-time exercise posture correction. While we successfully developed a foundational system, challenges arose in accurately detecting subtle postural deviations and refining feedback mechanisms. Key areas for future improvement include expanding the exercise library, enhancing robustness to environmental factors, integrating more sophisticated machine learning models, and conducting extensive user studies. Despite these challenges, this project has laid a strong foundation, and the insights gained will be invaluable for creating more advanced and impactful systems that can significantly benefit individuals in their fitness journeys.

6.4 Future work

The successful development of this initial exercise posture correction system provides a strong foundation for future advancements. Key areas for future work include expanding the exercise library to encompass a wider range of exercises, refining posture analysis through the integration of more sophisticated AI models, enhancing user experience with features like personalized training plans and mobile app integration, and improving robustness to environmental factors.

Incorporating wearable sensors, such as inertial measurement units (IMUs) and electromyography (EMG) sensors, can provide valuable additional information about the user's movement and muscle activity, enriching the data used for posture analysis. Furthermore, personalizing feedback by tailoring it to individual needs and preferences can significantly enhance the effectiveness of exercise correction systems. Finally, utilizing multimodal learning approaches that combine information from multiple modalities, such as video, audio, and wearable sensor data, can improve the accuracy and robustness of exercise analysis, leading to more comprehensive and effective feedback for users.

PROJECT METRICS

7.1. Challenges Faced

Some of the key challenges that were encountered during the development of this exercise posture correction system.

1. Accurately detecting and tracking key body joints in real-time, especially in dynamic environments with varying lighting and user movements.
2. Creating reliable algorithms for angle calculation and posture analysis that can accurately identify subtle postural deviations across a wide range of exercises.
3. Developing clear, concise, and actionable feedback mechanisms that are easily understandable and applicable by users.
4. Ensuring the system's robustness to variations in lighting, background clutter, user clothing, and camera placement.
5. Designing an intuitive and user-friendly interface that is easy to navigate and understand with minimal loading time.

7.2 Inter-disciplinary Knowledge Sharing

1. Fosters Innovation: By combining diverse perspectives and methodologies, interdisciplinary knowledge sharing can lead to novel solutions and breakthroughs that would be unattainable within the confines of a single discipline.

2. Enhances Problem-Solving: Complex challenges often require a multi-faceted approach. Interdisciplinary collaboration allows for a more holistic understanding of the problem and the development of more comprehensive and effective solutions.

3. Breaks Down Boundaries: It encourages communication and collaboration across disciplinary boundaries, fostering a more interconnected and collaborative research environment.

4. Promotes Deeper Understanding: By integrating knowledge from different fields, researchers gain a deeper and more nuanced understanding of complex phenomena.

5. Drives Interdisciplinary Research: It encourages the pursuit of research questions that transcend traditional disciplinary boundaries, leading to new areas of inquiry and discovery

7.3 Work Schedule (Gantt Chart)

It provides a visual representation of a project's schedule, outlining tasks, dependencies, durations, and milestones. It helps project managers and teams plan, track, and manage the development process, ensuring that tasks are executed in a logical order and are completed on time. It assists in identifying critical paths, resource allocation, and progress monitoring, contributing to effective project management and successful software development.

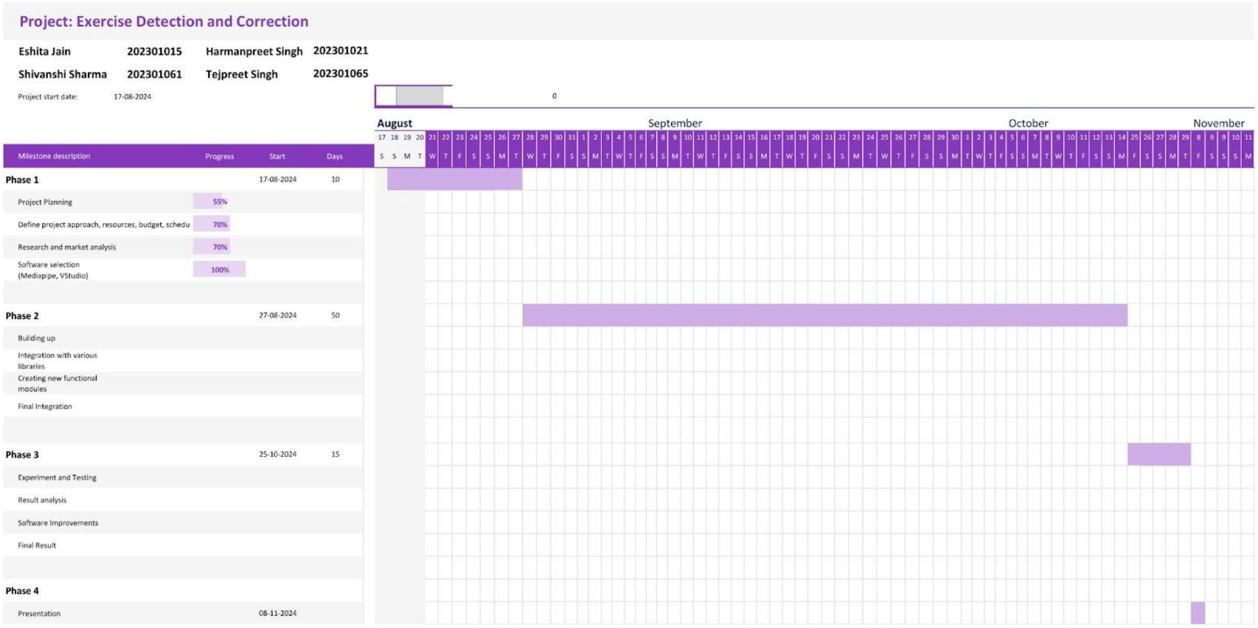

Figure 31 : Gantt chart

REFERENCES

[1] Shrey Sitapara; Akarsha D.P.; Mahika Jain; Sakshi Singh; Nitesh Gupta, “ Novel Approach for Real-Time Exercise Posture Correction Using Computer Vision and Cnn” 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Publisher: IEEE, DOI: 10.1109/AIKIIE60097.2023.10389979

[2] Rahul Ravikant Kanase, Akash Narayan Kumavat, Rohit Datta Sinalkar, Sakshi Somani, “Pose Estimation and Correcting Exercise Posture” Publisher: ITM Web of Conferences, DOI: 10.1051/itmconf/20214003031

[3] Steven Chen, Richard Yang, “Pose Trainer: Correcting Exercise Posture using Pose Estimation” Department of Computer Science, Stanford University, Publisher: DOI: 10.48550/arXiv.2006.11718

[4] Urmi Dedhia, Pratham Bhoir, Prateek Ranka, Pratik Kanani,”Pose Estimation and Virtual Gym Assistant Using MediaPipe and Machine Learning”, September 2023, DOI: 10.1109/NMITCON58196.2023.10275938

[5] Hitesh Kotte, Milos Kravcik, Nghia Duong-Trung, “Real-Time Posture Correction in Gym Exercises: A Computer Vision-Based Approach for Performance Analysis, Error Classification and Feedback”, Educational Technology Lab, German Research Center for Artificial Intelligence (DFKI), Alt-Moabit 91C, 10559 Berlin, Germany, October 2023

[6] Tathagat Maitray; Utkarsh Singh; Vaidansh Kukreja; Ashish Patel; Vishnu Sharma, “Exercise posture rectifier and tracker using MediaPipe and OpenCV”,2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Publisher: IEEE

[7] Michel Magdy Helmy; Mary Saad Yousef; Mina Anis Shokry; Mina Khalifa Gendy; Mina Lotfy Favez; Kirollos Aziz Galal, “PoseRight: An AI Empowered Automatic Posture Estimation and Correction Model”, 2024 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Publisher: IEEE

[8] Mukundan Chariar; Shreyas Rao; Aryan Irani; Shilpa Suresh; C S Asha, “AI Trainer: Autoencoder Based Approach for Squat Analysis and Correction” Volume 11: 2023, Publisher: IEEE

[9] Guttikonda Gowrish; Yenduri Sesha Vineeth; M. Thenmozhi, “AI Powered gym tracker using AI Pose estimation” , 2024 2nd International Conference on Networking and Communications (ICNWC), Publisher: IEEE

[10] GitHub: augmentedstartups/Pose-Estimation , “Pose-Estimation using OpenPose”

[11] GitHub: ruitaiS/repCounter , “Squat Rep Counter Using OpenCV and MediaPipe for Posture Detection”

[12] Medium: Side-Lying-Leg-Lift Exercise Detection Using MediaPipe OpenCV Python

[13] Numpy: https://numpy.org/