AI PERSONAL TRAINER

An AI-driven web app solution for optimizing workout experience

Duy Le - BDS21DON032
Thao Chu - BDS21DON022

23/04/2024

S P JAIN SCHOOL OF GLOBAL MANAGEMENT_SYDNEY

Project Capstone

ACKNOWLEDGEMENTS

We extend our heartfelt gratitude to Professor JP Aggarwal for his invaluable guidance and unwavering support throughout the development of this software plan. Professor Aggarwal's weekly meetings provided a platform for insightful discussions, feedback, and recommendations, significantly enriching our project's scope and direction. His expertise and mentorship have been instrumental in shaping the quality and depth of our work.

TABLE OF CONTENTS

I. INTRODUCTION

The AI fitness project aims to innovate the fitness industry by leveraging cutting-edge technologies to provide personalized posture correction and exercise plans. In today's sedentary lifestyle, maintaining proper posture during gym workouts is often overlooked, leading to various health issues and injuries. Recognizing this gap, our project endeavors to address these challenges by developing an innovative AI-powered fitness application.

Background and Rationale of the Project:

The AI fitness project is born out of a profound understanding of the challenges individuals face in maintaining proper posture and exercise techniques, especially during unprecedented times like the COVID-19 pandemic. With lockdowns restricting access to gyms and personal trainers, many individuals have resorted to home-based workouts. However, exercising without proper guidance can lead to incorrect postures, increasing the risk of injuries and hindering fitness progress. Improper posture not only compromises the effectiveness of exercises but also heightens the likelihood of sustaining injuries, thus impeding individuals' fitness journeys. Additionally, beginners in the gym environment often struggle with mastering correct techniques, while individuals who prefer exercising alone may lack the accountability and feedback provided by personal trainers. Recognizing these difficulties, our project aims to bridge the gap between individuals' fitness goals and their ability to exercise safely and effectively, regardless of their location or experience level. By leveraging AI-driven posture correction and personalized exercise plans, we seek to empower users to achieve optimal fitness outcomes and enhance their overall well-being.

Objectives and Goals:

The primary objective of the project is to create a comprehensive solution that not only detects and corrects improper posture in real-time but also delivers personalized exercise plans tailored to individual users' needs and fitness goals. By integrating computer vision algorithms, such as OpenCV and MediaPipe, our application can accurately analyze users' movements and provide instant feedback to improve their posture.

Furthermore, the project aims to utilize the LLM Large Language Model to generate personalized weekly exercise plans based on users' fitness levels, preferences, and progress. This personalized approach ensures that users receive tailored recommendations that align with their goals, maximizing the effectiveness of their fitness routines.

Through this project, we envision empowering users to take control of their fitness journey, achieve optimal results, and reduce the risk of injuries associated with improper posture and exercise techniques. By combining advanced technologies with fitness expertise, we strive to revolutionize the way people approach fitness and wellness.

II. PROJECT OVERVIEW

Overview of the AI Fitness Web Application:

The AI Fitness application is a modernized solution designed to revolutionize the way individuals approach their fitness routines. By leveraging advanced technology, including computer vision algorithms and machine learning models, the application offers users personalized guidance and real-time feedback to enhance their workout experience. Whether users are beginners seeking proper form guidance or seasoned athletes aiming to optimize their performance, the AI Fitness application caters to diverse fitness needs and preferences.

The AI Fitness application provides personalized correction for five core exercises and generates customized workout plans based on user input. These plans encompass a diverse range of exercises targeting major muscle groups, complete with recommended sets and repetitions, providing comprehensive fitness support in an online-optimized format.

Description of Key Features and Functionalities:

1. Real-time Posture Correction: The application utilizes advanced computer vision algorithms to analyze users' movements and provide immediate feedback on their posture during workouts. Through visual cues and audio prompts, users can correct form deviations and minimize the risk of injuries.

2. Personalized Exercise Plans: Driven by the LLM Model, the application creates bespoke exercise plans tailored to individual user goals, preferences, and fitness levels. These plans are crafted based on user input, including age, weight, strengths, and preferred workout duration.

3. Intuitive User Interface: With a user-friendly web interface built using Streamlit, the application offers seamless navigation and accessibility across devices. Users can easily access key features, track their progress, and engage with the community, fostering a supportive and motivating environment for achieving their fitness goals.

III. TECHNOLOGY STACK

The AI fitness application harnesses a powerful technology stack to deliver seamless user experiences and cutting-edge functionality.

1. OpenCV and MediaPipe:

These computer vision libraries are the backbone of the posture correction feature, enabling real-time detection and analysis of users' body postures during workouts. OpenCV provides robust image processing capabilities, while MediaPipe offers advanced pose estimation models, allowing the application to accurately identify and correct improper postures.

2. LLM - Large Language Model:

The LLM Model serves as the core engine behind the personalized exercise plans offered by the application. Leveraging machine learning techniques, the model analyzes user data, including age, weight, and fitness goals, to generate tailored workout routines that optimize effectiveness and minimize injury risk.

3. Streamlit:

Streamlit is employed to develop the user interface of the application, facilitating the creation of intuitive web pages for seamless interaction. With its simplicity and flexibility, Streamlit enables rapid development and deployment of feature-rich frontends, ensuring a smooth and engaging user experience.

IV. METHODOLOGY

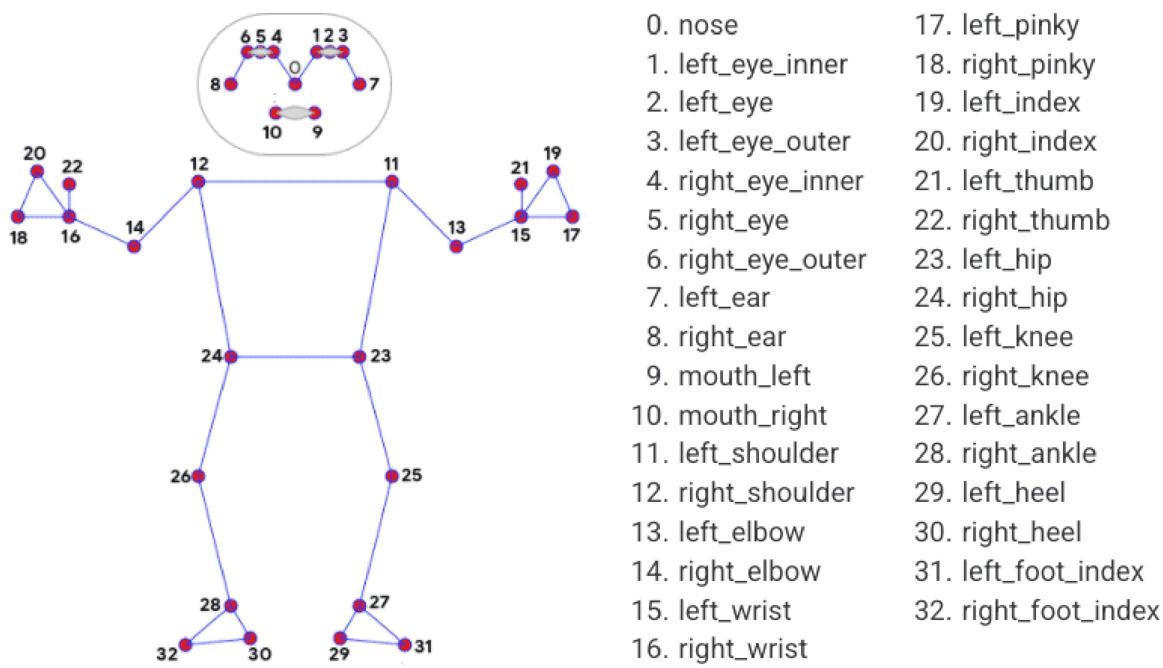
A. POSTURE CORRECTION FUNCTION

In our web application, we have integrated five primary exercises, namely Squats, Bicep Curl, Dumbbell Fly, Dumbbell kickback, and Push-ups, to cater to diverse fitness routines. Each exercise uses the same standardized logic, meticulously designed to ensure accuracy and effectiveness. For an easier explanation of this approach, we will employ the Bicep Curl as a paradigmatic example for elucidation within this report.

a. Body Pose Estimation with MediaPipe

MediaPipe Pose utilizes machine learning to achieve high-precision body pose tracking in videos. It leverages BlazePose, a technology encompassing COCO, BlazeFace, and BlazePalm functionalities, to infer 33 three-dimensional landmarks across the entire body in an RGB video frame.

The solution functions similarly to MediaPipe Hands and MediaPipe Face Mesh, employing a two-stage detection-tracking pipeline. In the first stage, a detector identifies the region of interest (ROI) within the frame, pinpointing the location of the person or pose. Subsequently, the tracker hones in on the ROI, utilizing the cropped frame as input to predict the pose landmarks and segmentation mask within that specific area.



b. Choosing Frontal and Side View for Posture Analysis

In developing our fitness analysis application, considerations arise regarding the viewpoint of the individual captured by the camera.

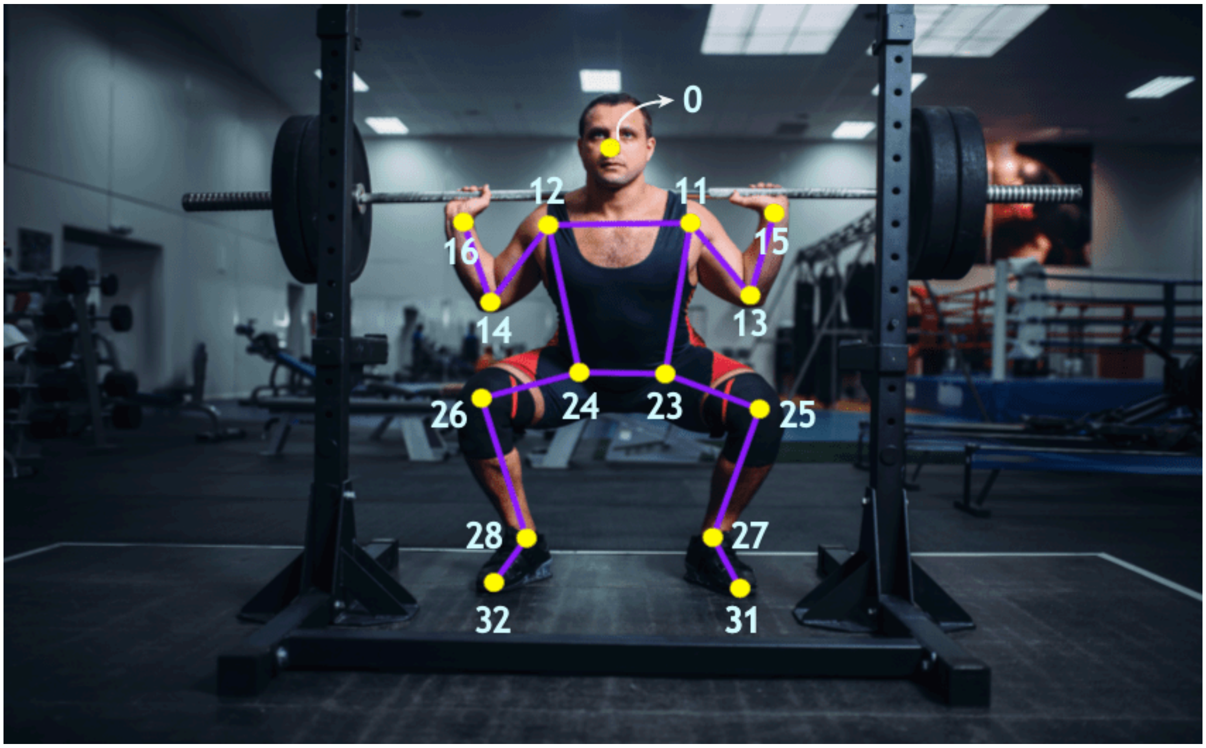
Utilizing the frontal view, we gain insights into both left and right sides, enabling us to utilize slopes and angles of landmark points like knee-hip and knee-knee lines. This information proves valuable in analyzing exercises such as overhead presses, side planks, crunches, and curls. In our web application, the frontal view is used for dumbbell fly exercise.

Meanwhile, the side view offers enhanced estimates of inclinations relative to verticals or horizontals, aiding in the analysis of exercises like deadlifts, pushups, squats, and dips.

Given our focus on Bicep Curl analysis and the necessity for precise inclinations with verticals, we prioritize the side view for optimal accuracy.

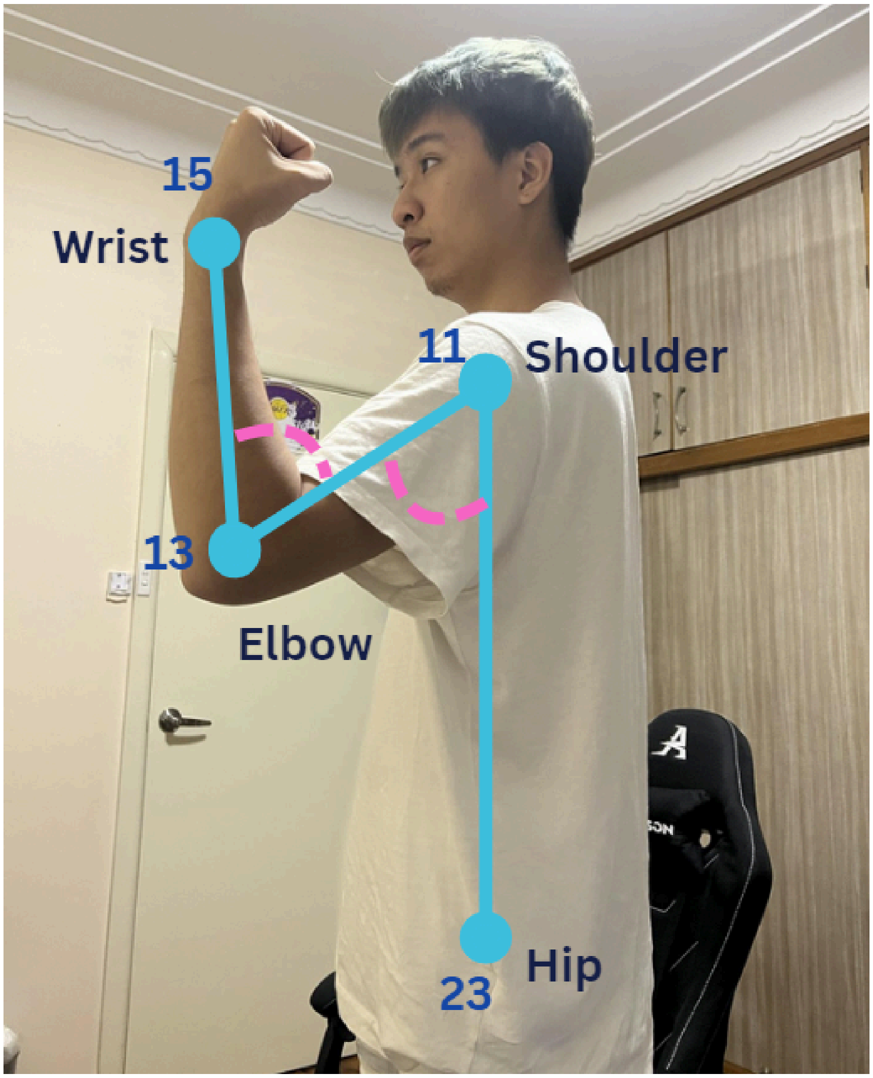
c. Using MediaPipe Pose to Analyze Bicep Curls

In this scenario, we'll concentrate on a specific set of body landmarks crucial for analyzing bicep curls. These landmarks will be highlighted in the below image:



To assess bicep curl form, we'll calculate the angles between the following body parts and vertical lines:

- Shoulder-elbow
- Elbow-wrist
- Shoulder-hip



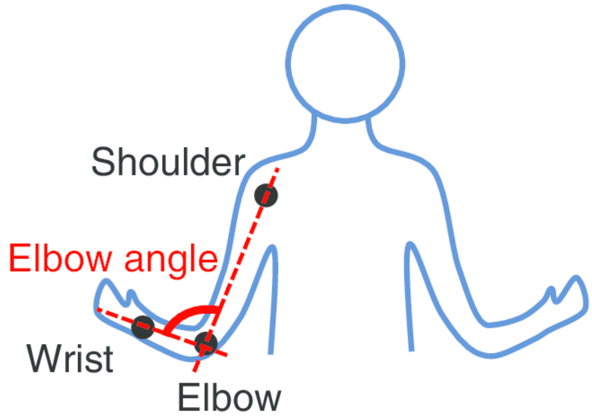
We'll use these angles, along with predefined thresholds (explained later), to determine if the bicep curl is performed correctly. This information will be used to generate real-time feedback messages.

Additionally, we'll calculate the offset angle between the nose and shoulders. This helps ensure the user maintains a proper side view for accurate pose estimation. We'll provide appropriate warnings if the user deviates significantly.

The system will also incorporate inactivity timers. If the user remains inactive for a set period, the counters for proper and improper bicep curls will reset.

d. Bicep Curls' State Tracking Diagram

This app employs a state transition diagram to delineate various states during bicep curl execution. This process will be calculated based on the elbow angle between the elbow-wrist landmark and the elbow-shoulder landmark.



Three states – s1, s2, and s3 – capture the progression of the exercise based on knee angles relative to the vertical. From the Normal phase (s1) to the Transition phase (s2) and finally the Pass phase (s3)

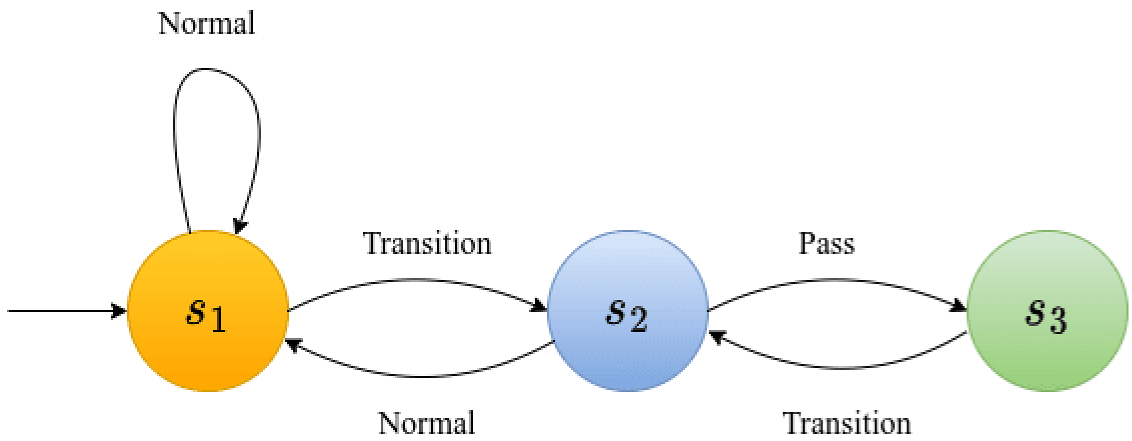
We will implement three distinct states within our application to monitor bicep curl performance. From the Normal phase (s1) to the Transition phase (s2) and finally the Pass phase (s3):

- **State (s1):** In the Normal phase (s1), indicates a standing position and straight arm with an elbow angle of less than 200° and greater than 145° degrees. This state serves as the baseline, where the application updates counters for proper and improper bicep curls.

- **State (s2):** Transition phase (s2) occurs when the elbow angle ranges between 144° and 85°. Users in this phase are transitioning between different stages of the exercise and subsequently advance to state s2.

- **State (s3):** The Pass phase (s3) is activated when the elbow angle falls within a specific range, such as 84° to 35°. In this phase, users have completed the bicep curl motion effectively and progress to state s3.

  States transition flow diagram:



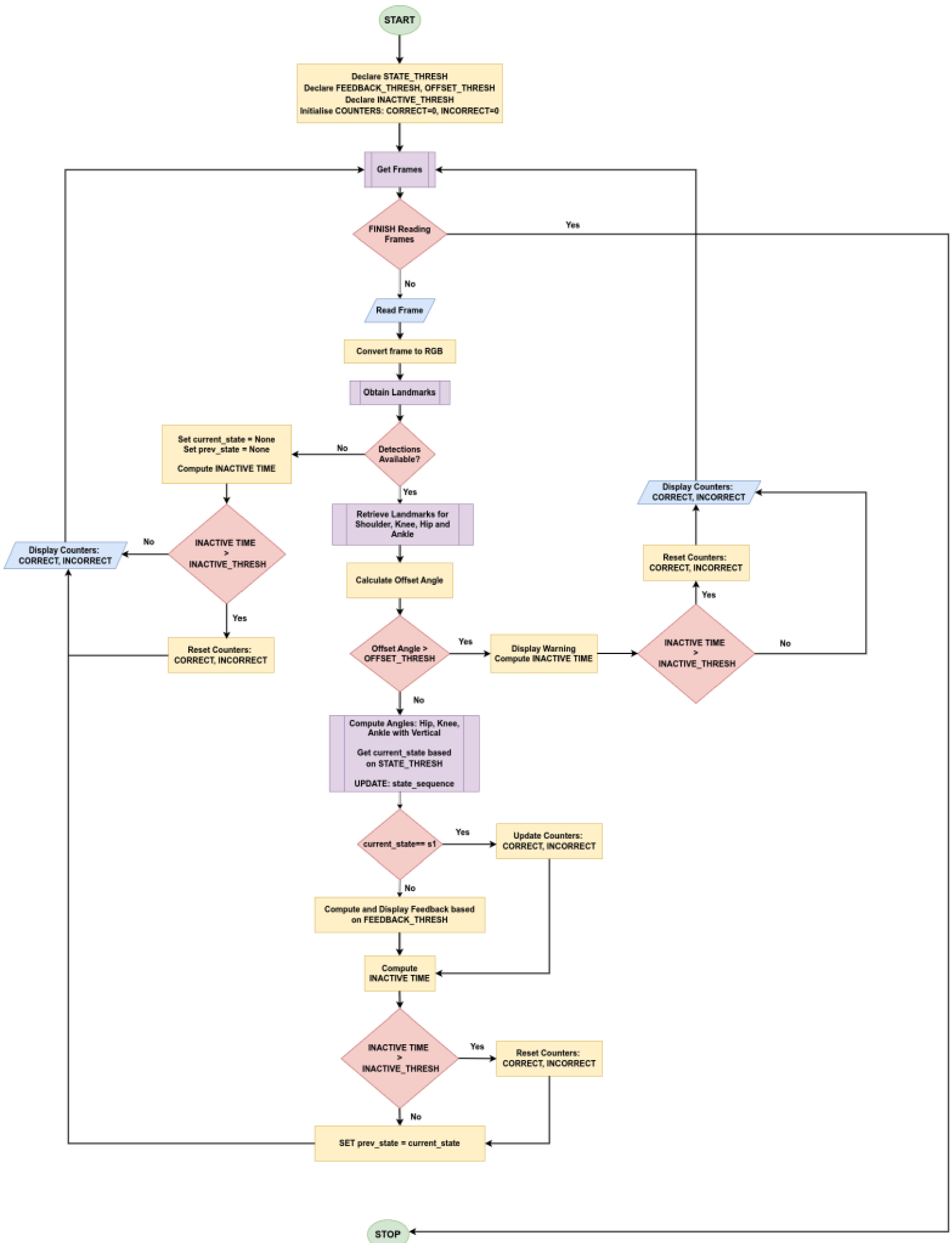
Note:

- ➔ All feedback-related calculations are performed for states s2 and s3.
- ➔ Throughout implementation, we maintain a list called state_sequence, which records the sequence of states as the individual transitions from states s1 to s3 and back to s1. The maximum length of state_sequence is 3 ([s2, s3, s2]). This list is instrumental in determining whether a correct or incorrect bicep curl is executed.

Upon encountering state s1, we reset the state_sequence to an empty list to begin tracking subsequent bicep curl counts.

e. Application Workflow

This application workflow follows a methodology inspired by the LearnOpenCV website (learnopencv.com)



1. We first establish the following thresholds alongside two counters:
    - STATE_THRESH: Thresholds determining the state assigned to each frame.

- FEEDBACK_THRESH: Thresholds determining the feedback information to display.
- OFFSET_THRESH: Threshold indicating if the person faces directly towards the camera.
- INACTIVE_THRESH: Threshold determining inactivity, where counters for CORRECT and INCORRECT are reset.
- Counters: CORRECT and INCORRECT track the number of proper and improper bicep curls, respectively.

2. Each frame from the webcam/video undergoes preprocessing and is passed through MediaPipe's Pose solution.

3. We extract landmarks for the Shoulders, Nose, Knee, Hip, and Ankle if detection landmarks are available; otherwise, we calculate INACTIVE TIME (in seconds) in the absence of detections.
    - If INACTIVE TIME exceeds INACTIVE_THRESH, we reset the CORRECT and INCORRECT counters.

4. We calculate the offset angle for the Nose and Shoulder coordinates.
    - If the offset angle surpasses OFFSET_THRESH, we display the relevant warning and compute INACTIVE TIME as in Step 3.

5. When the offset angle falls within OFFSET_THRESH, we proceed to calculate:
    - Angles of the shoulder-hip, hip-knee, and knee-ankle lines with the verticals.
    - current_state of the frame is based on STATE_THRESH.
    - We maintain a list called state_sequence (as discussed earlier).

6. Upon encountering state s1, we update the CORRECT and INCORRECT counters based on state_sequence. Otherwise, we display feedback messages based on FEEDBACK_THRESH and calculate INACTIVE TIME.

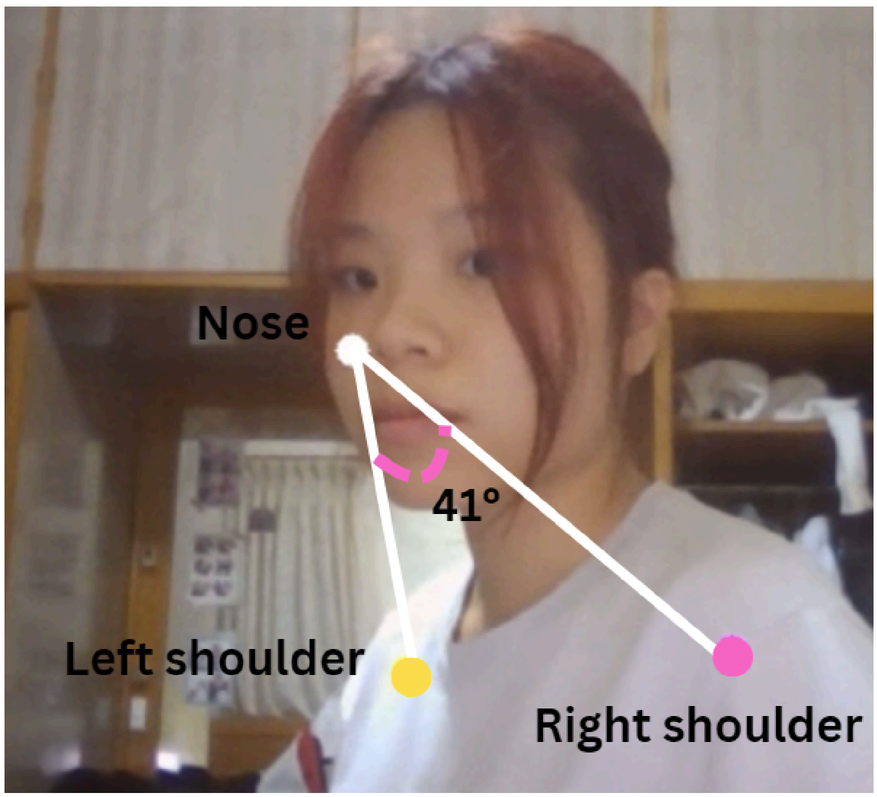
7. prev_state is assigned with current_state and proceeds to retrieve subsequent frames.

f. Angle Calculation concepts

To determine the angle between three points, with one serving as the reference point, we employ the equation:

$$\theta = \arccos \frac{\overrightarrow{P_{1ref}} \cdot \overrightarrow{P_{2ref}}}{|\overrightarrow{P_{1ref}}| \cdot |\overrightarrow{P_{2ref}}|}$$

For example, in calculating the offset angle, we ascertain the angle between the nose and the shoulders, with the coordinates of the nose as the reference point.



When the offset angle exceeds a certain OFFSET_THRESH, we infer that the individual is facing the front of the camera, prompting the display of an appropriate warning message.

The same applies when calculating other angles needed for other exercises.

g. Providing Feedbacks

Our application offers five distinct feedback messages to guide users during a bicep curl exercise:

1. Straighten your back

2. Move hand forward
3. Move hand backward
4. Curl your bicep higher

- Feedback 1 "Straighten your back": This feedback is triggered when the angle between the **shoulder-hip** and **vertical line** falls above a predefined threshold, such as 7°.

- Feedback 2 "Move hand forward": Displayed if meet two conditions i)when the angle between the shoulder-hip line and the shoulder-elbow line exceeds a certain threshold, for instance, 15°; ii) when the x-coordinate of the elbow joint is smaller than the x-coordinate of the shoulder joint.

- Feedback 3 "Move hand backward": Provided two conditions i) when the angle between the shoulder-hip line and the shoulder-elbow line exceeds specified thresholds, also 15°; ii) when the x-coordinate of the elbow joint is higher than the x-coordinate of the shoulder joint.

- Feedback 4 "Curl your bicep higher": This feedback is prompted when the elbow angle is between the elbow-wrist line and the shoulder-elbow line. This feedback will be displayed only in state s2, and happens when the transition from s2 to s3 (not vice-versa) is not met.

These feedback messages are essential for correcting improper bicep curl postures and ensuring effective workout sessions. The thresholds for each feedback are determined through research, real-life analysis, and experimentation.

h. Computing Inactive Times During Bicep Curls

As discussed previously, we implement a mechanism to detect periods of inactivity. Inactivity is defined as when the user maintains a certain state without performing any action for a specified duration, denoted as T. We have set the threshold T to be 15 seconds.

There are three scenarios in which our application identifies inactivity during a bicep curl exercise:

1. The user is facing directly toward the camera, indicated by an offset angle greater than OFFSET_THRESH, for a duration exceeding T seconds.
2. The user remains in the same state (such as holding the dumbbells at a certain position) without transitioning to the next phase of the exercise for more than T seconds.

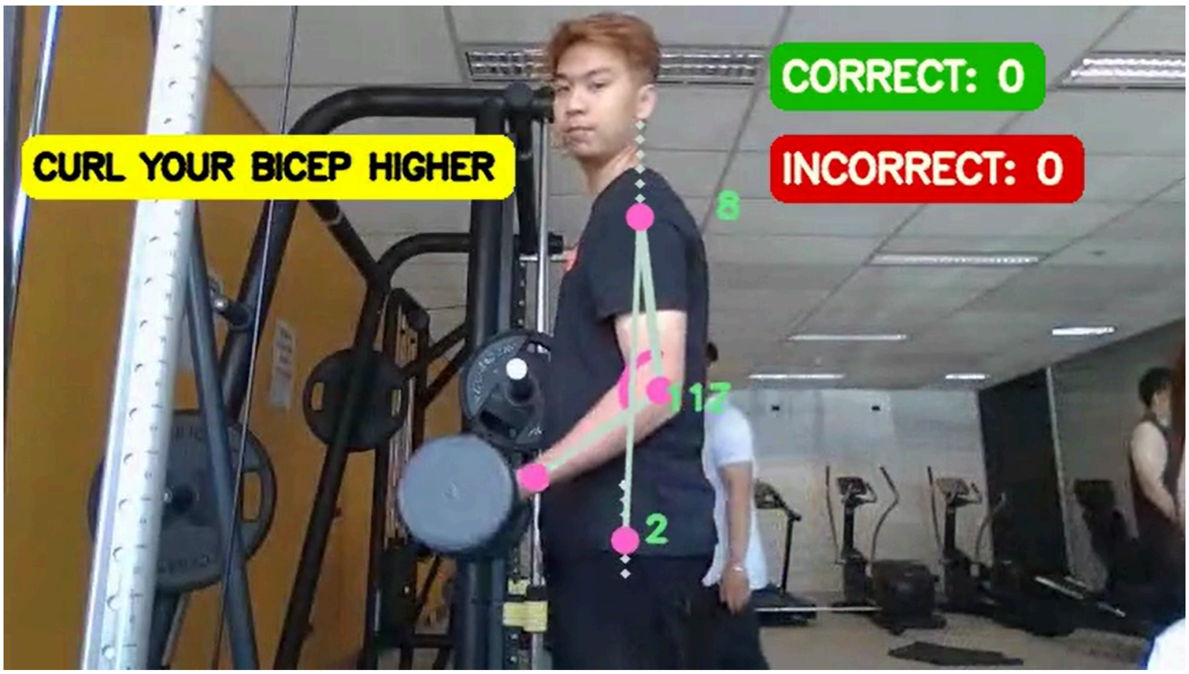
3. The application fails to detect any movements or changes in the user's posture for a period surpassing T seconds.

In each of these cases, the application initiates a reset of all relevant counters and parameters associated with tracking the user's performance during the bicep curl exercise.

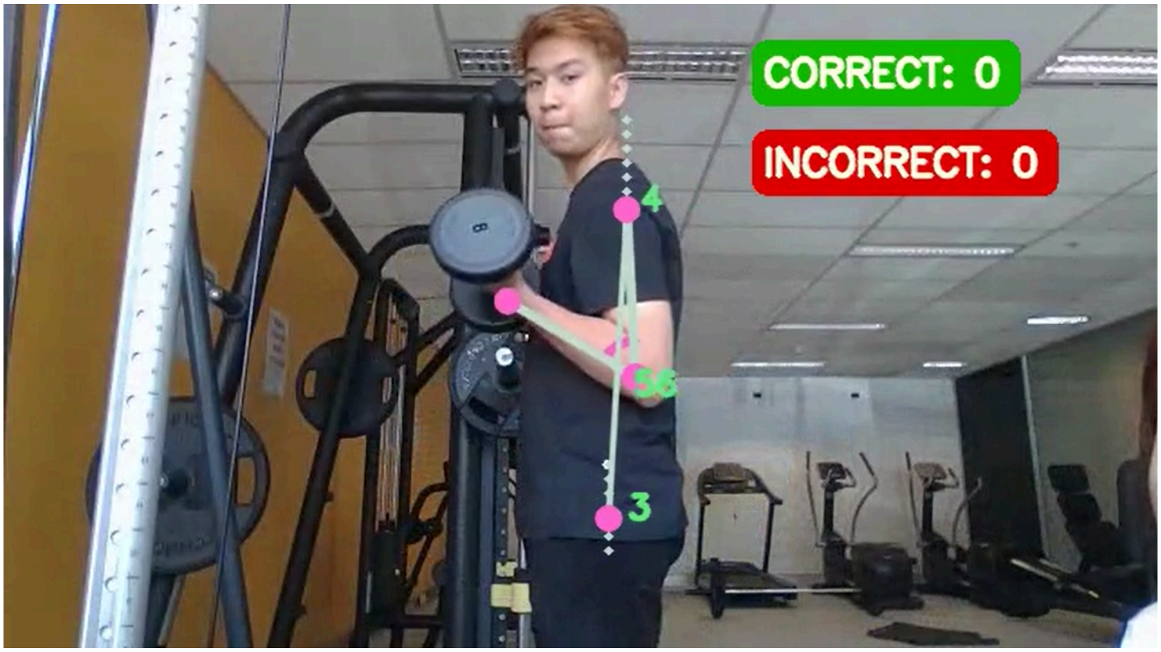
i. Test Cases

1. Correct posture

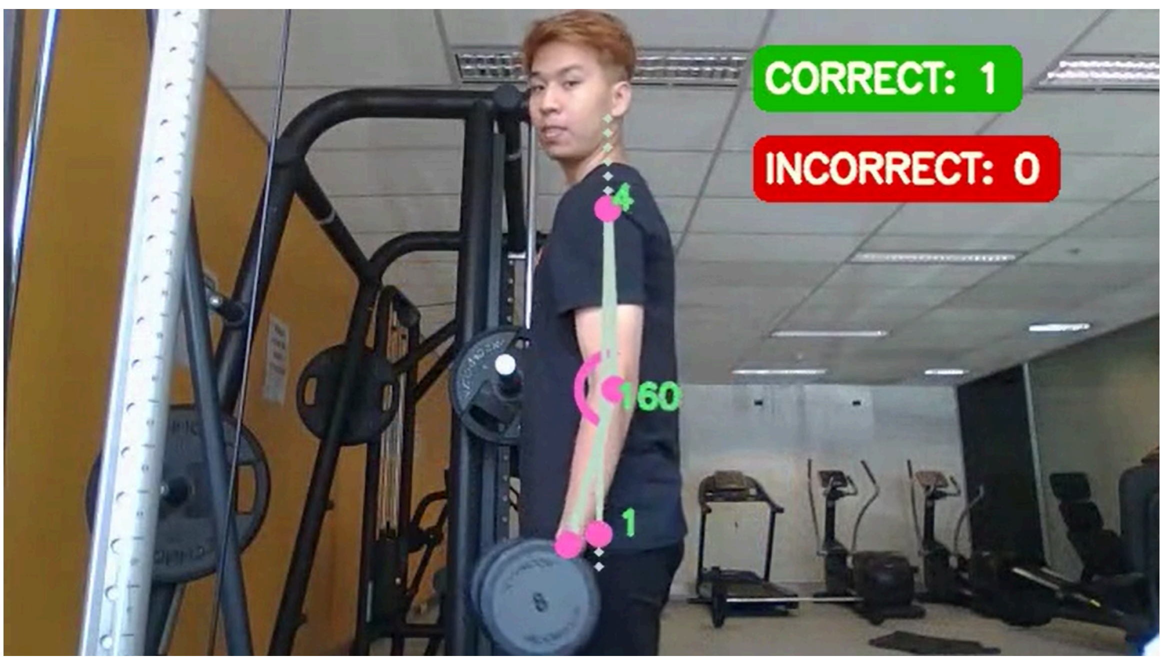
   From state 2 to state 3:



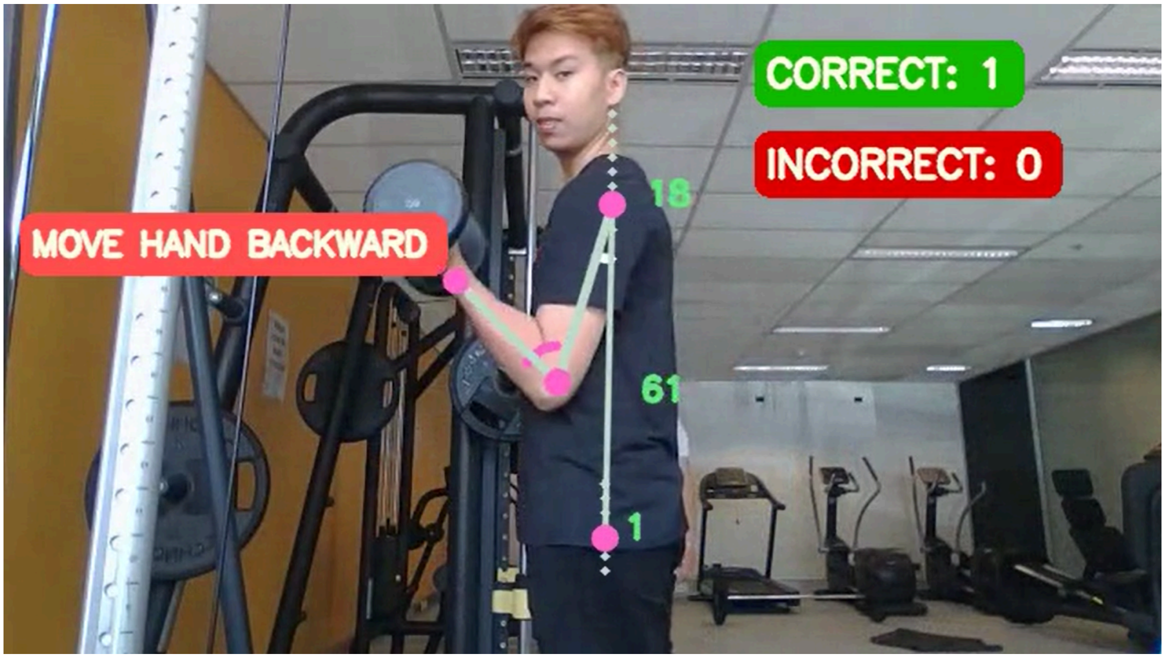
State 3:

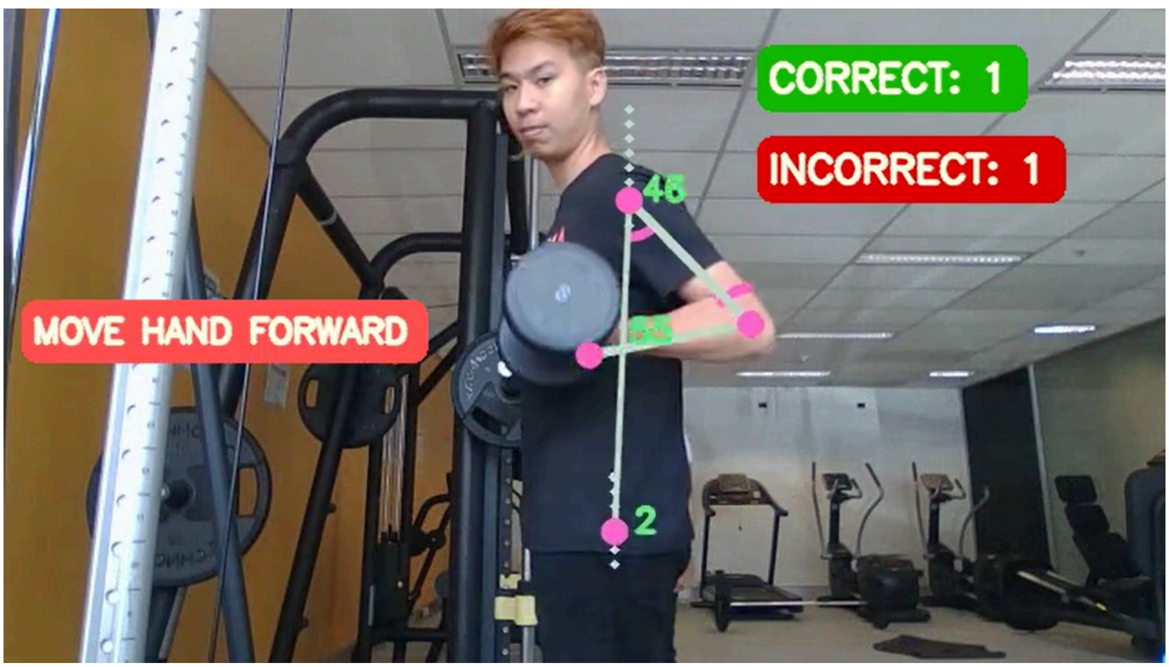


Back to state 2 -> One correct posture counted



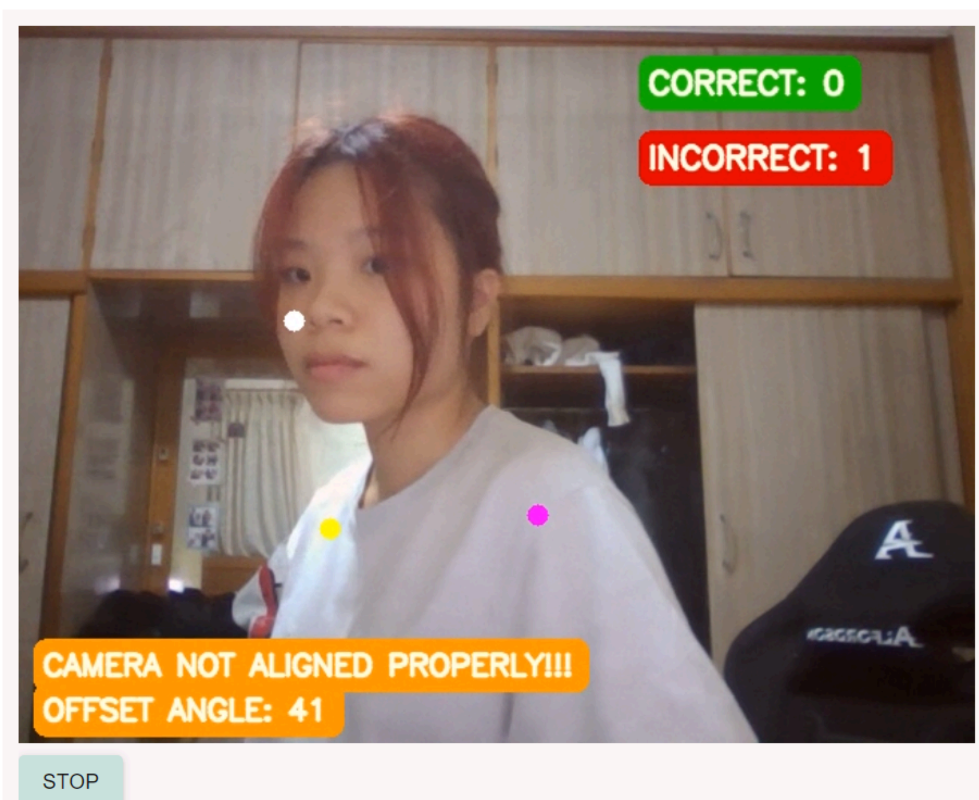
2. Incorrect posture





3. Incorrect camera view (frontal view warning)


CORRECT: 0
INCORRECT: 0
CAMERA NOT ALIGNED PROPERLY!!!
OFFSET ANGLE: 88
STOP


CORRECT: 0
INCORRECT: 1
CAMERA NOT ALIGNED PROPERLY!!!
OFFSET ANGLE: 41
STOP

-> Hence, in this case, in order to get the side view, the offset angle has to be 0
-> Otherwise, for the frontal view case, the offset angle will be around 90

j. Future Enhancements

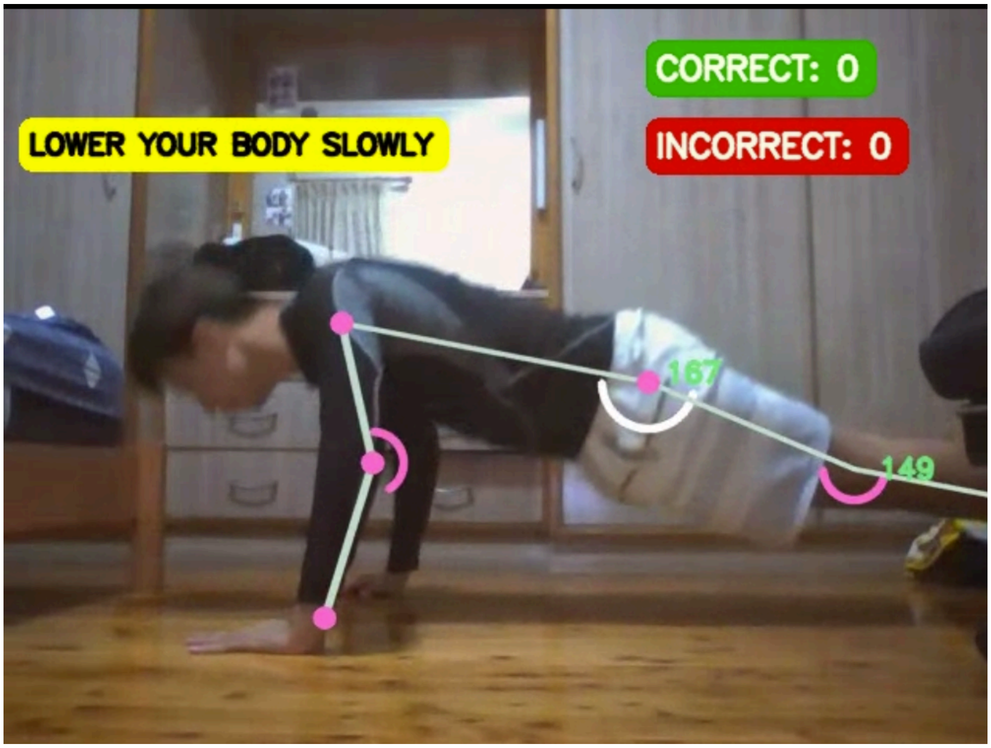
There are several areas where we can explore improvements to enhance the application's functionality and effectiveness generally, and in each exercise particularly:

- Enhanced Angle Tracking: Ensuring proper arm alignment and movement during bicep curls is crucial. Implementing advanced algorithms to track and analyze the alignment of the elbows and wrists can provide valuable feedback to users regarding their form.
- Multiple Camera Views: Incorporating multiple camera angles can provide a more comprehensive view of the user's posture and movements. This can enable more accurate analysis and feedback, especially for exercises where alignment and posture are critical, such as bicep curls.
- Integration of Advanced Techniques: We can explore integrating advanced techniques such as Human Action Recognition using Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) models. These models can analyze sequential data from the exercise movements to provide more nuanced feedback and insights.
- Wearable Sensors Integration: Utilizing wearable sensors like Inertial Measurement Units (IMUs) can offer additional data points for analysis. Integrating data from IMUs can enhance the application's capability to track movement dynamics and provide real-time feedback on exercise performance.

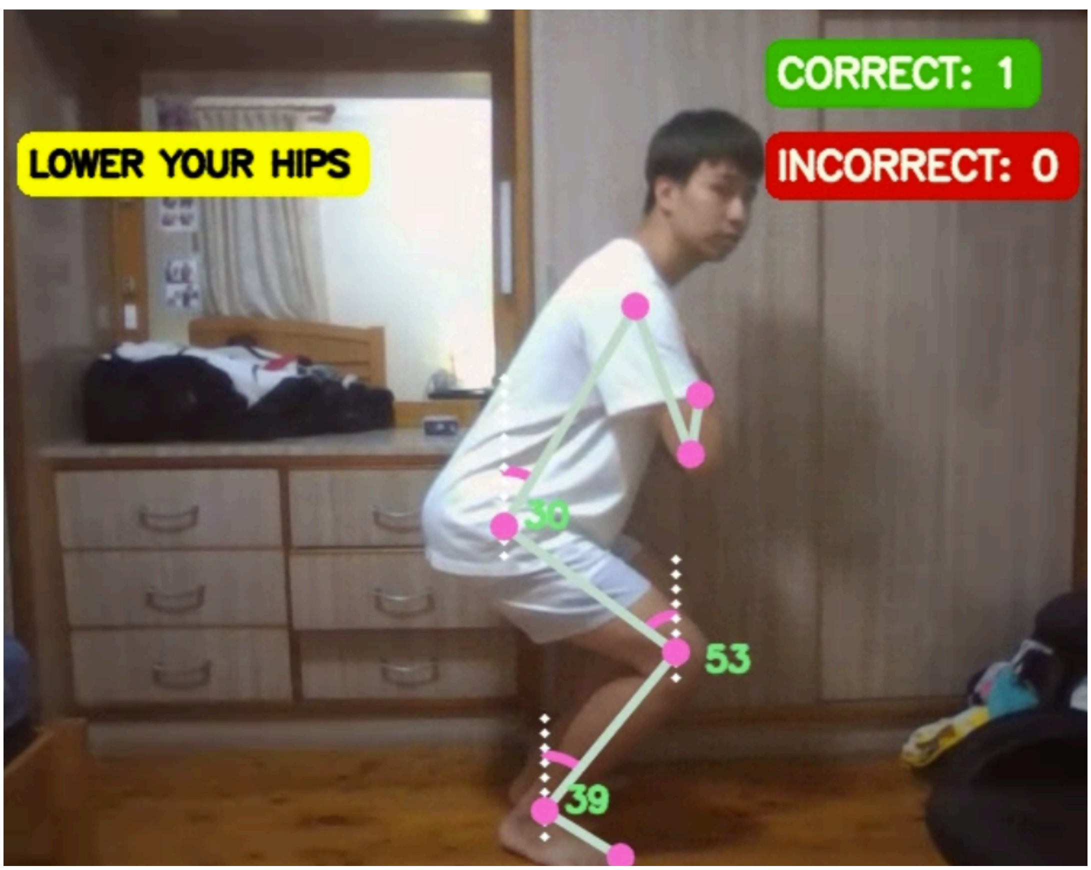
By incorporating these improvements and leveraging advanced technologies, we can enhance the functionality and effectiveness of our bicep curl exercise analysis application, providing users with valuable insights to improve their workout routines.

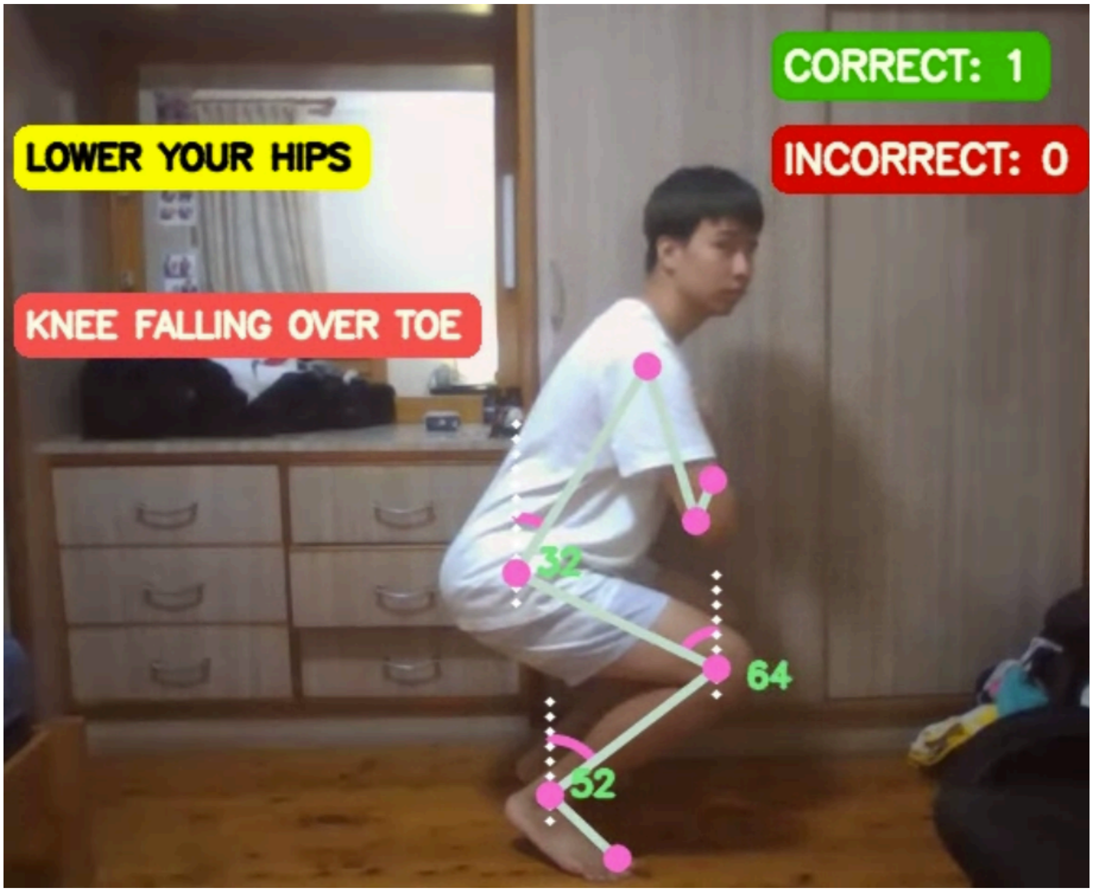
k. Brief overview of other exercises and code implementation

1. Push-ups


CORRECT: 0
LOWER YOUR BODY SLOWLY
INCORRECT: 0
167
149

2. Squat


CORRECT: 1
LOWER YOUR HIPS
INCORRECT: 0
30
53
39



B. PERSONALIZED EXERCISE PLANS GENERATOR

The "Personalized Exercise Plans Generator" feature harnesses the power of Large Language Models (LLM) to revolutionize the way users approach their fitness routines. By integrating cutting-edge Large Language Models (LLM) capabilities, this feature empowers users to input their personal details, fitness goals, and workout preferences through an interactive chat interface. Leveraging the sophisticated capabilities of LLMs, the application generates customized exercise plans tailored to each user's unique characteristics and objectives. This innovative approach not only simplifies the workout planning process but also ensures that users receive personalized guidance and support in achieving their fitness goals. Through the "Personalized Exercise Plans Generator," users can embark on a tailored fitness journey designed to maximize effectiveness and results.

This feature will provide users with two main functionalities: Firstly, a personalized weekly exercise plan tailored to target key muscle groups, incorporating specified sets, reps, and workout frequency based on individual preferences. Secondly, users will have access to a calendar view displaying their customized exercise regimen, facilitating easy visualization and scheduling of their workout routine.

The application leverages Large Language Model (LLM) techniques - a subset in the natural language processing (NLP) field and machine learning models to generate personalized workout routines tailored to individual fitness goals, experience levels, and preferences.

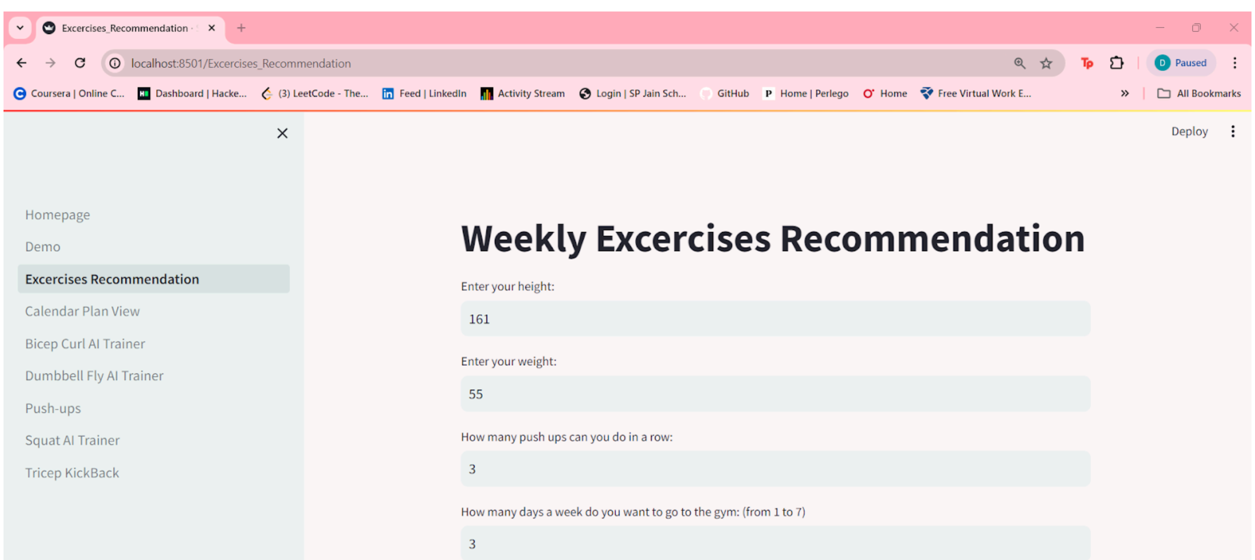
The code consists of several modules and libraries, including Streamlit, OpenCV, Langchain, and datetime. It begins with importing necessary libraries and setting up the Streamlit user interface. Users input their height, weight, fitness goals, workout preferences, and current fitness level. The application then interacts with an AI chatbot model to generate a customized weekly exercise plan in a calendar format.

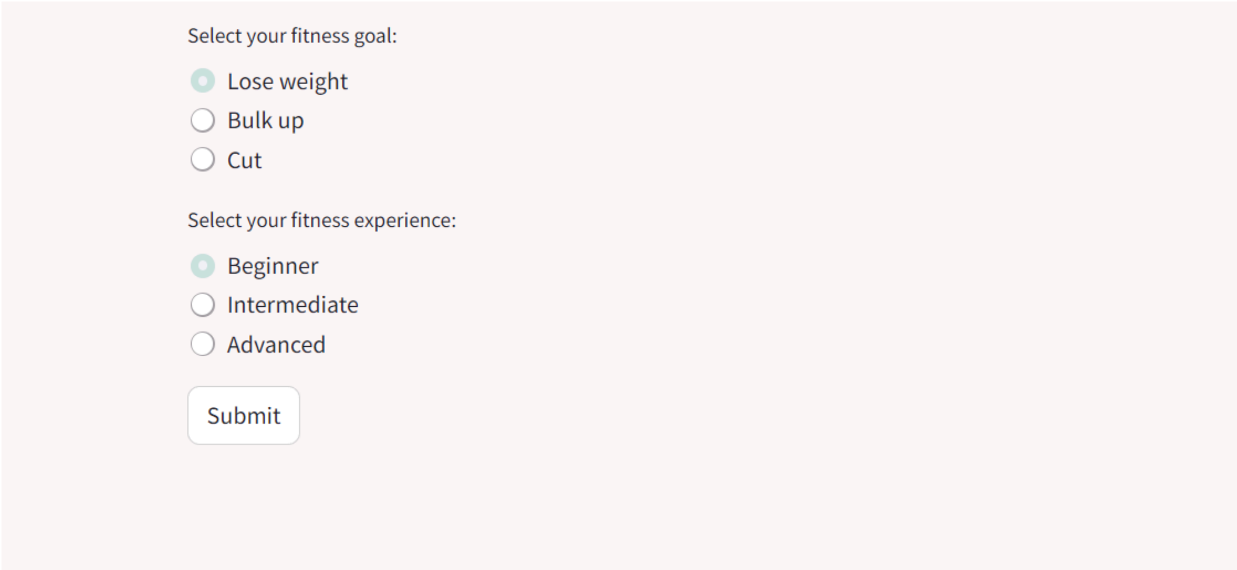
a. Workflow Description

1. User Input:

   Users provide their physical attributes (height, weight), strengths level, fitness goals, workout preferences (including the number of days per week they want to exercise), and experience level (categorized as beginner, intermediate, or advanced).

   Note that the user's strength level will be assessed based on their capacity to perform push-ups.

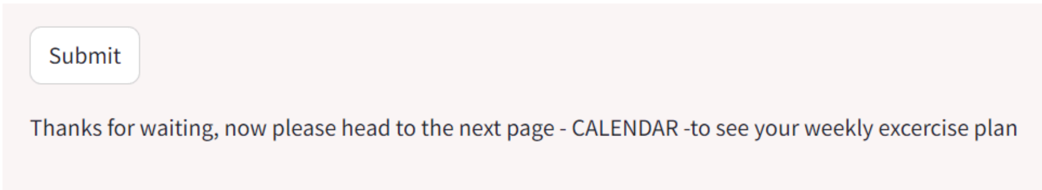


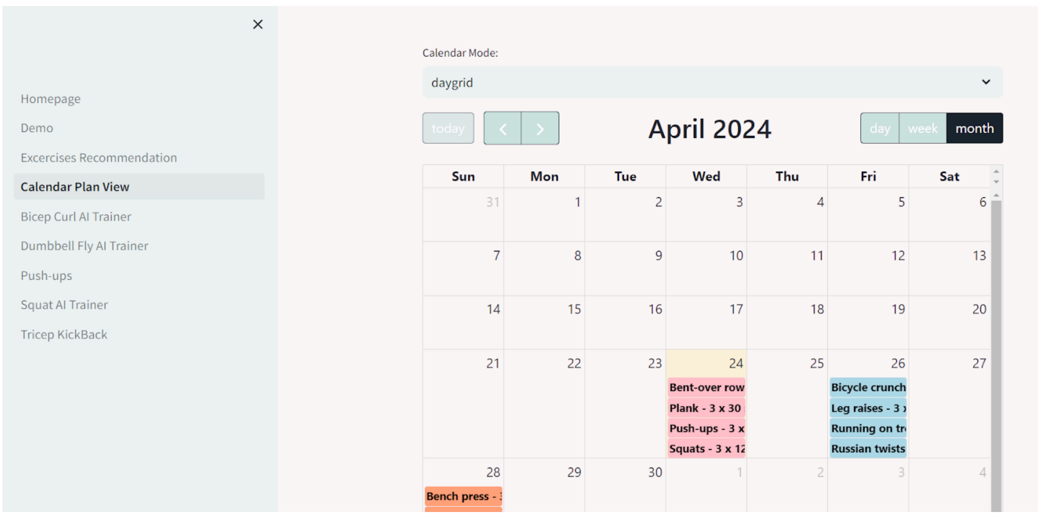


2. Chatbot Interaction:
    - Integration: The application seamlessly integrates with an AI-powered chatbot system.
    - User Input Processing: The chatbot system processes the user's input using advanced Large Language Model (LLM) algorithms.
    - Intent Recognition: It identifies the user's intent, understanding key elements such as fitness goals, workout preferences, and specific requests.
    - Contextual Understanding: The chatbot maintains context throughout the conversation, remembering previous interactions and user preferences.
    - Recommendation Generation: Based on the user's input and identified intent, the chatbot generates personalized exercise recommendations using a database of exercises, workout routines, and fitness guidelines.
    - Large Language Response: The chatbot formulates responses in natural language, providing clear explanations of the recommended exercises, sets, reps, and weights.
    - Iterative Interaction: Engaging in an iterative dialogue, the chatbot seeks clarification, feedback, and additional input from the user to refine its recommendations.
    - Feedback Incorporation: The chatbot incorporates user feedback into the conversation, adjusting its recommendations based on user interactions over time.

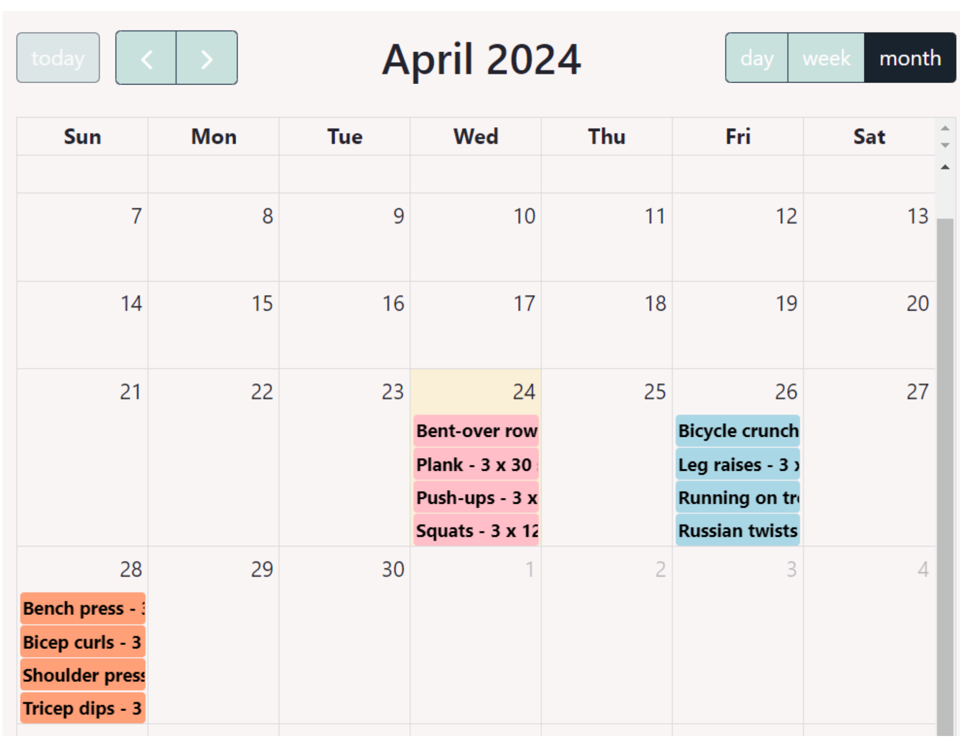
3. Plan Generation: Once the chatbot has gathered and processed the necessary information, it generates a detailed weekly exercise plan tailored to the user's goals, preferences, and

experience level. This plan includes exercise names, sets, reps, and recommended weights, and is formatted into a calendar view for easy visualization.





Calendar view:



List view:

b. Evaluation:

- User Feedback: The application's effectiveness and user satisfaction can be evaluated based on user feedback and engagement with the generated exercise plans.
- Usability: The ease of use, user interface design, and responsiveness of the application contribute to its overall usability and effectiveness.
- Accuracy: The accuracy of the exercise recommendations and their alignment with user preferences and goals are critical factors in evaluating the application's performance.

c. Future Enhancements

- Integration with Gym Posture Correction Data: Enhance the exercise plan generation process by incorporating insights and analysis gained from the gym posture correction feature. By leveraging data on users' posture and movement patterns, the application can tailor exercise recommendations to address specific areas of improvement, ensuring better alignment and effectiveness of the workout routines.
- Integration with User's Local Calendar: Enable users to seamlessly integrate their personalized exercise plans with their local calendar applications. This integration allows

users to receive notifications and reminders directly from their calendar, ensuring they stay on track with their fitness goals and exercise routines. Additionally, users can easily view their scheduled workouts alongside other commitments, facilitating better time management and adherence to their fitness plans.

- Progress Tracking and Analysis: Implement a progress tracking feature within the application, allowing users to monitor their fitness journey and track their progress over time. By including tick boxes or checkboxes for completed exercises, users can mark off completed workouts and visualize their progress. Furthermore, the application can analyze users' exercise history and performance data to provide insights into their progress, identify trends, and offer recommendations for continued improvement. This feature empowers users to stay motivated, set achievable goals, and make informed decisions to optimize their fitness outcomes.

C. FRONT-END OF THE WEB APP

a. Homepage.py Overview:

- Imports: The necessary libraries are imported, including Streamlit, Pandas, PIL, and OS.
- Initial State: The initial state of the session is defined, including variables to store dataframes, model information, evaluation metrics, and other flags.
- New Line Function: A utility function to add a specified number of newline characters.
- Logo: Displays the logo of the Gym Posture Correction App, providing visual branding.
- Description: A welcoming message and brief introduction to the app's purpose and functionality.
- Dataframe Selection: Provides guidance on getting started with the app, explaining the options for submitting data either through live streaming or file upload.
- Pages Folder: Mentions the existence of a folder named "pages" containing various subpages of the app, such as Homepage, Demo, Exercise Recommendation, etc.

b. Front-End Structure:

- Header: The app starts with a header displaying the logo and a brief description.
- Data Submission: Users are guided on how to submit their data, either through live streaming or file upload.
- Sidebar Navigation: Users can navigate between different pages of the app using the sidebar. Each page corresponds to a specific functionality or feature, such as Exercise Recommendation, Calendar Plan View, AI Trainers for various exercises, etc.

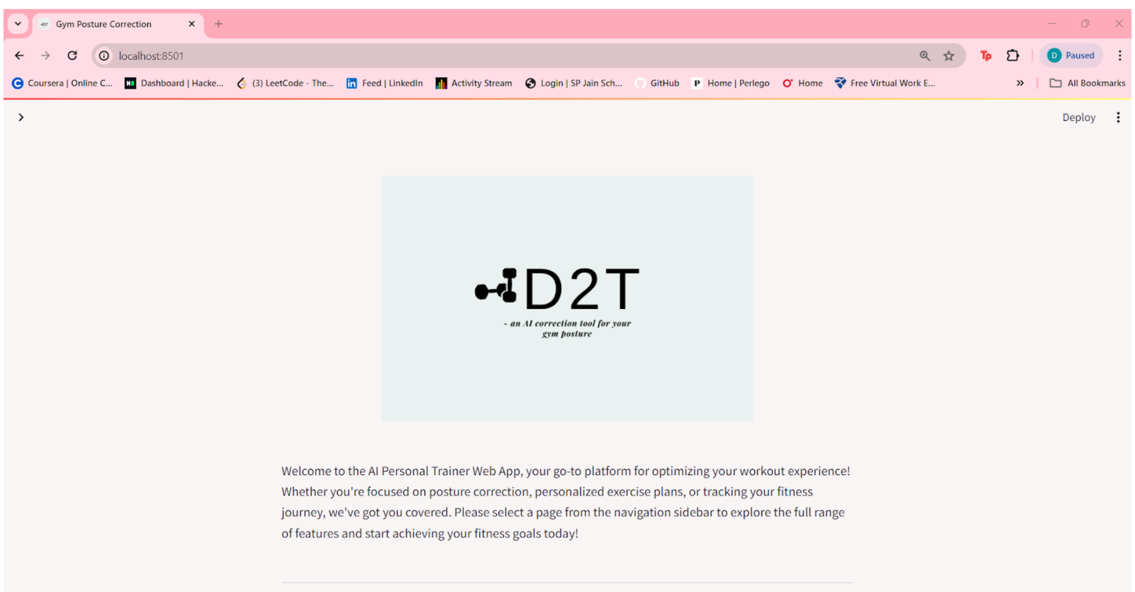
c. Streamlit Integration:

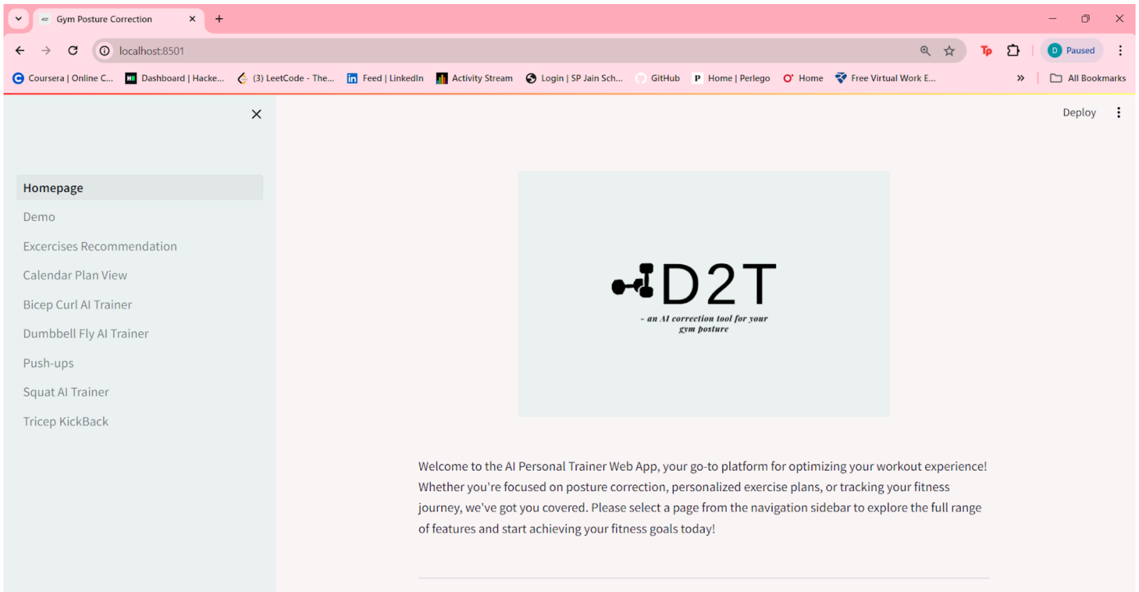
- The application is built using Streamlit, a Python library for creating web applications with minimal effort. Streamlit allows developers to create interactive web apps using familiar Python syntax, making it suitable for rapid prototyping and development.

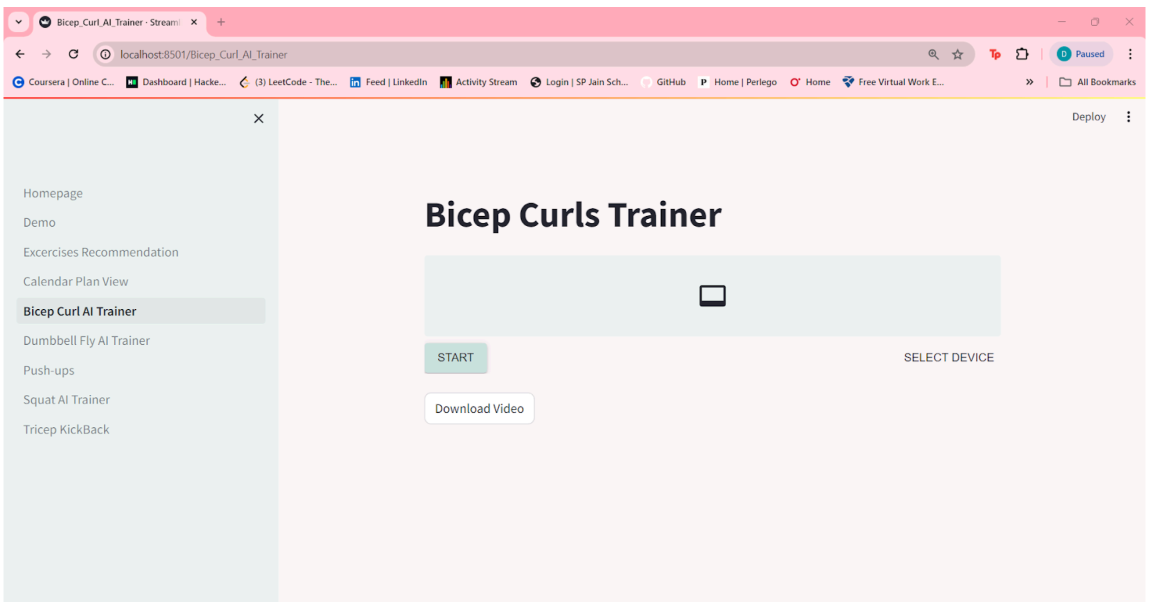
d. Running the App:

To run the app, one can execute the command: `streamlit run <path_to_homepage.py>` in the command prompt. This will launch the web app, displaying the homepage content along with the sidebar containing links to other pages.

Overall, the front-end of the Gym Posture Correction App provides a user-friendly interface for users to interact with the various features and functionalities offered by the application. It leverages the power of Streamlit to create an intuitive and responsive web app tailored to the needs of fitness enthusiasts and individuals looking to improve their gym posture.







V. RESULTS AND EVALUATION

The AI fitness application underwent a thorough evaluation to assess its effectiveness and user satisfaction.

Evaluation of the AI Fitness Application:

The application was evaluated based on its ability to accurately analyze and correct exercise postures, generate personalized exercise plans, and provide real-time feedback to users.

Performance metrics such as accuracy of posture analysis, responsiveness of feedback, and effectiveness of exercise plans were considered during the evaluation.

User Feedback and Satisfaction:

Feedback from users played a crucial role in evaluating the application's performance and user experience. Users provided insights into the application's usability, effectiveness in correcting postures, and overall satisfaction with personalized exercise plans. Their feedback helped identify areas for improvement and informed future enhancements to the application.

Overall, the evaluation process provided valuable insights into the strengths and weaknesses of the AI fitness application, guiding further development efforts to enhance user experience and satisfaction.

VI. FUTURE IMPROVEMENTS

Due to time and resource constraints, several areas of improvement and recommendations have been identified to enhance the user experience and functionality of the gym web application. These potential improvements include:

1. Enhancements to Gym Posture Correction:
    - Refinement of correction and tracking algorithms to provide more accurate feedback and analysis.
    - Integration of multi-camera views for comprehensive posture analysis from different angles.
2. User Activity Dashboard:
    - Development of a dashboard to display users' exercise activity and progress based on posture correction records.
    - Insights into improvement areas and achievements to motivate users and track their fitness journey.
3. Personalization Options for Weekly Plans:
    - Linking weekly exercise plans with results analysis from posture correction sessions.
    - Providing personalized recommendations based on users' exercise implementation count, common incorrect feedback, and other relevant metrics.
4. Database Integration and Data Security:
    - Implementation of a SQL database to securely store users' information, including personal details, exercise history, and progress tracking data.
    - Implementing robust data security measures to protect user privacy and sensitive information.
5. Enhance User Interfaces (UI):

    - Improvements to Streamlit web pages for enhanced user experience, including smoother navigation, faster loading times, and more intuitive interfaces.
    - Otherwise, consider other platforms with better offers related to the UI (such as React,...)
6. Mobile Version Development:
    - Creation of a mobile version of the application to cater to users who prefer to access the platform on their smartphones or tablets.
    - Optimization for various screen sizes and operating systems to ensure seamless functionality across devices.
7. Community Engagement Features:
    - Introduction of community engagement features, such as forums, challenges, and leaderboards.
    - Encouraging social interaction and collaboration among users to foster a sense of community, support, and accountability.
    - Facilitating peer-to-peer communication, knowledge sharing, and motivational support within the platform environment.
8. Continuous User Feedback and Iterative Development:
    - Establishing a feedback loop with users to gather insights, identify pain points, and prioritize feature enhancements.
    - Adopting an iterative development approach to regularly release updates and improvements based on user feedback and evolving needs.

VII. CONCLUSION

To sum up, this project has aimed to transform the fitness experience by leveraging technology to provide personalized guidance and support to users in achieving their fitness goals online. Throughout the development process, several key findings and outcomes have emerged, shaping the trajectory of the project and informing future directions.

Summary of Key Findings and Outcomes:

- The implementation of posture correction algorithms and real-time feedback mechanisms has proven effective in assisting users with maintaining proper form during exercises.
- The introduction of personalized exercise plans has enhanced user engagement and satisfaction by catering to individual preferences, goals, and fitness levels.
- The beginner-friendly user interface (UI) provides an easy-to-use platform for the users.

Reflection on the Project:

Throughout the project lifecycle, the team has encountered various challenges and obstacles, including team size, resource constraints, technical complexities, and evolving user requirements. However, these challenges have been met with resilience, creativity, and collaboration, ultimately contributing to the project's success.

The iterative development approach adopted has allowed for continuous refinement and improvement, ensuring that the final product meets the needs and expectations of its users.

Looking back, the project signifies a notable step forward in the intersection of fitness and technology, demonstrating the potential of AI-driven solutions to enhance health and wellness outcomes.

In conclusion, the project has achieved its objectives of developing a comprehensive and user-centric gym web application. Moving forward, there are ample opportunities for further innovation and enhancement, and the team remains committed to delivering an exceptional fitness experience for all users.

VIII. REFERENCES

1. Pose landmark detection guide. (n.d.). Google for Developers. https://developers.google.com/mediapipe/solutions/vision/pose_landmarker/
2. Dawn, K., & Dawn, K. (2024, January 9). AI Fitness Trainer - Build using MediaPipe for squat analysis. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow With Code, & Tutorials. https://learnopencv.com/ai-fitness-trainer-using-mediapipe/
3. CPT, A. M. W., & CPT, C. S. (2021, January 4). How to do Push-Ups to build Upper body strength. SELF. https://www.self.com/story/heres-exactly-how-to-do-a-push-up-correctly#:~:text=While%20a%2045%2Ddegree%20angle,feel%2C%20New%20York%20City%E2%80%93based
4. What is a large language model (LLM)? | Cloudflare. (n.d.). https://www.cloudflare.com/learning/ai/what-is-large-language-model/
5. Pandiyan, I. (2023, November 11). Demystifying GPT-3.5: A Deep Dive into OpenAI's LLM. Medium. https://medium.com/@indukishen/demystifying-gpt-3-5-a-deep-dive-into-openais-llm-5fe4a6977567