CHAPTER 1
INTRODUCTION

1.1 Introduction

In the realm of physical fitness, the correct execution of exercises is paramount for both efficacy and safety. An improper posture during gym and yoga sessions not only diminishes the potential benefits of the workout but also poses a risk of injury. Recognizing the importance of addressing this issue, our project endeavors to perfect the form of gym and yoga exercises through the integration of cutting-edge technology.

The foundation of our approach lies in the utilization of the Pose model from the Mediapipe library, a powerful tool capable of detecting the intricate landmarks or joints of the human body and the connections between them. These landmarks serve as crucial indicators of body parts such as hands, legs, elbows, shoulders, and hips. By accurately capturing the spatial relationships between these landmarks, we can calculate the angles formed by the connections, providing us with a comprehensive understanding of the user's body posture.

This project aims to develop software that focuses on assisting people in properly performing workout such as bicep curl and sit-up using Artificial Intelligence with Computer Vision technique. The goal of this project is to help prevent injuries and improve the quality of individual's workout by providing personalised feedback on their posture using only a computer and a webcam. Artificial Intelligence has always been discussed nowadays due to its ability to assist human with their daily activities. The capabilities of Artificial Intelligence giving humans hope to develop machines with human intelligence. One of the common Artificial Intelligence technologies is called Computer Vision, where machines can recognise photographs and videos in the same way that humans do.

Computer vision is a branch of Artificial Intelligence that aims to teach computers to see, identify, and interpret images in the same way that humans do, and then produce appropriate outputs. In other words, it combines a computer with human intelligence and sensibilities. Image classification, image localization, object detection, segmentation, and keypoint detection are some of the tasks included in computer vision.

1.2 Purpose

The primary purpose of our project is to develop a system that acts as a vigilant guide, assisting users in maintaining correct exercise postures. Aiming to enhance the overall exercise experience, our system employs a proactive approach by setting threshold values for key angles involved in various exercises. These threshold values are determined through a meticulous study of ideal exercise angles, ensuring that users receive immediate alerts when deviating from the prescribed form.

1.3 Aim

The overarching aim of our project is to contribute to the creation of a safer and more effective exercise environment. The specific objectives include:

- Implementing the Pose model to detect body landmarks and connections.
- Calculating relevant angles between body connections for various exercises.
- Establishing threshold values based on ideal exercise angles.
- Providing real-time alerts to users when their posture exceeds the defined thresholds.
- Empowering users to correct their form and reduce the risk of injuries.

1.4 Objective

The goal of this project is to develop an Artificial Intelligence-based system that assists people in performing home workouts such as bicep curl and sit-up by combining computer vision technology with human pose estimation technique, along with deep learning and machine learning approaches.

Following are the objectives of this project in order to achieve the mentioned goals:

- To investigate how computer vision can assist in detecting human exercise posture and incorrect posture.
- To develop an artificial intelligence-based software that uses camera to detect user's workout posture and provides personalized feedback on improving their exercise posture. 3
- To help preventing injuries and breaking the bad habits of exercising with incorrect posture.
- To evaluate the effectiveness of implementing this software in individual's health.

1.5 Scope

While our current focus is on perfecting exercise form through real-time feedback, the scope of our project extends beyond its immediate applications. In the future, we envision integrating an exercise and diet planner into the system, providing users with a comprehensive tool for optimizing their fitness routines.

As we delve into the subsequent sections, we will explore the technological and methodological aspects of our project, detailing the steps taken to transform this vision into a functional and user-friendly solution. Through this initiative, we aspire to contribute to the advancement of fitness technology and promote safer, more effective workout practices.

CHAPTER 2
LITERATURE SURVEY

The paper discusses the challenges of human pose estimation in computer vision, highlighting factors such as image scale, lighting changes, and cluttered backgrounds. It emphasizes the difficulty of automatically detecting a person's pose in an image due to various influencing factors. The paragraph references a method proposed by Yamakawa et al. from Waseda University in Tokyo, Japan, aimed at enhancing the accuracy of human pose estimation in videos. This method utilizes time series correlation to handle human pose as a set of time series data. Integration of a CNN-based model with a multiple-object tracking framework is employed to improve detection accuracy, interpolating undetected or incorrectly detected body joints using information from previous and following frames.

Author	Title	Year Of Publication	Method for Human Pose Estimation	Analysis/ Results
Mr.Kalyan D Bamane, Adarsh Bevore, Shashwat Upadhyay	Body Posture Guiding System	2020	. Deep Learning for Posture Analysis in Fall Detection	The work introduced in this paper is mainly focused on investigating a relatively low-cost and reliable fall detection approach for older adults based on computer vision techniques.
Ankita Rameshwar Mahajan, Vinod Agrawal	MOVEMENT DETECTION USING OPENCV	2022	Movement Detection using python	This Article can also be introduced as a better alternative to the expensive traditional security systems that take up much storage space and are not affordable for everyone.
Parag Tirpude, Sagar Sahu, Prof. Harshita Ragite	REAL TIME OBJECT DETECTION USING OPENCV-PYTHON	2022	ALGORITHMS FOR OBJECT DETECTION AND TRACKING	The Object Detection system in Images is web based application which mainly aims to detect the multiple objects from various types of images.

CHAPTER 3
PROBLEM STATEMENT

Develop an advanced Posture/Form Perfector System to address the prevalent issue of poor posture and form during everyday activities, leading to various health concerns. The system should encompass multiple major functionalities to provide real-time feedback, exercises, and recommendations, ultimately helping users maintain correct posture and form for improved well-being

CHAPTER 4
PROJECT REQUIREMENTS

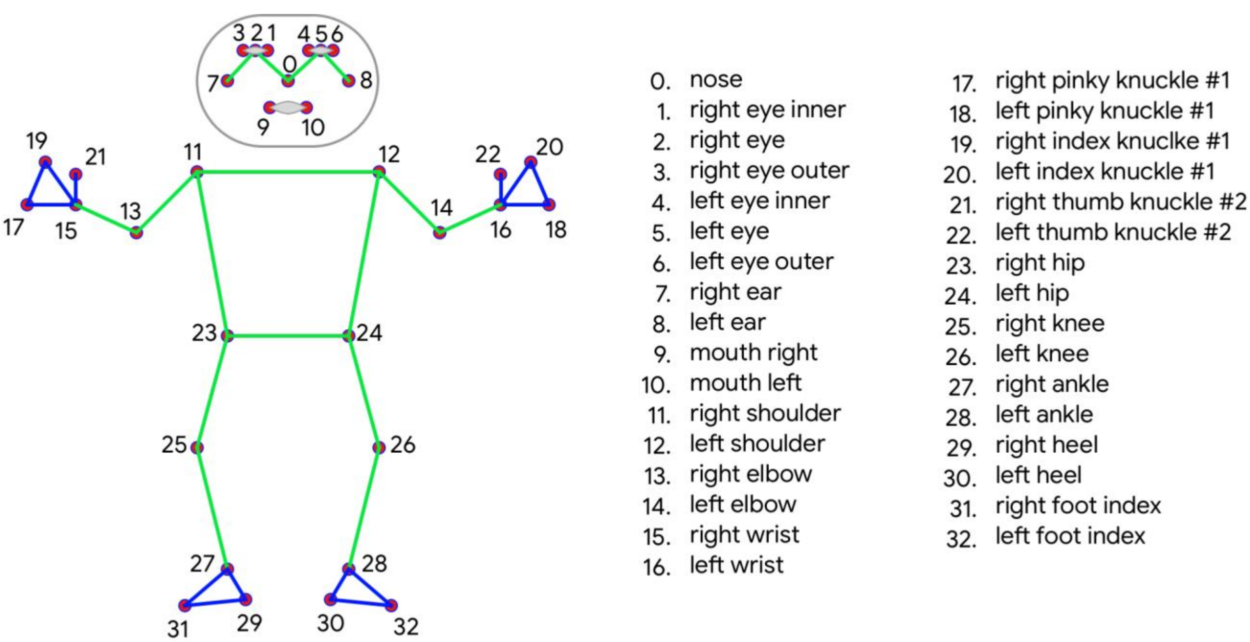
The project requirements are mentioned in the objectives section earlier, which is to investigate on how computer vision can assist in detecting human exercise posture and incorrect posture. To achieve this, the author must first achieve human pose estimation by using deep learning with CNN-based architecture as it is the common framework that specialized in image processing. The system should then be able to provide personalized feedback on their workout posture by detecting improper posture using various approaches. Machine learning and geometrical approaches are studied to find which approach is capable of achieving he author’s main goal. The selected approaches will be then implemented into a desktop application following the best practices of designing user interface for better user experience.

Mediapipe pose landmarker

The MediaPipe Pose Landmarker task lets you detect landmarks of human bodies in an image or video. You can use this task to identify key body locations, analyze posture, and categorize movements. This task uses machine learning (ML) models that work with single images or video. The task outputs body pose landmarks in image coordinates and in 3-dimensional world coordinates.

The pose landmarker model tracks 33 body landmark locations, representing the approximate location of the following body parts:

Diagram 4.1



Programming Language & Libraries

a. Python

Python is a programming language that was created by Guido van Rossum in 1991 and has a simple syntax similar to the English language.

Diagram 4.2

b. OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Diagram 4.3

c. MediaPipe

MediaPipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works on Desktop/Server, Android, iOS, and embedded devices like Raspberry Pi and Jetson Nano.

Diagram 4.4

d. NumPy

NumPy (pronounced NUM-py) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. We are mainly using this library to calculate the angle between connections.

Diagram 4.5

Software

a. Jupyter Notebook

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

Diagram 4.

Project Planner

Sr.No.	Dates		Project Task	Completion Date
	Month	Weeks		
1.	August	Week 0	Initial Discussion about 3 topics	1/08/2023
2.			Presentation of Respective topics to faculty panel	2/08/2023
3.		Week 1	Primary Discussion of Project Flow, Technologies, Use Cases	16/08/2023
4.			Selection of Final Topic and discussion about pros, cons, and future scope with project guide	29/08/2023
5.		Week 2	Problem statement Finalization	30/08/2023
	September		Submission of Problem Statement	1/09/2023
6.		Week 3	Literature Review/ Survey Discussion	5/09/2023
			Project Review 1	
7.		Week 4	Discussion Regarding Requirements	9/09/2023
8.		Week 5	Requirement Gathering and Analysis	11/09/2023
9.		Week 6	Prototype and Wireframing	15/09/2023
10.		Week 7	Tried Module 1 Implementation	17/08/2023
11.	October	Week 8-9	Initial Coding, made an functions required for our project.	22/08/2023
12.		Week 10	Project Review 2	1/08/2023
13.	Nov/Dec	Week 11-12	Front end prototype	14/11/2023

CHAPTER 5
SYSTEM DESIGN

Data Flow Diagram (DFD):
It's easy to understand the flow of data through systems with the right data flow diagram software.

DFD Level 0:
While higher-level diagrams provide a broader view of the system, lower-level diagrams offer more granular insights into individual processes. The level 0 data flow diagram, often referred to as the context diagram, is the highest-level diagram that provides an overall depiction of the system and its external entities.

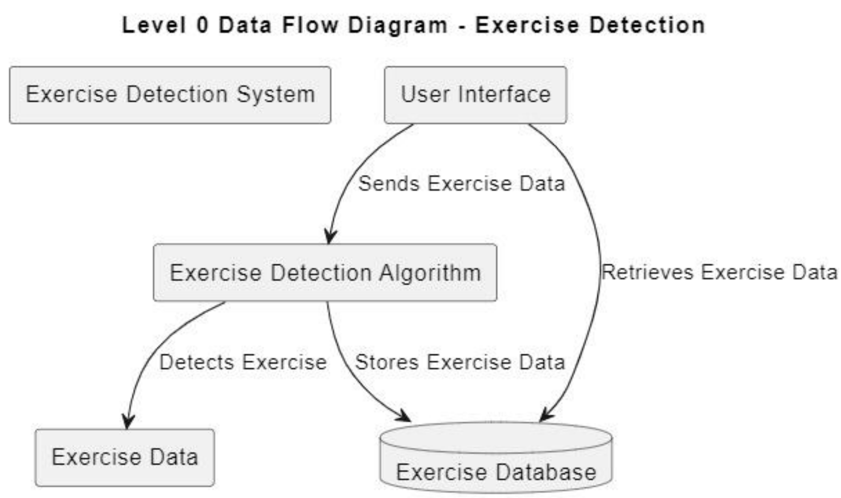


Diagram 5.1

DFD Level 1:
1-level DFD In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.

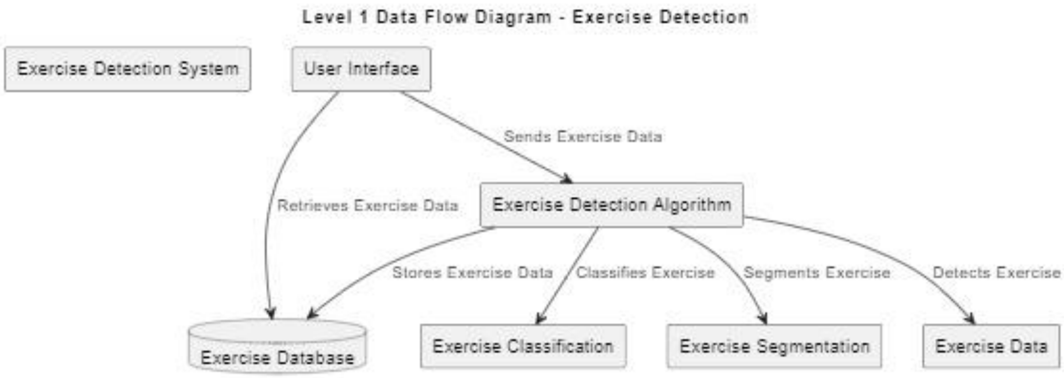


Diagram 5.2

DFD Level 2:
Deeper Dives (Level 2): the next level of DFDs provides even more detailed information by breaking down each Level 1 process into granular subprocesses. This level provides a deeper understanding of the system or process. It helps identify inefficiencies and areas for improvement.

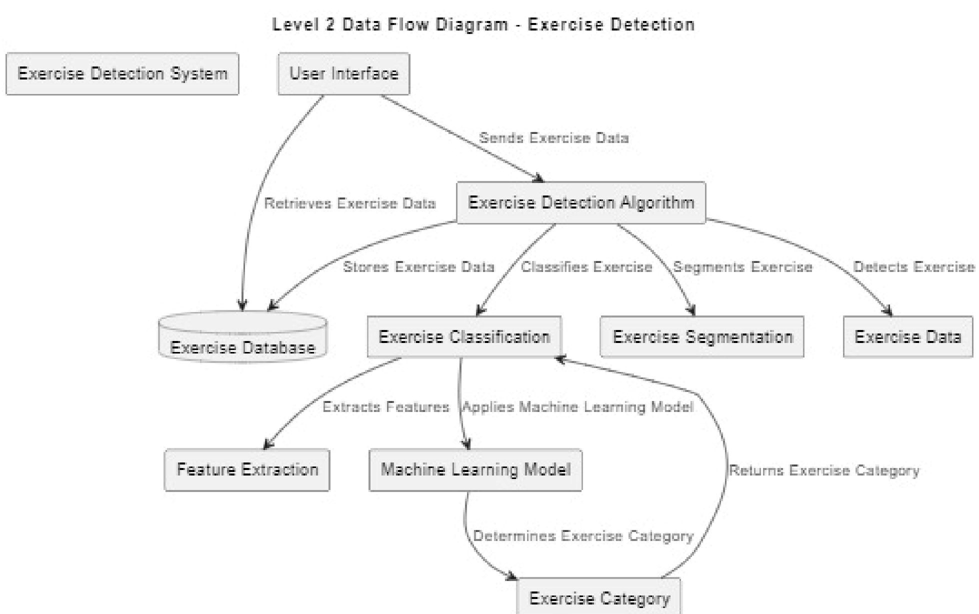


Diagram 5.3

1) CLASS UML DIAGRAM

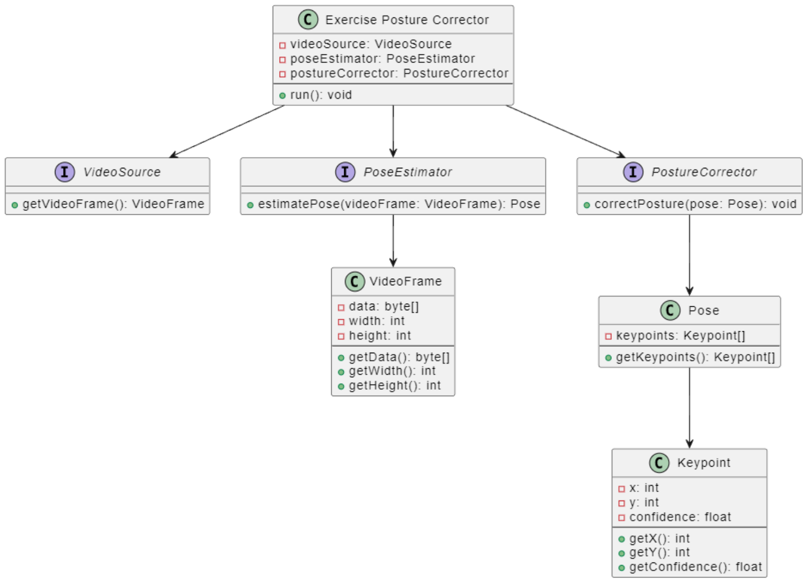


Diagram 5.4

2) ER DIAGRAM

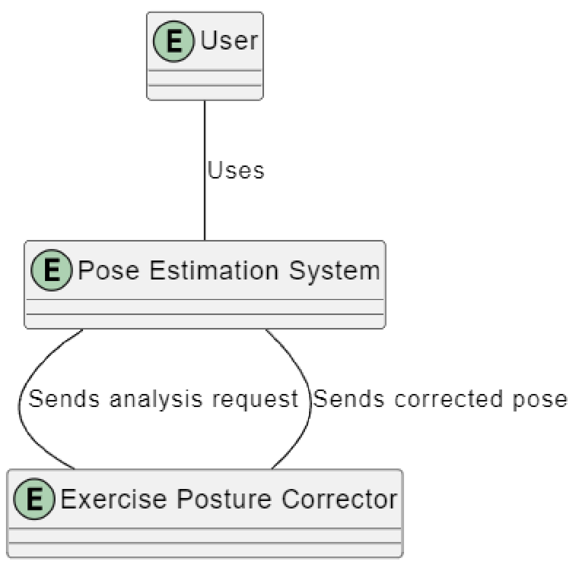


Diagram 5.5

3) SEQUENCE UML DIAGRAM

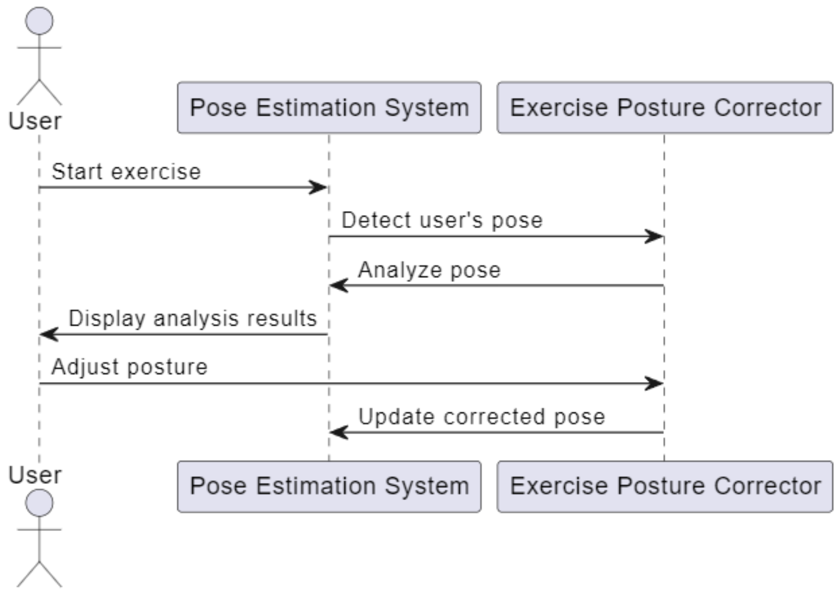


Diagram 5.6

4) STATE UML DIAGRAM

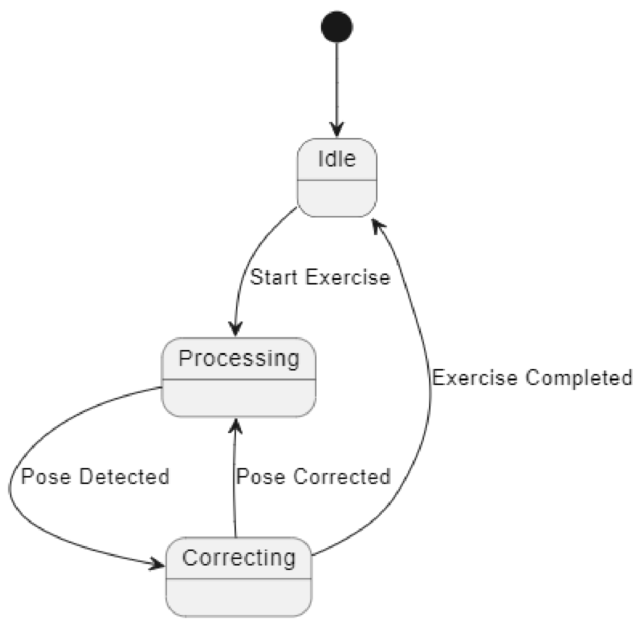
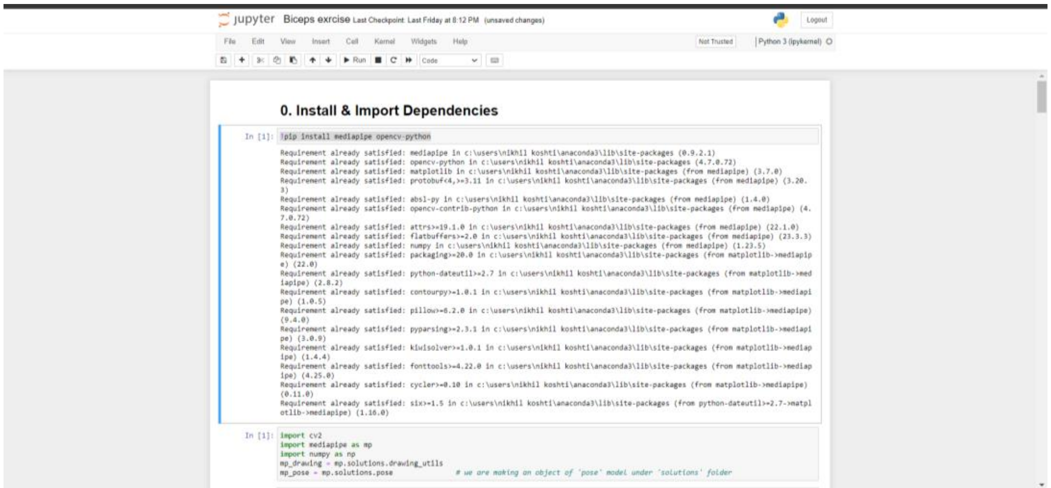


Diagram 5.7

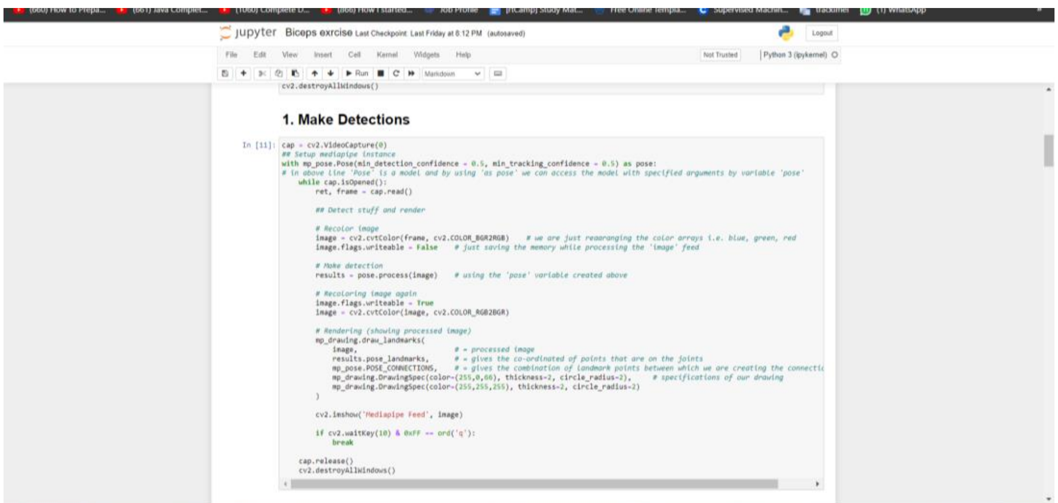
CHAPTER 6
PROJECT IMPLEMENTATION

In the first stage of our project implementation, we are going to access the webcam and will detect the joints and connections of user. Then we will apply our logic to analyze whether the user is doing the exercise in right way or wrong way. Following are the steps and screenshots of the respective code.

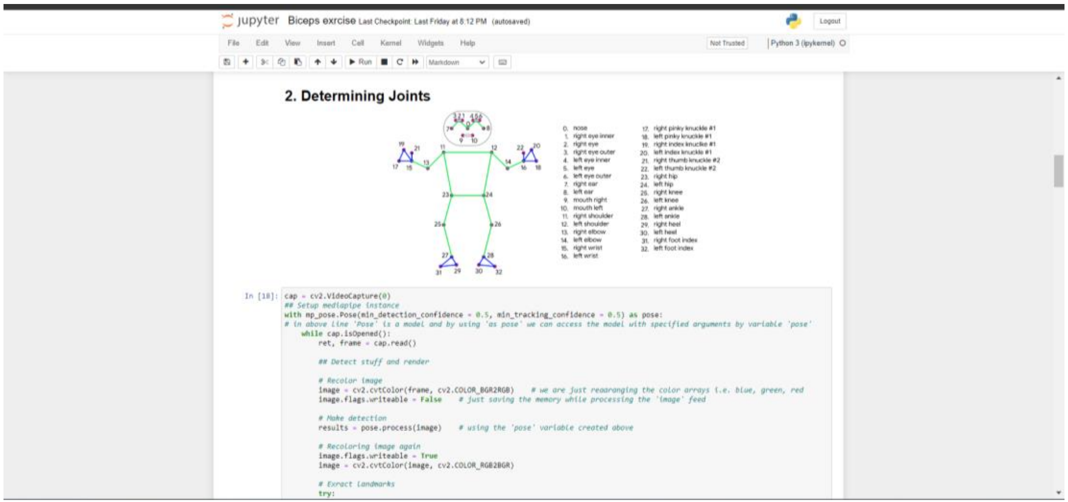
1. **Install & Import Dependencies:**
   We have installed the necessary libraries such as mediapipe and OpenCV as follows,



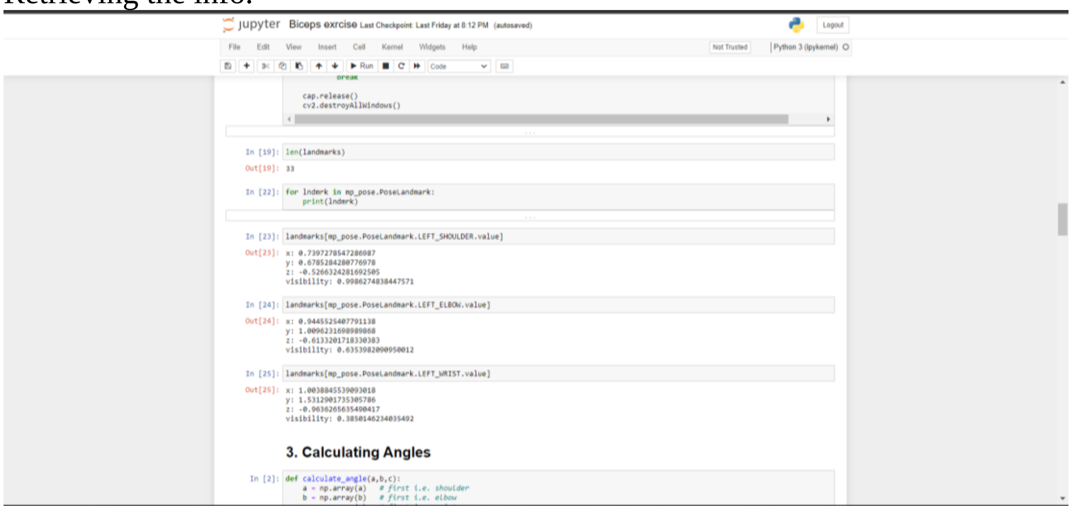
2. **Make Detections:**
   In this we have used solution.Pose model to detect the joints and body parts such as shoulder, elbow, leg, etc.



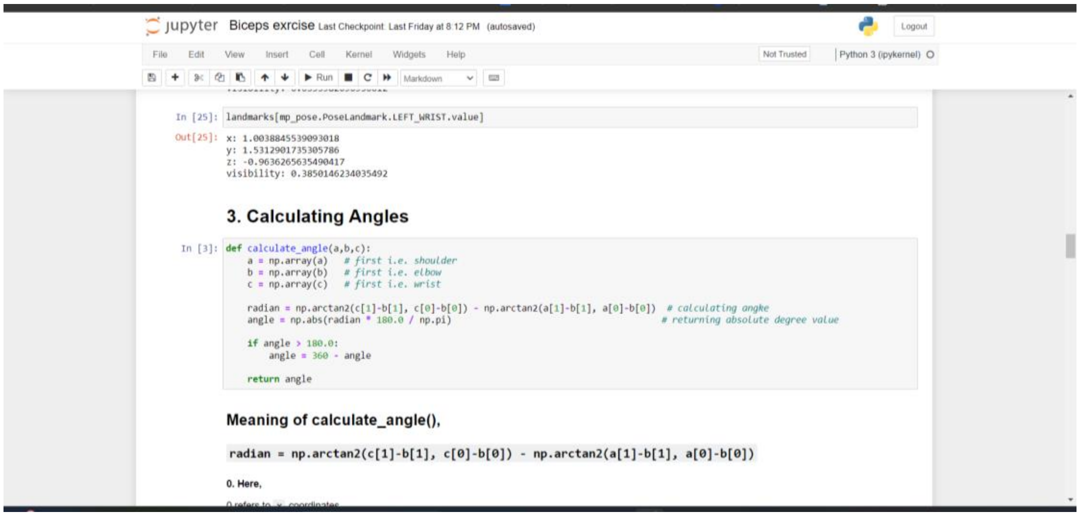
3. **Determining Joints:**
   This we studied the Pose model. Means there are how many detection landmarks are available, how many connection lines are available and how we can retrieve that information. The pose model diagram is shown in Chapter 4 also.



Retrieving the info.



4. **Calculating Angles:**
   In this I have written the calculate_angle() which takes the three point coordinates as parameters and returns the angle joining those point coordinates.



jupyter Biceps exrcise Last Checkpoint: Last Friday at 8:12 PM (autosaved)

In [25]:
```
landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value]
```

Out[25]:
```
x: 1.0038845539093018
y: 1.5312901735305786
z: -0.9636265635490417
visibility: 0.3850146234035492
```

3. Calculating Angles

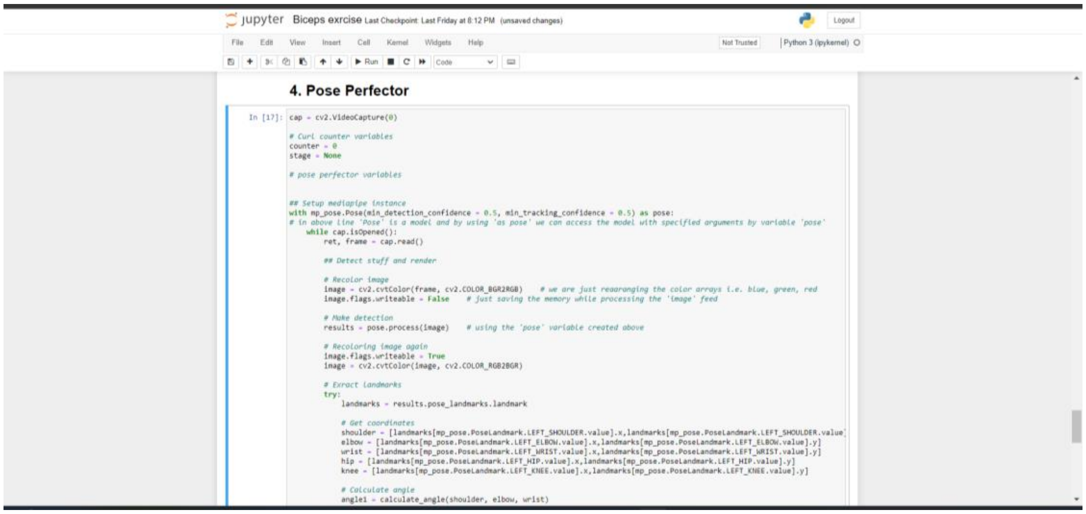
In [3]:
```
def calculate_angle(a,b,c):
    a = np.array(a)   # first i.e. shoulder
    b = np.array(b)   # first i.e. elbow
    c = np.array(c)   # first i.e. wrist

    radian = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])  # calculating angke
    angle = np.abs(radian * 180.0 / np.pi)                                         # returning absolute degree value

    if angle > 180.0:
        angle = 360 - angle

    return angle
```

Meaning of calculate_angle(),

```
radian = np.arctan2(c[1]-b[1], c[0]-b[0]) - np.arctan2(a[1]-b[1], a[0]-b[0])
```

0. Here,

5. **Pose Perfector Logic:**
   Let me explain it with an example.
   We have considered the exercise of biceps in which two body angles are important that are angle at elbow and hip. So, I have given the threshold values to these angles and if the user crosses these thresholds our system will give an alert that you are doing it wrong.

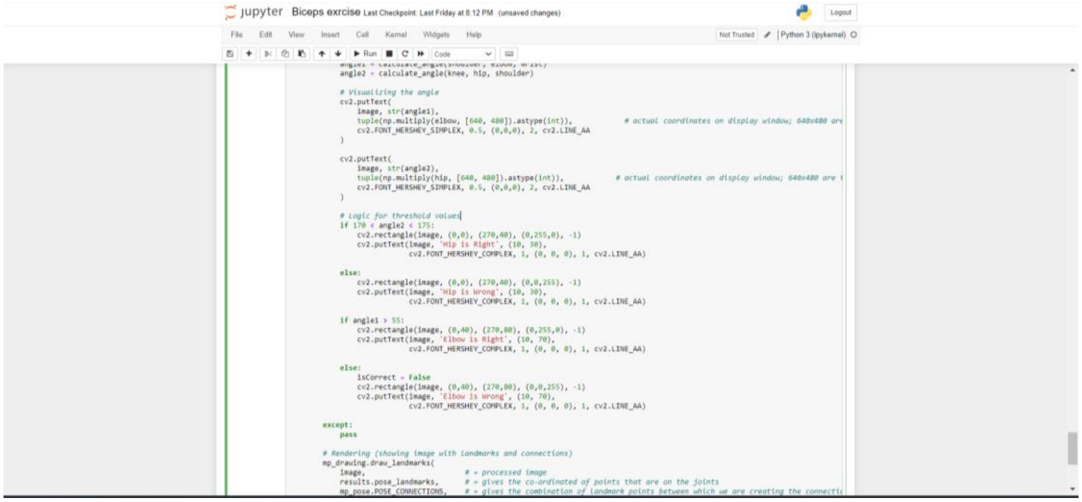


jupyter Biceps exrcise Last Checkpoint: Last Friday at 8:12 PM (unsaved changes)

4. Pose Perfector

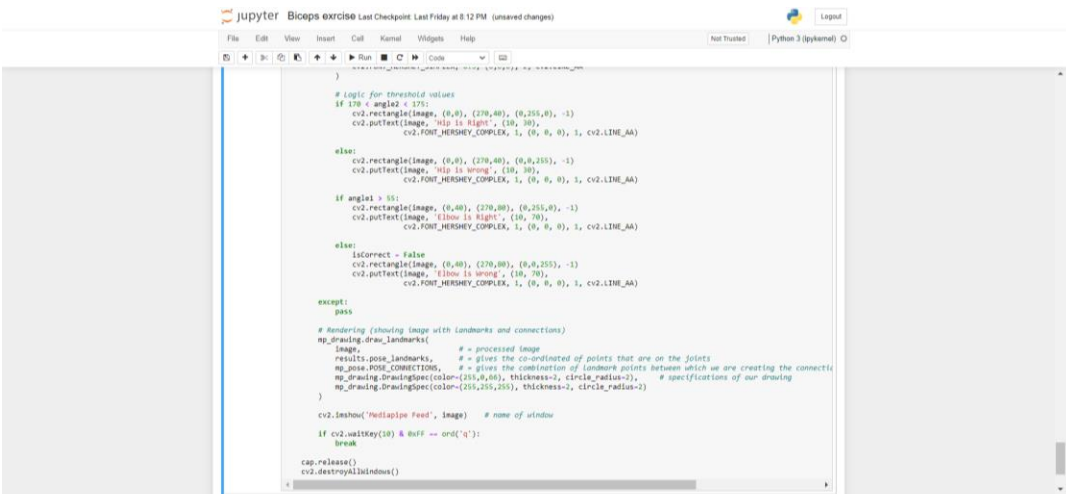
In [17]:
```
cap = cv2.VideoCapture(0)

# Curl counter variables
counter = 0
stage = None

# pose perfector variables


## Setup mediapipe instance
with mp_pose.Pose(min_detection_confidence = 0.5, min_tracking_confidence = 0.5) as pose:
# in above line 'Pose' is a model and by using 'as pose' we can access the model with specified arguments by variable 'pose'
    while cap.isOpened():
        ret, frame = cap.read()

        ## Detect stuff and render

        # Recolor image
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)    # we are just rearranging the color arrays i.e. blue, green, red
        image.flags.writeable = False    # just saving the memory while processing the 'image' feed

        # Make detection
        results = pose.process(image)    # using the 'pose' variable created above

        # Recoloring image again
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Exract landmarks
        try:
            landmarks = results.pose_landmarks.landmark

            # Get coordinates
            shoulder = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value
            elbow = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y]
            wrist = [landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].y]
            hip = [landmarks[mp_pose.PoseLandmark.LEFT_HIP.value].x,landmarks[mp_pose.PoseLandmark.LEFT_HIP.value].y]
            knee = [landmarks[mp_pose.PoseLandmark.LEFT_KNEE.value].x,landmarks[mp_pose.PoseLandmark.LEFT_KNEE.value].y]

            # Calculate angle
            angle1 = calculate_angle(shoulder, elbow, wrist)
```

Logic for thresholds:



Biceps exrcise Last Checkpoint: Last Friday at 8:12 PM (unsaved changes)

```python
        angle2 = calculate_angle(knee, hip, shoulder)

        # Visualizing the angle
        cv2.putText(
            image, str(angle1),
            tuple(np.multiply(elbow, [640, 480]).astype(int)),          # actual coordinates on display window; 640x480 are
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 2, cv2.LINE_AA
        )

        cv2.putText(
            image, str(angle2),
            tuple(np.multiply(hip, [640, 480]).astype(int)),          # actual coordinates on display window; 640x480 are t
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 2, cv2.LINE_AA
        )

        # Logic for threshold values
        if 170 < angle2 < 175:
            cv2.rectangle(image, (0,0), (270,40), (0,255,0), -1)
            cv2.putText(image, 'Hip is Right', (10, 30),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

        else:
            cv2.rectangle(image, (0,0), (270,40), (0,0,255), -1)
            cv2.putText(image, 'Hip is Wrong', (10, 30),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

        if angle1 > 55:
            cv2.rectangle(image, (0,40), (270,80), (0,255,0), -1)
            cv2.putText(image, 'Elbow is Right', (10, 70),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

        else:
            isCorrect = False
            cv2.rectangle(image, (0,40), (270,80), (0,0,255), -1)
            cv2.putText(image, 'Elbow is Wrong', (10, 70),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

    except:
        pass

    # Rendering (showing image with landmarks and connections)
    mp_drawing.draw_landmarks(
        image,                    # = processed image
        results.pose_landmarks,    # = gives the co-ordinated of points that are on the joints
        mp_pose.POSE_CONNECTIONS,   # = gives the combination of landmark points between which we are creating the connecti
```



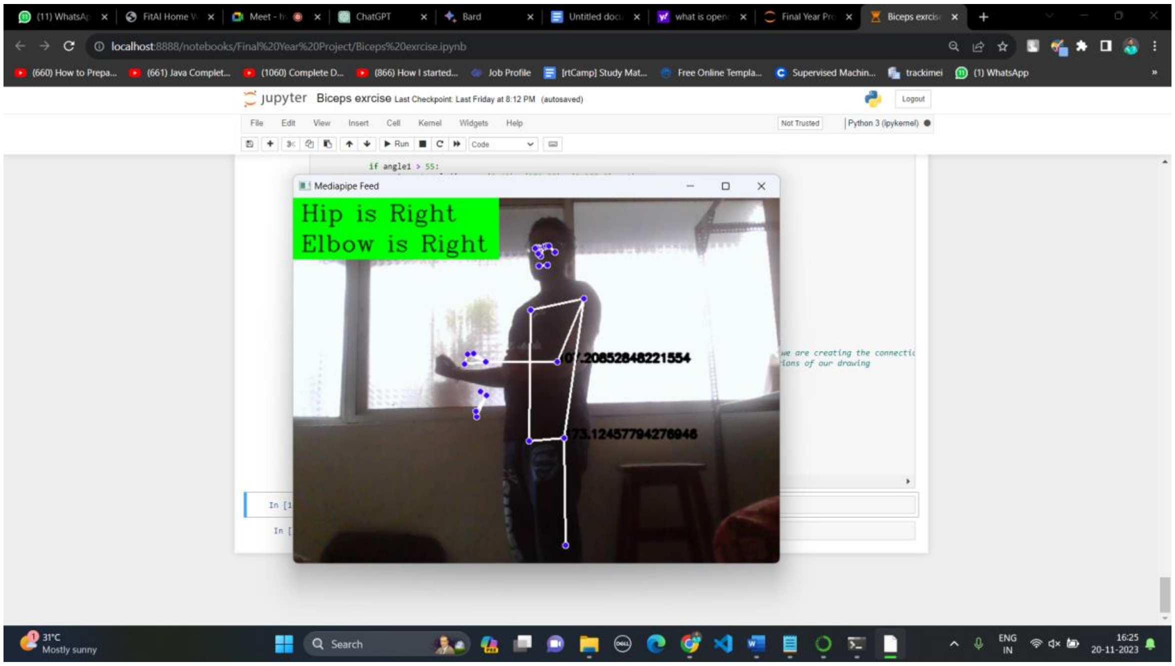
Biceps exrcise Last Checkpoint: Last Friday at 8:12 PM (unsaved changes)

```python
        )

        # Logic for threshold values
        if 170 < angle2 < 175:
            cv2.rectangle(image, (0,0), (270,40), (0,255,0), -1)
            cv2.putText(image, 'Hip is Right', (10, 30),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

        else:
            cv2.rectangle(image, (0,0), (270,40), (0,0,255), -1)
            cv2.putText(image, 'Hip is Wrong', (10, 30),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

        if angle1 > 55:
            cv2.rectangle(image, (0,40), (270,80), (0,255,0), -1)
            cv2.putText(image, 'Elbow is Right', (10, 70),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

        else:
            isCorrect = False
            cv2.rectangle(image, (0,40), (270,80), (0,0,255), -1)
            cv2.putText(image, 'Elbow is Wrong', (10, 70),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1, cv2.LINE_AA)

    except:
        pass

    # Rendering (showing image with landmarks and connections)
    mp_drawing.draw_landmarks(
        image,                    # = processed image
        results.pose_landmarks,    # = gives the co-ordinated of points that are on the joints
        mp_pose.POSE_CONNECTIONS,   # = gives the combination of landmark points between which we are creating the connecti
        mp_drawing.DrawingSpec(color=(255,0,66), thickness=2, circle_radius=2),    # specifications of our drawing
        mp_drawing.DrawingSpec(color=(255,255,255), thickness=2, circle_radius=2)
    )

    cv2.imshow('Mediapipe Feed', image)    # name of window

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

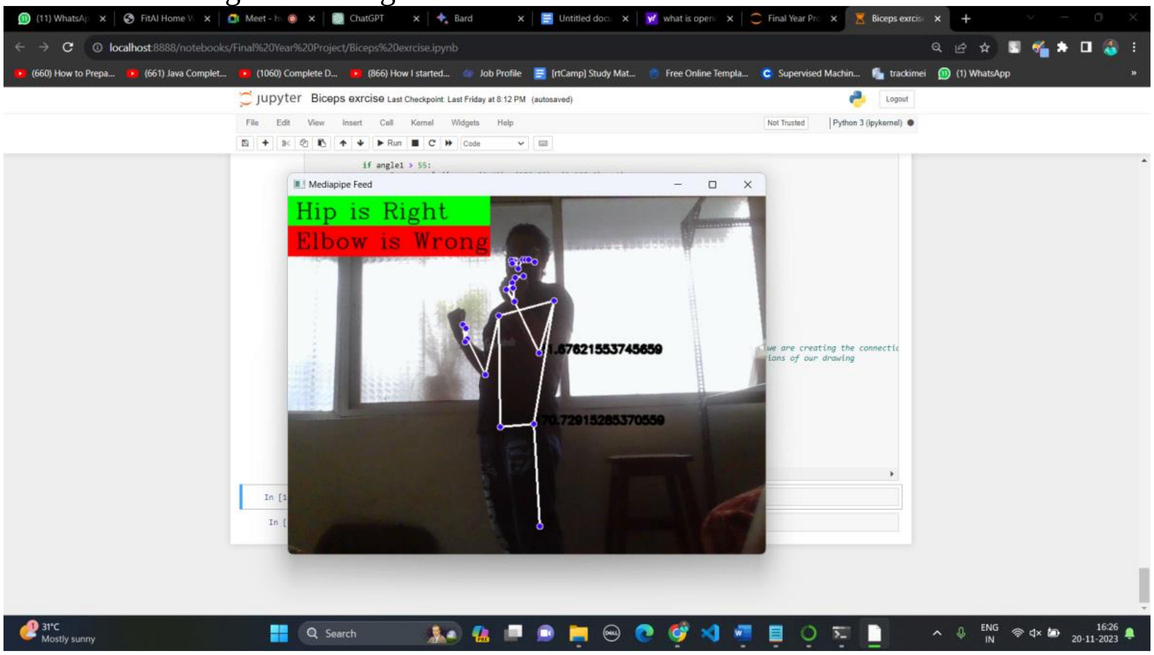
CHAPTER 7
RESULTS

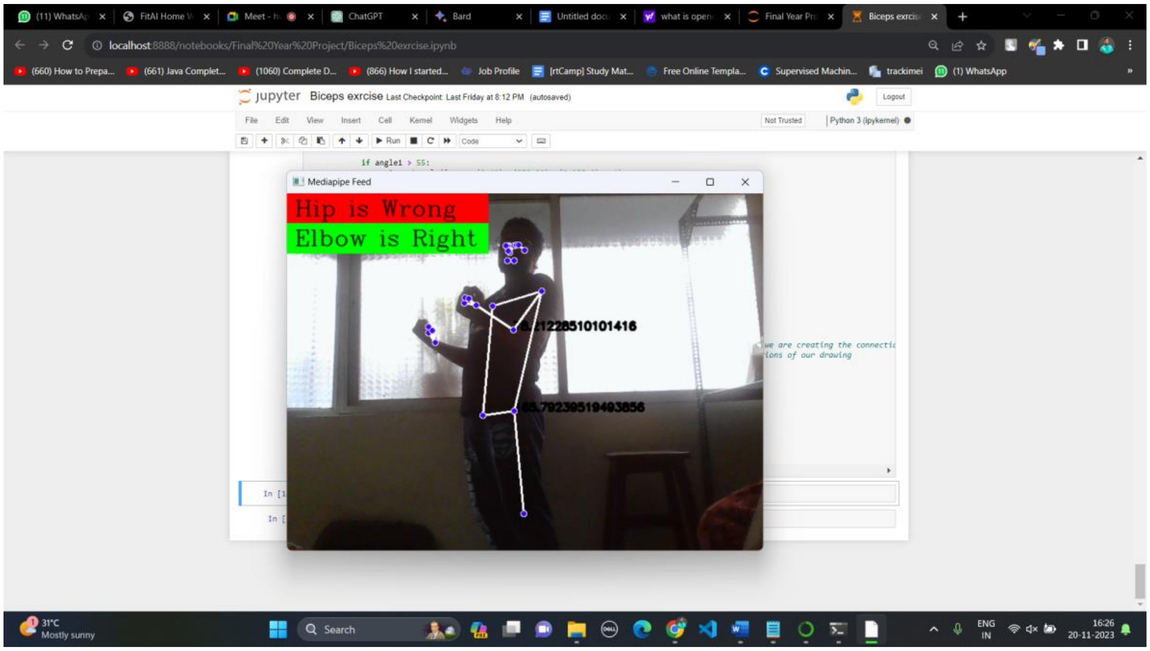
Here we are going to share some result screenshots in which we have shown we are going to give alerts and information about the angles on which user have to focus.
In our example of biceps exercise there are two angles elbow and hip. We do display the measure of angle to the user as shown in following diagrams.
In the section there are 4 sections as follows,

1. Both angles are right

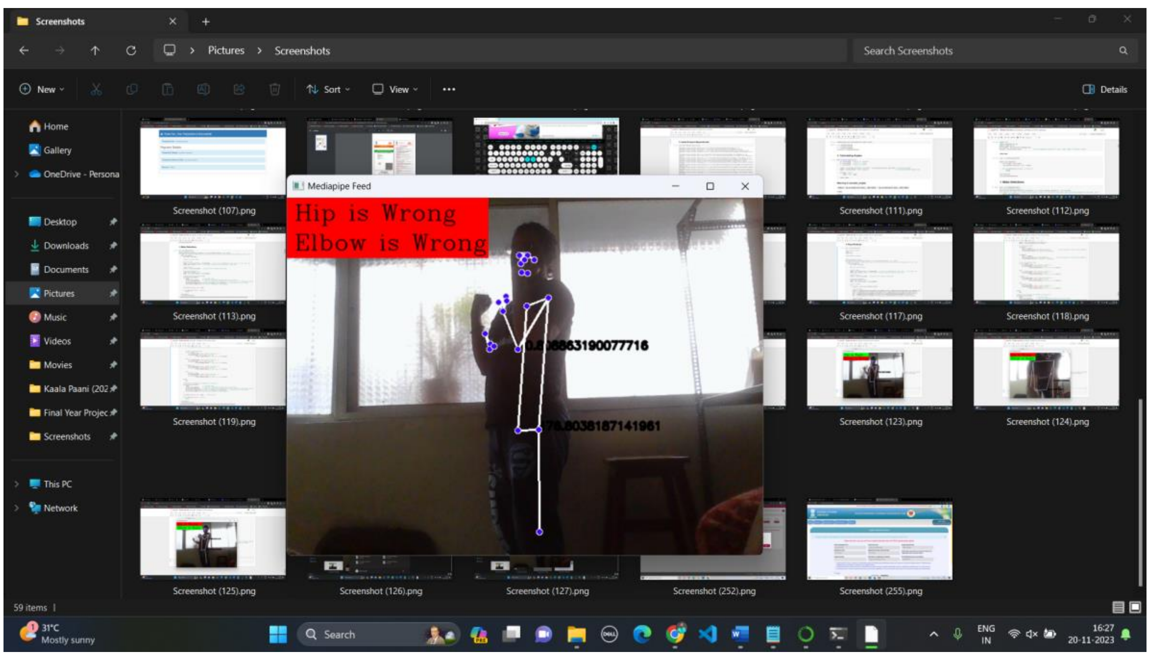


2. One of the angles is wrong





3. Both angles are wrong



CHAPTER 7
PLAGIARISM REPORT

Check Plagiarism

PLAGIARISM SCAN REPORT

Date	November 20, 2023
Exclude URL:	NO

Unique Content	**100**	Word Count	266
Plagiarized Content	**0**	Records Found	0

CONTENT CHECKED FOR PLAGIARISM:

In the realm of physical fitness, the correct execution of exercises is paramount for both efficacy and safety. An improper posture during gym and yoga sessions not only diminishes the potential benefits of the workout but also poses a risk of injury. Recognizing the importance of addressing this issue, our project endeavors to perfect the form of gym and yoga exercises through the integration of cutting-edge technology.

The foundation of our approach lies in the utilization of the Pose model from the Mediapipe library, a powerful tool capable of detecting the intricate landmarks or joints of the human body and the connections between them. These landmarks serve as crucial indicators of body parts such as hands, legs, elbows, shoulders, and hips. By accurately capturing the spatial relationships between these landmarks, we can calculate the angles formed by the connections, providing us with a comprehensive understanding of the user's body posture.

This project aims to develop software that focuses on assisting people in properly performing workout such as bicep curl and sit-up using Artificial Intelligence with Computer Vision technique. The goal of this project is to help prevent injuries and improve the quality of individual's workout by providing personalised feedback on their posture using only a computer and a webcam. Artificial Intelligence has always been discussed nowadays due to its ability to assist human with their daily activities. The capabilities of Artificial Intelligence giving humans hope to develop machines with human intelligence. One of the common Artificial Intelligence technologies is called Computer Vision, where machines can recognise photographs and videos in the same way that humans do.

MATCHED SOURCES:

Report Generated on **November 20, 2023** by https://www.check-plagiarism.com/ (https://www.check-plagiarism.com/)

Page 1 of 1