A MINI PROJECT REPORT

On

AI FITNESS TRAINEE

Submitted in partial fulfilment of the Requirements of the degree

Bachelor of Technology

In

Computer Science and Engineering

By

M.MADHU SUDHAN NAIK - O190399

B.V. CHAKRA PRIYA - O190228

J. THARUN - O190956

K.MAHITHA - O190949

K.BINDU - O190982

Under the guidance of

Mrs T.SIRISHA RANI M.Tech ,Assistant Professor

Dept of Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSIY OF KNOWLEDGE TECHNOLOGIES ONGOLE

2023-2024

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
DEPARTMENT OF COMPUTER SCEIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report entitled "AI FITNESS TRAINEE" Submitted by M Madhu Sudhan Naik (O190399), B V Chakra Priya(O190228), J Tharun(O190956), K Mahitha(O190949), K Bindu(O190982) partial fulfilment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering is a record of bonafide project work carried out under supervision during the academic year 2023-2024.

We are indebted to Mrs.T.Sirisha Rani,MTech, Assistant Professor,our project guide for conscientious guidance and encouragement to accomplish this project. We extremely thankful and pay our gratitude to Mr. Nandi Mallikarjuna, Asst. Professor Head of the Department CSE, for this valuable guidance and support on the completion of this project.

The report hasn't been submitted previously in part or full to this or any other university or institution for the award of any degree.

Mrs.T.Sirisha Rani ,M.Tech
Assistant Professor,
CSE Department,
RGUKT, ONGOLE.

Mr. N.Mallikarjuna ,
Head Of the Department,
CSE Department,
RGUKT, ONGOLE.

CERTIFICATE

This is to certify that the project entitled "**AI FITNESS TRAINEE"** being submitted by **M.MADHU SUDHAN NAIK** bearing the Id number **O190399** and **B.V CHAKRA PRIYA** bearing the Id number **O190228** and **J.THARUN** bearing the Id number **O190956** and **K.MAHITHA** bearing the Id number **O190949** and **K.BINDU** bearing the Id number **O190982** in partial fulfilment of the requirements for the award of the degree of the Bachelor of Technology in Computer Science and Engineering to Rajiv Gandhi University of knowledge Technologies is a record of bonafide work carried out by them under my guidance and supervision from during the academic year 2023-2024.

Project guide

Mrs.T.Sirisha Rani, M.Tech

Assistant professor,

Department of CSE,

RGUKT ONGOLE.

Head Of The Department

Mr.N.Mallikarjuna M.E

Assistant Professor,

Head of the Department of(I/C)

Department of CSE,

RGUKT ONGOLE.

APPROVAL

This report entitled “AI FITNESS TRAINEE” submitted by M.Madhu SudhanNaik(O190399),B.V Chakra Priya(O190228),J.Tharun(O190956),K.Mahitha(O190949),K.Bindu(O190982),Mrs.T.Sirisa Rani,M.tech ,Assistant professor RGUKT ONGOLE approval for degree of bachelor of technology in COMPUTER SCIENCE AND ENGINEERING.

Examiner (s)

_____

_____

_____

_____

Supervisor

_____

_____

Chairman

_____

Date: _____

Place: _____

ACKNOWLEDGEMENT

It is our privilege and pleasure to express a profound sense of respect, gratitude and in debtedness to our guide **Mrs T.Sirisha Rani** MTech ,Assistant Professor, Dept. of Computer Science and Engineering Rajiv Gandhi University Of Knowledge Technologies, for his/her indefatigable inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work. his dissertation work. We express our sincere gratitude to **Mr. N.Mallikarjuna** Assistant Professor & Head, Department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies, for his suggestions, motivations and co-operation for the successful completion of the work. We extend our sincere thanks to Dean **Mr. M. Rupas Kumar**, Dean Research and Development Rajiv Gandhi University Of Knowledge Technologies, for his encouragement and constant help .We extend our sincere thanks to **Dr.Bhaskar Patel** Director Rajiv Gandhi University Of Knowledge Technologies, for his encouragement.

M.MADHU SUDHAN NAIK - O190399

B V CHAKRA PRIYA - O190228

J.THARUN - O190956

K.MAHITHA - O190949

K.BINDU - O190982

DECLARATION

We hereby declare that the project work entitled " **AI FITNESS TRAINEE**" submitted to the **Rajiv Gandhi University Of Knowledge Technologies** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering is a record of an original work done by us under the guidance of **Mrs T. Sirisha Rani** MTech ,Assistant Professor and this project work have not been submitted to any other university for the award of any other degree.

SIGNATURE

M.MADHU SUDHAN NAIK - O190399	_____
B V.CHAKRA PRIYA - O190228	_____
J.THARUN - O190956	_____
K.MAHITHA - O190949	_____
K.BINDU - O190982	_____

Date: _____

Place: _____

ABSTRACT

Recently lot of people facing many health issues due to the lack of physical activity. Not getting enough physical activity can lead to heart disease, even for people who have no other risk factors. It can also increase the likelihood of developing other heart disease risk factors including obesity, high blood pressure, high blood cholesterol. Many people refrain from physical activities like workouts or yoga for various reasons like driving out to gym, trainer fees, social anxiety and hectic schedule.

Our objective is develop a solution which can guide the user on how to properly perform a particular exercise or a yoga asana. With the help of OpenCV library and Media pipe framework, we can develop an application that detects the user's exercise pose and provides personalized, detailed recommendations on how the user can improve their form.

The project aim is to validate and be able to quantify the physical movements of the end user. Mediapipe approach provides human pose tracking by employing machine learning (ML) to infer 33, 2D landmarks of a body from a single frame. With the help of this data, we can evaluate whether the user is performing task in a proper form and notice the montheir shortcoming The
practical application would require a webcam pointed towards the person in such a way that the face and the portion of the body associated with the task is clearly visible and can be marked with Mediapipe landmarks and later the data can be processed to guide them.

The project focuses on helping you exercise, by determining the quality and quantity of repetitions which is done by using pose estimation.

INDEX

1.INTRODUCTION

Exercising is most essential for proper health and fitness moreover, it is essential for every sphere of life. It’s important to remember that we have evolved from nomadic ancestors who spent all their time moving around in search of food and shelter, travelling large distances on a daily basis. Our bodies are designed and have evolved to be regularly active. Over time, people may come across problems if they sit down all day at a desk or in front of the TV and minimize the amount of exercise they do. Exercise is a bodily movement performed in order to develop or maintain physical fitness and good health overall. Here are some of the benefits of exercise:

1) Help you control your weight

2) Reduce your risk of heart diseases

3) Help your body manage blood sugar and insulin levels

4) Improve your mental health and mood

5) Strengthen your bones and muscles

6) Reduce your risk of cancer

7) Improve your sleep

According to the survey done by India news, it says that around 60% of people do not do regular exercise because they don't have that much time, it can be because of their family, work, and so on.

We all know that exercise is a very crucial part of our life and it is very to exercise with correct form. Sometimes we forget to keep a track on the number of reps.

1.1 Motivation

Al-powered personal training is an exciting i nnovation that combines the best of artificial intelligence and fitness. With ai trainee technology, a program can provide individualized feedback and guidance for every exercise and activity.

1.2 Problem Definition

A rtificial Intelligence (AI) is the ability of machines to learn and perform tasks. It has the potential to automate and supplement traditional fitness regimes, providing data-driven insights and adaptive personal training experiences.

This technology has already been employed for both commercial purpose, such as with fitness centres or at home for personal use. As al continue to evolves , so will its application in the fitness industry, changing how people train and move towards improved health.

1.3. Objective of the Project:

This technology has already been employed for both commercial purpose, such as with fitness centres or at home for personal use. As al continue to evolves , so will its application in the fitness industry, changing how people train and move towards improved health.AI can be used to track one’s performance, lifestyle, and diet; this information can then be used to give tailored advice and guidance on diet, sleep, exercise regimen, energy level, and more. AI is quietly revolutionizing the way people approach their health and well-being by providing real-time feedback systems designed to help habit formation.

2 . REQUIREMENT ANALYSIS

2.1 Existing System

There are numerous applications available in the market which guide the user about the exercises to be performed. All of the applications and services of fitness available to the public on play store and app store are focusing on describing the exercise routine to be followed and the number of repetitions for which it should be performed and the time duration in which you need to complete.

2.2 Scope of the Existing System

Scope of the Existing System The scope of the existing system is very limited. All of these application are too focused on telling users about what to do rather than how to do .There are many fitness apps available in market which allows user to keep track of their health and provide workout plan to user to achieve their personal fitness goal. These app have many functions that provide user with only data to user it does not actually provide a space where user can do their work out without going to gym or having a personal trainer. These apps do not overview whether the user is actually doing the exercise and if they are doing correctly. Their services do not reflect upon the performance of the users and it doesn't let the user know of their shortcomings in the posture they maintain while performing their exercises.

2.3 Proposed System

We all know it is very important to exercise with correct form. Nowadays people prefer working out at home. It is very convenient and time saving as well. We can workout anytime we want at home. But we also know that a trainer is very important to keep a track of our exercise. This project is our effort to make an AI based trainer. An AI-powered personal trainer that works simply by pointing a camera at a person completing a workout, and having a human pose estimation model (specific poses related to a workout regimen indicate whether or not a given exercise has been completed properly.

We aim to build this with the help of OpenCV and MediaPipe that would help you exercise more efficiently in your own homes. The project focuses on creating a customized function for each exercise using which it can help you exercise, by determining the quality and quantity of repetitions which is done by using pose estimation running on the CPU.

A system which will take care of all the parameters of workout like counting sets, repetitions and further giving suggestions to maximize user performance using AI and it will also check the form of exercise. The person will be doing his exercise live and live results will be given as outputs. There is a counter which will count the number of reps a person is doing while performing the workout. The status of the workout will also be displayed on the screen; the system will detect the motion of the exercise by calculating the angles between different points.

The process of proposed system is:

1. Real time video feed is captured through camera
2. Mediapipe establishes pose landmarks onto the person

3. User’s pose landmarks co-ordinates are extracted from real time
4. The co-ordinates are passed on the associated validation function.
5. After each successful completion of exercise, count is incremented.
6. If user commits any mistakes, they will notified to the user through voice commands
7. This can be used by the user to improve their performance

2.4 Novelty of Proposed System

The main novelty of this project is that it can challenge the traditional approach and provide an entirely new alternative to people through pose detection. This kind of application could enable safer and more inspirational home workout routines, while also increasing the accessibility and decreasing the costs associated with professional physical trainers This can also be an example of how we can use pose detection modules in the field of fitness industry to improve their mobility and efficiency. Our project overcomes the limitations of the existing fitness apps by utilizing the technology of computer vision and pose detection.

2.5 Software Requirements

1. Python 3.8
2. PyCharms IDE
3. OpenCV Library
4. Mediapipe Framewors

2.6 Hardware Requirements

System: OS Windows 10 or higher versions or ubuntu with webcam

Processor (CPU): i5 Processor and above

Memory: Minimum 8GB RAM

2.7 Usability

The system must be simple to use in terms of user interface and must avoid any complexity. It should be capable of having minimum interaction with the user to avoid much manual work, but at the same time should provide the best results possible. It should not take much time on processing or should stuck at many moments. In simple terms it should satisfy user needs with simplicity in terms of usability.

2.8 Reliability

The system must be reliable, it should not lead to unnecessary crashes and shouldn’t stuck at most cases of errors when occur, it should have good exception handling mechanisms.

The system should perform well in critical situations to most friendly experience in terms of handling the system with little or less failures.

2.9 Performance

The system or software designed shouldn’t slow down while performing. The system should be fast enough to produce results. Additionally, it should be able to perform well in case of more workloads. This attribute determines the overall functionality of the system in terms of resource consumption, response time and efficiency.

3.DIAGRAMS

3.1.UML DIAGRAMS

UML is the short form of Unified Modelling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group .The important goal for UML is to create a common modelling language for thesake of Object Oriented Software engineering .In its current form UML consists of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization ,Constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems .The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems .The UML is a very important part
of developing object oriented software andt he software development process. The UML uses mostlygraphical notations to express the design of software projects.It is divided into two types 1.Structural Diagrams. 2.Behavioral Diagram.

1.Structure Diagrams

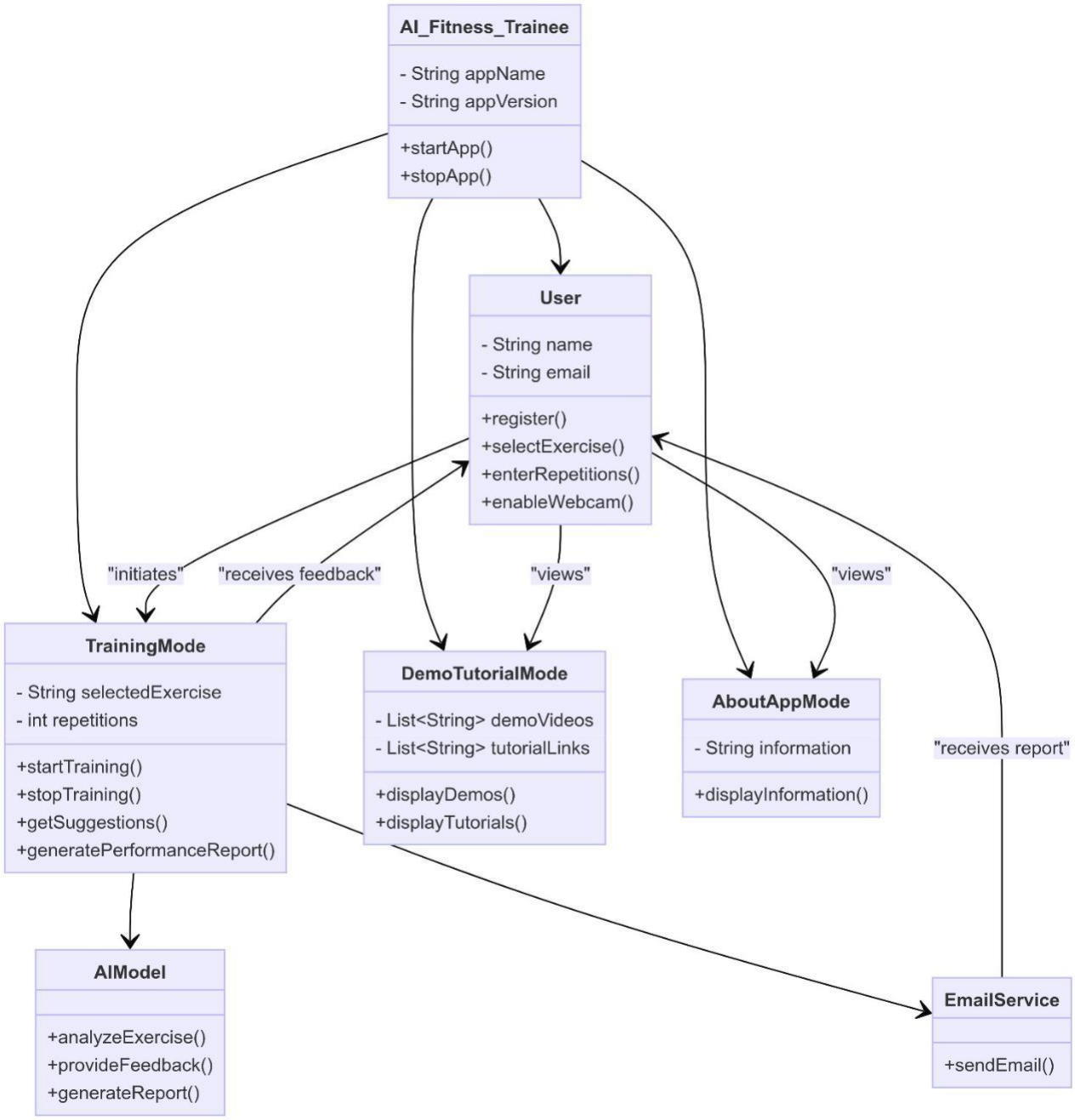
- Class Diagram
- Component Diagram
- Deployment Diagram
- Object Diagram
- Package Diagram
- Profile Diagram
- Composite Structure Diagram

2.Behavioral Diagrams

- Use Case Diagram
- Activity Diagram
- State Machine Diagram
- Sequence Diagram
- Communication Diagram
- Interaction Overview Diagram
- Timing Diagram

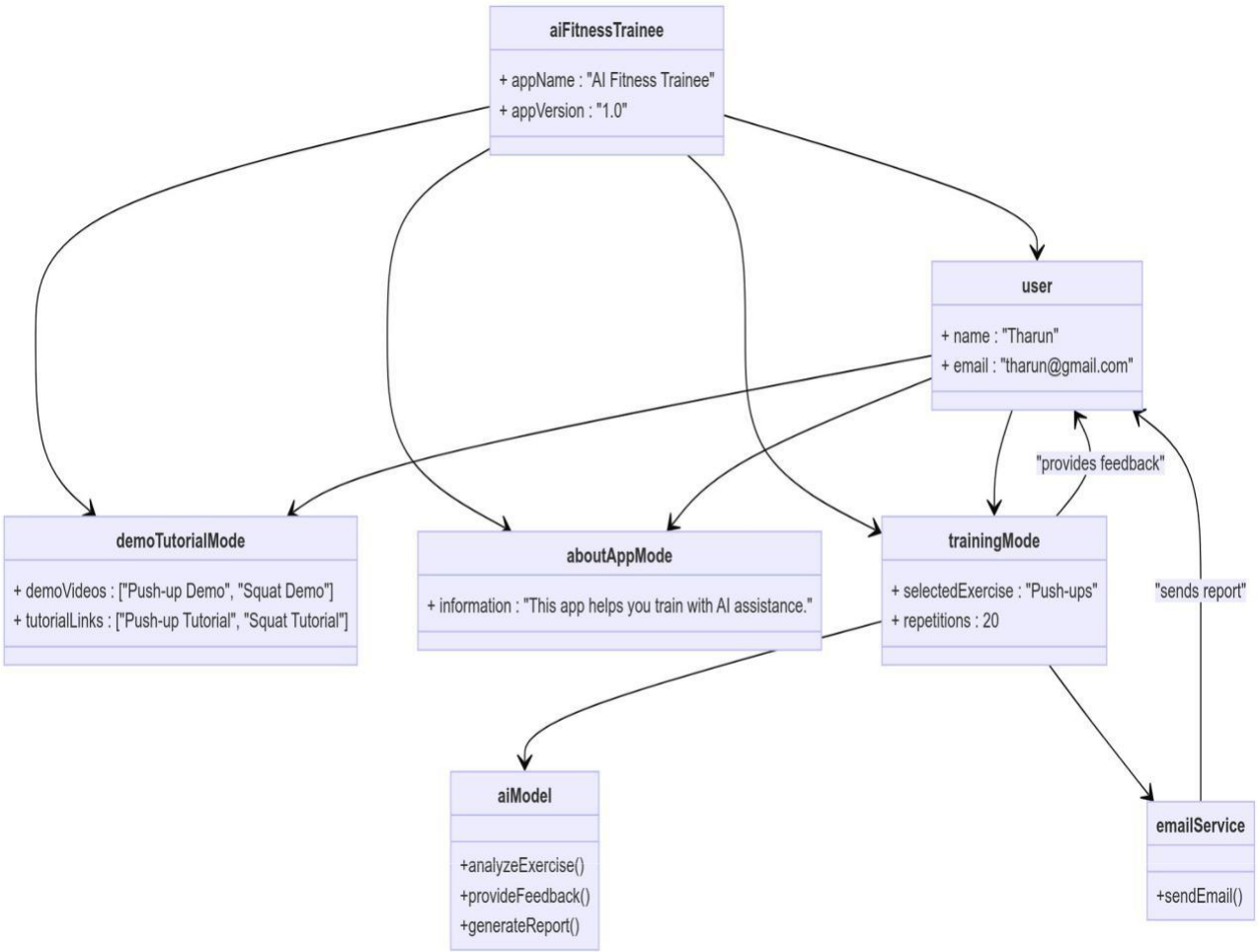
CLASS DIAGRAM

The class diagram is the main building block of object-oriented modelling .It is used for general conceptual modelling of the structure of the application ,and for detailed modelling translating the models into programming code. Class diagrams cam also be used for data modelling.



OBJECT DIAGRAM

It describes the static structure of a system at a particular point in time. It can be used to test the accuracy of class diagrams. It represents distinct instances of classes and the relationship between them at a time.

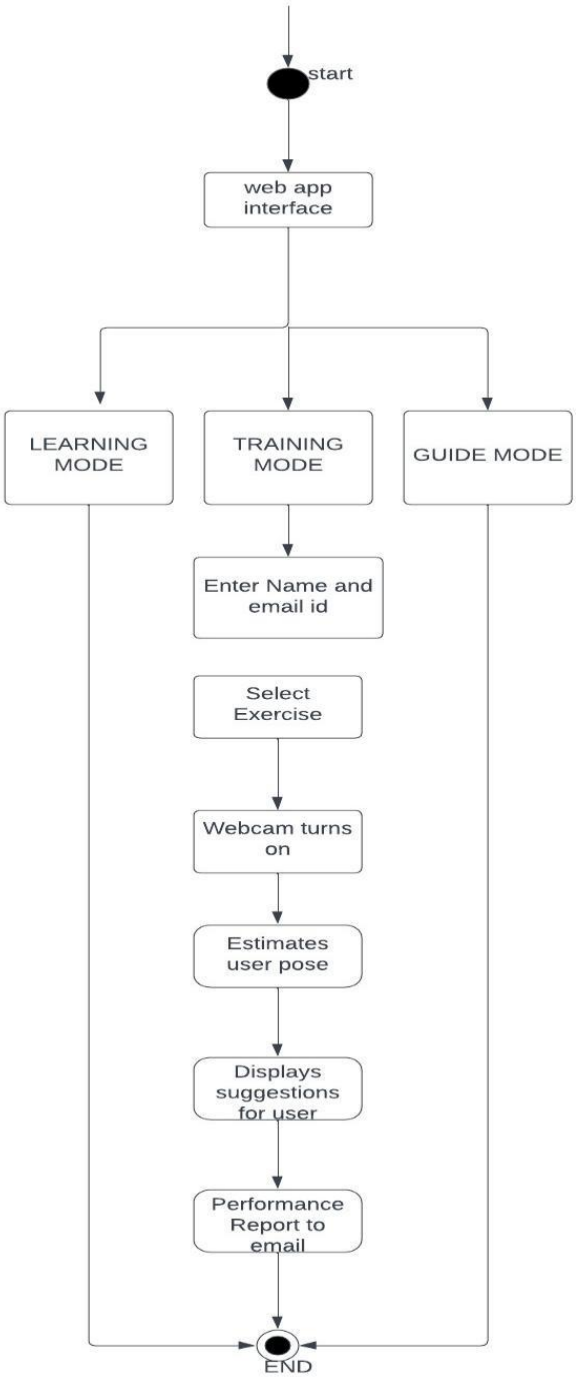


SEQUENTIAL DIAGRAM

A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects.

ACTIVITY DIAGRAM

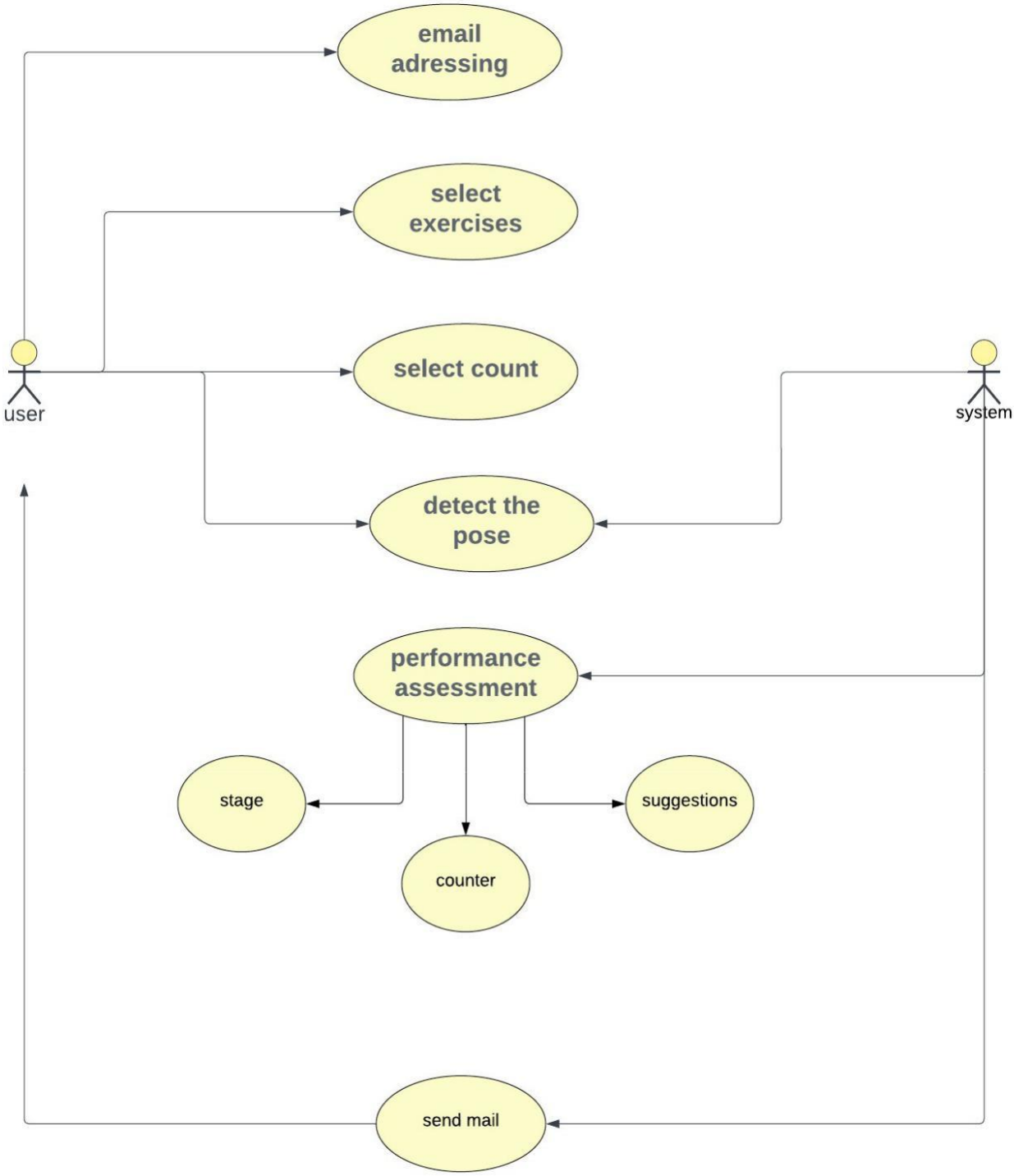
An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process



modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequeuncial.

USECASE DIAGRAM

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use- case diagrams describe what the system does and how the actors use it, but not how the system operates internall.



4. IMPLEMENTATION

This chapter explains the key findings, system behavior, and actions that were observed during the development and implementation of our project. The examples, procedures, and scenarios mentioned in this chapter are mostly based on the observations self-experiments and experiences of the author, the concrete suggestions from expert circle taken from tutorials or forums, as well as from the manufacturer documentation and manu -facturer software repositories. Occasionally, references are also made to theoritical approaches and fundamentals. The practical tests and research described in this chapter were all performed in the same system environment. All test results, operating system and software behavior mentioned in these subchapters are only valid i subchapters are only valid in an identical system environment.

4.1. SOFTWARES USED :

4.1.1. GIT :

Git is a distributed version control system used in software development to track changes made to source code over time. It was created by Linus Torvalds in 2005 and is widely used by developers to collaborate on projects, share code, and manage different versions of the same codebase.

Git allows developers to work on different branches of the codebase and merge changes together when ready, providing a way to manage conflicts and prevent code from being overwritten or lost. It also provides a history of changes made to the codebase, enabling developers to revert to previous versions if necessary.

Git is a command-line tool, but there are also many graphical user interfaces available that provide a more user-friendly experience. Git is free and open source, and is widely used by individual developers, teams, and large organizations

4.2. SOFTWARE INSTALLATION:

Before installing the openCV a Python environment is required which is why PyCharm has been downloaded and installed. After installing PyCharm, the Python version was downgraded to 3.9.2, because the later used Python package TensorFlow is not compatible with Python 3.10. It currently only supports Python versions 2.7, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9 (TensorFlow Community, n.d.).

The downgrade can be done using the following command in the terminal:

PS C: \project> pip install python=3.9.2

After the downgrade of Python the packages openCV and mediapipe were installed and the system was updated. If the installations went well, the next step is the installation of other Python packages. Among them important are numpy and streamlit. These Python packages can either be installed individually by command or all together by reading them from a file and installing them one after the other.

The list of packages that are used

NumPy

NumPy (Numerical Python) is a scientific computing library for Python that provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is a fundamental library for scientific computing with Python and is widely used in various fields, including physics, engineering, data science, and machine learning.

NumPy provides various tools for working with arrays, such as indexing,reshaping, slicing, and broadcasting. It also includes functions for linear algebra,Fourier analysis, random number generation, and more. The key data structure in NumPy is the ndarray (n-dimensional array), which is a container for homogeneous data and provides efficient storage and manipulation of numerical data.

Pandas

Pandas is a popular Python library for data manipulation and analysis. It provides powerful tools for working with structured data, such as data frames and series, making it a valuable tool for a wide range of applications. It provides efficient tools for data cleaning, filtering, merging, and reshaping, making it easy to work with messy or complex data.

Streamlit

A Python library for creating web applications with Python. It makes it easy to create web-based user interfaces with widgets, charts, tables, and more.

Mediapipe

It is an open-source framework developed by Google for building multimodal machine learning pipelines, particularly for processing perceptual data such as video and audio. It provides a set of reusable building blocks that can be combined to create complex pipelines for tasks such as object detection, hand tracking, facial recognition, and pose estimation.

The framework consists of a set of pre- built models and algorithms for common computer vision and machine learning tasks, as well as a set of customizable components that can be used to build more complex pipelines. It includes tools for processing video and audio streams in real-time, as well as tools for annotating and visualizing data.

Mediapipe is designed to be flexible and modular, allowing developers to easily plug in their own models and algorithms or modify existing ones. It also supports a variety of hardware platforms, including CPUs, GPUs, and specialized accelerators such as Google's Edge TPU.

Mediapipe has been used in a variety of applications, including virtual try-on, augmented reality, and interactive installations. It is available as a Python package and can also be used through a web API or on mobile devices through the TensorFlow Lite runtime.

Math

A built-in Python library for mathematical operations.

Time

A built-in Python library for time-related functions.

CV2

CV2 is a Python library for computer vision tasks. It is a wrapper around the OpenCV (Open Source Computer Vision) C++ library and provides a simple and intuitive Python interface for performing common computer vision operations such as image processing, object detection, and video analysis.

Some of the key features of cv2 include:

Image and video manipulation: cv2 provides a range of tools for manipulating and processing images and video streams, including operations such as resizing, cropping, color conversion, and filtering.

Object detection and tracking: cv2 includes pre-trained models and algorithms for detecting and tracking objects in images and videos, including face detection, pedestrian detection, and motion tracking.

Feature detection and matching: cv2 provides algorithms for detecting and matching features between images, which can be used for tasks such as image alignment, 3D reconstruction, and object recognition.

Camera calibration: cv2 includes tools for calibrating cameras, which can be used to correct for distortion and other imaging artifacts.

cv2 is widely used in academic research and industry for a variety of applications such as robotics, autonomous vehicles, and augmented reality. It is also commonly used in conjunction with other Python libraries such as NumPy, matplotlib, and scikit-image.

Matplotlib

This is a plotting library for Python. It provides a wide range of functions for creating different types of plots and visualizations, and is often used for data visualization and exploration.

smtplib Library

The smtplib library in Python provides a simple and efficient way to send emails using the SMTP protocol. It allows Python applications to connect to an SMTP server, authenticate with credentials, and send email messages.

Key Features and Concepts:

1. **SMTP Client Session (**smtplib.SMTP**)**:
    - smtplib.SMTP represents an SMTP client session that can be used to send emails.
    - It connects to the SMTP server specified by its hostname and port number (e.g., smtp.gmail.com on port 587 for Gmail's SMTP server).
2. **Secure Communication**:

- starttls(): Method used to upgrade an unencrypted connection to a secure one using Transport Layer Security (TLS). This is essential for securing email transmission over the internet.

3. **Authentication**:
    - login(username, password): Method to authenticate the client session with the SMTP server using the provided username (usually the email address) and password.
    - Alternatively, some SMTP servers support OAuth tokens or application-specific passwords for authentication.
4. **Sending Email**:
    - sendmail(from_addr, to_addrs, msg): Method to send an email message. It takes the sender's address (from_addr), recipient addresses (to_addrs), and the complete email message (msg) as arguments.
    - The message (msg) should be formatted according to RFC 2822 standards, which includes headers like From, To, Subject, and the email body.
5. **Email Components**:
    - MIME (Multipurpose Internet Mail Extensions) Support: The library supports MIME format for email messages. You can use email.mime modules (MIMEMultipart, MIMEText, etc.) to construct complex email structures with attachments and HTML content.
6. **Error Handling**:
    - Use try-except blocks to handle exceptions that may occur during SMTP operations, such as network errors, authentication failures, or SMTP server issues.
7. **Closing the Connection**:
    - Always close the SMTP connection gracefully using quit() after sending emails. This ensures that resources are released properly and the connection is terminated with the server.

Random

This is a Python module that provides functions for generating random numbers. It is used in the code to select a random image file to attach to the email

5.SOURCE CODE

File name: trainee.py

Description: This file consists of the user interface along with list of exercises that can be assessed. It displays the repetitions of exercise performed along with progress bar and suggestions indication direction.

Code:

```
import streamlit as st
import mediapipe as mp
import numpy as np
import math
import time
import cv2
from EmailingSystem import email_user
from graph import single_plot,double_plot
from calc import cals
flag=1
mp_pose = mp.solutions.pose
mp_drawing = mp.solutions.drawing_utils
# user position
stage = None
message=" "
# count of correct
movement counter = 0
l_angles=[] r_angles=[]
frames=[]
frame_count=0
@st.cache_resource()
def image_resize(image, width=None, height=None,
   inter=cv2.INTER_AREA): dim = None
   (h, w) = image.shape[:2]
   if width is None and height is
      None: return image
   if width is None:
      r = height / float(h)
      dim = (int(w * r), height)
else:
      r = width / float(w) dim
      = (width, int(h * r))
   resized = cv2.resize(image, dim,
   interpolation=inter) return resized
def calculate_angle(a, b, c):
   a = np.array(a)
   b = np.array(b) c
   = np.array(c)
   radians = np.arctan2(c[1] - b[1], c[0] - b[0]) - np.arctan2(a[1] - b[1], a[0] -
   b[0]) angle = np.abs(radians * 180.0 / np.pi)
```

```
    if angle > 180.0:
        angle = 360 - angle
    return angle
def get_pos(img, results):
    landmarks = []
    for id, lm in enumerate(results.pose_landmarks.landmark):
        h, w, c = img.shape
        cx, cy = int(lm.x * w), int(lm.y * h)
        landmarks.append([id, cx, cy])
    return landmarks
def drawOn(img, p1, p2, p3, angle,lmList):
    x1, y1 = lmList[p1][1:]
    x2, y2 = lmList[p2][1:]
    x3, y3 = lmList[p3][1:]
    cv2.line(img, (x1, y1), (x2, y2), (0, 255, 0), 1)
    cv2.line(img, (x3, y3), (x2, y2), (0, 255, 0), 1)
    cv2.circle(img, (x1, y1), 5, (255, 0, 0), cv2.FILLED)
    cv2.circle(img, (x2, y2), 5, (255, 0, 0), cv2.FILLED)
    cv2.circle(img, (x3, y3), 5, (255, 0, 0), cv2.FILLED)
    cv2.putText(img, str(int(angle)), (x2 - 50, y2 + 50),
                cv2.FONT_HERSHEY_PLAIN, 2, (0, 0, 255), 2)
st.markdown(
    """
    <style>
    [data-testid="stSidebar"][aria-expanded="true"] > div:first-child {
        width: 350px;
    }
    [data-testid="stSidebar"][aria-expanded="false"] > div:first-child {
        width: 350px;
        margin-left: -350px;
    }
    </style>
    """,
    unsafe_allow_html=True,
)
st.sidebar.title('AI Fitness Trainer\n using MediaPipe and OpenCV')
app_mode = st.sidebar.selectbox('', ['Training', 'About App','Demos and Tutorials'])
if app_mode == 'About App':
    st.title("AI FITNESS TRAINEE")
    st.markdown(
        'In this application we are using Mediapipe for detecting exercise gestures and opencv for
webcam reading and StreamLit for creating the Web Graphical User Interface (GUI)')
    st.markdown(
        """
    <style>
    [data-testid="stSidebar"][aria-expanded="true"] > div:first-child {
        width: 350px;
    }
    [data-testid="stSidebar"][aria-expanded="false"] > div:first-child {
        width: 350px;
        margin-left: -350px;
    }
```

```
    </style>
    """,
        unsafe_allow_html=True,
    )
    text_html = '<h3 style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none;
padding: 5px;">INSTRUCTIONS TO USE</h3>'
    st.markdown(text_html, unsafe_allow_html=True)
    st.markdown('step-1: Choose training mode to start your fitness today')
    st.markdown('step-2: If you are not aware of what you are intended to do,be comfortable to
go our demos and tutorial section')
    st.markdown('step-3: Now,it is the time to start your exercise,please provide Name and email')
    st.markdown('step-4: Select exercise fom dropdown')
    st.markdown('step-4: Select how many time you want to repeat the exercise')
    st.markdown('step-5: Tick on the start')
    st.markdown('step-6: Now,our trainee can be able to capture your pose')
    st.markdown('step-7: start doing exercise')
    st.markdown('step-8: count will be displayed on screen how many times you repeated')
    st.markdown('step-9: Finish those repetitions you mentioned,our trainee will count for you')
    st.markdown('step-10: Check your given mail inbox to see your performance report')
    text_html = '<h3 style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none;
padding: 5px;">BE AWARE...</h3>'
    st.markdown(text_html, unsafe_allow_html=True)
    st.markdown('Make sure that you have proper Lighting')
    st.markdown('Ensure that your camera is working properly and in correct angle to capture you')
elif app_mode == 'Demos and Tutorials':
        #HIGH KNESS
        st.title("HIGH KNEES")
        link_html = '<a href="https://youtu.be/oDdkytliOqE?si=Mferh8hYmi14Rh7m"
style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding:
5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        hk_video_url = "https://www.shutterstock.com/shutterstock/videos/1060403879/preview/ stock-
footage-athletic-woman-doing-high-knee-exercise-at-home-home-training-workout-home-fitness-there-
is-some.webm"
        st.video(hk_video_url)
        #SQUATS
        st.title("SQUATS")
        link_html = '<a href="https://youtu.be/4KmY44Xsg2w?si=eh8-DmSgjZptopSs"
style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding:
5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        s_video_url = "https://www.shutterstock.com/shutterstock/videos/1102020407/preview/ stock-
footage-female-butt-workout-squats-at-home-athletic-asian-woman-squats-workout-in-living-room-
female.webm"
        st.video(s_video_url)
        #SHOULDER PRESS
        st.title("SHOULDER PRESS")
        link_html = '<a href="https://youtu.be/xe19t2_6yis?si=SIpCkA04UFBoJfs_"
style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding:
5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
```

```
        shoulder_video_url = "https://ak.picdn.net/shutterstock/videos/1104019445/preview/stock-footage-medium-shot-of-young-sportsman-performing-barbell-shoulder-press-in-gym-man-holds-barbell-on-the.mp4"
        st.video(shoulder_video_url)
        #LATERAL CURLS
        st.title("LATERAL CURLS")
        link_html = '<a href="https://youtu.be/PzsMitRdI_8?si=Xt_bteImzFf_vP7m" style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding: 5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        lcurls_video_url = "https://www.shutterstock.com/shutterstock/videos/1012307186/preview/stock-footage-bodybuilder-performs-seated-side-lateral-raise-reflection-of-exercise-in-mirror.webm"
        st.video(lcurls_video_url)
        #CURLS
        st.title("CURLS")
        link_html = '<a href="https://youtu.be/sYV-ki-1blM?si=Oi9cMYUAKZnPggP _" style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding: 5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        curls_video_url = "https://www.shutterstock.com/shutterstock/videos/1097465161/preview/ stock-footage-athlete-exercising-with-dumbbells-in-indoor-gym-fit-man-doing-biceps-triceps-curls-workout-in-gym.webm"
        st.video(curls_video_url)
        #LEFT CURLS
        st.title("Left curls")
        link_html = '<a href="https://www.youtube.com/shorts/cHxRJdSVIkA?feature=share" style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding: 5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        lecurls_video_url ="https://www.shutterstock.com/shutterstock/videos/1030360235/preview/stock-footage-fitness-sport-weightlifting-and-bodybuilding-concept-man-exercising-with-dumbbells-at-home.webm"
        st.video(lecurls_video_url)
        #PUSHUPS
        st.title("PUSH UPS")
        link_html = '<a href="https://youtu.be/_UBOxUl0Sl4" style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding: 5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        pushups_video_url = "https://www.shutterstock.com/shutterstock/videos/1100041747/preview/stock-footage-calisthenics-outdoor-bodyweight-workout-doing-push-ups-in-sunny-day.webm"
        st.video(pushups_video_url)
        #PLANK
        st.title("PLANK")
        link_html = '<a href="https://youtu.be/pvIjsG5Svck?si=ry3kol3TxHF7hu3g" style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding: 5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        p_video_url = "https://www.shutterstock.com/shutterstock/videos/1063989619/preview/ stock-footage-fitness-indian-woman-doing-plank-exercise-workout-in-gym-indoors.webm"
        st.video(p_video_url)
```

```
        #SIT UPS
        st.title("SIT UPS")
        link_html = '<a href="https://youtu.be/1fbU_MkV7NE?si=bcHWl9xa7PkFSsAu"
style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding:
5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        situps_video_url = "https://cdn.coverr.co/videos/coverr-ab-workout-in-the-
park-7546/1080p.mp4"
        st.video(situps_video_url)
        #KNEE BEND
        st.title("KNEE BEND")
        link_html = '<a href="https://youtu.be/W4LuS9rK0gU?si=A5dTYz_OHyQmvHHd"
style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding:
5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        kb_video_url ="https://www.shutterstock.com/shutterstock/videos/1061478490/preview/ stock-
footage-caucasian-woman-spending-time-at-home-in-living-room-exercising-with-dumbbells-doing-
squats-in.webm"
        st.video(kb_video_url)
        #CRUNCHES
        st.title("CRUNCHES")
        link_html = '<a href="https://youtu.be/Xyd_fa5zoEU?si=uK846BgYpam0H4X3"
style="background-color: rgb(0, 128, 131); color: #fff; text-decoration:none; padding:
5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        cru_video_url = "https://www.shutterstock.com/shutterstock/videos/1041475927/preview/ stock-
footage-long-haired-girl-doing-bicycle-crunches-exercise-working-out-at-empty-studio-in-front-the-
windows.webm"
        st.video(cru_video_url)
        #HALF PLOUGH POSE
        st.title("HALF PLOUGH POSE")
        link_html = '<a href="https://youtu.be/uqKbLaXSfWE" style="background-color: rgb(0,
128, 131); color: #fff; text-decoration:none; padding: 5px;">Video tutorial &rarr;</a>'
        st.markdown(link_html, unsafe_allow_html=True)
        hp_video_url = "https://www.shutterstock.com/shutterstock/videos/1105807867/preview/ stock-
footage-video-of-woman-performing-alternate-leg-ardha-halasana-this-exercise-strengthen-the-muscles-
of.webm"
        st.video(hp_video_url)
elif app_mode == 'Training':
    st.title('Select Exercise')
    st.set_option('deprecation.showfileUploaderEncoding', False)
    suggestion = " "
    left, center, right = st.columns(3)
    name = st.text_input("Name", "")
    email = st.text_input("Email address", "")
    option = st.selectbox(
        'Exercise',
        ("Select One", "High Knees","Squats","Shoulder Press","Lateral Curls", "Curls", "Left
Curl", "Push Ups", "Plank","Sit Up",
         "Knee Bend", "Crunches","Half Plough Pose"))
    press_time = st.select_slider(
        'How many times you want to perform',
```

```
        options=['0', '2', '10', '15', '20', '25', '30'])
    cal_value=cals(option,int(press_time))
    message = "You have done " + option + " for " + press_time + " times and burnt
"+str(cal_value) +" calories"
    run = st.checkbox("Start")
    if option == "Left Curl":
        count = 0
        direction = 0
        prev_per = 0
        e = 0
    if option == "Push Ups":
        dir = 0
    if option == "Knee Bend":
        relax_counter = 0
        bent_counter = 0
        bent_time = 0
        relax_time = 0
    if option == "Crunches":
        f=0
    path=0
    # path="test/left_curl_demo.mp4"
    # path = "test/letral.mp4"
    # path = "test/curls_demo.mp4"
    # path = "test/squats1.mp4"
    # path = "test/shoulderpress_demo.mp4"
    # path = "test/highkness.mp4"
    # path = "test/pushup2_demo.mp4"
    # path = "test/KneeBendVideo.mp4"
    # path = "test/situps.mp4"
    # path = "test/crunches 1.mp4"
    # path = "test/plough.mp4"
    # time.sleep(2)
    while run:
        stframe = st.empty()
        cap = cv2.VideoCapture(path)
        width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
        height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
        fps_input = int(cap.get(cv2.CAP_PROP_FPS))
        fps = 0
        i = 0
        ltitle, ctitle, rtitle = st.columns(3)
        with ltitle:
            st.markdown("**Frame Rate**")
            kpi1_text = st.markdown("0")
        with ctitle:
            st.markdown("**Count**")
            kpi2_text = st.markdown("0")
        with rtitle:
            st.markdown("**Target**")
            kpi3_text = st.markdown("0")
        st.markdown("**Suggestions**")
        sugg = st.empty()
```

```
        sugg.markdown("---")
        st.markdown("<hr/>", unsafe_allow_html=True)
        with mp_pose.Pose(static_image_mode=False,
                          model_complexity=1,
                          smooth_landmarks=True,
                          enable_segmentation=False,smooth_segmentation=True,
min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:
            prevTime = 0
            while cap.isOpened():
                ret, frame = cap.read()
                frame_count+=1
                if not ret:
                    continue
                frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                result = pose.process(frame)
                try:
                    landmarks = get_pos(frame, result)
                    rshoulder = landmarks[12][1:]
                    relbow = landmarks[14][1:]
                    rwrist = landmarks[16][1:]
                    lshoulder = landmarks[11][1:]
                    lelbow = landmarks[13][1:]
                    lwrist = landmarks[15][1:]
                    rhip = landmarks[24][1:]
                    rknee = landmarks[26][1:]
                    rankle = landmarks[28][1:]
                    lhip = landmarks[23][1:]
                    lknee = landmarks[25][1:]
                    lankle = landmarks[27][1:]
                    # angle_right_elbow = round(calculate_angle(rshoulder, relbow, rwrist))
                    # angle_left_elbow = round(calculate_angle(lshoulder, lelbow, lwrist))
#
                    # angle_right_knee = round(calculate_angle(rhip, rknee, rankle))
                    # angle_left_knee = round(calculate_angle(lhip, lknee, lankle))
                    #
                    # angle_right_shoulder = round(calculate_angle(rwrist, rshoulder, rhip))
                    # angle_left_shoulder = round(calculate_angle(lwrist, lshoulder, lhip))

                    # cv2.putText(frame, str(angle_right_elbow), (relbow[0], relbow[1] - 10),
                    #             cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
                    # cv2.putText(frame, str(angle_left_elbow), (lelbow[0], lelbow[1] - 10),
                    #             cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
                    # cv2.putText(frame, str(angle_right_knee), (rknee[0], rknee[1] -
10), cv2.FONT_HERSHEY_SIMPLEX,
                    #             0.5, (255, 255, 255), 1)
                    # cv2.putText(frame, str(angle_left_knee), (lknee[0], lknee[1] -
10), cv2.FONT_HERSHEY_SIMPLEX, 0.5,
                    #             (255, 255, 255), 1)
                    # cv2.putText(frame, str(angle_right_shoulder), (rshoulder[0] + 10, rshoulder[1] + 10),
                    #             cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
                    # cv2.putText(frame, str(angle_left_shoulder), (lshoulder[0] - 50, lshoulder[1] + 10),
                    #             cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
```

```
            if option == "Left Curl":
                angle_left_elbow = round(calculate_angle(lshoulder, lelbow, lwrist))
                l_angles.append(angle_left_elbow)
                frames.append(frame_count)
                per = np.interp(angle_left_elbow, (45, 155), (100, 0))
                bar = np.interp(angle_left_elbow, (45, 155), (60, 420))
                drawOn(frame, 11, 13, 15, angle_left_elbow, landmarks)
                 color = (255, 0, 255)
                if per == 100:
                    color = (0, 255, 0)
                    if direction == 0:
                        direction = 1
                        counter += 1
                    else:
                        suggestion="Lift Arm Down"
                if per == 0:
                    if direction == 1:
                        direction = 0
                    else:
                        suggestion="Lift Arm Up"
                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2,
                            cv2.LINE_AA)
                cv2.rectangle(frame, (520, 60), (590, 420), color, 2) cv2.rectangle(frame, (520,
                int(bar)), (590, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per)} %',
                (520, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
            if option == "Curls":
                angle_right_elbow = round(calculate_angle(rshoulder, relbow, rwrist))
                angle_left_elbow = round(calculate_angle(lshoulder, lelbow, lwrist))
                drawOn(frame, 11, 13, 15, angle_left_elbow, landmarks)
                l_angles.append(angle_left_elbow)
                r_angles.append(angle_right_elbow)
                frames.append(frame_count)
                drawOn(frame, 12, 14, 16, angle_right_elbow, landmarks)
                color = (255, 0, 255)
                if angle_left_elbow >= 165 and angle_right_elbow >= 165:
                    stage = "open"
                    suggestion = "Keep elbows close."
                elif angle_left_elbow <= 34 and angle_right_elbow <= 34 and stage == "open":
                    stage = "close"
                    counter += 1
                    suggestion = "Make sure you are lifting the weights fully"
                # visualizing angle_left_elbow
                per1 = np.interp(angle_left_elbow, (34, 165), (100, 0))
                bar1 = np.interp(angle_left_elbow, (34, 165), (60, 420))
                per2 = np.interp(angle_right_elbow, (34, 165), (100, 0))
                bar2 = np.interp(angle_right_elbow, (34, 165), (60,
                420)) if per1 == 100 and per2==100:
                    color=(0,255,0)
```

```
                cv2.rectangle(frame, (520, 60), (590, 420), color, 2) cv2.rectangle(frame, (520,
                int(bar1)), (590, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per1)} %',
                (520, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
                cv2.rectangle(frame, (20, 60), (90, 420), color, 2) cv2.rectangle(frame, (20,
                int(bar2)), (90, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per2)} %',
                (20, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                        (0, 0, 255), 2, cv2.LINE_AA)
            if option == "Lateral Curls":
                angle_right_shoulder = round(calculate_angle(rwrist, rshoulder, rhip))
                angle_left_shoulder = round(calculate_angle(lwrist, lshoulder, lhip))
                l_angles.append(angle_left_shoulder)
                r_angles.append(angle_right_shoulder)
                frames.append(frame_count)
                drawOn(frame, 15, 11, 23, angle_left_shoulder, landmarks)
                drawOn(frame, 16, 12, 24, angle_right_shoulder, landmarks)
                color = (255, 0, 255)
                if angle_right_shoulder >= 100 and angle_left_shoulder >= 100 and stage
=="down":
                    stage = "up"
                    counter += 1
                    suggestion = "Down"
                elif 30 >= angle_right_shoulder and 30 >= angle_left_shoulder:
                    stage = "down"
                    suggestion = "Up"
                 # visualizing angle_left_shoulder
                per1 = np.interp(angle_left_shoulder, (30, 100), (0, 100))
                bar1 = np.interp(angle_left_shoulder, (30, 100), (420, 60))
                per2 = np.interp(angle_right_shoulder, (30, 100), (0, 100))
                bar2 = np.interp(angle_right_shoulder, (30, 100), (420,
                60)) if per1 == 100 and per2==100:
                    color=(0,255,0)
                cv2.rectangle(frame, (520, 60), (590, 420), color, 2) cv2.rectangle(frame, (520,
                int(bar1)), (590, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per1)} %',
                (520, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color,2)
                cv2.rectangle(frame, (20, 60), (90, 420), color, 2) cv2.rectangle(frame, (20,
                int(bar2)), (90, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per2)} %',
                (20, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
  cv2.putText(frame, f"stage: {str(suggestion)}", (10, 100), cv2.FONT_HERSHEY_SIMPLEX,
                        1, (0, 0, 255), 2, cv2.LINE_AA)
 if option == "Squats":
                angle_right_knee = round(calculate_angle(rhip, rknee, rankle))
                angle_left_knee = round(calculate_angle(lhip, lknee, lankle))
                l_angles.append(angle_left_knee)
                r_angles.append(angle_right_knee)
                frames.append(frame_count)
                drawOn(frame, 24, 26, 28, angle_right_knee, landmarks)
```

```
                drawOn(frame, 23, 25, 27, angle_left_knee, landmarks)
                color = (255, 0, 255)
                if angle_right_knee >= 173 and angle_left_knee >= 173:
                    stage = "down"
                    suggestion = "Go Down"
                elif angle_right_knee <= 90 and angle_left_knee <= 90 and stage == 'down':
                    stage = "up"
                    counter += 1
                    suggestion = "Go Up"
                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                            (0, 0, 255), 2, cv2.LINE_AA)
                # visualizing angle_left_knee
                per1 = np.interp(angle_left_knee, (90, 173), (100, 0))
                bar1 = np.interp(angle_left_knee, (90, 173), (60, 420))
              # visualizing angle_right_knee
                per2 = np.interp(angle_right_knee, (90, 173), (100, 0))
                bar2 = np.interp(angle_right_knee, (90, 173), (60,
                420)) if per1 == 100 and per2==100:
                    color=(0,255,0)
                cv2.rectangle(frame, (520, 60), (590, 420), color, 2) cv2.rectangle(frame, (520,
                int(bar1)), (590, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per1)} %',
                (520, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
                cv2.rectangle(frame, (20, 60), (90, 420), color, 2) cv2.rectangle(frame, (20,
                int(bar2)), (90, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per2)} %',
                (20, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
            if option == "Push Ups":
                angle_left_elbow = round(calculate_angle(lshoulder, lelbow, lwrist))
                drawOn(frame, 11, 13, 15, angle_left_elbow, landmarks)
                color = (255, 0, 255)
                l_angles.append(angle_left_elbow)
                frames.append(frame_count)
                per = np.interp(angle_left_elbow, (70, 140), (100, 0))
                bar = np.interp(angle_left_elbow, (70, 140), (60,
                420)) if per == 100:
                    color = (0, 255, 0)
                    if dir == 0:
                        counter += 1
                        dir = 1
                    suggestion="Up"
                if per == 0:
                    if dir == 1:
                        dir = 0
                    suggestion="Down"
                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                            (0, 0, 255), 2, cv2.LINE_AA)
                cv2.rectangle(frame, (520, 60), (590, 420), color, 2)
                cv2.rectangle(frame, (520, int(bar)), (590, 420), color, cv2.FILLED)
```

```
                cv2.putText(frame, f'{int(per)} %', (520, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
   if option == "Shoulder Press":
                angle_right_elbow = round(calculate_angle(rshoulder, relbow, rwrist))
                angle_left_elbow = round(calculate_angle(lshoulder, lelbow, lwrist))
                l_angles.append(angle_left_elbow)
                r_angles.append(angle_right_elbow)
                frames.append(frame_count)
                drawOn(frame, 11, 13, 15, angle_left_elbow, landmarks)
                drawOn(frame, 12, 14, 16, angle_right_elbow, landmarks)
                color=(255,0,255)
                if angle_left_elbow > 170 and angle_right_elbow > 170:
                   stage = "Up"
                if angle_left_elbow < 90 and angle_right_elbow < 90 and stage == 'Up':
                   stage = "Down"
                   counter += 1
                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                        (0, 0, 255), 2, cv2.LINE_AA)
                # visualizing angle_left_elbow
                per1 = np.interp(angle_left_elbow, (90, 170), (0, 100))
                bar1 = np.interp(angle_left_elbow, (90, 170), (420, 60))
                per2 = np.interp(angle_right_elbow, (90, 170), (0, 100))
                bar2 = np.interp(angle_right_elbow, (90, 170), (420,
                60)) if per1==100 and per2==100:
                   color=(0,255,0)
                cv2.rectangle(frame, (520, 60), (590, 420), color, 2) cv2.rectangle(frame, (520,
                int(bar1)), (590, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per1)} %',
                (520, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
 # visua;izing angle_right_elbow
                cv2.rectangle(frame, (20, 60), (90, 420), color, 2) cv2.rectangle(frame, (20,
                int(bar2)), (90, 420), color, cv2.FILLED) cv2.putText(frame, f'{int(per2)} %',
                (20, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
 if option == "High Knees":
                angle_left_hip = round(calculate_angle(lshoulder, lhip, lknee))
                drawOn(frame, 11, 23, 25, angle_left_hip, landmarks)
                if angle_left_hip > 160:
                   stage = "Down"
                if angle_left_hip < 80 and stage == 'Down':
                   stage = "Up"
                   counter += 1
                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                        (0, 0, 255), 2, cv2.LINE_AA)
                per = np.interp(angle_left_hip, (80, 160), (0, 100)) bar =
                np.interp(angle_left_hip, (80, 160), (420, 60)) cv2.rectangle(frame, (520, 60),
                (590, 420), color, 2) cv2.rectangle(frame, (520, int(bar)), (590, 420), color,
                cv2.FILLED) cv2.putText(frame, f'{int(per)} %', (520, 50),
                cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)
```

```
            if option == "Knee Bend":
                angle_left_knee = round(calculate_angle(lhip, lknee, lankle))
                drawOn(frame, 23, 25, 27, angle_left_knee, landmarks)
                if angle_left_knee > 140:
                    relax_counter += 1
                    bent_counter = 0
                    stage = "Relaxed"
                    suggestion = ""
                if angle_left_knee < 140:
                    relax_counter = 0
                    bent_counter += 1
                    stage = "Bent"
                    suggestion = ""
                    # rep
                if bent_counter == 8:
                    counter += 1
                    suggestion = 'Rep completed. Relax knee'
                elif bent_counter < 8 and stage == 'Bent':
                    suggestion = 'Keep Your Knee Bent'
                else:
                    suggestion = " "
                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                            (0, 0, 255), 2, cv2.LINE_AA)
            if option == "Sit Up":
                shoulder_avg = [(rshoulder[0] + lshoulder[0]) / 2, (rshoulder[1] + lshoulder[1]) /
                2] hip_avg = [(rhip[0] + lhip[0]) / 2, (rhip[1] + lhip[1]) / 2]
                knee_avg = [(rknee[0] + lknee[0]) / 2, (rknee[1] + lknee[1]) / 2]
                angle = calculate_angle(shoulder_avg, hip_avg, knee_avg)
                drawOn(frame, 11, 23, 25, angle_left_knee, landmarks) if angle
                <= 84 and status=="Up":
                    counter += 1
                    status = "Down"
                    suggestion = "Down"

                if angle >= 90:
                    status = "Up"
                    suggestion = "Up"

                cv2.putText(frame, f"stage: {str(suggestion)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                            (0, 0, 255), 2, cv2.LINE_AA)
                per = np.interp(angle, (84, 90), (0, 100))
                bar = np.interp(angle, (84, 90), (420, 60))
                cv2.rectangle(frame, (520, 60), (590, 420), color, 2)
                cv2.rectangle(frame, (520, int(bar)), (590, 420), color, cv2.FILLED)
                cv2.putText(frame, f'{int(per)} %', (520, 50), cv2.FONT_HERSHEY_PLAIN, 2,
color, 2)

            if option == "Crunches":
                x1 = landmarks[0][1]
```

```
                x2 = landmarks[12][1]

                length = x1 - x2
                print(length)
                if length >= 0 and f == 0:
                    f = 1
                    stage = "Bend Forward"
                    suggestion=stage
                elif length < 0 and f == 1:
                    f = 0
                    stage = "Relax"
                    suggestion=stage
                    counter += 1
                drawOn(frame, 0, 12, 24, length, landmarks)

                cv2.putText(frame, f"stage: {str(stage)}", (10,
100), cv2.FONT_HERSHEY_SIMPLEX, 1,
                            (0, 0, 255), 2, cv2.LINE_AA)


            if option == "Plank":
                angle1 = round(calculate_angle(lshoulder, lelbow, lwrist))
                angle2 = round(calculate_angle(lshoulder, lhip, lknee))
                angle3 = angle_left_knee = round(calculate_angle(lhip, lknee, lankle))
                drawOn(frame, 11, 13, 15, angle1, landmarks)
                drawOn(frame, 11, 23, 25, angle2, landmarks)
                drawOn(frame, 23, 25, 27, angle3, landmarks)
                if not 75 <= angle1 <= 105:
                    suggestion1='Bring your shoulder vertically above your
                    elbow' cv2.putText(frame, f"{str(suggestion1)}", (10, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                                (0, 0, 255), 2, cv2.LINE_AA)
                elif angle2 < 140:
                    suggestion2='Make your back straight. Bring your buttocks
                    DOWN' cv2.putText(frame, f"{str(suggestion2)}", (10, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                                (0, 0, 255), 2, cv2.LINE_AA)

                    elif angle2 > 170:
                    suggestion3='Make your back straight. Bring your buttocks UP'
                    cv2.putText(frame, f"{str(suggestion3)}", (10, 150),
        cv2.FONT_HERSHEY_SIMPLEX, 1,
                      (0, 0, 255), 2, cv2.LINE_AA)
                elif angle3 <= 160:
                    suggestion4='Do not bend your knee. Stretch your legs'
                    cv2.putText(frame, f"{str(suggestion4)}", (10, 200),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                                (0, 0, 255), 2, cv2.LINE_AA)
                else:
                    counter+=1
                    time.sleep(1)
```

```
            if option == "Half Plough Pose":
                # Calculate angles
                angle_left_hip = round(calculate_angle(lshoulder, lhip, lknee))
                drawOn(frame, 11, 23, 25, angle_left_hip, landmarks)

                print(lhip[1], lknee[1])
                # Determine if Half Plough Pose is being
                performed if angle_left_hip<83:
                    suggestion1="Half Plough Pose detected!"
                    cv2.putText(frame, f"{str(suggestion1)}", (10, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                                (0, 0, 255), 2, cv2.LINE_AA)
                    if lhip[1] > lknee[1]:
                        suggestion2="Your form looks good."
                        cv2.putText(frame, f"{str(suggestion2)}", (10, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                                    (0, 0, 255), 2, cv2.LINE_AA)
                        counter += 1
                        time.sleep(1)
                    else:
                        suggestion3="Your hips should be below your knees for proper form."
                        cv2.putText(frame, f"{str(suggestion3)}", (10, 150),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                                    (0, 0, 255), 2, cv2.LINE_AA)
                else:
                    suggestion4="Not Half Plough Pose."
                    cv2.putText(frame, f"{str(suggestion4)}", (10, 200),
cv2.FONT_HERSHEY_SIMPLEX, 1,
                                (0, 0, 255), 2, cv2.LINE_AA)
                 if counter == int(press_time):
                counter = 0
                press_time=0
                suggestion = "Completed"+message
                if option =="Left Curl":
                    single_plot(frames, l_angles, "Left Elbow", 45, 155)
                if option =="Curls":
                    double_plot(frames, l_angles, r_angles, "Left Elbow", "Right Elbow", 34, 165)
                if option =="Lateral Curls":
                    double_plot(frames, l_angles, r_angles, "Left Shoulder", "Right Shoulder", 30, 85)
                if option =="Squats":
                    double_plot(frames, l_angles, r_angles, "Left Knee", "Right Knee", 90, 173)
                if option =="Shoulder Press":
                    double_plot(frames, l_angles, r_angles, "Left Knee", "Right Knee", 90, 170)
                if option =="Push Ups":
                    single_plot(frames, l_angles, "Left Elbow", 70, 140)
                option = False
                if flag:
                    email_user(email, name, message)
                    flag=0
                run = False
                frame_count = 0
                frames = []
```

```
                l_angles = []
                r_angles = []
                cap.release()

            except:
                pass

            mp_drawing.draw_landmarks(frame,
result.pose_landmarks, mp_pose.POSE_CONNECTIONS)
            currTime = time.time()
            fps = 1 / (currTime - prevTime)
            prevTime = currTime
            kpi1_text.write(f"<h1 style='text-align: left; color:
red;'>{int(fps)}</h1>", unsafe_allow_html=True)
            kpi2_text.write(f"<h1 style='text-align: left; color:
red;'>{counter}</h1>", unsafe_allow_html=True)
            kpi3_text.write(f"<h1 style='text-align: left; color:
red;'>{press_time}</h1>", unsafe_allow_html=True)
            sugg.write(f"<h1 style='text-align: left; color:
red;'>{suggestion}</h1>", unsafe_allow_html=True)
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            # frame = cv2.resize(frame,(0,0),fx = 0.8 , fy = 0.8) frame =
            image_resize(image=frame, width=640) stframe.image(frame,
            channels='BGR', use_column_width=True)
```

File name: Cal.py

Description: This file receives parameters name and reps and depending on the exercise performed. It return the number of calories spent during the exercise

Code:

```
    def cals(name,reps):
  if name=="Lateral Curls":
    return reps*1
  if name=="Curls":
    return reps *2
  if name=="Squats":
    return reps * 15
  if name=="Left Curl":
    return reps *1
  if name == "Shoulder Press":
    return reps *15
  if name == "Push Ups":
    return reps *15
  if name == "High Knees":
    return reps *8
  if name == "Knee Bend":
    return reps *2
  if name == "Sit Up":
    return reps *10
```

```
    if name == "Plank":
        return reps *1
    if name == "Half Plough Pose":
        return reps *1
```

File EmailingSystem.py:

Description: This file uses the SMTP library to send an email to a user with a performance summary .The email message includes a greeting, performance summary, and message. A random funny image is attached to the email before being sent to the user .The main() function is used to test the emailing feature by sending a test message to the specified email address.

Code:

```
# email_utils.py

from dotenv import load_dotenv
import os
import random
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText from
email.mime.base import MIMEBase
from email import encoders

# Load environment
variables load_dotenv()

# Environment variables for email credentials
EMAIL_ADDRESS = os.getenv('EMAIL_ADDRESS')
EMAIL_PASSWORD = os.getenv('EMAIL_PASSWORD')

# Check if environment variables are loaded correctly
if not EMAIL_ADDRESS or not EMAIL_PASSWORD:
    raise ValueError("Email address or password environment variables are not set.")

def email_user(email, name, message):
    funny_images = ["fitness.jpg", "fitness1.jpg", "fitness2.jpg",
    "fitness3.jpg"] random_img = random.randint(0, len(funny_images) - 1)
    attachment_path = funny_images[random_img]

    try:
        # Set up the MIME msg =
        MIMEMultipart() msg['From'] =
        EMAIL_ADDRESS msg['To'] =
        email
        msg['Subject'] = "Performance Summary"

        # Attach the body with the msg instance
        greeting = f"<p style=\"font-size:30px; text-align: center; color:Red;\"> <b>Amazing
Workout {name}! </b> </p> <br>"
```

```
        performance_summary = f"<p style=\"font-size:20px; text-align: center;\"><b>Here is
your Performance Summary<b>:<br><br>{message}</p>"
        msg.attach(MIMEText(greeting + performance_summary, 'html'))

        # Attach the file
        if attachment_path:
            with open(attachment_path, "rb") as attachment:
                part = MIMEBase('application', 'octet-stream')
                part.set_payload(attachment.read())
                encoders.encode_base64(part)
                part.add_header(
                    'Content-Disposition',
                    f"attachment; filename= {os.path.basename(attachment_path)}",
                )
                msg.attach(part)

        # Create the SMTP session
        with smtplib.SMTP('smtp.gmail.com', 587) as server:
            server.starttls() # Secure the connection
            server.login(EMAIL_ADDRESS, EMAIL_PASSWORD) # Login to the email server

            # Send the email
            text = msg.as_string()
            server.sendmail(EMAIL_ADDRESS, email, text)

        print(f"Email sent successfully to {email}")

    except Exception as e:
        print(f"Error creating or sending email: {e}")


def main():
    # Replace with your desired message content
    message = "This is your performance summary message. You can add more details
and personalize it further."
    email_user("o190228@rguktong.ac.in", "Priya", message)

if __name__ == "__main__":
    main()
```

File name: graph.py

Description: This file consists of functions that are essential for plotting graphs describing the performance of the user in his workout

Code :

```
import streamlit as st
import matplotlib.pyplot as plt
```

```
def single_plot(frames,left_angle,label1,y1,y2):
    plt.rcParams["figure.figsize"] = (20, 5)
    fig, ax = plt.subplots()
    ax.plot(frames, left_angle, '-', color='red', label=label1)
    ax.axhline(y=y1, color='g', linestyle='--')
    ax.axhline(y=y2, color='g', linestyle='--')
    ax.legend(loc='center left')
    ax.set_xlabel('Frames')
    ax.set_ylabel('Angle')
    st.pyplot(fig)

def double_plot(frames,left_angle,right_angle,label1,label2,y1,y2):
    plt.rcParams["figure.figsize"] = (20, 5)

    fig, ax = plt.subplots()
    ax.plot(frames, left_angle, '-', color='red', label=label1)
    ax.plot(frames, right_angle, '-', color='blue', label=label2)
    ax.axhline(y=y1, color='g', linestyle='--')
    ax.axhline(y=y2, color='g', linestyle='--')
    ax.legend(loc='center left')
    ax.set_xlabel('Frames')
    ax.set_ylabel('Angle')
    st.pypl
```

5. TESTING

5.1 Introduction

Software Testing is a method is to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software system components using manual or automated toolsto evaluate one or more properties of interest. The purpose of software testing is toidentify errors, gaps, or missing requirements in contrast to actual requirements.

Some prefer saying Software testing definition as a White Box and Black BoxTesting. In simple terms, Software Testing means the Verification of ApplicationUnder Test (AUT). This is Software Testing course introduces testing software to theaudience and justifies the importance of software testing.

Software Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customersatisfaction. Software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of such examples.

5.1.1 Black Box Testing and White Box Testing

5.1.1.1 Black Box testing

The primary source of black-box testing is a specification of requirements that are stated by the customer. It is another type of manual testing. It is a software testing technique that examines the functionality of the software without knowing its internal structure or coding. It does not require programming knowledge of the software. All test cases are designed by considering the input and output of a particular function. In this testing, the test engineer analyses the software against requirements, identifies the defects or bugs, and sends it back to the development team

5.1.1.2 White Box Testing

The term 'white box' is used because of the internal perspective of the system. The clear box or white box, or transparent box name denotes the ability to see through he software's outer shell into its inner workings.

It is performed by Developers, and then the software will be sent to the testing team, where they perform black-box testing. The main objective of white-box testing is to test the application's infrastructure. It is done at lower levels, as it includes unittesting and integration testing. It requires programming knowledge, as it majorly focuses on code structure, paths, conditions, and branches of a program or software.The primary goal of white-box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

5.2 Types of testing

1. Unit Testing

2. Integration Testing

3.System Testing

4.Functional Testing

5. Acceptance Testing

6. Regression Testing

7.Performance Testing

8. Security Testing

9. User Acceptance Testing

1. Unit Testing

Unit testing is a method of testing individual units or components of a software application. It is typically done by developers and is used to ensure that the individual units of the software are working as intended. Unit tests are usually automated and are designed to test specific parts of the code, such as a particular function or method. Unit testing is done at the lowest level of the software development process, where individual units of code are tested in isolation.

2. Integration Testing

Integration testing is a method of testing how different units or components of a software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software are combined. Integration testing is done after unit testing and before functional testing, and is used to verify that the different units of the software work together

3. Regression Testing

Regression testing is a method of testing that is used to ensure that changes made to the software do not introduce new bugs or cause existing functionality to break. It is typically done after changes have been made to the code, such as bug fixes or new features, and is used to verify that the software still works as intended.

4. Alpha Testing

This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.

5. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software.
This version is released for a limited number of users for testing in a real- time environment.

6. System Testing

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.

7. Object-Oriented Testing

This testing is a combination of various testing techniques that help to verify and validate object-oriented software. This testing is done in the following
manner:

Testing of Requirements,
Design and Analysis of Testing,
Testing of Code,
Integration testing,
System testing,

6. OUTPUT SCREENS

MediaPipe Pose is a computer vision system developed by Google that can detect and track human body poses in real-time using machine learning algorithms. The system uses a set of 33 key points, also known as landmarks, to identify the position and orientation of different parts of the human body.

Here's a high-level overview of how MediaPipe Pose works:

1. Input data: The system takes in a video stream or a series of images containinga human body.

2.Preprocessing: The input data is preprocessed to enhance the image quality,remove noise, and the lighting conditions.

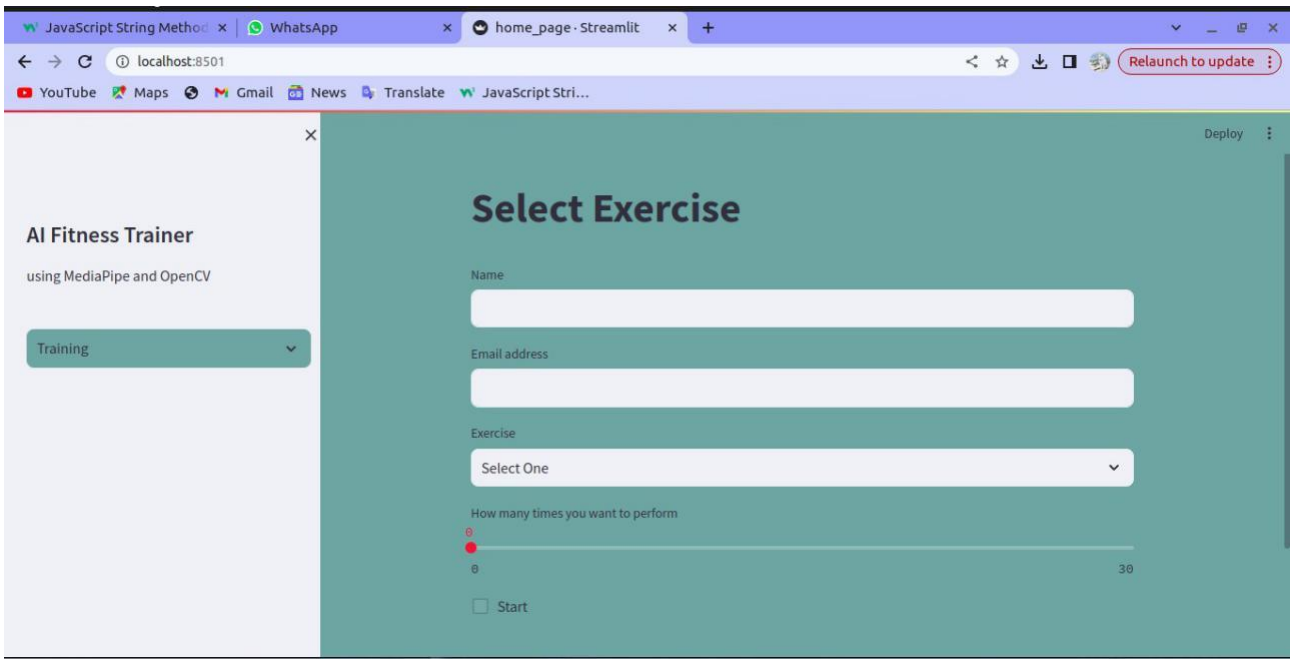
3.Pose estimation: The system uses a deep neural network model trained on thousands of labeled images to estimate the position of the 33 key points on the human body.

4.Pose tracking: The system uses a Kalman filter to track the movements of thekey points over time, reducing jitter and improving the overall accuracy of the pose estimation.

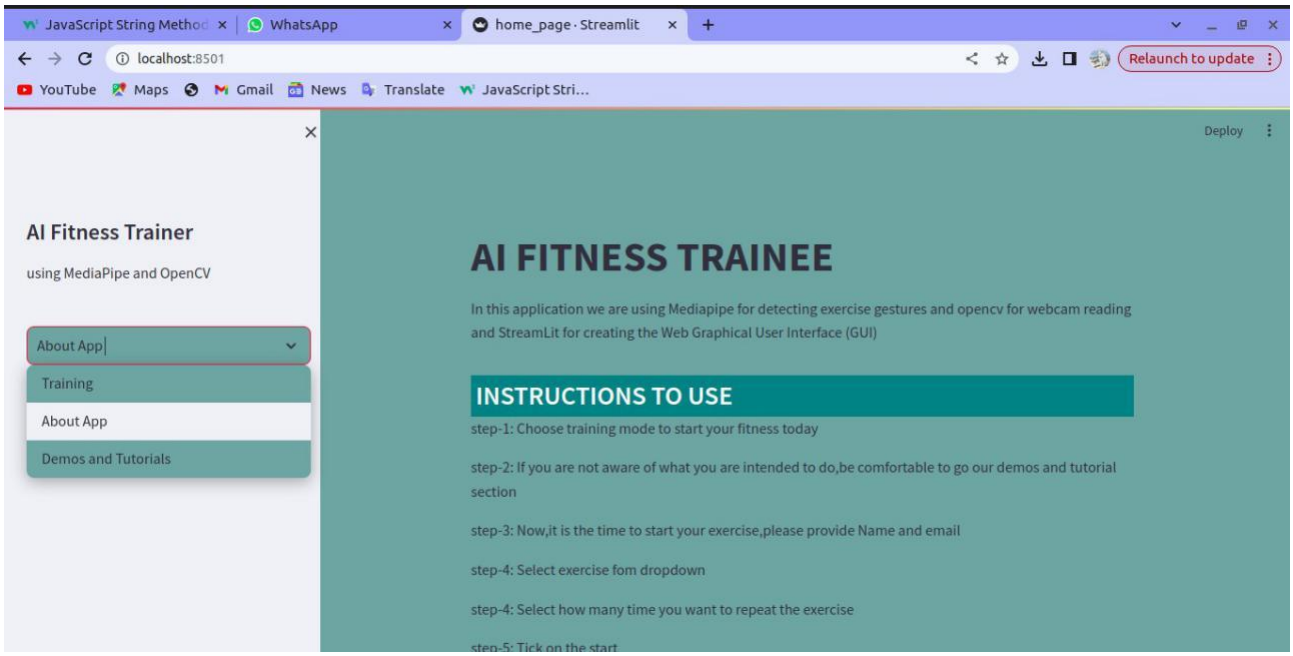
5.Output: The system outputs the estimated pose in real-time as a set of 33 key points, which can be used for various applications such as augmented reality, fitness tracking, or gesture recognition.

Project Execution Demo

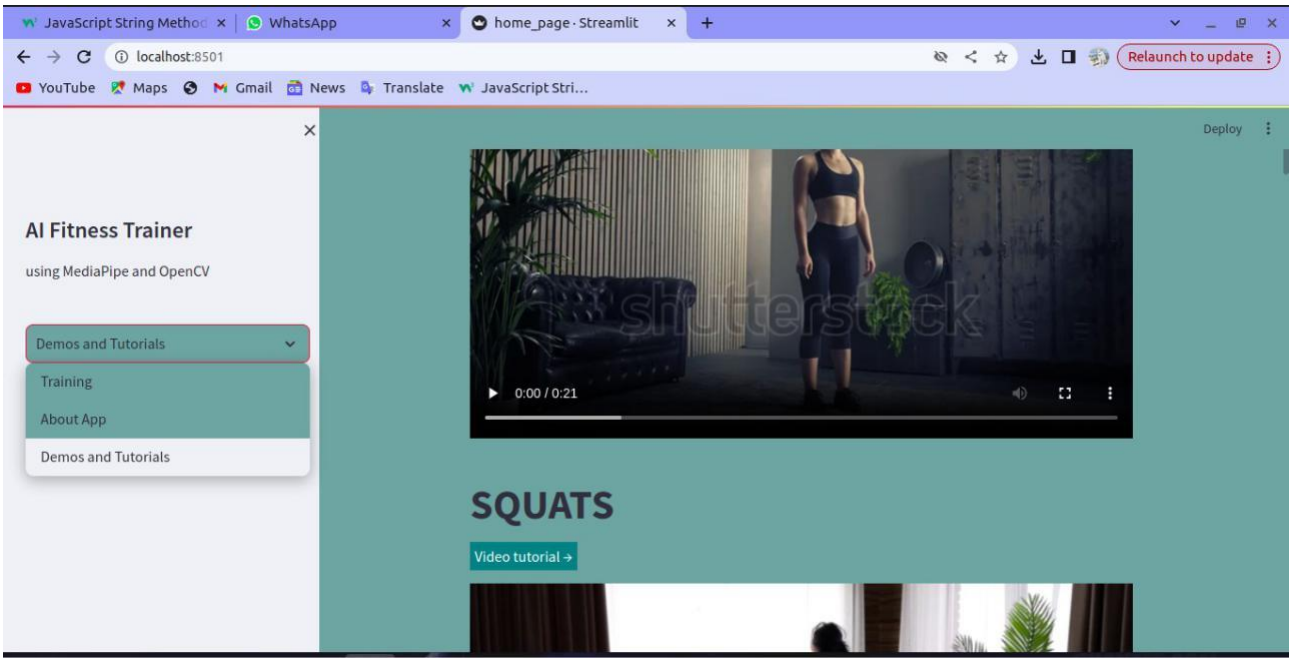
1.Interface



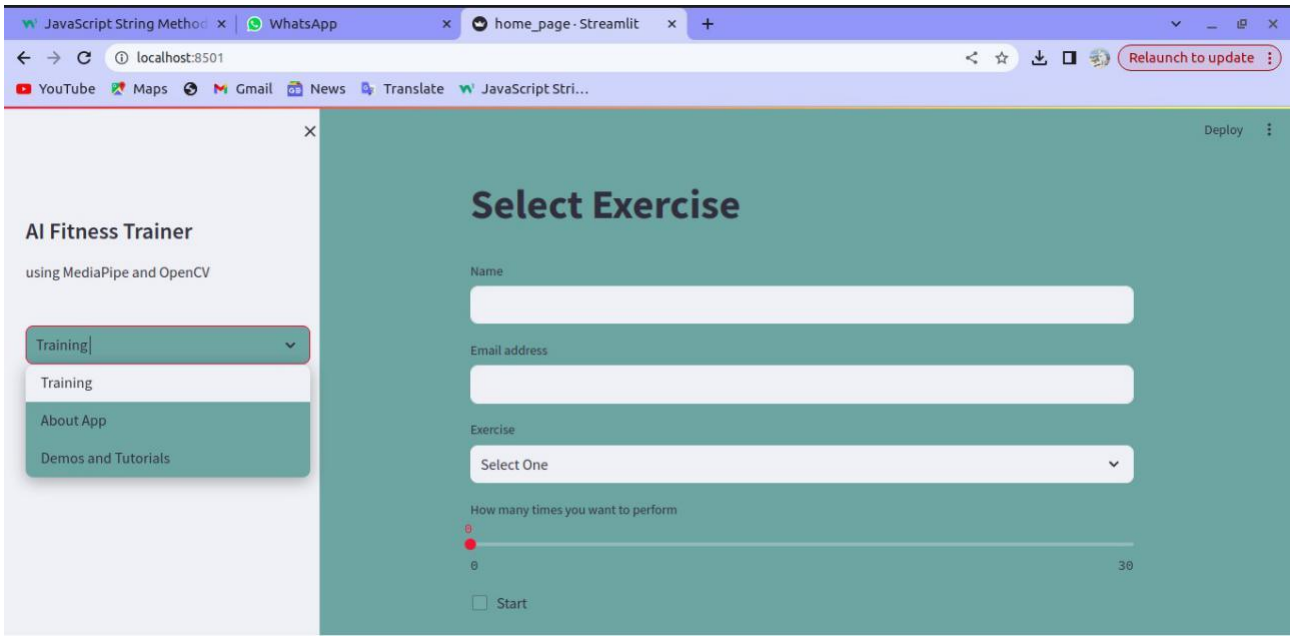
2.About app mode interface.



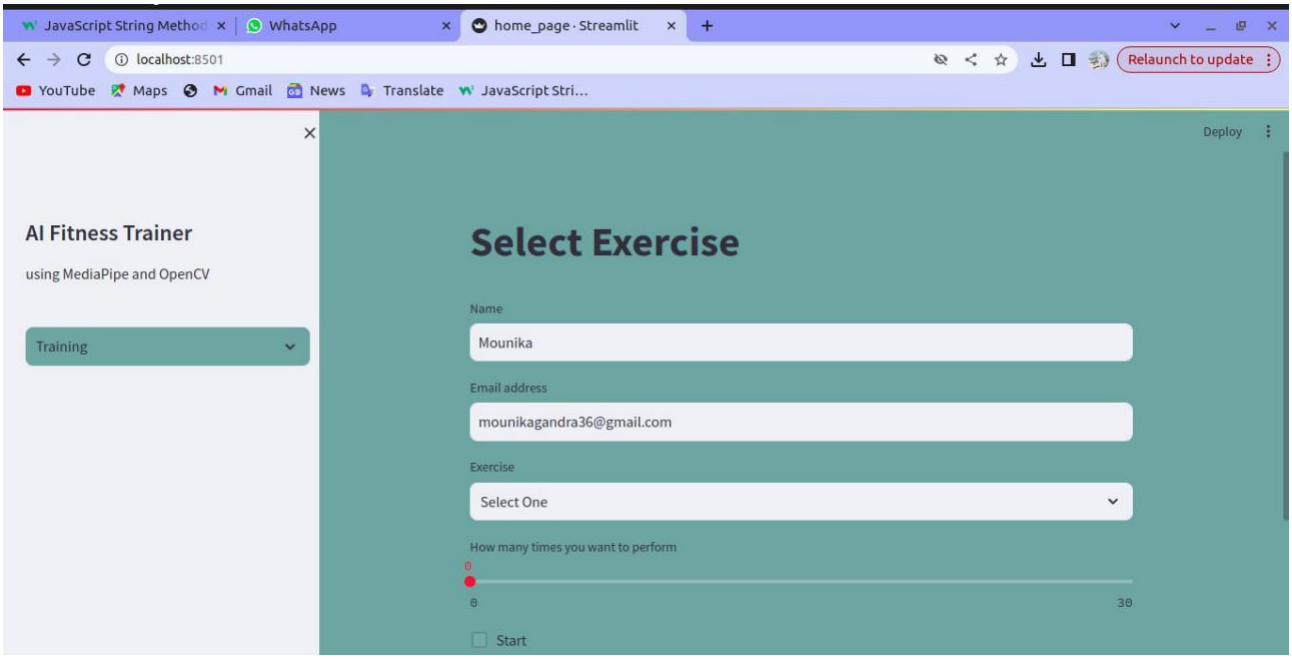
3.Demos and Tutorials mode interface.



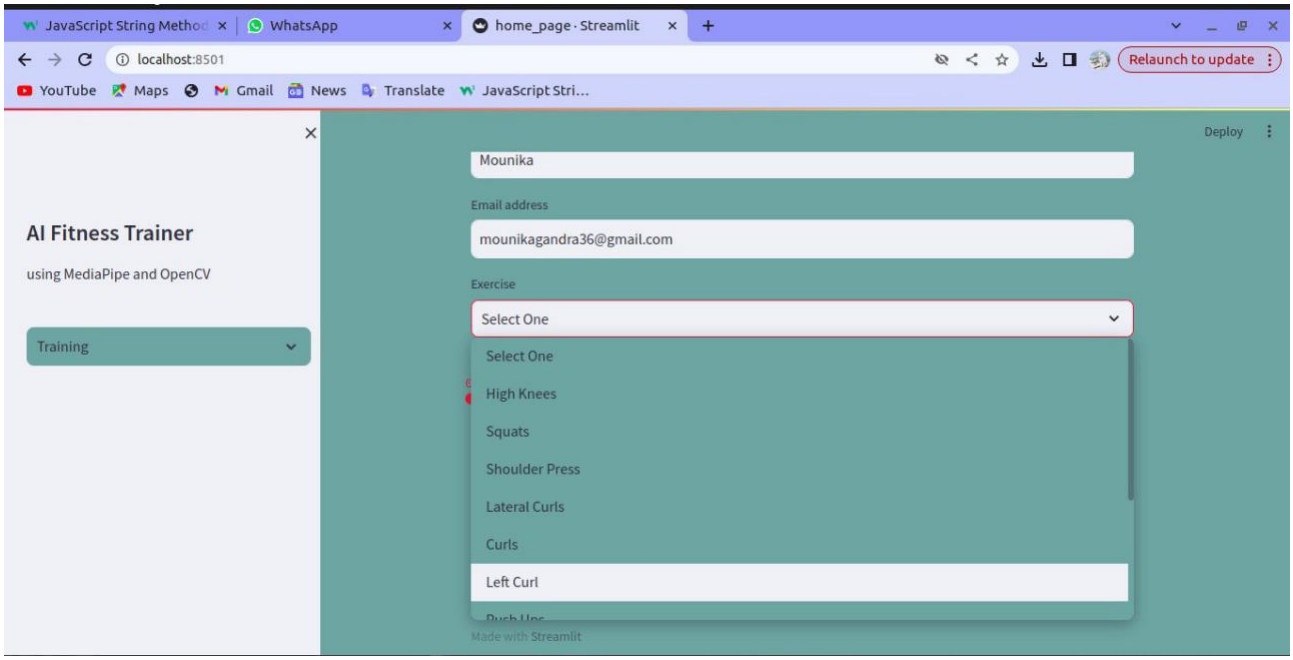
4.Training mode interface.



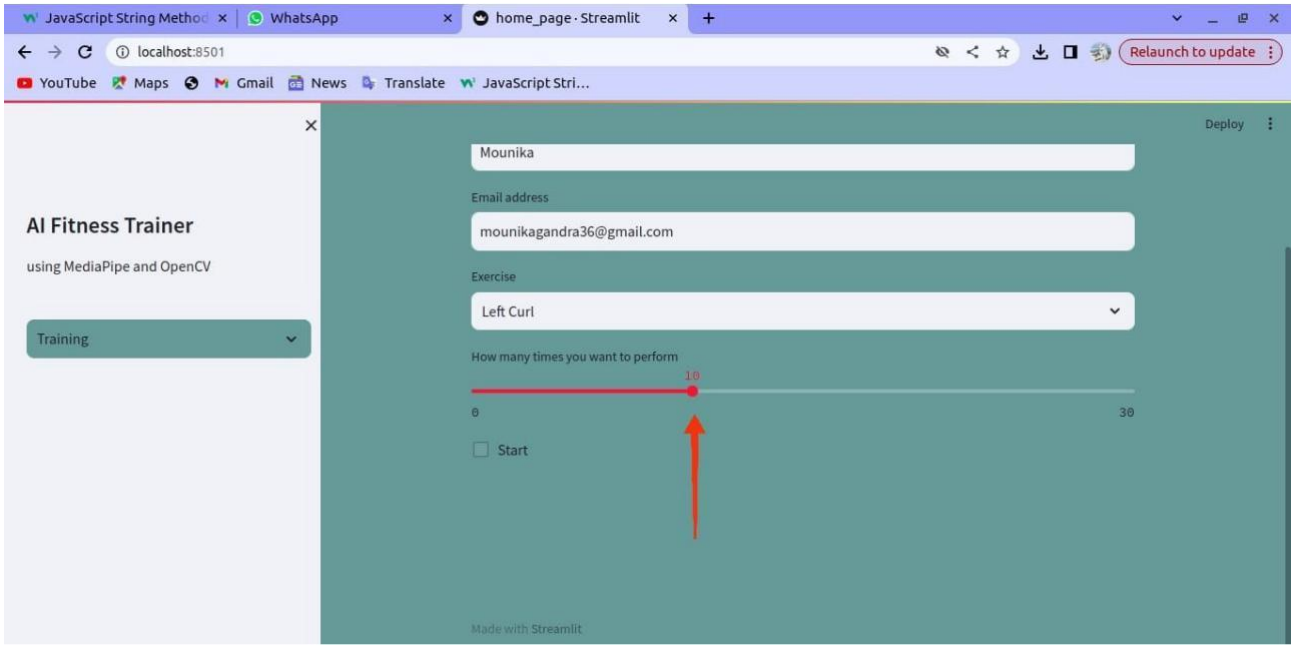
5.Enter details: Name and email



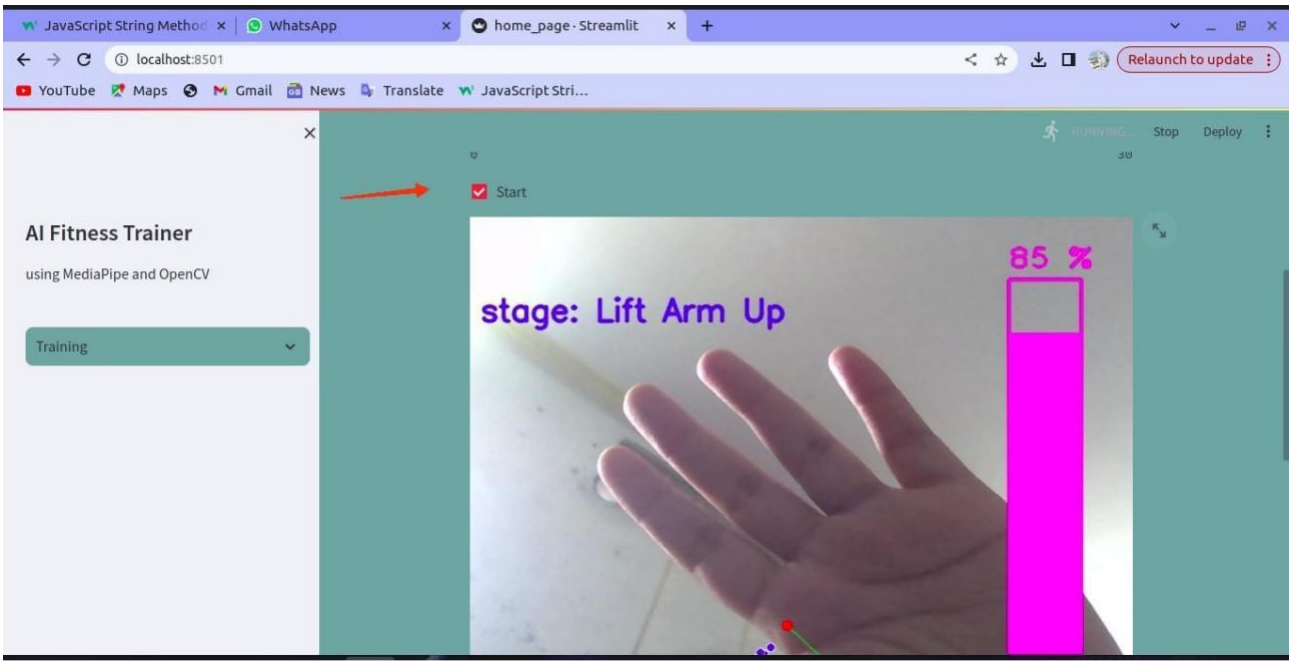
6.Select Exercise



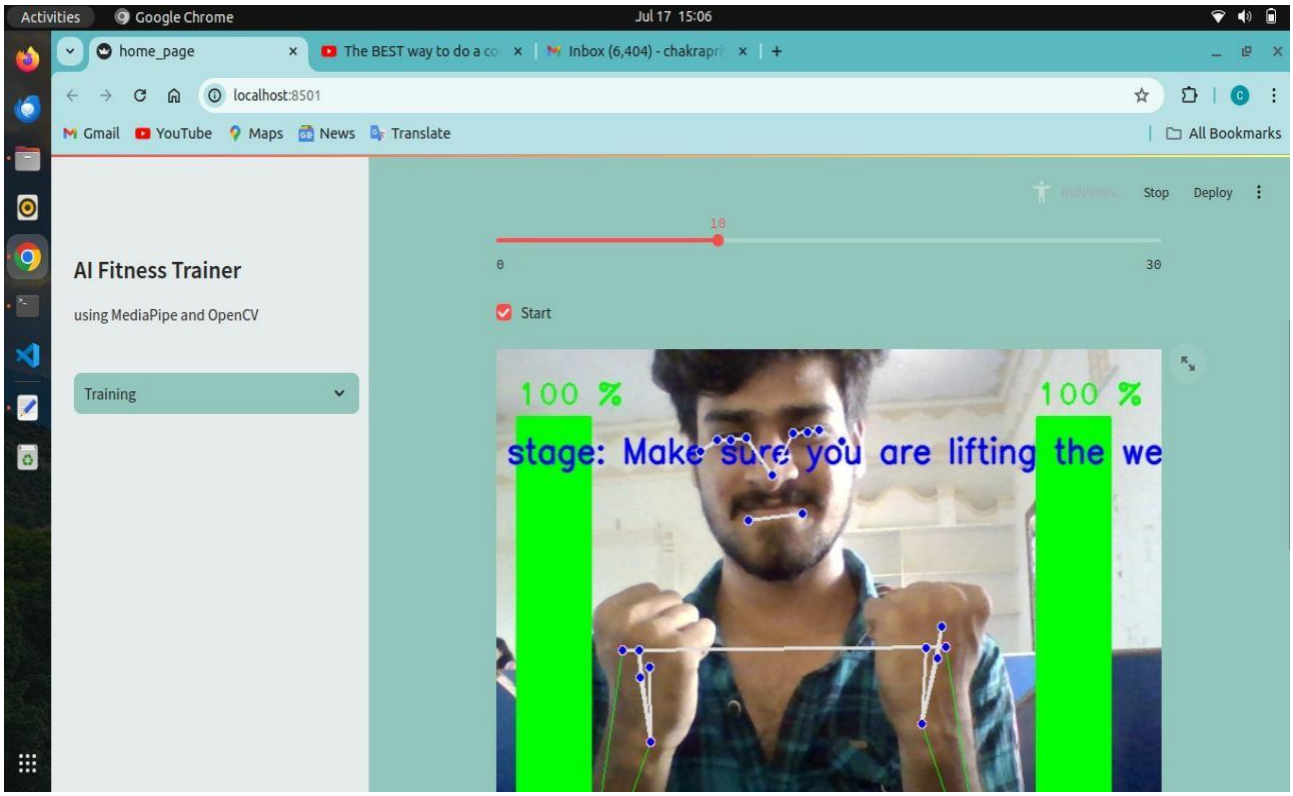
7.Select count



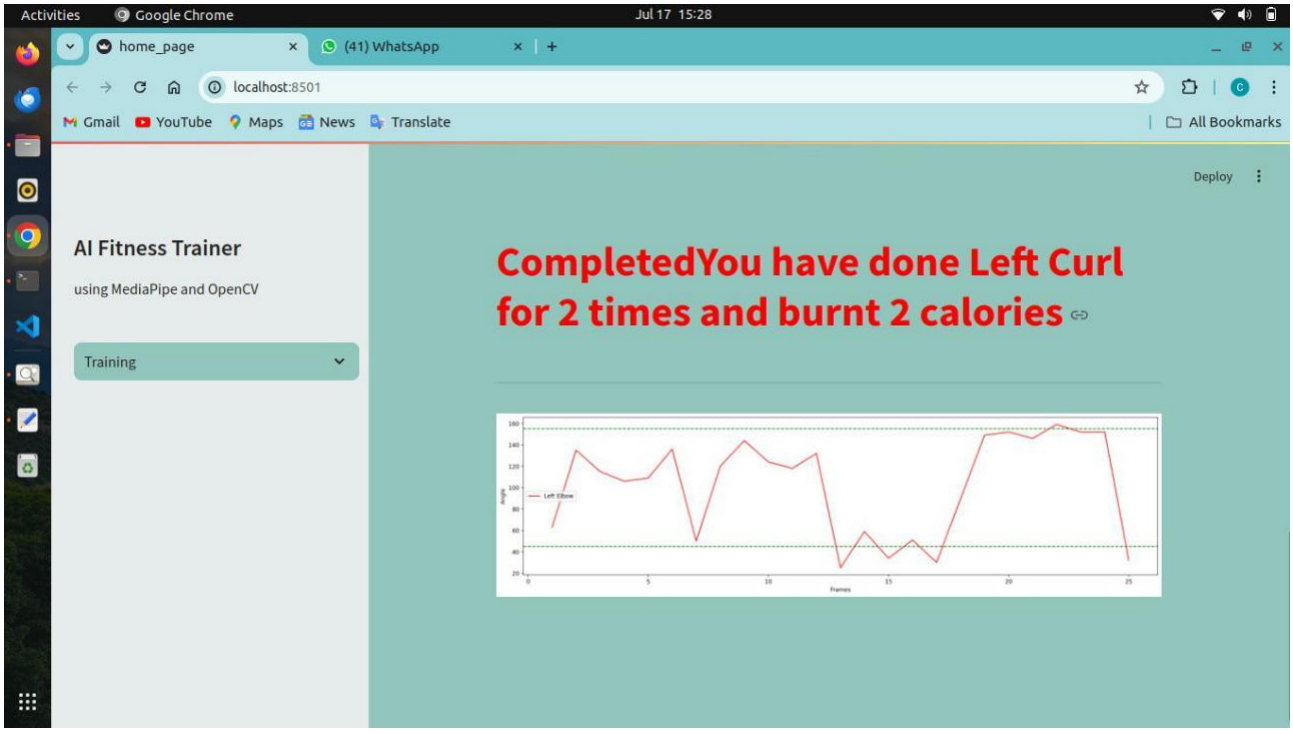
8.Check start button.



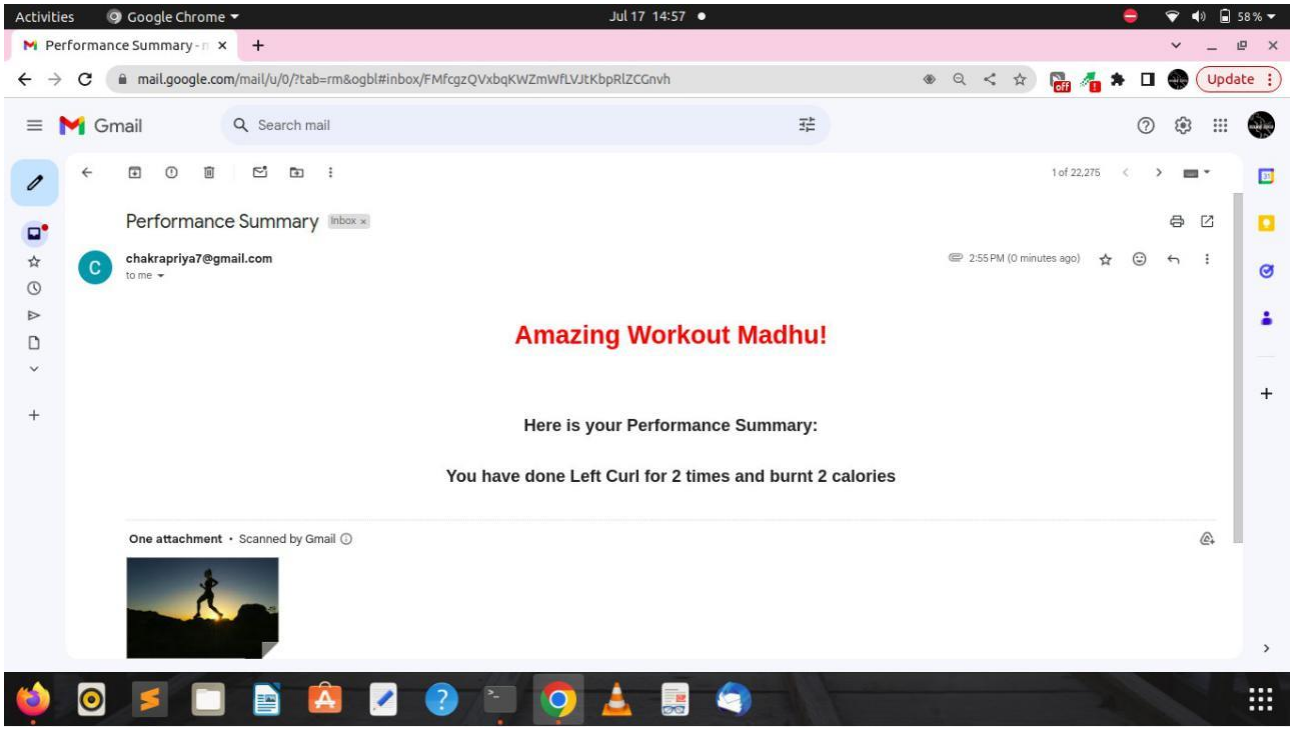
9.start doing exercise



10.Performance graph



11.Performance report to your inbox.



7.CONCLUSION

Nowadays our life is becoming busier and we hardly find time in our schedules to be healthy and fit and exercise daily. This has caused many diseases and health issues. Implementation of Artificial Intelligence in the field of fitness can solve many problems. The health-related applications and devices are making our lives easier andeases our fitness journey. Individuals can use this application in their own workouts ,Hence making them more efficient are less error-prone. There is a lot of scope of development in this project like it can be upgraded to support more exercises. The data collected by the AI Fitness Trainer can be saved and processed for the next sessions. The trainer will suggest you work out plan and its intensity according toyour body type and weight. This application can be developed into a complete android/iOS application for ease of use. The application can not only be used at homes but by increasing the scope can be used in gyms as smart trainers thus reducing the human intervention. Further application will also be able to provide personalized workout and diet plans. This application will offer different styles of yoga to give useran overall fitness regime.

We all know it is very important to exercise with correct form. Nowadays people prefer working out at home. It is very convenient and time saving as well. We can workout anytime we want at home. But we also know that a trainer is very important to keep a track of our exercise. This project is our effort to make an AI based trainer. An AI-powered personal trainer that works simply by pointing a camera at a person completing a workout, and having a human pose estimation model (specific poses related to a workout regimen) indicate whether or not a given exercise has been completed properly.

8.FUTURE ENHANCEMENT

Real-Time Feedback and Corrections:

Implement real-time feedback during the exercises to correct the users pose. The AI could provide visual or auditory cues to guide the users into the correct position, enhancement the effectiveness of the workout.

Adaptive Workouts:

Develop an adaptive system that tailors workouts based on the users performance and progress the ai could adjust exercise difficulty ,duration ,and intensity to ensure continual improvement and prevent plateaus

Augmented Reality(AR) Integration:

Explore AR Technology to provide Users with a more immersive experience.AR could overlay exercise instruction directly onto the users environment. Making it easier for them to follow instructions and maintain proper form.

AI-powered Personal Trainer Conversations:

Develop a natural language processing (NLP) system that follows users to interact with a virtual personal trainer the Al could answer questions ,provide encouragement ,and offer personality advice based on the users fitness journey

In our modern, hectic lives, finding time to prioritize our health and engage in regular exercise has become increasingly challenging. This lack of focus on fitness often leads to various health issues. Our primary objective is to raise awareness about the significance of good health and fitness among the general population and assist them in achieving their wellness goals. By harnessing the power of Artificial Intelligence (AI) and Machine Learning (ML) in the realm of fitness, we can address many of these challenges. Fitness applications and devices have simplified our lives and streamlined our fitness journeys. These tools empower individuals to conveniently perform workouts at home, increasing efficiency and reducing the risk of errors. Throughout this process, we have acquired knowledge on utilizing various Python libraries and packages and have witnessed the immense benefits that machine learning can offer in improving human well-being.

9. REFERENCES

1) https://google.github.io/mediapipe/solutions/pose
2) https://learnopencv.com/introduction-to-mediapipe/
3) “BlazePose: On-device Real-time Body Pose tracking.” V.Bazarevsky, I.Grishchenko, K.Raveendran, T.Zhu, F. Zhang, M.Grundmann.
4) F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.L. Chang and M. Grundmann, “MediaPipe Hands: On#device Real-time Hand Tracking.” ArXiv, 2020
5) Ammar Anuar, Khairul Muzzammil Saipullah, Nurul Atiqa h Ismail, Yewguan Soo “OpenCV Based Real#Time Video Processing Using Android Smartphone” , IJCTEE, Volume 1, Issue 3