MAHARASHTRA STATE BOARD OF

TECHNICAL EDUCATION

Latthe Education Society's Polytechnic, Sangli.

DEPARTMENT OF COMPUTER ENGINEERING

Capstone Project Execution & Report Writing

(22060)

Project Report on

AI –Yoga Application

Under Guidance of

Mr.A.N.MUJAWAR

Submitted By:-

ROLL NO	STUDENT NAME	Enrolment
223345	Harish S. Hulyalkar	2000430230
223346	Vaibhav S. Rajput	2000430209
223347	Ayan M. Mujawar	2000430216
223348	Pratik P. Kale	2000430222

Academic Year: 2022-2023

Latthe Education Society's Polytechnic, Sangli.

Department of Computer Engineering

CERTIFICATE

This is to certify **that Mr.Harish Hulyalkar, Mr.Vaibhav Rajput, Mr.Ayan Mujawar, Mr.Pratik Kale** from (Institute) **Latthe Education Society's Polytechnic College** having Enrolment No**: 2000430230, 2000430209, 2000430216, 2000430222** has completed **Project** title **AI-Yoga Application** in a group consisting of **4** candidates under the guidance of theFaculty Guide.

ROLL NO	STUDENT NAME	Enrolment no.	Seat no.
223345	Harish S. Hulyalkar	2000430230	172973
223346	Vaibhav S. Rajput	2000430209	172961
223347	Ayan M. Mujawar	2000430216	172966
223348	Pratik P. Kale	2000430222	172955

Project Guide	**Head of the Department**	**Principal**
(Mr.A.N.Mujawar)	(Mrs. M.S. Ranade)	(Mr. A.B. Gaji)

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our respected guide **Mr.Aftab N. Mujawar Sir** for their valuable help and guidance. We thankful to them for encouragement they have giventowards the planning of our capstone project.

We are also grateful to respected **Mrs. Medha S. Ranade** (HOD, CO) and to our respected principal sir **Mr. Annasaheb Gaji** for permitting us to utilize all the necessary faculties of our institution.

We are also thankful of all the other faculties and staff members of our department for their kind co-operation and help.

Lastly, we would like to express our deep appreciation towards our classmates and indebtedness to our parents for providing us the moral support and encouragement.

DECLARATION

I, **Mr.Harish Hulyalkar, Vaibhav Rajput,Ayan Mujawar,Pratik Kale** bearing Enrollment no. **2000430230, 2000430209, 2000430236, 2000430222** studying in VI semester of Diploma in Computer Engineering atLESP, Sangli, declare that the work of Capstone project entitle "**AI-Yoga Application**" which is being submitted in the fulfillment for the award of the diploma in Computer Engineering of MSBTE is an original carried out by me during the academic year 2022-23 under the guidance of Mrs. P. D. Patil, Lecturer, Dept. of Computer Engg., LESP, Sangli.

I further declare that the work of similar topic or the matter embodied in the Capstone project report has notbeen submitted previously for the award of any diploma by me to any other university or institution.

Date: Signature

Place: (name of student)

The CPE viva-voce examination of Ms. /Mr._____Was conducted on Date_____

	Name	Signature
Internal examiner:-	**Mr.A.N.Mujawar**	……..

Abstract

Yoga has many benefits for physical and mental health, but not everyone has access to yoga studios or teachers. That's why we've created an AI-powered yoga Android application that allows users to practice yoga from the comfort of their homes.
Our AI yoga application features a user-friendly interface that makes it easy to choose from a variety of yoga asanas tailored to your skill level and preferences. Whether you're a beginner or an experienced yogi, our AI yoga application can help you improve your yoga practice.

In addition to providing a wide range of yoga practices, our application uses artificial intelligence to track co-ordinates and provide personalized recommendations based on your performance. With our AI-powered yoga application, you can receive feedback on your poses and overall performance, helping you to improve your yoga practice over time.

By using our AI yoga application, you can enjoy the benefits of yoga without having to leave your home or pay for expensive classes. Whether you're looking to improve your flexibility, reduce stress, or simply enjoy a relaxing yoga practice

INDEX

Chapter -1

Introduction

AI-yoga is an application that uses artificial intelligence (AI) technology to help people practice yoga more effectively. The app uses computer vision algorithms to detect the coordinates of the user's body during a yoga pose and provides feedback on the correct alignment. The AI-yoga app uses the camera on the user's device to capture a video of the user performing a yoga pose. The computer vision algorithms then analyze the video and detect the coordinates of the user's body parts. The app then compares the user's body position to the correct alignment for the pose and provides feedback on any deviations from the correct form.

To provide feedback, the app displays a timer that starts when the user's body is in the correct position and stops when the user's form deviates from the correct alignment. The app also displays a skeletal line graphic overlay on the video feed to show the user how their body is aligned in relation to the correct form. In addition to providing feedback on alignment, the AI-yoga app can also suggest modifications or adjustments to help the user improve their form. The app can also track the user's progress over time and provide personalized recommendations for poses or sequences to help the user meet their goals. The app can also display a reference image at the top of the screen while the user is performing the asanas. This reference image shows the correct alignment for the pose and serves as a visual guide for the user to follow.

The AI-yoga app uses Tensorflow, an open-source machine learning framework developed by Google, to power its computer vision algorithms. Tensorflow is one of the most popular frameworks for building and training machine learning models, and it is particularly well-suited for image and video recognition tasks.

The computer vision algorithms in the AI-yoga app use Tensorflow to detect the coordinates of the user's body parts and compare them to the correct alignment for the pose. The algorithms are trained on a large dataset of yoga poses and sequences, which allows the app to accurately recognize and provide feedback on a wide variety of poses.

The use of Tensorflow in the AI-yoga app is a significant advantage because it allows the app to continually improve and adapt to user feedback. As more users use the app and provide feedback on their experience, the app can use this data to refine its computer vision algorithms and improve its accuracy and effectiveness.

In addition to Tensorflow, the AI-yoga app likely uses other machine learning and computer vision technologies, such as Convolutional Neural Networks (CNNs) and OpenCV, to power its algorithms and provide accurate feedback to users. By leveraging the latest in machine learning and computer vision research, the app is able to provide a cutting-edge yoga experience that is personalized and effective.

Overall, the AI-yoga app is a useful tool for yoga practitioners who want to improve their form and get more out of their yoga practice. With its advanced computer vision algorithms and personalized feedback, the app can help users achieve their yoga goals more effectively and efficiently.

Chapter -2

Literature Survey

1. SMART BADUANJIN

Smart Baduanjin is a modern, technology-based approach to the traditional Chinese qigong exercise known as Baduanjin or Eight-Section Brocade. The aim of Smart Baduanjin is to enhance the health benefits of the exercise by incorporating wearable sensors and other technologies to monitor and analyze the user's movements, breathing, heart rate, and other physiological data.

The Smart Baduanjin group used wearable sensors to monitor their movements and receive real-time feedback on their performance, while the traditional group received verbal instructions and feedback from an instructor.

Features:

- Smart Baduanjin typically includes a mobile app that the user can use to access the exercise program, track their progress, and receive feedback on their performance.
- Some Smart Baduanjin systems include a virtual coach that uses artificial intelligence and machine learning to provide personalized feedback and coaching to the user.
- Smart Baduanjin may also include features for monitoring the user's health, such as heart rate monitoring, breathing rate monitoring, and sleep tracking.

2. BMW GESTURE RECOGNITION

BMW gesture recognition is a technology that allows drivers to control certain functions of their BMW vehicle using hand gestures, without the need to physically interact with buttons or touchscreens. The system uses a 3D sensor to detect and interpret the driver's hand gestures, and translates them into commands that control features such as audio volume, phone calls, and navigation.

The results of the study showed that BMW gesture recognition was usable and user-friendly for older drivers, and that it improved their driving experience by reducing the need for physical interaction with the vehicle's controls. The researchers concluded that gesture recognition has the potential to be an effective and convenient way to interact with in-vehicle technology, particularly for older drivers who may have difficulty with traditional controls.

Features:

- BMW gesture recognition allows drivers to control certain functions of their vehicle using hand gestures, without the need to physically interact with buttons or touchscreens.
- BMW gesture recognition uses 3D sensor technology to detect and interpret the driver's hand gestures.
- BMW gesture recognition allows drivers to customize the hand gestures used to control different functions of the vehicle.

3. MEDICAL OPERATIONS USING GESTURES

Medical operations using gestures refers to the use of gesture recognition technology to control medical devices and perform surgical procedures. This technology has the potential to improve surgical accuracy and efficiency, while also reducing the risk of infection and other complications.

Medical operations using gestures have the potential to improve surgical accuracy, efficiency, and safety. By allowing surgeons to control medical devices using intuitive hand gestures, gesture recognition technology can reduce the risk of infection and other complications, while also improving surgical outcomes and patient outcomes.

Features:

- Medical operation using gestures allows for hands-free control of medical devices and equipment, which can improve safety and reduce the risk of infection and other complications.

- Gesture recognition technology allows for a more intuitive and natural user interface, with surgeons using hand gestures to control medical devices and perform surgical procedures.

- Gesture recognition technology has the potential to improve surgical accuracy by allowing for more precise control of medical devices and equipment. Completely digitized financial transactions

4. SOCIAL MEDIA FILTERS:-

Social media filters are a popular feature on platforms such as Snapchat and Instagram, allowing users to add animated overlays to their photos and videos. Many of these filters use gesture recognition technology to track the user's movements and respond with animated effects in real-time.

The results of the study showed that gesture recognition technology was effective in enhancing the usability and user experience of the social media app. Participants reported that the use of hand gestures to control the app's filters felt intuitive and natural, and that the real-time response of the app's animations added to the overall fun and engaging experience.

Features:

- Social media filters using gestures rely on real-time gesture recognition technology to track the user's movements and apply the desired filter or effect.
- Social media filters using gestures typically rely on simple and intuitive hand or facial gestures, making them easy for users to learn and use.
- Some social media filter apps allow users to customize the gestures they use to apply filters and effects, allowing them to create a personalized and unique user experience.

Chapter -3

3.1) Problem Definition:

Yoga is a popular form of exercise that can help improve physical fitness, mental well-being, and overall health.

3.2) Scope:

In today's fast-paced world, people are leading sedentary lifestyles, which has led to various health issues like stress, anxiety, and obesity. To counteract these health problems, people are turning to yoga, which is an ancient practice that combines physical postures, breathing techniques, and meditation to improve overall health and well-being.

However, due to the busy schedules, people often find it challenging to attend regular yoga classes. To solve this problem, we have created an AI yoga application that can be accessed from anywhere and anytime. The AI yoga application is designed to provide personalized yoga sessions to individuals based on their specific needs and goals. The AI yoga application offers a range of yoga sessions that cater to different levels of expertise, from beginners to advanced practitioners. The application uses machine learning algorithms to analyze the user's performance and provide personalized feedback to improve their practice.

Additionally, the AI yoga application offers various meditation and breathing exercises that are designed to reduce stress and anxiety. These exercises are tailored to the individual's specific needs and can be accessed anytime, anywhere .Using the AI yoga application not only saves time and money, but it also offers a convenient and effective way to improve overall health and well-being. With regular use, individuals can expect to see improvements in their flexibility, strength, and overall mental and physical health.

3.3) Objective:

- Providing users with personalized feedback on their form and technique, allowing them to improve their yoga practice and avoid injury.
- Allowing users to practice yoga from the comfort of their own home, at any time that suits their schedule.
- Offering customized yoga sequences and poses based on the user's fitness level, goals, and preferences.
- Creating a more engaging and immersive experience that encourages users to practice yoga more regularly and consistently.
- Using yoga as a tool to promote physical fitness, stress reduction, and overall mental well-being.

.

Chapter -4

Methodology

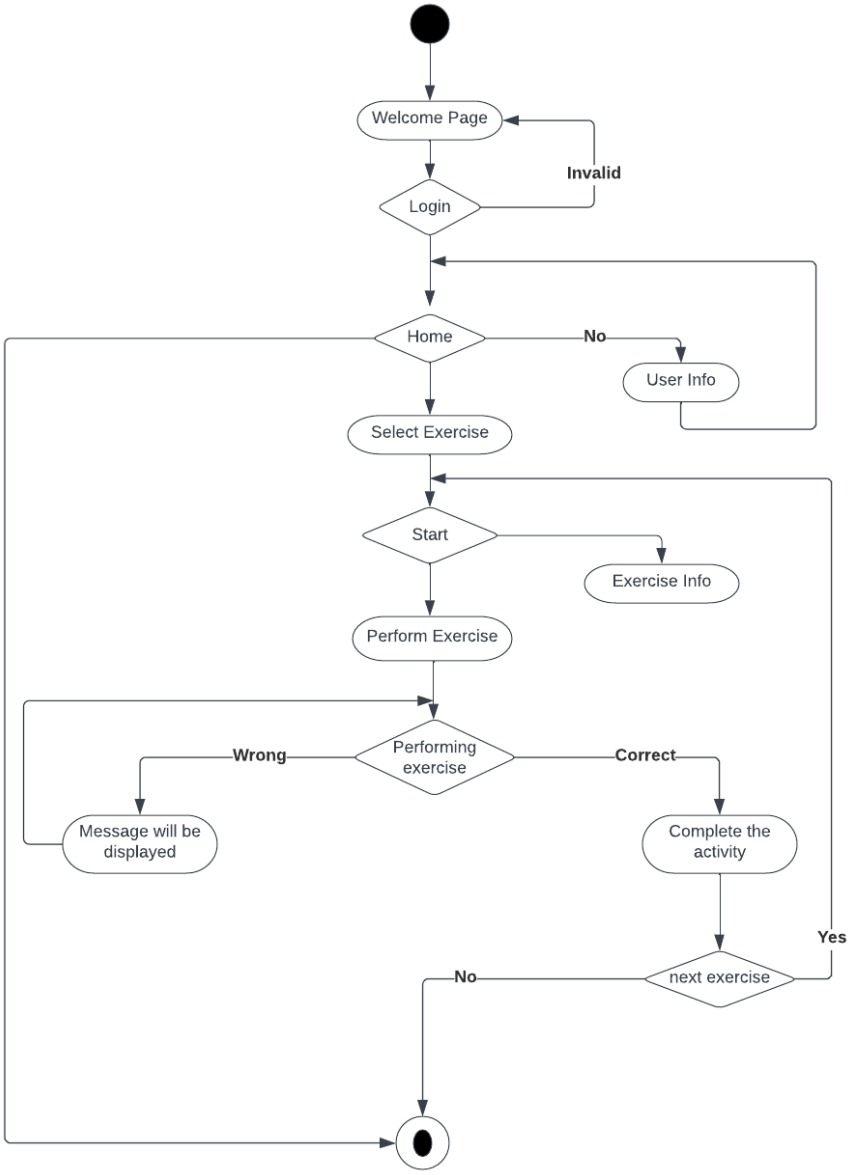


Fig4.1 Activity Diagram.

4.1) Activity Diagram:-

An activity diagram is a type of UML (Unified Modeling Language) diagram that illustrates the flow of activities or processes in a system. It shows the steps involved in a process or use case, and the actions taken at each step. The diagram is made up of nodes, which represent the activities or actions, and edges, which connect the nodes to show the flow of the process.

Chapter- 5

Details of designs, working and processes

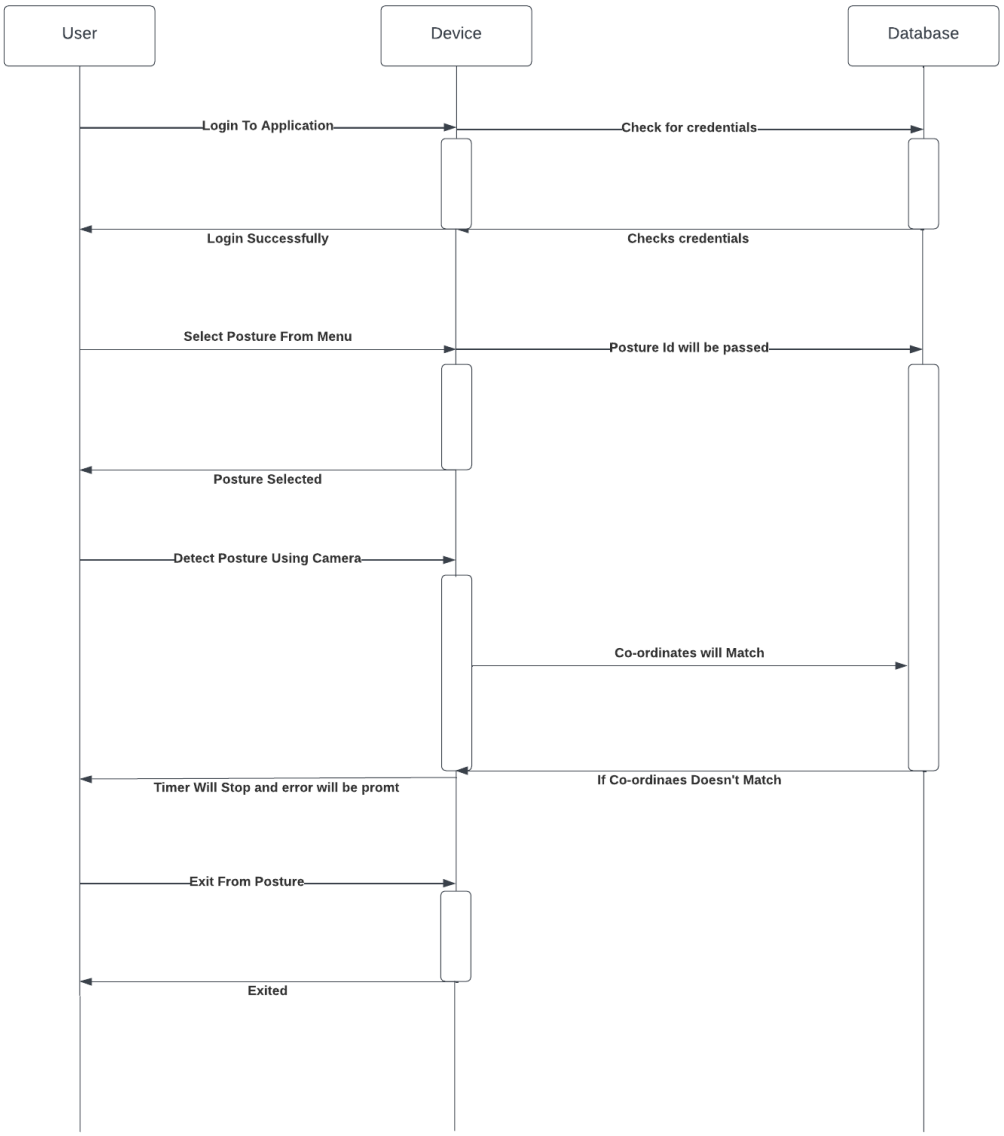


Fig. 5.1: Sequence Diagram

5.1) Sequence Diagram:-

A sequence diagram is a type of interaction diagram that shows the interactions between objects in a system, arranged in a chronological sequence. It depicts the objects and messages passed between them. Here, interaction between each and every components in our system will be shown.

5.4) Data Flow Diagram:-

- **Zero Level DFD :-**

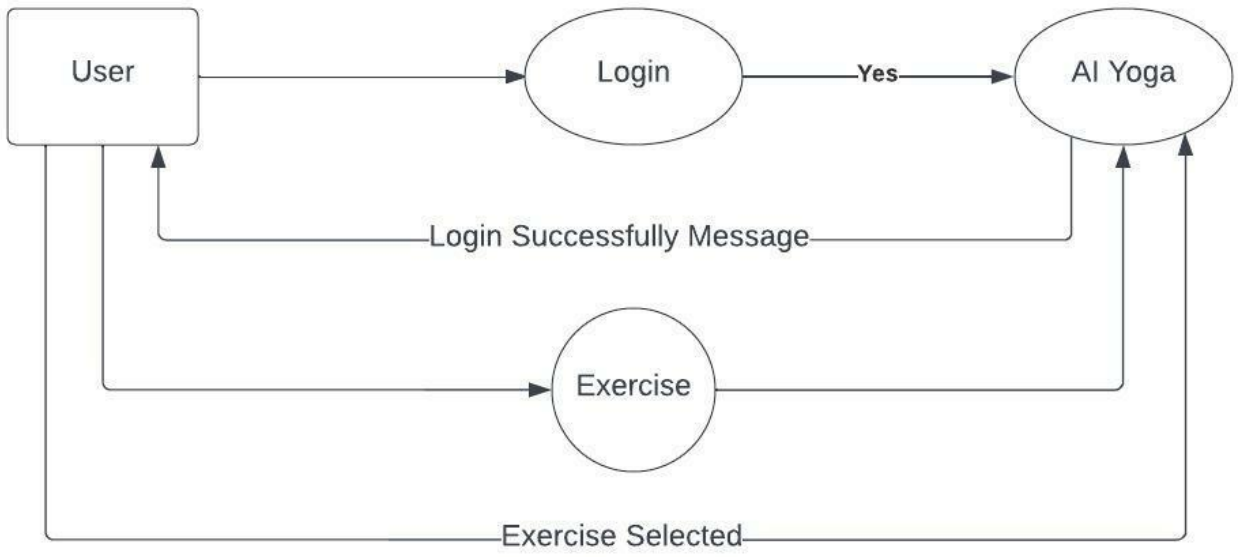


Fig. 5.2.1 Zero level DFD

A zero-level data flow diagram provides an overview of the high-level process of a system, whereas AI-Yoga is a system that uses AI and machine learning technology to provide personalized feedback and coaching to users during their yoga practice.

- **First Level Data Flow Diagram:-**

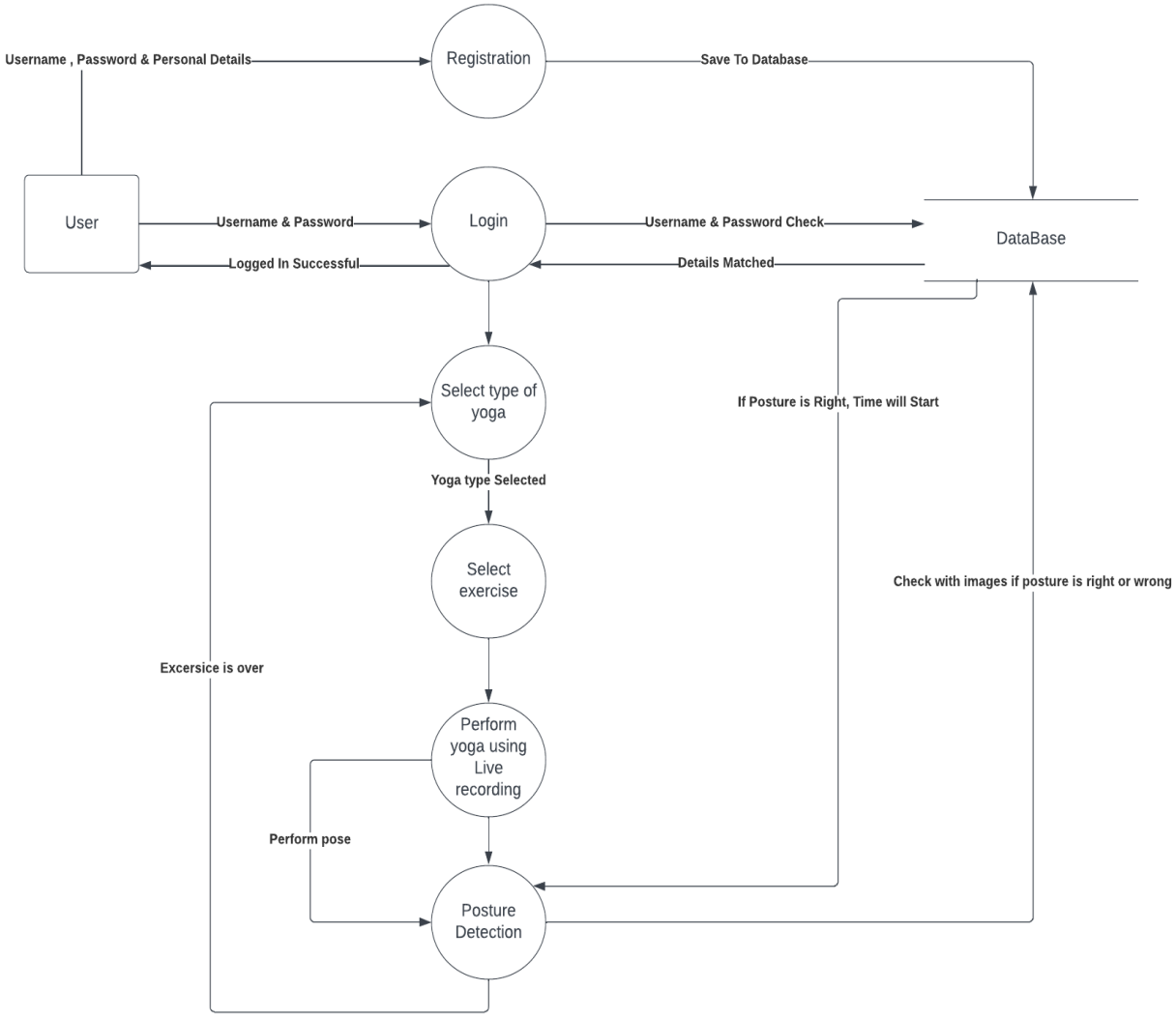
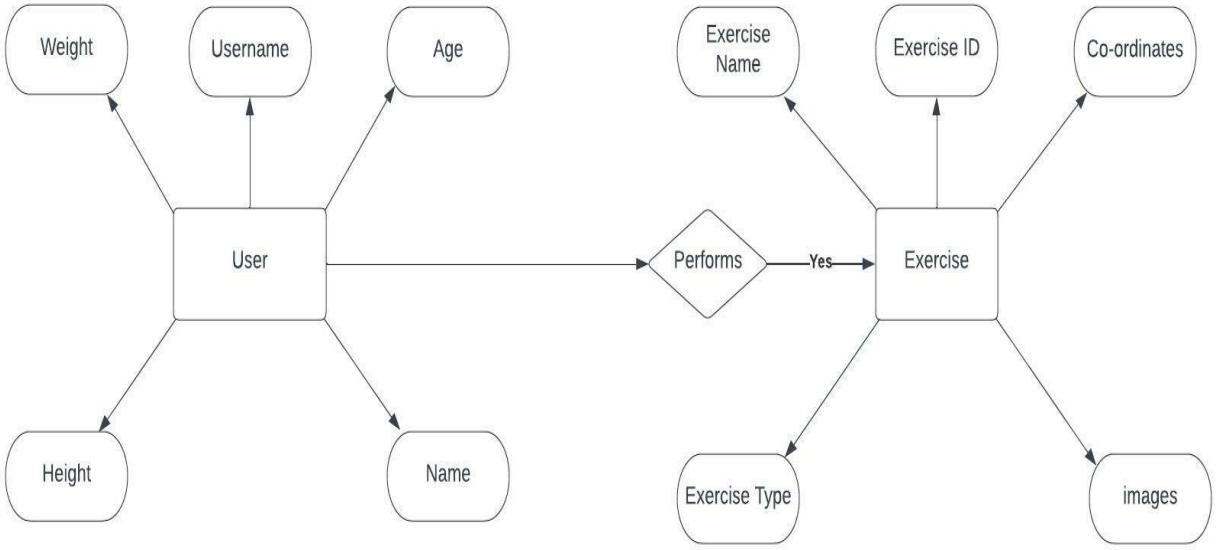


Fig 5.2.2 First Level Data Flow Diagram

First level Data Flow Diagram of AI-Yoga shows how the system is divided into sub-system, each of which deals with one or more of the data flows, and which together provider all of the functionality to system. It also identifies internal data stores of Login, Playback credentials that must be present in order for renewable system to do its job, and shows the flow of the data between the various parts of user of the system. Data Flow Diagram level 1 provides a more detailed breakout of Pieces of the 1st Level DFD.

- **ER Diagram (Entity Relationship):**



5.1) Entity Relationship Diagram

Entity relationship diagram is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology system. It show the relationship in between all modules in our system.

Chapter – 6

Resources Required

S. R.	Requirements	Specification	Remark
1	**Computer system with broad Specification**	Intel® Core ™ I3-4136 with 8GB RAM and 500 GB SSD	
2	**Software**	Android Studio, VS Code	
3	**Any other resources used**	Laser Printer	
4	**Printing Material**	Printing Papers, ink, toner	

Chapter -7

Functionality Code and Application

7.1) Android studio Functionality code-

1. Activity_begginer.xml

```
<RelativeLayout
        android:id="@+id/relative10"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/relative9"
        android:padding="20dp">

        <Button
           android:id="@+id/next"
           android:layout_alignParentRight="true"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Next"
           android:textColor="@color/blue_bg" />

        <Button
           android:id="@+id/back_2"
           android:layout_alignParentLeft="true"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Back"
           android:textColor="@color/blue_bg" />

    </RelativeLayout>
```

2. Begineer.java

```
package com.example.ai_yoga;

    import androidx.appcompat.app.ActionBar;
    import androidx.appcompat.app.AppCompatActivity;

    import android.annotation.SuppressLint;
    import android.content.Intent;
    import android.net.Uri;
    import android.os.Bundle;
    import android.view.View;
    import android.widget.Button;

    public class beginner_1 extends AppCompatActivity {
    private Button ic_btn1;
    private Button next;
       @SuppressLint("MissingInflatedId")
       @Override
       protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_begineer_1);


          ActionBar actionbar = getSupportActionBar();
          actionbar.hide();

          ic_btn1 = findViewById(R.id.back_2);
          next = findViewById(R.id.next);

          ic_btn1.setOnClickListener(new View.OnClickListener() {
             @Override
             public void onClick(View view) {
             Intent intent = new Intent(beginner_1.this, homescreen.class );
             startActivity(intent);
           }
        });

          next.setOnClickListener(new View.OnClickListener() {
             @Override
             public void onClick(View view) {
                Intent intent1 = new Intent(beginner_1.this, WebView.class);
                startActivity(intent1);

             }
          });
      }
     }
```

3. WebView.java

```
package com.example.ai_yoga;

    import androidx.appcompat.app.ActionBar;
    import androidx.appcompat.app.AppCompatActivity;
    import androidx.core.app.ActivityCompat;
    import androidx.core.content.ContextCompat;

    import android.annotation.SuppressLint;
    import android.content.pm.PackageManager;
    import android.os.Build;
    import android.os.Bundle;
    import android.view.KeyEvent;

    import android.webkit.PermissionRequest;
    import android.webkit.WebChromeClient;
    import android.webkit.WebSettings;
    import android.webkit.WebViewClient;

    public class WebView extends AppCompatActivity {
       private android.webkit.WebView web;



       @SuppressLint("MissingInflatedId")
       @Override
       protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_web_view);
          web = findViewById(R.id.webview);

          WebSettings webSettings = web.getSettings();
          webSettings.setJavaScriptEnabled(true);
          web.setWebViewClient(new Callback());
          web.getSettings().setJavaScriptEnabled(true);
          web.getSettings().setAllowFileAccess(true);
          web.getSettings().setAllowContentAccess(true);
          web.getSettings().setAllowFileAccessFromFileURLs(true);
          web.getSettings().setAllowUniversalAccessFromFileURLs(true);
          web.getSettings().setMediaPlaybackRequiresUserGesture(false);
          web.setWebChromeClient(new WebChromeClient() {
             @Override
             public void onPermissionRequest(final PermissionRequest request) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
                   request.grant(request.getResources());


                }
```

```
            }
        });

        web.loadUrl("https://tree.aiyoga.tech/");

        ActionBar actionbar = getSupportActionBar();
        actionbar.hide();



    }

    private class Callback extends WebViewClient {
        @Override
        public boolean shouldOverrideKeyEvent(android.webkit.WebView view, KeyEvent
event) {
            return false;
        }
    }
}
```

7.2) Functionality Code:

React.js:-

```
import * as poseDetection from '@tensorflow-models/pose-detection';
import * as tf from '@tensorflow/tfjs';
import React, { useRef, useState, useEffect } from 'react'
import Webcam from 'react-webcam'
import { count } from '../../utils/music';
import './Tree.css'
import { poseImages } from '../../utils/pose_images';
import { POINTS, keypointConnections } from '../../utils/data';
import { drawPoint, drawSegment } from '../../utils/helper'

let skeletonColor = 'rgb(255,255,255)'
let interval
let flag = false

function Yoga() {
    const webcamRef = useRef(null)
    const canvasRef = useRef(null)
    const [startingTime, setStartingTime] = useState(0)
    const [currentTime, setCurrentTime] = useState(0)
    const [poseTime, setPoseTime] = useState(0)
    const [bestPerform, setBestPerform] = useState(0)
    const [currentPose, setCurrentPose] = useState('Tree')
    const [isStartPose, setIsStartPose] = useState(false)

    useEffect(() => {
        const timeDiff = (currentTime - startingTime) / 1000
        if (flag) {
            setPoseTime(timeDiff)
        }
        if ((currentTime - startingTime) / 1000 > bestPerform) {
            setBestPerform(timeDiff)
        }
    }, [currentTime])
```

```
useEffect(() => {
   setCurrentTime(0)
   setPoseTime(0)
   setBestPerform(0)
}, [currentPose])

const CLASS_NO = {
   Tree: 6,
 }

function get_center_point(landmarks, left_bodypart, right_bodypart) {
   let left = tf.gather(landmarks, left_bodypart, 1)
   let right = tf.gather(landmarks, right_bodypart, 1)
   const center = tf.add(tf.mul(left, 0.5), tf.mul(right, 0.5))
   return center
}

function get_pose_size(landmarks, torso_size_multiplier = 2.5) {
   let hips_center = get_center_point(landmarks, POINTS.LEFT_HIP, POINTS.RIGHT_HIP)
   let shoulders_center = get_center_point(landmarks, POINTS.LEFT_SHOULDER,
  POINTS.RIGHT_SHOULDER)
   let torso_size = tf.norm(tf.sub(shoulders_center, hips_center))
   let pose_center_new = get_center_point(landmarks, POINTS.LEFT_HIP, POINTS.RIGHT_HIP)
   pose_center_new = tf.expandDims(pose_center_new, 1)

   pose_center_new = tf.broadcastTo(pose_center_new, [1, 17, 2])
      // return: shape(17,2)
   let d = tf.gather(tf.sub(landmarks, pose_center_new), 0, 0)
   let max_dist = tf.max(tf.norm(d, 'euclidean', 0))

   // normalize scale
   let pose_size = tf.maximum(tf.mul(torso_size, torso_size_multiplier), max_dist)
   return pose_size
}

function normalize_pose_landmarks(landmarks) {
   let pose_center = get_center_point(landmarks, POINTS.LEFT_HIP, POINTS.RIGHT_HIP)
   pose_center = tf.expandDims(pose_center, 1)
   pose_center = tf.broadcastTo(pose_center, [1, 17, 2])
   landmarks = tf.sub(landmarks, pose_center)
```

```
    let pose_size = get_pose_size(landmarks)
    landmarks = tf.div(landmarks, pose_size)
    return landmarks
}

function landmarks_to_embedding(landmarks) {
    // normalize landmarks 2D
    landmarks = normalize_pose_landmarks(tf.expandDims(landmarks, 0))
    let embedding = tf.reshape(landmarks, [1, 34])
    return embedding
}

const runMovenet = async() => {
    const detectorConfig = { modelType:
  poseDetection.movenet.modelType.SINGLEPOSE_THUNDER };
    const detector = await poseDetection.createDetector(poseDetection.SupportedModels.MoveNet,
  detectorConfig);
    const poseClassifier = await tf.loadLayersModel('https://models.s3.jp-tok.cloud-object-
  storage.appdomain.cloud/model.json')
    const countAudio = new Audio(count)
    countAudio.loop = true
    interval = setInterval(() => {
        detectPose(detector, poseClassifier, countAudio)
    }, 100)
}

const detectPose = async(detector, poseClassifier, countAudio) => {
    if (
        typeof webcamRef.current !== "undefined" &&
        webcamRef.current !== null &&
        webcamRef.current.video.readyState === 4
    ) {
        let notDetected = 0
        const video = webcamRef.current.video
        const pose = await detector.estimatePoses(video)
        const ctx = canvasRef.current.getContext('2d')
        ctx.clearRect(0, 0, canvasRef.current.width, canvasRef.current.height);
        try {
            const keypoints = pose[0].keypoints
            let input = keypoints.map((keypoint) => {
                if (keypoint.score > 0.4) {
                    if (!(keypoint.name === 'left_eye' || keypoint.name === 'right_eye')) {
```

```
                drawPoint(ctx, keypoint.x, keypoint.y, 8, 'rgb(255,255,255)')
                let connections = keypointConnections[keypoint.name]
                try {
                    connections.forEach((connection) => {
                        let conName = connection.toUpperCase()
                        drawSegment(ctx, [keypoint.x, keypoint.y], [keypoints[POINTS[conName]].x,
                            keypoints[POINTS[conName]].y
                        ], skeletonColor)
                    })
                } catch (err) {

                }

            }
        } else {
            notDetected += 1
        }
        return [keypoint.x, keypoint.y]
    })
    if (notDetected > 4) {
        skeletonColor = 'rgb(255,255,255)'
        return
    }
    const processedInput = landmarks_to_embedding(input)
    const classification = poseClassifier.predict(processedInput)

    classification.array().then((data) => {
        const classNo = CLASS_NO[currentPose]
        console.log(data[0][classNo])
        if (data[0][classNo] > 0.97) {

            if (!flag) {
                countAudio.play()
                setStartingTime(new Date(Date()).getTime())
                flag = true
            }
            setCurrentTime(new Date(Date()).getTime())
            skeletonColor = 'rgb(0,255,0)'
        } else {
            flag = false
            skeletonColor = 'rgb(255,255,255)'
```

```
                countAudio.pause()
                countAudio.currentTime = 0
              }
            })
        } catch (err) {
          console.log(err)
        }


    }
}

function startYoga() {
    setIsStartPose(true)
    runMovenet()
}

function stopPose() {
    setIsStartPose(false)
    clearInterval(interval)
}



if(isStartPose) {
    return (
      <div className="yoga-container">
        <div className="performance-container">
          <div className="pose-performance">
            <h4>Pose Time: {poseTime} s</h4>
          </div>
          <div className="pose-performance">
            <h4>Best: {bestPerform} s</h4>
          </div>
        </div>
        <div>

          <Webcam
          width='400px'
          height='912px'
          id="webcam"
```

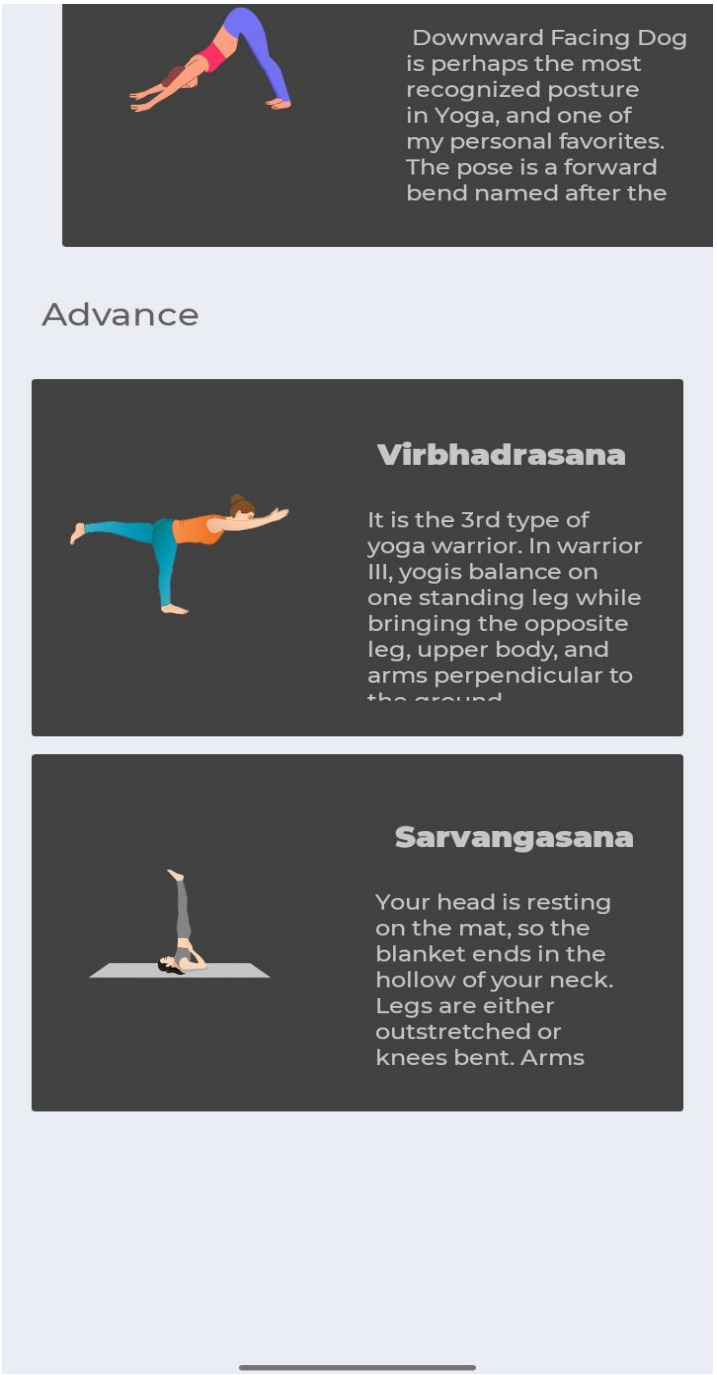
```
            ref={webcamRef}
            style={{
              position: 'absolute',
              top: '-110px',
              padding: '0px',
            }}
          />
            <canvas
              ref={canvasRef}
              id="my-canvas"
              width='502px'
              height='590px'
              style={{
                position: 'absolute',
                zIndex: 1,
                left: '-50px',
                top: 50,
              }}
            >
            </canvas>
          <div>
            <img
              src={poseImages[currentPose]}
              className="pose-img"
              style={{
                width: 120,
                height: 120,
                top: 75,
                left: 1,
              }}
            />
            </div>

          </div>
          <button
            onClick={stopPose}
            className="secondary-btn"
          >Stop Pose</button>
        </div>
      )
    }
```

```
    return (
      <div
        className="yoga-container"
      >
        <button
            onClick={startYoga}
            className="secondary-btn"
          >Start Pose</button>
      </div>
    )
  }
      export default Yoga
```

7.3) Output

-Homepage


Downward Facing Dog is perhaps the most recognized posture in Yoga, and one of my personal favorites. The pose is a forward bend named after the
Advance
Virbhadrasana
It is the 3rd type of yoga warrior. In warrior III, yogis balance on one standing leg while bringing the opposite leg, upper body, and arms perpendicular to
Sarvangasana
Your head is resting on the mat, so the blanket ends in the hollow of your neck. Legs are either outstretched or knees bent. Arms

upward alongside the ears, while the knees

Trikonasana

Trikonasana is a standing posture of the Hatha Yoga type of Yoga. 'Tri' means three and 'Kona' means 'corner'. So, trikonasana literally translates to 'Three-corners'. It is name

Moderate

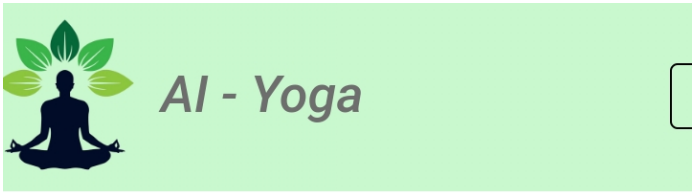
Bhujangasana

Cobra pose is a gentle backbend practiced from a face-down position that warms and strengthens the spine while opening the chest. To enter the pose, lie down on

Svanasana

Downward Facing Dog is perhaps the most recognized posture in Yoga, and one of my personal favorites. The pose is a forward bend named after the

-Selecting asana



Artificial Intiligence Trainer

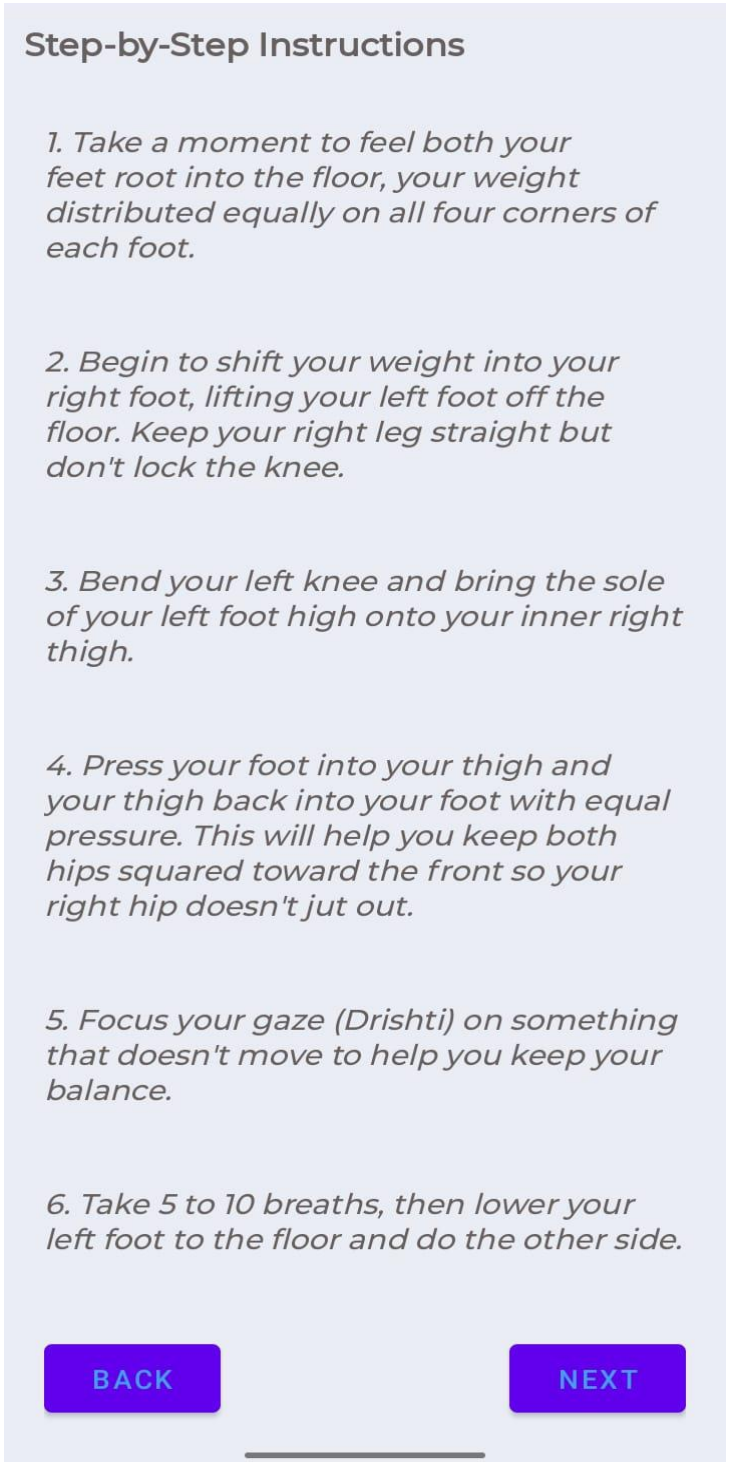
Tree

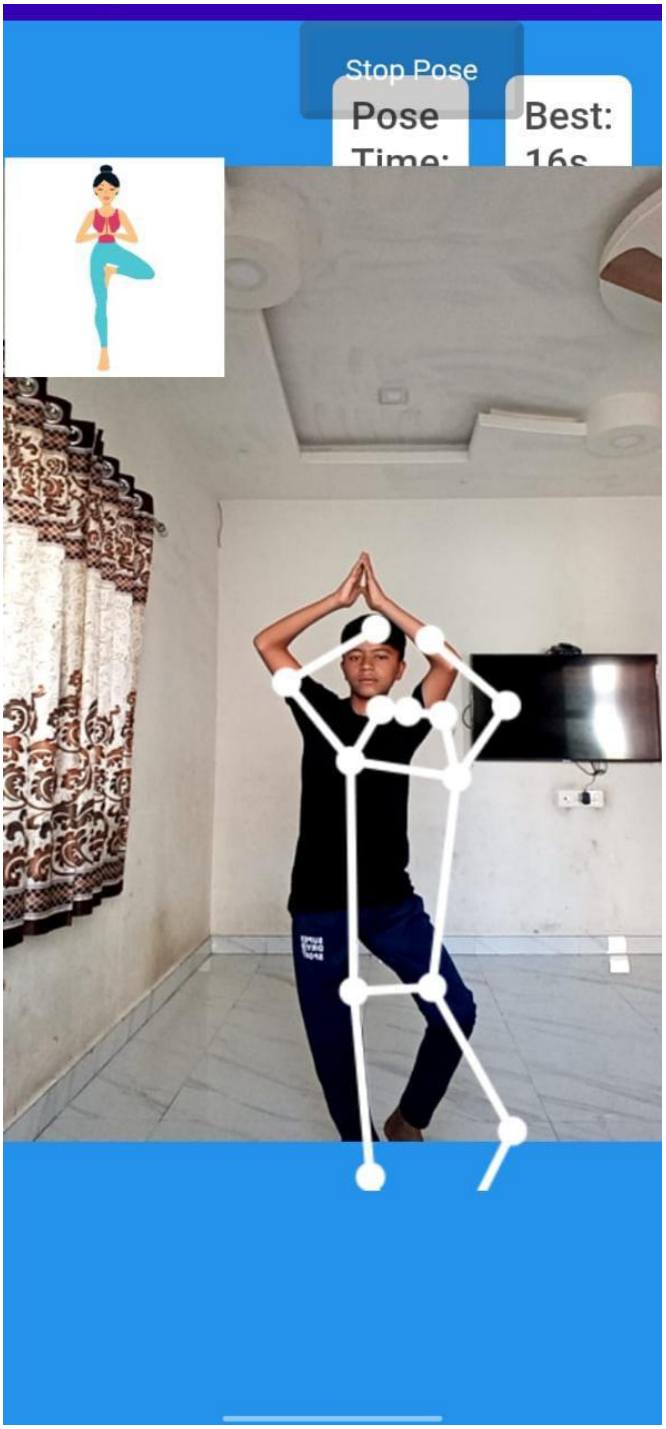
-Selected yoga asana

Tree Yoga

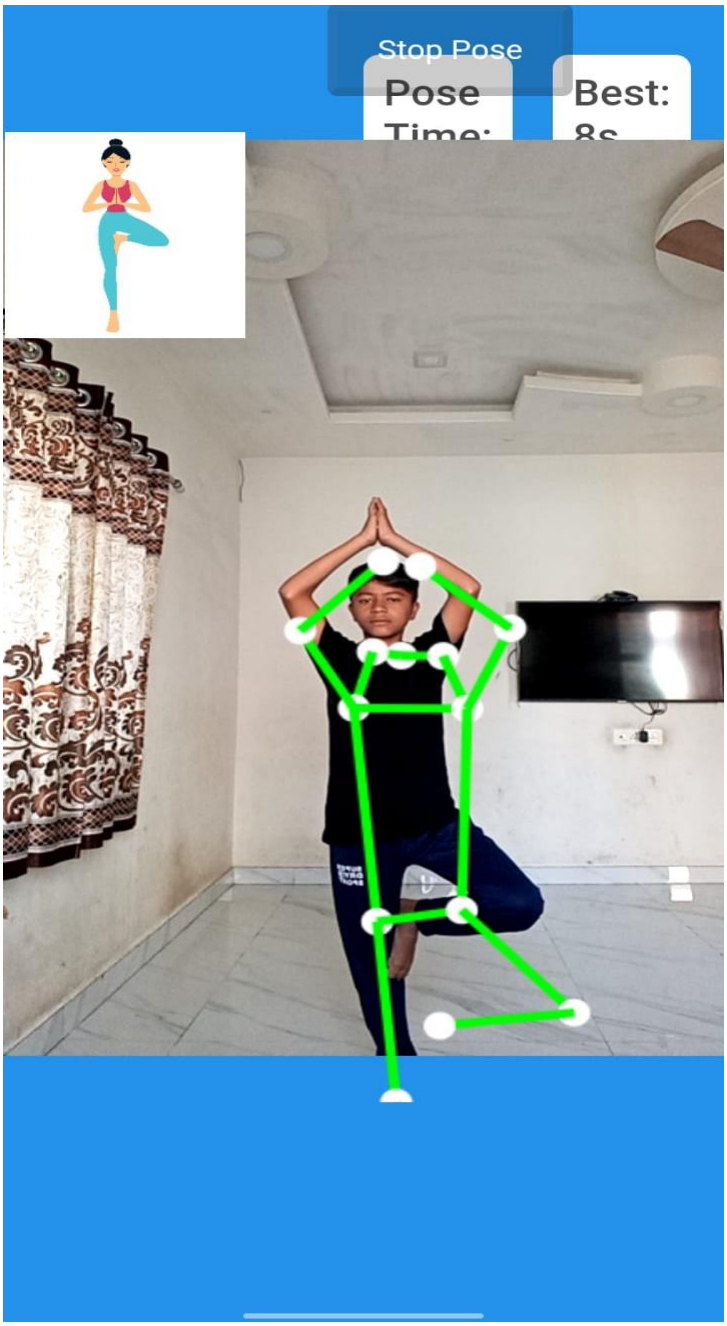
Tree Pose (Vrksansana) is usually the first standing balance pose that is taught to yoga beginners because it's the simplest. Keep your sense of humor about learning to stand on one leg. It's harder than it looks at first and will be different every day. Don't get frustrated if you wobble or even fall over at first. If you are building a sequence around Tree Pose, start with some seated hip openers such as Cobbler's Pose and Eye of the Needle Pose to prepare you.

Step-by-Step Instructions


Step-by-Step Instructions
1. Take a moment to feel both your feet root into the floor, your weight distributed equally on all four corners of each foot.
2. Begin to shift your weight into your right foot, lifting your left foot off the floor. Keep your right leg straight but don't lock the knee.
3. Bend your left knee and bring the sole of your left foot high onto your inner right thigh.
4. Press your foot into your thigh and your thigh back into your foot with equal pressure. This will help you keep both hips squared toward the front so your right hip doesn't jut out.
5. Focus your gaze (Drishti) on something that doesn't move to help you keep your balance.
6. Take 5 to 10 breaths, then lower your left foot to the floor and do the other side.
BACK
NEXT


Stop Pose
Pose
Best:

-performing asana



Chapter -8

Conclusion and Future

8.1) Conclusion: scope

AI-Yoga is a promising application that combines the benefits of yoga practice with the power of artificial intelligence. By utilizing computer vision and machine learning algorithms, AI-Yoga can recognize and analyze the user's yoga poses and provide real-time feedback on their form and technique. This personalized feedback can help users improve their practice, reduce the risk of injury, and achieve their fitness goals.

The future scope for AI-Yoga is vast and exciting. As the technology continues to evolve, we can expect to see more advanced features added to the application, such as customized workout plans, virtual yoga classes, and the ability to track and analyze the user's progress over time. The application could also be integrated with wearable devices and smart home technology to create a seamless and interactive yoga experience.

Furthermore, AI-Yoga has the potential to improve accessibility to yoga practice by allowing users to practice at their own pace and in the comfort of their own home. This could benefit individuals who are unable to attend traditional yoga classes due to location, time constraints, or physical limitations.

Overall, AI-Yoga has the potential to revolutionize the way we approach yoga practice and fitness training. With ongoing research and development, we can expect to see even more innovative and impactful features added to the application in the future.

8.2) Objective Achieved:

Our proposed system is based on movement recognition. The goal of the application is to provide image guides, which helps user to understand the poses and postures.

When user starts using application camera captures their movements during exercise if the posture is wrong the user receives a description of the errors made and recommendations to remove them.

- Provide real-time feedback on user's form and technique during yoga practice.
- Help users improve their practice and reduce the risk of injury.
- Provide personalized yoga workout plans based on the user's goals and fitness level.
- Increase accessibility to yoga practice by allowing users to practice at their own pace and in the comfort of their own home.
- Create a seamless and interactive yoga experience through the integration of advanced technology, such as computer vision and machine learning algorithms.
- Improve overall health and wellness of users by promoting regular and safe yoga practice.

8.3) Future Scope:

AI-yoga has a promising future scope as the demand for online wellness and fitness programs is growing rapidly, especially during the pandemic. As more people turn to digital platforms to maintain their physical and mental health, AI-yoga can provide a unique and personalized experience that adapts to the user's needs. In the future, AI-yoga can incorporate advanced technologies such as machine learning and artificial intelligence to provide a more personalized experience.

The AI algorithm can analyze user data, such as fitness levels, health conditions, and preferences, to create a customized yoga routine that meets the user's specific needs. This will make the program more effective and engaging for users.

Moreover, AI-yoga can incorporate virtual reality and augmented reality technologies to create an immersive experience that simulates a yoga studio or retreat. This can enhance the user's experience and make them feel more connected to the practice. With the increasing penetration of smartphones and affordable internet plans, AI-yoga can leverage these technologies to reach a wider audience. The program can be accessed from anywhere, anytime, making it convenient for users to maintain their yoga practice.

Furthermore, the integration of virtual reality and augmented reality technologies in AI-yoga can create a more immersive experience for users. With VR/AR, users can feel like they are practicing yoga in a real-life setting, such as on a beach or in a forest. This can add a new dimension to the practice and help users to relax and de-stress more effectively.

Additionally, AI-yoga can expand its reach by partnering with other companies and organizations. For example, it can collaborate with fitness wearable’s manufacturers to provide real-time tracking and monitoring of users' yoga sessions. It can also partner with healthcare providers to offer yoga as a complementary therapy for various health conditions such as stress, anxiety, and depression. Finally, AI-yoga can also offer online yoga teacher training programs, enabling people to become certified yoga instructors from the comfort of their homes. This can open up new career opportunities for individuals and help to meet the growing demand for qualified yoga teachers.

In conclusion, the future scope of AI-yoga is promising as it can incorporate advanced technologies to provide a personalized and immersive experience for users. As the demand for online wellness programs continues to grow, AI-yoga can become a significant player in the industry.

8.4) System Requirements:

External Requirements:

- **Front End :**

1) React Js.
2) Android Studio.

- **Back End:**

1) Firebase.
2) Java.
3) Tensor Flow.
4) JavaScript.

➢ **Software and Hardware Requirements at User's side.**

Software Requirements:

- Operating System (Windows 10 and above).
- Internet connection.

Hardware Requirement:

- Ram at least 4gb
- Any processor.

Chapter -9

Reference and Bibliography

[1] Dev-simplified. "React-Router and React-Link." Dev-simplified. Accessed May 5, 2023.
This source provides information about react-router and react-link, which can be useful for building the project.

[2] Banks, Alex. Book on React-JS. O'Reilly Media, 2018.
This book can provide detailed information about React-JS and JSX, which can help in developing the project.

[3] Vaughn, Brian. "Data Rendering." Brian Vaughn. Accessed May 5, 2023.
This source can help in understanding how data rendering works, which is important for displaying information in the project.

[4] Ruan, Luna. "Front End Whiz." Roman-devs. Accessed May 5, 2023.
This article provides information and tips about UI/UX design, which can be useful for creating an appealing and user-friendly interface for the project.

ACTION PLAN for Capstone project Execution and report writing

Planned Duration	Actual Duration	Action Planned	Actual outcome
-23 to -23	23 To 23	Finalizing abstract for report and execution of project	Finalized abstract for report and execution of project.
23 to 23	23 To 23	Review of introduction, literature survey. Execution of project	Reviewed of introduction, literature survey. Execution of project.
23 to 23	23 To 23	Problem statement, scope and objectives of project. Execution of project	Problem statement, scope and objectives of project and Execution of project successfully.
23 to 23	23 To 23	Actual methodology and execution	Actual methodology and execution done successfully.
-23 to -23	23 To 23	Module description, system specifications and project execution	Module description, system specifications and project executed successfully.
-23 to -23	23 To 23	Execution of project	Executed of project done.
-23 to -23	23 To 23	Result analysis	Result analysis done successfully.
-23 to -23	23 To 23	Applications, conclusion and future scope of project	Application , conclusion and future scope of project done successfully
-23 to -23	23 To 23	Finalization of report	Finalized of project is done successfully.

-23 to -23	23 To 23	Plagiarism checking	Plagiarism checking done successfully.
-23 to -23	23 To 23	Presentation	Presentation done successfully.