Robotics Interest Group

Induction Report **– Round 3**

Wireless Robotic Hand Control System

Team Members:	**Rachel Aji Joseph (CS), Sooraj SB Nair (EE), Senthilnathan PL(ME)**
Roll No:	**B240086CS, B241228EE, B241177ME**
Phone No:	**6282901948, 8590159640, 8667767324**
Duration:	**15 days (23rd March - 6th April, 2025)**

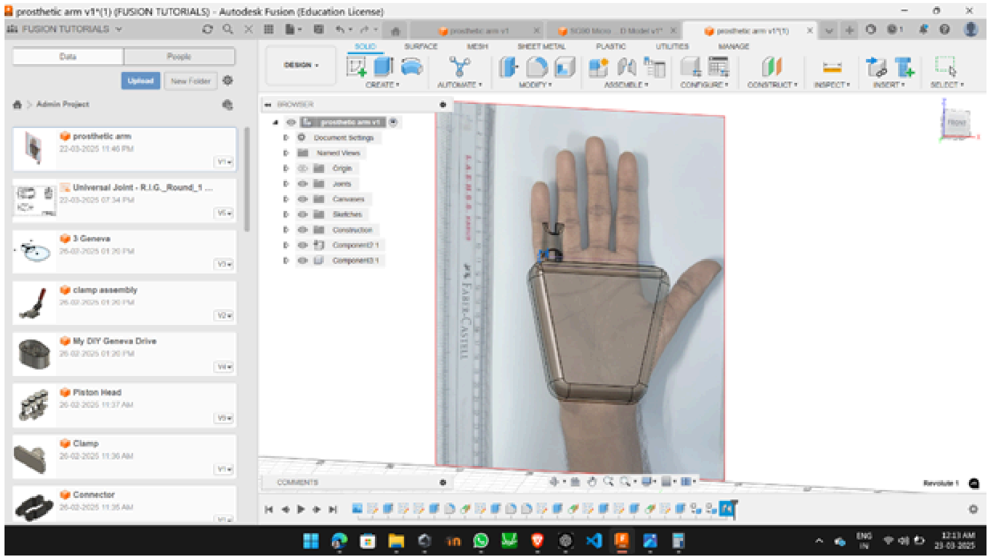
Google Drive Link:	Round 3

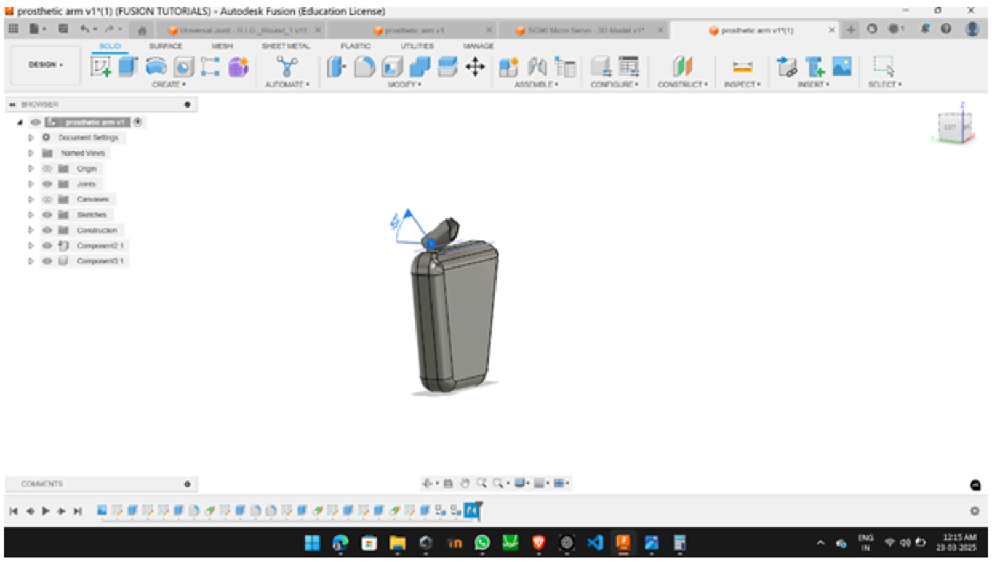
Your Problem Statement:

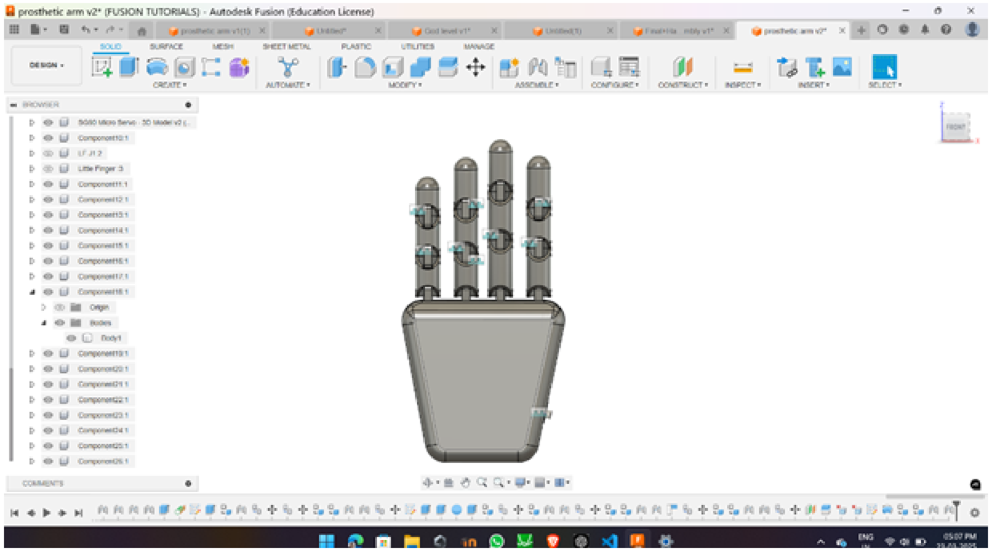
Wireless Robotic Hand Control System

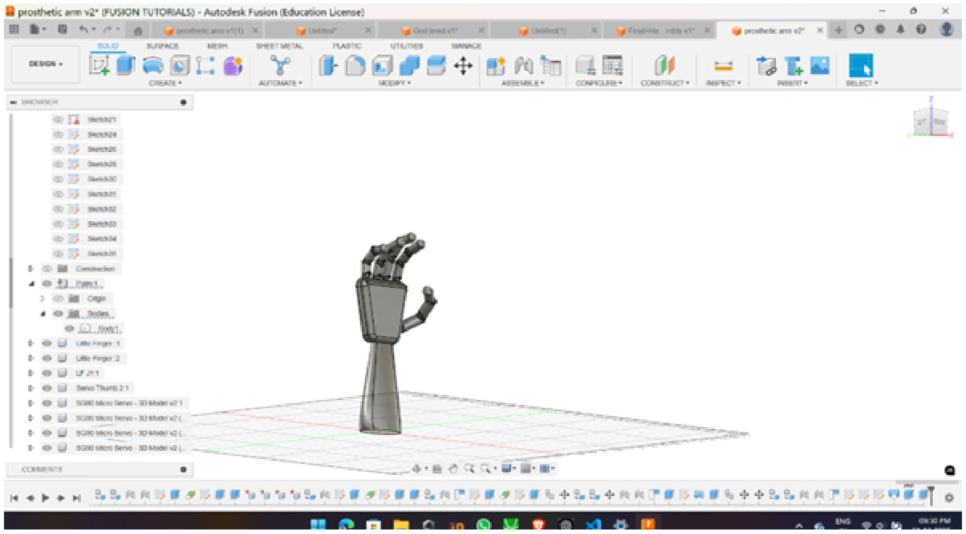
This project aims to develop a wireless control system that enables intuitive operation of a Robotic Hand by mapping the user's hand movements to corresponding Robotic Hand actions through wireless communication.

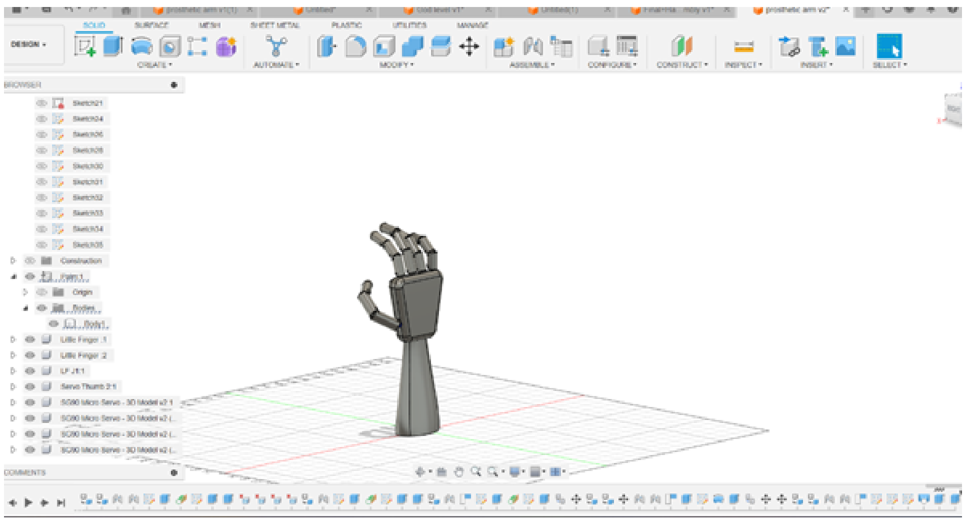
Screenshots of CAD model

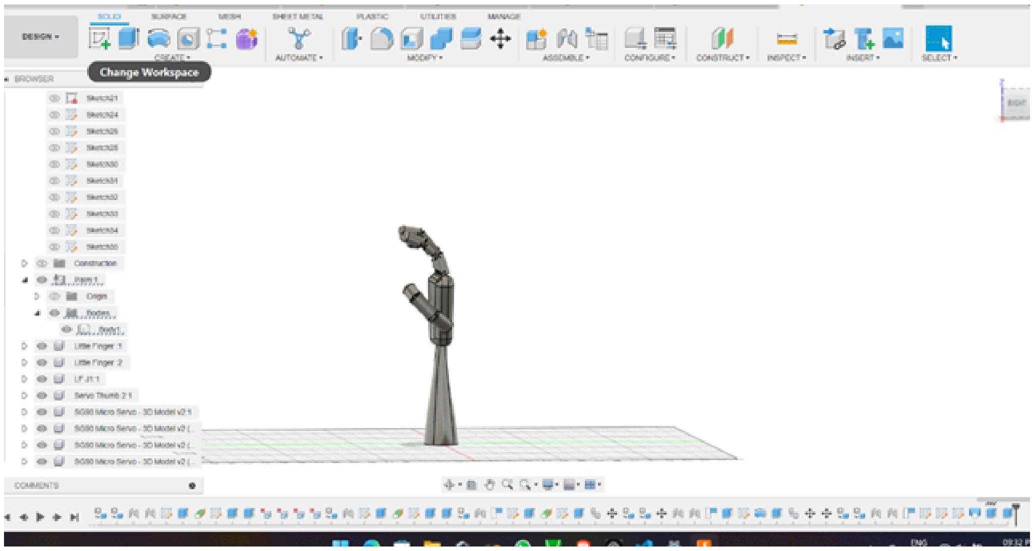

prosthetic arm v1*(1) (FUSION TUTORIALS) - Autodesk Fusion (Education License)
FUSION TUTORIALS
Data
People
Upload
New Folder
Admin Project
prosthetic arm
Universal Joint - R.I.G._Round_1 ...
3 Geneva
clamp assembly
My DIY Geneva Drive
Piston Head
Clamp
Connector
DESIGN
SOLID
SURFACE
MESH
SHEET METAL
PLASTIC
UTILITIES
MANAGE
CREATE
AUTOMATE
MODIFY
ASSEMBLE
CONFIGURE
CONSTRUCT
INSPECT
INSERT
SELECT
BROWSER
prosthetic arm v1
Document Settings
Named Views
Origin
Joints
Canvases
Sketches
Construction
Component2:1
Component3:1
COMMENTS
Revolute 1

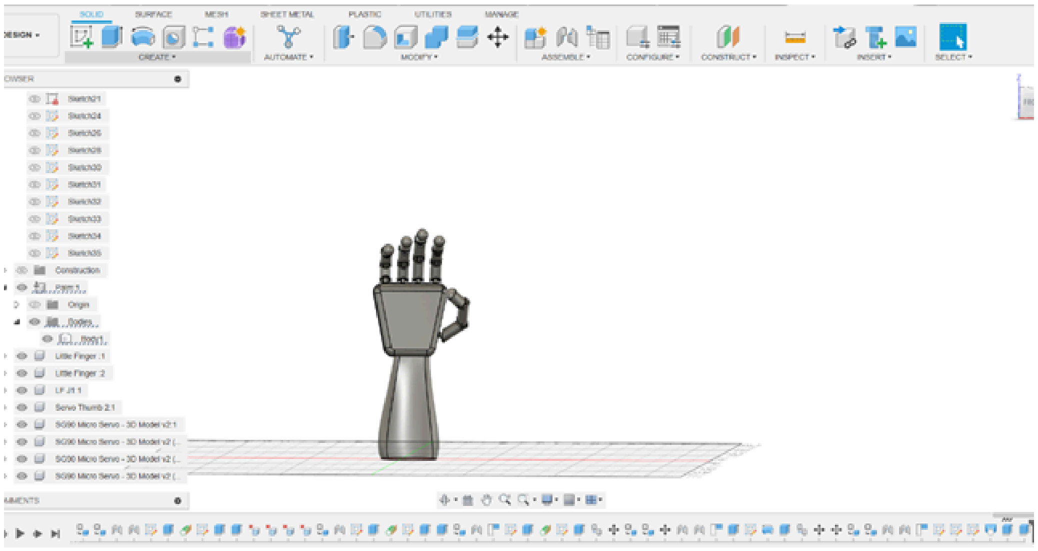

prosthetic arm v1*(1) (FUSION TUTORIALS) - Autodesk Fusion (Education License)
Universal Joint - R.I.G._Round_1 v11
prosthetic arm v1
SG90 Micro Servo - 3D Model v1*
prosthetic arm v1*(1)
DESIGN
BROWSER
prosthetic arm v1
Document Settings
Named Views
Origin
Joints
Canvases
Sketches
Construction
Component2:1
Component3:1
COMMENTS

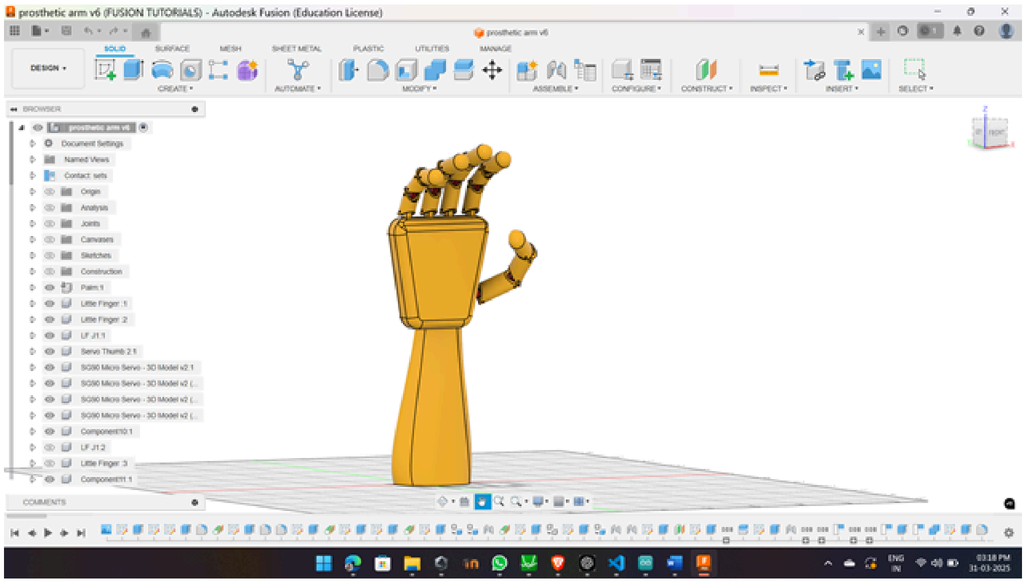

prosthetic arm v2* (FUSION TUTORIALS) - Autodesk Fusion (Education License)
DESIGN
BROWSER
SG90 Micro Servo - 3D Model v2
Component10:1
LF Jt 2
Little Finger 3
Component11:1
Component12:1
Component13:1
Component14:1
Component15:1
Component16:1
Component17:1
Component18:1
Origin
Bodies
Body1
Component19:1
Component20:1
Component21:1
Component22:1
Component23:1
Component24:1
Component25:1
Component26:1
COMMENTS


prosthetic arm v2* (FUSION TUTORIALS) - Autodesk Fusion (Education License)
SOLID
SURFACE
MESH
SHEET METAL
PLASTIC
UTILITIES
MANAGE
DESIGN
CREATE
AUTOMATE
MODIFY
ASSEMBLE
CONFIGURE
CONSTRUCT
INSPECT
INSERT
SELECT
BROWSER
Sketch21
Sketch24
Sketch26
Sketch28
Sketch30
Sketch31
Sketch32
Sketch33
Sketch34
Sketch35
Construction
Origin
Bodies
Body1
Little Finger :1
Little Finger :2
LF J1:1
Servo Thumb 2:1
SG90 Micro Servo - 3D Model v2:1
COMMENTS


SOLID
SURFACE
MESH
SHEET METAL
PLASTIC
UTILITIES
MANAGE
DESIGN
CREATE
AUTOMATE
MODIFY
ASSEMBLE
CONFIGURE
CONSTRUCT
INSPECT
INSERT
SELECT
BROWSER
Sketch21
Sketch24
Sketch26
Sketch28
Sketch30
Sketch31
Sketch32
Sketch33
Sketch34
Sketch35
Construction
Origin
Bodies
Body1
Little Finger :1
Little Finger :2
LF J1:1
Servo Thumb 2:1
SG90 Micro Servo - 3D Model v2:1
COMMENTS


DESIGN
SOLID
SURFACE
MESH
SHEET METAL
PLASTIC
UTILITIES
MANAGE
CREATE
AUTOMATE
MODIFY
ASSEMBLE
CONFIGURE
CONSTRUCT
INSPECT
INSERT
SELECT
Change Workspace
BROWSER
Sketch21
Sketch24
Sketch26
Sketch28
Sketch30
Sketch31
Sketch32
Sketch33
Sketch34
Sketch35
Construction
Palm 1
Origin
Bodies
Body1
Little Finger 1
Little Finger 2
LF J1:1
Servo Thumb 2:1
SG90 Micro Servo - 3D Model v2:1
SG90 Micro Servo - 3D Model v2 (
SG90 Micro Servo - 3D Model v2 (
SG90 Micro Servo - 3D Model v2 (
COMMENTS
RIGHT


DESIGN
SOLID
SURFACE
MESH
SHEET METAL
PLASTIC
UTILITIES
MANAGE
CREATE
AUTOMATE
MODIFY
ASSEMBLE
CONFIGURE
CONSTRUCT
INSPECT
INSERT
SELECT
BROWSER
Sketch21
Sketch24
Sketch26
Sketch28
Sketch30
Sketch31
Sketch32
Sketch33
Sketch34
Sketch35
Construction
Palm 1
Origin
Bodies
Body1
Little Finger 1
Little Finger 2
LF J1:1
Servo Thumb 2:1
SG90 Micro Servo - 3D Model v2:1
SG90 Micro Servo - 3D Model v2 (
SG90 Micro Servo - 3D Model v2 (
SG90 Micro Servo - 3D Model v2 (
COMMENTS



CAD Design:

- I **(Senthilnathan PL)** used my hands with the actual scale as a canvas to trace out the outline of the palm and fingers.

- We used Trapezoidal and Cylindrical Profiles to develop the sketches of the palm and the fingers (except for the thumb where we used the Elliptical Cylinder) into 3D bodies.The tips of the fingers are covered with spheres and ellipsoids respectively.

- Then the bodies were converted to components and assembled with proper joints and motion constraints. The CAD work was completed within the first 2 days.

Electronics and Circuitry:

- The initial plan was to use flex sensors attached to a glove, which shows linear variation in resistance when bent, to control the servo motors remotely using an Esp 32 WROOM DA module for transmission and an Esp 8266 Node MCU as receiver using Esp Now protocol.

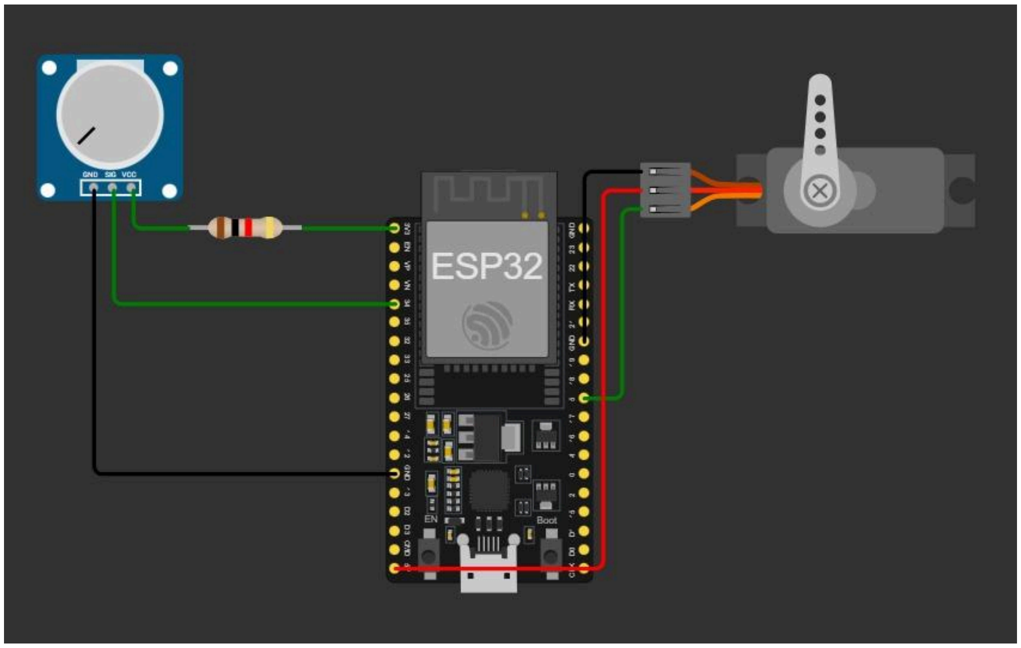
- I (**Sooraj SB Nair)** simulated the circuit in TinkerCAD (using Arduino UNO R3 and flex sensors) and in Wokwi (using ESP32 and potentiometers), as TinkerCAD had flex sensors but no ESP32, and Wokwi had ESP32 but no flex sensors.

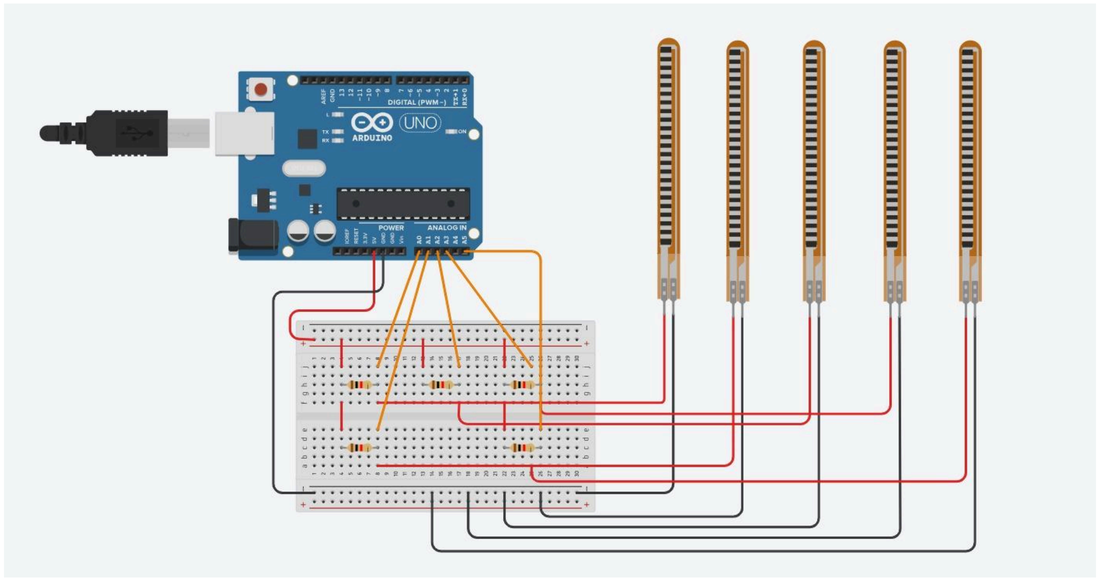
- When we began to work with the actual circuit, the flex sensors provided to us were damaged and hence they didn’t show a linear variation.

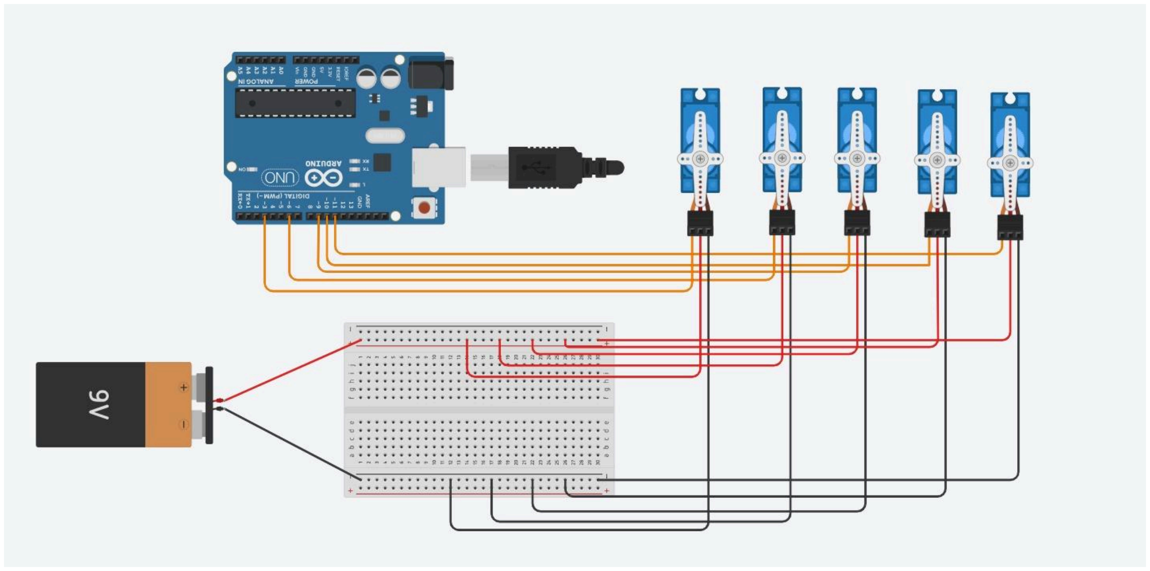
- So, our mentor advised us to use OpenCV instead of Flex sensors. So I modified the circuit accordingly.

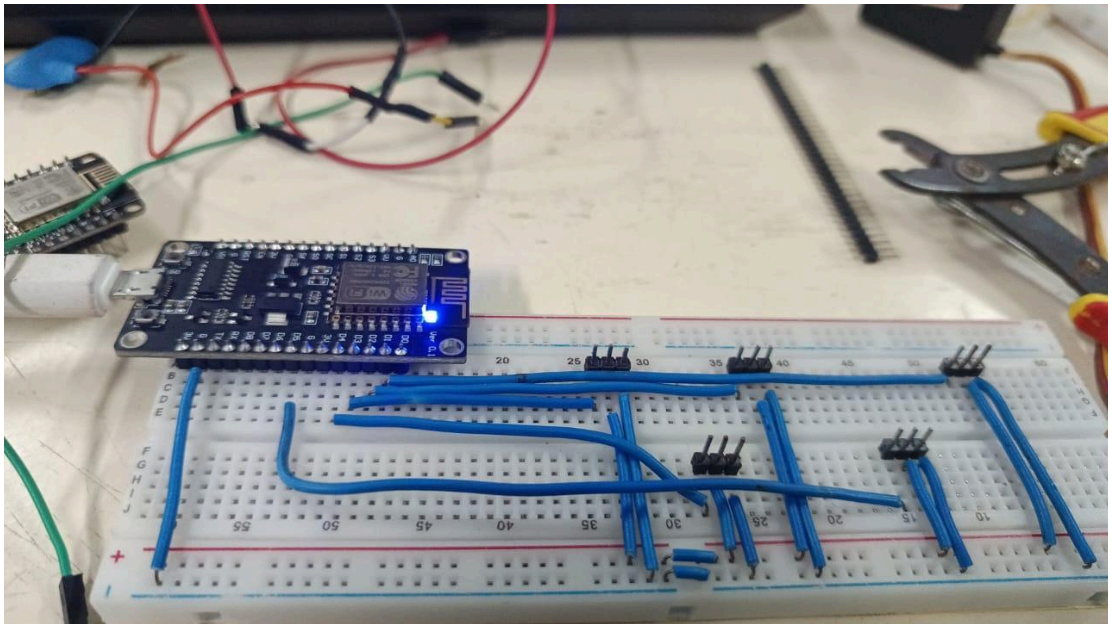
- We used a Python program for transmitting the data from the computer to the NodeMCU via HTTP protocol.The data received was then used to control five servo motors connected to the digital PWM pins (D1–D5) of the NodeMCU.

- For stable operation, an external 5V voltage source was used to efficiently power the servo motors, while the NodeMCU was powered separately using a USB connection from the computer.




DIGITAL (PWM~)
ARDUINO
UNO
POWER
ANALOG IN


DIGITAL (PWM~)
ARDUINO
UNO
POWER
ANALOG IN
9V



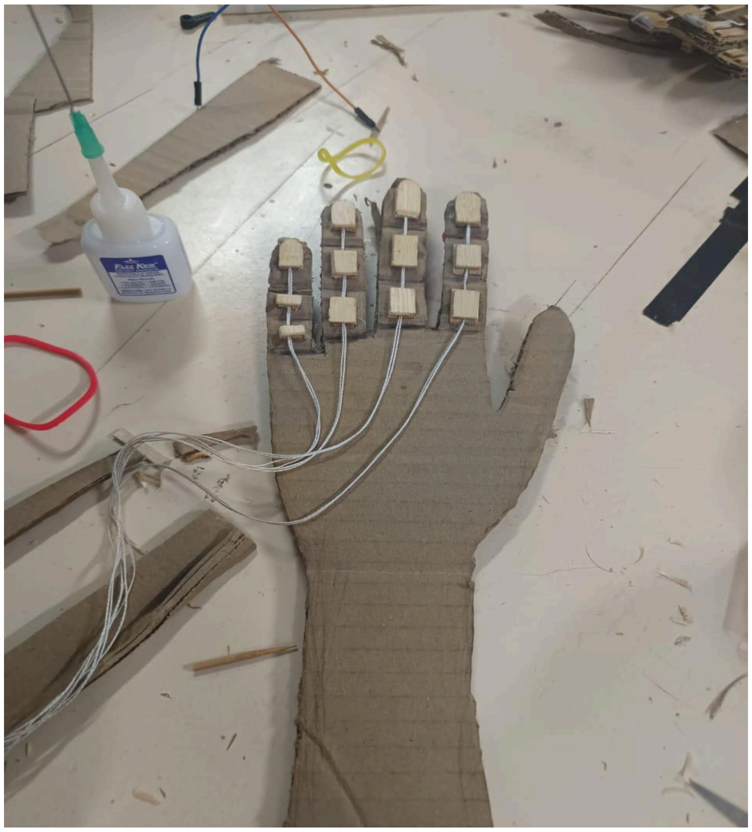
<u>Designing and Fabrication:</u>

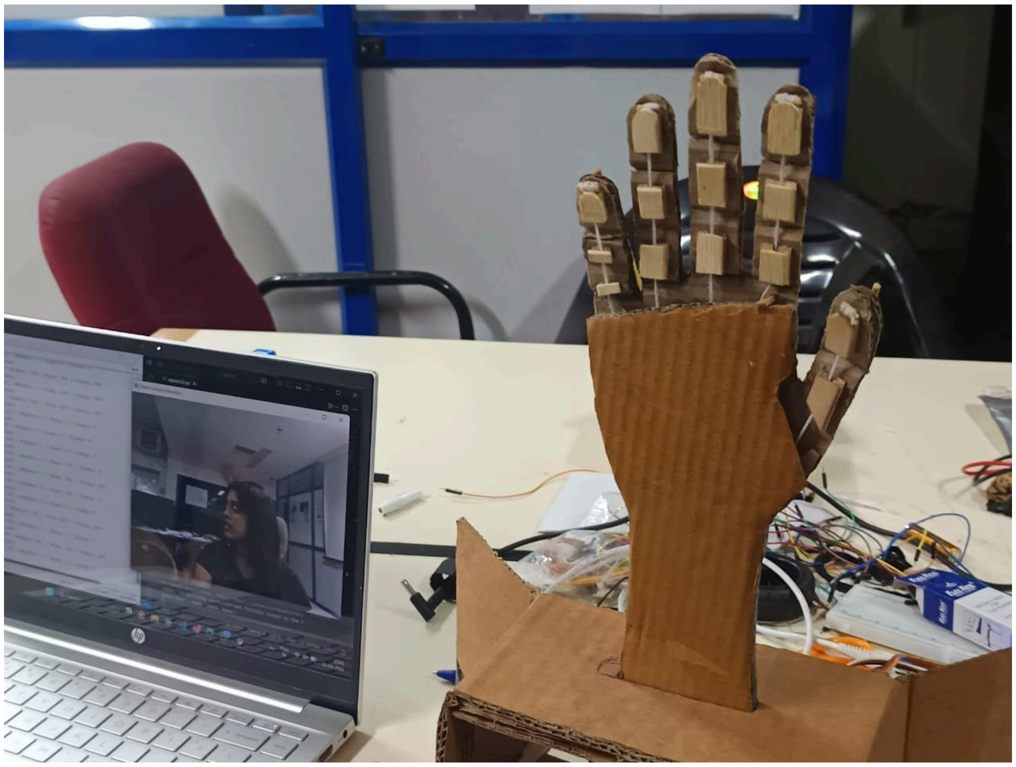
- The very first finger design was designed by **Rachel Aji Joseph**.

- Followed by the 3 model hands were also designed by **her** and **Sooraj SB Nair**, where we experimented with various materials like Foam, Cardboard and with various techniques to achieve the finger motions.

- We used Strings, Rubber bands and Popsicle sticks to achieve the desired motion and scraps of cardboard to give the motion constraints.

- The final model was designed by **Senthilnathan PL**. Later he made the bottom Servo box where servos are fixed at their positions and the strings are attached to them with the help of a single sleeve servo attachment at their calibrated angles.





Software and Programming:

- The initial plan in the programming side was to just use Arduino IDE to take in reading from the five flex sensors, map the values to each of their corresponding motors, so that the motors move according to the flex sensors input.

- As we were told to change the plan to OpenCV, the programming side took much more weight now, incorporating Python too.

- The Python program included the OpenCV and mediapipe modules initially where it used OpenCV to capture camera footage which was sent frame by frame to mediapipe, which used its AI trained model to identify hands and draw a skeleton on them.

- Mediapipe would keep track of our hand movement and do nothing else until we move a certain finger up or down(change the previous position of any finger basically) after which it sends a request to send a signal documenting the difference(in a url format) to the Nodemcu, using HTTP protocol.

- The Nodemcu,connected to the same wifi as the computer, accepts the request and uses the Arduino program it already has uploaded in itself to decide what to do with the url it just received.

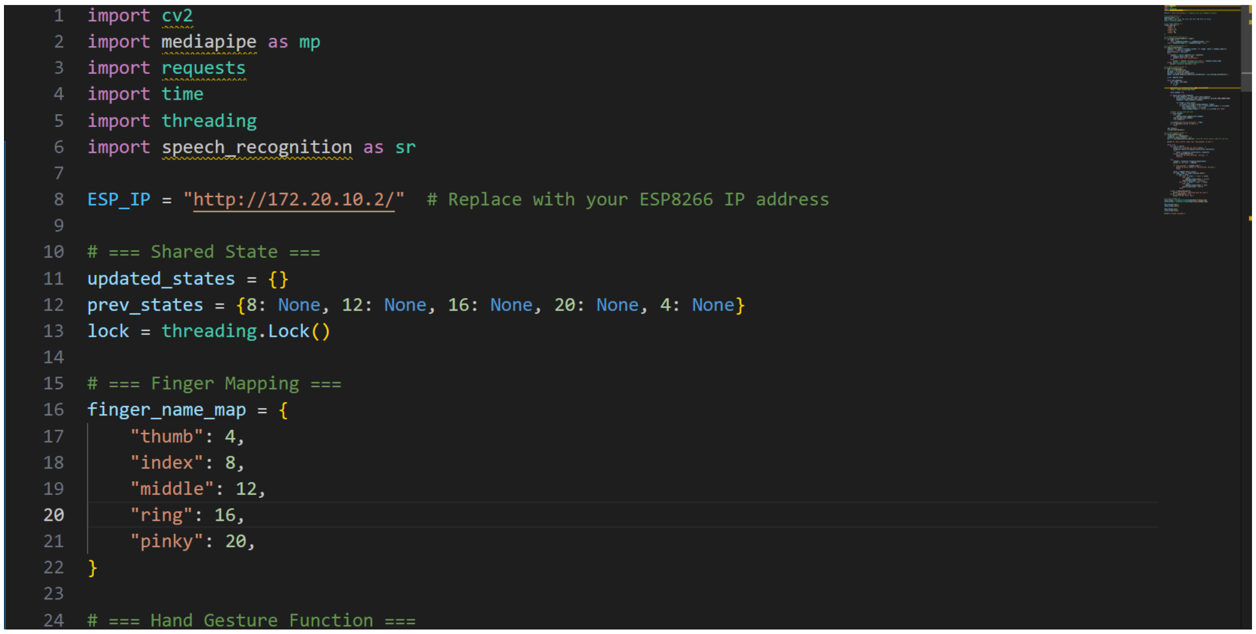
- The Arduino program basically reads the question tag part of the url(the part which tells what change occurred compared to the previous state of our hand)and recreates that same difference by moving the fingers assigned motor, hence resulting in the prototype moving.

- Once this part was done, on week 2, I (**Rachel Aji Joseph**) worked on incorporating the google speech-to-text library to enable voice commands to control the hand as well.

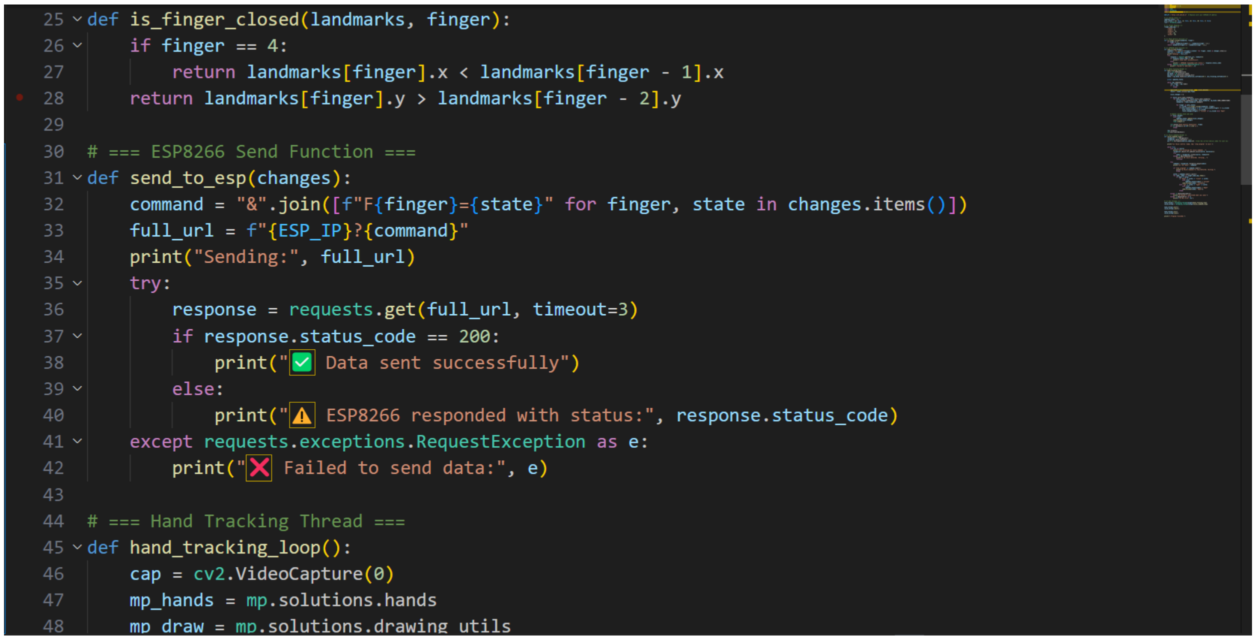
- The voice command part works pretty similar just that instead of a continuously video capturing camera I have a constantly listening microphone as well. Once the mic hears a sentence with “Open” , “Up” , “Close” or “Down” in it, it moves the finger mentioned in the same sentence accordingly by making the exact same url format the OpenCV program created.

- Finally I’ve resulted in a Python program using two separate threads for video and audio controls for the hand meaning I can control my arm with any of the two options anytime while the program is running.

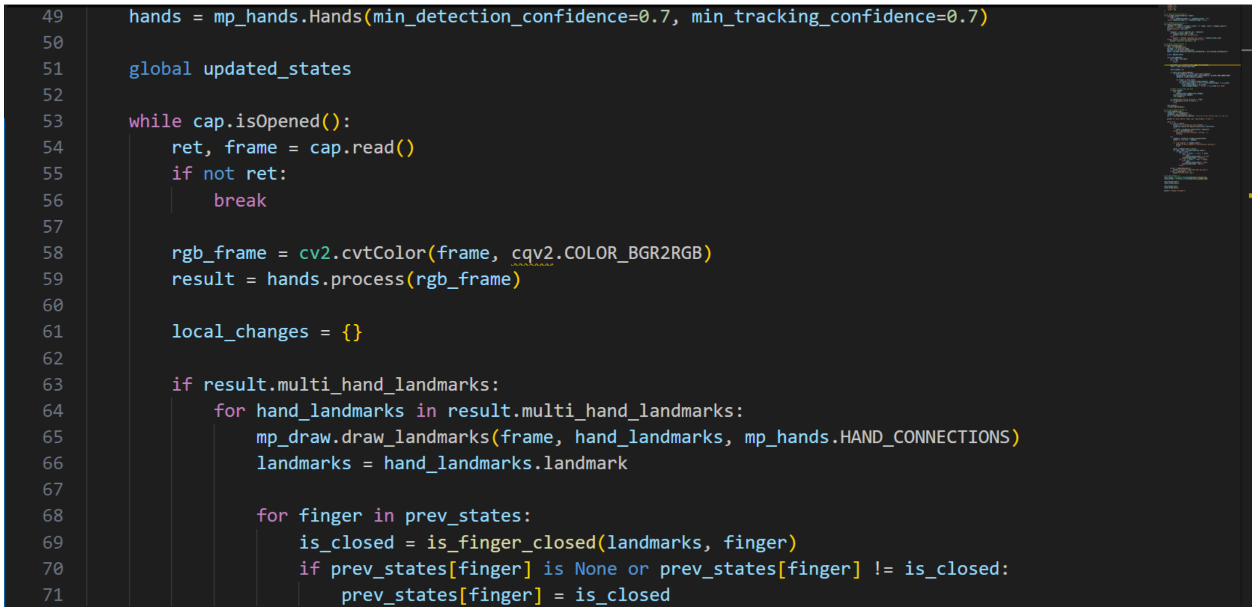
PYTHON CODE



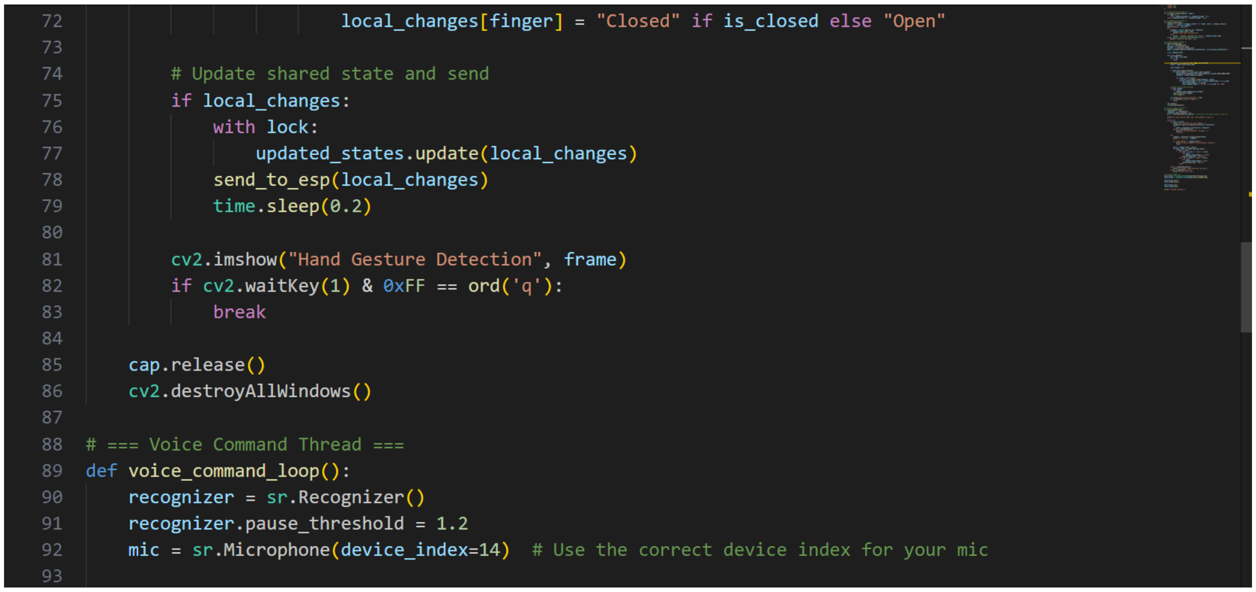
```python
import cv2
import mediapipe as mp
import requests
import time
import threading
import speech_recognition as sr

ESP_IP = "http://172.20.10.2/"  # Replace with your ESP8266 IP address

# === Shared State ===
updated_states = {}
prev_states = {8: None, 12: None, 16: None, 20: None, 4: None}
lock = threading.Lock()

# === Finger Mapping ===
finger_name_map = {
    "thumb": 4,
    "index": 8,
    "middle": 12,
    "ring": 16,
    "pinky": 20,
}

# === Hand Gesture Function ===
```



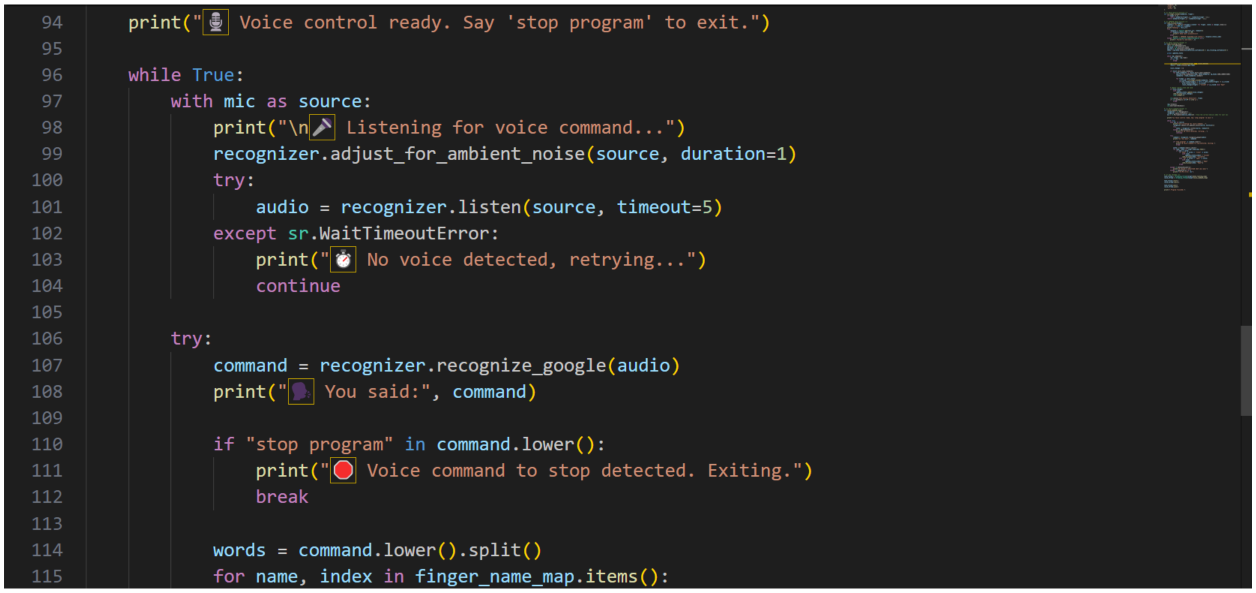
```python
def is_finger_closed(landmarks, finger):
    if finger == 4:
        return landmarks[finger].x < landmarks[finger - 1].x
    return landmarks[finger].y > landmarks[finger - 2].y

# === ESP8266 Send Function ===
def send_to_esp(changes):
    command = "&".join([f"F{finger}={state}" for finger, state in changes.items()])
    full_url = f"{ESP_IP}?{command}"
    print("Sending:", full_url)
    try:
        response = requests.get(full_url, timeout=3)
        if response.status_code == 200:
            print("✅ Data sent successfully")
        else:
            print("⚠️ ESP8266 responded with status:", response.status_code)
    except requests.exceptions.RequestException as e:
        print("❌ Failed to send data:", e)

# === Hand Tracking Thread ===
def hand_tracking_loop():
    cap = cv2.VideoCapture(0)
    mp_hands = mp.solutions.hands
    mp_draw = mp.solutions.drawing_utils
```



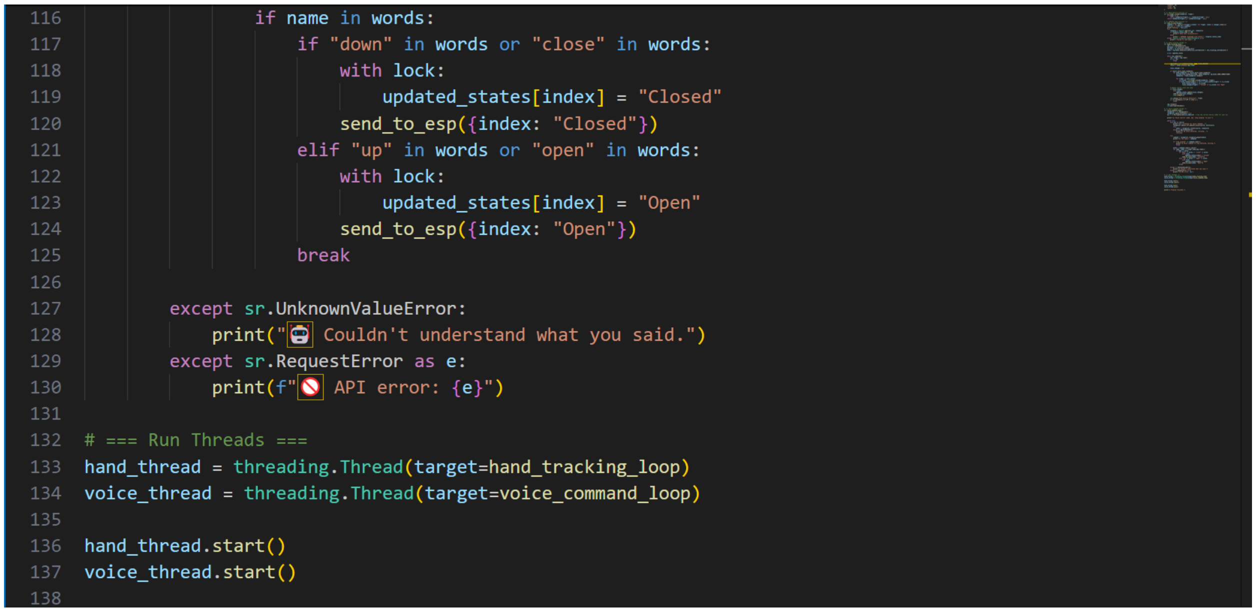
```python
    hands = mp_hands.Hands(min_detection_confidence=0.7, min_tracking_confidence=0.7)

    global updated_states

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        rgb_frame = cv2.cvtColor(frame, cqv2.COLOR_BGR2RGB)
        result = hands.process(rgb_frame)

        local_changes = {}

        if result.multi_hand_landmarks:
            for hand_landmarks in result.multi_hand_landmarks:
                mp_draw.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)
                landmarks = hand_landmarks.landmark

                for finger in prev_states:
                    is_closed = is_finger_closed(landmarks, finger)
                    if prev_states[finger] is None or prev_states[finger] != is_closed:
                        prev_states[finger] = is_closed
```



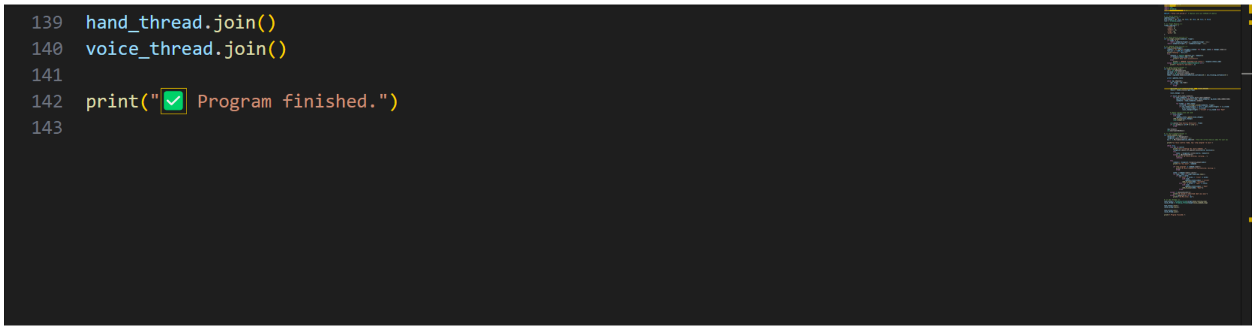
```python
                        local_changes[finger] = "Closed" if is_closed else "Open"

        # Update shared state and send
        if local_changes:
            with lock:
                updated_states.update(local_changes)
            send_to_esp(local_changes)
            time.sleep(0.2)

        cv2.imshow("Hand Gesture Detection", frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

# === Voice Command Thread ===
def voice_command_loop():
    recognizer = sr.Recognizer()
    recognizer.pause_threshold = 1.2
    mic = sr.Microphone(device_index=14)  # Use the correct device index for your mic

```



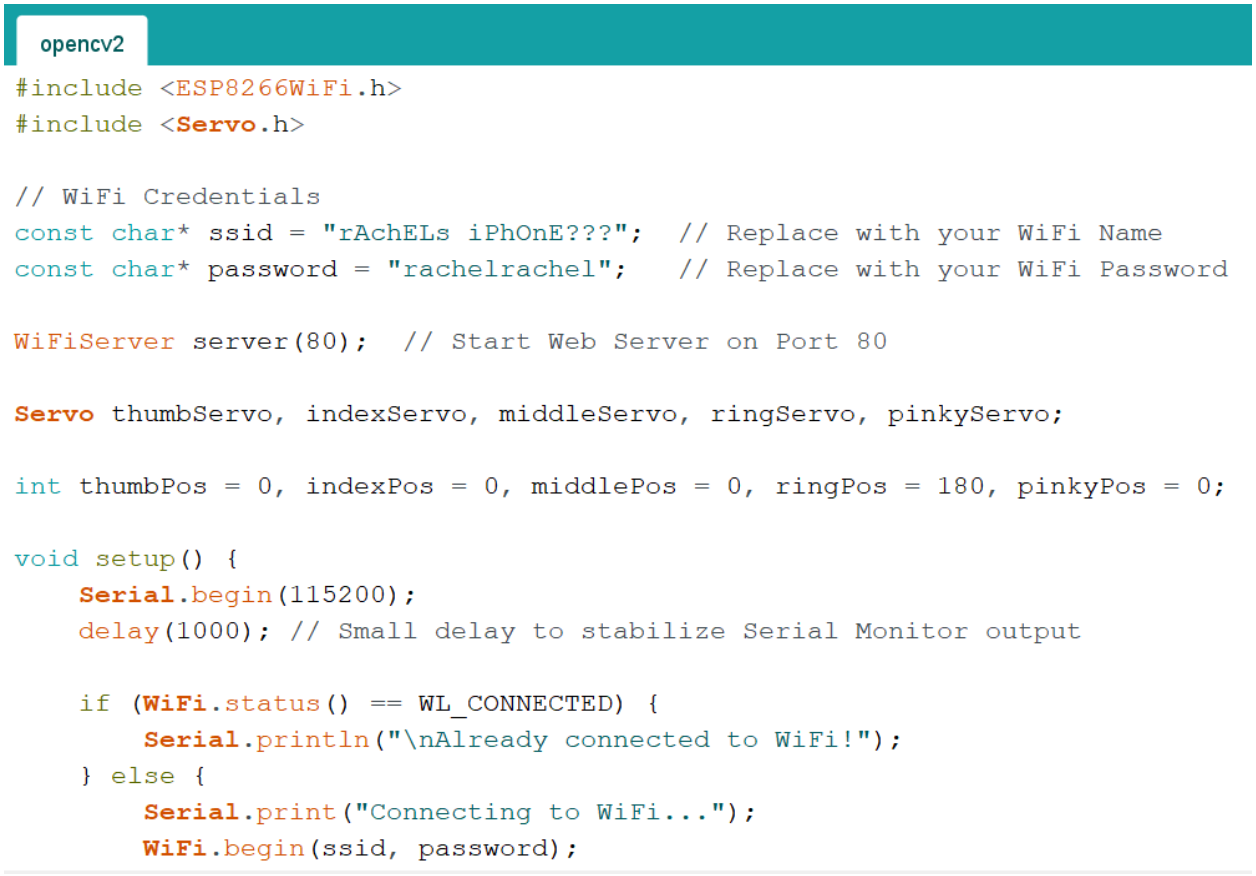
```python
    print("🎙 Voice control ready. Say 'stop program' to exit.")

    while True:
        with mic as source:
            print("\n🎤 Listening for voice command...")
            recognizer.adjust_for_ambient_noise(source, duration=1)
            try:
                audio = recognizer.listen(source, timeout=5)
            except sr.WaitTimeoutError:
                print("⏱ No voice detected, retrying...")
                continue

        try:
            command = recognizer.recognize_google(audio)
            print("🗣 You said:", command)

            if "stop program" in command.lower():
                print("🔴 Voice command to stop detected. Exiting.")
                break

            words = command.lower().split()
            for name, index in finger_name_map.items():
```



```python
                if name in words:
                    if "down" in words or "close" in words:
                        with lock:
                            updated_states[index] = "Closed"
                        send_to_esp({index: "Closed"})
                    elif "up" in words or "open" in words:
                        with lock:
                            updated_states[index] = "Open"
                        send_to_esp({index: "Open"})
                    break

        except sr.UnknownValueError:
            print("🤖 Couldn't understand what you said.")
        except sr.RequestError as e:
            print(f"🚫 API error: {e}")

# === Run Threads ===
hand_thread = threading.Thread(target=hand_tracking_loop)
voice_thread = threading.Thread(target=voice_command_loop)

hand_thread.start()
voice_thread.start()

```



```python
hand_thread.join()
voice_thread.join()

print("✅ Program finished.")

```



```
#include <ESP8266WiFi.h>
#include <Servo.h>

// WiFi Credentials
const char* ssid = "rAchELs iPhOnE???";  // Replace with your WiFi Name
const char* password = "rachelrachel";   // Replace with your WiFi Password

WiFiServer server(80);  // Start Web Server on Port 80

Servo thumbServo, indexServo, middleServo, ringServo, pinkyServo;

int thumbPos = 0, indexPos = 0, middlePos = 0, ringPos = 180, pinkyPos = 0;

void setup() {
    Serial.begin(115200);
    delay(1000); // Small delay to stabilize Serial Monitor output

    if (WiFi.status() == WL_CONNECTED) {
        Serial.println("\nAlready connected to WiFi!");
    } else {
        Serial.print("Connecting to WiFi...");
        WiFi.begin(ssid, password);
```

```
        while (WiFi.status() != WL_CONNECTED) {
            delay(500);
            Serial.print(".");
        }
        Serial.println("\nWiFi connected!");
    }

    // Print IP address regardless of when it connects
    Serial.print("ESP8266 IP: ");
    Serial.println(WiFi.localIP());

    // Start Server
    server.begin();

    // Attach Servos to GPIO Pins
    thumbServo.attach(D1);
    indexServo.attach(D2);
    middleServo.attach(D3);
    ringServo.attach(D4);
    pinkyServo.attach(D5);
    // Move servos to initial positions
    updateAllServos();
}

void loop() {
    WiFiClient client = server.available();
    if (!client) return;

    while (!client.available()) {
        delay(1);
    }

    String request = client.readStringUntil('\r');
    client.flush();

    // Check which fingers are open/closed and update positions
    if (request.indexOf("F8=Open") != -1) indexPos = 180;
    if (request.indexOf("F8=Closed") != -1) indexPos = 0;

    if (request.indexOf("F20=Open") != -1) pinkyPos = 0;
    if (request.indexOf("F20=Closed") != -1) pinkyPos = 180;
    if (request.indexOf("F16=Open") != -1) ringPos = 0;
    if (request.indexOf("F16=Closed") != -1) ringPos = 180;

    if (request.indexOf("F12=Open") != -1) middlePos = 0;
    if (request.indexOf("F12=Closed") != -1) middlePos = 180;

    if (request.indexOf("F4=Open") != -1) thumbPos = 180;
    if (request.indexOf("F4=Closed") != -1) thumbPos = 0;

    // Move all servos simultaneously
    updateAllServos();

    // Send HTTP Response
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println();
    client.println("OK");
    client.stop(); // Close the connection
}


// Function to move all servos at once
void updateAllServos() {
    thumbServo.write(thumbPos);
    indexServo.write(indexPos);
    middleServo.write(middlePos);
    ringServo.write(ringPos);
    pinkyServo.write(pinkyPos);

    Serial.println("Updated Servo Positions:");
    Serial.print("Thumb: "); Serial.print(thumbPos);
    Serial.print(" | Index: "); Serial.print(indexPos);
    Serial.print(" | Middle: "); Serial.print(middlePos);
    Serial.print(" | Ring: "); Serial.print(ringPos);
    Serial.print(" | Pinky: "); Serial.println(pinkyPos);
}
```

Everyday Progress - Documentation:

Robotic Arm Record Book

Senthilnathan’ s Experience:

CAD:

When I started working on the CAD model, I was very confident that I could make this very easily. But the problem was, I learnt CAD by designing 3D bodies and some motion mechanisms where I get the dimensions from their Engineering Drawing Data sheet or from other internet sources. Only while working on this project, I came to know that I lack the skill to design my own ideas with proper dimensions as a CAD designer. I chose geometric profiles for developing the 3D shape of the hands because I had never known about Forms developing in Fusion. I felt it was too late to start learning forms developing and to use it to develop the palm’s profile. This project spotlighted the blindspots in my skills as a CAD designer.

Software:

This project showed me that there are many things I have to learn about programming. Especially

OpenCV and languages like HTML, CSS, XML, JavaScript. Because I thought of creating a web server with a single user interface where a User can switch between Visual and Voice commands and to provide some buttons to do some pre-programmed tasks like waving each finger and to do a countdown upto 5. But this was way beyond my coding knowledge to achieve.

- ➢ The fabrication part made me realize how wonderfully designed the human hands are. They are a truly amazing bio-engineering marvel for their fine motor skills.

<u>Sooraj's Experience</u>

Working on this project was a great learning experience for me. As someone new to electronics, I began by studying the basic working principles of the ESP8266 microcontroller and servo motors.Simulating the circuit in TinkerCAD and Wokwi helped me understand the connections required to build the circuit.Each platform had its own limitations, so switched between them to make the best use of it.

This project helped me understand electronics and a bit of programming that’s required to integrate hardware and software.I'm thankful to my teammates and mentor for guiding me throughout the project. This experience not only helped me learn but also made me more excited and confident to work on similar projects in the future.

Rachel’s Experience

Fabrication

Working on different prototypes and coming up with new ideas for a more stable model was not a very foreign task for me, as I’ve worked a lot with arts and crafts in my past(especially with cardboard). This part may have been a tedious, time consuming process but I do find fun in it.

Electronics

Even though I was assigned the software part of the project, I would say I learnt much more from electronics and different microcontrollers and motors in this project. Being grouped with mechanical and electrical team members who already knew well about circuits and simple electrical components, me not having gone to a single electrical lab in my life, I learnt alot from them. I may have taken a bigger part in fabrication and software in terms of doing actual work and contributing to the project, but the area I improved the most in doing this project was the electrical side.

I would end up watching the other team members construct circuitry one day and once I'm done with my work the next day, I try to recreate and learn to do what they did, by myself, which worked well. I find myself to have an interest in circuits now.

Software

I did have much prior knowledge on OpenCV and mediapipe in my past but google speech-to-text was completely new to me. I very much enjoyed learning about google speech-to-text.

Incorporating two softwares to run a program was also something I've never done before which took me a couple days to fully understand. I also finally have a decent understanding on how protocol works and how important it is, mostly the HTTP protocol as it's what was used in the project. I've been taught a lot about protocol but really didn't understand what it was for and how it works until I worked on this project.

I would say in total I learnt a lot working on this project and I look forward to learn more in the future.