International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Smart Glove for Gesture Recognition using IOT

B. Sarada M.E[1] (Assistant Professor), Ch. Sri Divya [2], Shamshad Begum [3], Potala Kousalya [4], Eluru Simhadri [5]
DEPARTMENT OF ELECTRONICS AND COMMUNATION ENGINEERING
SANKETIKA VIDYA PARISHAD ENGINEERING COLLEGE
VISAKHAPATNAM, INDIA

ABSTRACT:

Communication challenges faced by individuals with speech and hearing impairments create barriers to effective interaction in daily life. This project, "Smart Glove for Gesture Recognition using IOT," introduces an assistive technology solution that translates hand gestures into text and speech, facilitating seamless communication. The proposed system integrates flex sensors, an MPU6050 accelerometer and gyroscope, and an ESP32 microcontroller to capture finger movements and hand orientation. The ESP32 processes sensor data and transmits it to an Android application via Bluetooth or Wi-Fi. The application then converts the recognized gestures into real-time text and speech output, allowing individuals with communication disabilities to convey messages effectively. The smart glove is designed to recognize alphabetic characters, numerical digits, and essential phrases in sign language. The Android application features a user-friendly interface for ease of use, along with text-to-speech functionality for improved accessibility. This project aims to develop a cost-effective, portable, and reliable assistive device that enhances independent communication for individuals with speech and hearing impairments. By leveraging IoT capabilities, the system allows for data storage, remote access, and potential future enhancements, contributing significantly to inclusive communication, education, and healthcare.

1. INTRODUCTION:

In today's rapidly advancing technological landscape, accessibility remains a crucial focus, particularly for individuals with hearing and speech impairments. Communication barriers persist between the deaf/mute community and the general population due to the reliance on sign language, which is not universally understood. Traditional solutions, such as human interpreters or text-based communication, are not always feasible in daily interactions, necessitating the development of assistive technologies. The concept of wearable smart gloves designed to translate sign language into text or speech has gained significant attention in recent years. Research has explored various approaches to this problem, leveraging sensor-based, deep learning, and radiofrequency (RF) technologies. Several studies have demonstrated the effectiveness of gloves equipped with flex sensors and accelerometers to detect hand gestures and translate them into text or speech in real time. Abougarair & Arebi (2022) developed a smart glove capable of translating sign language using flex sensors and accelerometers, facilitating improved communication for the hearing-impaired community [1]. Similarly, Ambar et al. (2023) introduced a wearable sensor glove that integrates with an Android application for real-time sign language translation [2]. Recent advancements in artificial intelligence and deep learning have further enhanced the accuracy of sign language recognition. Chen & Yin (2024) proposed a deep learning-enabled smart glove that utilizes sensor fusion techniques to improve real-time gesture classification, making translation more efficient and reliable [3]. Meanwhile, Bodda et al. (2020) developed a sensor-driven architecture for sign language-to-speech translation, incorporating gyroscopes and accelerometers to track finger orientations and hand movements with high precision [4]. Beyond wearable technology, alternative approaches such as RF sensing have been explored for sign language recognition. Grubuz et al. (2020) demonstrated the use of RF sensors to recognize American Sign Language without requiring physical wearables, offering a privacy-friendly solution for non-invasive gesture recognition [5]. Additionally, Sun et al. (2023) introduced a novel self-supervised learning model, SignBERT+, which enhances sign language understanding through hand pose modelling, further improving recognition accuracy [6]. 3 This project aims to develop a smart glove equipped with flex sensors, an ESP32 microcontroller, and Bluetooth communication to translate sign language gestures into both text and speech. By integrating advanced sensor technologies and real-time processing techniques, this glove seeks to bridge the communication gap between the hearing-impaired and the wider community. Our work builds upon prior research in sign language recognition, with a focus on enhancing usability, portability, and accessibility.

2. LITERATURE REVIEW:

The field of gesture recognition using smart wearable devices has seen significant advancements over the past decade, with numerous studies focusing on improving accuracy, efficiency, and real-time processing. Early research on smart gloves primarily explored the use of flex sensors to capture finger movements, as demonstrated in the work of Kim et al. (2016), who developed a glove-based system for basic sign language recognition. However, these early models lacked real-time connectivity and suffered from limited scalability. With the advent of microcontrollers like ESP32 and Arduino, researchers have shifted towards integrating wireless communication, allowing gesture data to be transmitted instantly to mobile or web applications. Zhou et al. (2019) highlighted the role of low-power microcontrollers in enhancing real-time data processing, enabling seamless interaction between the user and digital interfaces. The integration of Inertial Measurement Units (IMUs), such as the MPU6050, has further improved accuracy in gesture tracking by capturing not only finger bends but also hand orientation and motion. Chen et al. (2020) demonstrated the effectiveness of IMUs in reducing misinterpretation errors in complex sign language gestures, making them an essential component in modern smart gloves. Sensor technology plays a crucial role in the performance of wearable gesture recognition systems. Al-Ghamdi et al. (2018) investigated the use of flex sensors for finger movement detection, emphasizing their reliability in identifying static and dynamic gestures. Complementary studies by Patil et al. (2020) explored how IMU sensors enhance the detection of subtle wrist and hand movements, particularly in cases where multiple gestures appear similar. Additionally, Li et al. (2017) introduced the concept of pressure and capacitive touch sensors in smart gloves, allowing for more intuitive gesture-based interactions and improved detection of hand postures. The role of the Internet of Things (IoT) in smart gloves has been extensively studied in recent years. Singh & Sharma (2022) explored how IoT-enabled wearable devices can transmit real-time gesture data via Wi-Fi and Bluetooth, reducing the need for wired connections and enabling remote accessibility. Kumar et al. (2023) further extended this approach by integrating cloud-based processing, allowing gesture data to be stored and analysed for continuous learning and optimization. Their study demonstrated how cloud platforms could enhance the adaptability of gesture recognition systems by enabling updates and refinements over time. 6 Artificial intelligence (AI) and machine learning (ML) techniques have also contributed significantly to improving recognition accuracy in gesture-based communication systems. Reddy et al. (2021) applied deep learning models, specifically convolutional neural networks (CNNs), to classify sign language gestures, achieving higher precision compared to traditional rule-based approaches. Similarly, Zhang et al. (2023) investigated the use of AI-driven adaptive systems, which allow smart gloves to learn new gestures dynamically, making them more versatile for various applications, including assistive communication and human-computer interaction. Despite these advancements, several challenges remain. Studies have pointed out that accuracy can be affected by sensor drift, environmental interference, and individual variations in gesture execution. Furthermore, ergonomic concerns regarding the comfort and wearability of smart gloves have been raised, emphasizing the need for lightweight and flexible materials. The issue of data processing latency, as noted by multiple researchers, also poses a limitation, particularly in real-time communication applications. Future developments in 5G technology and edge computing could address these concerns by offering ultra-fast data transmission and localized processing, thereby reducing delays in gesture recognition. Overall, the literature underscores the rapid evolution of smart gloves from simple sensor-based systems to highly sophisticated, AI-driven, and IoT-enabled solutions. As research continues to push the boundaries of wearable technology, the integration of advanced sensors, machine learning algorithms, and real-time communication networks will play a crucial role in enhancing the accuracy and usability of sign language recognition systems. These developments hold significant promise for improving accessibility for individuals with hearing and speech impairments, bridging the communication gap through innovative technological solutions.

2.1 Working of the Smart Glove

2.1.1 Overview of Functionality

The smart glove functions by detecting hand gestures using embedded sensors and translating them into meaningful speech or text. It operates in the following sequence: Gesture Detection: The glove is equipped with flex sensors and an MPU6050 accelerometer and gyroscope, which detect finger bending and hand orientation.

Data Processing: The ESP32 microcontroller collects the sensor data and processes it to recognize specific gestures.
Wireless Communication: The processed data is transmitted via Bluetooth to a smartphone application.
Speech/Text Output: The smartphone application receives the data and converts the corresponding gesture into speech or text.
This real-time conversion of hand gestures into speech allows individuals with speech impairments to communicate more effectively with people who do not understand sign language.

2.1.2 System Components

The smart glove consists of several hardware and software components, each playing a crucial role in its operation:

A. Hardware Components

1. ESP32 Microcontroller – The brain of the system, responsible for processing sensor data and transmitting it to the mobile application.

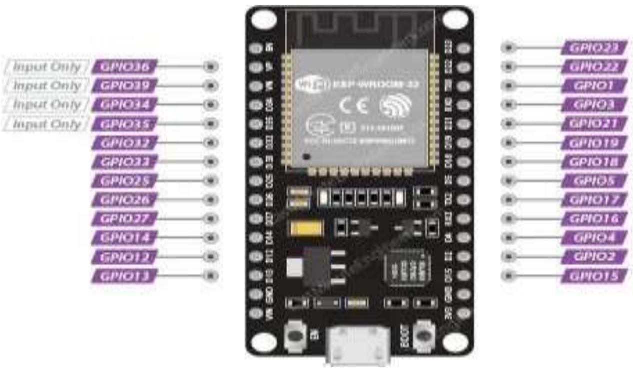

Figure 1. ESP32

2. Flex Sensors – Detect finger bending and hand movement patterns.

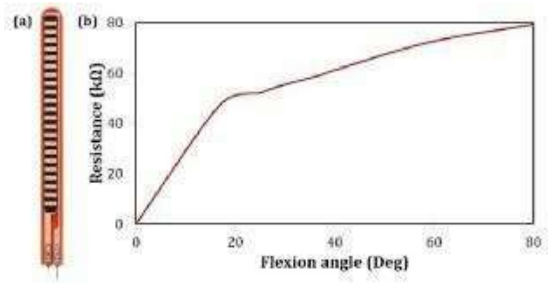

Figure 2. Flex sensor

3. MPU6050 (Accelerometer & Gyroscope) – Captures the orientation and motion of the hand.

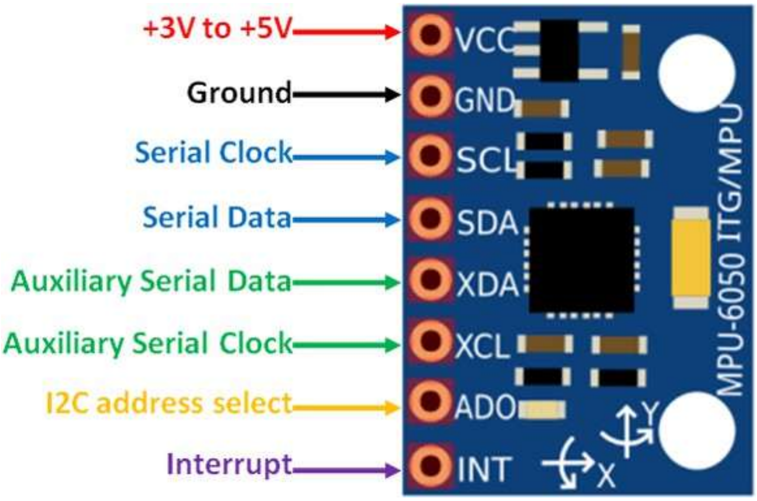

Figure 3. Mpu6050

4. Bluetooth Module (Built-in ESP32) – Enables wireless communication between the glove and the smartphone application.
5. Power Source – A rechargeable battery powers the glove.

B. Software Components

1. Embedded Code (Arduino IDE) – The ESP32 is programmed using the Arduino IDE to read sensor data and send it via Bluetooth.
2. Mobile Application (MIT App Inventor) – The app receives data from the ESP32, processes it, and converts gestures into speech.

2.2 Block Diagram of the Smart Glove System

2.2.1 System Architecture

The block diagram below illustrates the working of the smart glove system:

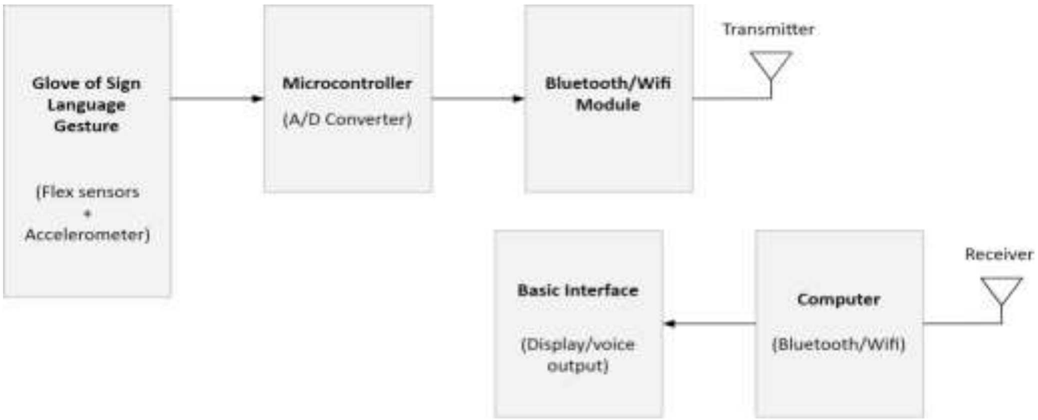

Figure .4 Block Diagram of Smart Glove

2.2.2 Explanation of the Block Diagram

The block diagram consists of the following key components:

1. Input Stage:

○ The flex sensors and MPU6050 detect hand gestures.

○ The collected data is transmitted to the ESP32 microcontroller.

2. Processing Stage:

○ The ESP32 processes the sensor inputs and determines which gesture is being made.
○ The processed data is encoded for transmission.

3. Communication Stage:

○ The processed gesture data is transmitted via Bluetooth to the smartphone application.

4. Output Stage:

○ The mobile application receives the data, converts it into text, and then uses text-to-speech (TTS) conversion to generate spoken output.
○ The user can hear the generated speech or read the translated text.

This system enables real-time gesture recognition and voice output, significantly improving accessibility for individuals with speech impairments.

2.3 Understanding Sign Language

Sign language is a visual form of communication that relies on hand gestures, facial expressions, and body movements to convey meaning. There are over 135 different sign languages worldwide, including American Sign Language (ASL), British Sign Language (BSL), Australian Sign Language (Auslan), and Indian Sign Language (ISL). Additionally, there are adapted forms like Signed Exact English (SEE) and hybrid systems such as Pidgin Signed English (PSE) that blend spoken and signed elements.

While sign language is primarily used by individuals who are deaf or hard of hearing, its impact extends far beyond. It enables seamless communication, fosters inclusivity, and even enhances listening and cognitive skills among hearing individuals. Unlike spoken languages, sign language requires constant visual attention, promoting active engagement in conversations. The practice of maintaining eye contact, essential in sign language, also helps improve focus and interpersonal communication in everyday interactions.

Indian Sign Language (ISL) is the primary sign language used by the deaf and hard-of- hearing community in India. It differs significantly from ASL and BSL and has its own grammar, gestures, and regional variations. Despite its growing recognition, ISL is still in the process of being widely standardized across India. Efforts by the Indian Sign Language Research and Training Centre (ISLRTC) have contributed to its development, promoting inclusivity and accessibility in education and communication.

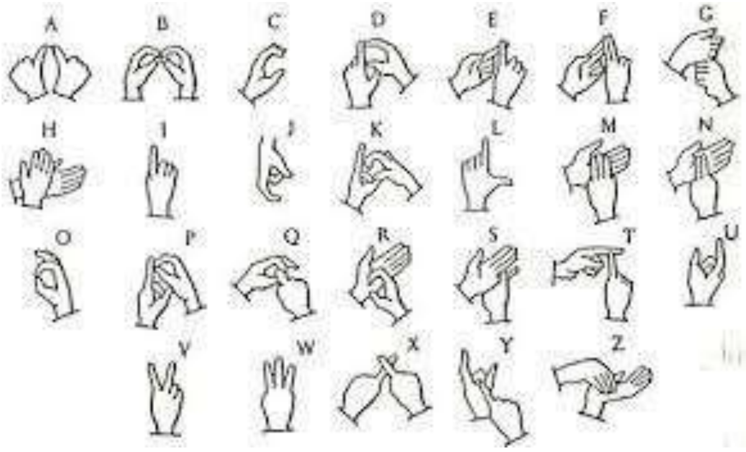

Figure .5 Indian Sign Language

2.4 CIRCUIT SIMULATION

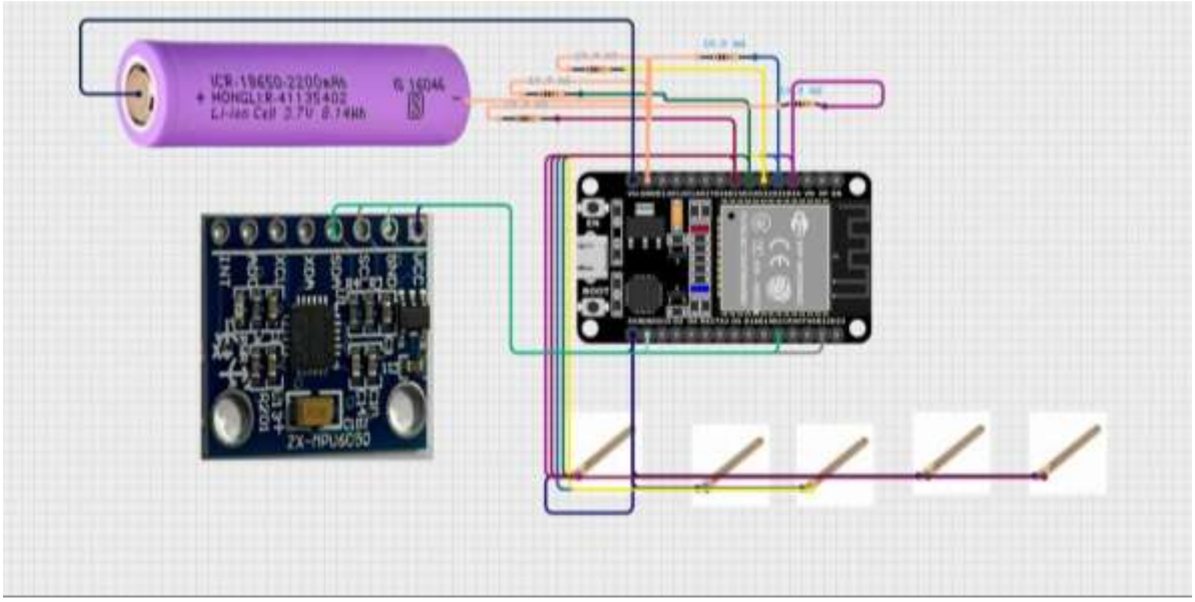

Figure 6 Circuit Diagram for Smart Glove

Connections:

Power Supply:

- The ESP32 is powered by a 3.7V 18650 Li-ion battery.
- Vin Pin → Connected to the positive terminal of the battery.
- GND Pin → Connected to the negative terminal (ground).

Flex Sensors (Finger Bending Detection):

- Five flex sensors are used to detect finger bending.
- One end of each flex sensor is connected to ESP32's analog input pins (A0, A1, A2, etc.).
- The other end is connected to GND through a 10kΩ pull-down resistor to ensure stable voltage readings.

MPU6050 Accelerometer & Gyroscope (Motion Detection):

- This sensor is used for tracking hand orientation and movement.
- VCC Pin → Connected to 3.3V of ESP32.
- GND Pin → Connected to ESP32 GND.
- SDA Pin → Connected to ESP32 GPIO21 (I2C Data Line - SDA).
- SCL Pin → Connected to ESP32 GPIO22 (I2C Clock Line - SCL).

3. Text-to-Speech Performance

The app successfully converted the recognized gestures into text, which was then converted into speech output. The voice synthesis was tested for clarity, pronunciation, and volume levels. The results showed that the speech output was clear and understandable, making communication more effective.

Figure 7 Smart Glove Gesture for Hello

Figure .8 Smart glove gesture for Bye

Figure .9 smart glove gesture for help

Figure .10 Smart Glove gesture for victory

4. Software Code

This code reads flex sensor values and MPU6050 motion data, processes them, and sends recognized characters over Bluetooth (BLE).

```
#include <Wire.h> #include<MPU6050.h>
#include <BluetoothSerial.h>   // Library for Bluetooth communication

// Initialize MPU6050 object MPU6050 mpu;
BluetoothSerial SerialBT; // Create a Bluetooth serial object

// Define flex sensor pins

const int flexSensorPins[] = {32, 33, 34, 35, 36}; // Flex sensor connected to these pins

const int numFlexSensors = sizeof(flexSensorPins) / sizeof(flexSensorPins[0]);


// Variables to store last detected gesture and time unsigned long lastGestureTime = 0;
const int gestureCooldown = 1500; // Time (ms) before detecting a new gesture

String lastGesture = "";

void setup() {
Serial.begin(115200); // Start serial communication SerialBT.begin("SmartGlove"); // Start Bluetooth with the
name "SmartGlove"
Wire.begin(21, 22, 100000); // Initialize I2C communication (SDA=21, SCL=22)


mpu.initialize(); // Initialize MPU6050 mpu.setSleepEnabled(false); // Prevent MPU6050 from
sleeping

mpu.CalibrateAccel(6); // Calibrate accelerometer mpu.CalibrateGyro(6); // Calibrate gyroscope
// Set up flex sensor pins as input with pull-down resistors

for (int i = 0; i < numFlexSensors; i++) { pinMode(flexSensorPins[i], INPUT_PULLDOWN);
}

}

// Function to detect gestures based on MPU6050 sensor data String detectMPU6050Gesture() {
int16_t ax, ay, az, gx, gy, gz; mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
// Define movements for specific gestures based on acceleration data

if (ax > 10000) return "Yes";                          // Forward tilt motion if (ay > 10000) return
"No";                                                  // Right tilt motion if (ay < -10000) return
"Hello";   // Left tilt motion if (az > 10000) return "ThankYou"; // Upward motion
if (az < -10000) return "Please";                          // Downward motion

return ""; // Return empty string if no valid gesture detected

}
```

```
// Function to detect gestures based on flex sensor values String detectFlexGesture() {
int flexValues[numFlexSensors];

// Read values from each flex sensor

for (int i = 0; i < numFlexSensors; i++) { flexValues[i] = analogRead(flexSensorPins[i]);
}

// Conversational gestures

if (flexValues[1] > 1100) return "Help"; if (flexValues[2] > 1100) return "Sorry"; if (flexValues[3] > 1100) return "Food";
if (flexValues[4] > 1100) return "Water"; if (flexValues[0] > 1200) return "Bye"; if (flexValues[1] > 1200) return "I";
if (flexValues[2] > 1200) return "You"; if (flexValues[3] > 1200) return "Love";
if (flexValues[4] > 1200) return "Need";



return "";

}



// Function to detect gestures and send data via Bluetooth void detectGesture() {
// Wait for cooldown before detecting the next gesture if (millis() - lastGestureTime < gestureCooldown) {
return;

}

String mpuGesture = detectMPU6050Gesture(); // Detect gesture using MPU6050

String flexGesture = detectFlexGesture();                                   // Detect  gesture  using  flex
sensors

String currentGesture = mpuGesture + " " + flexGesture; // Combine both gestures

currentGesture.trim(); // Remove extra spaces


// Ensure only one gesture is displayed at a time if (!currentGesture.isEmpty() && currentGesture !=
lastGesture) {

SerialBT.println(currentGesture); // Send detected gesture over Bluetooth

Serial.println(currentGesture); // Print the detected
gesture
lastGesture = currentGesture; // Update last detected
```

```
gesture

lastGestureTime = millis(); // Update the timestamp

}

}



void loop() {

detectGesture(); // Continuously detect gestures delay(50); // Small delay to stabilize readings
}
```

Conclusion

The development of the Smart Glove represents a significant step forward in bridging the communication barrier faced by individuals with hearing or speech impairments. By leveraging embedded systems, sensor integration, and wireless communication, this glove is capable of recognizing both alphabetic gestures (A–Z) and commonly used conversational gestures such as "Help", "Food", "Water", "Yes", "No", "Thank You", and others. These gestures are captured using a combination of flex sensors to monitor finger bends and the MPU6050 accelerometer and gyroscope module to detect wrist and hand motions.

The system successfully interprets sign language and transmits the interpreted data via Bluetooth using the ESP32 module to a custom-designed Android application. The application, built using MIT App Inventor, receives and displays the translated gestures in a user-friendly interface. This ensures that the glove serves as a real-time translator, helping non-sign language users understand sign language speakers with ease.

The glove was tested under multiple gesture scenarios, and results demonstrated satisfactory performance in gesture recognition, minimal delay in transmission, and a smooth user experience through the mobile app. Additionally, the glove’s design is lightweight, cost- effective, and easily wearable, making it a feasible prototype for day-to-day usage and further commercial development. Overall, the project has demonstrated the successful implementation of a practical, assistive wearable that can positively impact the lives of specially abled individuals.