1. Project Title: Smart Blind Cane

2. Purpose

Objective: The primary goal of the Smart Blind Cane is to enhance the mobility, independence, and safety of visually impaired individuals. It will help the user to navigate by providing its real-time location and inform the user with different buzzer sounds by its obstacle detection system. SOS system will be implemented which will send a text message to a dear one in times of urgency or accident. This project has been designed to eradicate the common issues that a blind person may face while moving on their own.

Scope: Intended use and application is to provide -

- *Obstacle Detection :* Use of ultrasonic sensors that will be mounted on a rotating servo motor to continuously scan the surroundings of the user to detect obstacles at close-up distance and alert the user through buzzers to avoid any mobility issues for the user.
- *Location Awareness :* Use of a GPS module that will constantly update the location of the user which can be seen through any output device as it supports google maps interface system. This ensures both the user and their family members can track the user's movement if needed.
- *Emergency Alert (SOS) :* The system will be configured with a contact number of a trusted and capable individual to the user. An emergency button will be installed in the device and upon pressing the emergency button, our GSM module will send an emergency text to that preconfigured contact number that will include any customized message along with the location of the user that will be taken through the GPS module.
- *Efficiency and Usability:* In terms of efficiency, our project will focus mainly on the low power consumption, lightweight design and affordability. Furthermore, the components of our projects are very easily accessible and inexpensive compared to the other working blind canes in the market. This makes our device a suitable option as it ensures low maintenance cost and longer operating life for its daily uses.

Significance: It will significantly reduce the dependency of blind people over other capable persons as it will give them a better sense of mobility and safety. The Smart Blind Cane acts as both a preventive and reactive support system. Accidents will be evaded as its continuously moving sensors will keep the user updated about its vicinity. And even for the worst case scenarios, a capable individual will be informed about the whereabouts of the user in case of emergencies.

3. Components

- Microcontroller : Arduino Uno R3
- Sensors :
  - Ultrasonic SONAR Sensor (HC-SR04) (1x)
  - GPS Module (NEO-7M V2) (1x)
  - GSM Module (SIM800L) (1x)
- Actuators :
  - Servo motor - Micro SG90 (1x)
  - DC Shaftless Vibration motor - D34 (1x)
  - Buzzer Module (Passive) (1x)
- Body/Chassis : Stock Blind Stick
- Additional Components : Jumper wires, Breadboard, Dip Switches, Power Supply (9V and 3.7V Batteries) and Battery Connectors.

4. Cost Breakdown

No	Components	Quantity	Unit Cost (BDT)	Total Cost (BDT)
1	Arduino Uno R3	1	995	995
2	Ultrasonic SONAR Sensor HC-SR04	1	99	99
3	Servo Motor Micro SG90	1	135	135
4	Shaftless Vibration Motor D34	1	150	150
5	Blind Stick (PVC)	1	50	50
6	GSM Module (SIM800L)	1	399	399
7	GPS Module (NEO-7M V2)	1	1050	1050
8	Breadboard and Jumper wires	2	100	200
9	3.7V Lithium Ion Battery	1	690	690
10	Battery (9V)	1	80	80
11	Battery Connector	1	25	25
12	Passive Buzzer 5V	1	15	15
13	Dip Switch (1x)	2	16	32
Total Cost (BDT)				3920

5. System Code Implementation

```
#include <Servo.h>
#include <TinyGPS++.h>
```

```
#include <SoftwareSerial.h>

// -------------------- GPS Module (NEO-7M) -------------------- //
static const int GPS_RX = 7;  // Arduino receives GPS data (from GPS TX)
static const int GPS_TX = 6;  // Arduino sends data to GPS (rarely used)
static const uint32_t GPSBaud = 9600;
SoftwareSerial ss(GPS_RX, GPS_TX); // GPS Serial
TinyGPSPlus gps;

// -------------------- GSM Module (SIM800L) -------------------- //
const int GSM_TX = 2;  // Arduino TX to SIM800L RX
const int GSM_RX = 3;  // Arduino RX from SIM800L TX
SoftwareSerial mySerial( GSM_RX,GSM_TX); // SIM800L Serial

// -------------------- Other Hardware -------------------- //
const int trig_pin = 11;
const int echo_pin = 5;
const int buzzer = 9;
const int buttonPin = 4;
const int ledPin = 13;
Servo myServo;
float timing = 0.0;
float distance = 0.0;
int buttonState = 0;

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  ss.begin(GPSBaud);

  myServo.attach(10);
  pinMode(buzzer, OUTPUT);
  pinMode(echo_pin, INPUT);
  pinMode(trig_pin, OUTPUT);
  digitalWrite(trig_pin, LOW);
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);

  Serial.println("System Ready.");
}
```

```
void updateSerial() {
  while (Serial.available()) {
    mySerial.write(Serial.read());
  }
  while (mySerial.available()) {
    Serial.write(mySerial.read());
  }
}

void sendSMS(String message) {
  mySerial.listen();
  mySerial.println("AT");
  delay(500);
  mySerial.println("AT+CMGF=1");
  delay(500);
  mySerial.println("AT+CMGS=\"+8801730957339\""); // <- Replace with
actual number
  delay(500);
  mySerial.print(message);
  delay(500);
  mySerial.write(26); // CTRL+Z
  delay(5000);
}

void displayInfo() {
  ss.listen();
  Serial.print(F("Location: "));
  if (gps.location.isValid()) {
    double latitude = gps.location.lat();
    double longitude = gps.location.lng();
    Serial.print(latitude, 6);
    Serial.print(F(","));
    Serial.print(longitude, 6);

    String googleMapsURL = "https://www.google.com/maps?q=" +
String(latitude, 6) + "," + String(longitude, 6);
    Serial.print(F("\nOpen the following link to see the location: "));
    Serial.println(googleMapsURL);
  } else {
    Serial.print(F("INVALID"));
    Serial.println(F("\nNo valid location data available."));
  }

  Serial.print(F("  Date/Time: "));
  if (gps.date.isValid()) {
    Serial.print(gps.date.month());
    Serial.print(F("/"));
    Serial.print(gps.date.day());
    Serial.print(F("/"));
    Serial.print(gps.date.year());
  } else {
    Serial.print(F("INVALID"));
  }

  Serial.print(F(" "));
  if (gps.time.isValid()) {
    if (gps.time.hour() < 10) Serial.print(F("0"));
    Serial.print(gps.time.hour());
    Serial.print(F(":"));
    if (gps.time.minute() < 10) Serial.print(F("0"));
    Serial.print(gps.time.minute());
    Serial.print(F(":"));
    if (gps.time.second() < 10) Serial.print(F("0"));
    Serial.print(gps.time.second());
    Serial.print(F("."));
    if (gps.time.centisecond() < 10) Serial.print(F("0"));
    Serial.print(gps.time.centisecond());
  } else {
    Serial.print(F("INVALID"));
  }

  Serial.println();
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
    Serial.println("Button Pressed - Sending SMS and Collecting GPS
Data...");

    sendSMS("I'm in danger. Please help me!");
    updateSerial();// Alert message

    // Get GPS data for 5 seconds
    ss.listen();
    unsigned long start = millis();
    while (millis() - start < 5000) {
      while (ss.available()) {
        gps.encode(ss.read());
      }
    }

    displayInfo();

    if (millis() > 30000 && gps.charsProcessed() < 10) {
      Serial.println(F("No GPS data detected: check wiring or baud
rate."));
      Serial.println(F("Ensure NEO-7M has power and clear sky view."));
      delay(2000);
    }
  } else {
    digitalWrite(ledPin, LOW);

    // Sweep left to right
    for (int angle = 0; angle <= 180; angle += 10) {
      myServo.write(angle);
      delay(15);

      digitalWrite(trig_pin, LOW); delay(2);
      digitalWrite(trig_pin, HIGH); delay(10);
      digitalWrite(trig_pin, LOW);

      timing = pulseIn(echo_pin, HIGH);
      distance = (timing * 0.034) / 2;

      Serial.print("Distance: ");
      Serial.print(distance);
      Serial.print(" cm | ");
      Serial.print(distance / 2.54);
      Serial.println(" in");

      if (distance < 30) {
        for (int i = 0; i < 3; i++) {
          tone(buzzer, 800);
          delay(150);
          noTone(buzzer);
          delay(100);
        }
      }

      delay(100);
    }

    // Sweep right to left
    for (int angle = 180; angle >= 0; angle -= 10) {
      myServo.write(angle);
      delay(15);

      digitalWrite(trig_pin, LOW); delay(2);
      digitalWrite(trig_pin, HIGH); delay(10);
      digitalWrite(trig_pin, LOW);

      timing = pulseIn(echo_pin, HIGH);
      distance = (timing * 0.034) / 2;

      Serial.print("Distance: ");
      Serial.print(distance);
      Serial.print(" cm | ");
      Serial.print(distance / 2.54);
      Serial.println(" in");

      if (distance < 30) {
        for (int i = 0; i < 2; i++) {
          tone(buzzer, 1200);
          delay(400);
          noTone(buzzer);
          delay(200);
```

```
          }
        }

        delay(100);
      }
    }
}
```

6. Functionality Breakdown

- **Functionality 1: Obstacle Detection and Alerting**
  - **Overview :** This function focuses primarily on assisting the visually challenged user to detect and avoid obstacles on their walking path through immediate vibratory and auditory alerts. The system ensures this safety through relentless checking of obstacles through SONAR sensors in a safety range and alerting the user on nearby objects before collisions which helps them to navigate without accidents. HC-SR04 Ultrasonic Sensor and a Micro SG90 Servo Motor is used to scan the environment while the Passive Buzzer (5V) is acting as an auditory actuator that is alerting the user when an obstacle is detected within the safety range, depending on the direction.
  - **Working Procedure :**
    - **Workflow :**
      - Ultrasonic sensor relentlessly sends ultrasonic waves throughout the area by mounting on the rotating servo motor which performs a continuous sweep from **0° to 180°** and then back again from **180° to 0°**.
      - At every angular position the motor rotates, the device takes in the measure for any obstacle that the sensor can find. If the waves hit an obstacle, it bounces back and the sensor then calculates the distance of the obstacle.

    - The safety range we have given to the device is around 30 cm, so if an object is detected within 30 cm, the alerting system will be triggered.
    - As the alerting system is triggered, the buzzer will emit distinct beep patterns based on the servo's sweep direction (left to right and vice versa) to provide directional awareness to the user.

- **Sensor Integration :**
    - Ultrasonic sensor emits ultrasonic pulses from its TRIG pin.
    - The ECHO pin of the ultrasonic sensor receives those reflected pulses that come from nearby objects.
    - Arduino Uno processes the data provided by the sensor, takes the time taken for the pulse that is being received and converts the taken distance in centimeters using the formula -

$$\text{Distance} = (\text{timing} \times 0.034) / 2$$

- **Actuator Logic :**
    - Upon turning on the robot, the Micro SG90 Servo Motor will keep on rotating in horizontal motion to help the sensor to circulate.
    - The Passive Buzzer generates the alerting beeps through tone() and noTone(), which produces and stops its tones respectively.
    - Whenever an object is detected close enough, the beeping is performed based on the direction the sensor has detected. If the obstacle is on the left, a 800 Hz beep is triggered and if it is on the right, 1200 Hz beep is triggered which is differentiable enough for the user to understand where the obstacle is.

- **Control Logic :**

        - The Servo angle of the motor is increased or decreased by 10 degrees per instance which covers the entire forward area of the user.
        - Our safety range is around 30 cm, which makes the threshold for the sensor.
        - Upon measuring the distance, if Distance < threshold, the buzzer generates alert through instructed beeps.
        - If Distance > threshold, no buzzer alerts will be activated.
        - The sweep mechanism produces a wide range of FOV (Field of View) and real-time buzzer response for the user to detect obstacles and avoid them.

    - Advantages :
        - Obstacles are detected in real-time.
        - Servo motor's sweep mechanism brings in Wide Coverage for the sensors.
        - Direction-based beeping patterns help the user to get feedback on the exact position of the obstacle.
        - Sensors only actively emit pulses and only trigger alerts when an object is detected, costing very low energy.
        - It is the most effective feature of our project as its contribution is immense compared to the cost of the components making it highly cost effective.

- **Functionality 2 : GPS Tracking Assistance**
    - **Overview :** This function is based on GPS based service to determine and display the user's real-time geographic location using NEO-7M V2 GPS module. The device takes in the coordinates of the location of the user and outputs a clickable

google maps link on any output device which will potentially help to track the user safely by a verified person like the user themselves or their trusted contacts.

- **Working Procedure :**
  - **Workflow :**
    - The Arduino Uno R3 initializes communication with the GPS module via SoftwareSerial.
    - When the GPS tracking is asked to the device, the module will directly communicate with the satellite and begin collecting location data for a defined time of 5 seconds.
    - Upon completion of the above process, the Arduino will take in the latitude and longitude of the device and then display it in both plain text and google maps interface.
    - As GPS data is not always available due to certain environments and conditions, an error message will be shown.
  - **Sensor Integration :**
    - The NEO-7M GPS Module is connected in the digital pins 6 (TX) and 7 (RX) for transmitting and receiving signals respectively, with a 9600 baud rate.
    - NMEA data that is received from the satellites are decoded using TinyGPSPlus library.
    - gps.location.lat() and gps.location.lng() is used to get the latitude and longitude of the device.
    - It also extracts the date and time that might be helpful for the user.
  - **Control Logic :**
    - The GPS data that it collects for 5 seconds in a non-blocking way, millis() is used to measure the time.

        - The device continuously listens for incoming GPS data upon any changes of position on SoftwareSerial and the data is processed within the microcontroller of the device.
        - After 5 seconds, the system shows the valid data and suggests proper visibility and system integration issues for invalid data.

    - Advantages :
        - Provides real-time geographical information of the device.
        - A clickable google maps IRL is provided by the device for easy and subtle tracking.
        - It can also be used for travel tracking, fall detection and also when the device location is unknown to the user.
        - It does not require any internet connection as it directly connects with the GPS satellites and serial communication.

- **Functionality 3 : SOS Emergency SMS system**
    - **Overview :** This emergency system is built upon the input of the user which will trigger the SIM800L GSM module to send a predefined text message containing the user's real-time GPS coordinates that is extracted from the NEO-7M GPS Module to a trusted individual, which is a pre-saved emergency contact number, in case of emergency.

    - **Working Procedure :**

        - **Workflow :**

            - Upon pressing the emergency button (digital pin 4), the microcontroller detects it through digitalRead().

    - An LED (pin 13) is turned on to indicate the activation of the emergency system of the device.
    - The GPS tracking system is also triggered in that instance, causing it to listen for 5 seconds to gather location data. And then, the GPS coordinates and a Google Maps Link is generated.
    - The SIM800L GSM module will then send the emergency text : "I'm in danger. Please help me!" along with the GPS coordinates of the device that is extracted from the NEO-7M GPS Module to a pre-defined number (in our test case, it is given 01730957339).

- **Sensor Integration :**
    - The push button will be taken as a sensor as it is taking input and data from the user that they are in trouble.
    - The SIM800L GSM module is connected via SoftwareSerial in pins 2 (TX) and 3 (RX) for transmitting and receiving data respectively. It is initialized with AT commands to configure the SMS mode so that the text message can be successfully sent.
    - The GPS module (NEO-7M) activates immediately, gathers and locks in to the coordinates at the time the SOS button is pushed.
    - Emergency contact storage is implemented in the device to give one trusted contact (any) to send the text message to.

- **Control logic :**
    - The main loop of the code continuously checks the activation of the push button as an emergency case can be critical.
    - If the button is pressed, GPS data collection is initiated with a time window where the device fetches the location of the user, turns into google maps URL and sends them through SMS to a trusted contact using sendSMS().

    - Throughout the whole operation, the LED light remains on which gives us feedback on error free execution.

- Advantages :
    - Helps the user by instantly alerting a dear one during emergencies.
    - The location feature enhances the credibility of the SOS function to a greater extent as it guides the rescuers accurately to the user.
    - Faster response time and reliability is achieved using SMS systems as it allows the device to function smoothly even in a 2G network area.

7. Project Challenges

- **Technical Challenges:**
    - GSM Module Instability : The SIM800L GSM module requires a stable 3.7 - 4.2V power supply all along with a high current of 2 Ampere peak. Initial tests proved to be a failure because we could not generate a proper stable voltage that falls in the range or sufficient enough to run the module smoothly. A buck converter could have been helpful in this situation as we found after researching, but we refrained from using it as our Lithium Ion battery was later successful in running the module smoothly. Also our initial tests using Arduino's 5V pin for the module (without knowing much about the consequences) proved to be fatal. Furthermore, GSM modules may not work in weak signal areas.
    - GPS Signal Delay : The NEO-7M GPS module sometimes takes a lot of time to actually acquire the valid location, especially while indoors and in bad weather which causes a delay in tracking. This can potentially cause hindrance in the accuracy of the location.
    - Sensor Noise and Precision : The HC-SR04 ultrasonic sensor may not always give the best results when it is placed in front of reflective surfaces, sharp angles, or

environmental noise. During our test runs, it was even sometimes (although not often) detecting the dust accumulating on the sensors and also the wind from the ceiling fan. For this issue, a cover for the sensor has been used to avoid dust and the sensor is placed at a degrading angle so that no unnecessary wind gets struck against our sensor.

- Servo Motor Jitter : Servo motor (Micro SG90) occasionally showed jittery behavior and unnecessary movement right when the system was turned on.
- Rain or dust may impact the electronics and the fragile components may detach or loosen due to collision.

- **Design Challenges:**
  - Fitting all the modules which are Arduino Uno R3, SIM800L GSM, NEO-7M V2 GPS, Shaftless Vibration Motor, Passive Buzzer and other sensors in our heavy moduled standard stick, in our case a PVC pipe caused us a thoughtful layout and wire management.
  - Not all modules could run on the same power supply as they all contain different operating ranges. Voltage mismatches in power and also in transmission and receiving signals were managed carefully one by one through using separate power supply or through voltage divider using resistors as required.

- **Integration Challenges:**
  - As the GPS and GSM modules both are using SoftwareSerial, switching between them has to be carefully timed so that we do not lose any necessary data in between.
  - The frequent consumption of power from the servo motor affected the other modules, especially the GSM module which had to be later fixed by using a separate voltage input.

- **Budget Constraints**: We initiated our project with a budget not exceeding over 4000 Tk and we successfully succeeded in that term. However, it caused us to choose the most inexpensive components in the market in order to build our project which limited us to use higher-end components which could substantially reduce the troubleshooting and creative workarounds for our project.

- **Risk Mitigation:** Power issues could occur among the components thus separate power supplies have been provided to the components as needed rather than one power source running all the components. Its modular design really helps the device function even if one of the components fails to perform. For instance, even if the GPS module fails to extract the data of the user, the GSM module still sends the emergency text by taking the last found GPS data or no data at all. Components chosen for the device are the most available components in the market which gives the device a thorough standard, replaceable and even upgradable if needed.