CODING AND SOLUTION

PROJECT TITLE :
IoT Based Safety Gadget for Child Safety Monitoring and Notification

TEAM ID : PNT2022TMID42176
TEAM LEADER : SAMEERA N
TEAM MEMBER 1 : MYTHILI T
TEAM MEMBER 2 : YOGESHWARI V
TEAM MEMBER 3 : SOWMIYA S

Functional features:

- The safety device protects individuals from potential harms and dangers.
- Then, send collected data to parents' smartphones by SMS using GSM shield.
- Sensor inside the smartwatch senses the heartbeat of a child or woman who wears it [11]. When he/she is exposed to attacks, heartbeat rate will be high
- When this is detected, alarm sound will be triggered
- it will then automatically make calls to registered contact and to the nearest police station [11]. Based on the location provided by GPS, police will arrive soon at the correct destination.

Code layout :

```
 #include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include<TinyGPS++.h>
#define RXD2 16
#define TXD2 17
HardwareSerial neogps(1);
TinyGPSPlus gps;  char arr[100];
const char* ssid = "Redmi"; const
char* password = "krish@08";
#define ID "17cmwk"
#define DEVICE_TYPE "Tracker"
#define DEVICE_ID "gps1" #define
TOKEN "childtracker1"
char server[] = ID ".messaging.internetofthings.ibmcloud.com";
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json"; char
publish_Topic2[] = "iot-2/evt/Data2/fmt/json"; char
authMethod[] = "use-token-auth";  char token[] = TOKEN;
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;
```

```
WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);
```

```
void setup() {
Serial.begin(115200);
Serial.println();
wifi_init(); } long
previous_message = 0; void
loop() {  client.loop();
String payload = getLocationPayload();
if(payload=="{}"){  return;
}
Serial.print("Sending payload: ");
```

```
Serial.println(payload);
if (client.publish(publish_Topic1, arr)) {
Serial.println("Published successfully");
} else {
Serial.println("Failed");
}  delay(2000);
}
void wifi_init(){ WiFi.begin(ssid,
password);
neogps.begin(9600,SERIAL_8N1,RXD2,TXD2);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println(WiFi.localIP()); if
(!client.connected()) {
Serial.print("Reconnecting client to "); Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");  delay(500);
}
Serial.println("Connected TO IBM IoT cloud!");
}
}
String getLocationPayload(){ boolean
newData = false;
for(unsigned long start = millis();millis()-start<1000;){
while(neogps.available()){ if(gps.encode(neogps.read())){
newData = true;
}
}
}
String payload;  if(newData
== true){  newData = false;
payload = locationPayloadGenerator();
}  else{
Serial.println("No data");  payload
="{}";
}
return payload;
}
String locationPayloadGenerator(){ String payload = "{}";
if(gps.location.isValid()){  float lat =
gps.location.lat();  float lon = gps.location.lng();
payload = "{\"latitude\" : "+String(lat)+",\"longitude\" :
"+String(lon)+"}";  create_json(lat,lon);
} return
payload;  }
void create_json(float lat,float lon){ StaticJsonDocument<100>
doc;
JsonObject root = doc.to<JsonObject>();
root["name"]="Child"; root["latitude"] =
lat;
root["longitude"] = lon;
serializeJsonPretty(doc,arr);
```

}

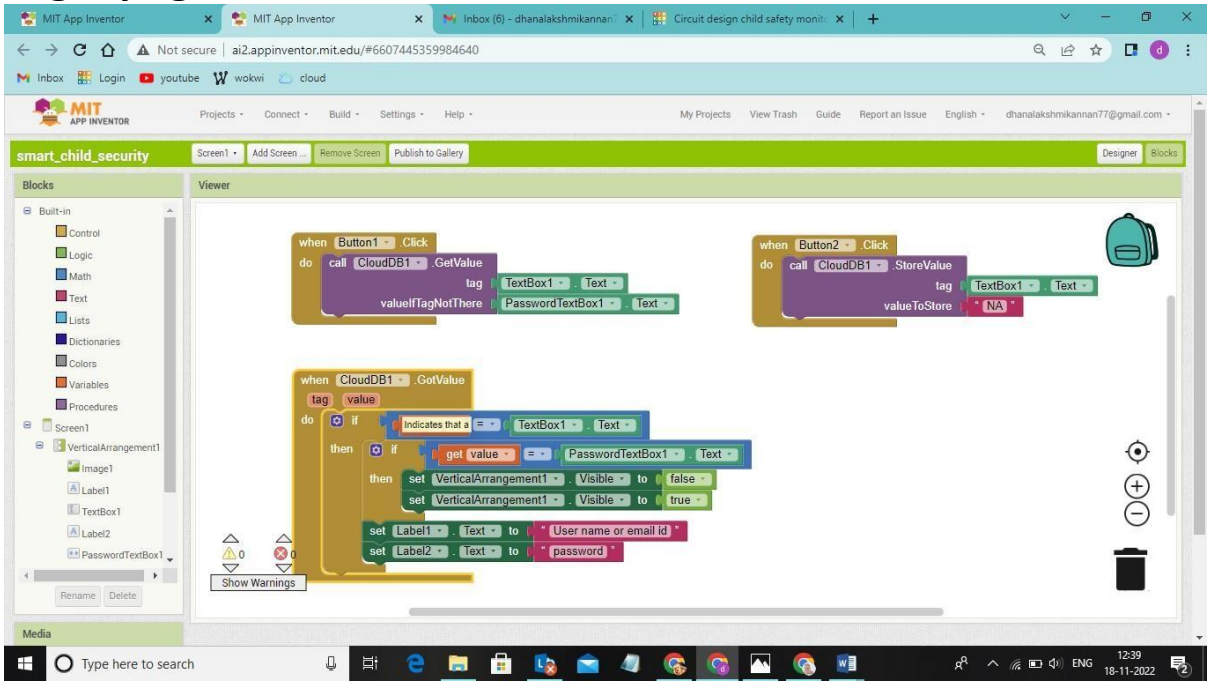
Utilization Algorithm:

1. Start the process
2. Initialize power is supplied microcontroller
3. Check the speed and location of the children
4. Send all data gathered by text message
5. Display real time that is sent to mobile phone
6. Check if the location will be different from the location
7. Send the message
8. After the process completed, it moves to original state
9. Stop the process

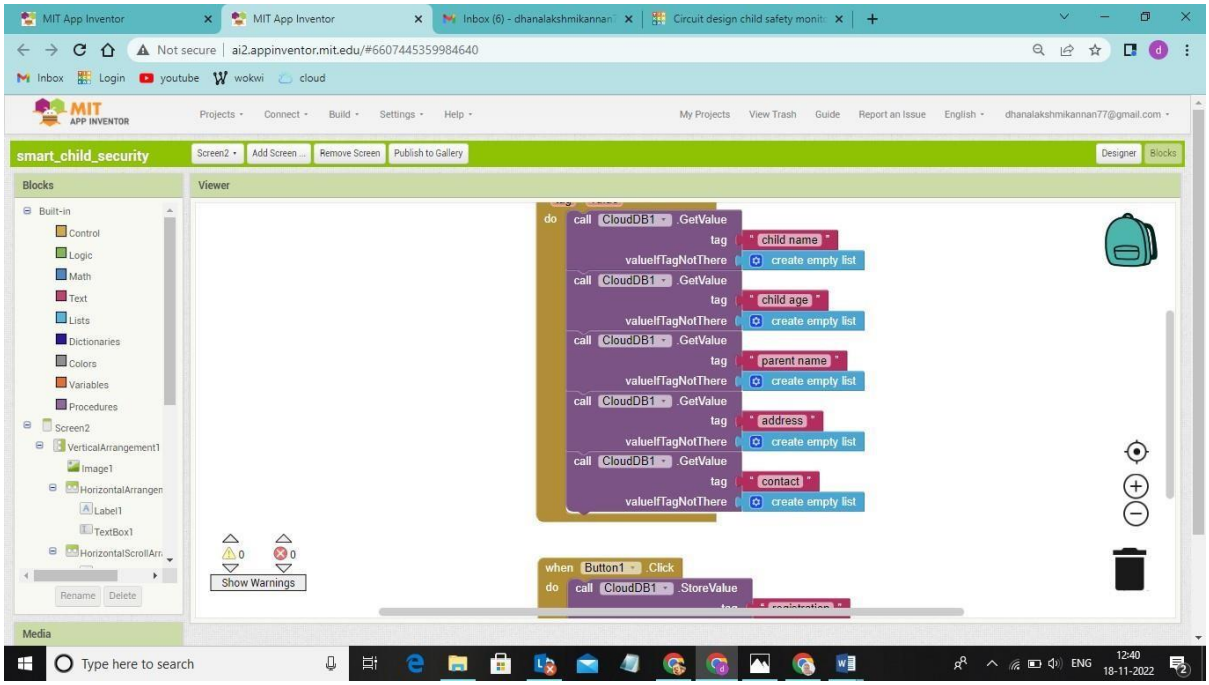
Dynamic Programming:

IoT devices may have the capability to dynamically adapt and change based on their conditions.

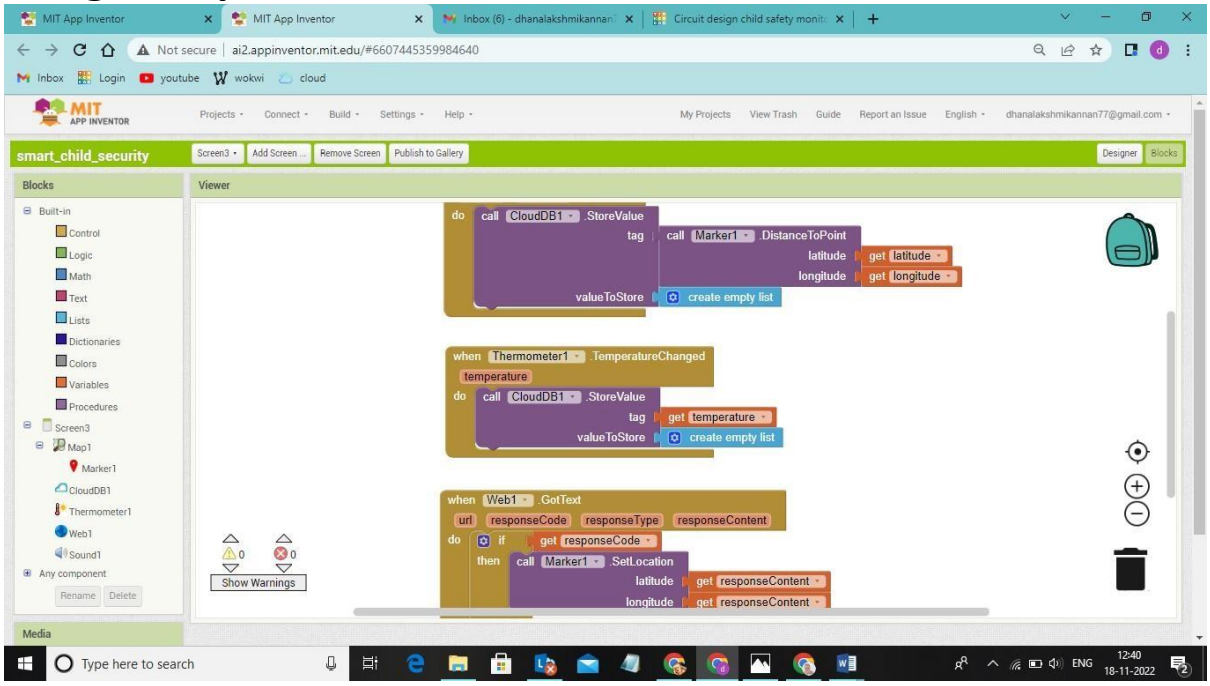
Login page



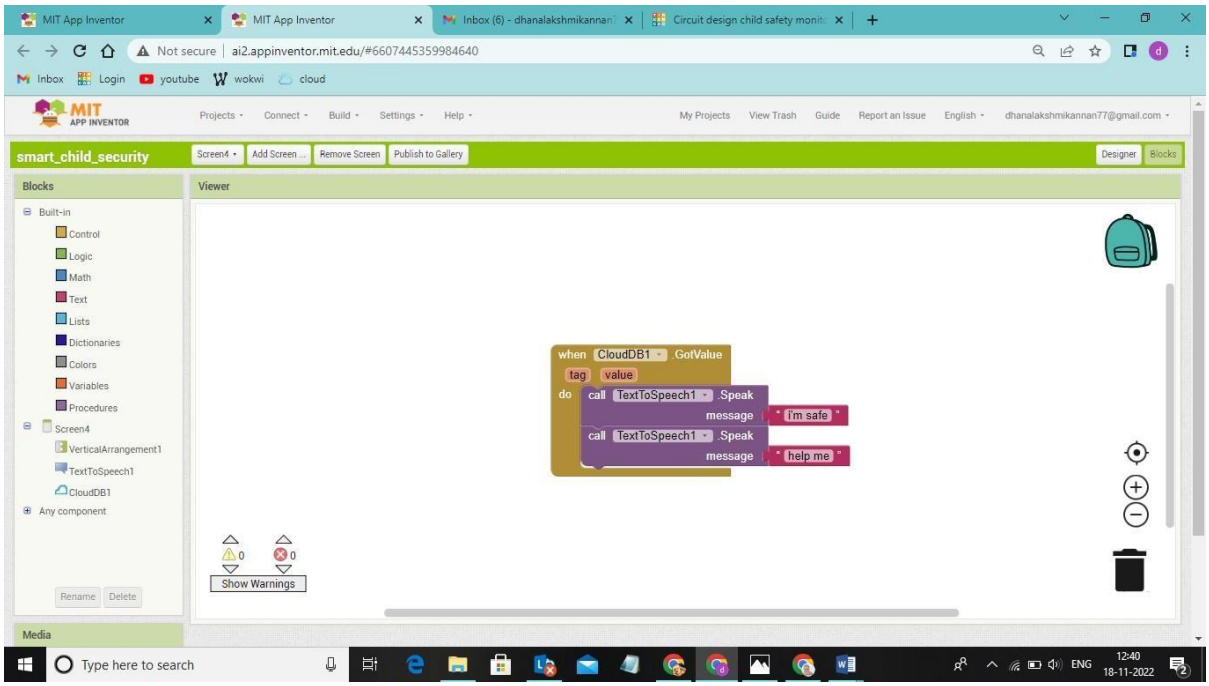
Registration page:



Google map:



Message Receiver:



Traceablity:

The accessibility should cover a wide range of users. A user with disabilities should be able to experience the same benefits as other users.