```
//Libraries required
#include <Wire.h>
#include <Adafruit_PN532.h>
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

int menuChoice = 0; //For menu switch
int pushbutton= D5; //Pushbutton Pin
int buzzer = D7; // Buzzer Pin
bool oPressed = false;//Variable declared to keep track of whether push
button is pressed or not

// Wifi network station credentials
#define WIFI_SSID "SMART-7543"
#define WIFI_PASSWORD "051266303928777711557"

// Telegram BOT Token (Get from Botfather)
#define BOT_TOKEN "7165516005:AAHbNzySc6Y7Teg1vHJBSwul91tOg_kFArI"
const unsigned long BOT_MTBS = 1000; // mean time between scan messages
X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);
unsigned long bot_lasttime; // last time messages' scan has been done

bool stopCommandReceived = false;
bool shouldPrintMenu = false;
String chat_id = ""; // store the chat_id for sending messages from
serial input

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(128,64, &Wire, -1);

//Function for handling telegram chat input and output
void handleNewMessages(int numNewMessages)
{
    Serial.print("handleNewMessages ");
    Serial.println(numNewMessages);

    for (int i = 0; i < numNewMessages; i++)
    {
```

```
        chat_id = bot.messages[i].chat_id; // Save the chat_id to use
in serial communication
        String text = bot.messages[i].text;

        String from_name = bot.messages[i].from_name;
        if (from_name == "")
            from_name = "Guest";

        if (text =="/menu")
        {
            String menu = "**************HealthyIndex*************";
            menu += "\nMenu : \n ";
            menu += "1. HealthCalc \n";
            menu += "2. NutriBreak \n";
            menu += "3. NutriText \n";
            bot.sendMessage(chat_id, menu);
        }
        else if (text == "1")
        {
          displayOutput("Select choice : 1");
          displayOutput("Please Scan your product!");
          bot.sendMessage(chat_id,"You have Selected  HealthCalc! \n
Please scan your product and once done with scanning press the
pushbutton!");
            menuChoice = 1;
            scanCard();
        }
        else if (text == "2")
        {
             displayOutput("Select choice : 2");
             bot.sendMessage(chat_id,"You have Selected  NutriBreak! \n
Please scan your product and once done with scanning Please send the
/stop command to return to the main menu!");
            menuChoice = 2;
            scanCardWithInfo();
        }
        else if (text == "3")
        {
             displayOutput("Select choice : 3");
               bot.sendMessage(chat_id,"You have Selected  NutriText!
\n Please scan your product and once done with scanning Please send the
/stop command to return to the main menu!");
            menuChoice = 3;
```

```
            scanCardWithInfo2();
        }
        else if (text == "stop")
        {
            printMenu();
            menuChoice = 0;


        }
        else
        {
            bot.sendMessage(chat_id, "Unknown command: " + text, "");
        }
    }
}



// Create the PN532 instance
Adafruit_PN532 nfc(-1, -1);

// Constants
const int MAX_VALUES = 10; // Maximum number of values to store in the
array

// Global variables
uint8_t scannedValues[MAX_VALUES]; // Array to store scanned values
int numScannedValues = 0; // Number of scanned values stored in the
array
int totalSum = 0; // Total sum of scanned values

void setup(void) {
  Serial.begin(115200);
  Serial.println();
  pinMode(pushbutton,INPUT);
  delay(10);
    if(!display.begin(SSD1306_SWITCHCAPVCC,0x3c)) {
    Serial.println(F("SSD1306 allocation failed"));
    while(true);
    }

  // attempt to connect to Wifi network:
  configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
```

```
  secured_client.setTrustAnchors(&cert); // Add root certificate for
api.telegram.org
  Serial.print("Connecting to Wifi SSID ");
  Serial.print(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.print("\nWiFi connected. IP address: ");
  Serial.println(WiFi.localIP());

  // Check NTP/Time
  Serial.print("Retrieving time: ");
  time_t now = time(nullptr);
  while (now < 24 * 3600)
  {
    Serial.print(".");
    delay(100);
    now = time(nullptr);
  }
  Serial.println(now);
  pinMode(buzzer, OUTPUT);
  Serial.begin(115200);
  while (!Serial)
    delay(10);

  Serial.println("Hello!");

  nfc.begin();

  uint32_t versiondata = nfc.getFirmwareVersion();
  if (!versiondata) {
    Serial.print("Didn't find PN53x board");
    while (1); // halt
  }
  Serial.println("Waiting for an NFC Card ...");

  // Print the menu options
  printMenu();

}
```

```
void loop(void){
    if (millis() - bot_lasttime > BOT_MTBS)
    {
        int numNewMessages = bot.getUpdates(bot.last_message_received +
1);

        while (numNewMessages)
        {
            Serial.println("got response");
            handleNewMessages(numNewMessages);
            numNewMessages = bot.getUpdates(bot.last_message_received +
1);
        }

        bot_lasttime = millis();
    }

//For Serial handling serial/telegram window input
    if (menuChoice == 0) {
        if (Serial.available() > 0) {
            char inputChar = Serial.read();
            if (inputChar == '1' || inputChar == '2' || inputChar ==
'3') {
                menuChoice = inputChar - '0';
                Serial.print("Selected option:" + String(menuChoice));
                Serial.println(menuChoice);
            } else {
                Serial.println("Invalid choice, please select 1, 2 or
3.");
                printMenu();
            }
        }
    } else if (menuChoice == 1) {
        Serial.print("Please Scan each product in front of scanner and
press pushbutton once you are done!");
        displayOutput("Please scan your \nproduct\n\nand once
done\npress the pushbutton");
        int digitalState=digitalRead(pushbutton);
        if (!oPressed) {
            if (digitalState==0){
                oPressed = true; // Set the flag
                Serial.print("Final total sum: ");
```

```
                Serial.println(totalSum);
                if (totalSum > 150) {
                    displayOutput("Healthy!\n\nYour Health Score is
:"+String(totalSum));
                    bot.sendMessage(chat_id, "Fantastic choices! You're
a Health Rockstar! Keep shining!");
                } else if (totalSum > 100 && totalSum < 1502) {
                    displayOutput("Moderatly Healthy!\n\nYour Health
Score is :"+String(totalSum));
                    bot.sendMessage(chat_id, "You're making good
choices! Keep going, you're on your way to becoming a Health
Rockstar!");
                } else {
                   displayOutput("Not Healthy!\n\nYour Health Score is
:"+String(totalSum));
                    bot.sendMessage(chat_id, "Your choices aren't the
best :(, but don't worry! Start your health journey today, and every
step counts!");
                    digitalWrite(buzzer, HIGH);
                    delay(1000);
                    digitalWrite(buzzer, LOW);
                }
                ESP.restart();
            }
        }
        if (!oPressed) {
            scanCard();
        }
        delay(1000);
    } else if (menuChoice == 2) {

            scanCardWithInfo();
        }
     else if (menuChoice == 3) {
            scanCardWithInfo2();
        }

}



void printMenu() {
```

```
  displayOutput("\n******* Menu ********\n1.HealthCalc
\n\n2.NutriBreak\n\n3.NutriText");
}

void scanCard() {
    if (oPressed) {
        return; // Exit the function if 'o' is pressed
    }

    uint8_t success;
    uint8_t uid[] = {0, 0, 0, 0, 0, 0, 0}; // Buffer to store the
returned UID
    uint8_t uidLength;                       // Length of the UID (4 or
7 bytes depending on ISO14443A card type)

    // Wait for an NTAG203 card. When one is found 'uid' will be
populated with
    // the UID, and uidLength will indicate the size of the UUID
(normally 7)
    success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
&uidLength);

    if (success) {
        if (uidLength == 7) {
            uint8_t data[4]; // Assuming each page contains 4 bytes of
data
            String extractedData = ""; // Initialize an empty string to
store the extracted data

            // Iterate through pages 6 and 7 to extract data
            for (uint8_t i = 6; i <= 7; i++) {
                success = nfc.ntag2xx_ReadPage(i, data);

                // Extract data and append to extractedData string
                if (success) {
                    // Print raw data for debugging
                    for (int j = 0; j < 4; j++) {
                        Serial.print(data[j], HEX); Serial.print(" ");
                        // Filter out non-numeric characters (ASCII
values 48-57 correspond to digits '0' to '9')
                        if (data[j] >= 48 && data[j] <= 57) {
                            extractedData += char(data[j]);
                        }
```

```
                    }
                    Serial.println();
                } else {
                    Serial.println("Unable to read the requested
page!");
                }
            }

            // Limit the length of the extracted data to avoid
concatenating from multiple reads
            if (extractedData.length() > 2) {
                extractedData = extractedData.substring(0, 2);
            }

            // Print extracted data for debugging
            Serial.print("Extracted Data: ");
Serial.println(extractedData);

            // Convert the extracted data to an integer value
            int scannedValue = extractedData.toInt();

            // Store the scanned value in the array if there is space
available
            if (numScannedValues < MAX_VALUES) {
                scannedValues[numScannedValues] = scannedValue;
                numScannedValues++;

                // Add the scanned value to the total sum
                totalSum += scannedValue;
            }

            // Print the scanned value and total sum
            Serial.print("Health Index: ");
            Serial.println(scannedValue);
            Serial.print("Index Sum : ");
            Serial.println(totalSum);
            displayOutput("Health Index " + String(scannedValue));
            if(scannedValue==63)
            {
              displayOutput("\nScanned Product :\n\nGrilled Chicken");
            }
            else if(scannedValue==62)
            {
```

```
              displayOutput("\nScanned Product \n\nLettuce");
            }
            else if(scannedValue==70)
            {
              displayOutput("\nScanned Product :\n\nTomatoes");
            }
            else
            {
              displayOutput("Please scan it again!");
            }

        } else {
            Serial.println("This doesn't seem to be an NTAG203 tag
(UUID length != 7 bytes)!");
        }
    }

    // Delay before scanning for another tag
    delay(2000);
}

void scanCardWithInfo() {


    uint8_t success;
    uint8_t uid[] = {0, 0, 0, 0, 0, 0, 0}; // Buffer to store the
returned UID
    uint8_t uidLength;                     // Length of the UID (4 or
7 bytes depending on ISO14443A card type)

    success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
&uidLength);

    if (success) {
        if (uidLength == 7) {
            uint8_t data[4]; // Assuming each page contains 4 bytes of
data
            String extractedData = ""; // Initialize an empty string to
store the extracted data

            for (uint8_t i = 6; i <= 7; i++) {
                success = nfc.ntag2xx_ReadPage(i, data);
```

```
                if (success) {
                    for (int j = 0; j < 4; j++) {
                        if (data[j] >= 48 && data[j] <= 57) {
                            extractedData += char(data[j]);
                        }
                    }
                } else {
                    Serial.println("Unable to read the requested
page!");
                }

                // Check if the stop command has been received and exit
if so
            }

             if (extractedData.length() > 2) {
                extractedData = extractedData.substring(0, 2);
            }

            // Print extracted data for debugging
            Serial.print("Extracted Data: ");
Serial.println(extractedData);


            int scannedValue = extractedData.toInt();

            Serial.print("Scanned value: ");
            Serial.println(scannedValue);

            displayInfo1(scannedValue);
        } else {
            Serial.println("This doesn't seem to be an NTAG203 tag
(UUID length != 7 bytes)!");
        }
    }
}

void scanCardWithInfo2() {

    uint8_t success;
    uint8_t uid[] = {0, 0, 0, 0, 0, 0, 0}; // Buffer to store the
returned UID
```

```
    uint8_t uidLength;                        // Length of the UID (4 or
7 bytes depending on ISO14443A card type)

    success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
&uidLength);

    if (success) {
        if (uidLength == 7) {
            uint8_t data[4]; // Assuming each page contains 4 bytes of
data
            String extractedData = ""; // Initialize an empty string to
store the extracted data

            for (uint8_t i = 6; i <= 7; i++) {
                success = nfc.ntag2xx_ReadPage(i, data);

                if (success) {
                    for (int j = 0; j < 4; j++) {
                        if (data[j] >= 48 && data[j] <= 57) {
                            extractedData += char(data[j]);
                        }
                    }
                } else {
                    Serial.println("Unable to read the requested
page!");
                }


            }
             if (extractedData.length() > 2) {
                extractedData = extractedData.substring(0, 2);
            }

            // Print extracted data for debugging
            Serial.print("Extracted Data: ");
Serial.println(extractedData);


            int scannedValue = extractedData.toInt();

            Serial.print("Scanned value: ");
            Serial.println(scannedValue);
```

```
            displayInfo2(scannedValue);
        } else {
            Serial.println("This doesn't seem to be an NTAG203 tag
(UUID length != 7 bytes)!");
        }
    }
}


void displayInfo1(int scannedValue) {
  // Display related information about the scanned value
  // This is just a placeholder, you can customize this to display
actual information
  displayOutput("Product ID : \n "+String(scannedValue));
  if(scannedValue==70)
  {
    displayOutput("Name: Tomatoes\n--------------\nCalories : 32 \nFat
0.4g : 1%\nCarbohydrate 7g: 3 %\nFiber 2.2g : 8 %\nProtein 1.6g : 3
%");
  }
  else if(scannedValue==62)
  {
    displayOutput("Name: Lettuce\n--------------\nCalories : 5.4\nFat
0.4g : 0% \nCarbohydrate 7g: 0 %\nFiber 2.2g : 2 %\nProtein 1.6g : 1
%");
  }
  else if(scannedValue==63)
  {
    displayOutput("Name:Grilled Chicken\n--------------\nCalories :
128\nFat 2.7g : 3%\nCarbohydrate 0g: 0 %\nFiber 0g : 0 %\nProtein 26g :
52 %");
  }

}


void displayInfo2(int scannedValue) {
  // Display related information about the scanned value
  // This is just a placeholder, you can customize this to display
actual information
  displayOutput("Check your phone!");
  if(scannedValue==70)
  {
```

```
    bot.sendMessage(chat_id,"Name: Tomatoes\n--------------\nCalories
: 32 \nFat 0.4g : 1%\nCarbohydrate 7g: 3 %\nFiber 2.2g : 8 %\nProtein
1.6g : 3 %");

  }
  else if(scannedValue==62)
  {
    bot.sendMessage(chat_id,"Name: Lettuce\n--------------\nCalories :
5.4\nFat 0.4g : 0% \nCarbohydrate 7g: 0 %\nFiber 2.2g : 2 %\nProtein
1.6g : 1 %");

  }
  else if(scannedValue==63)
  {
    bot.sendMessage(chat_id,"Name: Grilled
Chicken\n--------------\nCalories : 128\nFat 2.7g : 3%\nCarbohydrate
0g: 0 %\nFiber 0g : 0 %\nProtein 26g : 52 % ");

  }

}


//Function for displaying output OLED display
void displayOutput(String output) {
    Serial.println(output);
    display.clearDisplay();
    display.setTextSize(0.2);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.println(output);
    display.display();
}
```

```
/*
Code References
[1] Witnessmenow. (n.d.). GitHub - witnessmenow/Universal-Arduino-Telegram-Bot: Use Telegram on your Arduino (ESP8266 or Wifi-101 boards). GitHub. https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot
[2] For code improvisation- OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. https://chat.openai.com/chat
[3] elechouse/PN532: NFC library for Arduino using PN532. (n.d.). GitHub. https://github.com/elechouse/PN532
[4] Adafruit. (n.d.). Adafruit/ADAFRUIT_SSD1306: Arduino Library for SSD1306 monochrome 128x64 and 128x32 oleds. GitHub. https://github.com/adafruit/Adafruit_SSD1306
*/
```