NANDHA ENGINEERING COLLEGE

ERODE–638052 (Autonomous)

(Affiliated to Anna University, Chennai)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PBL REPORT

22AIC14 – INTERNET OF THINGS AND ITS APPLICATION

MINI PROJECT REPORT ON

TOPIC - SMART HEALTH MONITORING SYSTEM

Submitted by

NAME	REGISTER NUMBER
22AI041	ROHAN.M
22AI055	THARANIDHARAN.M
22AI057	VISHNU.A

NANDHA ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

This is to certify that the project work entitled "SMART HEALTH MONITORING SYSTEM" is the Bonafide work of ROHAN.M (22AI041), THARANIDHARAN.M (22AI055), VISHNU.A (22AI057) who carried out the work under my supervision.

Signature of the Supervisor
Dr. K. Lalitha,
Professor,
Department of AI & DS,
Nandha Engineering College,
Erode – 638052.

Signature of the HOD
Dr. P. Karunakaran,
Head of the Department,
Department of AI & DS,
Nandha Engineering College,
Erode – 638052.

Submitted for End semester PBL review held on _____

SMART HEALTH MONITORING SYSTEM

AIM:

To develop a Smart Health Monitoring System using ESP32, a pulse sensor, andBlynk software for real-time pulse monitoring. The system integrates with a website to display the patient's past medical history alongside current health datafor efficient healthcare management

SCOPE:

The **Smart Health Monitoring System** uses an **ESP32 microcontroller** and a **pulse sensor** to monitor and display a patient's real-time **pulse rate** via the **Blynk mobile app**. The system also integrates with a **dedicated website** that stores the past medical report along with live pulse rate.

BRIEF HISTORY:

The development of smart health monitoring systems has advanced with innovations in wearable technologies and IoT. Early systems focused on basic health parameters, while modern solutions integrate real-time monitoring using devices like the ESP32 and pulse sensors. Tools like Blynk software allow for easy mobile app creation to visualize data. These systems enable continuous health tracking and provide both real-time and historical data through web and mobile platforms, offering enhanced remote healthcare management.

PROPOSED METHODOLOGY:

1. **Hardware Setup**:

   ESP32 Microcontroller: Connects the pulse sensor to the system and processes data.

   Pulse Sensor: Monitors the user's pulse rate in real-time**.**

2. **Data Collection & Transmission**:

   The pulse sensor sends data to the ESP32, which transmits it to the Blynk app for real-time monitoring and a cloud database for storing historical data.

3. **Mobile App Integration (Blynk)**:

   The pBlynk app displays the current pulse rate and provides alertsfor abnormal reading.

4. **Web Interface**:

   A dedicated website shows the current pulse rate from Blynk andthe patient's past medical history from the database.

5. **Data Synchronization**:

   Data is continuously synced between the ESP32, Blynk app, and website, ensuring up-to-date information across platforms.

6. **Security**:

   Data encryption and secure authentication ensure the privacy andsafety of patient information.

COMPONENTS REQUIRED:

S.NO	COMPONENTS	QUANTITY
1	ESP32	1
2	PULSE SENSOR	1
3	BLYNK SOFTWARE	SOFTWARE

DESCRIPTION:

The Smart Health Monitoring System is designed to provide real-time tracking and management of a patient's health using an ESP32 microcontroller, a pulse sensor, and Blynk software. The system continuously monitors the patient's pulserate and transmits the data to the Blynk app for instant feedback.

Additionally, the system features a dedicated website that displays the current pulse rate in real-time, fetched directly from the Blynk app. The website also shows the patient's past medical history, which is securely stored in a cloud-based database or server.

This system offers the following benefits:

Real-time Monitoring: Patients can track their pulse rate in real-time usingthe Blynk app on mobile devices.

Historical Data Access: Healthcare providers or patients can access past medical records via the website, allowing for continuous health monitoringand informed decision-making.

Remote Health Management: The system enables remote health monitoring, making it easy for patients to track their health at home and forhealthcare providers to manage patient data remotely.

User-friendly Interface: Both the Blynk app and the website are designed to be intuitive, displaying health data clearly and simply

Security and Privacy: The system ensures the privacy of sensitive medicalinformation through data encryption and secure access controls.

This integrated solution improves healthcare efficiency, empowers patients with self-monitoring, and facilitates better management by healthcare professionals through accessible, real-time data.

PROTOCOLS:

The system utilizes several protocols to ensure smooth functionality: HTTP is used for communication between the website and Blynk's cloud API to fetch real-time pulse rate data; Wi-Fi connects the ESP32 to the internet for data transmission to the Blynk cloud; SQL queries retrieve and manage patient medical history from the database; and REST API integrates real-time Blynk data into the website. Optionally, MQTT can be employed for efficient, lightweight, real-time messaging between the ESP32 and the server. These protocols enable seamless data exchange and system integration

CODING:

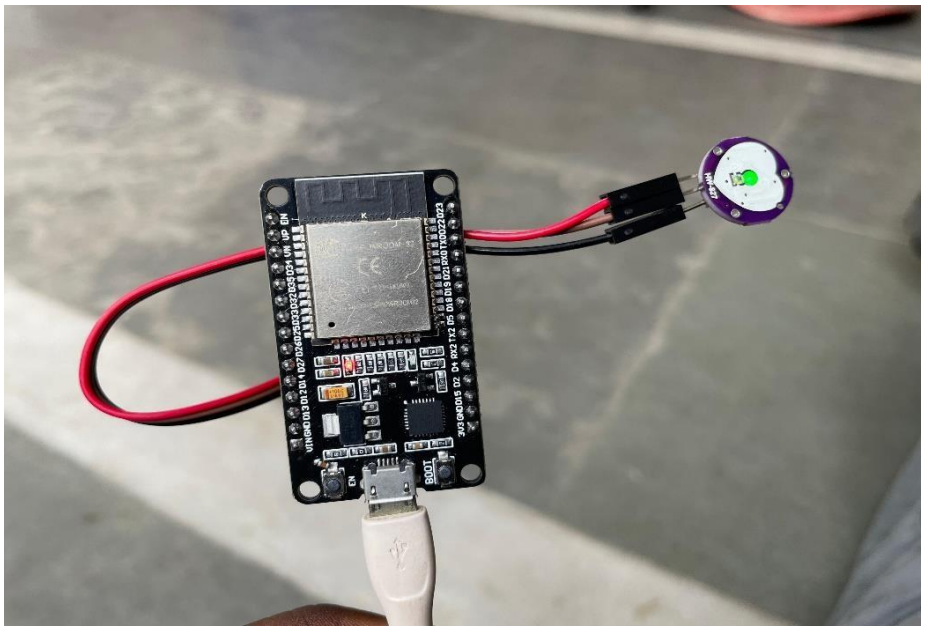
```
#define BLYNK_TEMPLATE_ID "TMPL3U7cpL6MD"
#define BLYNK_TEMPLATE_NAME "Pulse BPM"
#define BLYNK_AUTH_TOKEN "Ggl1cp59zomRm0moNL2gQ2BuILQL-Eue"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// Wi-Fi credentials
const char* ssid = "vivot15g";          // Replace with your Wi-Fi SSID
const char* password = "";  // Replace with your Wi-Fi password
```

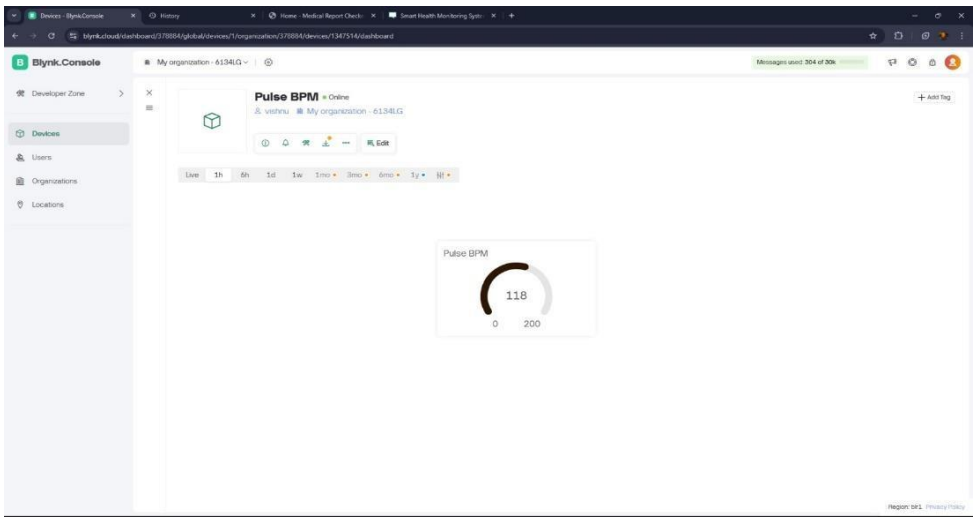
```
// Pulse Sensor Pin

const int pulsePin = 34; // GPIO34 (analog input on ESP32)
volatile int pulseCount = 0; // Counts the pulses
unsigned long lastPulseTime = 0; // Tracks the last pulse time

const int debounceInterval = 100; // Minimum interval between pulses (ms)
unsigned long lastTime = 0;
const int interval = 10000; // Update interval in milliseconds (10 seconds)

// Interrupt to count pulses with debouncing
void IRAM_ATTR countPulse() {
  unsigned long currentTime = millis();
  if (currentTime - lastPulseTime > debounceInterval) {
    pulseCount++;
    lastPulseTime = currentTime;
  }
}

void setup() {
  // Start Serial Monitor
  Serial.begin(115200);

  // Pulse Sensor Pin Setup
  pinMode(pulsePin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(pulsePin), countPulse, RISING);

  // Connect to Wi-Fi
  Serial.print("Connecting to Wi-Fi");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("\nConnected to Wi-Fi");

  // Connect to Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, password);
  Serial.println("Connected to Blynk!");
}

void loop() {
  Blynk.run(); // Run Blynk

  unsigned long currentTime = millis();
  if (currentTime - lastTime >= interval) {
    // Calculate BPM (pulses in 10 seconds * 6 to get BPM)
    int bpm = (pulseCount * 12000) / interval;
    pulseCount = 0; // Reset pulse count
    lastTime = currentTime;
```

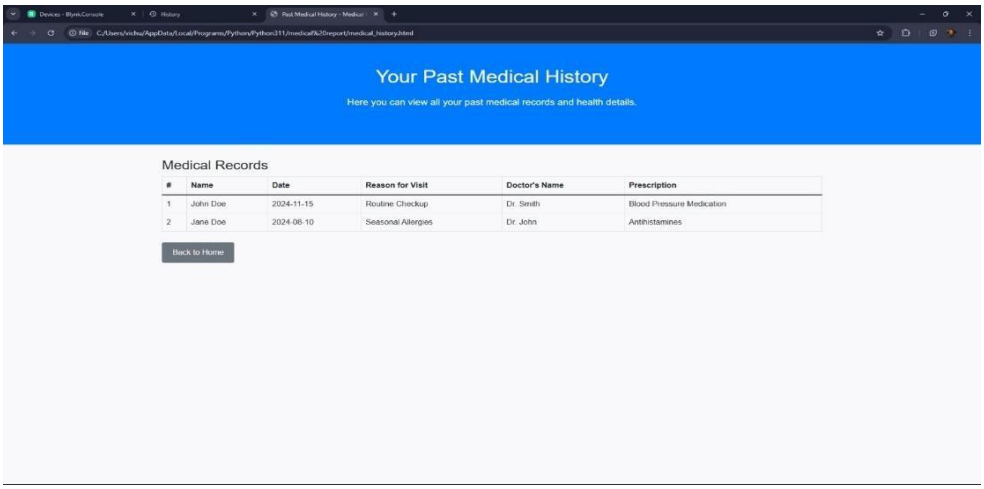
```
// Display BPM on Serial Monitor
    Serial.print("Pulse BPM: ");
    Serial.println(bpm);


// Send BPM to Blynk (Virtual Pin V0)
    Blynk.virtualWrite(V1, bpm);
  }
}
```

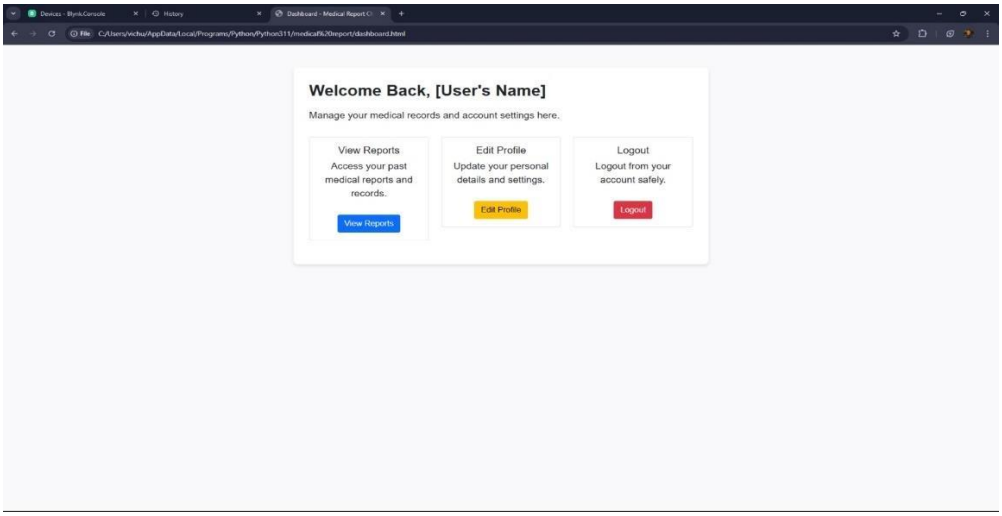
OUTPUT SCREENSHOT:

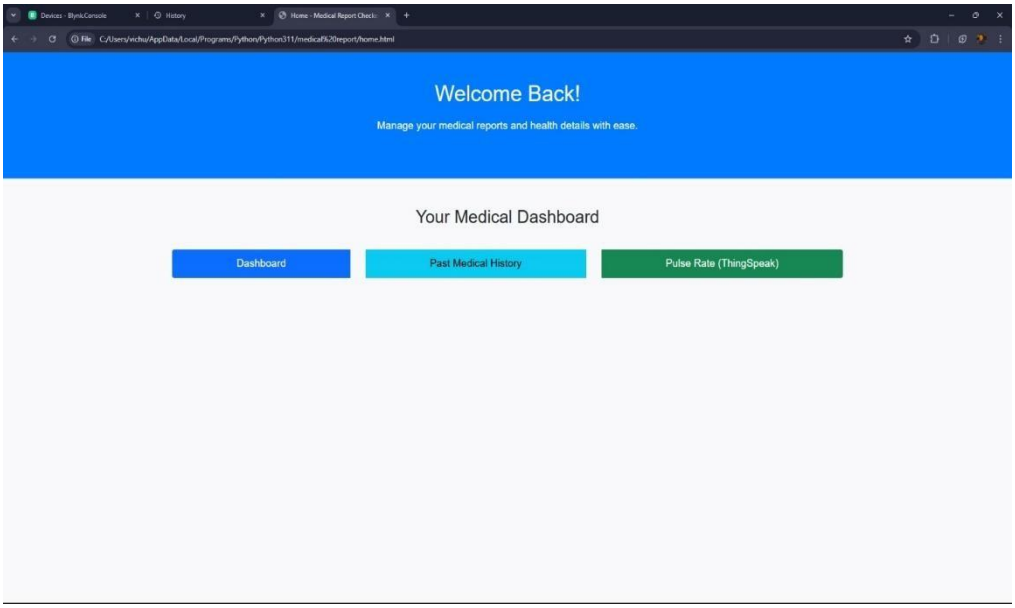
`

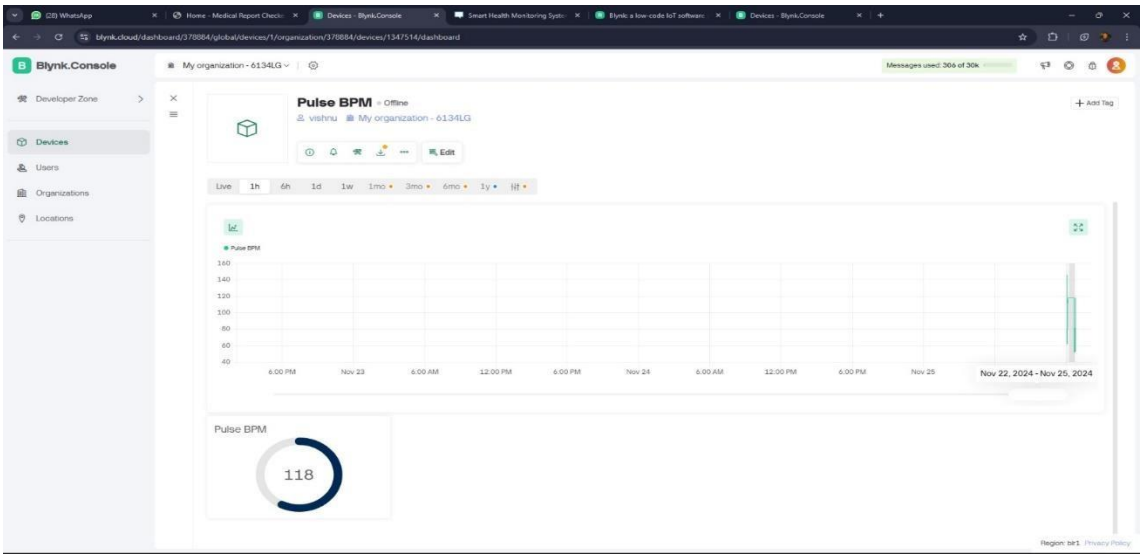



Blynk.Console
Developer Zone
Devices
Users
Organizations
Locations
Pulse BPM
Online
Edit
Pulse BPM
118
0
200


Your Past Medical History
Here you can view all your past medical records and health details.
Medical Records
Name Date Reason for Visit Doctor's Name Prescription
1 John Doe 2024-11-15 Routine Checkup Dr. Smith Blood Pressure Medication
2 Jane Doe 2024-08-10 Seasonal Allergies Dr. John Antihistamines
Back to Home


Welcome Back, [User's Name]
Manage your medical records and account settings here.
View Reports
Access your past medical reports and records.
View Reports
Edit Profile
Update your personal details and settings.
Edit Profile
Logout
Logout from your account safely.
Logout


Welcome Back!
Manage your medical reports and health details with ease.
Your Medical Dashboard
Dashboard
Past Medical History
Pulse Rate (ThingSpeak)


Blynk.Console
Developer Zone
Devices
Users
Organizations
Locations
Pulse BPM
Offline
Edit
Pulse BPM
118
Nov 22, 2024 - Nov 25, 2024

LIMITATIONS:

1. Limited to pulse rate monitoring; lacks support for other vital signs like bloodpressure or oxygen levels.
2. Relies on stable internet connectivity; disruptions impact performance.
3. Battery dependency can lead to frequent recharging or power issues.
4. Potential risks of data breaches despite encryption.
5. Limited scalability for handling multiple patients simultaneously.
6. Elderly or non-tech-savvy users may find the system difficult to use.
7. No advanced analytics for predictive health insights.

FUTURE ENHANCEMENTS:

1.Additional Sensors: Include blood pressure, SpO2, and temperature sensors for comprehensive monitoring.
2.AI Analytics: Implement machine learning for health risk predictions and insights.
3.Offline Functionality: Enable offline data storage and synchronization for uninterrupted use.
4.Wearable Integration: Support wearables like smartwatches for portability.
5.Enhanced Security: Use multi-factor authentication and blockchain for improved data protection.

CONCLUSION:

The Smart Health Monitoring System offers real-time tracking of vital signs like pulse rate using ESP32 and pulse sensors, with data displayed on the Blynk app and a dedicated website. It enables remote monitoring and provides easy access to both currentand historical health data. The system enhances healthcare accessibility, empowers informed decision-making, and ensures secure data management, ultimately improving patient care and health management.