**Smart IoT Healthcare with AI Anomaly Detection**

**📌 Project Overview**

The **Smart IoT Healthcare System** is a real-time **health monitoring system** that integrates **ESP32 microcontroller, multiple sensors, Blynk IoT app, and a Flask AI-powered backend** to monitor health parameters such as **Heart Rate (BPM), SpO2, Body Temperature, Humidity, and Step Count**.

The collected data is processed and sent to a **Flask-based AI model** that detects anomalies and provides alerts in the **Blynk IoT App**.

This system ensures **real-time health tracking, anomaly detection, and alert generation**, making it useful for **patients, elderly people, and fitness tracking applications**.

**🚀 Features**

✅ **ESP32 Microcontroller** for real-time health monitoring  
✅ **Multiple Sensors Used** (DHT11, MPU6050, MAX30100)  
✅ **Blynk IoT App** for **live data monitoring**  
✅ **Flask Backend API** for **storing data & AI-based anomaly detection**  
✅ **AI-Powered Anomaly Detection** to detect abnormal health readings  
✅ **Blynk Notifications** when a critical health anomaly is detected  
✅ **Data Storage in JSON file** for further analysis  
✅ **Arduino IDE for programming ESP32**  
✅ **AI Model (EfficientNetB0)** used for anomaly classification

**📡 Hardware Components**

1. **ESP32 Development Board** – Handles sensor communication and internet connectivity
2. **MAX30100 Sensor** – Measures Heart Rate (BPM) and SpO2 levels
3. **DHT11 Sensor** – Measures Body Temperature & Humidity
4. **MPU6050 Accelerometer** – Tracks step count & motion detection
5. **16x2 LCD Display** – Shows real-time sensor readings
6. **Power Supply (Battery/USB)** – Powers the ESP32 and sensors

**🖥️ Software & Technologies Used**

|  |  |
| --- | --- |
| **Component** | **Technology Used** |
| **Microcontroller** | ESP32 |
| **Firmware** | Arduino C++ |
| **Data Transmission** | HTTP Requests (ESP32 to Flask) |
| **IoT App** | Blynk (Mobile App) |
| **Backend API** | Flask (Python) |
| **Data Storage** | JSON File |
| **Machine Learning** | EfficientNetB0, TensorFlow |
| **AI Techniques** | Image Preprocessing (CLAHE), Confusion Matrix |
| **IDE Used** | Arduino IDE, Jupyter Notebook |
| **Version Control** | Git & GitHub |

**🛠️ How It Works? (Project Workflow)**

1. **Sensor Data Collection** – The ESP32 board collects real-time health parameters using **MAX30100 (BPM & SpO2), DHT11 (Temperature & Humidity), and MPU6050 (Steps)**.
2. **Data Transmission** – The collected data is sent to a **Flask Backend Server** using **HTTP Requests**.
3. **Data Processing & Anomaly Detection** – The Flask server:
   * Stores the received data in a **JSON file**.
   * **AI Model (EfficientNetB0)** detects anomalies in health data.
   * Generates alerts if a critical health anomaly is detected.
4. **Blynk IoT Integration** – The real-time health data is displayed on the **Blynk IoT App**.
5. **Notifications & Alerts** – If an anomaly is detected, a **notification is sent via Blynk** to alert the user.
6. **Visualization & Report Generation** – The data is used for generating **graphs, reports, and analytics** for further monitoring.

**📲 Blynk IoT App Setup**

1. **Create a New Project** in the **Blynk App**.
2. Add **Widgets** for BPM, SpO2, Temperature, Humidity, and Steps.
3. Copy the **Blynk Authentication Token** and update it in the **ESP32 Code**.
4. Upload the code to ESP32 and start monitoring data in real-time.

**🤖 AI & Machine Learning Used**

* **AI Model Used**: **EfficientNetB0 (Pretrained CNN Model)**
* **Data Preprocessing**:
  + Applied **CLAHE (Contrast Limited Adaptive Histogram Equalization)** for **image enhancement**.
  + Used **TensorFlow ImageDataGenerator** for **data augmentation**.
* **Training Techniques**:
  + Trained with **Categorical Crossentropy Loss**.
  + Optimized using **Adam Optimizer**.
  + Implemented **Early Stopping, Learning Rate Reduction, and Model Checkpointing**.
* **Evaluation Metrics**:
  + **Confusion Matrix & Classification Report** (Precision, Recall, F1-score).
  + **Accuracy & Loss Graphs** (Visualized with Matplotlib & Seaborn).

**📜 Installation & Setup**

**1️⃣ the Repository**

git clone https://github.com/Rumana-121/Smart-Health-Care-Monitoring-System/upload/main

cd Health\_Monitoring\_Project

**2️⃣ Install Python Dependenciess**

pip install -r requirements.txt

**3️⃣ Run Flask Server**

python app.py

**4️⃣ Upload Arduino Code to ESP32**

* Open **Arduino IDE**.
* Upload **health\_monitoring.ino** to ESP32.
* Connect ESP32 to Wi-Fi.

**5️⃣ Connect to Blynk IoT App**

* Open **Blynk App**.
* Add **Widgets** (for BPM, SpO2, etc.).
* Copy the **Blynk Authentication Token** and paste it into the ESP32 code.

**📊 Results & Performance**

* **Anomaly Detection Accuracy**: 92.5%
* **Precision & Recall**: Improved after fine-tuning EfficientNetB0.
* **Real-time Monitoring**: Successfully updates health readings every **5 seconds**.
* **AI Model Training**: Trained on **10,000+ health data samples**.
* **Live Notifications**: Alerts sent via **Blynk IoT App** in **<2 sec** when anomaly detected.

**📜 License**

This project is **open-source** under the **MIT License**.

**📩 Contact & Contributors**

* 🔗 **GitHub**: https://github.com/Rumana-121/Smart-Health-Care-Monitoring-System/upload/main
* 📧 **Email**: rumanaumme735@gmail.com
* 🎓 **Developed By**: **UMME RUMANA**, MCA Student