

# Phase 5: Project Demonstration & Documentation

Title:

## **Healthcare Diagnosis and Treatment System**

---

Abstract:

The Healthcare Diagnosis and Treatment System is an intelligent AI-based solution that assists in early diagnosis, provides treatment recommendations, and monitors patient vitals in real-time using IoT technology. In this final phase, the system demonstrates AI logic in diagnosing conditions, IoT data simulation for patient monitoring, and encryption of sensitive health records. It is designed to support hospitals, telemedicine platforms, and personal healthcare devices. This document includes technical walkthroughs, test reports, and simulation using Wokwi to replicate real-time monitoring devices.

## **Index**

1. Project Demonstration
2. Project Documentation
3. Feedback and Final Adjustments
4. Final Project Report Submission
5. Project Handover and Future Works
6. Wokwi Simulation Integration

## 1. Project Demonstration

### Overview:

The system simulates patient symptom input and wearable sensor data to produce a complete health diagnosis and treatment output.

### Demonstration Details:

Symptom Input & AI Prediction: User inputs fever and fatigue, and the AI predicts influenza or anemia.

Personalized Treatment: Based on age and allergies, a treatment plan is recommended.

IoT Monitoring: Real-time data from simulated heart rate, oxygen saturation, and temperature readings trigger alerts.

Encryption Layer: Health reports are encrypted and decrypted to simulate secure medical record transfer.

### Outcome:

The system responded accurately and securely to patient symptoms and health metrics during testing.

## 2. Project Documentation

### Includes:

System Flow Diagram: From user input → diagnosis → treatment → encrypted output.

Python Source Code: Single script file combining AI diagnosis, treatment engine, IoT simulation, and encryption.

User Guide: For doctors and patients interacting with the system.

Admin Manual: Maintenance and encryption key handling.

Testing Report: Encrypted sample output, vital alert triggers, AI predictions, and user feedback logs.

Outcome:

Documentation ensures the project is usable, maintainable, and scalable by future teams or hospitals.

### **3. Feedback and Final Adjustments**

Steps Taken:

Feedback from peers included improving system clarity and treatment explanations.

Thresholds for IoT alerts were refined (e.g., heart rate  $> 100$ , SpO<sub>2</sub>  $< 92$ ).

Added data decryption log for admin review of secure messages.

Outcome:

Final version provided clearer output and faster processing under test load.

### **4. Final Project Report Submission**

Content Summary:

Phase-wise developments

Challenge examples (e.g., overlapping symptoms, sensor lag)

Use of encryption and IoT device simulation

Final integrated workflow

Outcome: Submitted final report documents the entire system lifecycle and realtime use cases.

## **5. Project Handover and Future Works**

Future Scope:

Link to wearable device SDKs (Fitbit, Apple Health).

Add live chatbot for conversational health support.

Expand to mental health and chronic condition monitoring.

Multilingual AI support for diverse populations.

Outcome: System prepared for deployment and long-term improvement.

## **6. Wokwi Simulation Integration**

Overview:

Wokwi (online electronics simulator) was used to simulate real-world IoT sensor data from wearables.

Simulated Components:

LM35 – Body temperature sensor

Pulse sensor (via potentiometer) – Simulates variable heart rate

Analog input – For SpO2 (Oxygen level)

Sample ESP32 Arduino Code:

```
#define TEMP_PIN 34
#define HEART_PIN 35
#define OXYGEN_PIN 32

void setup() { Serial.begin(115200);
}

void loop() {
  int tempRaw = analogRead(TEMP_PIN);
  float tempC = (tempRaw / 4095.0) * 3.3 * 100;

  int heartRaw = analogRead(HEART_PIN);
  int bpm = map(heartRaw, 0, 4095, 60, 120);

  int oxyRaw = analogRead(OXYGEN_PIN);
  int spo2 = map(oxyRaw, 0, 4095, 85, 100);

  Serial.println("---- Patient Vitals ----");
  Serial.print("Temp: "); Serial.println(tempC);
  Serial.print("Heart Rate: "); Serial.println(bpm);
  Serial.print("Oxygen Level: "); Serial.println(spo2);
  delay(2000);
}
```

Outcome:

Wokwi successfully simulated health monitoring, allowing validation of AI response and alerts before real hardware integration.

Note: Attach screenshots of:

AI diagnosis output

Encrypted and decrypted health data

