

## **Phase 5: Project Demonstration & Documentation**

Title:

Natural Disaster Prediction and Management System

Abstract:

The Natural Disaster Prediction and Management System integrates AI, real-time IoT sensor data, and secure encryption to detect and respond to threats such as floods, earthquakes, and cyclones. In this final phase, system capabilities were demonstrated using simulated sensor data (via Python and Wokwi), and encrypted alerts were generated. The project is designed to aid public authorities with quick, accurate forecasting and response planning. Technical documentation, test reports, and simulation diagrams are included to support future deployment and development.

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 demonstration covered AI predictions, real-time sensor simulation, and response alert generation.

### Details:

Prediction Engine: Used simulated values to predict disasters like Flood and Earthquake.

IoT Data: Simulated readings for rainfall, river level, wind speed, and seismic activity.

Response System: Generated tailored recommendations such as evacuation alerts.

Encryption: Alerts were securely encrypted and decrypted using cryptography.fernet.

### Outcome:

The system functioned accurately in real-time with test inputs and showed secure data handling capabilities.

## **2. Project Documentation**

### Includes:

Architecture Diagram: Flow from sensor to AI to alert output.

Source Code: Combined Python script with all logic.

User Manual: For government/emergency staff usage.

Admin Guide: Deployment and alert configuration.

Testing Logs: Output samples and system performance charts.

### Outcome:

Comprehensive documentation allows for future scaling and easy deployment.

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

Enhancements Made:

Refined disaster thresholds to reduce false positives.

Improved encryption error handling.

Increased output clarity for non-technical users.

Outcome:

System performance improved, particularly in alert sensitivity and output formatting.

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

Contents:

Summary of all phases.

Key outcomes and AI logic.

Disaster scenarios tested.

Solutions to real-time sensor data inconsistencies.

Outcome:

Report is complete and well-structured for academic and practical use.

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

Next Steps:

Integrate with government sensor networks (e.g., IMD, NASA).

Enable SMS and voice alerts via Twilio or similar.

Add web dashboard for live disaster updates.

Outcome:

System is ready for pilot testing in rural and coastal areas.

## 6. Wokwi Simulation Integration

### Overview:

To simulate real-time sensor inputs, Wokwi (online electronics simulator) was used with an ESP32 microcontroller.

### Simulated Components:

DHT22 – Simulates temperature/humidity for storm detection.

HC-SR04 – Simulates river water level.

Piezo or Vibration Sensor – Simulates earthquake tremors.

### Wokwi Code Highlights:

```
#define TRIG 12
#define ECHO 14
#define VIBRATION_PIN 27

void loop() {
  // Ultrasonic sensor for river level
  digitalWrite(TRIG, LOW); delayMicroseconds(2);
  digitalWrite(TRIG, HIGH); delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  long duration = pulseIn(ECHO, HIGH);
  float distance = duration * 0.034 / 2;

  int vibration = digitalRead(VIBRATION_PIN);

  Serial.print("River Level: "); Serial.println(distance);
  Serial.print("Vibration: "); Serial.println(vibration ? "YES" : "NO");
}
```

### Outcome:

Wokwi provided a reliable, cost-free simulation of environmental sensors for project validation and demonstration purposes.