COLLEGE CODE : 1128

COLLEGE NAME : T.J.S.E.C

DEPARTMENT : AI&DS

STUDENT NM-ID : aut112823AIDS12

aut112823AIDS34

aut112823AIDS51

aut112823AIDS52

aut112823AIDS28

DATE : 09-05-2025

TECHNOLOGY-PROJECT

NAME :

NATURAL DISASTER PREDICTION AND

MANAGEMENT

SUBMITTED BY,

NAME:

(112823243012) Evangelin.G

(112823243032)Pathakunta Venkateswarlu Reddy

(112823243050)Tharunkumar.D

(112823243051)Ummiti Narendra

(112823243058)Vupati Mohan Krishna

Phase 5: Project Demonstration & Documentation

Title:
Natural Disaster Prediction and Management System

Abstract
The *Natural Disaster Prediction and Management System* project is designed to enhance disaster preparedness and response through the integration of Artificial Intelligence (AI), Machine Learning (ML), Geographic Information Systems (GIS), and Internet of Things (IoT). The final phase brings together predictive modeling, real-time environmental monitoring, and automated alert systems in a unified platform. This document covers a comprehensive demonstration of the system, complete technical documentation, performance analysis, source code, and final testing results. With an emphasis on scalability, accuracy, and timely response, the system provides actionable insights for government agencies, emergency responders, and the general public. Diagrams, dashboards, and codebase visuals are included to present the architecture and operational flow.

Table of Contents

1. Project Demonstration Overview

The system will be demonstrated to showcase its ability to predict natural disasters (earthquakes, floods, wildfires, etc.), process real-time sensor and satellite data, and trigger automated alerts.

Demonstration Details:

- **System Walkthrough:** A step-by-step walkthrough simulating a disaster event, from data ingestion to alert generation and disaster response planning.
- **AI Prediction Accuracy:** Display of real-time model predictions using historical and live data, including early warnings for potential threats.
- **IoT Integration:** Demonstration of real-time monitoring from deployed environmental sensors (e.g., seismic activity, water levels, temperature, wind).
- **Performance Metrics:** Presentation of system latency, accuracy, and response time under simulated disaster load conditions.

- **Security & Privacy:** Explanation of secure data transmission, access control for agencies, and privacy measures for sensitive geolocation data.

Outcome:

Stakeholders will gain confidence in the system's ability to handle high-pressure scenarios, predict threats with high accuracy, and support coordinated emergency responses.

2. Project Documentation Overview

This section provides detailed documentation covering the full architecture and all functional components of the system.

Documentation Sections:

- **System Architecture:** Visual diagrams of data pipelines, AI model workflows, and IoT device communication flow.
- **Code Documentation:** In-depth explanations of modules including ML training scripts, disaster prediction algorithms, alert systems, and dashboard UI.
- **User Guide:** Manual for civilians and agencies on how to use the platform, interpret warnings, and respond appropriately.
- **Administrator Guide:** Operational procedures for maintaining hardware, updating models, and monitoring system performance.
- **Testing Reports:** Summarized results from load testing, model accuracy validation, system uptime monitoring, and data integrity checks.

Outcome:

A thorough documentation set that supports maintainability, reproducibility, and future system upgrades.

3. Feedback and Final Adjustments Overview

Post-demonstration feedback from domain experts, emergency services, and pilot users will be used to fine-tune the system.

Steps:

- **Feedback Collection:** Feedback via surveys, interviews, and performance observations from pilot deployments and demo sessions.
- **Refinement:** Addressing prediction inaccuracies, UI/UX feedback, false alert reduction, and enhancing multi-language support for alerts.
- **Final Testing:** Complete testing cycle to ensure refinements are stable, performance-optimized, and ready for field use.

Outcome:

The final system reflects practical usability and reliability enhancements based on real stakeholder inputs.

4. Final Project Report Submission Overview

A detailed report summarizing the full development cycle, key achievements, challenges, and readiness for real-world deployment.

Report Sections:

- **Executive Summary:** Overview of the project vision, features, and achievements.
- **Phase Breakdown:** Detailed account of each development phase — data collection, AI model training, alert logic, GIS integration, and sensor deployment.

- **Challenges & Solutions:** Discussion on technical and logistical challenges (e.g., real-time data noise, alert sensitivity tuning) and how they were resolved.
- **Outcomes:** Summary of operational capabilities, including predictive accuracy, real-time alerting efficiency, and multi-agency interoperability.

Outcome:

A comprehensive report that captures the journey from conception to a near-deployable solution with national or regional applicability.

5. Project Handover and Future Works Overview

The project is formally handed over with a vision for future enhancement and broader adoption.

Handover Details:

- **Next Steps:** Recommendations for expanding coverage (more sensors, more regions), enhancing prediction accuracy with more data sources, and integrating with national emergency databases and communication systems.
- **Sustainability Guidelines:** Steps for ensuring long-term operability, periodic model retraining, and system updates.

Outcome:

The system is ready for pilot deployment and further scaling, with a clear path for future research and operational integration.

```python
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# 1. Simulated Dataset (you can replace this with real-time data)
data = {
    'rainfall_mm': np.random.randint(50, 300, 100),
    'river_level_m': np.random.uniform(1.0, 5.0, 100),
    'soil_moisture_pct': np.random.randint(10, 100, 100),
    'humidity_pct': np.random.randint(40, 100, 100),
    'flood_risk': [1 if x > 200 and y > 3 else 0 for x, y in zip
                   (np.random.randint(50, 300, 100), np.random.uniform(1.0, 5.0, 100))]
}

df = pd.DataFrame(data)
```

```python
# 2. Split data
X = df[['rainfall_mm', 'river_level_m', 'soil_moisture_pct', 'humidity_pct']]
y = df['flood_risk']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 3. Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# 4. Predict and evaluate
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")

# 5. Disaster Management Logic
def manage_disaster(rainfall, river_level, soil_moisture, humidity):
    prediction = model.predict([[rainfall, river_level, soil_moisture, humidity]])[0]
    if prediction == 1:
        print("⚠️ Flood risk detected! Initiate disaster response protocol.")
```

```python
        # Simulated actions
        print("- Alert emergency services")
        print("- Notify citizens via SMS & public channels")
        print("- Activate evacuation plan")
    else:
        print("✅ No immediate flood risk.")

# Example usage
manage_disaster(rainfall=250, river_level=4.2, soil_moisture=85, humidity=90)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\naren\OneDrive\Desktop\html.demo> & C:/Users/naren/AppData/Local/Programs/Python/Python313/python.exe c:/Users/naren/OneDrive/Desktop/html.demo/nm.py
Model Accuracy: 70.00%
C:\Users\naren\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature names, but RandomForestClassifier
as fitted with feature names
  warnings.warn(
✅ No immediate flood risk.
```