COLLEGE CODE:3108

COLLEGE NAME:-Jeppiaar Engineering College

DEPARTMENT::Information Technology

STUDENT NM-ID :
F8226D1E2F2723194184E48093E5379F

ROLL NO:310823205069

DATE:14-05-2025

TECHNOLOGY-PROJECT NAME:
Natural Disaster Prediction and Management

SUBMITTED BY,

Pravin N
Tamil.M
Udhaya
surya.S
Ragul

Phase 5: Project Demonstration & Documentation

Title: Natural Disaster Prediction and Management

Abstract:

The Natural Disaster Prediction and Management system leverages machine learning, satellite data, and real-time environmental sensor networks to predict and manage the impact of natural disasters. This final project phase outlines the full system capabilities, including disaster detection algorithms, emergency alert systems, real-time data analytics, and community response tools. The solution is designed for scalability, accuracy, and integration with emergency response infrastructure. Screenshots and code snapshots will support system explanation.

Index

1. Project Demonstration

Overview:

The Natural Disaster Prediction and Management system will be demonstrated in a live environment, showcasing prediction accuracy, user alerts, and dashboard analytics.

Demonstration Details:

- System Walkthrough: From input data ingestion to disaster prediction output.
- Prediction Engine: Demonstration of ML model performance with historical data.
- Real-time Alerts: Display of notification system for emergency response.

Phase 5: Project Demonstration & Documentation

- Performance Metrics: Accuracy, false positives, and system response time.

- Security: Measures for securing sensitive location and population data.

Outcome:

The demonstration confirms system readiness for real-time deployment in disaster-prone regions.

2. Project Documentation

Overview:

Complete technical and user documentation accompanies this project, covering system architecture, algorithms, deployment steps, and maintenance.

Documentation Sections:

- Architecture: Flowcharts and system diagrams.

- Codebase: Modular code breakdown, ML models, and alert systems.

- User Manual: Guide for community and admin portal users.

- Admin Guide: Instructions for model retraining and monitoring.

- Testing Reports: Model validation, system load testing, and latency reports.

Outcome:

The documentation ensures smooth deployment, support, and further development.

3. Feedback and Final Adjustments

Overview:

User and stakeholder feedback during demonstration phases will guide final tweaks and tuning.

Phase 5: Project Demonstration & Documentation

Steps:

- Feedback Collection: Surveys from potential users, NGOs, and emergency teams.

- Refinement: Algorithm fine-tuning and UI adjustments.

- Final Testing: System retesting under simulated disaster events.

Outcome:

The final version incorporates practical suggestions for robustness and usability.

4. Final Project Report Submission

Overview:

A comprehensive final report consolidates all progress and technical insights.

Report Sections:

- Executive Summary

- Phase Summaries

- Challenges & Solutions

- Outcomes and Impact Analysis

Outcome:

This final report reflects the full development lifecycle and readiness for real-world use.

5. Project Handover and Future Works

Overview:

The project is prepared for handover with guidelines for future improvements.

Phase 5: Project Demonstration & Documentation

Handover Details:

- Next Steps: Suggestions like multi-language support, AI-enhanced rescue routing, and satellite integration.

Outcome:

A ready-to-deploy system with potential for expansion into broader disaster management platforms.

```python
# app.py
from flask import Flask, request, jsonify
import joblib

app = Flask(__name__)
model = joblib.load('flood_model.pkl')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    prediction =
model.predict([[data['rainfall'],
data['river_level'],
data['soil_moisture']]])
    return jsonify({'flood_risk': 'Yes'
if prediction[0] == 1 else 'No'})

if __name__ == '__main__':
    app.run(debug=True)
```

OUT PUT


```
{
  "rainfall": "heavy",
  "river_level": "high",
  "soil_moisture": "wet"
}
```