Phase 3: Implementation of Project

Title: Natural Disaster Prediction and Management System

Objective

The goal of Phase 3 is to implement the core components of the Natural Disaster Prediction and Management System. This includes the deployment of AI-based prediction models, multichannel alert systems, community reporting platforms, and foundational blockchain integration for relief management.

1. AI Model Development

Overview:

The system will utilize AI/ML models trained on historical and real-time data for early detection of disasters such as floods, earthquakes, and cyclones.

Implementation:

- Collect and preprocess historical and real-time data.
- Train AI models to identify patterns and provide predictive alerts.
- Validate prediction accuracy with recent case studies.

Outcome:

AI models will provide short-term predictions for natural disasters and support risk scoring in vulnerable regions.

2. Alert System Development

Overview:

A multichannel alert system will be built to disseminate early warnings to the public.

Phase 3: Implementation of Project

Implementation:

- Develop SMS and mobile app-based notifications.

- Use geofencing to target messages to specific locations.

- Integrate with local radio and sirens for remote areas.

Outcome:

A working prototype capable of delivering region-specific alerts through multiple channels.

3. Community Reporting and Dashboard

Overview:

Create a platform where citizens can report real-time information, including photos and safety status.

Implementation:

- Develop an intuitive web/mobile interface.

- Enable map-based input and viewing.

- Include AI moderation of user-submitted reports.

Outcome:

An operational community reporting system with map and real-time data visualization.

4. Blockchain for Relief Management

Overview:

Implement a blockchain ledger to track disaster relief distribution.

Phase 3: Implementation of Project

Implementation:

- Use blockchain for tamper-proof recording of aid flow.

- Develop smart contracts for transparency in fund allocation.

Outcome:

A basic blockchain system for tracking aid materials and financial disbursement post-disaster.

5. Testing and Feedback Collection

Overview:

System components will undergo initial testing for functionality, reliability, and user interaction.

Implementation:

- Conduct user testing in selected regions.

- Collect feedback from government agencies and public users.

Outcome:

Feedback-driven improvements for Phase 4 and successful pilot test results.

Challenges and Solutions

1. Data Availability

   - Challenge: Limited access to real-time disaster data.

   - Solution: Collaborate with meteorological departments and use open datasets.

2. Infrastructure Limitations

Phase 3: Implementation of Project

- Challenge: Internet and mobile access may be unreliable in rural areas.
- Solution: Use SMS and radio-based alert systems to cover non-internet regions.

3. Public Engagement

- Challenge: Low participation in community reporting.
- Solution: Awareness campaigns and gamification to encourage user reporting.

Outcomes of Phase 3

- Functional AI models with predictive capabilities.
- Operational multichannel disaster alert system.
- Live dashboard with community reports and geospatial data.
- Blockchain system prototype for relief tracking.
- User feedback collected from simulations and pilot runs.

Next Steps for Phase 4

- Enhancing AI accuracy with larger datasets.
- Expanding reach with multilingual and voice-based interfaces.
- Scaling up system deployment in collaboration with national disaster agencies.

SCREENSHOTS OF CODE and PROGRESS - MUST BE ADDED HERE FOR PHASE 3

SCREENSHOTS OF CODE AND PROGRESS

Sample Code: Flood Risk Prediction using Random Forest

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

# Load dataset
data = pd.read_csv('flood_data.csv')

# Preprocess
features = data[['rainfall', 'river_level', 'soil_moisture']]
labels = data['flood_risk']

# Split data
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.3, random_sta

# Train model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Evaluate
predictions = model.predict(X_test)
print(classification_report(y_test, predictions))
```