Project Title:

Smart Prediction System for Building and Property Collapse Detection

---

Objective:

To design and implement an intelligent system capable of predicting potential building collapses by collecting real-time sensor data and analyzing it using machine learning models.

Problem Statement:

Many buildings and residential properties in Egypt and similar countries are at risk of collapse due to poor maintenance, aging infrastructure, and undetected structural weaknesses. These failures often occur without early warning, leading to devastating losses. The aim of this project is to build a predictive system that provides early alerts to prevent casualties and material damage.

System Overview:

The system combines:

- Embedded hardware to collect real-world signals.
- AI models to predict collapse probability.
- Cloud/Local dashboard to visualize and monitor the status.

Hardware Components:

1. ESP32 or Raspberry Pi – main controller to handle sensor data.
2. Vibration sensor (e.g., SW-420 or ADXL345) – detect unusual vibrations.
3. Accelerometer/Gyroscope (MPU6050) – measure tilt, lean, or sudden motion.
4. Ultrasonic sensor – measure wall displacement or movement.
5. Moisture sensor / Humidity sensor (DHT11 or DHT22) – check humidity and water exposure in walls.
6. Temperature sensor (optional) – track environmental changes.
7. Camera (optional) – to visually detect cracks using image analysis.
8. Battery pack or solar power (for deployment in field)
9. SD card or Wi-Fi module for data logging and transmission

Software Components:

1. Firmware for ESP32/Pi – written in C++ or MicroPython to read and send sensor data.
2. Backend and Data pipeline – Python scripts to collect, preprocess and store data.
3. Machine Learning model:
   - Trained using scikit-learn or TensorFlow.
   - Input: sensor readings (vibration, humidity, tilt, displacement...)
   - Output: risk score (Low – Medium – High risk of collapse)
4. Web Dashboard / Mobile App:
   - Framework: Flask / Streamlit / Firebase
   - Features: Real-time graph, Alerts, History log
5. Notification System:
   - SMS or email alerts (via Twilio or similar)
   - Optional: IoT buzzer or LED indicator on hardware

Steps and Timeline:

Step 1: Research and Design (Week 1–2)

- Study structural failure patterns
- Define key features to detect (e.g., vibration thresholds, tilt angle)
- Draft system architecture

Step 2: Hardware Assembly (Week 2–3)

- Connect sensors to ESP32 / Raspberry Pi
- Write test code to read each sensor independently

Step 3: Data Collection (Week 3–5)

- Place sensors on different test structures (e.g., old buildings, simulations)
- Log sensor data under normal and stressed conditions

Step 4: Dataset Preparation (Week 5)

- Label collected data manually (stable vs. unstable states)
- Normalize and clean the dataset

Step 5: Model Development (Week 6–7)

- Choose ML model (Random Forest, SVM, or Neural Network)
- Train and evaluate accuracy
- Export model for use on-device or via server

Step 6: Integration (Week 8)

- Merge ML output with dashboard & hardware alerts
- Test full pipeline

Step 7: Testing & Deployment (Week 9–10)

- Simulate real-world conditions (humidity, tremors, shifting)
- Measure precision and false positives

Step 8: Report, Documentation, and Presentation (Week 11–12)

- Write final report and system documentation
- Prepare presentation and poster for evaluation

Dataset Sources:

1. Custom-collected dataset – via your own sensors placed on test structures.
2. Kaggle or research datasets:
   - "Concrete Crack Images for Classification"
   - "Building Structural Health Monitoring"
   - Search terms: "Building collapse prediction", "Structural sensor dataset", "Crack detection in buildings"
3. Synthetic dataset – simulate fake data that mimics sensor patterns for training

---

Optional Extensions:

- Image processing using CNN to detect cracks
- GPS tagging and mapping risk levels per region
- Edge AI model using TinyML for offline prediction

Expected Outcome:

A fully working prototype that can monitor physical building properties, predict risk levels, and alert residents or engineers to take action early – potentially saving lives and property.