Project Name: Salinity Sentinel – A Smart Soil Monitoring System for Coastal Bangladesh

Group Name: Innovation Instigators

Group Members:
1. Name: S.M Azman Sikder Durjay | ID: 20245103133
2. Name: Mostar-Shid Billah | ID: 20245103139
3. Name: Dipto Dey | ID: 20245103143
4. Name: Shudipto Ghosh | ID: 20245103151
5. Name: Yeasin Arafat Nayem | ID: 20245103160

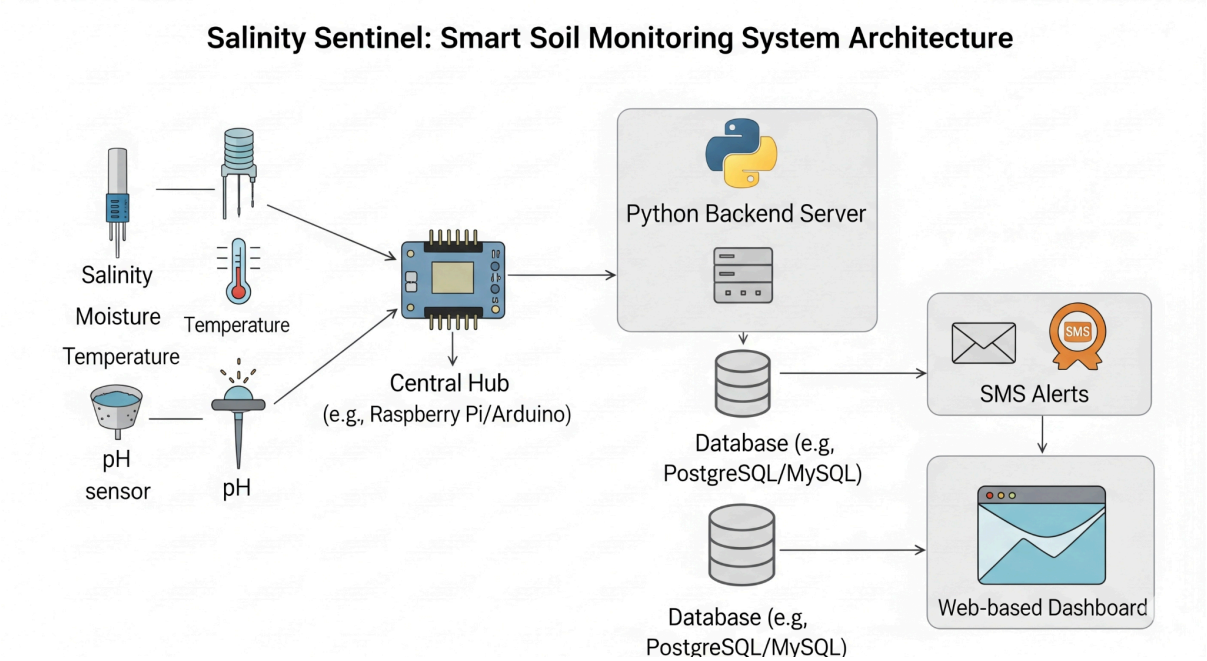
Introduction

The coastal regions of Bangladesh are severely affected by soil salinity, which directly hampers agricultural productivity and the livelihoods of farmers. Due to climate change, rising sea levels, and saline water intrusion, this problem is worsening. Most farmers are unaware of these changes in real time, which leads to crop damage and financial losses. Salinity Sentinel is a smart, low-cost soil monitoring system that uses sensors and Python-based software to monitor soil conditions. It can measure salinity, moisture, temperature, and pH levels in real time and send alerts when necessary, helping users take preventive measures to protect crops.

Key Features

● Real-time measurement of soil salinity, moisture, temperature, and pH
● Web-based dashboard for viewing live and historical data
● Alert system via Email or SMS when thresholds are crossed
● Integration with weather APIs for contextual insights
● Database storage for long-term trend analysis
● (Optional) Machine Learning for forecasting future salinity levels

Diagram



Technologies To Be Used

For Front-End:
● **HTML:** "skeleton" of the dashboard, defining where text, images, and data visualizations go.
● **CSS:** make the dashboard look good, applying colors, fonts, layouts, and ensuring it's responsive on different screen sizes (e.g., desktop, mobile).
● **JavaScript Framework (e.g., React, Vue.js, or Angular):** a structured way to manage the user interface
● **Charting Library (e.g., Chart.js, D3.js, or Plotly.js):** create beautiful and interactive data visualizations.

For Back-End :
● **Python Framework (e.g., Flask or Django):** This is the core of our back end.
● **Database (e.g., PostgreSQL or MySQL):** to store the long-term sensor data.
● **API (Application Programming Interface).**
● **Weather API:** To provide contextual insights, the back end will make requests to an external weather service