Salinity Sentinel – A Smart Soil Monitoring System for Coastal Bangladesh

SUBMITTED BY

Name	ID	Intake	
Yeasin Arafat Nayem	ID: 20245103160	Intake: 53	Sec: 2
Tasnuba Tarannum Taha	ID: 20245103071	Intake: 53	Sec: 2
Mansura Akter Mim	ID: 20244103173	Intake: 54	Sec: 5
Israt Jahan	ID: 20244103167	Intake: 54	Sec: 5

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)

Date: 18th November, 2025

Abstract

This project focuses on developing a Soil Management System that analyzes soil health based on stored soil data and recommends suitable crops according to soil pH and type. The system retrieves soil properties from a database, processes the values, and provides scientific decision-support for farmers. This digital solution aims to replace traditional manual soil testing difficulties with an automated, reliable, and efficient model.

Table of Contents

Chapter 1: INTRODUCTION

1.1 Problem Specification

Soil plays a crucial role in agriculture, especially in coastal areas of Bangladesh where salinity is a major concern. Most farmers still rely on manual notes or assumptions when managing soil conditions. Without a proper system, critical information such as soil pH, salinity level, moisture, crop history, and fertilizer usage is often lost or not recorded correctly. This leads to poor decision-making, improper fertilizer use, unsuitable crop selection, and long-term soil degradation.

1.2 Objectives

- Provide a smart, database-driven soil monitoring system for coastal regions.
- Store, track, and update soil data such as pH, salinity, moisture, soil type, fertilizer use, and crop history.
- Analyze soil conditions automatically and give meaningful recommendations.
- Warn users about over-fertilization or unsuitable soil conditions.
- Suggest suitable crops based on soil type and pH.

1.3 Scope

- Manual soil data entry with fields such as soil type, region, pH level, moisture, fertilizer used, and previous crop.
- Automatic analysis: pH classification, salinity evaluation, soil-type-based recommendations.
- Warning system for overuse of fertilizers.
- Crop recommendation engine based on soil type and pH.
- Data visualization using charts for better interpretation.
- Useful for farmers, agriculture students, researchers, and organizations.

1.4 Organization of Project Report of Project Report**

- **Chapter 1** explains problem, objective, and scope.
- **Chapter 2** discusses existing solutions and theoretical background.
- **Chapter 3** includes tools, models, system diagrams, and database schema.
- **Chapter 4** describes front-end and back-end implementation.
- **Chapter 5** provides system requirements and user guide.
- **Chapter 6** includes conclusion, limitations, and future improvements.

Chapter 2: BACKGROUND

2.1 Existing System Analysis

Existing soil testing systems include:

- **Manual soil testing labs** – accurate but time-consuming and costly.
- **Portable soil kits** – quick but often inaccurate.
- **Mobile apps** – many lack scientific accuracy and depend on user input.

Pros of digital systems:
✔ Faster processing
✔ Automated analysis
✔ Easy access to data

Cons:
✘ Dependency on device and database
✘ Requires initial calibration and setup

2.2 Supporting Literature

This project uses:

- Soil science (pH scale, crop suitability).
- Database management (db.sqlite).
- Web technologies (HTML, CSS, JS, Django).
- Algorithms for decision-making.
  These tools ensure efficient data storage, accurate soil analysis, and real-time results.

Chapter 3: SYSTEM ANALYSIS & DESIGN

3.1 Technology & Tools

- **Backend:** Python (Django Framework)
- **Frontend:** HTML, CSS, JavaScript
- **Charts/Visualization:** Chart.js
- **Database:** SQLite (lightweight and easy to deploy)
- **Tools Used:** VS Code, GitHub, Browser

3.2 Model & Diagram**

3.2.1 SDLC Model (Waterfall)

The project follows the Waterfall Model due to its structured phases:

- Requirements → Design → Implementation → Testing → Deployment

3.2.2 System Architecture

- User Interface
- Backend Logic Layer
- Soil Analyzer Module
- Crop Recommendation Engine
- Database Layer

3.2.3 Use Case Diagram

Actors: User, System
Features: Soil data input, view analysis, crop recommendation

3.2.4 Context Level Diagram (DFD-0)

Shows interaction between user and Soil Management System.

3.2.5 Data Flow Diagram (DFD-1)

- Soil Data → Analyzer → Recommendation Engine → Output

3.2.6 Database Schema

Tables:

- **soil_data** (id, ph_value, soil_type, moisture, nitrogen, date)
- **crop_list** (id, crop_name, ph_min, ph_max, soil_type)

3.2.7 Algorithms/Flowchart

Basic crop recommendation algorithm:

```
IF ph_value BETWEEN ph_min AND ph_max AND soil_type MATCHES
    SHOW suitable crop
ELSE
    SHOW no match found
```

Chapter 4: IMPLEMENTATION

4.1 Interface/Front-End Design

- Simple, clean UI using HTML and CSS.
- User-friendly soil data input forms.
- Chart-based soil analysis reports using Chart.js.

4.2 Back-End Design

- Built with Python Django to manage soil entries and compute recommendations.
- Uses SQLite database for storing soil records, fertilizer data, and crop history.
- Implements logic for analyzing soil pH, salinity, and fertilizer frequency.

4.3 Modules/Features**

- Soil Data Entry Module
- Soil Health Analyzer
- Crop Suggestion Module
- Database Management
- User Dashboard

Chapter 5: USER MANUAL

5.1 System Requirements

5.1.1 Hardware

- Any modern PC or laptop

5.1.2 Software

- Browser
- XAMPP / Python / Node environment

5.2 User Interfaces

Panel A: Soil Data Viewer

Panel B: Crop Recommendation Output

Login Credentials: admin / user

Chapter 6: CONCLUSION

6.1 Conclusion

The Soil Management System provides automated soil health analysis using scientific methods. It supports sustainable agriculture and improves decision-making for farmers.

6.2 Limitations

- Depends on database accuracy.
- Limited to pH and soil type.

6.3 Future Work

- Add fertilizer recommendations.
- Integrate real-time sensor data.
- Add weather-based predictions.