SMART SOIL

A PROJECT REPORT
BY

TEAM NO. 1

SHASHWAT RANJAN (E23CSEU0067)
SWAYAM AGARWAL (E23CSEU0063)
SABHYA RAJVANSHI(E23CSEU0079)

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

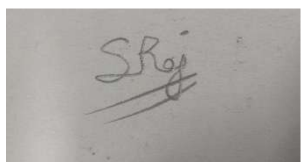
April 2025

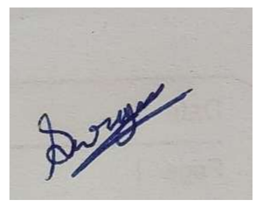
DECLARATION

We hereby declare that the work which is being presented in the report entitled

"Smart Soil", is an authentic record of our own work carried out during the period from JAN, 2025 to April, 2025 at School of Computer Science and Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report has not been submitted by us for the award of any other degree elsewhere.





Signature of Candidate

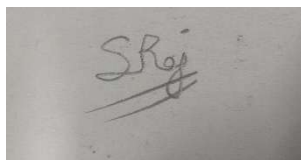
Shashwat Ranjan
(Enroll. No. E23CSEU0067)

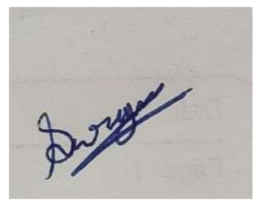
Swayam Agarwal
(Enroll. No. E23CSEU0063)

Sabhya Rajvanshi
(Enroll. No. E23CSEU0079)

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deepest gratitude to our mentor, **Dr. Nitin Shelke (Faculty)** for guiding, supporting, and helping us in every possible way. We were extremely fortunate to have him as our mentor as he provided insightful solutions to problems faced by us thus contributing immensely towards the completion of this capstone project. We would also like to express our deepest gratitude to VC, DEAN, HOD, faculty members and friends who helped us in successful completion of this capstone project.





Signature of Candidate

Shashwat Ranjan
(Enroll. No. E23CSEU0067)

Swayam Agarwal
(Enroll. No. E23CSEU0063)

Sabhya Rajvanshi
(Enroll. No. E23CSEU0079)

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The **Smart Soil** project presents an innovative approach to modern agriculture by integrating soil health monitoring with artificial intelligence and sensor-based technology. The goal of the project is to assist farmers and agricultural professionals in making informed decisions to improve crop yield, reduce resource wastage, and promote sustainable farming practices.

This system utilizes a combination of IoT sensors embedded in the soil to capture real-time data on key parameters such as moisture levels, temperature, pH, and nutrient content. The collected data is processed using a machine learning model trained to analyze patterns and diagnose soil conditions. Based on this analysis, the system provides actionable recommendations to the user through an intuitive web interface or mobile application.

To enhance decision-making accuracy, the project incorporates a custom-built AI chatbot capable of addressing user queries related to soil health, crop compatibility, and treatment suggestions. This chatbot leverages natural language processing to interpret user input and deliver context-aware responses, thereby improving user engagement and accessibility. Additionally, the system includes a visual analytics dashboard that displays graphs and insights related to soil parameters over time, enabling users to track changes and take proactive measures.

The significance of the **Smart Soil** project lies in its potential to transform traditional agricultural practices through automation, data-driven insights, and user-friendly interfaces. By bridging the gap between technology and agriculture, this project offers a scalable solution for efficient soil management and supports the broader goal of achieving food security and environmental conservation.

1. INTRODUCTION

Agriculture lies at the center of international food security, but agriculture is threatened by climate change, volatile weather, and wasteful water use. Experience-based decision making is the basis for traditional farming practices, and this is bound to result in poor resource allocation. Among the most important factors that control crop yield is soil moisture, which has a direct influence on plant growth and irrigation scheduling.

To overcome this challenge, AI-based soil moisture forecasting has become a game-changing technique in precision agriculture. Farmers can schedule irrigation, minimize water loss, and maximize crop yield with machine learning and real-time environmental information. This project will create an AI model that forecasts soil moisture level at a specific location and offers data-driven crop advice based on the forecasted moisture level.

Major Features of the Project:

- Soil Moisture Forecasting: Applies AI algorithms to simulate environmental factors (temperature, humidity, etc.) and predict soil moisture.

- Crop Recommendation System: Suggests suitable crops in expectation of future moisture levels for maximum yield and effective use of resources.

- Sustainable Farming Practices: Minimizes water consumption and promotes precision agriculture.

This project will make a big impact in smart farming, and farming decisions will be assisted by AI intelligence instead of intuition.

1.1 Problem Statement

Agriculture is the backbone of global food supply, but farmers are often overwhelmed with inefficient irrigation and variable crop selection based on unpredictable soil moisture levels. Traditional farming relies on soil observation or stationary weather predictions, leading to over-irrigation, water waste, and inefficient crop yield.

There is no existing, AI-driven system that can determine soil moisture content precisely and provide data-driven crop recommendations based on real-time environmental conditions. Farmers need an efficient solution to simplify irrigation schedules and select the most suitable crops based on soil moisture content to achieve sustainable agriculture and higher productivity.

This project attempts to address this issue by developing an AI-driven soil moisture forecasting model that takes various environmental factors such as temperature, humidity, rainfall, and soil type into consideration. The system will recommend crops that are most suitable for the soil based on these forecasts, allowing farmers to make the correct decision and reduce wastage of resources.

- Critical Issues: Lack of adequate appropriate real-time soil moisture prediction tools for farmers.

- Excessive use of water leading to over-irrigation or water stress in plants.

- Poor crop selection due to insufficient data-driven insights.

  Need for a smart, AI-based system to boost agricultural productivity and sustainability.

By solving these problems, our system will provide farmers precision agriculture capabilities and thus improved yield, water efficiency, and sustainable farming.

2. BACKGROUND RESEARCH

India is predominantly an agrarian economy, with nearly 60% of the rural population dependent on agriculture for their livelihood. However, the sector faces numerous challenges including fragmented land holdings, unpredictable rainfall, and limited access to real-time soil data, especially in remote and underdeveloped regions. Despite the increasing availability of satellite imagery and open data sources, the lack of localized, tech-enabled decision-making tools has led to inefficient farming practices and suboptimal yields. Recognizing this gap, I chose to design Smart Soil with a focus on Indian geography, crop patterns, and climatic zones. By using region-specific rules for crop recommendation and leveraging India's diverse landscape in the training dataset, the model becomes not just a proof of concept, but a practically usable system tailored to the needs of Indian farmers. The project aligns with national goals like Digital India and Doubling Farmers' Income, aiming to democratize precision agriculture using open-source AI tools.

2.1 Proposed System

This project aims to develop an AI-based soil moisture prediction and crop recommendation system to enhance precision agriculture. The system, based on machine learning algorithms, considers real-time environmental information, historical soil moisture trends, and weather forecasts to predict soil moisture and recommend the optimal crops.

Vision

Our aim is to equip farmers with evidence-based information to enhance the efficiency of irrigation, minimize wastage of water, and enhance crop selection. This will result in an increase in agricultural output, minimized consumption of resources, and sustainable farming.

Problem Solved

Conventional farming techniques are based on instinctive soil testing and plant choice, and these usually lead to inefficient watering, poor harvests, and excess water use. This project is addressing all these problems through the provision of an AI-supported real-time decision-making system for intelligent irrigation scheduling and enhanced plant suitability analysis.

2.2 Goals and Objectives

The main objective of the project is the creation of an AI-driven soil moisture prediction and crop advisory system that provides precision agriculture by giving data-driven input to farmers. The system should be scalable, accurate, and easy to use and make the most use of water and maximize the yield.

3. PROJECT PLANNING

This section covers the details of the project planning. Selecting the lifecycle of the development, project stakeholders, resources required, assumptions made (if any) are detailed in the sections below.

3.1. Project Lifecycle

The creation of the AI-based Soil Detection and Crop Recommendation System is an Agile project, with flexibility, iterative enhancement, and constant feedback incorporation. The project is divided into sets of phases, each with defined objectives and deliverables. The team operated in a SCRUM-like manner, with high-intensity meetings, collaboration, and incremental enhancement.

The project life cycle is divided into the following major phases:

- Week 1: Project Idea & Research

  The project idea of soil detection from Sentinel-1 and Sentinel-2 satellite data and an AI prediction model was solidified. Initial studies were conducted on Google Earth Engine API, Sentinel bands, their application in soil and crop prediction, and existing similar solutions to determine areas of differentiation for this project.

- Week 2: Technology Stack & Literature Review
  1. Python-based Backend AI Processing
  2. API Integration: Flask
  3. Frontend Development: HTML, CSS, JavaScript

Experiments were carried out on soil moisture forecasting methods, Sentinel-2 pre-processing, and machine learning algorithms like XGBoost, Random Forest, and TensorFlow. The team also worked on the Google Maps and Google Earth Engine APIs to allow effective data extraction and processing.

- Week 3: Planning & Frontend Development Starts
  1. System design structure like activity selection diagrams, data flow diagrams, and pseudocode for major functions.

  2. Frontend development started with the basic features of login and registration pages with OTP verification.

  3. UI dashboard, research area, and map interface design was planned in order to design a user-friendly and seamless experience.

- Week 4: Feature Design & Prototyping
  - Wireframes and prototypes of key features such as real-time location display, weather condition integration, and AI-based soil & crop prediction dashboard were created.

- Backend architecture for user authentication, API calls, and data processing was implemented.

- Google Maps API was utilized for real-time geolocation monitoring.

- Week 5: Prototype Testing & Refinements

    1. Usability testing was conducted to enhance UI performance and responsiveness.

    2. Improvements in OTP login and verification system offered a simple user authentication process.

    3. API calls were optimized to enhance the precision of location-based forecasts and weather fetching.

    4. Security enhancements were implemented to prevent unauthorized access.

- Week 6: Data Collection & Preprocessing

    - Satellite imagery was downloaded from Google Earth Engine.

    - Sentinel-2 bands useful for crop forecasting and soil moisture were identified and processed.

    - Geolocation mapping and preprocessing of data were done to provide organized data.

    - A formatted CSV dataset was used for model training.

Week 7: Backend Integration, Feature Engineering & EDA

- Exploratory Data Analysis (EDA) was conducted in order to plot soil moisture distribution and crop types.

- Feature engineering methods were used to improve model input features.

- Backend integration began, which encompassed:

- Post-login rendering

- Razorpay integration and API installation

- Google Earth Engine Sentinel-2 API setup

- Simulation of research tab for scientific paper presentations

Week 8: Integrating Google Maps API, Training & Tuning Certain of the machine learning algorithms (XGBoost, Random Forest) were attempted for the problem of detecting soil and compared. Hyperparameter tuning was also performed to obtain maximum model accuracy. The previous AI model was deployed to the backend for real-time prediction retrieval based on the user geolocation. Final testing and performance verification were conducted to ensure system efficiency and accuracy.

3.2 Future Updates

- Using CNN based model training to classify soils and its types for better and uttermost crop recommendation.

4. SYSTEM ANALYSIS AND DESIGN

This section describes in detail about the design part of the system.

4.1 Overall Description

This project will combine data engineering, machine learning, and frontend development to develop a smart, user-friendly system that improves decision-making and automation. The system will use AI/ML models trained on structured data to enable predictive insights without compromising the ease of use through an intuitive frontend. The data pipeline, developed by data engineers, will enable efficient data fetching, transformation, and storage in a manner that delivers high-quality and well-formatted input to train and run models. The system, using AI techniques, will analyze patterns and provide actionable recommendations, leading to enhanced precision and efficiency in decision-making.

The project will have a very structured workflow beginning from data acquisition and preprocessing, followed by model development and deployment. AI/ML engineers will deploy and tune the models in a manner that they execute optimally in real-time scenarios. Frontend engineers will develop an interactive and responsive interface that displays insights in a manner understandable by end users. The team will emphasize scalability, usability, and inclusion of automated workflows in the development process. The ultimate goal is to provide a robust platform to enable data-driven decision-making and an optimal and delightful user experience.

4.2 Design diagrams/Architecture/ UML diagrams/ Flow Charts/ E-R diagrams

- Use Case: Use Case Diagram
- Activity Selection: Activity Selection Diagram

4.2.1 Product Backlog Items

- As a farmer, I want to scan my soil using the AI tool, so that I can get instant analysis results.
- As a researcher, I want to upload soil data for analysis, so that I can study different soil compositions.
- As a farmer, I want the system to suggest fertilizers based on my soil's condition, so that I can improve soil health.
- As a farmer, I want the system to recommend the best crops for my soil type, so that I can maximize my yield.

4.2.1 Use Case Diagram

Smart Soil System

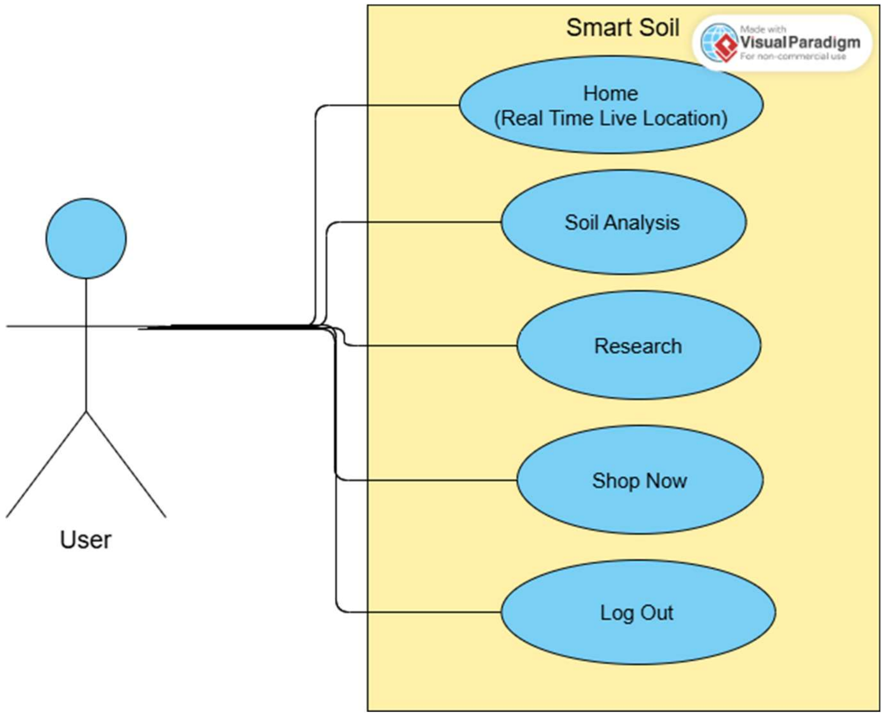


Figure 1: sample 1

4.2.1 Activity Diagrams

Smart Soil System

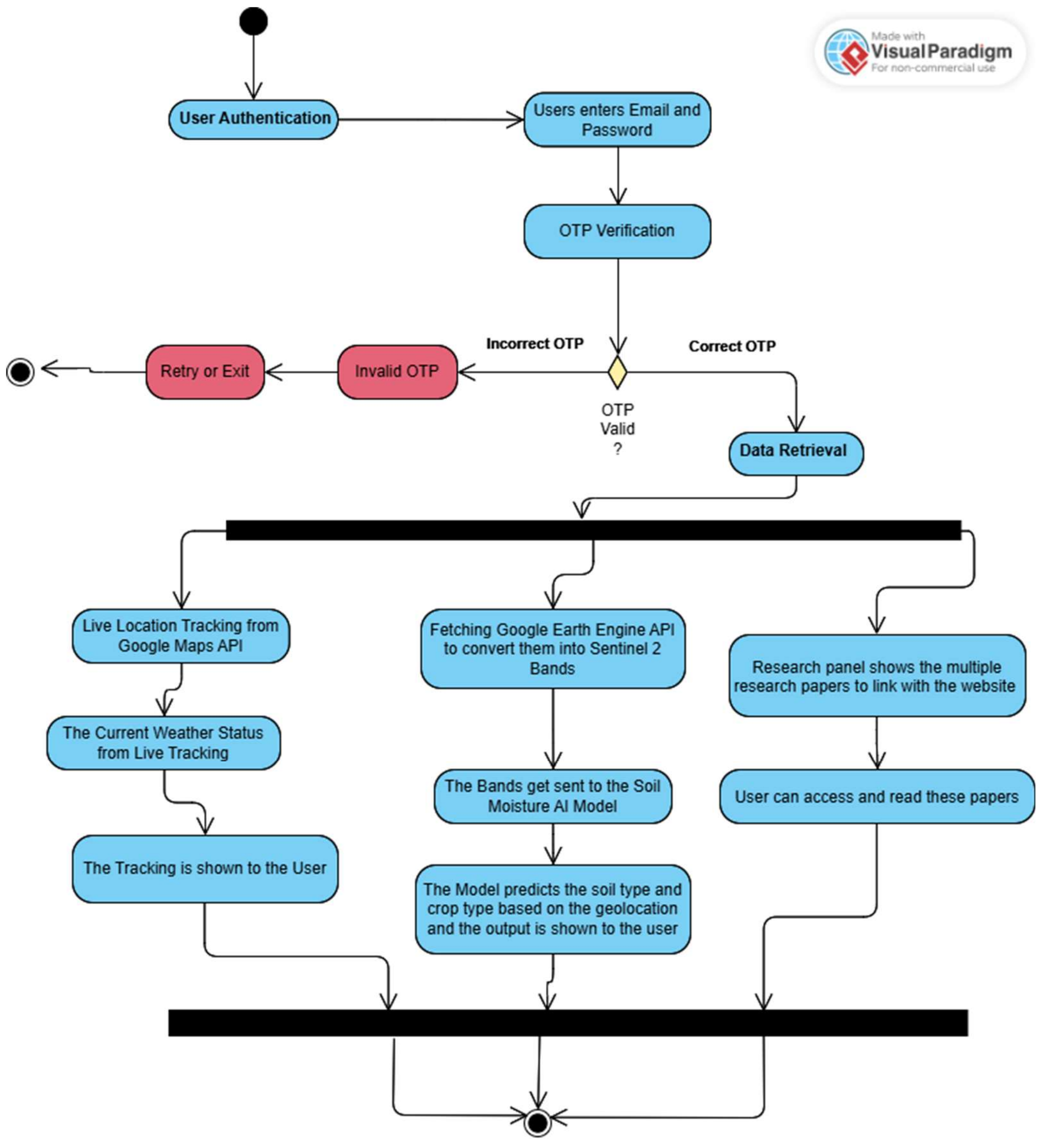


Figure 2: sample 2

5. USER INTERFACE

5.1 UI Description

The user interface of this project will be designed to provide a natural and seamless experience of handling AI-driven insights. The frontend, developed using the newest web technologies, will provide a responsive dashboard via which users will be able to visualize data, and handle machine learning models. The most prominent features will be real-time data, predictive analytics screens, and interactive controls for users to change inputs and view results in real time.

Frontend engineers will design a clean and organized UI with an emphasis on usability so that AI/ML outputs are easily understandable. Users will be able to interact with the system via form fields, drop-downs, and graphical displays of AI predictions. The interface will be accessible and responsive, enabling it to be used on a variety of devices. Integration with backend APIs will also enable real-time updates, so the user will always be able to access the freshest data and model insights.

5.2 UI Mockup

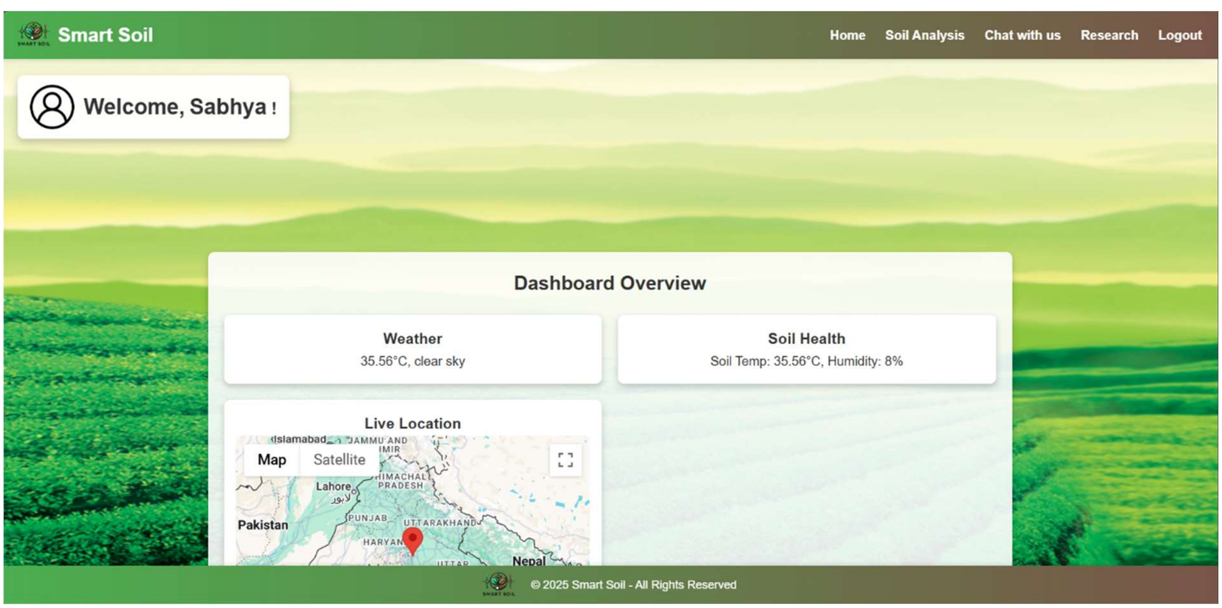


Figure 3: Sample 3

6. ALGORITHMS/PSEUDO CODE OF CORE FUNCTIONALITY

1. HARDWARE LAYER (IoT Devices)

```
class SoilSensor {
   // Sensor Components
   Initialize:
      - Temperature Sensor
      - Moisture Sensor
      - pH Sensor
      - Nutrient Sensors (N, P, K)
      - Light Sensor

   Functions:
      collectSensorData():
         Read all sensor values
         Package data with timestamp and sensor ID
         Return formatted sensor data

      calibrateSensors():
         Perform sensor calibration routines
         Store calibration values

      transmitData():
         Establish connection with gateway
         Send encrypted sensor data
         Receive acknowledgment
}
```

2. DATA COLLECTION AND PROCESSING LAYER

```
class DataProcessor {
   Initialize:
      - Database Connection
      - Data Validation Rules
      - Analysis Models

   Functions:
      processRawData(sensorData):
         Validate data integrity
         Apply calibration corrections
         Store in time-series database

      analyzeSoilHealth():
```

```
            Calculate soil health metrics
            Generate soil quality score
            Identify anomalies
            Return analysis results

        generateRecommendations():
            Compare with optimal values
            Check historical trends
            Generate actionable insights
            Return recommendations
}
```

3. BACKEND API LAYER

```
class SmartSoilAPI {
    Initialize:
        - Authentication System
        - Route Handlers
        - WebSocket Server

    Endpoints:
        // User Management
        POST /api/auth/register
        POST /api/auth/login

        // Sensor Data
        GET /api/sensors/data
        POST /api/sensors/configure

        // Analysis
        GET /api/analysis/soil-health
        GET /api/analysis/recommendations

        // Real-time Updates
        WebSocket /ws/sensor-updates

    Functions:
        handleRealTimeUpdates():
            Listen for new sensor data
            Process and validate data
            Broadcast to subscribed clients
}
```

```
4. FRONTEND APPLICATION
class SmartSoilUI {
   Initialize:
      - User Authentication State
      - Real-time Data Connection
      - Dashboard Components

   Components:
      DashboardView:
         Display real-time sensor readings
         Show soil health indicators
         Present recommendations

      AnalyticsView:
         Display historical data graphs
         Show trend analysis
         Present prediction models

      ConfigurationView:
         Manage sensor settings
         Set alert thresholds
         Configure notification preferences

      NotificationSystem:
         Monitor threshold violations
         Send push notifications
         Display in-app alerts
}


5. MACHINE LEARNING MODULE
class PredictiveAnalytics {
   Initialize:
      - Training Data
      - ML Models

   Functions:
      trainModels():
         Preprocess historical data
         Train prediction models
         Validate model accuracy

      predictFutureConditions():
         Analyze current trends
         Generate predictions
         Return confidence intervals
```

```
  optimizeRecommendations():
      Consider local conditions
      Apply ML insights
      Generate personalized advice
}
```

6. MAIN SYSTEM CONTROLLER

```
class SmartSoilSystem {
  Initialize:
    - All System Components
    - System Configuration
    - Error Handling

  Functions:
    startSystem():
      Initialize all components
      Establish connections
      Start monitoring services

    handleErrors():
      Log error details
      Implement failsafe measures
      Notify administrators

    maintainSystem():
      Monitor system health
      Perform regular backups
      Update configurations
}
```

7. DATA STORAGE

```
Database Schema:
  Users:
    - UserID
    - Authentication details
    - Preferences
    - Farm details

  Analysis:
    - AnalysisID
    - Soil health metrics
    - Recommendations
    - Historical trends

  SensorData:
    - SensorID
    - Timestamp
    - All sensor readings
    - Data quality indicators
```

7. PROJECT CLOSURE

This section elucidates the overall lookup at the project and some of the future works that may enhance the solution.

7.3Goals / Vision

Our initial intention was to create a model whereby users would upload an image of the soil, and the system would provide soil moisture predictions and offer crop advice based on image processing. The design was meant to provide a simple but effective way whereby users would decide on soil condition.

As the project moved forward, we improved our model by incorporating the Google Earth Engine API to obtain real-time location information. We then processed that information into Sentinel-2 satellite imagery spectral bands, which we converted into a CSV dataset for training our machine learning model. This improvement helped us to grow the accuracy and reliability of the soil analysis. The final model now predicts the soil moisture and gives crop advice based on real-time satellite images to offer an improved and scalable solution for farming insights.

8.2 Delivered Solution

Our proposed solution is predominantly made up of an AI system for soil analysis utilizing Google Earth Engine API and Sentinel-2 satellite imagery for predicting soil moisture and suggesting proper crops. The backend is Python-based for AI processing, Flask-based for API integration, and a frontend developed using HTML, CSS, and JavaScript. The system receives real-time geolocation, filters out satellite data into spectral bands, converts the data into a CSV dataset to train the model, and offers accurate soil moisture prediction and crop suggestions. The optimized final model was trained using machine learning algorithms like XGBoost and Random Forest for improved accuracy.

Week 1: Project Idea & Research

We completed the project concept of soil detection with the help of Sentinel-1 and Sentinel-2 satellite imagery and an AI model. We started our research by learning Google Earth Engine API, Sentinel bands, and their applications in soil and crop prediction. We also studied competing products already available in the market and determined our differentiator.

Week 2: Literature Review & Technology Stack

We chose Python for AI processing, Flask for API integration, and HTML, CSS, and JavaScript for the frontend. We studied research papers on soil moisture prediction, pre-processing of Sentinel-2, and AI models like XGBoost, Random Forest, and TensorFlow. We also studied Google Maps and Google Earth Engine APIs for implementation.

Week 3: Planning & Frontend Development Basics

We designed activity selection diagrams, data flow diagrams, and pseudocode for major functions. Frontend development began with login and registration pages and OTP authentication. Initial UI design for the dashboard, research area, and map interface was developed to give a simple user interface.

Week 4: Feature Design & Prototyping

Wireframes and prototypes of the most important features were created, including real-time location display, integration of weather conditions, and an AI-driven soil and crop prediction dashboard. The backend data processing architecture, API request handling, and the user authentication was set up. Google Maps API was used for retrieving real-time geolocation.

Week 5: Prototype Testing and Refinements

Prototypes were tested for usability and user experience. Performance and responsiveness of UI were enhanced. OTP verification and login were optimized, and API calls were optimized for fetching accurate geolocation and weather. Security was improved to avoid unauthorized access.

Week 6: Data Preprocessing & Collection

Satellite data was sourced from Google Earth Engine, the Sentinel-2 bands applicable in crop forecasting and soil moisture forecasting. Geolocation mapping and data preprocessing were accomplished, and satellite features extracted were formatted into a CSV dataset used for training the model.

Week 7: Backend Integration, Feature Engineering & EDA Exploratory Data Analysis (EDA)

EDA was conducted to graph crop types and patterns of soil moisture. Feature engineering techniques were employed in the preprocessing of input variables. Backend integration was started, including post-login rendering, Razorpay integration, Google Earth Engine Sentinel-2 API integration, and research tab simulation for scientific paper demonstrations.

Week 8: Google Maps API Integration, Model Training, Evaluation & Optimization

Some machine learning algorithms such as XGBoost and Random Forest were tested and contrasted with one another based on soil moisture prediction accuracy. Model performance was optimized by tuning the hyperparameters. The finished AI model was then implemented in the backend to send predictions efficiently in accordance with the geolocation of the user. Final testing and performance evaluation were conducted to identify accuracy and performance.

8.3 Remaining Work

Our project has succeeded in integrating an AI-enabled soil analysis system with real-time geolocation-based crop and soil moisture forecasting. Further fine-tuning and tweaking are still necessary to improve its accuracy, ease of use, and scalability. The remaining work of the project is described by the next steps as follows:

Week 9: Model Testing & Accuracy Improvement

We will experiment with various AI models to determine how well they perform in soil moisture forecasting and crop suitability. We will carry out hyperparameter tuning for accuracy and trustworthiness. We will experiment with various performance metrics in a bid to identify the optimal model to employ.

Week 10: Deployment of Model & Integration into Website

The trained AI model will be hosted on a cloud server to provide easy access. The model will be integrated into our website, and the customers will be able to input their geolocation to receive real-time soil and crop guidance. API endpoints will be used to enable seamless communication between the frontend and the backend.

Week 11: API Test, Debug and Experimental Features

API integrations will be thoroughly tested for smooth data fetching from Google Earth Engine and proper processing. System debugging will be conducted for eliminating system bugs. Other experimental features will be tried out, such as integration with an AI-driven agricultural advisory chatbot, model translation to FastAPI for enhanced response time, and integration with a marketplace for farm produce.

Week 12: Last Bug Fixes & Site Optimization

Final debugging and bug fixing will be performed to provide a smooth user experience. The site will be optimized for performance, speed, and usability. Load testing will be performed to test the system's performance under heavy traffic to see if it can efficiently handle traffic. Security features will be tightened to provide data privacy and security before final release.

REFERENCES

In the face of rapid climate change and increasing food security concerns, precision agriculture has emerged as a vital area of technological innovation. At the heart of this transformation lies the need for accurate soil analysis—specifically, the prediction of soil moisture levels and crop recommendation using satellite data. My project, titled Smart Soil, leverages Sentinel-2 multispectral imagery, SMAP soil moisture data, and machine learning to create a decision-support system for farmers. This section outlines the literature and existing research that shaped this idea.

Remote Sensing and Soil Moisture Prediction

One of the most extensively studied aspects in agritech is the use of Synthetic Aperture Radar (SAR) and optical satellite data for soil moisture estimation. According to the paper "Are the current expectations for SAR remote sensing of soil moisture using machine learning over-optimistic?" (2022), researchers argue that while SAR datasets have shown promise, machine learning models trained on them often underperform in real-world settings due to spatial heterogeneity and data noise. This finding aligns with our choice to utilize Sentinel-2 optical bands (B2, B3, B4, B8) and SMAP passive microwave data, which are less impacted by such limitations and offer broader spatio-temporal coverage with a more interpretable structure.

The paper further discusses that the generalization ability of ML models across regions is a major challenge, with most models showing strong performance only in the regions they were trained on. This motivated me to construct a diverse dataset covering Kharif and Rabi seasons, using random geospatial sampling across India to capture the country's vast agricultural diversity. By training models on this heterogeneous dataset, the goal is to enhance model robustness and reduce overfitting.

Integration of Machine Learning in Agriculture

There has been a growing body of work on applying supervised learning for both regression (e.g., predicting soil moisture) and classification (e.g., crop recommendation). Studies have shown that models like Random Forest, XGBoost, and Artificial Neural Networks (ANNs) provide significant accuracy improvements over traditional statistical methods. Inspired by this, I first experimented with traditional ML models and then transitioned to ANN-based approaches for better generalization, especially after evaluating performance using MAE, RMSE, and accuracy metrics.

Moreover, papers like "Machine learning for soil moisture retrieval: Challenges and future prospects" highlight the importance of feature engineering and incorporating multi-source data for better soil condition predictions. This directly influenced my approach of combining Sentinel-2 reflectance values with SMAP data and rule-based logic to define both soil moisture labels and recommended crops based on seasonal context and known crop-soil suitability zones.

Motivation and Project Direction

Upon reviewing multiple papers and datasets from platforms like NASA EarthData, Copernicus Open Access Hub, and Kaggle, I identified a research gap: most soil prediction systems focus solely on classification or regression—not both. Therefore, Smart Soil was designed to be a multi-output system, offering insights into both soil moisture content (regression) and suitable crop type (classification) based on the same set of inputs.

Additionally, the temporal frequency and free accessibility of Sentinel-2 data made it the ideal choice. While SAR provides data under cloud cover, its preprocessing is more complex and less intuitive for integration into lightweight ML pipelines, especially for real-time deployment. By contrast, Sentinel-2 images offer faster integration with tools like Google Earth Engine (GEE), which was used to export over 25,000 georeferenced data points per season.

Finally, inspiration was drawn from open-source agritech solutions such as Plantix, Farmonaut, and OneSoil, which demonstrate the power of AI-backed insights but often remain closed in terms of their model architecture. Smart Soil's objective is not only to match these tools in accuracy but also to remain transparent, reproducible, and open for further research or integration into mobile/web platforms.

Link of the research Paper :- https://ieeexplore.ieee.org/abstract/document/10852337