

BhoomiAI: Technical Documentation & Project Report

Project: BhoomiAI - An AI-Powered Crop Yield Prediction and Optimization Platform

Team: GreenBots

Version: 1.0 (Smart India Hackathon 2025 Submission)

1. Project Overview

1.1 Introduction

BhoomiAI is a full-stack web application developed for the Smart India Hackathon 2025 under the theme of **Agriculture, FoodTech & Rural Development**. It provides small-scale farmers with data-driven insights for crop yield prediction, using a custom-trained machine learning model, making precision agriculture accessible and simple.

1.2 Problem Statement

Small-scale farmers in India face uncertainty due to unpredictable weather, soil variability, and non-optimized fertilizer usage. Lack of affordable and easy-to-use predictive tools leads to suboptimal yields and financial instability.

1.3 Proposed Solution

BhoomiAI offers:

- Data-Driven Prediction:** A Random Forest ML model trained on Odisha's historical data predicts crop yields in kg/ha using features like fertilizer usage, cropping intensity, area, and season.
- Actionable Insights:** Integrates live weather data to reduce manual input and improve prediction accuracy.

2. System Architecture

BhoomiAI uses a decoupled three-tier architecture:

- **Frontend:** Static web app (HTML, CSS, JavaScript)
- **Backend:** Spring Boot REST API (Java, PostgreSQL)
- **ML Service:** Python FastAPI microservice for AI computations

2.1 Architectural Flow

Flow 1: User & Weather

Frontend !' Backend !' PostgreSQL !' External Weather API

Flow 2: AI Prediction

Frontend !' ML Service !' Returns predicted yield

2.2 Technology Stack

Frontend:

- HTML5, CSS3, JavaScript (ES6+)
- Axios for API calls, GSAP for animations

Backend:

- Spring Boot 3+, Java 21
- Spring Web, Spring Data JPA, Hibernate, PostgreSQL

Machine Learning Service:

- Python 3.11+, FastAPI
- scikit-learn, pandas, joblib, uvicorn, pydantic

3. Machine Learning Pipeline

3.1 Data Sourcing & Pre-processing

Datasets sourced from Odisha:

- Seasonal Rice Yield (multi-level CSV)
- State-wide Fertilizer Consumption (N, P, K)
- Land Usage Statistics (cropping intensity)
- Cleaned seasonal yield dataset

Pre-processing:

- Standardize Year column, rename columns, merge datasets
- Output: `odishamasterdata_final.csv`

3.2 Model Training & Evaluation (`train_master.py`)

- One-hot encode `Season` column
- Train Random Forest Regressor (80% train, 20% test)
- Performance: R^2 "H0.81, MAE "Hl 43 kg/ha
- Serialize model: `bhoomia/final_model.joblib`

3.3 API Deployment (`main.py`)

- FastAPI POST endpoint `/predict`
- Validates input using pydantic
- One-hot encodes Season and predicts yield
- Returns JSON response

4. Backend Service (Spring Boot)

Core Responsibilities:

- **User Management:** ``/api/v1/users/identify``
- **External Weather API Integration:** ``/api/v1/weather/fetch-current``
- **Data Persistence:** Spring Data JPA + Hibernate
- **Global CORS Configuration**

5. Frontend Application (HTML, CSS, JS)

5.1 Structure & User Flow

- **Pages:** ``index.html`` (landing), ``login.html``, ``dashboard.html``
- **State Management:** Uses ``localStorage`` to transfer user data

5.2 Styling & Animation

- **Custom CSS:** Unique styles for each page
- **Design:** Vibrant greens, blues, glassmorphism effects
- **GSAP:** Scroll animations, transitions, countdown effects

5.3 Functionality & API Integration (``dashboard.js``)

- Event-driven user interactions
- Async API calls via Axios
- Updates UI dynamically based on API responses

6. Conclusion & Future Scope

BhoomiAI demonstrates a complete AI-powered agricultural tool for small-scale farmers.

Future Enhancements:

- Models for other crops (Wheat, Maize)
- District-level historical weather integration
- N-P-K recommendations module
- Progressive Web App for offline and mobile use