AI-Based Crop Recommendation for Farmers

Comprehensive Hackathon Solution Plan

Problem ID: 25030

Hackathon Duration: 36 hours (Late 2025)

1. How to Solve: Approach, Methodology, or Solution Strategy

1.1. Core Idea & Innovation

The core idea is to develop an AI-driven mobile platform that provides hyper-localized crop recommendations for farmers by integrating soil data, weather forecasts, past crop rotations, and market trends.

- AI-based predictive models
- Image recognition for crop and soil health
- Voice/chat interfaces in regional languages

This allows farmers to receive real-time, actionable, offline-friendly advice, making the solution personalized, sustainable, and accessible to low-connectivity rural regions.

1.2. Technology Stack Recommendation

Component	Recommendation
Frontend	Mobile App: Flutter (cross-platform, supports offline and multilingual UI), React Native as alternative
Backend	FastAPI (Python) for rapid API development, integrates easily with ML models
Database	PostgreSQL (structured farmer, soil, and crop data), SQLite or local storage for offline caching
AI/ML Models	TensorFlow Lite (on-device inference), PyTorch (server-side ML), Scikit-learn (soil property prediction), Hugging Face Transformers (local language NLP)
APIs & Data Sources	SoilGrids API, Bhuvan APIs, OpenWeatherMap API, IMD API, Agri-market APIs, web scraping
Deployment	Backend: AWS EC2/Elastic Beanstalk or GCP Cloud Run; Frontend: App stores or Firebase Hosting

1.3. Step-by-Step Execution Plan

Phase 1: Foundation (First 8 hours)

- Finalize features: crop recommendation, soil & weather input, market trend suggestions, multilingual support, offline mode
- UI/UX mockups: farmer-friendly interface with large buttons and visual indicators
- Project setup: Initialize Flutter project + Git repo, Setup FastAPI backend, Integrate PostgreSQL or SQLite
- Dataset exploration: Download soil, weather, and crop datasets; clean and pre-process for ML model training

Phase 2: Core Module Development (Next 16 hours)

- Soil & Weather Analysis: Build lightweight regression model for soil suitability prediction
- Crop Recommendation Engine: Train ML model using historical crop yield, soil type, weather, and market data
- Image Recognition: TensorFlow Lite model for crop/soil disease detection with camera integration
- Multilingual NLP: Hugging Face models for voice/chat interface in Hindi and regional languages
- APIs integration: Connect soil, weather, and market APIs for real-time data

Phase 3: Integration & Polish (Next 8 hours)

- Connect frontend and backend: Fetch recommendations and send user input to ML engine
- Offline Mode: Implement local caching for key recommendations & FAQs
- UI polish: Visualize soil health, predicted yield, and profit clearly with icons and infographics
- Testing: Basic end-to-end flow with sample farmers; bug fixing
- Presentation prep: Prepare screenshots, demo video, and feature explanation slides

Phase 4: Buffer & Pitch Prep (Final 4 hours)

- Run final tests with multiple data inputs
- Fine-tune AI model thresholds for accuracy

- Rehearse live demo: image input, voice query, recommendation output
- Prepare slides highlighting impact, innovation, and scalability

2. Available Resources: Datasets, APIs, Tools, Platforms, or References

2.1. Datasets

- SoilGrids: Global soil properties (pH, organic carbon, texture) → soilgrids.org
- Agri-market data (India): Crop prices → agmarknet.gov.in
- Weather data: Historical and forecast data → OpenWeatherMap API
- PlantVillage dataset: Crop disease images → Kaggle

2.2. APIs & Tools

- Bhuvan APIs (ISRO satellite data)
- OpenWeatherMap API (weather forecasts)
- Razorpay API (optional for seed purchases)
- Tesseract OCR (optional, for handwritten farm records)
- BeautifulSoup/Scrapy for market trend scraping

2.3. Platforms & Frameworks

- Flutter for mobile app development
- FastAPI for backend API
- PostgreSQL + SQLite for database
- TensorFlow Lite for on-device ML
- PyTorch for server-side ML
- Hugging Face Transformers for local language NLP
- Google Colab / Kaggle Notebook for ML prototyping

2.4. Key References

- Research Paper: "AI for Precision Agriculture: Crop Recommendation Systems" → ScienceDirect
- GitHub Example: "Plant Disease Detection using TensorFlow Lite" → GitHub
- Tutorial: "Build a Crop Recommendation System using Python and ML" → Kaggle

Summary: This plan enables your team to prototype a working mobile AI system for crop recommendation within 36 hours, featuring multilingual

support, offline capability, real-time soil & weather integration, and actionable farmer-friendly insights. It balances innovation, technical feasibility, and real-world impact for rural India.