edunet foundation

Smart Irrigation System for Efficient Agriculture

Linkedin: PRAKHAR SHARMA

Github link: github project link

Name:PRAKHAR SHARMA
Mentor Name:Raghunandan M

Learning Objectives:

GOAL: To engineer a comprehensive, professional-grade Smart Irrigation Dashboard that not only automates watering but also empowers farmers with actionable, real-time insights through live weather integration, system health monitoring, and multi-modal control.

- To gain hands-on experience in data preprocessing and EDA of sensor data.
- To build and train a multi-output classification model using Python's scikit-learn library.
- To deploy a machine learning model into a feature-rich, interactive web application using Streamlit.
- **To integrate third-party APIs for live data feeds and implement security best practices using secrets management.**

Source : www.freepik.com/

Tools and Technology use

- ***Python Libraries:***

I. Pandas: For efficient data manipulation and analysis.
II. Scikit-learn: For machine learning functionalities and loading the model with Joblib.
III. Matplotlib & Seaborn: For initial data visualizations and plots.
IV. Requests: For making HTTP requests to external APIs.
V. Plotly: For creating interactive charts in the final application.

- ***Streamlit:***

I. For building the interactive web application and dashboard.

- ***IDE/Environment:***

I. Google Colab: Used for developing, training, and evaluating the machine learning model, and for generating the .pkl file.
II. Local Terminal/Command Prompt: Used to run the Streamlit web application on a local machine.

- ***APIs & Services:***

I. OpenWeatherMap API: To fetch live weather data for intelligent decision-making.
II. GitHub: Used for hosting the project's source code and managing different versions.

Methodology

The project is built upon a :

- MultiOutputClassifier wrapping a RandomForestClassifier.
- This model is trained on sensor data to predict the ON/OFF status for three farm parcels.
- The model is saved as a
- .pkl file for deployment.

Application Architecture:

- A sophisticated multi-page application was developed using Streamlit's tabbed layout for a clean user experience (Dashboard, Analytics, System Health, Settings).
- Application state (like the current irrigation mode) is managed across user interactions using st.session_state.
- API keys and sensitive data are securely managed using Streamlit's secrets management, preventing exposure in the source code.

Methodology

Feature Integration:

- **Live Weather:** Integrates the OpenWeatherMap API to fetch and display real-time temperature, humidity, and rain forecasts, providing crucial context.
- **Multi-Modal Control**: Implemented logic for three distinct operational modes:
- **Automatic:** AI-driven predictions.
- **Manual:** Direct user control with override toggles.
- **Scheduled:** Time-based irrigation set by the user.
- **System Diagnostics:** A sensor health algorithm checks for abnormal readings (e.g., values at 0.0 or 1.0) and flags them for user attention.

Problem Statement:

Overview:

The Challenge: Modern farming requires more than simple automation. Farmers face a complex set of challenges including water scarcity, unpredictable weather, and a lack of trust in "black box" automated systems. There is a critical need for an intelligent system that is not only efficient but also transparent, controllable, and context-aware.

Problem Statement:

To address this gap by developing an integrated dashboard that provides intelligent AI suggestions, full manual control, and vital real-time data (like weather and sensor health), fostering trust and empowering the farmer to make optimal decisions.

Solution:

The "Agri-Smart Pro" Dashboard: A comprehensive, multi-faceted solution was developed to tackle the challenges of modern irrigation.

Intelligent Dashboard: Displays AI-powered predictions alongside live weather metrics, offering a holistic view for decision-making.

Total User Control: Features three operational modes (AI, Manual, Scheduled) and allows for immediate manual override of individual sprinklers.

Deep Analytics: Provides interactive charts for current sensor data, historical water usage trends, and a detailed event log for auditing every action.

System Diagnostics: A dedicated "System Health" tab actively monitors all 20 sensors and flags potential malfunctions, enabling proactive maintenance.

Customization & Security: Includes tailored advice based on selected crop profiles and ensures secure API key handling via Streamlit Secrets.

GitHub Link: Click here for the github link.

API KEY: 02073b5867c5940938ee3b84954e47d7

Click here if the code does not visible.It has drive link for every file related to this internship.

Screenshot of Output:

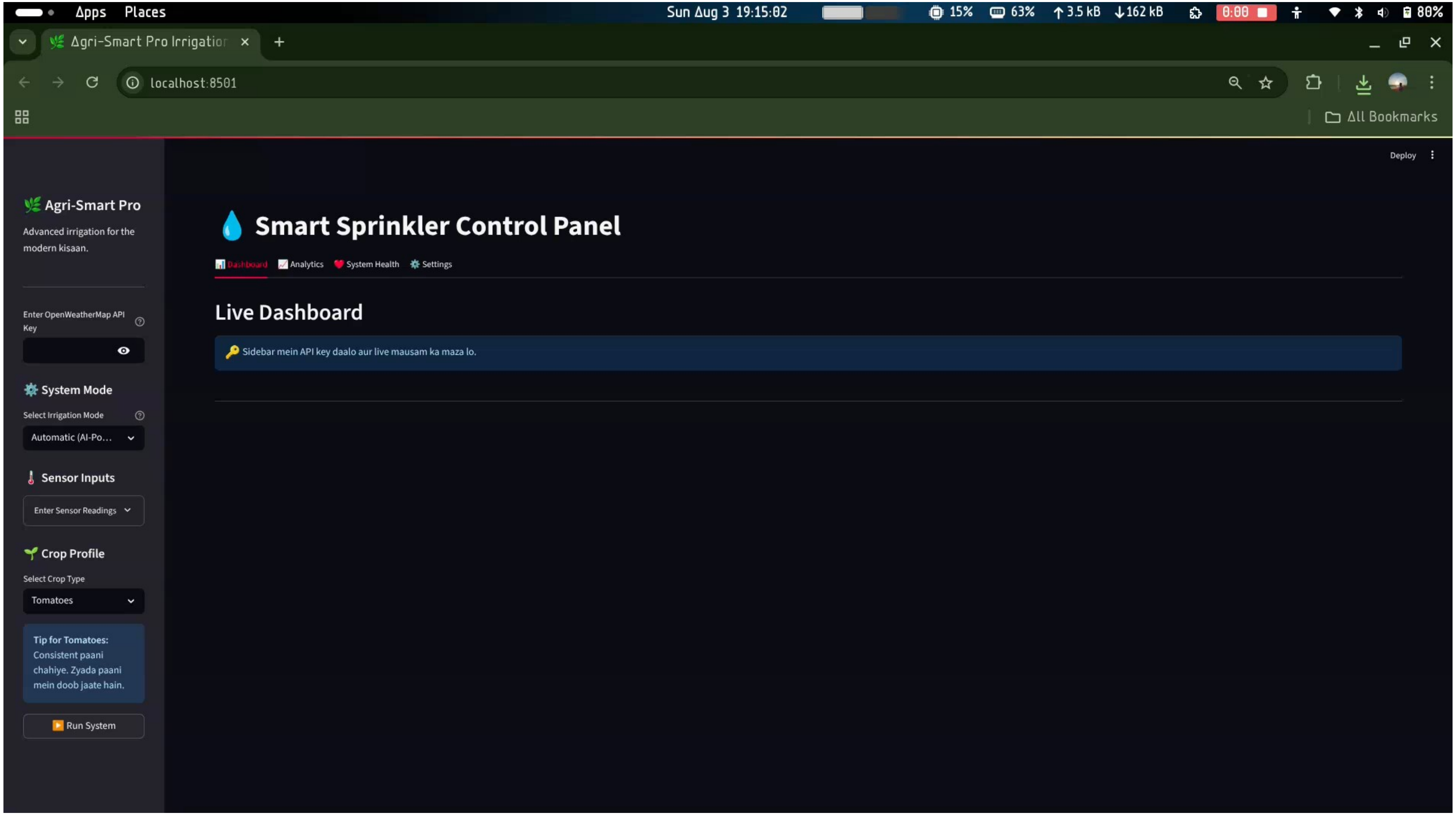
Local URL: http://localhost:8501 Network URL: http://192.168.1.40:8501

This is a video
Of the output.

I had to add it
because my
project had
multiple features.

Pl note that the
API key is:

02073b5867c5940938ee3b84954
e47d7



CONCLUSION: This project successfully evolved from a simple proof-of-concept into a robust prototype for a comprehensive farm management dashboard. By integrating a machine learning model with live APIs, advanced user controls, and system diagnostics, it demonstrates a clear path toward building practical, trustworthy, and highly efficient agricultural technology.

Future Scope:

Predictive Maintenance: Utilize historical sensor data to train a model that predicts sensor failures before they occur.

Dynamic Cost-Benefit Analysis: Enhance the savings calculator to use real-time water prices and historical usage data for more accurate financial insights.

Automated Alerting System: Integrate email or SMS notifications to alert farmers of critical events, such as system malfunctions or irrigation cycles skipped due to rain.

Advanced Soil-Moisture Models: Incorporate more complex models that factor in soil type, evaporation rates, and crop growth stage for even finer control over water usage.