TECHFIESTA 2025
Precision Irrigation System

- **Problem Statement ID – T2K25A2**
- **Problem Statement Title- Precision Irrigation System**
- **Domain - Agriculture**
- **Team Name - Tech Titans**
- **Team Leader Name - Soham Penshanwar**

Idea/Approach Details

Proposed Solution

- **Smart Irrigation**: Combines soil moisture, weather data, and crop stages to optimize water use.
- **Automated Scheduling**: Raspberry Pi processes data and adjusts irrigation schedules automatically.
- **Weather Adaptation**: Integrates forecasts to adjust irrigation based on predicted rainfall.
- **User Interface**: Local and remote dashboards for easy monitoring and control.

.

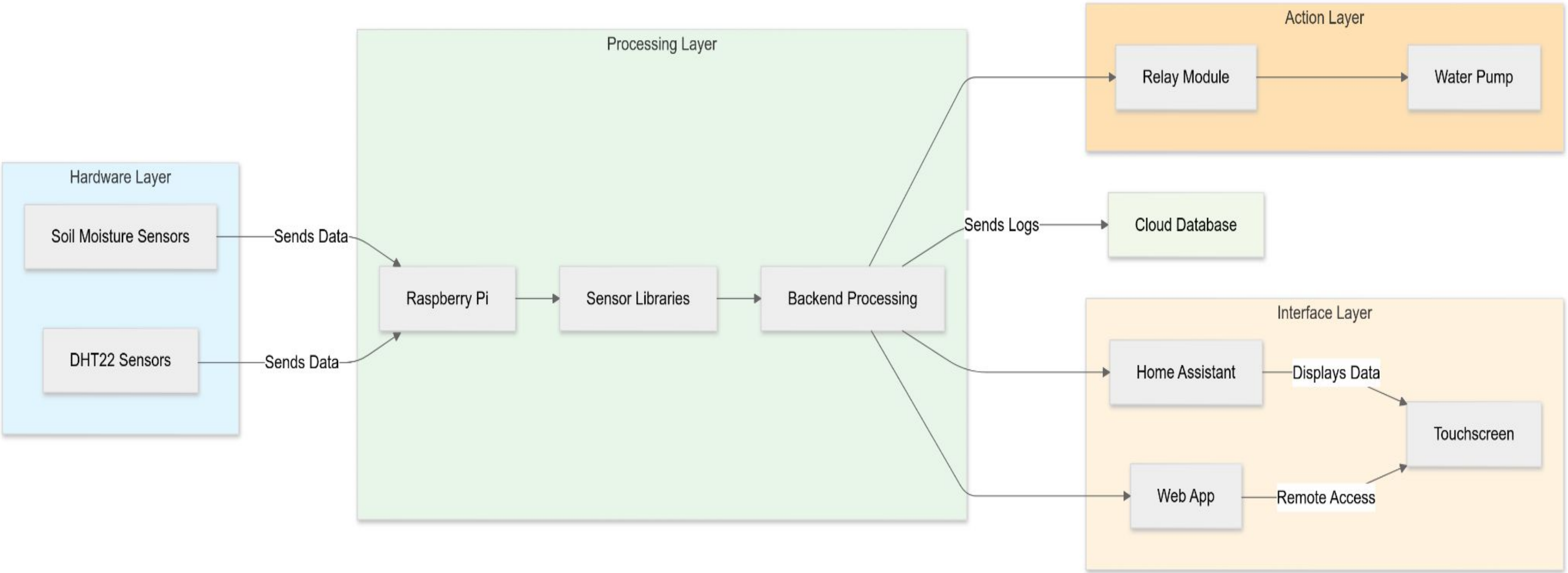
Addressing the Problem

- **Efficient Water Management**: Minimizes water waste by irrigating only when necessary.
- **Weather Adaptability**: Prevents over-irrigation by anticipating rainfall.
- **Crop-Specific Irrigation**: Customizes irrigation schedules for different crops and growth stages.
- **Automated Precision**: Reduces human error by automating irrigation decisions.

Innovation & Uniqueness

- **Real-Time Data**: Collects and processes weather, soil, and crop data in real-time.
- **Remote Control**: Enables monitoring and management via mobile/web dashboard.
- **Weather Integration**: Adjusts irrigation based on weather forecasts.
- **Voice Assistance**: Provides hands-free control and updates.

Process flow diagram



Methodology used

1.Data Collection

Provides real-time environmental data for irrigation decisions.
Sensors: Soil Moisture Sensor, DHT22

2.Data Acquisition

Raspberry Pi collects and processes sensor data, ensuring accuracy.

3.Data Analysis

Compares soil moisture & environmental conditions to determine irrigation needs.

4.Decision Making

Evaluates data against thresholds, calculates optimal irrigation timing.

5.Hardware Integration

Raspberry Pi controls Relay Module to activate the Water Pump

6.User Interface

Web dashboard displays real-time data and allows system control.

Solution Concept and Feasibility

Concept	Feasibility
Smart Irrigation System	Built on **Raspberry Pi** and **low-cost sensors**, ensuring **affordability** and **scalability**.
Automated Decision-Making	**Simple automation** reduces labor and integrates seamlessly into existing systems.
Voice-Controlled Operation	**Cost-effective** integration with voice assistants makes it intuitive for farmers with minimal technical skills.
Seamless Monitoring	Accessible via **mobile apps** or **desktop dashboards**, enabling flexibility and remote control.
Environmental & Economic Impact	Achieves **long-term savings** in water and energy, making it **financially viable** and eco-friendly.

Use cases & description

Use Case 1: Real-Time Soil Monitoring

- **Description:**
  - Continuously monitors soil moisture levels and environmental parameters like temperature and humidity.
  - Provides accurate, real-time data to optimize irrigation schedules.
- **Benefit:**
  - Prevents over-irrigation or under-irrigation, ensuring better crop health.

Use Case 2: Remote Monitoring and Control

- **Description:**
  - Provides a web or mobile dashboard for real-time monitoring of soil conditions.
  - Allows users to manually control the irrigation system remotely if needed.
- **Benefit:**
  - Increases convenience and ensures system usability from anywhere.

Use Case 3: Automated Irrigation Control

- **Description:**
  - Automatically activates and deactivates the water pump based on soil moisture thresholds.
  - Tailors irrigation to specific crop requirements and growth stages.
- **Benefit:**
  - Reduces manual effort and improves irrigation accuracy.

Technology stack used

Category	Technology/Tool	Purpose
Hardware	Raspberry Pi	Central controller for the system.
	Soil Moisture Sensor	Measures real-time soil moisture levels.
	DHT22 Sensor	Captures temperature and humidity.
	Relay Module	Controls the water pump.
	Water Pump	Executes irrigation based on control signals.
Programming	Python	Core logic, sensor interfacing, and data processing.
Backend	Django	REST API for processing sensor data and managing logic.
Frontend	HTML/CSS/JavaScript	Web dashboard for monitoring and control.
	React.js or Vue.js (Optional)	For modern and dynamic UI/UX.
Data Visualization	Matplotlib/Plotly	Visualizes sensor data in graphs or charts.
Local Database	SQLite	Logs sensor data locally on the Raspberry Pi.
	MySQL/PostgreSQL (Optional)	For advanced data handling and storage.
Cloud Services	Google Cloud / AWS IoT (Optional)	Remote data storage, analysis, and real-time monitoring.
	Firebase Realtime Database (Optional)	Simplified cloud storage and synchronization.
Weather API	OpenWeatherMap or WeatherAPI	Fetches real-time weather data for irrigation planning.
IoT Protocol	MQTT (Optional, if ESP32 is used)	Lightweight IoT communication protocol.
	HTTP/WebSocket	Communication between Raspberry Pi and cloud/local interfaces.

Constraints

Challenge	Strategy
Data Accuracy and Sensor Reliability	- Use high-quality, calibrated sensors. - Regular maintenance and calibration. - Implement data validation algorithms.
Integration with Existing Systems	- Design a modular and compatible system. - Provide retrofit kits for older infrastructure. - Offer custom integration.
Technical Skills of Farmers	- Create user-friendly interfaces (mobile/web). - Provide training programs and manuals. - Offer 24/7 customer support.
Dependence on Internet Connectivity	- Implement offline functionality for core operations. - Use SMS alerts and local notifications.