Project Proposal

Request for Proposal (RFP) - Farm Management Dashboard System

Name: Hai Hoang Le

ID: 103542974

Classroom: SAT 7:00 Wagga Wagga (6th floor)

Lecturer: Dr. Thomas Hoang

Background

GCFood Company is a leading agricultural enterprise in Vietnam, and the business is looking for offers from competent vendors to provide a Farm Management Dashboard System. To reach this goal, the company needs a Farm Management Dashboard System, which will enable their farmers to better monitor and manage their operations. The solution will allow them to use IoT sensor data and AI models to make more informed decisions and maximize resource utilization.

Scope

Objective:

The primary objective of this project is to develop and implement a Farm Management Dashboard System that will streamline farm operations, provide real-time insights, and enhance productivity for farmers. The system aims to achieve the following objectives:

- Centralized Data Aggregation: The system will collect data from numerous IoT sensors and, an easy-to-use dashboard, providing a comprehensive perspective of farm operations.
- Farmers will get real-time and historical insights into crop health, soil conditions, weather forecasts, and equipment performance, allowing them to make more educated decisions.
- Decision Support: Based on data analysis, the system will deliver intelligent recommendations and insights to farmers to help them make more informed decisions about crop management, irrigation scheduling, and resource allocation.
- Efficiency and Productivity: By streamlining procedures and minimizing waste, the system attempts to improve agricultural efficiency and overall productivity.

Functions and Features Requirements:

To achieve the project objectives, the Farm Management Dashboard System should include the following functions and features:

1. Data Integration & Aggregation:
    a. Integration of IoT sensors to monitor elements like as temperature, humidity, soil moisture, and crop health.
    b. AI models will be used to detect pests, identify crop diseases, and predict yield.
    c. Implementation of data validation and cleansing procedures to ensure data accuracy and reliability.
2. User-Friendly Dashboard:
    a. Customizable dashboards allow farmers to tailor their views to their own preferences and needs.
    b. Real-time data visualization using interactive charts, graphs, and maps to aid comprehension.
    c. Historical data is stored for trend analysis and decision-making purposes.

3. Alerts and notifications:
    a. Implementation of an alerting system with customized mechanisms for informing farmers of crucial occurrences or abnormalities.
    b. Mobile accessibility allows farmers to receive warnings and access the dashboard via cellphones and tablets.
4. Decision Support and Insights:
    a. AI-powered recommendations for crop management, irrigation timing, and resource allocation.
    b. Weather forecasts are being integrated to assist farmers plan their activities and adjust to changing weather conditions.
5. User Management and Security:
    a. Role-based access control ensures data privacy and security.
    b. Sensitive data is secured using robust encryption mechanisms.
6. Reporting and analysis:
    a. Reporting tools for creating custom reports and exporting data.
    b. Data analytics skills include trend analysis and performance evaluation.

Project Timeline:

Phase 1: Implement an IOT system (2 weeks)

Phase 2: Design a database and design structure (1 week)

Phase 3: Build API system for CRUD (1 week)

Phase 4: Implement a website with a dashboard with requirements (4 weeks)

Phase 5: Testing and Quality Assurance (2 weeks)

Phase 6: User Acceptance Testing(1 week)

Phase 7: Deployment and Training (1 week)

Total: 12 weeks (3 months)

Technical Specification:

- Embedded language, IOT sensors
- Database system design
- Backend Development
- Web development and Framework
- Server management and deployment
- Web security, business data, and IoT data security

Risk and Mitigation:

- Data leak
- Server downtime, server load capacity
- User information leak
- Database overload

Deliverables and schedule

During this project as an individual project, the project's deliverables will include a fully functional prototype of a software application that includes most of the required functionalities listed in the Scope section above. The

prototype will show how all the IOT sensors are connected to a microcontroller as well as the way that all IOT data is sent and stored on a free noSQL database. A Vuejs framework will be used to implement the user interface in this case the dashboard, the library Chartjs will be used to show the real-time IOT data on the interface. The backend system will include an API system so developers can interact with the data in the database by using the fetching method. The chatGPT 3.5 is realized it is an alternative to be integrated into the application. Users can take advantage of it to generate reports of IOT data over some time. In addition, it can give out some recommendations based on the IOT data sent from the user.

Initial Release Schedule of the Product Backlog items

No.	Item	Dependencies	Business Value (1 least – 10 most)	Release Schedule (Sprint 1 | 2 | 3 | …)
F1	Thinking about which attributes are needed for farm management		8	Sprint 1
F2	List all components and sensors	F1	8	Sprint 1
F3	Collect and prepare sensors and components	F2	9	Sprint 1
F4	Design the model for the farm		7	Sprint 1
F5	Prepare to set up sensors and components together	F3	7	Sprint 1
F6	Wire up all sensors to a micro-controller	F5	8	Sprint 1
F7	Start to collect to see how sensors return value	F6	5	Sprint 1
F8	Structure the IOT data in a structured way		8	Sprint 1
F9	Send all Data to a node in this case an ESP8266 to store in a database via TX TR port	F8	8	Sprint 2
F10	Choose a database to store these Data in real-time		9	Sprint 2
F11	Start to reading the provided document about the database		6	Sprint 2
F12	Design the database to store IOT data in a structured way	F10	8	Sprint 2
F13	Start to connect to the database on ESP8266		7	Sprint 2
F14	Send all IOT data to store on the database		7	Sprint 2
F15	Choose a way to interact with IOT data in the database like API		8	Sprint 3
F16	Choose a service to build the API backend system	F15	8	Sprint 3
F17	Select a language to write like NodeJS	F16	8	Sprint 3
F18	Start to code APIs to CRUD IOT data		9	Sprint 3
F19	Fix and Testing APIs	F18	9	Sprint 3
F20	Design a wireframe for a dashboard UI		9	Sprint 4
F21	Choose a web framework that supports creating one		8	Sprint 4
F22	Think about which technology will be used	F21	8	Sprint 4
F23	Create an authentication and authorization system		9	Sprint 4
F24	Create a user-friendly dashboard		9	Sprint 5
F25	Create real-time data charts fetched from the API		8	Sprint 5
F26	Integrated AI bot to the dashboard		8	Sprint 5

F27	Create a feature that user can send IOT data to AI bot to get reports or recommendation	F26	9	Sprint 5
F28	Create some widgets for users to directly control sensors		8	Sprint 5
F29	Create a notification, alert system for farmers		9	Sprint 5
F30	Quality, security, currency testing	F23 ,F24 ,F25, F27, F28, F29	8	Sprint 5
F31	Hire a domain name and a host server		7	Sprint 6
F32	Setup server for web services, attach the domain name to the server	F31	7	Sprint 6
F33	Secure the server for against cyber attack	F32	8	Sprint 6
F34	Upload the website to the server for global use	F33	9	Sprint 6
F35	Write an instruction paper manually use		7	Sprint 6