Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	03 October 2022
Team ID	PNT2022TMID47947
Project Name	SmartFarmer - IoT Enabled Smart Farming Application
Maximum Marks	4 Marks

Technical Architecture:

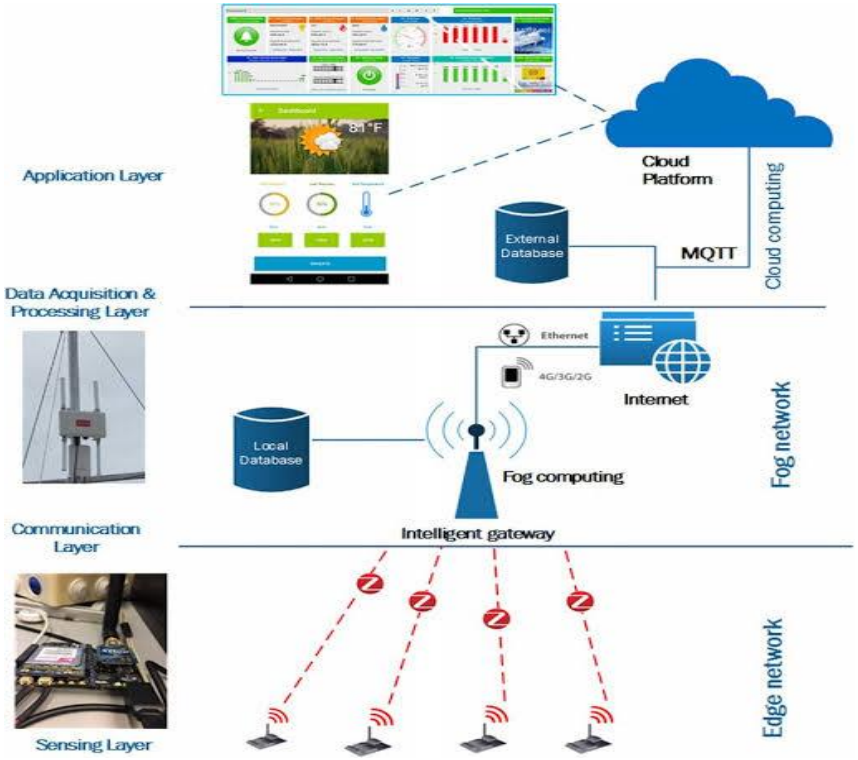

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Mobile App Front End Client side code, which is downloaded and executed by the browser. Back End Server side code, which is executed on the server.	HTML, CSS, JavaScript / Angular Js / React Js Node Js / PHP / Java / Python
2.	Application Logic-1	Collect sensor data from various field sensors and save to a remote database	Java / Python
3.	Application Logic-2	Provide information to farmers about the current conditions of their crops	IBM Watson STT service
4.	Application Logic-3	Help farmers in making decisions about irrigation, fertilization, and pest control	IBM Watson Assistant
5.	Database	Database Persistent storage to store the data The database should be able to store data relating to soil moisture, temperature, humidity, light intensity, and water level.	MySQL, NoSQL, etc.
6.	Cloud Database	MongoDB - Used for storing data	IBM DB2, IBM Cloudant etc.
7.	File Storage	The app also includes a database of crop information, so farmers can quickly look up the best practices for growing their crops	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	The purpose of this API is to provide a means for developers to interact with the Smartfarmer application in order to automate various tasks related to smart farming	IBM Weather API, etc.
9.	External API-2	This includes retrieving data from sensors, controlling actuators, and managing user account	Aadhar API, etc.
10.	Machine Learning Model	The purpose of machine learning in the Smartfarmer application is to provide farmers with predictions about crop yields, based on data collected by sensors in the field. This information can help farmers to make decisions about when to plant, how to irrigate, and what type of fertilizer to use.	Object Recognition Model, etc.

11.	Infrastructure (Server / Cloud)	Local Server Configuration: Prerequisites :- You need to have java installed on your system. How to deploy :- Step 1 :- Download the source code zip file and extract it. Step 2 :- Open terminal and navigate to the extracted folder location. Step 3 :- Run the command 'mvn install' to install the dependencies. Step 4 :- Run the command 'mvn spring-boot:run'. The application will start running on port 8080. Cloud Server Configuration : There are a few steps that need to be taken in order to deploy a cloud for the Smartfarmer iot enabled smart farming application. 1. Create a new project in the Google Cloud Platform console. 2. Within the project, create a new App Engine application. 3. Download and install the Google Cloud SDK. 4. Within the App Engine application, create a new service. 5. Within the service, create a new version. 6. Deploy the application to the App Engine service	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Crop Sensor Data Analysis	Python, Tkinte
2.	Security Implementations	Use of firewalls: Firewalls are used to protect the network from unauthorized access	SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	To ensure scalability, the application will need to be designed for horizontal scaling. This means that the application can be run on multiple servers at the same time, with each server handling a portion of the traffic	Using MongoDB, NodeJS, ReactJS, ExpressJS, Redux
4.	Availability	The application would allow farmers to track the status of their crops and soil health in real-time, as well as receive information on the best practices for crop care. The application would also provide alerts to farmers in the event of changes in weather or pests.	sensors, software applications, and cloud-based data storage and analytics.
5.	Performance	The performance of an IoT enabled smart farming application depends on a number of factors, including the quality of the sensors and devices used, the connectivity of the system, the amount of data being collected, and the algorithms used to analyze the data.	soil moisture levels, temperature, and air quality sensors.