This writeup outlines the system requirements for the Digital Farm Management System (DFMS).

1. **Frontend:**

* **Next JS** as the Main frontend framework - Next JS which is a React framework is popular for creating fast web UI frontends with no reload making it versatile for our Digital Farm Management System.
* **Typescript** as our type safe framework. It will help write less bug code ensuring also less runtime errors.
* **Tailwind CSS** as the Cascading Style Sheets framework to help write less CSS for faster rendering.
* **Shadcn** for UI components.
* **Material UI** will also be used for adding themes to our DFMS to make it look nice and attractive.

1. **Backend:**

* **PostgreSQL** Database with **PostGIS** extension shall be used as the main database to store farm polygons, store farm activities such as planting, addresses to GIS layers such as soil moisture and also store important farmer details such as farm number.
* **FastAPI** – FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints. We shall use FastAPI to call API such as OpenWeatherAPI for real-time weather information.
* An Integration of **FastAPI** and **Apache Airflow** shall also be used to build a special backend to automatically schedule retrieval of satellite imagery specifically Sentinel, run an algorithm (algorithms shall be developed from the GEE analysis code being done by Antoinette, Anne and Edna) every 5 days, produce outputs which shall be saved to Geoserver and used in the frontend for farmer’s view.
* **Geoserver** – Geoserver shall be used to manage our GIS layers once they’ve been published there by our backend. Geoserver will also aid in providing the OGC protocols such as WMS, WFS, TMS for rendering satellite imagery on Leaflet or OpenLayers map.
* **Supabase** - We shall use Supabase which is a managed backend for data storage and authentication. This free platform shall help us authenticate farmers using Phone numbers and farm number without requiring username, email or password. OTP based login shall be used to make the system onboarding as smooth as possible.
* **Docker, Docker Stack & Kubernetes** – This shall be used to containerize different components of the application and ensure they communicate efficiently in an isolated environment. Kubernetes shall be used for autoscaling the infrastructure from frontend to backend making sure as traffic increases, the web application (DFMS) shall remain up, lighting fast and working as expected (at least 99.95% uptime)
* **Github Actions** – This shall be used for Continuous Deployment and Deployment with at least three code admins i.e. Nancy, Kelvin and Walter to make sure code follows the best practices and that the suggested features have been added. Only upon verification by at least two admins, the code shall proceed to production where now the farmers can receive updated Digital Farm Management System (DFMS).

Write-up prepared by Louis Wambua, GIS Developer Intern, Statsspeak

February 12, 2025

