**SOFTWARE ENGINEERING**

**FERTPPM– MVP specification**

Tagline: IMPROVE YOUR CROPHEALTH, SAVE A COIN

Project team members:

**Benson Mwangi** - Full stack Software Engineer and Artificial intelligence programmer

Benson Mwangi is a well-rounded developer with a passion for building efficient and user-friendly websites and applications which are powered by Artificial intelligence. He has extensive knowledge of frontend, backend development and machine learning, and is fluent in programming languages like Python, C/C++, HTML, CSS, and JavaScript and platforms like Django, Flask, TensorFlow, Keras, pytorch and Neural networks, Tkinter.

Benson is renowned for delivering high-quality projects on time. He is self-sufficient and can work well independently or in groups. His attention to detail and ability to solve complex problems makes him an asset to any project. In addition to his technical skills, Benson also has excellent communication and collaboration skills. He can communicate technical concepts clearly and concisely, and is always open to comments and suggestions.

Overall, Benson is a dedicated and trusted full stack Software Engineer and AI programmer who is committed to building exceptional websites and applications that meet and exceed customer expectations.

His role in this project is to develop backend on the system

**Stephanie Iman -** Front End Engineer and Web developer

Stephanie Iman is a highly technical and motivated individual with ambitions of coming up with very user-friendly platforms, be it applications or websites. She aspires to work on applications friendly to the disabled and lesser people in the community. She is well educated on Python and its libraries, CSS, HTML, Bootstrap, JavaScript as well as working with Rest API and is able to write Object Oriented code. In addition, she has knowledge in unit testing and database like MySQL

In a work environment, Stephanie is friendly, and accommodative, works well in groups and is very open minded and innovative. She is adventurous and wants to understand hence explore opportunities of creativity in development. She has excellent communication skills and is committed to learn as much as possible to get to her targets

Her role in this project is to develop front end of the system

Technologies

- Languages:  Python, HTML, CSS, JavaScript, Django template language, php

- Frameworks: Django, Flask, node.js

-Database: PostgreSQL + ORM concept

- Libraries: Bootstrap.

- Platforms: Virtual environment, Amazon Web Services, VSCode,

- Hardware: Web Servers, Application Server, databases, Firewalls, Load balancer.

- Books/Resources: <https://www.w3schools.com/Django>

Technology Compromise

Flask vs. Django

When choosing which Python web framework to use for our project's backend, we looked at both Flask and Django. Flask is known for being lightweight and flexible, while Django is more comprehensive and opinionated. In the end, we decided to use Flask because our project requirements were relatively simple and we wanted the flexibility to customize our implementation as needed. In addition, Flask's modular design makes it easy to integrate the specific libraries and tools we need for our project. Bootstrap vs Materialize

For our front-end design framework, we looked at both Bootstrap and Materialize. Bootstrap is widely used and well-documented, with a large number of templates and components available. On the other hand, materialize offers a more modern design aesthetic and focuses on mobile-first design. In the end, we chose to use Bootstrap because of its larger community and rich documentation, which makes it easier to find solutions to common problems and integrate it with other libraries and frameworks. However, we like Materialize focus on accessibility and might consider using it in future projects.

In making these decisions, we considered the specific requirements of the project, our team's experience with different technologies, and the documentation and support available for each option. We also consider factors such as ease of use, flexibility, and community support. By carefully considering our technology options, we were able to create a project that met our needs and provided a solid foundation for future growth.

Challenge

The portfolio project aims to solve the challenge of delivering nutrients efficiently and reliably to the crops. The traditional method of determining what amount of fertilizers element to apply on the crop to achieve the required amount of nutrients measured in terms of PPM (parts per million) over the time or in daily feeding regimes can be time consuming and unreliable, leading to under or over feeding the crop thus causing unhealthy and imbalanced growth. By creating this website application for calculating and tracking the optimal amount of fertilizer elements to use per day as per the requirement by the crop tissue test analysis laboratories like Cropnuts or Eurofins Scientific laboratory. The project aims to streamline the process and provide a more convenient and seamless experience for crop growers and agricultural departments across the world.

However, it is important to note that the wallet project will not solve the challenge of accurately measuring the amount of element already available in the crop or the growing media to determine what extra element is needed to top up to achieve the required target. This will always have to incorporate regular growing media and crop tissue laboratory test either internally within the organizations or by external crop science laboratories. The portfolio project aims to help crop growers and their customers, including Horticultural growers, flower growers, department of agriculture, environmental regulatory bodies and other businesses that rely on crop production to operate. By providing a centralized platform for calculating Fertigation feeding regimes, planning and tracking fertilizers consumption, the project aims to simplify the process for all stakeholders, while improving efficiency and accuracy and overall reliability.

This project is relevant and location-specific, as it will primarily serve crop growers in a specific region or continent. The specific location may vary depending on the scope and size of the project, but will likely be limited to a defined geographical area to ensure an efficient and timely crop feeding regime and economical routine.

Risks

Technical risk

1. Technology Compatibility:

The technical risk to full stack developers is the compatibility of different technologies used in the project. Using different technologies such as programming languages, frameworks, and databases can cause compatibility issues that lead to conflicts or errors. To prevent this risk, we will thoroughly research and select compatible technologies and ensure that they can work together seamlessly.

2. Ability of extension:

Another technical risk is the project's ability to scale as user demand increases. If the system is not designed with scalability in mind, the system may become slow or unresponsive as the number of users and data increases. To avoid this risk, we will design the project to be scalable from the ground up, use technologies capable of handling high volumes of data and traffic, and implement load balancing techniques and self-expanding.

 Non-technical risk

1. Project Management:

Non-technical risk is the ability to effectively manage projects, including meeting deadlines, managing resources, and communicating effectively with team members and stakeholders. Poor project management can lead to delays, cost overruns, and quality loss. To prevent this risk, we will implement effective projects

Management practices, such as the use of flexible methods, regular progress reports and effective communication channels.

2. Fast Range:

Another non-technical risk is scope change, where the scope of the project extends beyond the original plan, resulting in additional work and costs. This can lead to delays and increase project complexity. To prevent this risk, we will clearly define the scope of the project and communicate it effectively to team members and stakeholders. We will also monitor and control changes to the scope of the project, ensuring that any changes are reasonable and in line with the overall project goals.

In summary, as a full stack developer, it is important to know and minimize technical and non-technical risks. By choosing compatible technologies, designing for scalability, implementing effective project management practices, and controlling project scope, we can reduce the potential impact of risks and ensure the success of the project.

Infrastructure

Branching and merging

As a full stack developer, my team's branching and merging process depends on the complexity and size of the project. For smaller projects, we usually follow the GitHub process, which is to fork features from the master branch, commit the changes, checkout them, and then merge the changes back to the master branch. This allows us to always maintain a stable and deployable codebase.

For larger and more complex projects, we may adopt a more complex branching and merge strategy, such as Gitflow, which includes multiple branches for different development and release phases. This strategy helps us more efficiently manage and coordinate feature development, testing, and deployment.

Deployment strategy

Our deployment strategy involves automated continuous integration (CI/CD) and delivery processes. We use our special tools to automate building, testing, and deploying our application code. This ensures that the application is always in a deployable state and any changes are automatically deployed to the production environment.

We also use container technologies like Docker to package the application and its dependencies into a portable image that can be deployed consistently across different environments.

Population data

To populate our application with data, we use a combination of manual and automated methods. For testing and development purposes, we use dummy data generated by scripts or tools. For production data, we may import data from external sources or migrate data from an existing database. We ensure that the data is clear, consistent, and consistent with the application schema.

Testing tools and procedures

We use a variety of testing tools and processes to ensure that our application code is reliable, performant, and error-free. Including:

##### Unit test:

We write unit tests for each individual component of the application using tools like Jest, Mocha or PHPUnit.

##### Integration testing:

We perform integration testing to verify that the different components of our application work together as expected. We use tools like Cypress or Selenium to automate this process.

##### Performance test:

We test the performance and scalability of the application using tools like Apache JMeter or Gatling.

##### Security check:

We perform security testing to ensure that our application is safe and free of vulnerabilities. We use tools like OWASP ZAP or Burp Suite for this purpose. In a nutshell, as full stack developers, we ensure that our infrastructure is scalable, maintainable, and reliable using consolidation and distribution strategies, appropriate branching, automated deployment, and powerful testing tools and workflows.

Existing solutions:

As a full stack developer, we consider ourselves developing web or native desktop apps. In this context, here are some existing solutions that might be similar to our proposed project:

##### Fertilizer Optimizer

This app assists you in using fertilizer more efficiently to optimize your fertilizer investments. The app asks you for information on crops grown, area planted, expected crop sale prices, fertilizers costs and the budget you have to invest in fertilizer products. Based on robust crop response functions, it will calculate the most profitable combination of fertilizers to purchase and advise you on crop and site- specific application rates. The app can also take into account any integrated soil fertility management (ISFM) practices to tailor the fertilizer recommendation to your farm.

**Similarities:**

Both Fertilizer Optimizer and our proposed project leverage technology to provide a seamless and convenient service for users.

**Difference:**

Our project focuses on giving the amount of fertilizer to use in terms of ppm and converting this amount into kilograms to achieve the required nutrients needed to feed the crop per day. It also integrates other agricultural technologies like fertilizer recycle and UV treatments, while Fertilizer Optimizer focuses on providing a combination of fertilizers to purchase and advise you on crop and site-specific application rates.

##### Cropnuts LabPro:

Cropnuts (www.cropnuts.com) is a leading agricultural, environmental, and food safety testing laboratory & farmer advisory service, based in Kenya and working with clients across Africa.

Cropnuts LabPro offers agronomists and agents (including agro-dealers and commercial agents) access to Cropnuts' laboratory and agronomy platform which provides complete lab results and crop/soil specific recommendations to maximize yields.

**Similarities:**

Cropnuts LabPro and our proposed project are both applications that offer support to Crop growers.

**Difference:**

Our project focuses on giving the amount of fertilizer to use in terms of ppm and converting this amount into kilograms to achieve the required nutrients needed to feed the crop per day while Cropnuts LabPro offers agronomists and agents (including agro -dealers and commercial agents) access to Cropnuts' laboratory and agronomy platform

##### 3. FARMLAB

FARMLAB is a B2B solution which uses infra-red technology to provide on-site, quick, affordable and accessible soil analysis for fertilizer recommendations and soil correction input advice to farmer interest organizations.

The FARMLAB model allows each lab to be self-owned and operated as mobile or stationary laboratory units, or contracted for on-site soil testing.

**Similarities:**

FARMLAB and our proposed project both leverage technology to provide a convenient Fertigation and crop nutrition experience for users.

**Difference:**

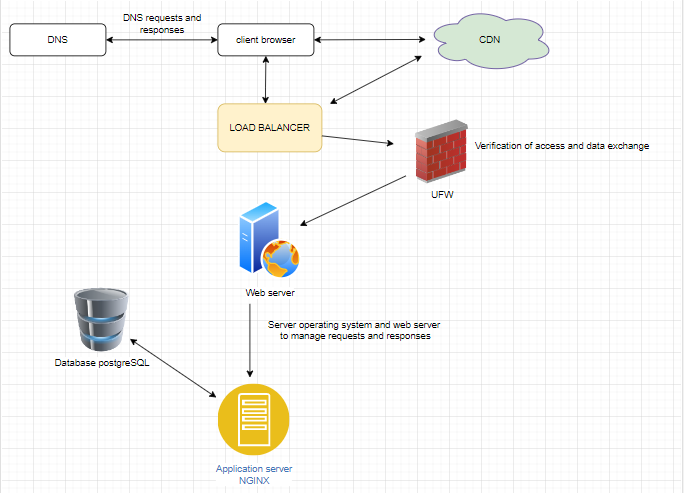
Our project application Focuses on giving the amount of fertilizer to use in terms of ppm and converting this amount into kilograms to achieve the required nutrients needed to feed the crop per day while FARMLAB allows each crop science lab to be self-owned and operated as mobile or stationary laboratory units, or contracted for on-site soil testing.

If we re-implement a proven solution, we evaluate different options and choose the one that best suits our specific requirements and specifications. For example, if we are developing an image compression algorithm, we will evaluate different classes of image compression, such as lossy and lossless compression, and different types of compression algorithms, such as Run Length Encoding (RLE), Huffman encoding and Transform encoding.

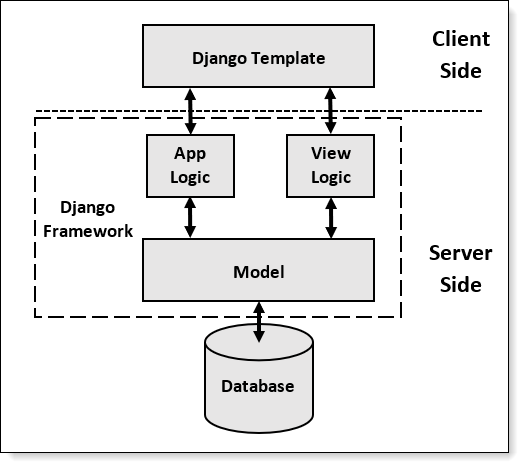
After evaluating each option, we may choose to implement a conversion encoding because it offers a balance between image qualities and file size. Transform coding involves converting image data into a new representation using a mathematical function, such as the Discrete Cosine Transform (DCT), which can reduce the amount of data required to represent the image while still maintaining high image quality. Other unique aspects of Transform encoding include the ability to compress images without losing important information and the ability to work well with natural and composite images. However, transform encoding may require more computational resources than other compression algorithms, which is an important consideration when developing web, Desktop apps or mobile apps.

ARCHITECTURE

 The Below diagram is a representation of what happens when you type in the URL leading to the FERTPPM website. To start from the very beginning when the user types in a URL, a DNS lookup is triggered to convert to an IP address of the website domain name which is well interpreted by the computer. Depending on what the user wants, a get request is initiated. The legitimacy of the end user will be questioned at the firewall which will be the access point of the web server. If the get request in the web server is not successful, a servlet request will be made to the application server where the data will be processed and the results delivered to the client as requested.



Web Application Architecture Overview



APIs

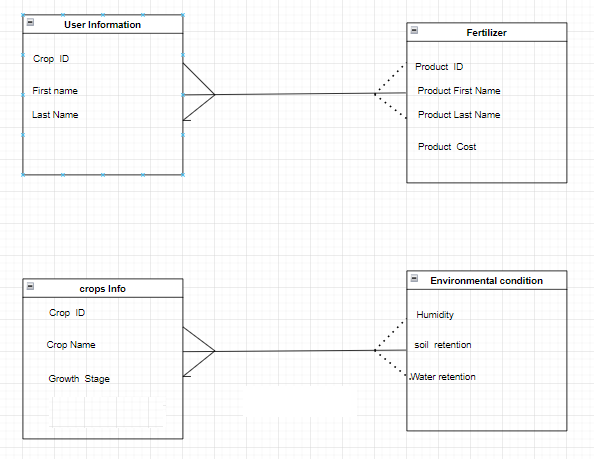
Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

Our application will be communicating with the following APIs

1. <https://api.whatsapp.com/>

To connect our system to WhatsApp for customer support

Data Model



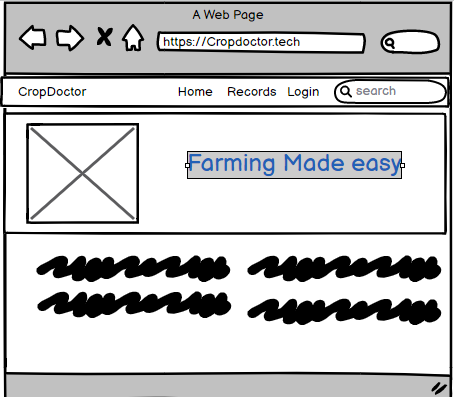
Above is a representation of how data will be interacting with each other. To start with, the first square represents the credentials of the individual interacting with the system. We also included a primary key to deal with scenario where tabulated data is similar. The relationship between the user and the Fertilizer which in this case we refer to them as products, is seen where there is need to add information regarding the fertilizer like the scientific name, i.e., Calcium Nitrate which needs to be done manually.

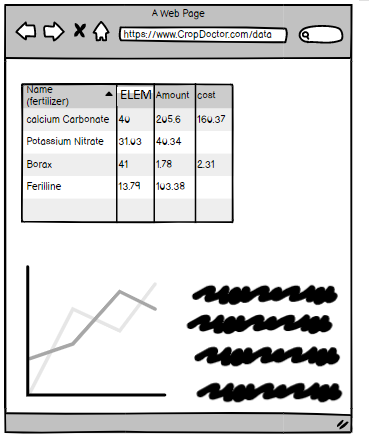
In the crop info container, it is only logical to identify the type of plants that are available. We also made sure to include the growth stage seeing to the fact that different crop stages need different nutrients. The last container comprises the environmental conditions present as some weather conditions may affect the amount of Fertilizers nutrient being absorbed by the plants.

User Story

|  |  |
| --- | --- |
| User Story | As an aspiring modern farming, I want to be able to keep track of the nutrient intake of my plants and know when I need to increase or decrease the amount of fertilizer to ensure healthy plants |
| Accepted criteria | * Be able to securely log in as an admin to change costs or quantity of fertilizers * User is able to append values of products * User is able to get quantities without having to calculate manually * Accounts are updated concurrently after every validation |

MOCK-UP





# Portfolio Project week 1

Being able to write a project from scratch with 12 months of learning is so eye opening because you get to have a personal understanding of the functionality of each piece of code. However you also get to face the wrath of debugging, logic and many more challenges that give you a headache. Here are the challenges we faced

## Challenges

When doing our research, we had an idea of utilizing a Chabot package to get automated answers. We failed to realize that the algorithm we had to use called Chatterbot only worked on python 2 and lower versions which had to force us to downgrade python to a lower version. This was definitely not something we were willing to cave in and work on it. In our opinion, we needed to advance with technology. So that said we had to change it.

When we got another project idea that we had to build using python GUI interface called tkinter, we noticed it was only capable of running as a desktop application and not a web application. We thought of it as a good challenge thinking of how we can change its functionality by trying to come up with a way to convert this desktop app to a web app that can run on a server and other internet stack. This has been quite a hustle with no success since the framework can never be translated by a web browser. We decided to take up the challenge of building a web app with different design and frameworks from the previous idea we had at the beginning. We took up the Django framework for backend with PostgreSQL database.

For the non-technical challenge aspect, we had a difficult time with time because most of the time, our free times are different. One of us has a day job and the other studies mostly at night, so it’s sometimes not convenient to have compatible time that matches both of us.

## Collaboration

“Watch the Master”

“Watch the Master” is a scenario that often occurs when one person in the pair is more experienced than the other. A common situation for this phenomena is a senior software engineer working with a novice software engineer. The senior software engineer may skip what they think to be unnecessary for explanation, while the novice software engineer may feel too timid to ask questions or challenge the approach taken because also the deadline for project delivery is tight, deferring to the senior member of the pair.

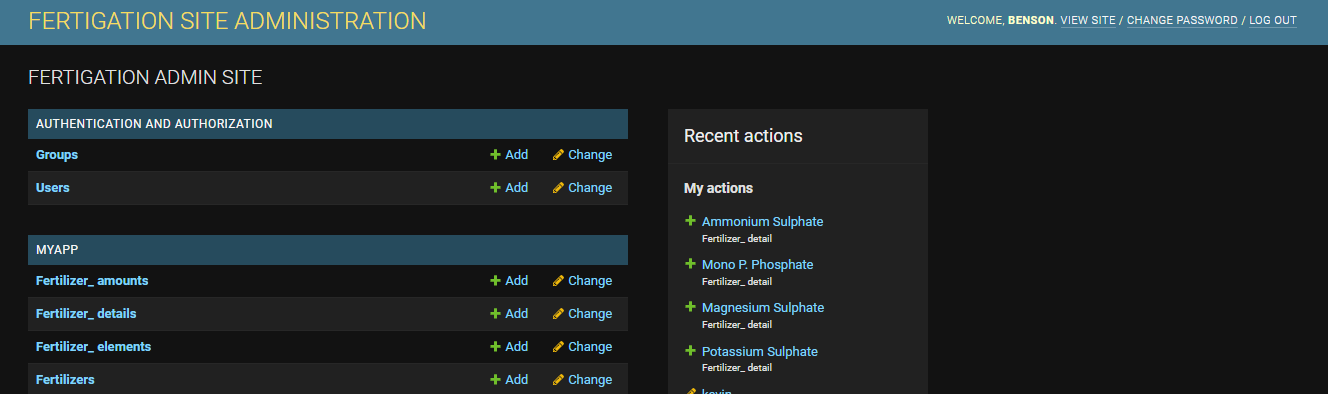
This is similar to the collaboration challenge our team encountered of silent disengagement where one of the team members had more experience in software development compared to the other one who was very fresh to the software development technology. This demanded more time to let both members are well knowledgeable with the technology to use before development starts.

## Project Updates

We do have a lot we are planning to do to vamp up our project. As of now, in the landing page, we have added authentication functionality and support details about the program. see below.



We have also added models to our app to help us communicate with database upon GET and POST requests made by the user to the server. See below



We plan to add more of what the app does and how it works to give a little bit more functionality.

In addition, we will add a tutorial video of how to navigate the app for a better user experience. Being that it is mostly an app that involves a lot of data manipulation and retrieval, it can be quite overwhelming using the app. We will also incorporate a chat bot which will make it easier for the stakeholders using the app to communicate updates and tips on better farming techniques.

## Project progress

On a scale of 1 – 10. I would rate 7 as I feel like we are progressing quite well. We are good in different aspects of the app so we sort of balance off each other. Considering that we are like 30% done we believe that we can finish the project in time.

# Progress Report: MVP Complete

**2nd week project progress**

On a scale of 1 to 10, this week's progress is rated at 8. 2.

We measured the progress by completing planned tasks, staying within the coordinated project schedule, and successfully resolving issues. Considering the results of the second week, we will give it an evaluation of 8th place. We have overcome challenges related to project dependencies, implemented dynamic sorting and filtering options, and integrated file upload functionality. We also made efforts to improve collaboration and communication with other developers. Despite some delays in schedule due to challenges faced, overall progress is on track.

In this second week of developing the project portfolio proposal website, we made further progress in implementing key features and addressing challenges encountered. We have completed the following tasks:

1. Project dependencies:

Extensive research has been done to identify and address the complexities involved in managing project dependencies. We successfully implemented a dependency management system to ensure smooth integration and functionality between the various components of the website.

2. Dynamic sorting and filtering and CRUD:

We attempted to implement dynamic sorting and filtering options for the project list. We explored a variety of techniques and successfully integrated a solution that allowed users to sort and filter data based on various criteria such as date, category and products.

3. File upload of project assets:

We solved the challenge of managing file uploads for project resources by spending time researching and implementing a suitable solution. We successfully integrated a file upload feature that allows users to upload relevant project resources such as images and documents and link them with different resources.

4. Cooperation and Communication:

Efforts continued to improve collaboration and communication with other developers. We held Virtual meetings regularly to discuss progress, address concerns and ensure smooth collaboration. We are using Instant messaging platforms to enable fast and efficient communication.

That said, we are happy to state that the login, registration of new user and logout process are complete and functional. In addition, we also were able to create a database functionality i.e. CRUD where the following were captured:

* The super user is able to add or remove other users who will be interacting with the system
* The user is able to insert and edit data into the system using both post and get methods.
* The database can smoothly receive data, and update the record after any alteration from the users.
* In case of any deletion needed, the database has been created to capture the changes made.
* The user can be able to navigate through different pages with the help of navigation bars and menus and links
* The system has also been integrated with social media APIs platform to redirect the user to any needed support.

The incomplete aspects of the website are as follows:

* In case user forgets password and username, we need a recovery mechanism
* We need to add resource pages with more information of how the system works and the purpose of the system for user to understand the system better
* We need a well laid color coordinated landing page.
* We also need a blueprint showing the system relationship and dependencies that can enable other stakeholders to understand the flow.
* We also need to polish up the main function that will drive our system business logic.

**CHALLENGES**

**Technical challenges**

To start with, we view our system as a full-stack web application. That said, it incorporates up to 5 front and backend languages and platforms, e.g. JavaScript, Python and Django, PHP, HTML, CSS among others. This makes it challenging to manage dependencies.

Among the many bugs we encountered, the one that stood out for us was where we had to migrate the project to the deployment environment. The issue come in getting the domain name to run in HTTPS protocol and SSH certificate.

Considering that the project is big, the amount of time it took for deployment to be initialized was extensive.

The challenges associated with managing project dependencies, implementing dynamic sorting and filtering, and handling file uploads were successfully overcome with a thorough study and adjustment of the project plan. The team remains vigilant and adapts to handle additional unforeseen complications.

**Non-Technical challenges.**

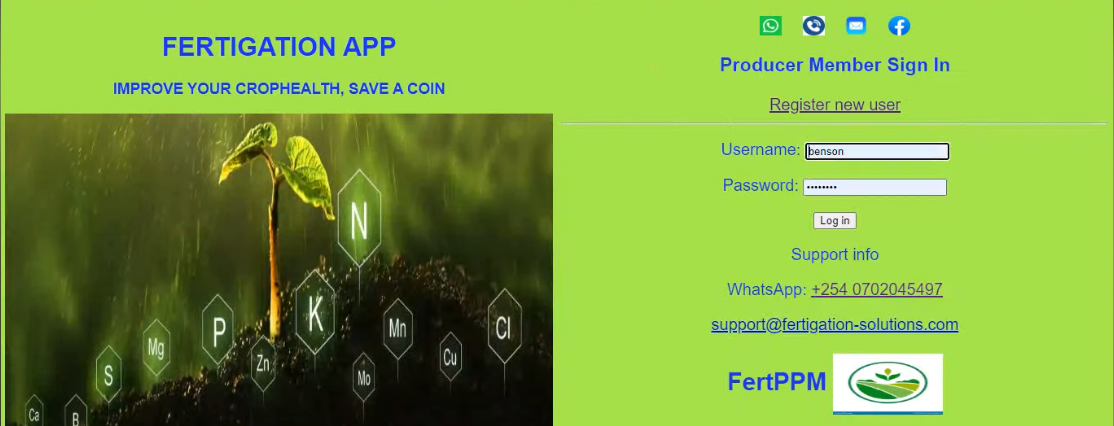
Based on the progress achieved in the second week, project completion and MVP definition results within the specified time period are evaluated as follows:

Considering that there are so many aspects of the project that need perfecting and the fact that we are working on a given time line, a bit of pressure is building up. We hope at the end of it all, we will meet the deadline and the app will be fully functional

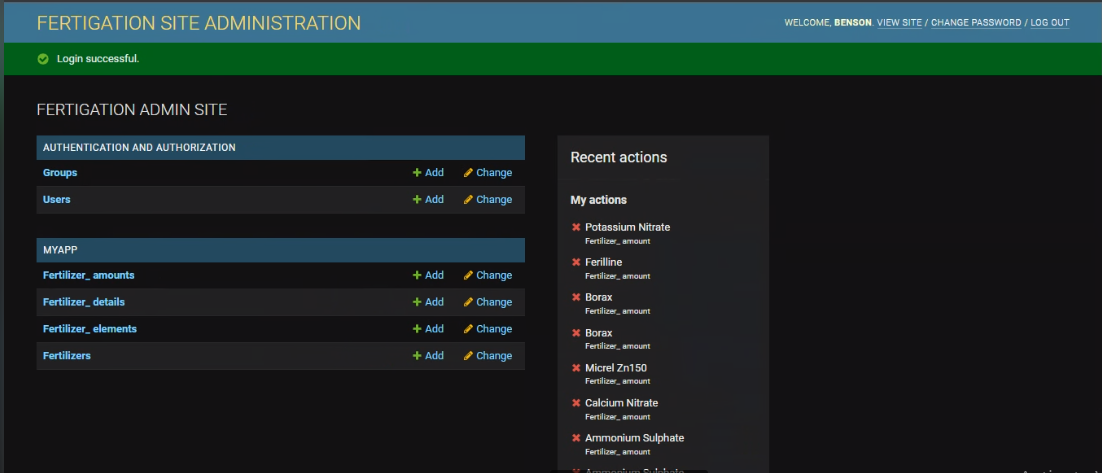
Efforts to improve collaboration and communication between team members have yielded positive results. Regular virtual meetings and the use of collaboration tools enabled smoother communication channels and minimized the impact of this non-technical challenge. However, our activities of daily living have been interfering with collaboration which delays our progress

**Screen Shots**

Below is a sneak peek of the application we are developing so far.







References

* <https://www.getapp.com/industries-software/farm-management/f/api/>
* <https://intranet.alxswe.com/concepts/135>
* <https://trello.com/b/qvvvw9uK/portfolio-project>