**GreenTech: IoT-Based Smart Hydroponic System with Mobile App Integration**

Ello, Jhusthine  
Muripaga, Raihanie  
Omandam, Richie Mae D.  
Plaza, Benjie T.  
Vergio, Hannah Grace D.

Department of Information Technology

College of Information Technology and Computing

**University of Science and Technology of Southern Philippines**

Cagayan de Oro City

May 2025

1. **System Overview**

The GreenTech Mobile Application that integrates backend, and IoT components monitors the real-time sensor data. This system is designed for convenience that allows users to view their device connected with hydroponics through a mobile interface.

1. **Frontend – Flutter:**

The mobile application is developed using Flutter as frontend for cross-platform interface for responsive UI. This retrieves and displays data on the dashboard in real-time, giving the users an up-to-date overview of environmental conditions for their hydroponics. After the user logs into their account, the dashboard will display sensor data such as:

* **Temperature**
* **Humidity**

1. **Backend – Fastapi:**

The data from sensors is sent to the fastapi backend via HTTP requests. The fastapi then parses and stores the data on the dashboard, allowing the frontend to access it.

1. **IoT Integration – ESP32 & Arduino IDE:**

The ESP32 microcontroller, programmed via Arduino IDE, reads environmental data using DHT11 for both temperature and humidity. These readings are sent to the fastapi backend in real time using HTTP requests.

1. **Development Challenges**

Developing a real-time environmental monitoring mobile application using Flutter and integrate it with fastapi, Arduino-based IoT devices is the challenge itself. From hardware components to backend integration and frontend visualization.

* **Real-Time Data Fetching from ESP32 to FastApi**:

Establishing and integrating a stable and efficient communication between microcontrollers and the FastApi was one of the most significant challenges our team encountered. With the microcontroller’s limitation, we only rely with managing power consumption and maintain consistent Wi-Fi connectivity to fetch real-time data.

* **UI/UX Styling**

While we already have the UI & UX visualization for our frontend, building it using the Flutter was still a challenge because while UI is visually appealing, we still need to keep the performance optimized and make sure to display the updated data.

1. **Future Improvements**

To improve the system’s performance, scalability, and user’s experience and convenience, the following are the proposed enhancements;

* **Sensor Diagnostics**

Add a diagnostic page to monitor the health or status of each sensor, and a push notification if one of the devices are not working.

* **Offline Mode**

The system’s limitations include the reliability of power consumption and internet connectivity, which means the user must have internet connections at all times for both the devices and mobile application to ensure a real-time data fetching. Enabling an offline caching in the app with auto-sync when availability of internet is present may enhance the system.