3 METHODOLOGY

3.1 Materials

3.1.1 Software

Describe all the software that you used including names and other details.

Table 1 Software Requirements: Development of Vertigrow Vertical Farming System

SOFTWARE	**SPECIFICATION**
Supported Browsers	Major Web browsers include Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
UX/UI Prototypes and mockups	Figma
Development Software's	Visual Studio Code
Programming Language	HTML, CSS, Javascript
DBMS	MySQL
Arduino IDE	Used for programming the Arduino microcontrollers.
Ubidots	A cloud-based platform for real-time data visualization and monitoring.
ESP8266/ESP32 Libraries	Libraries are essential for enabling Wi-Fi connectivity in IoT devices.
ThingSpeak	An IoT analytics platform for storing and analyzing sensor data.
Fuzzy Logic Toolbox	Software used to implement fuzzy logic algorithms for data processing.

The specified Table provides a clear overview of the software components utilized in the development and operation of VertiGrow. Each element is chosen thoughtfully to ensure efficiency, compatibility, and effectiveness throughout the Vertigrow system's lifecycle.

Table 2 Non-Functional Requirements: Development of Vertigrow Vertical Farming System

NON- FUNCTIONAL	DESCRIPTION
PERFORMANCE	• Process and respond to IoT sensor data in real time. • Efficiently handle multiple sensor inputs.
RELIABILITY	• High Availability and reliability. • Ensures continuous monitoring and automated control of the farming environment.
ENERGY EFFICIENCY	• This includes optimizing the power usage of sensors and controllers and implementing energy-efficient algorithms.
MAINTAINABILITY	• Designed for easy maintenance and updates, with clear code
SCALABILITY	• Ability to integrate future growth

The non-functional descriptions provide important insights into various elements of the system beyond its core functionalities, highlighting essential attributes that contribute to a well-rounded and user-focused software and hardware solution.

Table 3 Functional Requirements: Development of Vertigrow Vertical Farming System

USER TYPE	DESCRIPTION
Admin • Environmental Monitoring • Automation • Energy Management • User Interaction • Data Storage and Analysis	
Farmers Workers • Environmental Monitoring • Automated Farming Operations • Energy Management	

• User Customization	

3.1.2 Hardware (if there is a hardware component)

Describe the hardware that you used.

Table 3 Functional Requirements: Development of Vertigrow Vertical Farming System

HARDWARE	SPECIFICATION
ESP8266 /ESP32	A low-cost Wi-Fi microcontroller that enables wireless communication and control of the system. It can be programmed using Lua or Arduino IDE.
SOIL MOISTURE SENSOR	A sensor that measures the moisture level in the soil, helping to determine when irrigation is needed. It provides real-time data for optimal watering.
TEMPERATURE SENSOR	A device that measures the ambient temperature of the environment, crucial for maintaining ideal growing conditions.
HUMIDITY SENSOR	A sensor that monitors the humidity levels in the air, which affects plant growth and health. It helps in adjusting the environment accordingly.
LIGHT SENSOR	A sensor that measures the intensity of light, ensuring that plants receive adequate light for photosynthesis. It can trigger adjustments in artificial lighting.
WATER LEVEL SENSOR	A sensor that detects the water level in the reservoir, ensuring that the system maintains sufficient water supply for the plants.
Pumps and Valves	Water pumps used for irrigation, delivering water to the plants based on the readings from the soil moisture and water level sensors.
Solar Panels	A renewable energy source that converts sunlight into electricity, providing a sustainable power solution for the VertiGrow system. It helps reduce reliance on traditional power sources.

Led Grow Lights	Artificial lights that provide the necessary light spectrum for plant growth, especially in indoor vertical farming setups.
Power Supply	A power source that provides the necessary voltage and current to operate the NodeMCU and other components, typically using solar energy for sustainability.
Fuzzy Logic Controller	A computational model that processes data from various sensors to make decisions about irrigation and environmental control based on fuzzy logic principles.

This table provides a comprehensive overview of the hardware components used in the VertiGrow system, highlighting their roles and functionalities in supporting efficient and sustainable vertical farming practices.

3.1.3 Data

Indicate the source of the data, type of data, year of acquisition, and other pertinent details.

Data Sources	Type of Data	Year Of Acquisition	Details
Soil Moisture	Soil moisture percentage	2024	
Temperature Sensor	Temperature in degrees Celsius	2024	
Humidity Sensor	Humidity percentage	2024	
Light Sensor	Light intensity in lux	2024	
Water Level Sensor	Water levels in liters	2024	
Cloud Storage Platform	Data and Analysis	2024	
User Input	Operational Data	2024	
Historical Agricultural Data	Comparative Analysis	2024	
Energy Consumption Data	Performance metrics	2024	
Research Studies	Literature Review Data	2021-2024	

This structured overview provides a comprehensive understanding of the data utilized in the VertiGrow project, emphasizing the sources, types, and significance of real-time data analysis in enhancing vertical farming practices.

3.2 Methods

The headings given here for the methods are only suggestive. Adopt what is appropriate for your research. For example, you would have experimental design if you used one such as multivariate method and the like which would describe the number of tests you did and the conditions for testing.

3.2.1 Research Design

Sample: The study uses developmental research method

3.2.2 Fuzzy Logic and Low-Energy Adaptive Clustering Hierarchy Algorithm

There are two algorithms utilized in this research that encompass two different parameters. First is the Fuzzy Logic which will be used as a decision making algorithm for water irrigation of the vertical farm. Fuzzy Logic is optimal for the automation of vertical farm's water irrigation systems. Another algorithm that will be utilized is the Low-Energy Adaptive Clustering Hierarchy Algorithm (LEACH) which is used to lower the overall energy consumption of the vertical farm by optimizing the microcontrollers and sensors, using LEACH algorithm it makes the vertical farm more efficient and sustainable. These two algorithms will help the researchers objectives into developing a more efficient and sustainable Smart Vertical Farm.

3.2.3 Process Model

The process model will outline the baseline of the system software development of VertiGrow through a web application, as its purpose relies on the output data from the monitoring sensors. The flow and process of the system are continuous; therefore, an Agile method is a suitable approach. This methodology allows for easy identification and resolution of minor project issues, significantly reducing problems during the development stage.

3.2.4 Agile Methodology

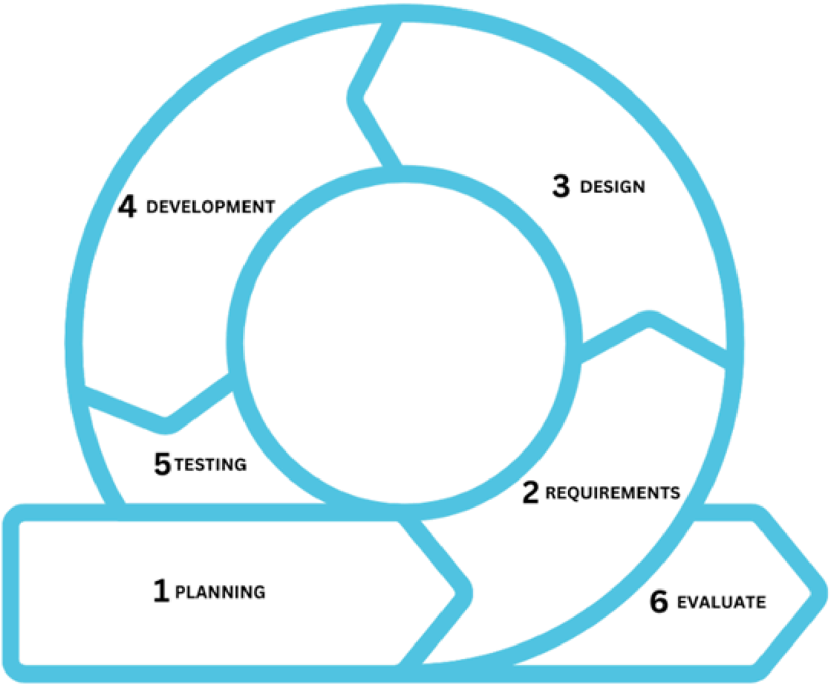

Figure ?.

The phases will include planning, requirements, design, development, testing, and evaluation. The planning stage involves formulating ideas for the process and concepts of the system, which includes development and utilization. Additionally, identifying core features of IoT monitoring, energy management and decision-making algorithms such LEACH and Fuzzy Logic.

3.2.4.1

During the requirements phase, this document will outline the functional and non-functional requirements of the system. The functional requirements involve monitoring soil, humidity, and light conditions. Additionally, the integration of Fuzzy logic for smart decision-making and LEACH for efficient energy consumption will be included. The entire system must be powered by solar energy with minimal downtime.

The design phase will outline the development of the architecture and system blueprint. This includes the system architecture, which involves a modular design for IoT devices, energy management, and algorithms. In the web system prototype, a dashboard will display real-time soil data, energy status, and alerts. In the algorithm design, we will define the fuzzy logic rules for soil condition decisions (e.g., watering when soil moisture is below a certain threshold) and outline the LEACH cluster-based routing for sensor communication to save energy. Lastly, in hardware design, we will plan the layout for sensors, solar panels, and microcontrollers.

In the development stage, integrating the system iteratively this includes IoT sensors that are deployed in the soil to gather real-time data on moisture, temperature, and nutrients. Next, the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol is implemented for energy-efficient communication between the sensors. Fuzzy Logic is then used to analyze the data and automate decision-making, such as optimizing irrigation and fertilization schedules. A solar energy management system is integrated to ensure sustainability by powering the sensors with renewable energy. Finally, a user-friendly interface is developed to allow users to easily access and visualize the data, enabling informed agricultural decisions.

Testing phase, will integrate a controlled vertical farm without the implementation of any IoT systems, algorithms, or solar technology. In contrast, a vertical farm that includes IoT, algorithms, and solar implementations is represented by VertiGrow. Through this testing, we can determine the effectiveness of VertiGrow in terms of crop quality and yields compared to the controlled vertical farm.

Lastly, the evaluation phase involves presenting the system to the client, the City of Agriculture, gathering their feedback, noting the system's shortcomings, and using this information as recommendations for improvement.

3.2.4.1 Requirement Analysis5

3.2.4.2 System Design

The user will manually login into the system entering the username and password. When the user is logged in successfully, it will then redirect to the dashboard of the system. The dashboard shows the analytics of the vertical farms such as the humidity, temperature, water quality & level, soil moisture, and light which is the justification for the monitoring system. Moreover, the system has an automated watering system so that the user will not manually water the vertical farms.

3.2.4.2.1 Flowchart

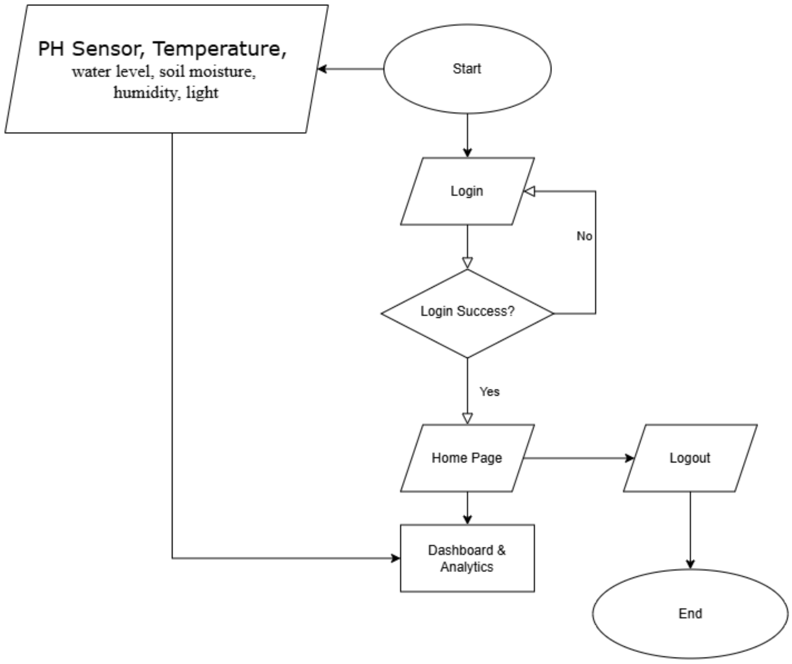

Figure ??? Flowchart: Development of the Vertical Farming Monitoring System

The figure above shows that the user has the access to the system to monitor their own respective farms. The main input for this system is the real-time data gathered from the crops which are the water quality & level, temperature, soil moisture, humidity, and light. These data will then gather and appear to the web system's dashboard & analytics.

3.2.4.2.2 Use Case Diagram

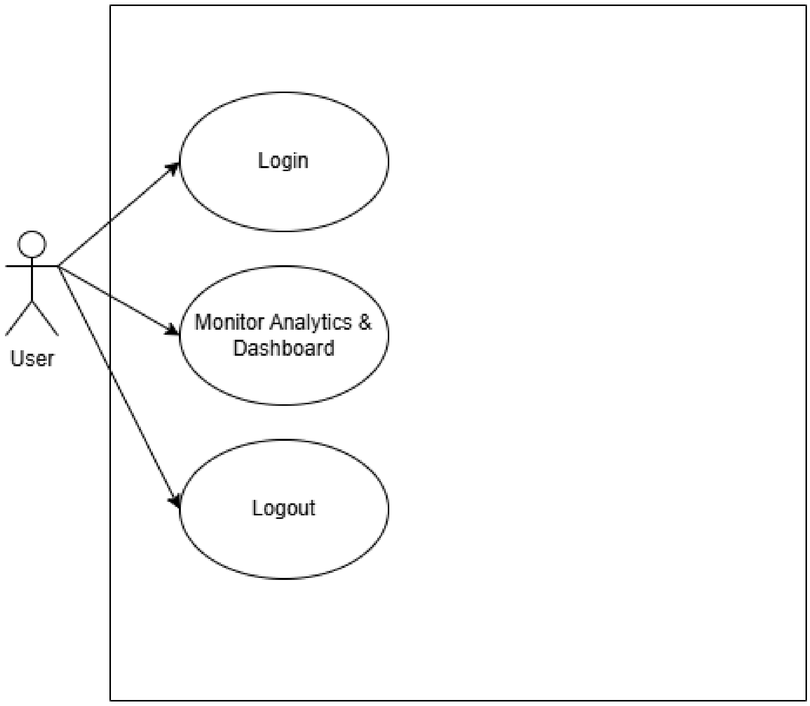

Figure ??? Use Case Diagram: Development of the Vertical Farming Monitoring System

3.2.4.3 System Development

The system will be developed using a variety of programming languages. The developers will incorporate diverse features to ensure the system's functionality and performance. Particularly, the front end, back end, and hardware will be implemented distinctively. To ensure the development of the system thoroughly, it must connect:

- Frontend will be using Laravel
- Backend will be using Laravel
- Hardware will be using C++

3.2.4.4 Testing

3.2.5 Evaluation