VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Multidisciplinary Project (CO3111)

Final Report

Indoor Smart Farm

Supervisor: Lê Thị Bảo Thu

Students: Trần Đình Đăng Khoa - 2211649
Vũ Trịnh Thanh Bình - 2252085
Trương Tuấn Kiệt - 2252411
Lê Trung Kiên - 2252394
Lê Hồng Anh Hoàng - 2211078
Nguyễn Thủy Tiên - 2252806

HO CHI MINH CITY, May 2025

Contents

Work Distribution

Name	ID	Tasks	Contribution
Trần Đình Đăng Khoa	2211649	- Frontend - Authentication - Device Log - Write and edit report	100%
Vũ Trịnh Thanh Bình	2252085	- Hardware design - Sending sensor data from the device to the IoT server - Write and edit report	100%
Trương Tuấn Kiệt	2252411	- Frontend - Dashboard - Visualization	100%
Lê Trung Kiên	2252394	- Frontend - Dashboard - Visualization	100%
Lê Hồng Anh Hoàng	2211078	- Backend - Database design - Connect database to IoT server	100%
Nguyễn Thủy Tiên	2252806	- UI design - Frontend - Write and edit report	100%

1 Introduction

This Indoor Smart Farm system is designed to automate and optimize indoor cultivation by continuously monitoring environmental conditions and controlling essential farming devices. Utilizing smart sensors and automation technologies, the system aims to enhance plant growth, reduce water and energy consumption, and minimize the need for manual labor.

Key features of the system include automatic control of lighting and irrigation, real-time monitoring of soil moisture and light intensity, and a user-friendly interface for system management and customization. By automating these critical processes, the system ensures ideal growing conditions, improves crop health, and reduces dependence on constant human supervision — contributing to a more efficient and sustainable farming method.

The hardware components of the system include a water pump motor, LED grow lights, soil moisture sensors, and light sensors, all integrated to create a responsive and intelligent farming environment.

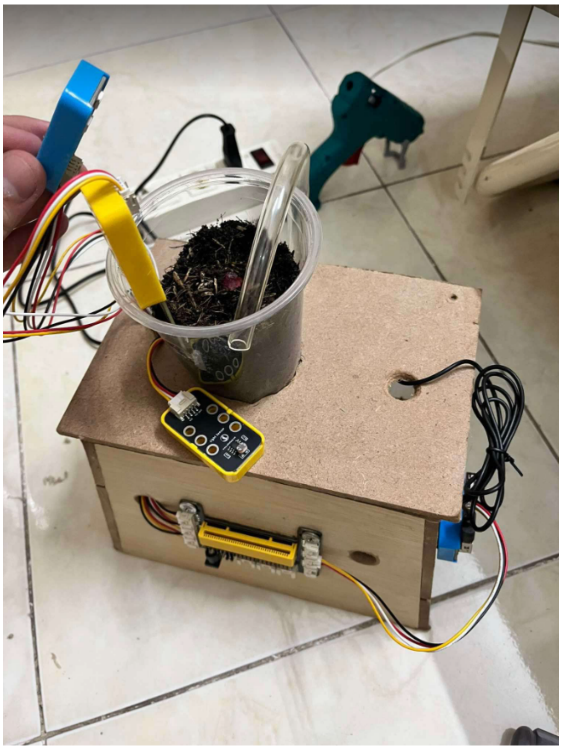


Figure 1: A sample model

2 Requirements

2.1 Functional Requirements

2.1.1 Internet of Things System

- **Soil Moisture Monitoring:** Continuously measures soil moisture levels and reports them back to the system.

- **Light Intensity Monitoring:** Monitors light levels and activates LED lighting when the intensity is below 2000 lux (threshold can be adjusted).
- **Automated Device Activation:** Activates devices when certain conditions are met (e.g., activate the water pump when soil moisture is below 30%).

2.1.2 Application

- **User Interface:**
  - Users can manually operate devices and create automation settings (e.g., turn on irrigation when soil moisture is below 30
  - Users can monitor real-time sensor data via an intuitive web-based application. The interface will include a dashboard, system status indicators, and graphical depictions of soil moisture and light intensity levels.
- **Security:** Allows users to log in, and log out securely.
- **Data:** Processes and stores all sensor data. Users can access and view all stored data.

2.2 Non-functional Requirements

2.2.1 Internet of Things System

- **Reliability:** The system should operate correctly 99% of the time.
- **Performance:**
  - User inputs are processed within 2 seconds.
  - The water pump/LED activates within 1 second of a threshold breach.
  - Sensor data refreshes every 1 minute.

2.2.2 Application

- **Usability:**
  - The UI shall be intuitive, with clear visual representations of farm conditions (moisture and light levels).
  - Alerts and notifications shall be easy to understand.
- **Security:** The system must include basic user authentication to ensure that only authorized users can access it.
- **Data Storage:** Stores sensor data locally for 24 hours (using an SD card or onboard memory).

3 Devices

3.1 Soil Moisture Sensor

- **Application**: Measures water level within the soil. This is the statistic that most directly affects irrigation decisions.
- **Input**: Soil moisture - the sensor is planted underground.
- **Output**: Local soil moisture, reported to the server as above.

3.2 Light Sensor

- **Application**: Measure sunshine intensity and detect daylight. Sunshine intensity also affects whether to irrigate or not. Daylight detection on the other hand dictate whether to turn grow lights on or off.
- **Input**: Light level - the sensor is placed outside, in exposure to the surrounding environment settings.
- **Output**: Processed by the controller to return sunshine level and additionally a grow light control signal when the light level dips below a set threshold.

3.3 Mini Pump

- **Application**: Pumps water to irrigate the farm
- **Input**: None (runs as long as there is electrical power).
- **Output**: None (electronically). Moves fluids.

3.4 LED

- **Application**: Turns on automatically when the light is below the threshold
- **Input**: Light sensor readings processed by a microcontroller to trigger LED control signals.
- **Output**: RGB LEDs light up with predefined colors or brightness levels when conditions are dark.

3.5 Microbit

- The hub for all sensors. It receives sensor readings through the expansion circuit board and perform preliminary processing (such as unit conversion, threshold and the like).
- Pre-processed data are then sent to the server for logging.

3.6 Expansion Circuit Board

-The backbone for Microbit and other sensors.

4 Use Case Details

4.1 Use Case Diagram

Use Case Diagram for Indoor Smart Farm System

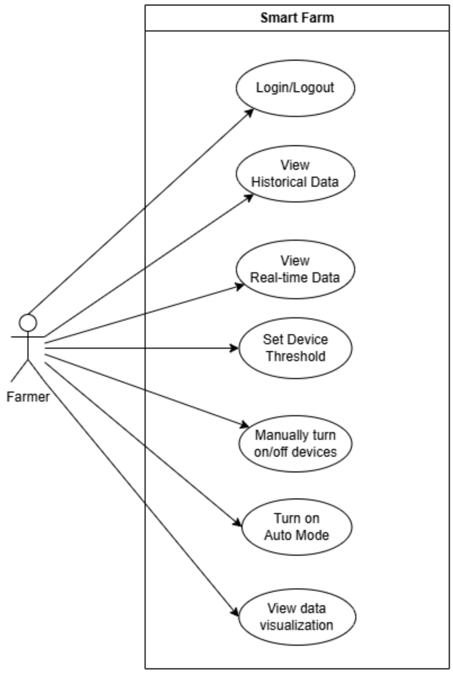


Figure 2: Use Case Diagram for Indoor Smart Farm System

4.2 Use Case Descriptions

Use Case: View Real-Time Data

Use Case Name	View Real-Time Data
Actor	Farmer
Description	The farmer views real-time soil moisture and light intensity data via the app.
Preconditions	Farmer is logged in, system is operational, sensors are active.
Normal Flow	1. Farmer logs in. 2. Selects "Dashboard". 3. System displays current data.
Postconditions	Farmer sees real-time soil moisture and light levels on the dashboard.
Alternative Flow	If no data is available, the system displays "No data available."

Use Case: View Data Visualization

Use Case Name	View Data Visualization
Actor	Farmer
Description	The farmer views summary data through graphs in the Dashboard.
Preconditions	Farmer is logged in, system is operational.
Normal Flow	1. Farmer logs in. 2. Selects "Dashboard". 3. Selects date of data to summary. 4. System displays average data for each 2 hours in that date.
Postconditions	Farmer sees summary data through graph on the dashboard.
Alternative Flow	If no data is available, the system displays graph with no data.

Use Case: View Historical Data

Use Case Name	View Historical Data
Actor	Farmer
Description	The farmer can view historical data in.
Preconditions	Farmer is logged in, historical data is stored.
Normal Flow	1. Farmer logs in. 2. Selects "Device log" in the sidebar. 3. System display the devices log in a table format.
Postconditions	Farmer can view a table of historical data.
Alternative Flow	If no data is stored, the system displays "No logs found."

Use Case: Set Device Threshold

Use Case Name	Set Device Threshold
Actor	Farmer
Description	The farmer sets and saves thresholds for automated device activation (e.g., soil moisture $< 30\%$).
Preconditions	Farmer is logged in, system is operational.
Normal Flow	1. Farmer logs in. 2. Navigates to dashboard. 3. Sets threshold. 4. System saves.
Postconditions	Thresholds are updated, devices activate based on settings.
Alternative Flow	If save fails, system retries and notifies the user.

Use Case: Manually turn on/off devices

Use Case Name	Manually turn on/off devices
Actor	Farmer
Description	The farmer turn on/off devices manually.
Preconditions	Farmer is logged in, system is operational, auto mode is off.
Normal Flow	1. Farmer logs in. 2. Navigates to dashboard. 3. Turns the device on/off.
Postconditions	Device is turned on/off.
Alternative Flow	If auto mode is on, the status of device will be automatically updated despite the farmer turn on/off.

Use Case: Turn on Auto Mode

Use Case Name	Turn on Auto Mode
Actor	Farmer
Description	The farmer turn on Auto Mode and system will automate device activation base on settings.
Preconditions	Farmer is logged in, system is operational.
Normal Flow	1. Farmer logs in. 2. Navigates to dashboard. 3. Turns Auto Mode on. 4. System saves.
Postconditions	Auto Mode is turned on, devices activate based on settings.
Alternative Flow	None

Use Case: Log In / Log Out

Use Case Name	Log In / Log Out
Actor	Farmer
Description	The farmer accesses the system by logging in with credentials and exits the system by logging out.
Preconditions	For Login: Farmer has a registered account. For Logout: Farmer is logged in.
Normal Flow	**Login:** 1. Farmer enters username and password. 2. System verifies the credentials. 3. Access is granted. **Logout:** 4. Farmer selects the logout option. 5. System logs the farmer out.
Postconditions	After Login: Farmer can access system features. After Logout: Farmer is returned to the login screen.
Alternative Flow	**Login Failure:** - If credentials are incorrect, the system displays an error message.

5 Mockup

5.1 Authentication page

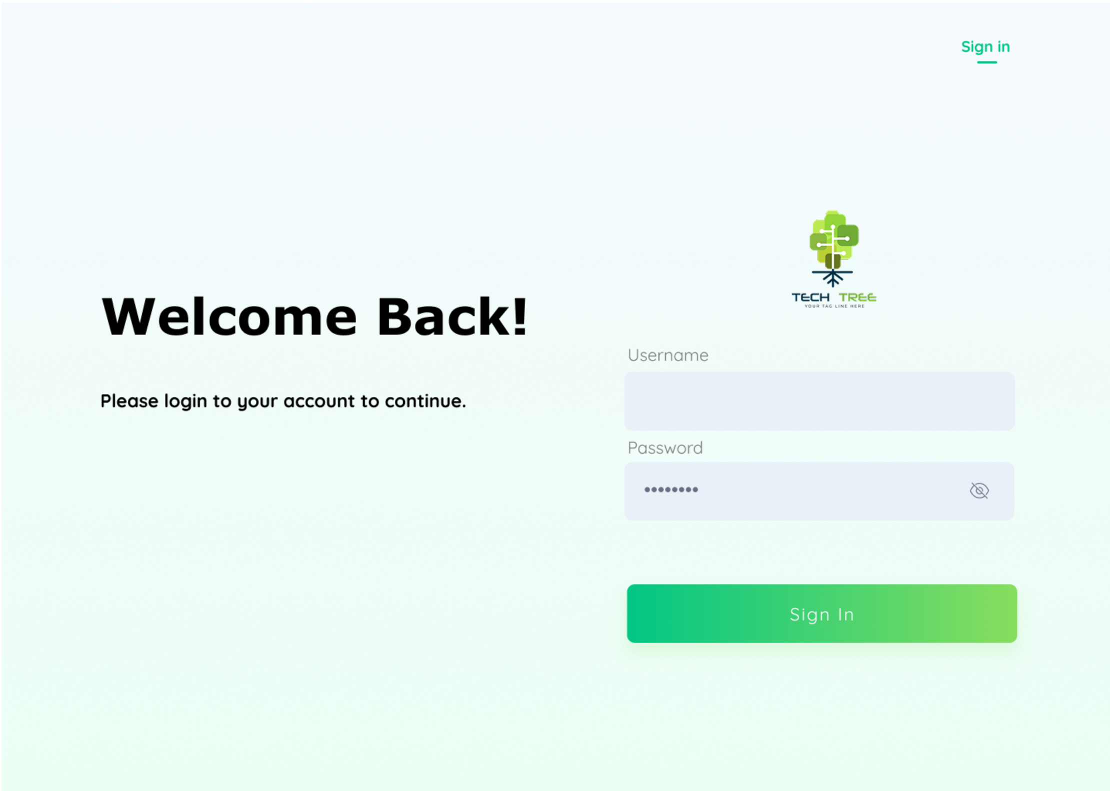


Figure 3: Authentication page

In this page, user will input **username** and **password** to continue.

5.2 Dashboard page

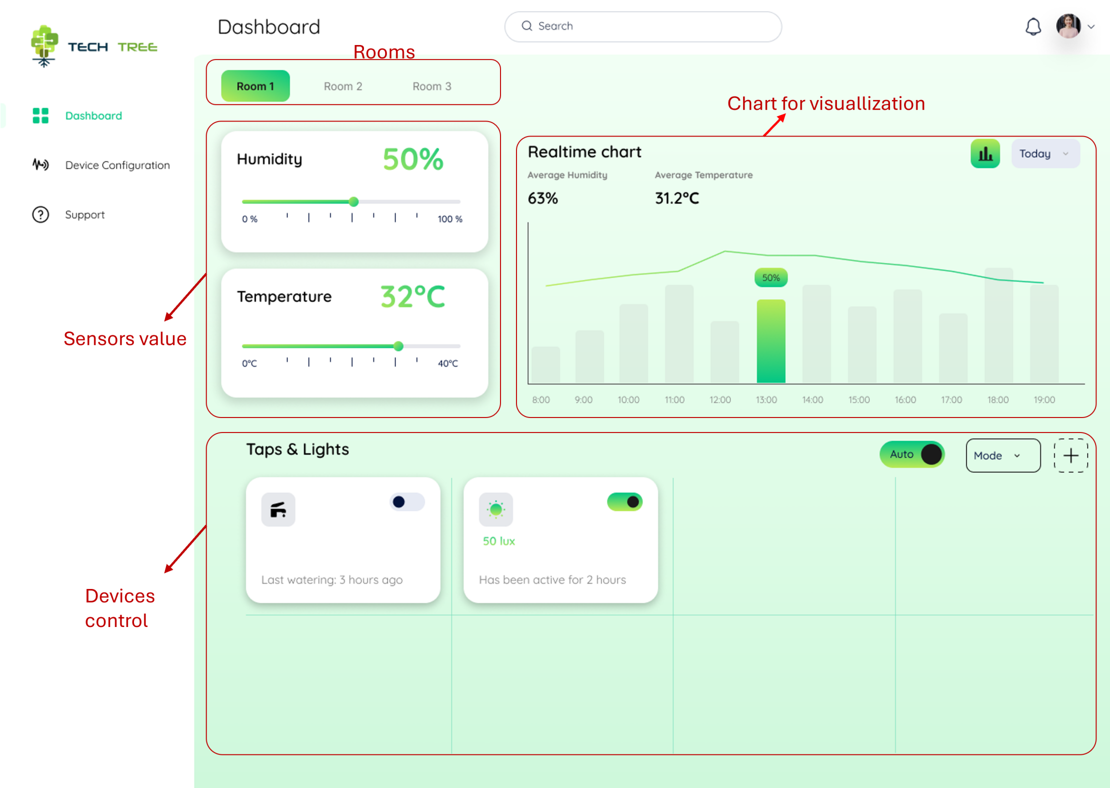


Figure 4: Caption

This page will be the main desktop for user. Including:

- **Rooms**: different work sheets for different room
- **Sensors value**: real-time value of sensors will be displayed here.
- **Chart**: charts for visualization will be drawn here.
- **Devices control**: Add and control devices.

5.3 Device Configuration page

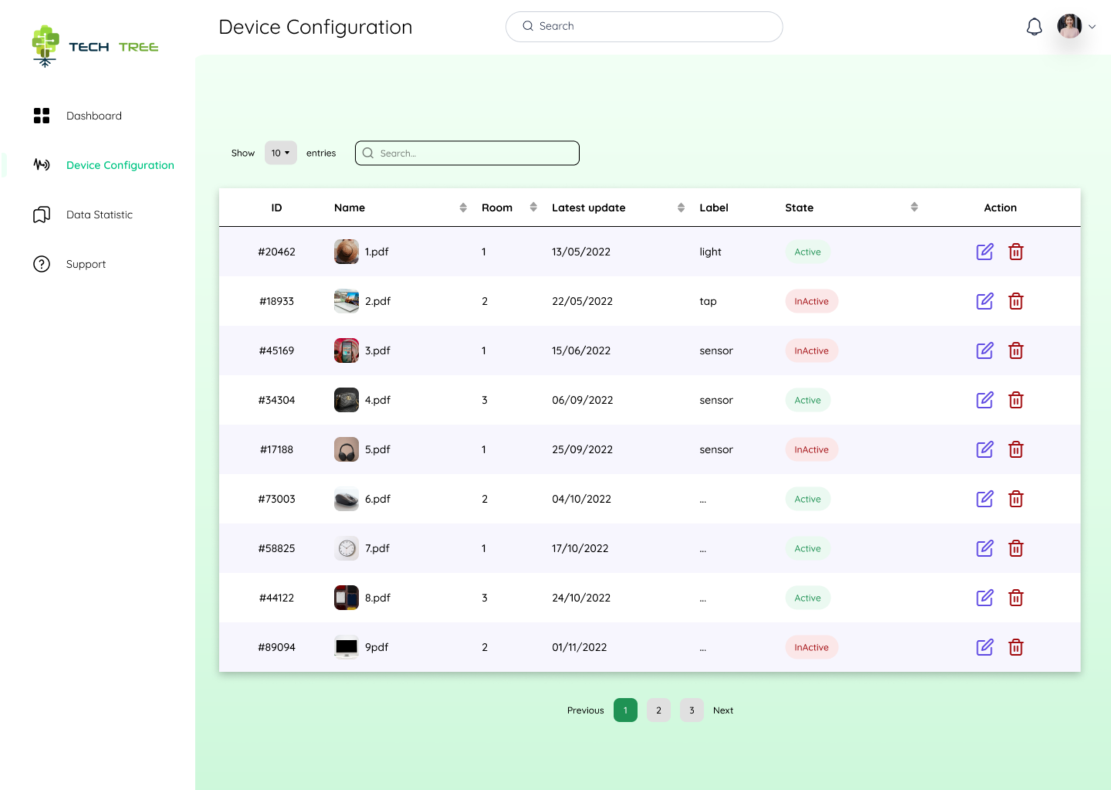


Figure 5: Devices configuration page

List of devices and actions to it will be displayed in this page in a table format.

6 General Design

The Indoor Smart Farm system is designed as a modular and layered architecture combining both hardware and software components. The system comprises **three main layers**: **the IoT Device Layer**, **the Server & Database Layer**, and **the User Interface Layer**. These layers interact through defined communication protocols to ensure seamless automation and user control.

6.1 IoT Device Layer

This layer consists of physical sensors and actuators connected to a microcontroller (Microbit), which acts as the control hub. The main components include:

- **Soil Moisture Sensor**: Continuously monitors the water level in the soil.
- **Light Sensor**: Detects ambient light intensity and triggers lighting systems when necessary.

- **Mini Pump**: Waters the soil based on moisture readings.
- **LED Grow Lights**: Automatically turn on when natural light is insufficient.
- **Microbit and Expansion Board**: Responsible for collecting sensor data, executing automation rules, and transmitting processed information to the server.

6.2 Server and Database Layer

This layer serves as the backend of the system and is responsible for:

- **Data Handling**: Collecting and storing environmental data (e.g., soil moisture, light intensity) and user actions.
- **Automation Logic**: Defining thresholds (e.g., soil moisture $< 30\%$) and triggering actions accordingly.
- **User Management**: Handling user accounts, authentication, and session control.
- **Database Structure**: Designed using an ERD-to-relational schema transformation. It includes tables for Farmer, Sensor, Output Device, Action Logs, and Environmental Conditions, with appropriate foreign key constraints.

6.3 User Interface Layer

A web-based dashboard is provided for users to interact with the system. Key features include:

- **Authentication Page**: Secure login interface for registered users.
- **Real-time Monitoring**: Live visualization of sensor data using graphs and indicators.
- **Device Control**: Manual toggling of water pumps and lights.
- **Automation Configuration**: Allows users to set thresholds for automatic device activation.

6.4 Communication and Data Flow

Sensor readings are collected every 2 seconds and transmitted to the server via wireless communication (Wi-Fi). The server processes incoming data and updates the database. The frontend periodically polls the server to fetch updated data, ensuring the user interface reflects the real-time status of the farm.

6.5 Automation Logic

Automation rules are evaluated on the server side. For example, if the moisture level falls below 30%, the server issues a command to activate the mini pump. Similarly, if light intensity drops below a configured threshold, the LED light is turned on automatically.

7 Database Overview

7.1 Model Description

The model consists of three main entities: **Farmer** (also referred to as *User*), **Output_Device**, and **Sensor** (serving as the input device).

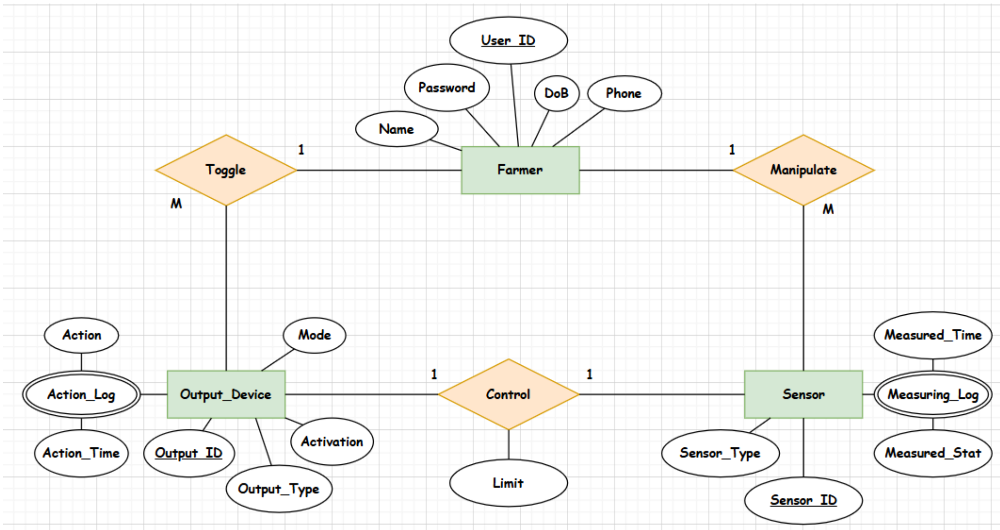


Figure 6: Entity-Relationship Diagram of the system

Farmer

This entity has the following attributes:

- **User_ID**: Primary key of the entity.
- **Name**: Full name of the current user.
- **Password**: Used for authentication; each user logs in with their `User_ID` and `Password`.
- **Phone**: The user's phone number.
- **DoB**: Date of birth of the user.

Sensor

This entity includes:

- **Sensor_ID**: Primary key of the entity.
- **Sensor_Type**: Indicates whether the sensor is for light or soil moisture (e.g., "light" or "soil moisture").
- **Environment_Condition** (multivalued attribute), containing:
  - **Measured_Time**: Timestamp of the recorded measurement.
  - **Measured_Stat**: Numeric value ranging from 0 to 100. Measurement units (Lux for light, % for soil moisture) are managed externally from the database.

Output_Device

This entity is structurally similar to `Sensor`, with the following attributes:

- **Output_ID**: Primary key of the entity.
- **Output_Type**: "water pump" or "light".
- **Activation**: Indicates whether the device is currently activated.
- **Mode**: Indicates if the device operates in automatic or manual mode.
- **Action_Log** (multivalued attribute), including:
  - **Action_Time**: Timestamp when the action occurred.
  - **Action**: Either "enable" or "disable". A constraint is enforced to prevent two consecutive identical actions.

Relationships

- **Control**: One-to-one relationship where a `Sensor` controls an `Output_Device` of the same type. Includes attribute:
  - **Limit**: Threshold value used to trigger the control action.
- **Toggle**: One-to-many relationship where a `Farmer` can turn on/off their `Output_Devices`.
- **Manipulate**: One-to-many relationship where a `Farmer` is associated with multiple `Sensors`.

7.2 EER Mapping

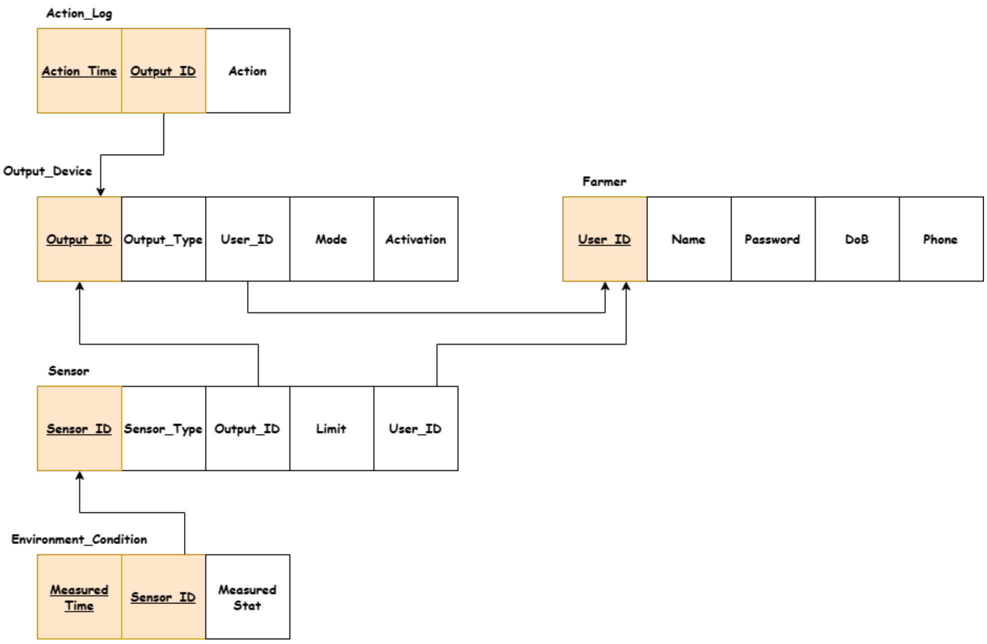


Figure 7: Relational Schema derived from EER model

- **Farmer** is mapped to a relation with attributes: `User_ID`, `Name`, `Password`, `DoB`, and `Phone`.
- **Sensor** is represented using two relations:
  - `Sensor (Sensor_ID, Sensor_Type, Output_ID, Limit, User_ID)`: `Output_ID` and `Limit` are included from the `Control` relationship (due to its 1:1 nature). `User_ID` is a foreign key referring to the associated `Farmer`.
  - `Environment_Condition(Measured_Time, Sensor_ID, Measured_Stat)`: A multi-valued attribute relation where `Sensor_ID` is a foreign key referencing `Sensor`. The combination of `Measured_Time` and `Sensor_ID` forms the super key.
- **Output_Device** is also mapped into two relations:
  - `Output_Device (Output_ID, Output_Type, User_ID, Mode, Activation)`: `User_ID` is a foreign key referencing `Farmer`.
  - `Action_Log(Action_Time, Output_ID, Action)`: `Output_ID` is a foreign key referencing `Output_Device`. The super key is the pair `(Action_Time, Output_ID)`.

8 Final Product

8.1 Authentication page

Our system is an internal system, the administrator will create the account for the user. Therefore, there is only login option.

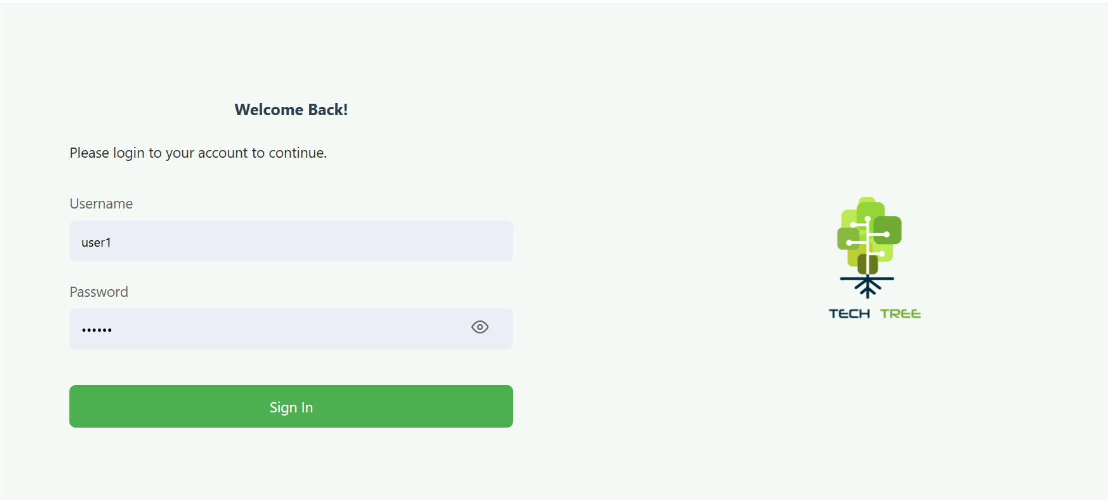


Figure 8: Login page

Users will enter `username` and `password` to authenticate and continue.

8.2 Sidebar

User can navigate between dashboard and device log page using the sidebar in the left of the screen.

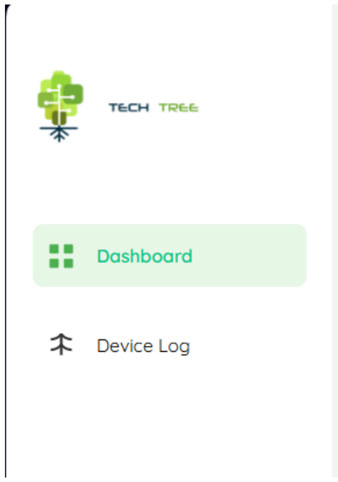


Figure 9: Sidebar for navigation

8.3 Dashboard page

User can view the entire real-time report of the system here.

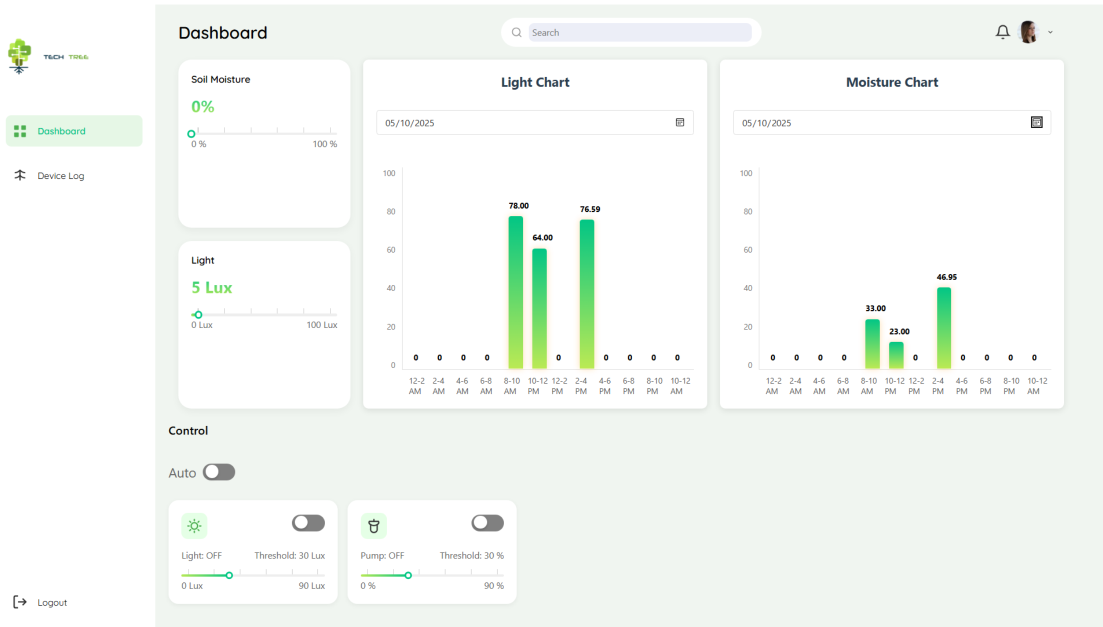


Figure 10: Dashboard overview

In this page, there are 3 sections:

1. **Sensors real-time value**

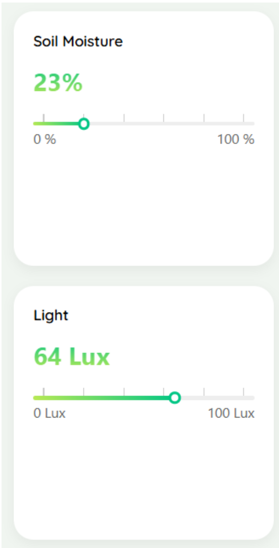


Figure 11: Real-time value from sensors

There are 2 type of sensors:

- **Soil moisture**: value in range from 0 to 100 percent(%).
- **Illuminance**: value in range from 0 to 100 lux.

2. **Output devices control**

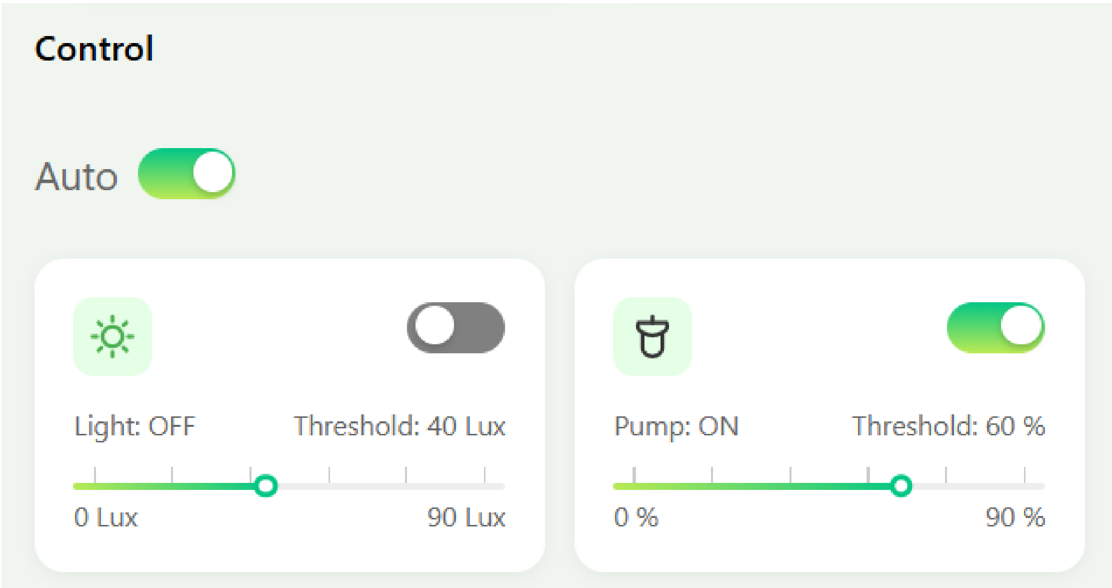


Figure 12: Output devices control section

In this section, we have:

- **Control mode**: user can choose between auto and manual mode to control the devices. When `Auto` is turned on, the device will turn on/off automatically base on the sensors value.
- **Trigger**: For each device, user can manually turn on/off.
- **Threshold**: Users can drag the slider to adjust the threshold (step is fixed to 10) for `Auto` mode.

3. **Diagrammatically visualize the sensors values**

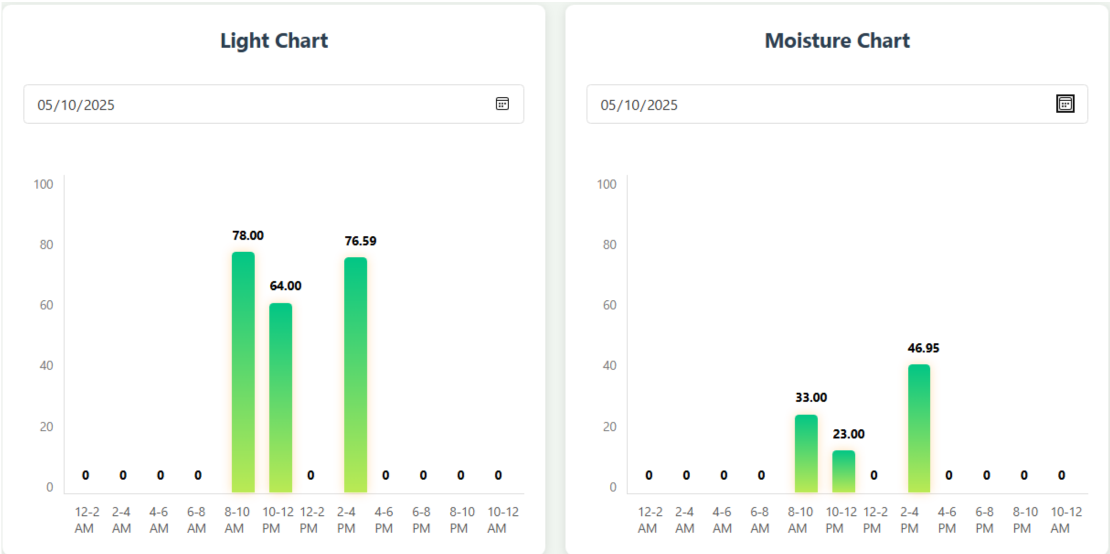


Figure 13: Value diagram of each sensors

For each sensors, user can:

- Choose date for the diagram report.

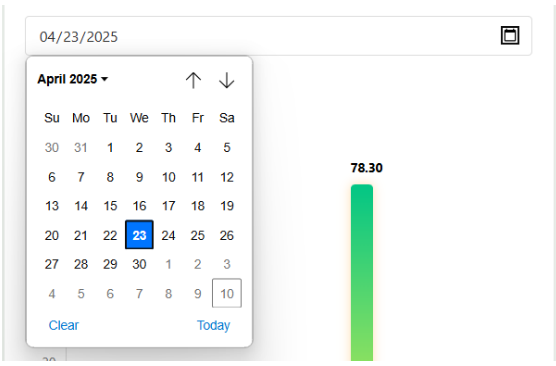


Figure 14: Choose date to display

- View the diagram of average value for each 2 hours in the day.

8.4 Devices log page

In this page, user can see the sensors and output devices log in a table format.

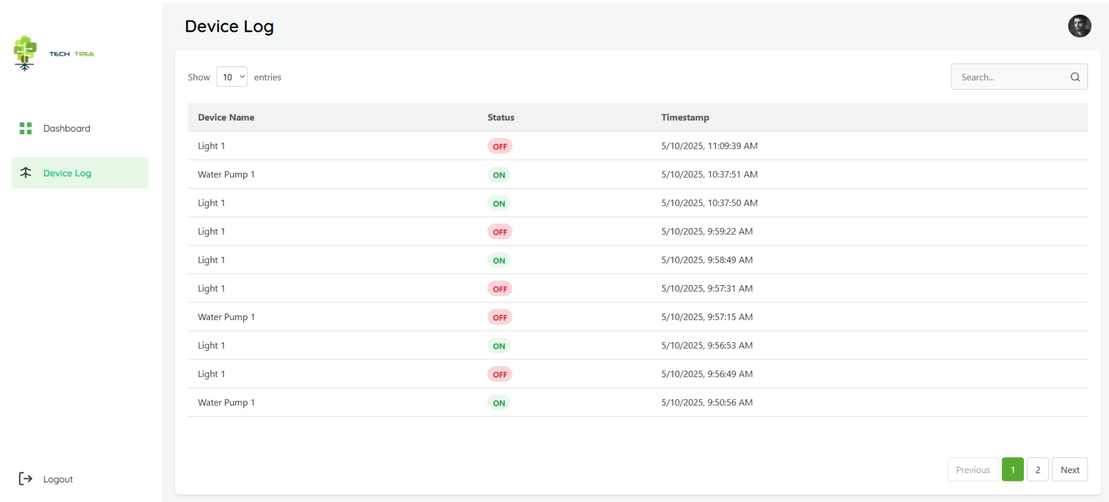


Figure 15: Device log table

- **Entries**: number of records to be shown.
- **Search**: user can search by:
    1. Device name:

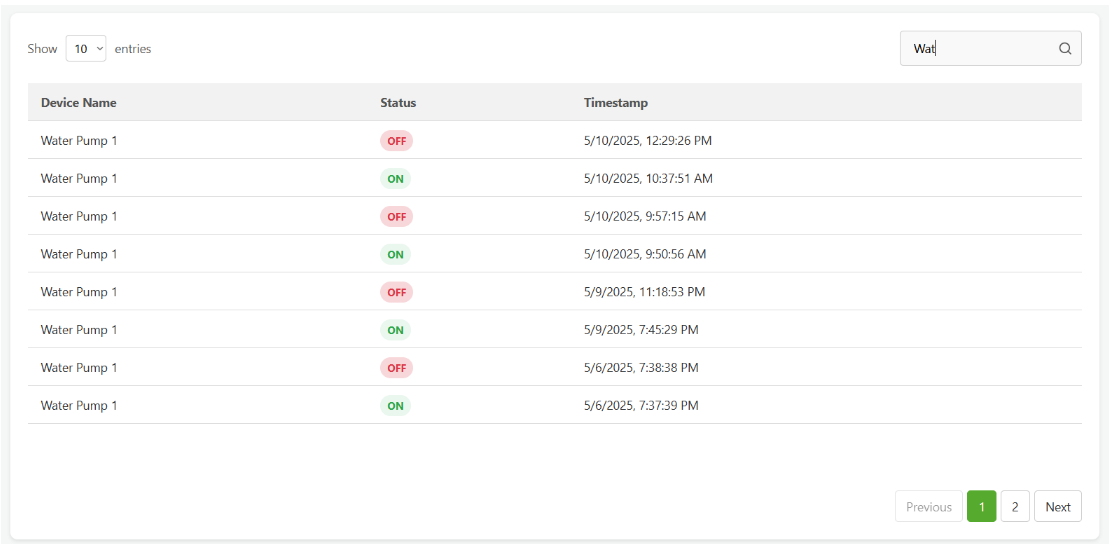
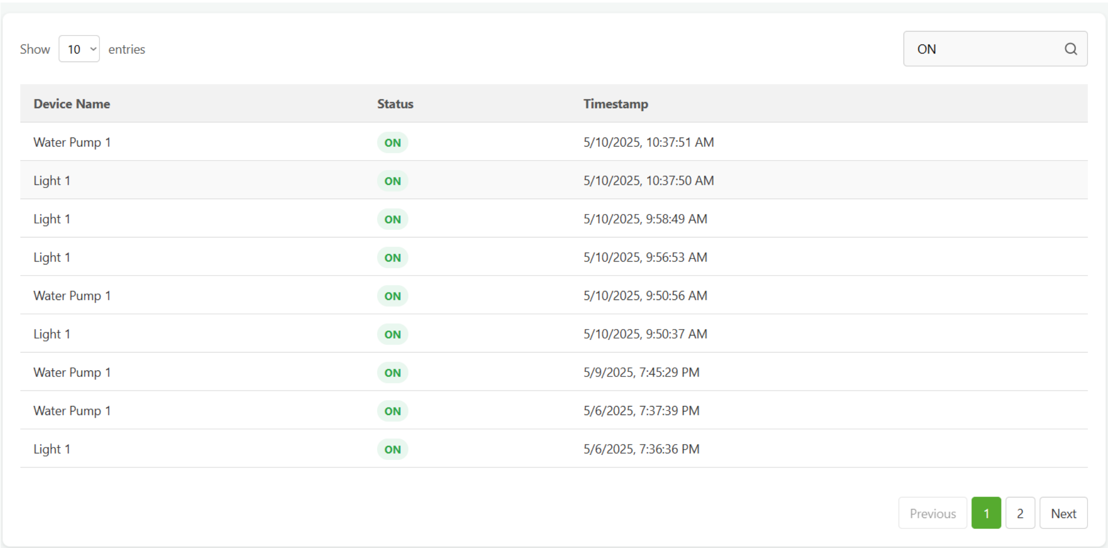


Figure 16: Search by device name

2. Status:


Show 10 entries

ON

Device Name	Status	Timestamp
Water Pump 1	ON	5/10/2025, 10:37:51 AM
Light 1	ON	5/10/2025, 10:37:50 AM
Light 1	ON	5/10/2025, 9:58:49 AM
Light 1	ON	5/10/2025, 9:56:53 AM
Water Pump 1	ON	5/10/2025, 9:50:56 AM
Light 1	ON	5/10/2025, 9:50:37 AM
Water Pump 1	ON	5/9/2025, 7:45:29 PM
Water Pump 1	ON	5/6/2025, 7:37:39 PM
Light 1	ON	5/6/2025, 7:36:36 PM

Previous 1 2 Next

Figure 17: Search by status

9 Techonologies

FrontEnd

- **Framework:** ReactJS
- **Routing:** React Router
- **Data Fetching:** Axios
- **Charting:** Recharts
- **Styling:** CSS
- **Icons:** React Icons, Lucide React

BackEnd

- **Framework:** Express.js
- **Database:** MongoDB
- **API Integrations:** Adafruit API
- **Deployment:** Vercel
- **Development Utilities:** Nodemon (for auto-reloading)

Version and Packages management

- **Version control & CI:** Git and GitHub Actions
- **Packages management:** npm

10 References

- Indoor Smart Farm Project Documentation, HackMD.
  https://hackmd.io/@lnu-iot/r1yEtcs55