Automated Indoor Nursery

Department of Electrical and Computer Engineering
University of Central Florida

Final Presentation
EEL4915 - Fall 2022
Group 10

Nicholas Leon - Electrical Engineer
Mariana Lozano - Electrical Engineer
Austen Ordos - Electrical Engineer
Hamzah Ullah - Computer Engineer

Overview

Product Description:

- ➔ An indoor plant system that provides enough water and sunlight to keep these herbs alive.
- ➔ This device monitors all the components needed to help these herbs grow and stay healthy.
- ➔ Indoor gardening allows for a stable food supply for consumers whether it's at home, a restaurant, or retailers.

Motivation:

- ➔ Provide ease of mind to customers so they can be worry free while they are away.

Project Objectives

➔ Our goal was to meet all of these objectives as well as provide functionality of our entire product.

Objective #	Description
01	Automatically provide water for the plants while the owner is away.
02	Automatically provide enough sunlight to keep herbs alive.
03	Include easy-to-use interface.
04	Users can view snippets of plants while they are away and decide if more or less water/sunlight is needed.
05	Ease of mind for the user while they are away.
06	Sensors that accurately provide levels and information about how the herbs are doing (pH, humidity, temperature, etc.)
07	Minimal latency displayed data on websites for accuracy.

Market Specifications

- The market requirements were important to lay out to give the product a more user friendly feel and to help us keep them in mind.

Market	Value
System must be of reasonable weight	<30 lbs
System must be of reasonable size	<10 cubic feet
System must moderate pH level	Approx. 1 degree of precision
System must hold an approximate voltage supply	Approx. 24 VDC
System must hold a reasonable power consumption	<400 Watts
System must treat segments of water to send to the herbs	2 Gallons
Software must display the necessary data	Water amount, light, pH levels, humidity levels
System shall be able to communicate with a device for controls and data and control system remotely	WiFi Module
Unit must not leak	N/A

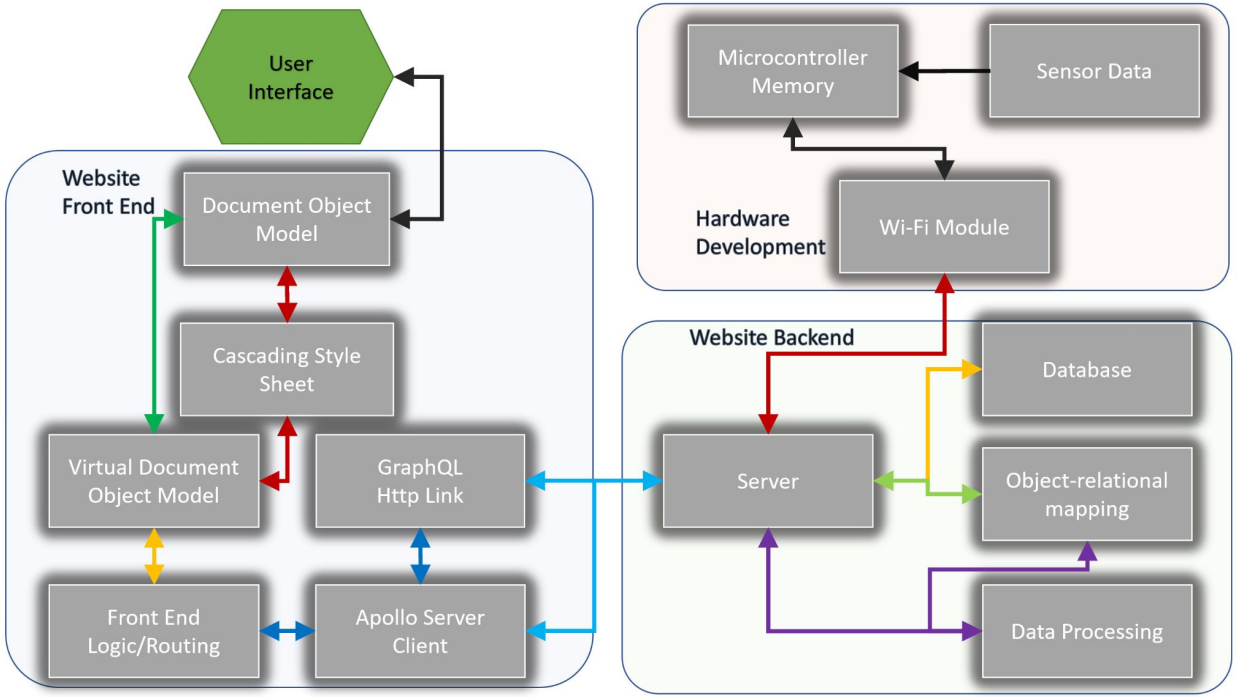
Engineered Specifications

- The engineering requirements were important to lay out because they were used to drive the project constraints.

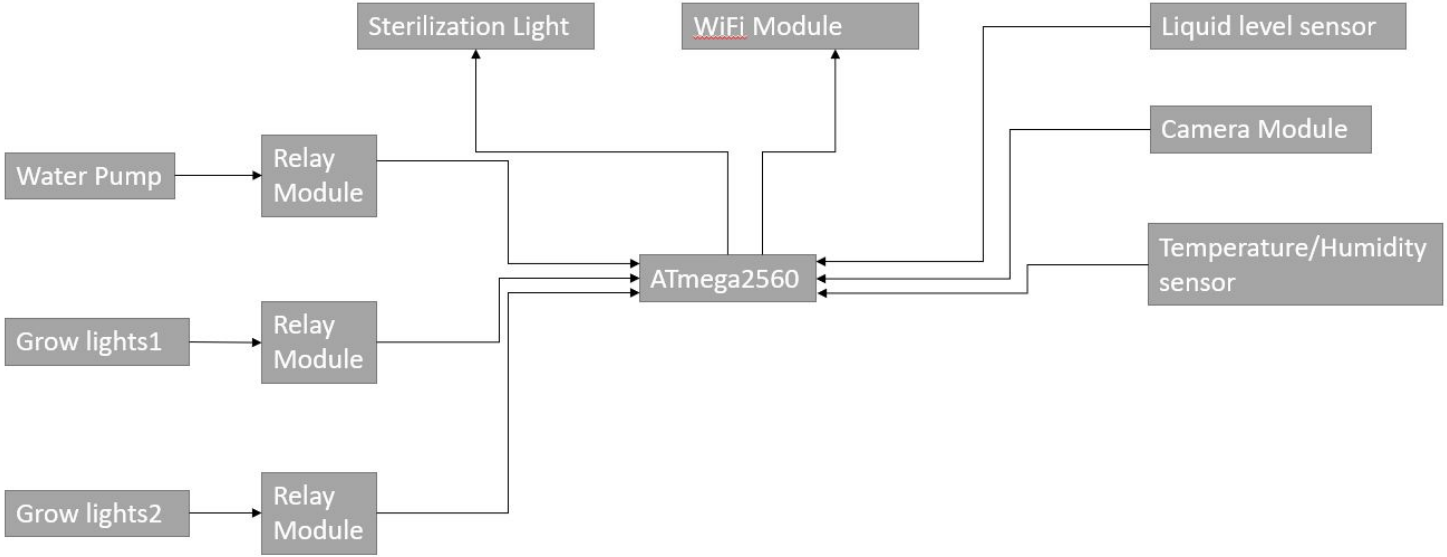
Engineering	Value
System must be able to change and detect pH levels	Approx. 1 degree of precision
System must measure and control ambient light	Approx. 3 degree of precision
Units must be able to communicate information back through the system	WiFi Module
System must control flow of water	1-2 Gallons per minute (GPM)
System will be able to monitor the quantity of water in the reservoir	Approx. 0.5 gallons of precision

Block Diagrams

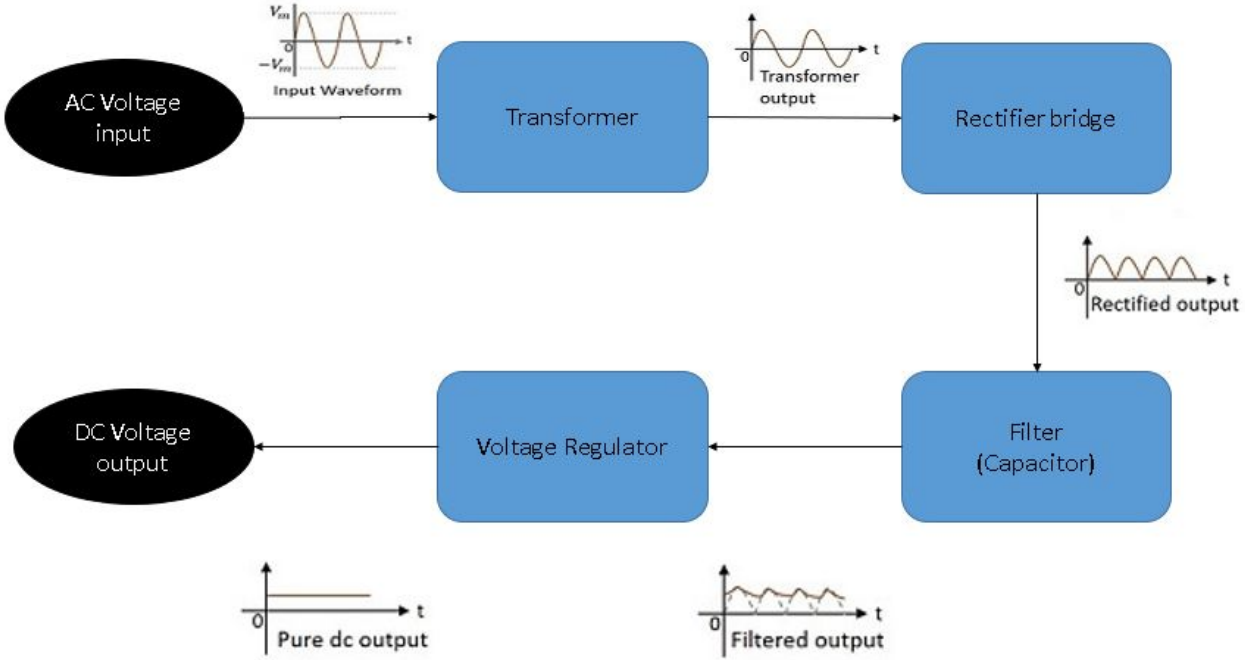
Software Block Diagram



Hardware Block Diagram



Power Block Diagram



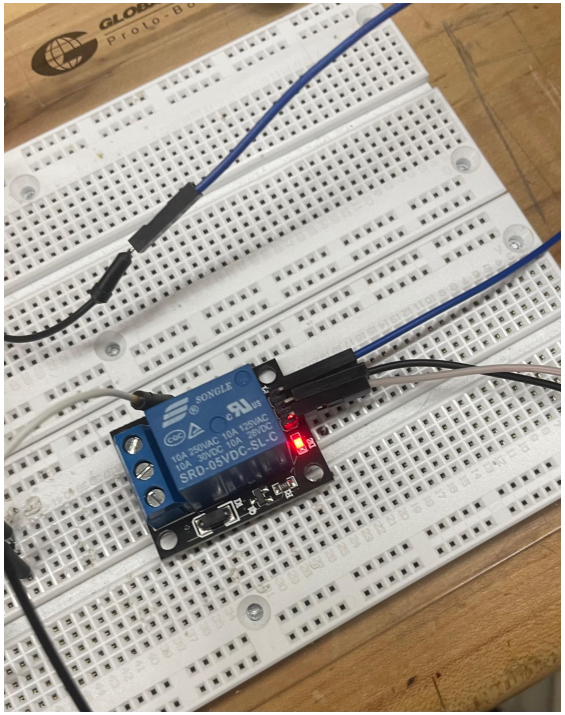
Enclosure

- ➔ PVC Pipe with elbows and T-fittings
- ➔ Corrugated Sheets
- ➔ Vinyl
- ➔ Drain Pan
- ➔ Drip Irrigation System
- ➔ Pots
- ➔ Tubing
- ➔ Printed Circuit Board

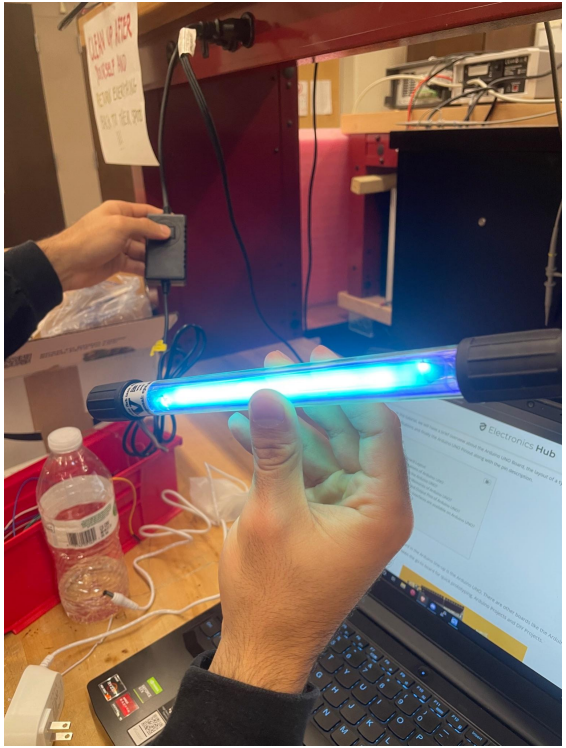
Hardware Design

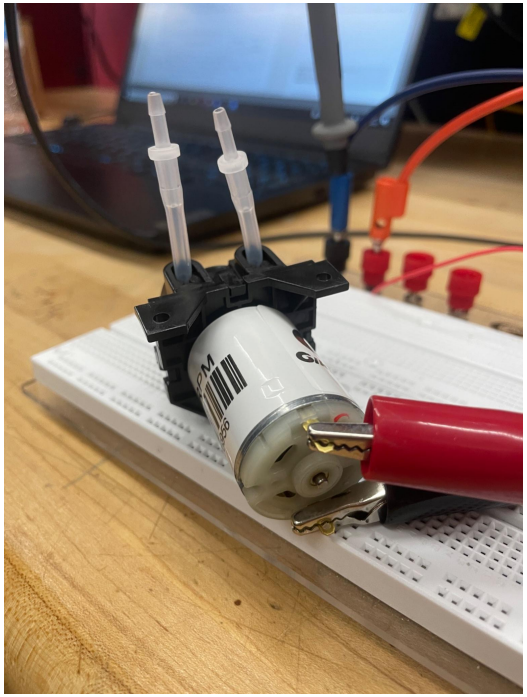
Hardware Testing

Relay Module



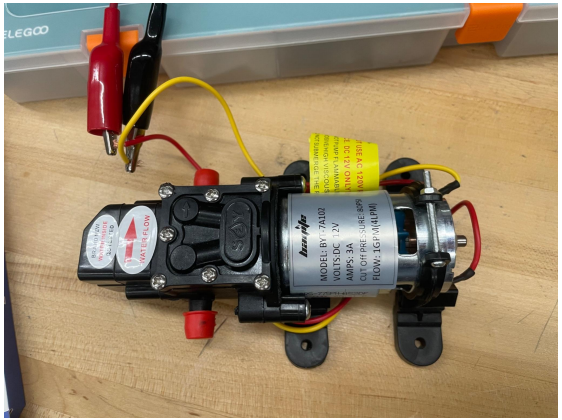
Green Killer





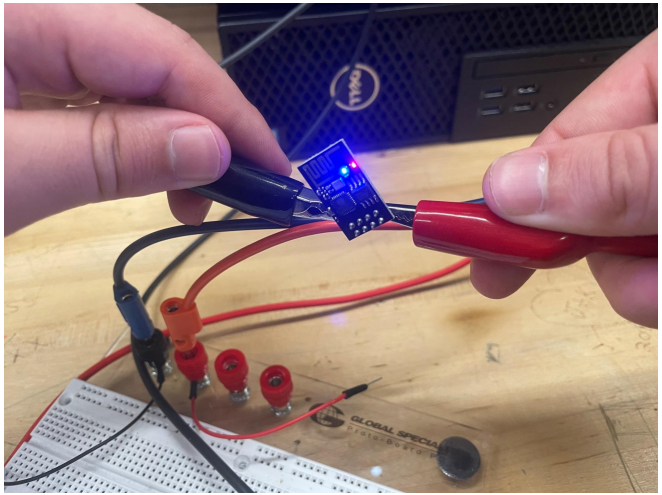
Peristaltic Pump

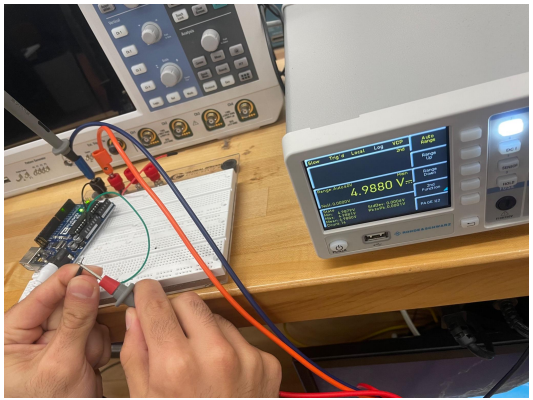
Hardware Testing (cont.)



Diaphragm Pump (12 V)

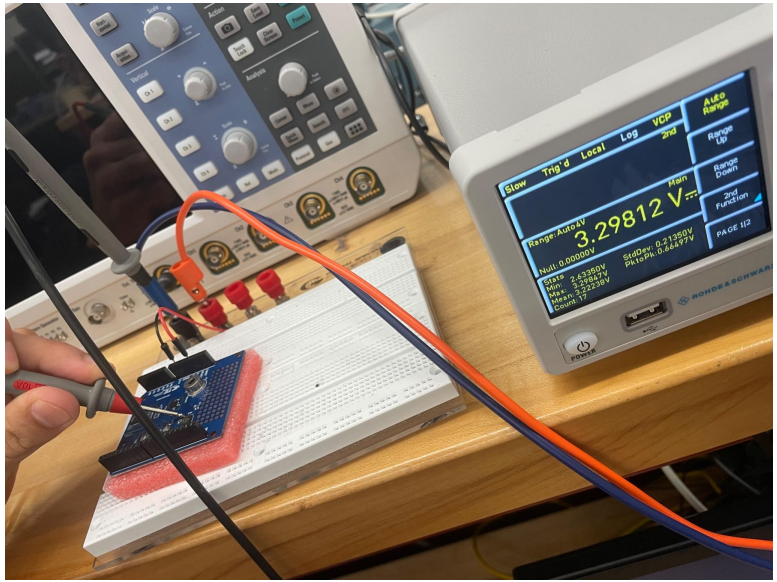
Wi-Fi Module (3.3 V)



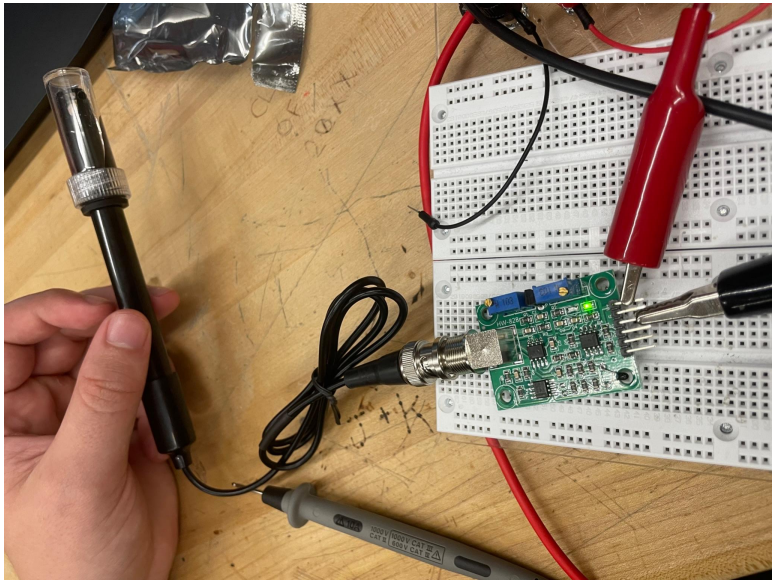


Arduino Uno R3 (shown working for 5 V also works for 3.3 V)

Hardware Testing (cont.)



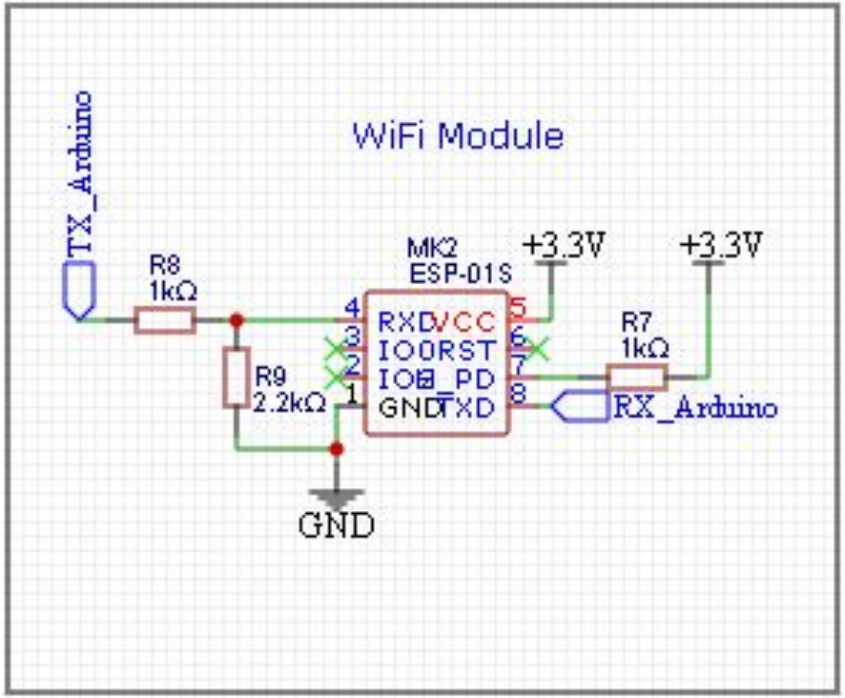
WeatherShield Arduino



pH Sensor (5 V)

PCB Schematic

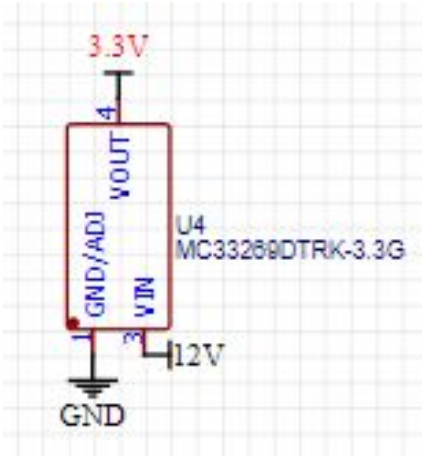
- WiFi module that establishes two-way communication between the ESP-01 and the device that is connected to it via Wi-Fi.

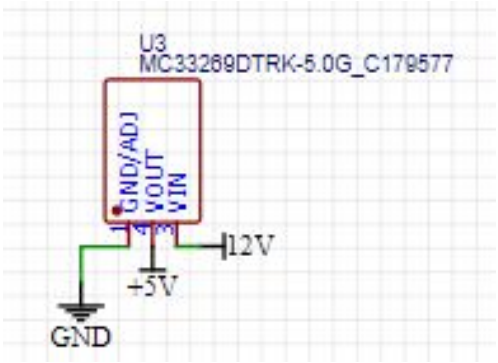


PCB Schematic

➔ Voltage regulators that converts +12V to +5V

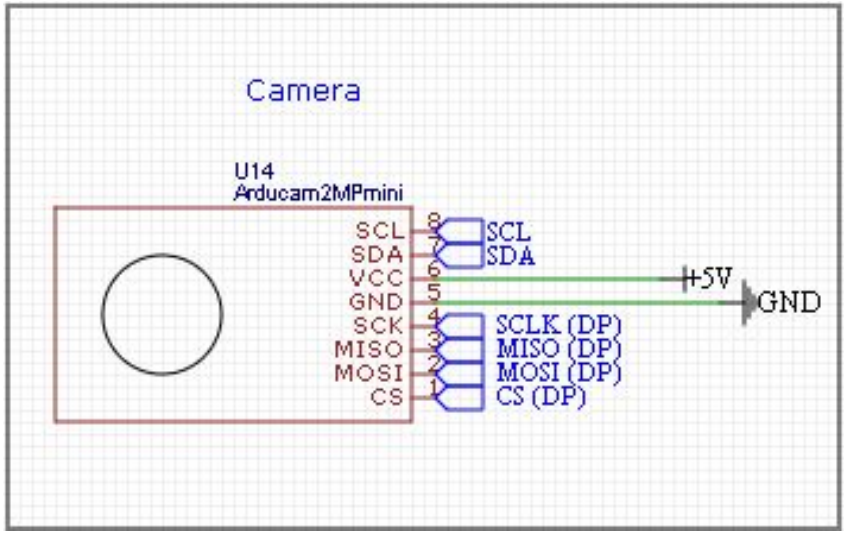
➔ Voltage regulator that converts +12V to +3.3V





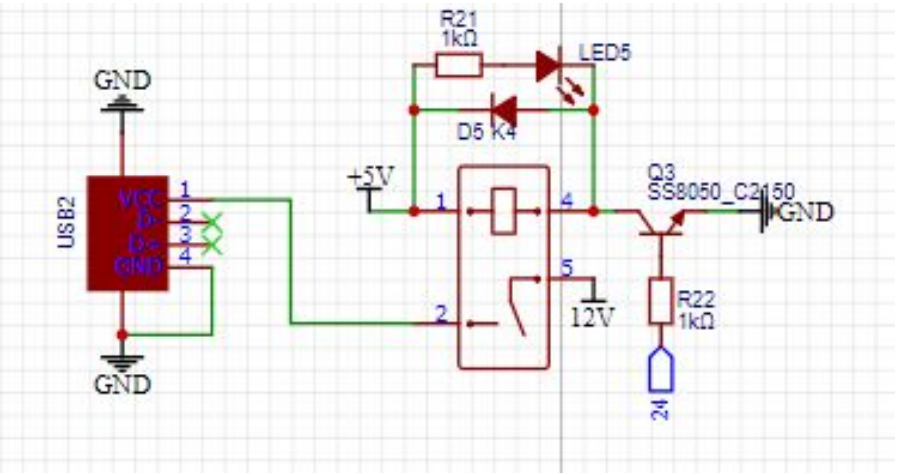
PCB Schematic

➔ The camera used to take pictures and livestream the enclosure to keep track of progress



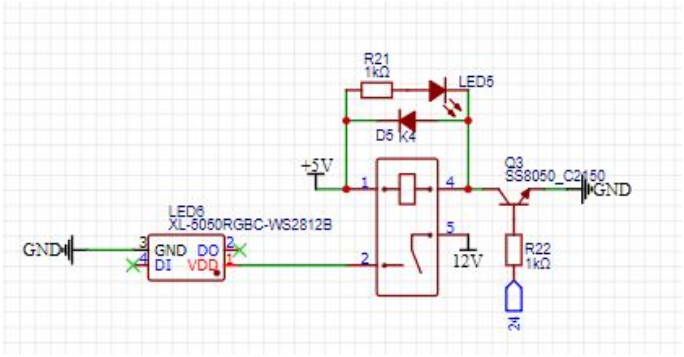
PCB Schematic

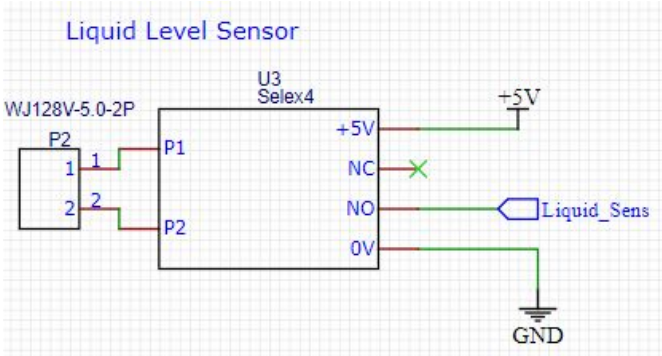
- 5 V pump operated via programmable relay that pumps into the drip irrigation system and back into the reservoir.

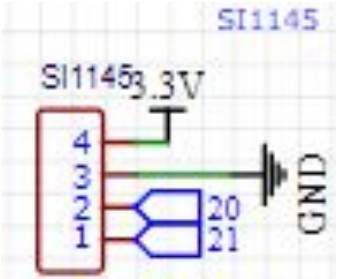


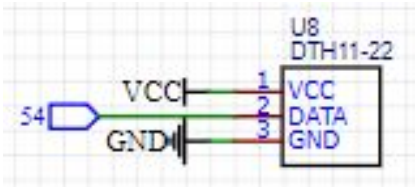
PCB Schematic

- ➔ 12 V Grow Lights operated via programmable relays (one each).
- ➔ Liquid Level Sensor operates via detecting on its probe (Liquid = 1 or No Liquid = 0)
- ➔ DTH11 detects Temperature and Humidity via 3.3 V
- ➔ Light Sensor detects visible and UV light



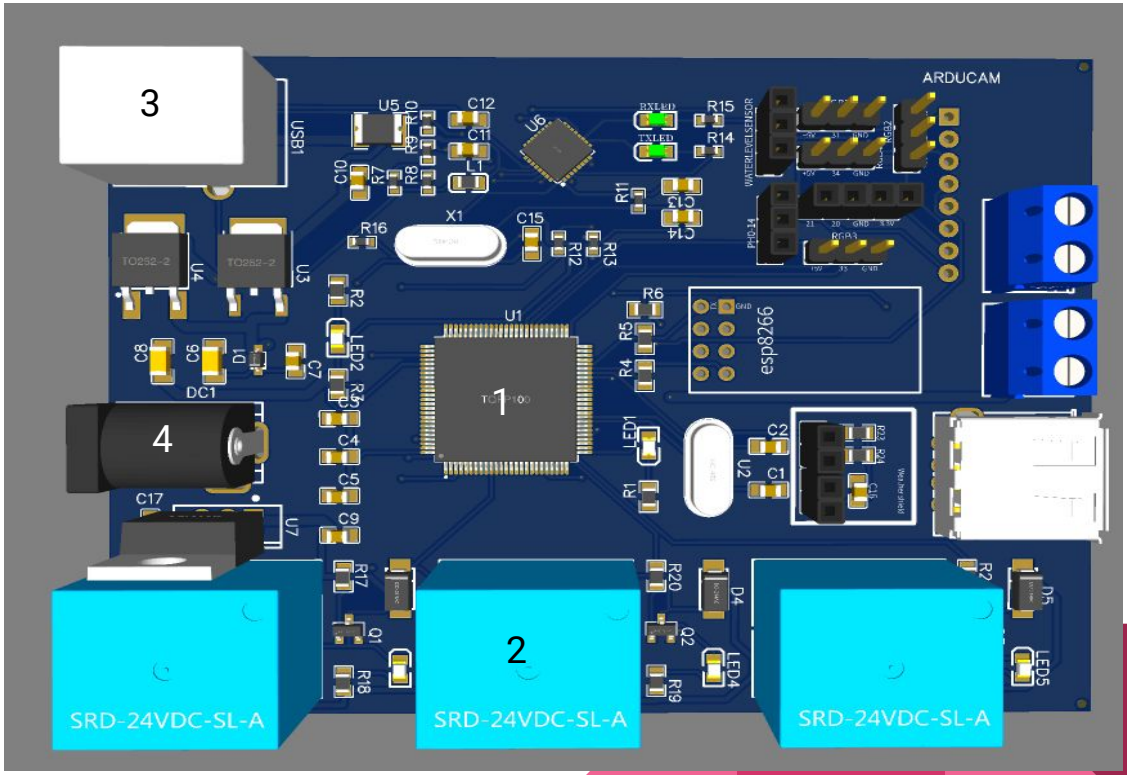






PCB Functionality

1. MCU - ATmega2560
2. Relays (via Uno R3)
3. USB type B - Flashing
4. Barrel plug



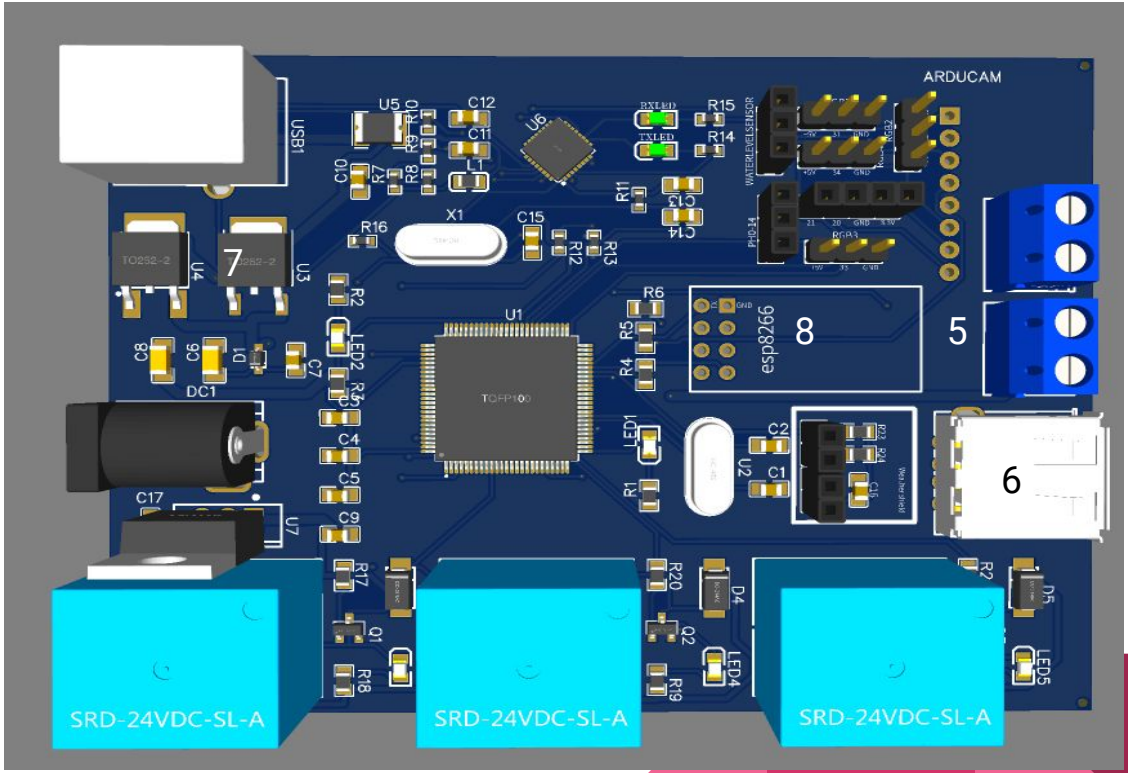
PCB Functionality

5. 12V UV Grow Light

6. 5V water pump

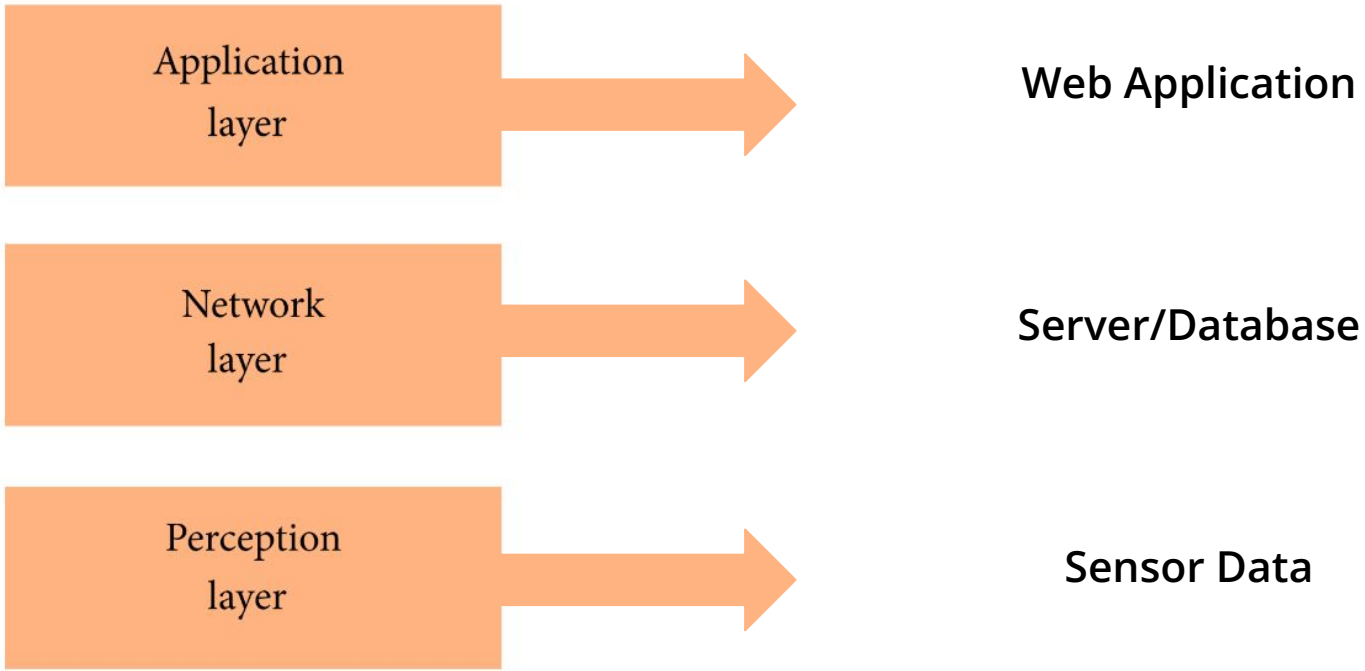
7. Voltage regulators

8. Wi-Fi module



Software Design

Software Design



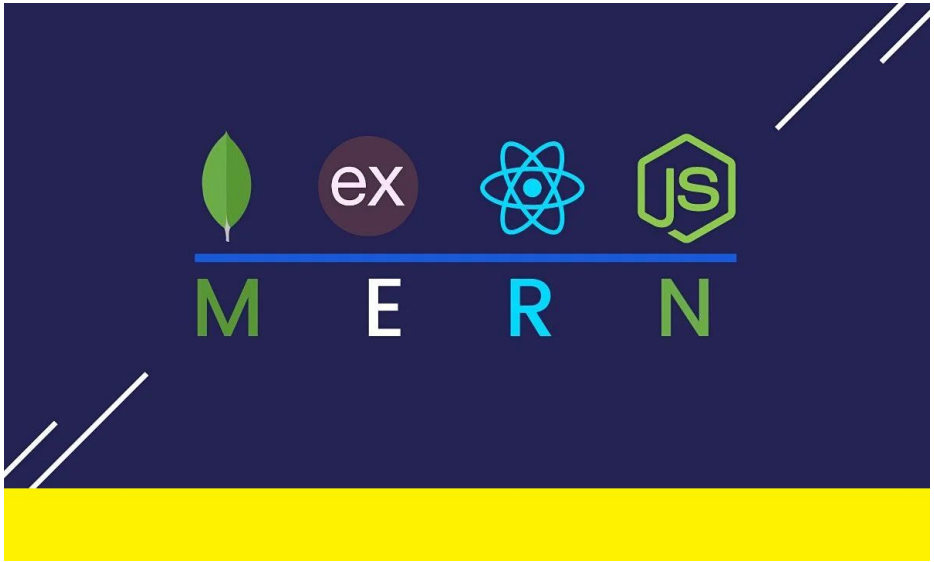
Web Application

- User Interface
  - React
  - UI Framework: Semantic UI
  - GraphQL Query Language
  - PWA - Progressive Web Application

- Backend Interface
  - ExpressJS Server
  - NoSQL Mongoose Database
  - NodeJS

Deployed through Heroku

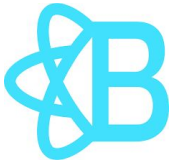
https://automatic-indoor-garden.herokuapp.com/



Web Application

CSS Framework Comparison: Form

React Bootstrap



vs

Semantic UI React

```jsx
import Button from 'react-bootstrap/Button';
import Form from 'react-bootstrap/Form';

function basicLogin() {
  return (
    <Form>
      <Form.Group className="mb-3" controlId="formBasicEmail">
        <Form.Label>Email address</Form.Label>
        <Form.Control type="email" placeholder="Enter email" />
      </Form.Group>

      <Form.Group className="mb-3" controlId="formBasicPassword">
        <Form.Label>Password</Form.Label>
        <Form.Control type="password" placeholder="Password" />
      </Form.Group>
      <Button variant="primary" type="submit">
        Submit
      </Button>
    </Form>
  );
}

export default basicLogin;
```

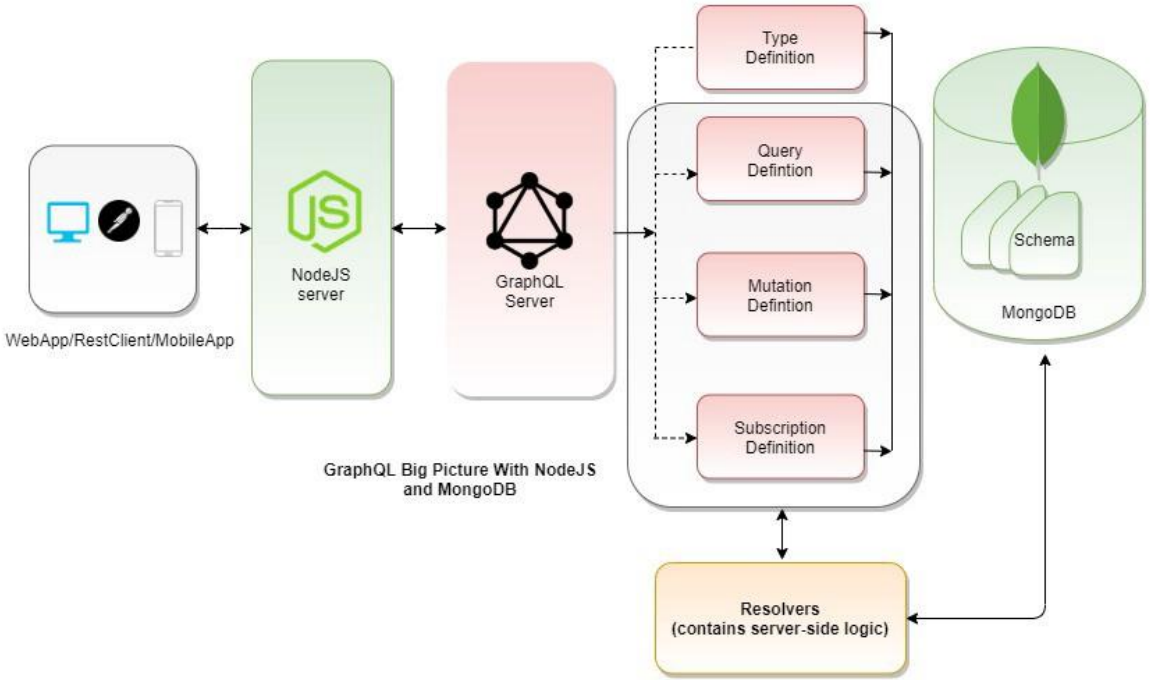
Source: https://react-bootstrap.github.io/forms/overview

```jsx
<Form name='signupContent' onSubmit={handleSubmit} className={animatedClass.props}>
    <Header>Sign Up</Header>
    <Form.Field
        required
        id='form-input-control-email'
        name='email'
        control={Input}
        type='email'
        label='Email'
        onChange={updateSubmission}
        placeholder='test@domain.com'
        error={
            (submission?.email?.split('').length > 0) &&
            !validateEmail(submission.email) &&
            { content: "Please enter a valid email address." }
        }
    />
    <Form.Field
        required
        id='form-input-control-password'
        name='password'
        control={Input}
        type='password'
        label='Password'
        onChange={updateSubmission}
        error={
            (submission?.password?.split('').length > 0) &&
            !validatePassword(submission.password) &&
            { content: "Password must contain uppercase, lowercase, number and  special character (@$!%*?&)" }
        }
    />
    <Form.Field
        id='form-button-control'
        control={Button}
        name='Submit'
        content='Register'
    />
</Form>
```

Source: https://github.com/brobro10000/automatic-indoor-garden/blob/main/client/src/components/login/index.js

Network: Server/Database

- Document Object Model of MongoDB in conjunction with a GraphQL querying language
- Allows us to pull a single document query instance per database call, while tailoring document for only relevant data with GraphQL
- Retool REST API Generator to create a layer sending and retrieving data between the website and Wi-Fi module



Plant Database Queries: Create User

Operation

Mutation

```
mutation Mutation($email: String!, $password: String!) {
  createUser(email: $email, password: $password) {
    token
    user {
      email
      password
    }
  }
}
```

Response

STATUS 200 | 207ms | 360B


```
{
  "data": {
    "createUser": {
      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJkYXRhIjp7ImVtYWlsIjoidGVzdEBtYWlsLmNvbSIsIl9pZCI6IjYzMjI3ZGE
xMTI4ODRjODVjNGUwNDc5OSJ9LCJpYXQiOjE2NjMyMDQ3NjksImV4cCI6MTY2Mz
IxMTk2OX0.wsNMF5JOYDiU3oaJTCOLv_5b29IBz0dMhTdU56aMjjY",
      "user": {
        "email": "test@mail.com",
        "password": "$2b$10$Bq11M1oHF5jDW3DMilk3.
ud60HdzJ2yoRTc6c4Z0yA/nGt2BTvNE2"
      }
    }
  }
}
```

Plant Database Queries: Create Plant

Operation

CreatePlant

```
mutation CreatePlant($name: String!, $temperature: Int, $pH: Float, $humidity: Int) {
  createPlant(name: $name, temperature: $temperature, pH: $pH, humidity: $humidity) {
    _id
    name
    temperature
    pH
    humidity
    history {
      _id
      createdAt
      temperature
      pH
      humidity
    }
  }
}
```

Variables | Headers

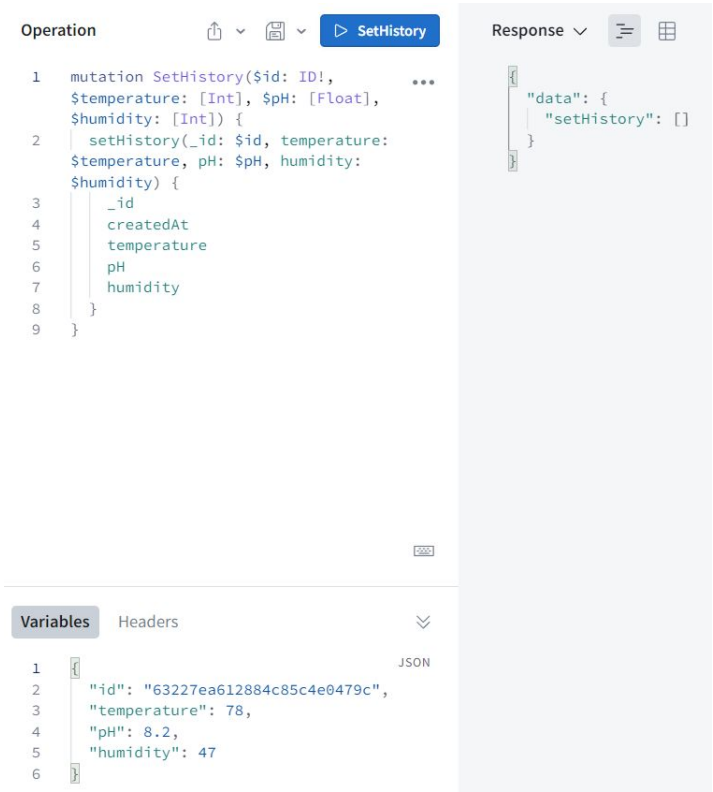
JSON

```
{
  "name": "Basic",
  "temperature": 78,
  "pH": 8,
  "humidity": 50
}
```

Response


```
{
  "data": {
    "createPlant": {
      "_id": "63227ea612884c85c4e0479c",
      "name": "Basic",
      "temperature": 78,
      "pH": 8,
      "humidity": 50,
      "history": []
    }
  }
}
```

Plant Database Queries: Set History



Plant Database Queries: Set History

Operation

Query

```
query Query($id: ID!) {
  historyById(_id: $id) {
    _id
    createdAt
    temperature
    pH
    humidity
  }
  plants {
    _id
    name
    temperature
    pH
    humidity
    history {
      _id
      createdAt
      temperature
      pH
      humidity
    }
  }
}
```

Variables | Headers

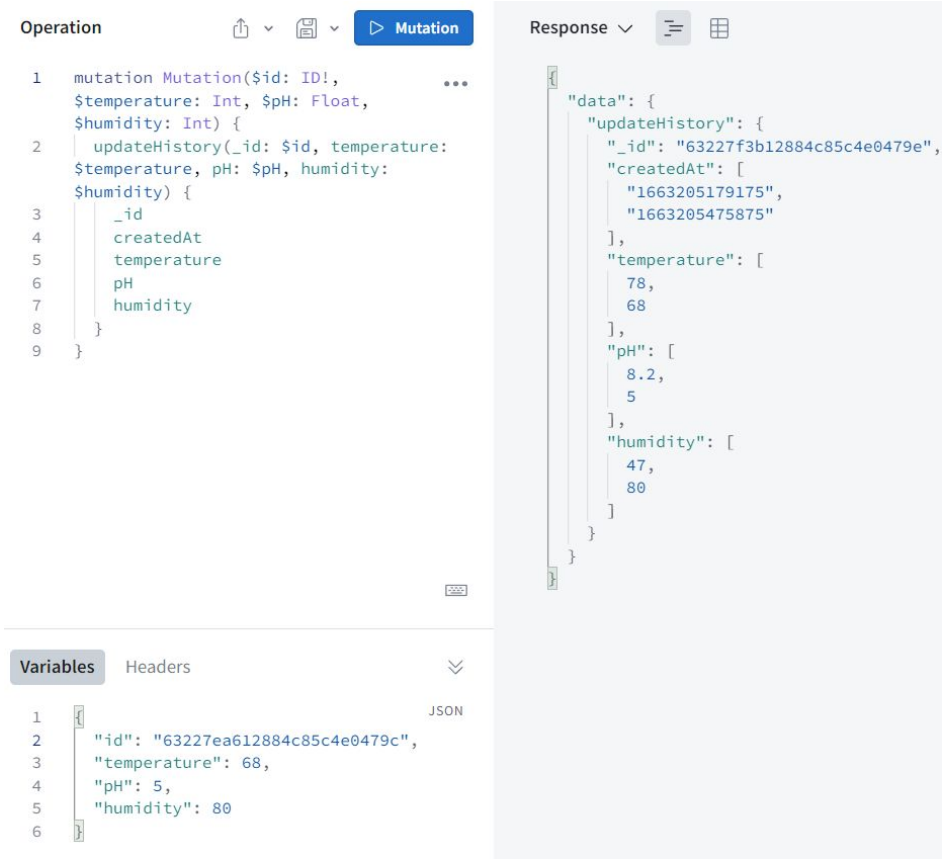
JSON

```
{
  "id": "63227ea612884c85c4e0479c"
}
```

Response


```
{
  "data": {
    "historyById": null,
    "plants": [
      {
        "_id": "63227ea612884c85c4e0479c",
        "name": "Basic",
        "temperature": 78,
        "pH": 8,
        "humidity": 50,
        "history": [
          {
            "_id": "63227f3b12884c85c4e0479e",
            "createdAt": [
              "1663205179175"
            ],
            "temperature": [
              78
            ],
            "pH": [
              8.2
            ],
            "humidity": [
              47
            ]
          }
        ]
      }
    ]
  }
}
```

Plant Database Queries: Update History



Plant Database Queries: Update History

Operation

Query

```
query Query($id: ID!) {
  historyById(_id: $id) {
    _id
    createdAt
    temperature
    pH
    humidity
  }
  plants {
    _id
    name
    temperature
    pH
    humidity
    history {
      _id
      createdAt
      temperature
      pH
      humidity
    }
  }
}
```

Variables | Headers

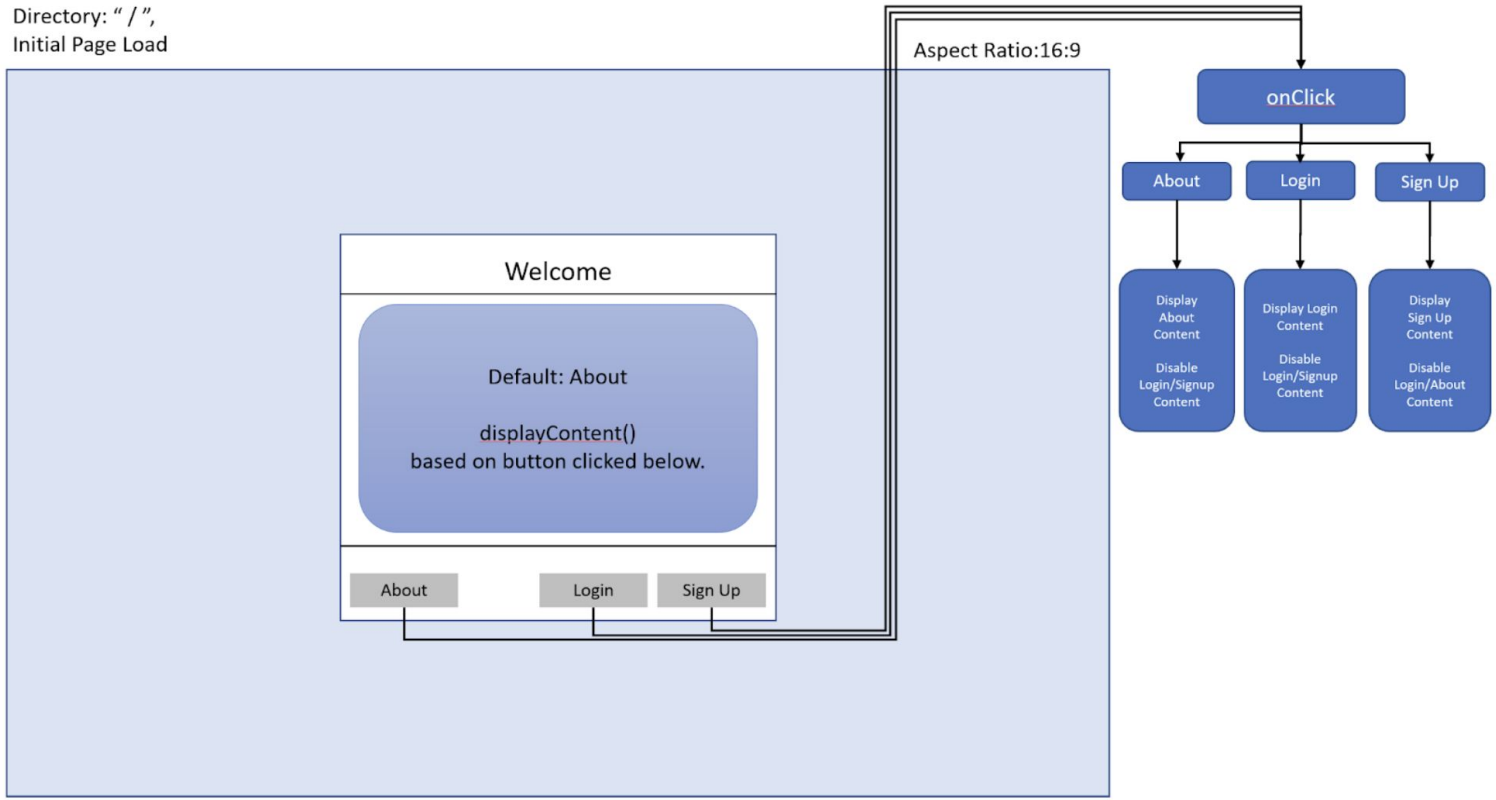
JSON

```
{
  "id": "63227ea612884c85c4e0479c"
}
```

Response

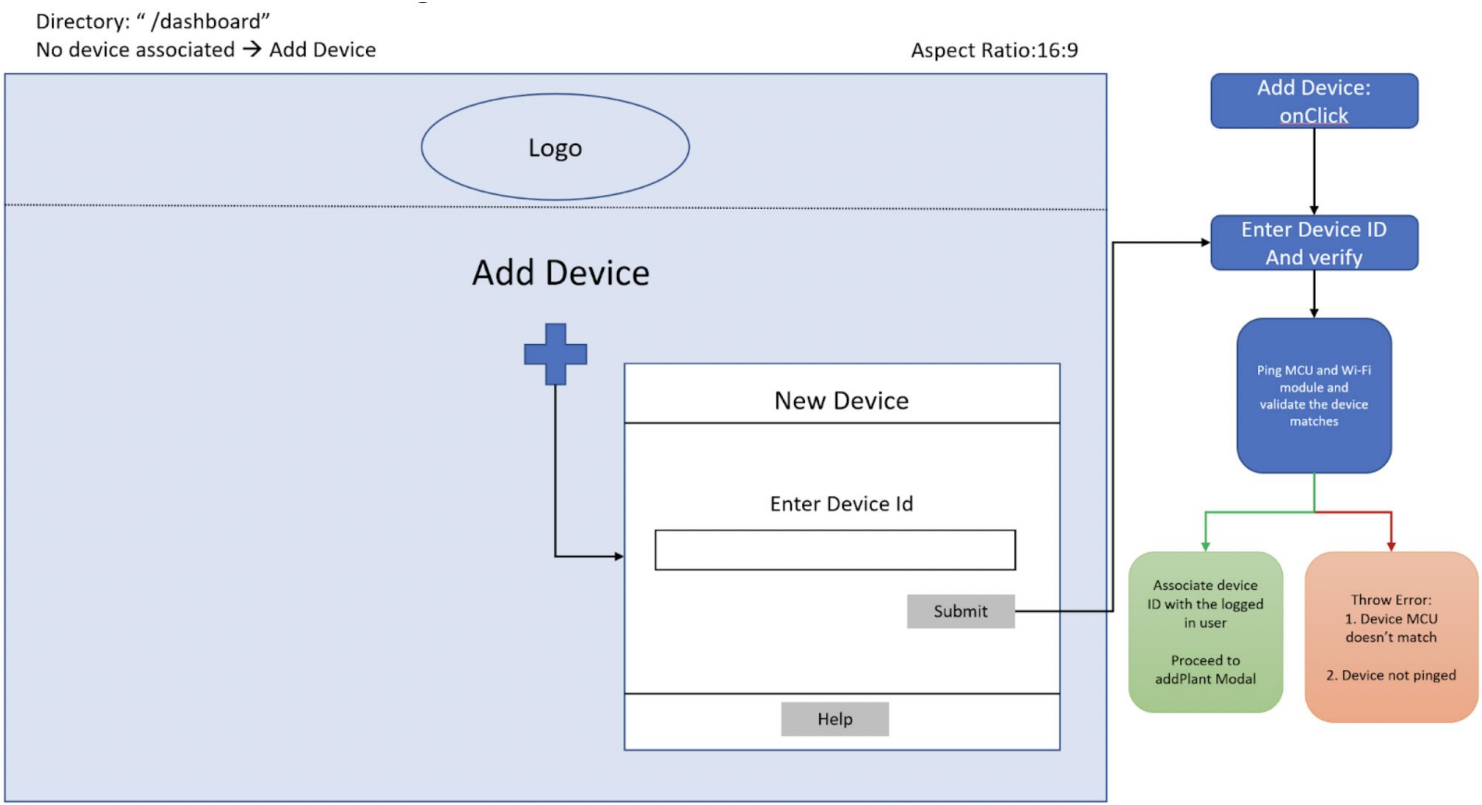

```
{
  "data": {
    "historyById": null,
    "plants": [
      {
        "_id": "63227ea612884c85c4e0479c",
        "name": "Basic",
        "temperature": 78,
        "pH": 8,
        "humidity": 50,
        "history": [
          {
            "_id": "63227f3b12884c85c4e0479e",
            "createdAt": [
              "1663205179175",
              "1663205475875"
            ],
            "temperature": [
              78,
              68
            ],
            "pH": [
              8.2,
              5
            ],
            "humidity": [
              47,
              80
            ]
          }
        ]
      }
    ]
  }
}
```

Website Flow: Initial Page Load

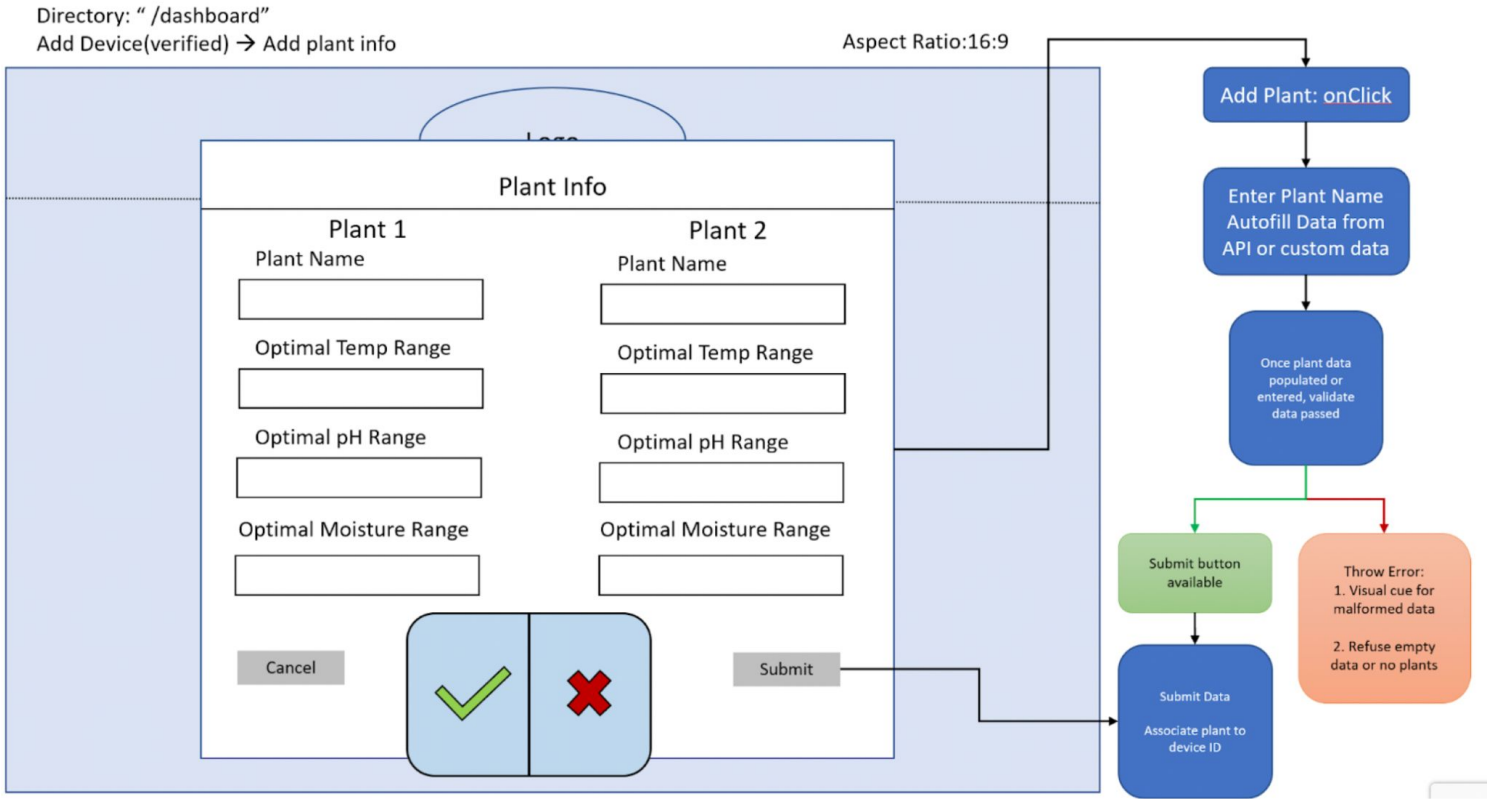


Website Flow: Login/Sign Up

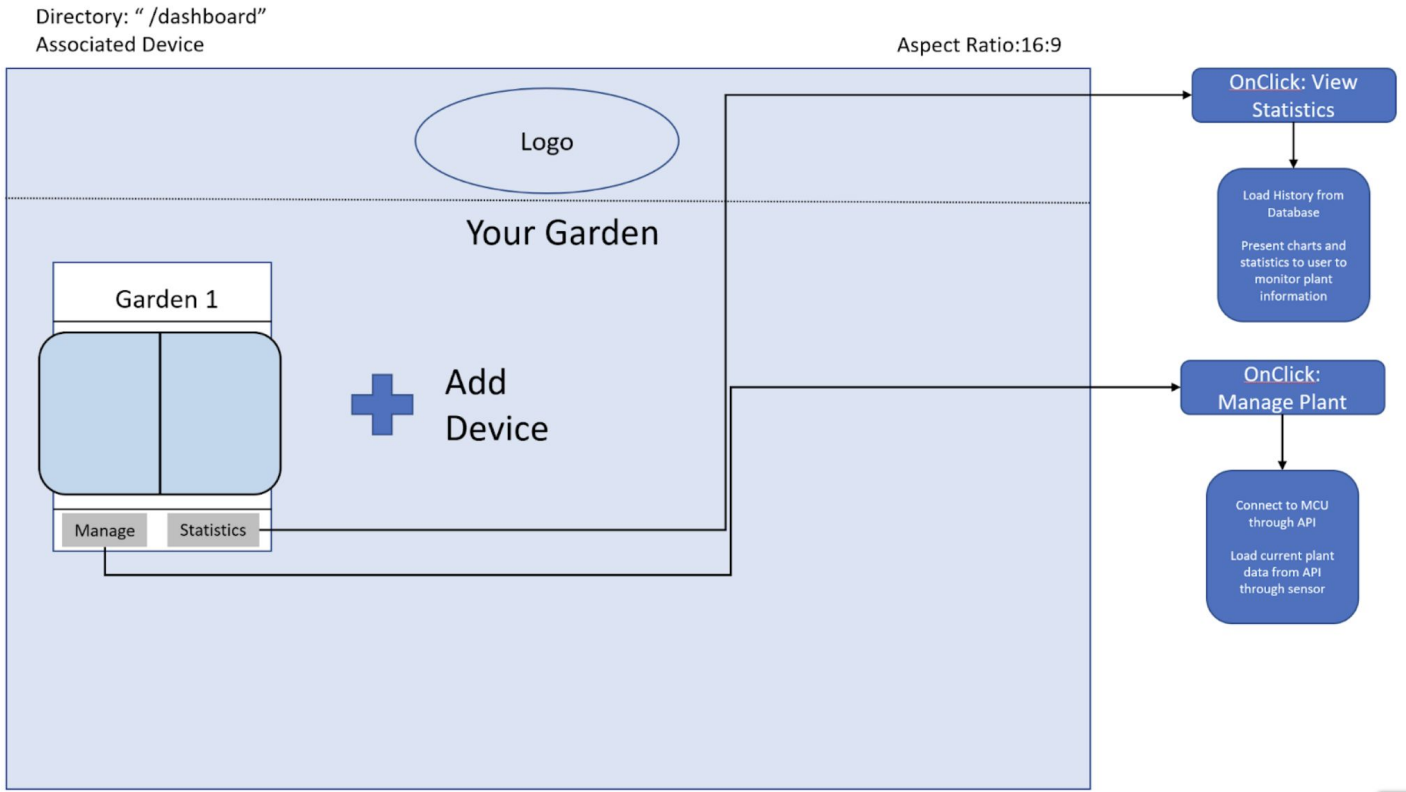
Website Flow: Dashboard, Add Device



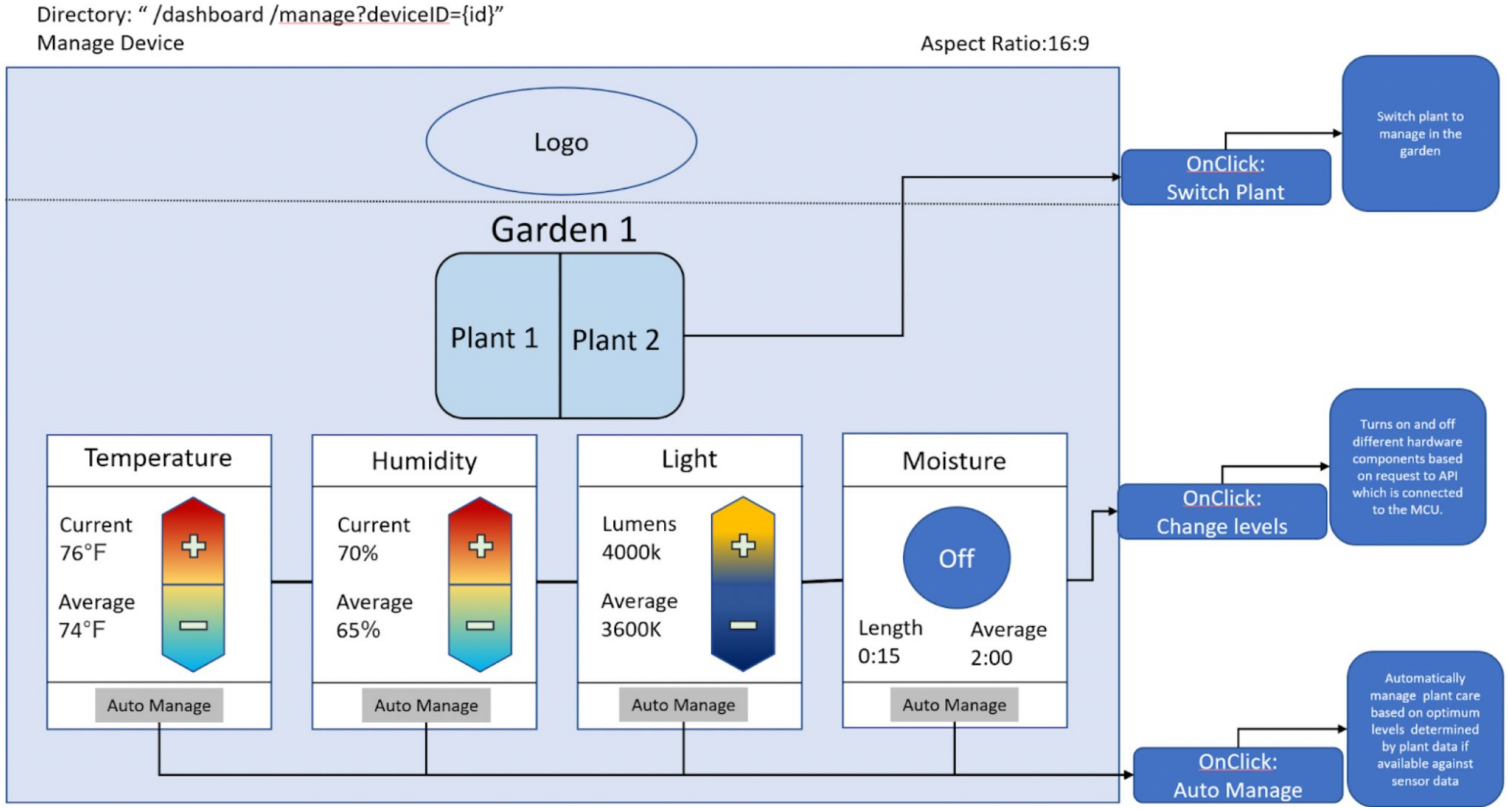
Website Flow: Dashboard, Add Plant



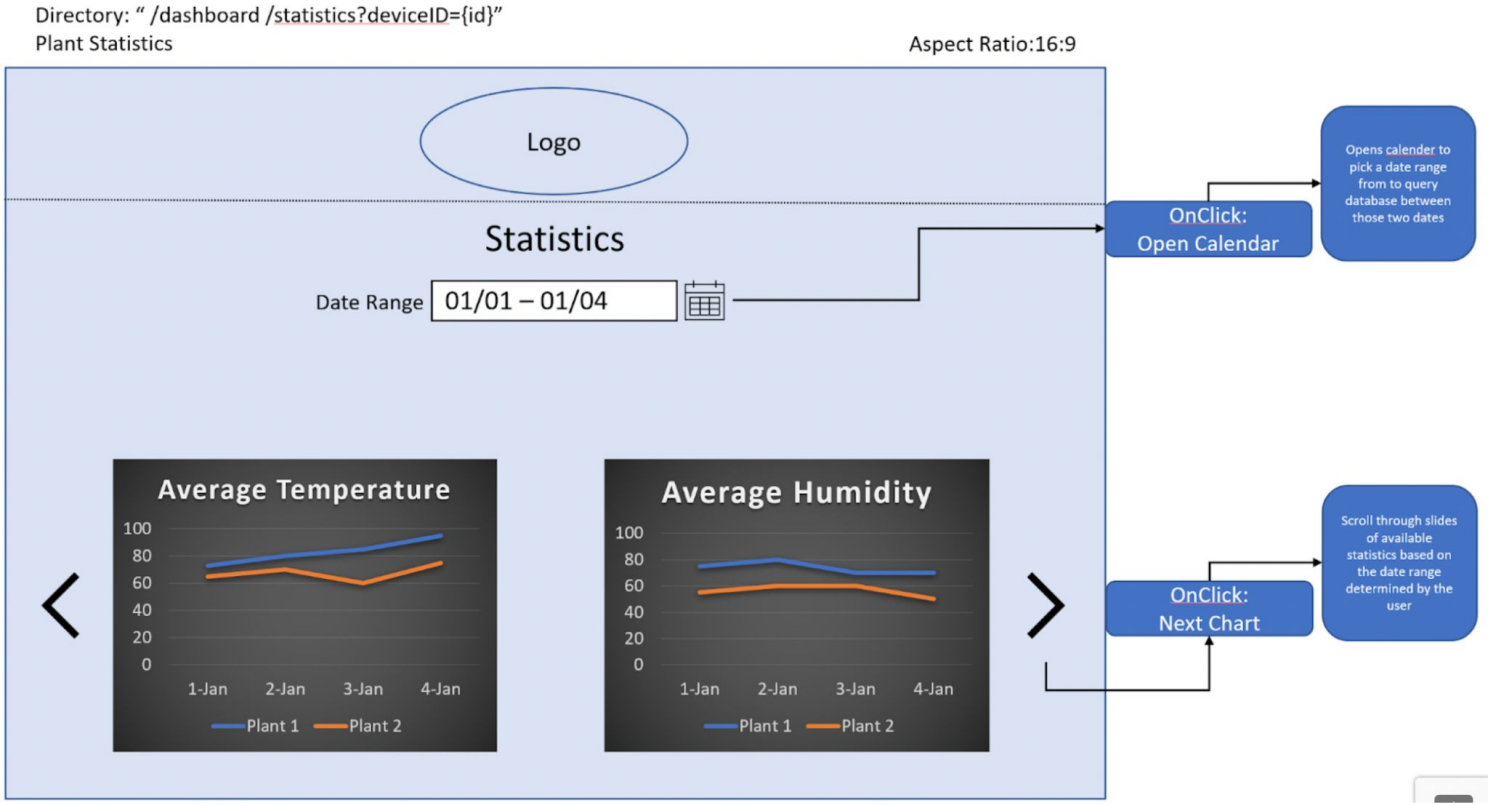
Website Flow: Dashboard, Device Added



Website Flow: Manage Plant



Website Flow: Plant Statistic



Administrative Content

Work Distribution

Mariana, Austen, and Nick are the three EEs therefore were responsible for the printed circuit board. The three worked together on the testing, PCB schematic, and prototyping.

Hamzah is the sole CPE therefore responsible for the software aspect of the project. He worked on our website and interface.

In the end we all contributed to every aspect of the overall project with the help of one another.

Budget

- ➔ Final budget cost for the customer.
- ➔ Not included cost for extra PCBs because they are not included in final product.

Item	Quantity	Price
Final PCB	1	50.00
Potting Mix	1	5.97
Planters	4	13.92
Miracle Grow Plant Food 2 lbs	1	10.97
Plant Seeds (Parsley and Basil)	6	10.14
PVC pipe + elbow fittings	35	43.81
3/4' to 3/8' adapter	1	10.64
Reservoir	1	38.56
Corrugated Sheet	2	15.16
Vinyl Tubing	1	16.19
Cat Water Fountain Pump	1	11.99
Vinyl (per yard)	3	45.10
Light Sensor	1	21.10
CQRobot Liquid Sensor	1	18.99
Coospider UV Green Killer	1	20.77
DHT11 Humidity and Temperature Sensor	1	free
Wifi Module	3	8.99
Grow Lights	2	32.00
Drip Irrigation System	1	21.84
Liquid Level Sensor	1	free
Camera	1	25.99
Barrel Plug (around 24 V)	1	8.89
Heroku Subscription	1	6.99
	Total:	438.01

Thank you for watching our presentation we hope you enjoyed it!