# Automated Indoor Nursery

Group 10:
Mariana Lozano - Electrical Engineer
Nicholas Leon - Electrical Engineer
Austen Ordos - Electrical Engineer
Hamzah Ullah - Computer Engineer

EEL4915 - Fall 2022

# Overview

Product Description:

- ➔ An indoor plant system that will provide enough water and sunlight to keep these herbs alive.
- ➔ This device will monitor 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.
- ➔ Show how these herbs will grow in and outside of their perfect environment.

# Objectives and Goals

| 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. |

➔ Our goal is to meet all these objectives as well as provide functionality of our entire product at the end of this semester.

# Market Specifications

- The market requirements are 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 |

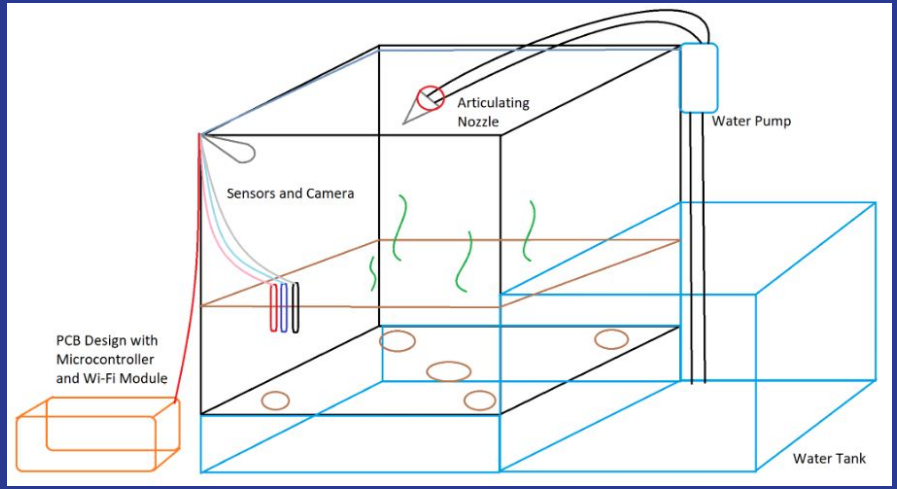
# Engineered Specifications

- The engineering requirements are important to lay out because they are 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 |

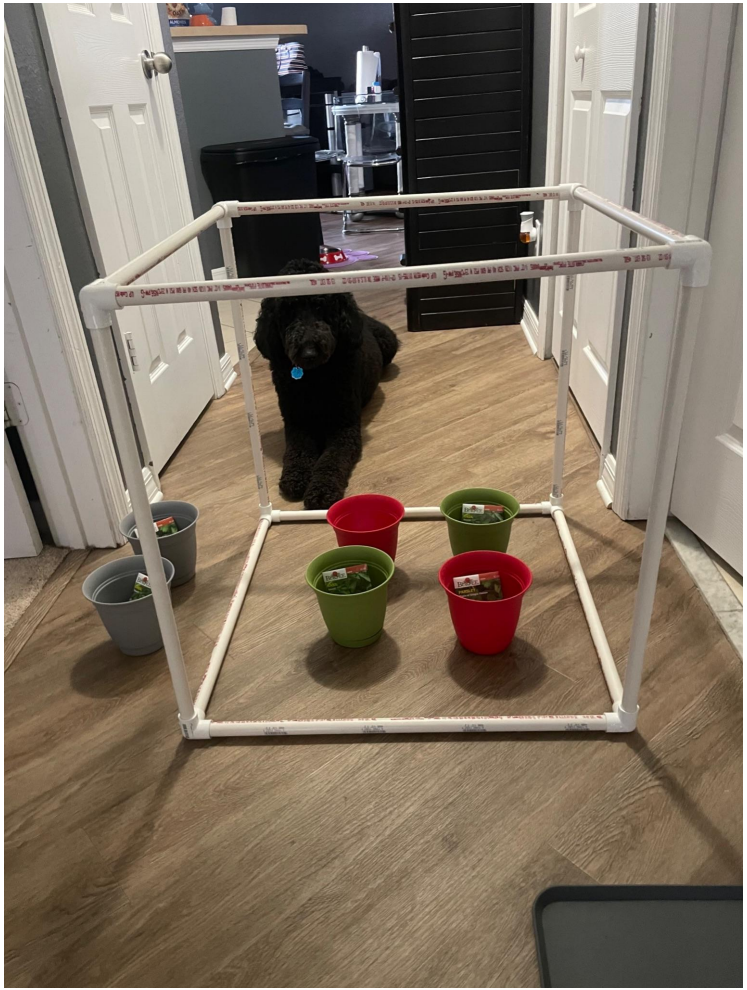
# System Design

Sketch of what we want the enclosure to look like



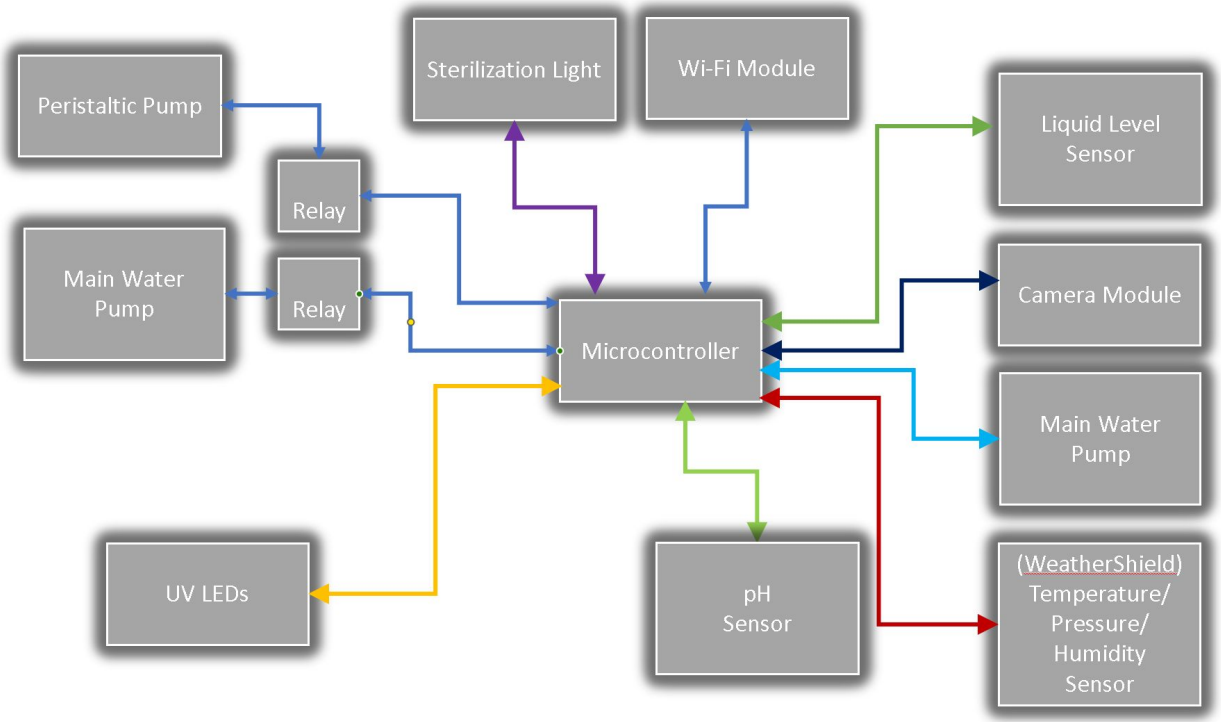
# Enclosure Building

- ➔ The picture on the right shows what the system looks like so far with the pieces we have already bought/acquired.
- ➔ We do not have the reservoir tank or the water tank yet but those are to be ordered soon.

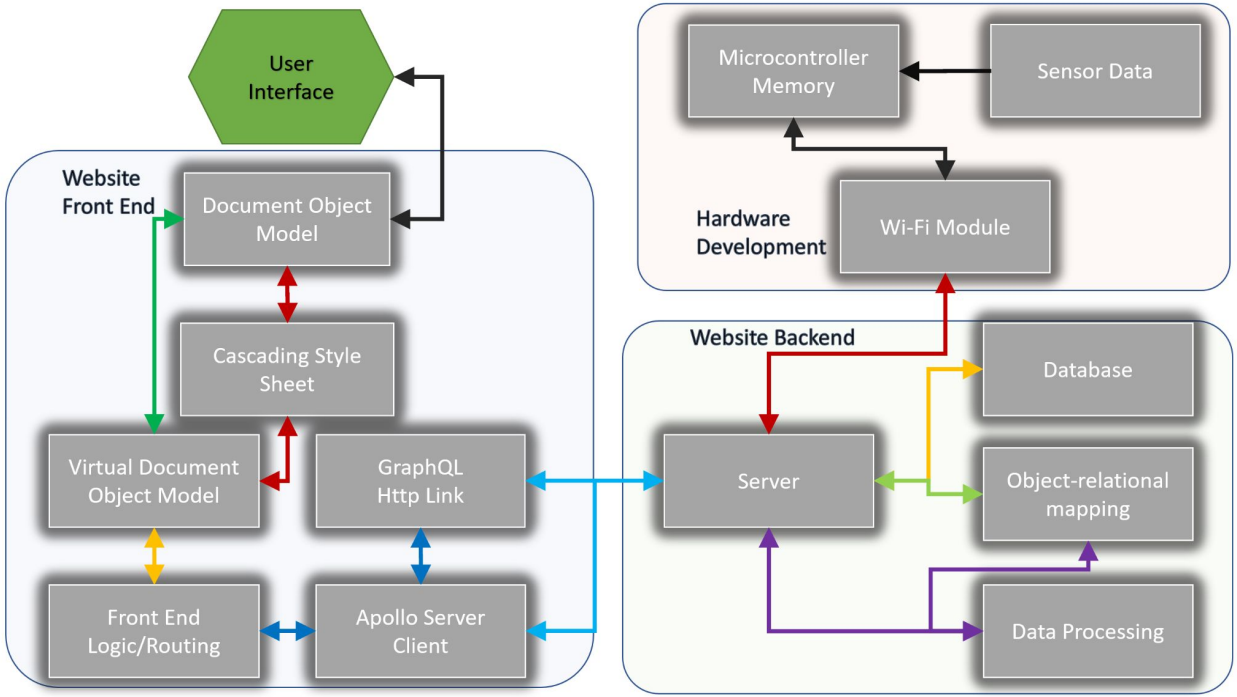



# Block Diagrams

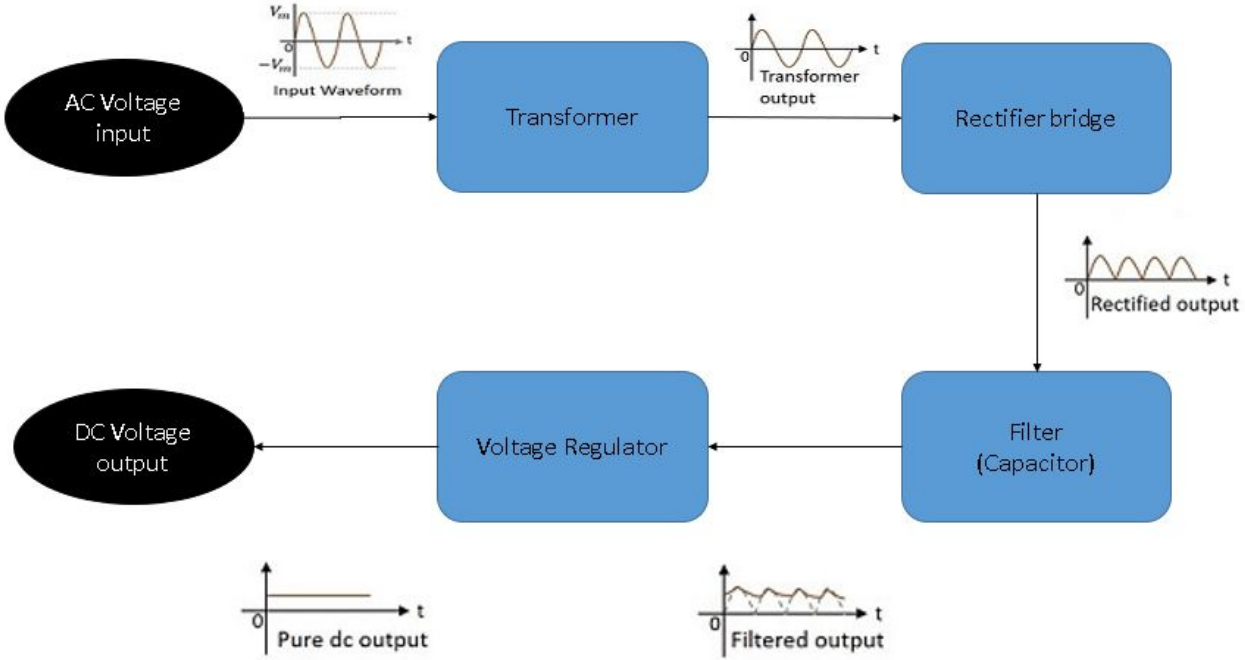
# Hardware Block Diagram



# Software Block Diagram



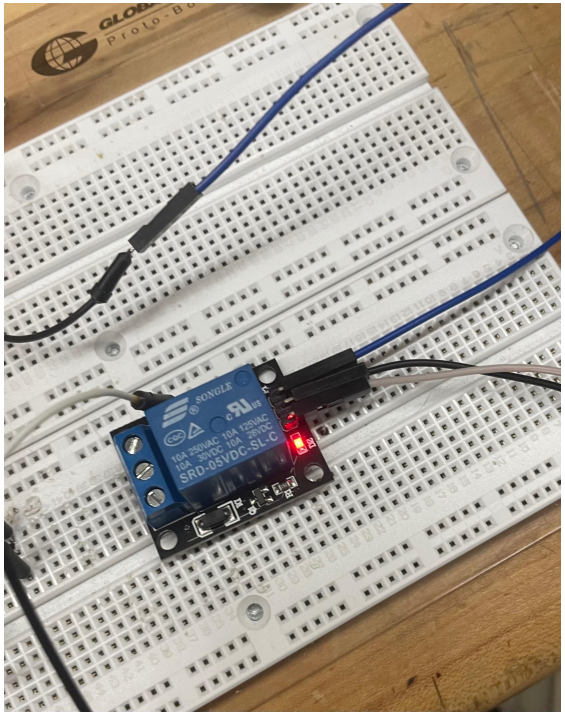
# Power Block Diagram



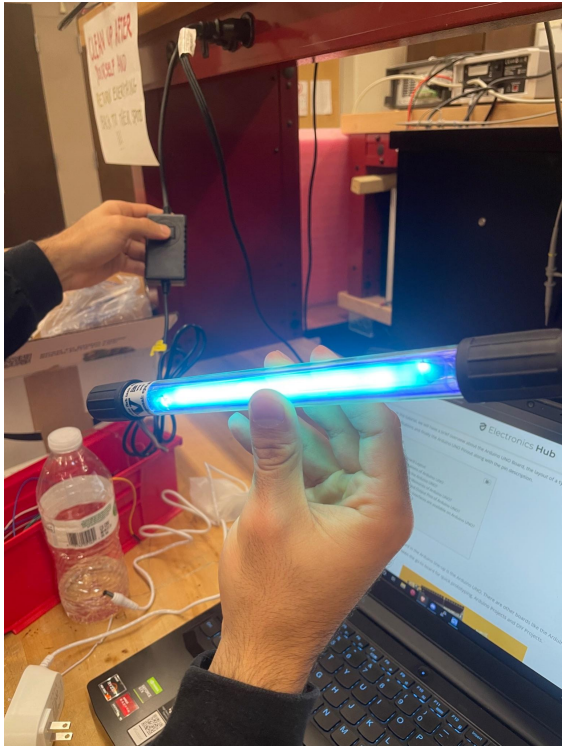
# Hardware Design

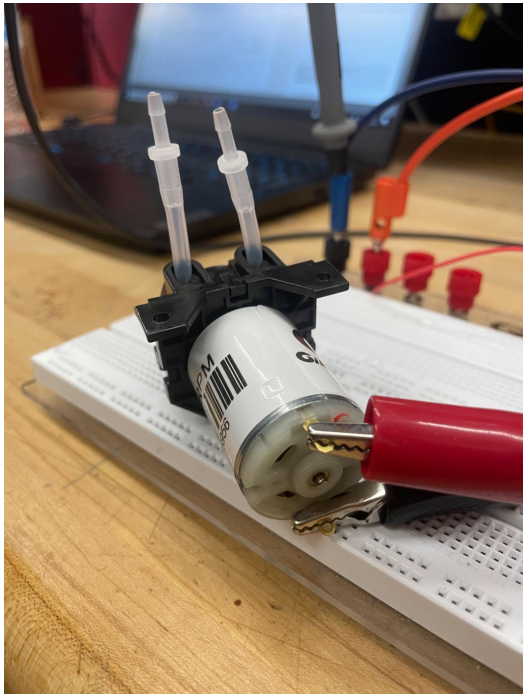
# Hardware Testing

Relay Module



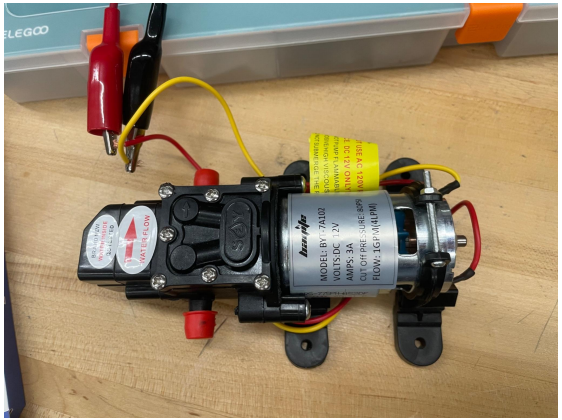

Green Killer





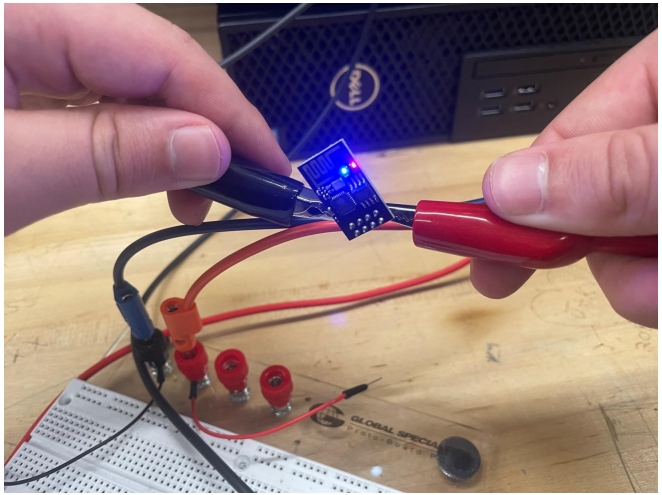
Peristaltic Pump

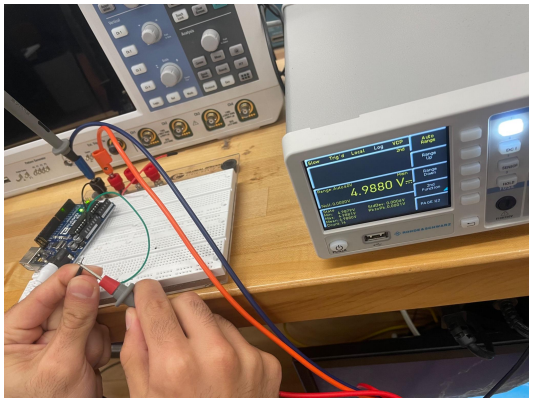
# Hardware Testing (cont.)



Diaphragm Pump (12 V)

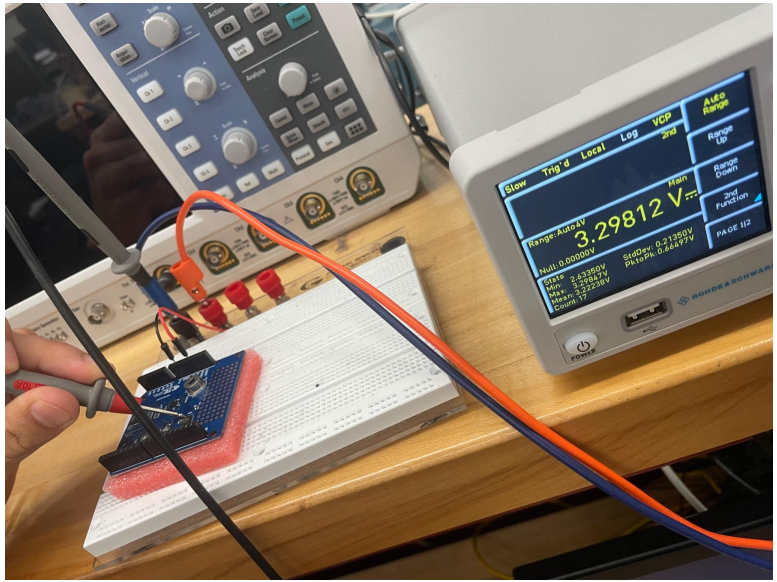
Wi-Fi Module (3.3 V)



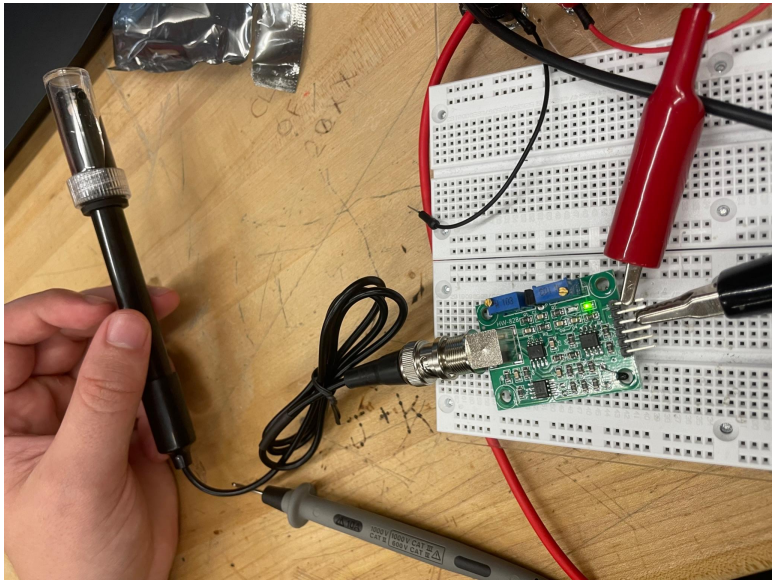



Arduino Uno R3 (shown working for 5 V also works for 3.3 V)

# Hardware Testing (cont.)



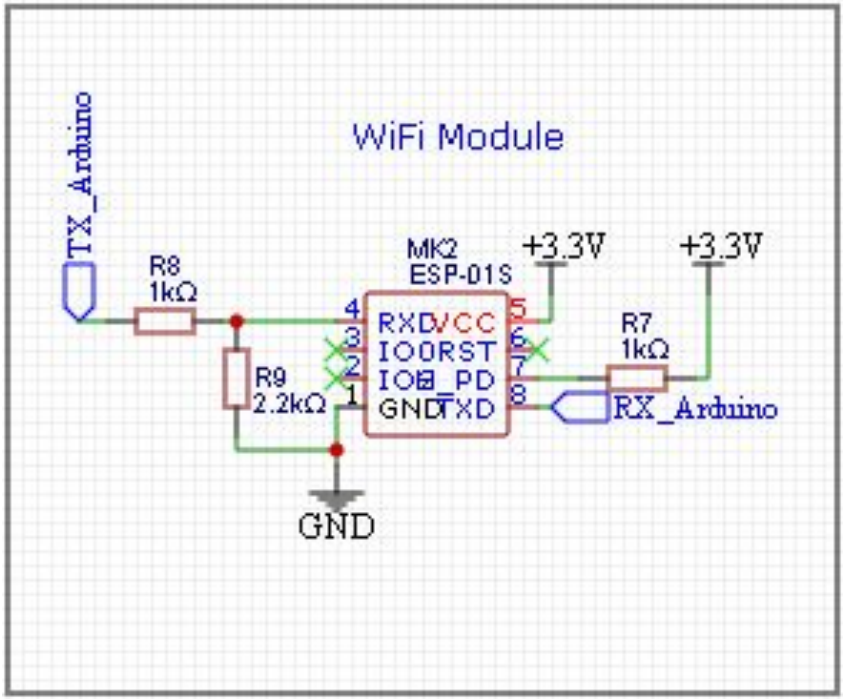

WeatherShield Arduino



pH Sensor (5 V)

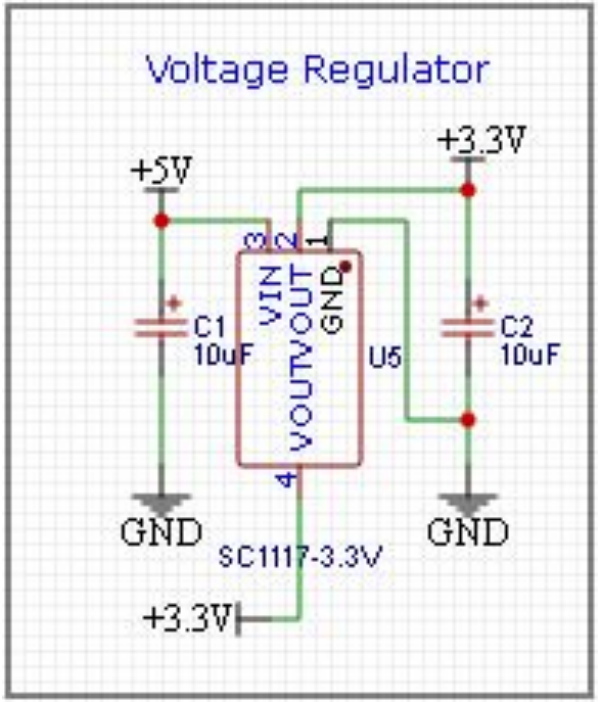
# PCB Schematic

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



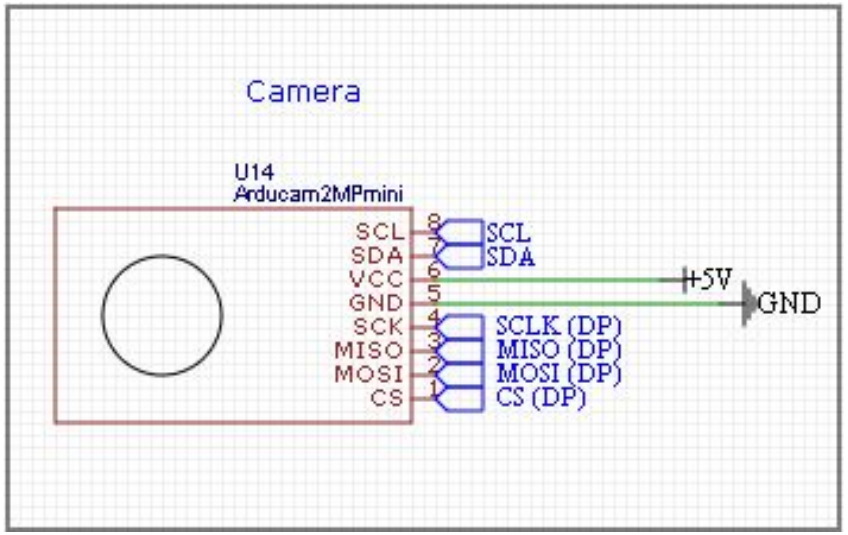
# PCB Schematic

- ➔ Voltage regulator that converts +5V into +3.3V



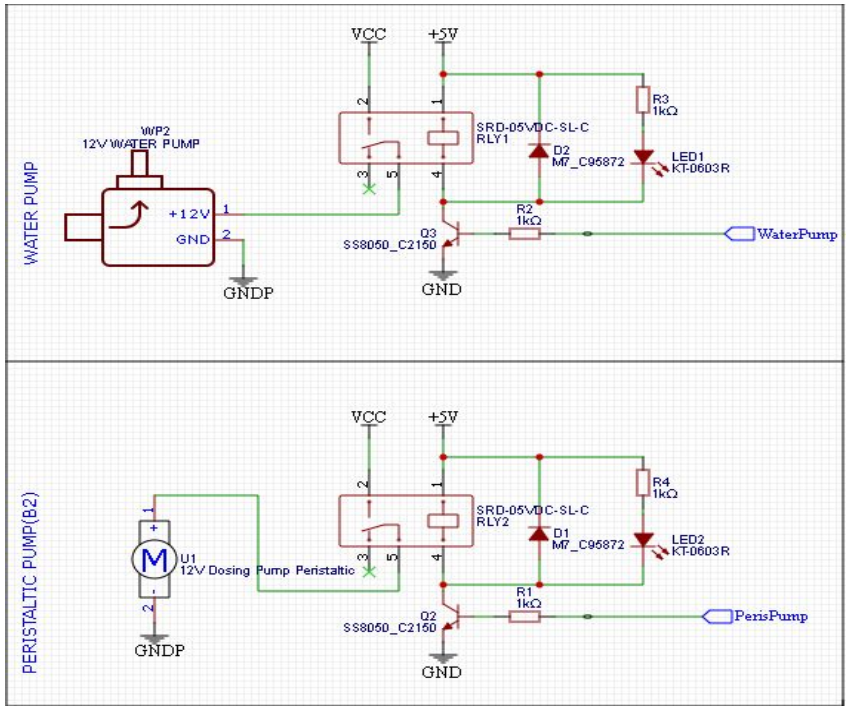
# PCB Schematic

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



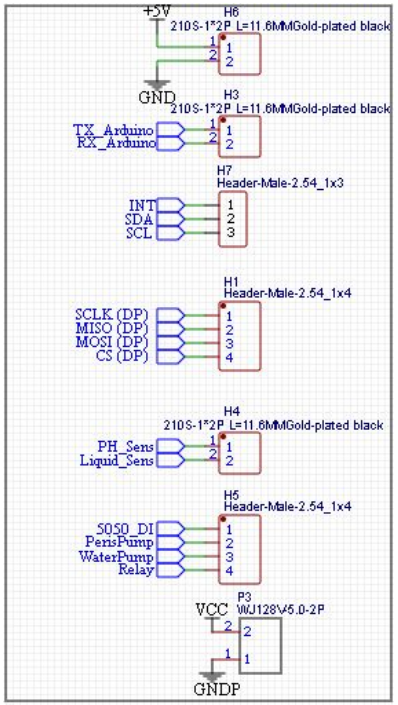
# PCB Schematic

- Both pumps that that regulate water ph-level and amount of irrigation that occurs.

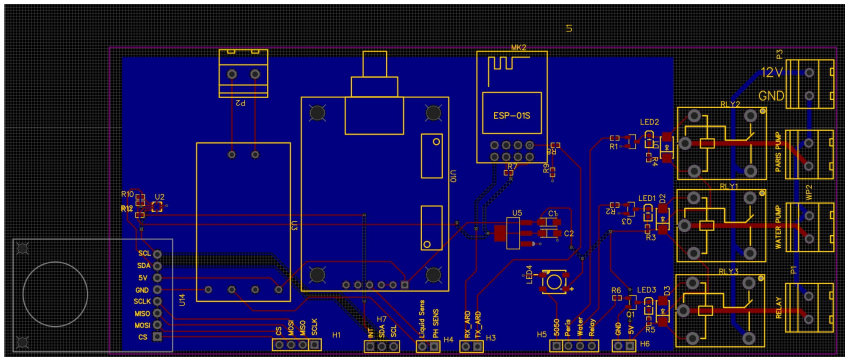


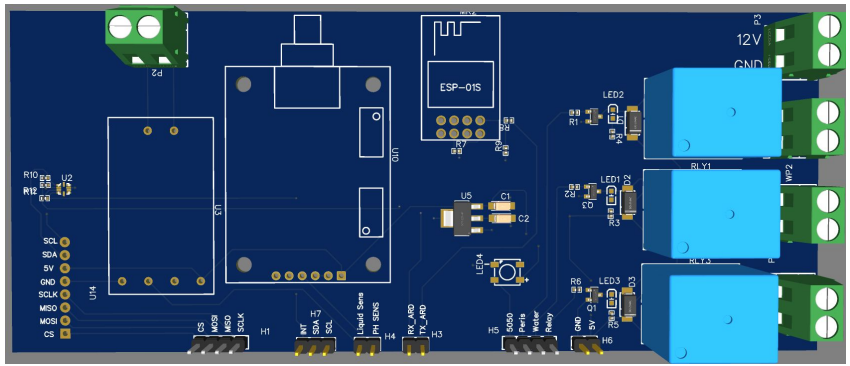
# PCB Schematic

- ➔ Connectors and pin headers that allow for external device connection.

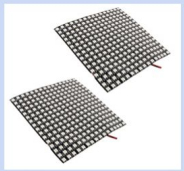


# PCB Layout (Attempt 1 Ordered)



# Components

| | Description | Brand | Dimensions | Details | Price |
|---|---|---|---|---|---|
|  | ACEIRMC 2pcs 16X16 RGB LED Flexible WS2812B 5050 Matrix Dream Color Individually Addressable LED Programmable Display Screen led Panel WS01 for Arduino (16X16) | ACEIRMC | 6.3 in x 6.3 in | | $33.99 QTY: 2 |
|  | Fresh Water Pressure Diaphragm Pump with Hose Clamps | bayite | | Self Priming<br>GPM: 1<br>Power Consumption: 12V DC<br>PSI: 80 | $20.95 |
|  | Dosing Pump Peristaltic Dosing Head with Connector For Arduino Aquarium Lab Analytic Diy AE1207 | Gikfun | | Voltage: 12V DC<br>Flow Rate: 0-100mL/min | $11.50 |

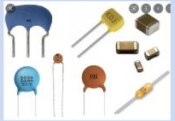
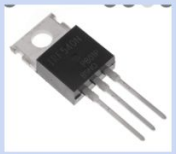
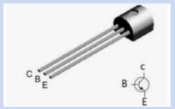
| | | | | | |
|---|---|---|---|---|---|
|  | PH0-14 Value Detect Sensor Module + PH Electrode Probe BNC For Arduino | GAOHOU | 4.5cm×3.2cm×20cm | Response Time: ≤5s<br>Power: ≤0.5W | $35.88 |
|  | Aquarium Clean Light Submersible Waterproof Lamp Water Clean Green Algae Clear for Fish Tank Pond (HUV-11) | Coospider | 11.6 x 3.7 x 1.7 inches | Power: 11 W | $20.77 |
|  | Ocean: Contact Water/Liquid Level Sensor Compatible with Raspberry Pi/Arduino | CQRobot | | Type: Contact<br>Input Voltage: 3.3 - 5 V DC | $14.11 |
|  | WeatherShield Arduino | DPP902S000 | | Forward Voltage: 1.5 - 3.6 V<br>Humidity Accuracy: 0-100%<br>Temp Accuracy: -40 to 125° C | $63.14 |

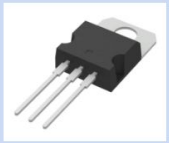
# Components (cont.)

| | | | | | |
|---|---|---|---|---|---|
|  | Adafruit Si1145 Light Sensor PCB | Adafruit | 20mm x 18mm x 2mm / 0.8" x 0.7" x 0.08"<br><br>Weight: 1.4g | SI1145 Digital UV Index / IR / Visible Light Sensor<br><br>Voltage Supply: Power with 3-5VDC<br><br>Output Type: I2C address 0x60 (7-bit)<br><br>Operating Temperature : -40°C ~ 85°C | $9.95 |
|  | ESP8266 (ESP-01) | HiLetgo | 0.98 x 0.59 x 0.39 inches | Forward Voltage: 3.3V (3.6 V MAX)<br><br>Wifi Security Modes: WPA, WPA2 | $8.99 |
|  | 12 V Power Supply, 12 V 2 A 24 W Switching Power Supply, Drive, Power Adapter, AC 110 V to 12 V Power Supply for LED Strip Light with 5.5/2.1 mm DC Female Barrel Connector | inShareplus | 5.5/2.1 mm | Voltage/Amps: 12 V / 2 A<br><br>Max Power of Device: 24 W | $8.99 |

| | | | | | |
|---|---|---|---|---|---|
|  | Relay Module | SunFounder | | # of Channels: 4<br><br>Voltage: 5 V<br><br>Current Rating: 10 A<br><br>Needs 15-20mA Driver Current. | $7.99 |
|  | Arducam Mini 2MP Plus – OV2640 SPI Camera Module for Arduino UNO Mega2560 Board & Raspberry Pi Pico | 2 megapixels image sensor OV2640 | | | $25.99 |
|  | Nozzle | | | | |
|  | Vinyl Tubing | | 10 feet<br><br>⅜ in | Clear Vinyl Tubing Flexible PVC Tubing, Hybrid PVC Hose, Lightweight Plastic Tubing, by 3/8 Inch ID, 10-Feet Length | $10.99 |

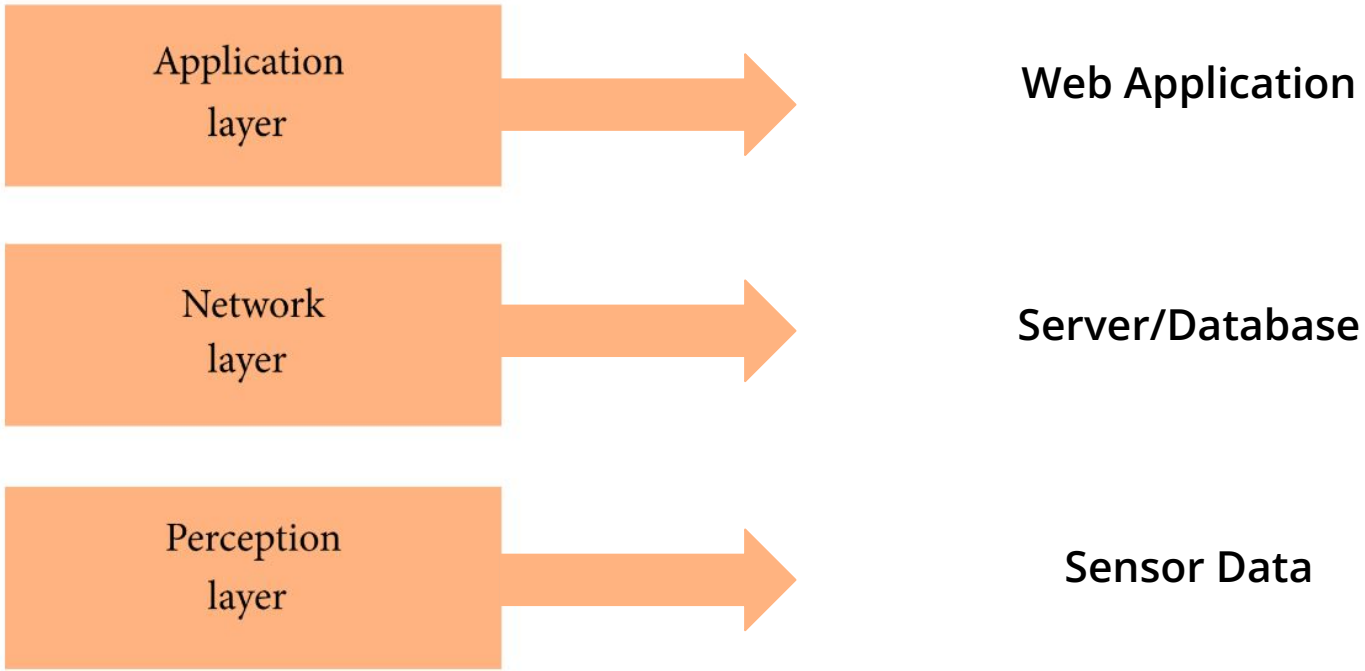
# Components (cont.)

| | | | | | |
|---|---|---|---|---|---|
|  | Resistors | | | Used for circuit design and power dissipation | |
|  | Capacitors | | | Used for circuit design, stabilizing, and decoupling | |
|   | MOSFET: Power Transistors | | | IRF540: Higher voltage source supplied to the pumps<br><br>Both: Can be controlled by MCU simultaneou sly | |
|  | BJT Transistor | | | Used to switch operations | |
|  | Zener Diodes | | | Used to switch operations | |

| | | | | | |
|---|---|---|---|---|---|
|  | Voltage Regulator | | | LM7805 | |

# Software Design

# Software Design



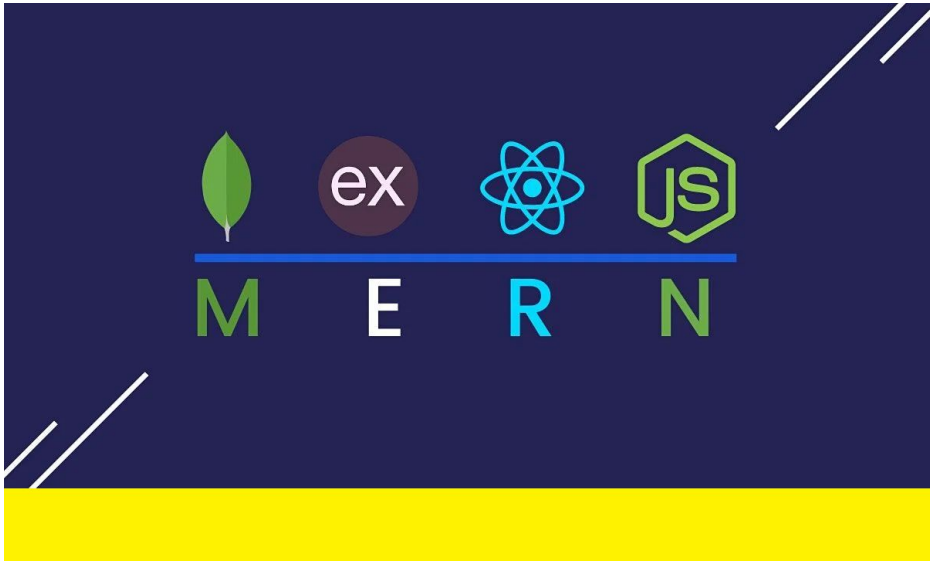
# Application: Web Application

- 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

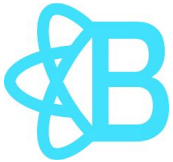
https://automatic-indoor-garden.herokuapp.com/



# Application: 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

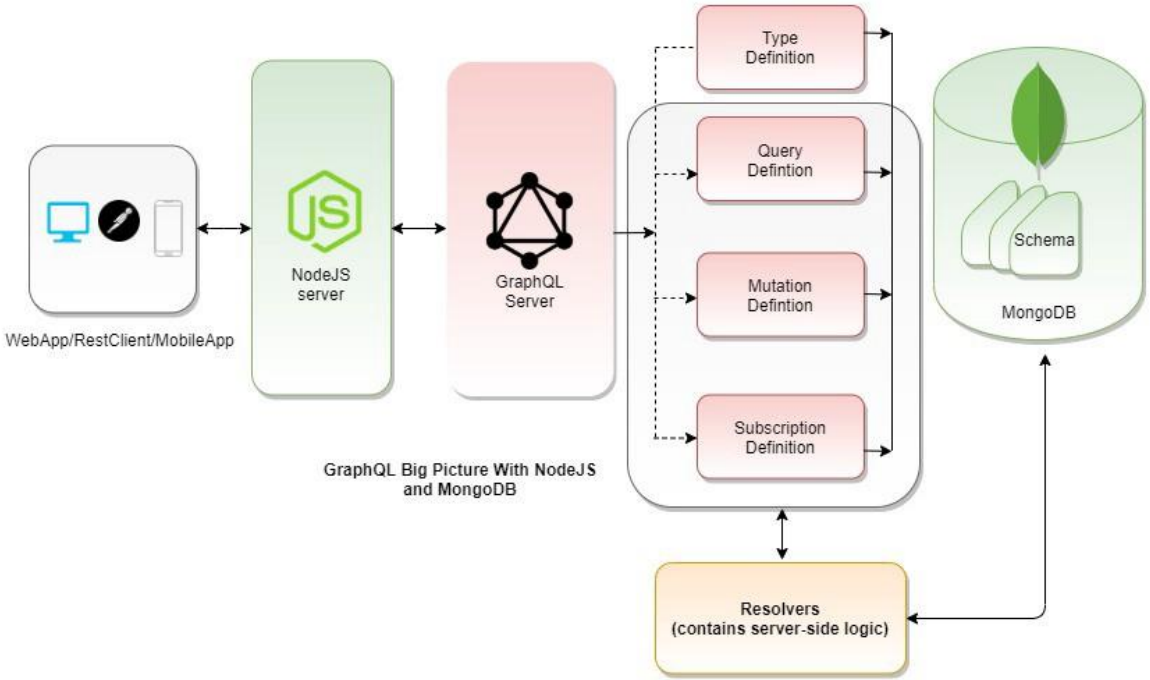
```jsx
<Form name='signupContent' onSubmit={handleSubmit} className={animatedClass.props}>
    <Header>Sign Up</Header>
    <Form.Field
        required
        id='form-input-control-email'
        name='email'
        control={Input}
        type='email'
        label='Email'
        onChange={updateSubmission}
        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



# 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

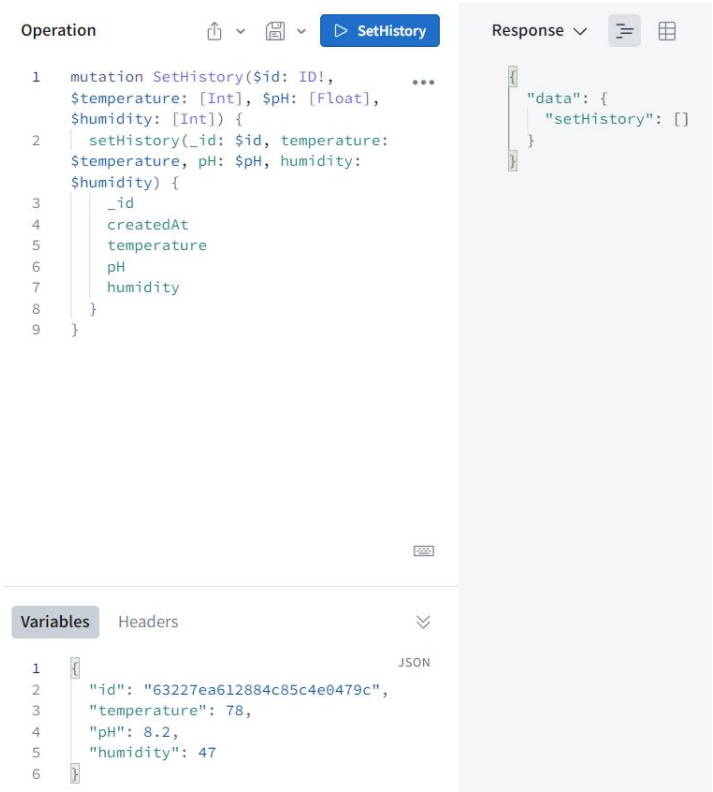
JSON

```
{
  "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

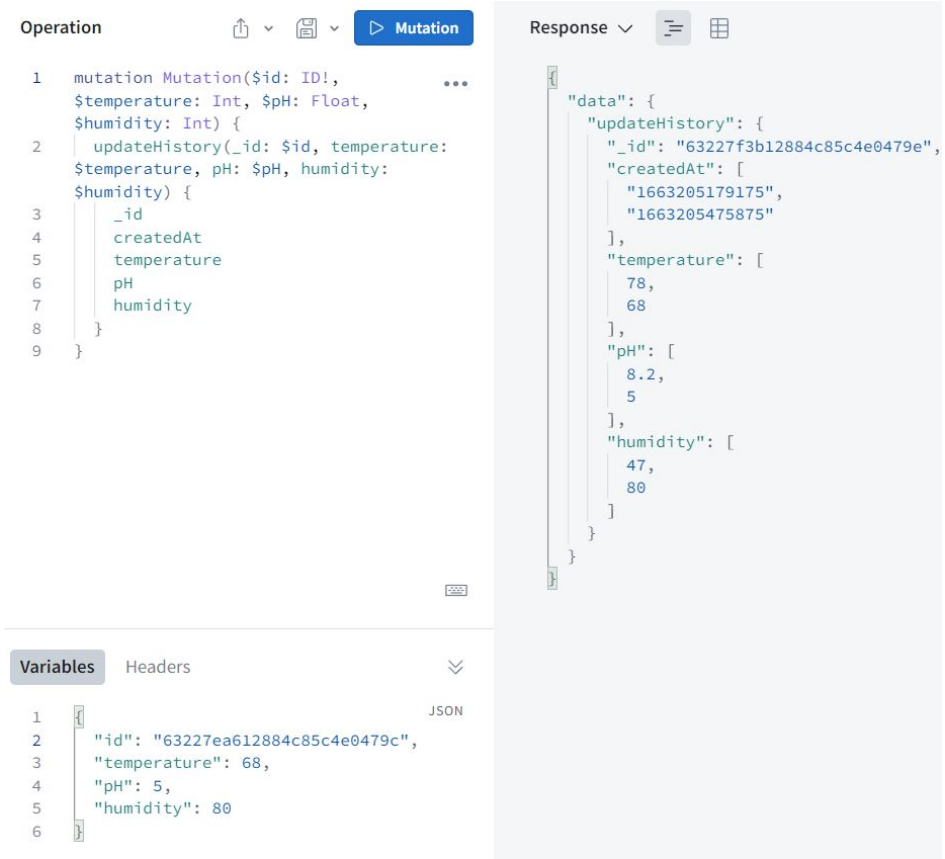
JSON

```
{
  "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

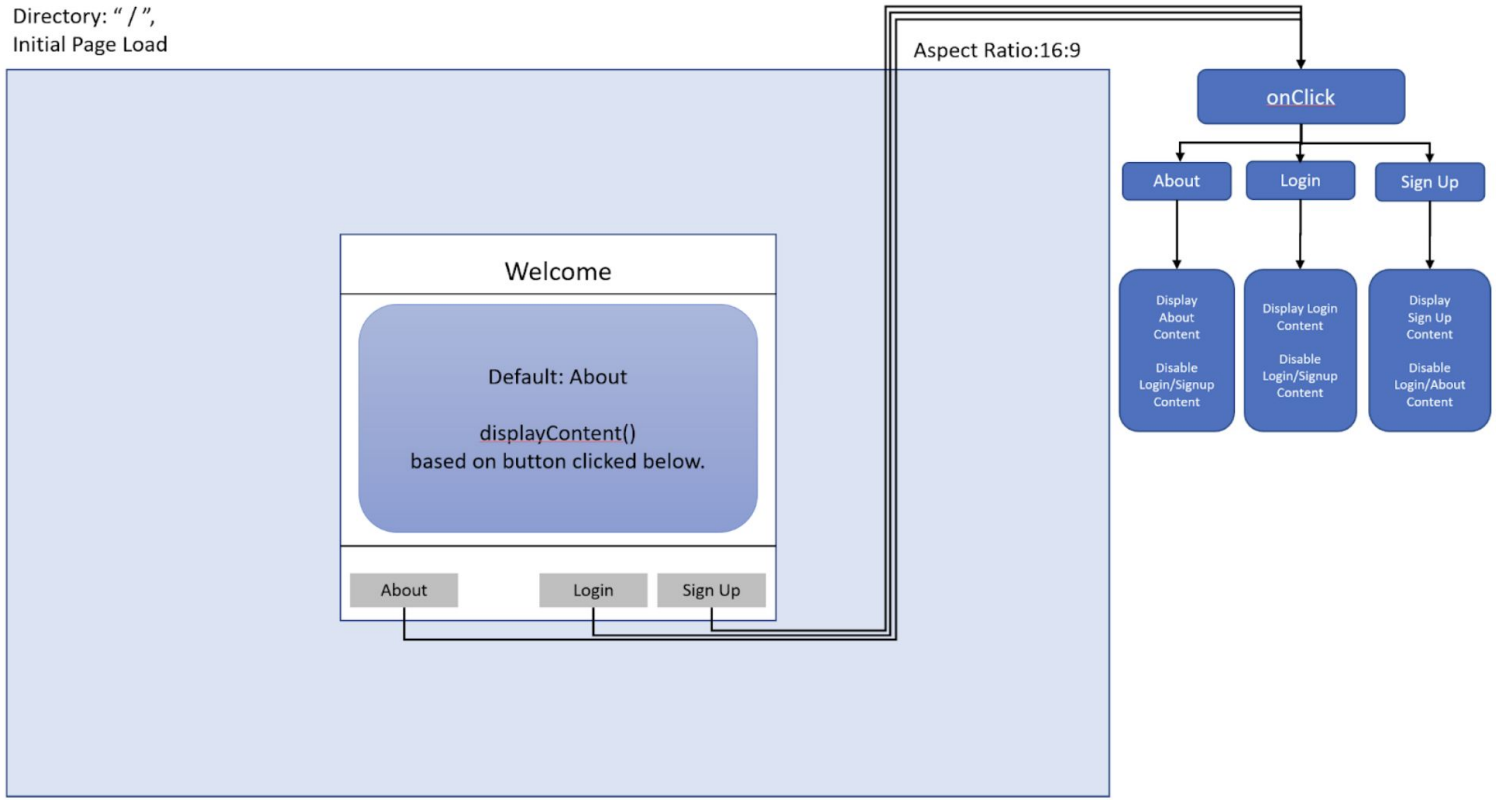
JSON

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

Response

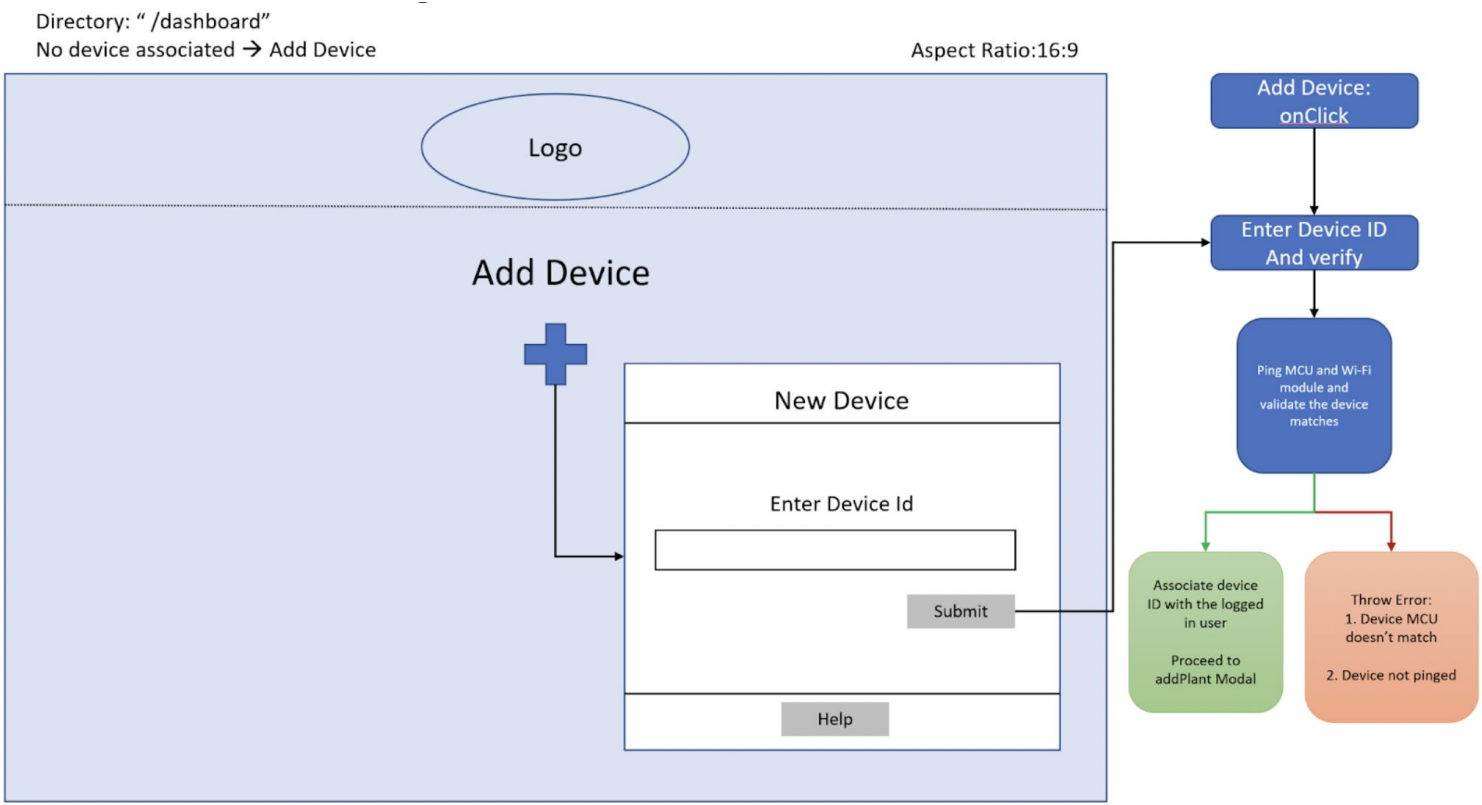

```
{
  "data": {
    "historyById": null,
    "plants": [
      {
        "_id": "63227ea612884c85c4e0479c",
        "name": "Basic",
        "temperature": 78,
        "pH": 8,
        "humidity": 50,
        "history": [
          {
            "_id": "63227f3b12884c85c4e0479e",
            "createdAt":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

# Website Flow: Initial Page Load

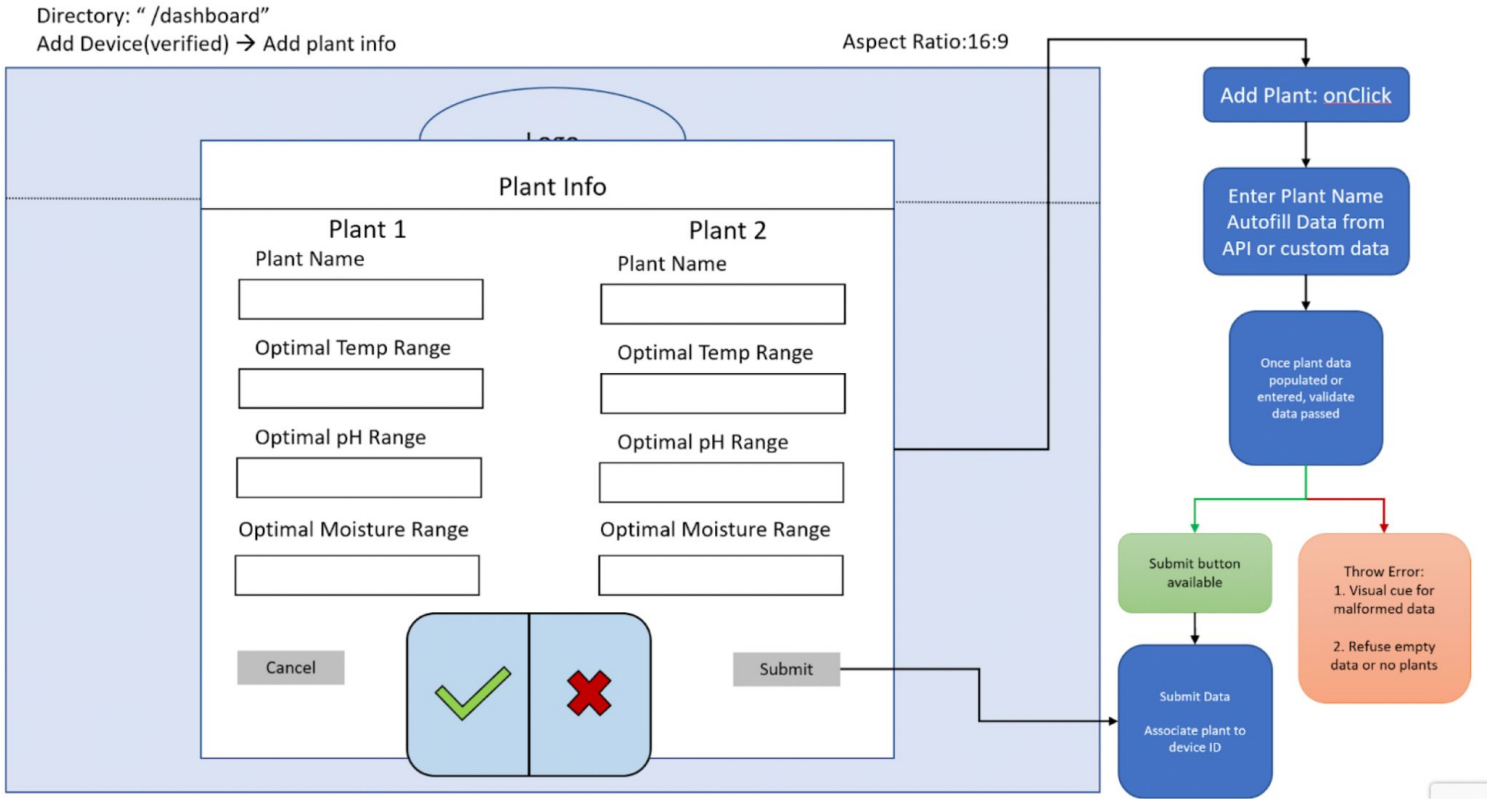


# Website Flow: Login/Sign Up

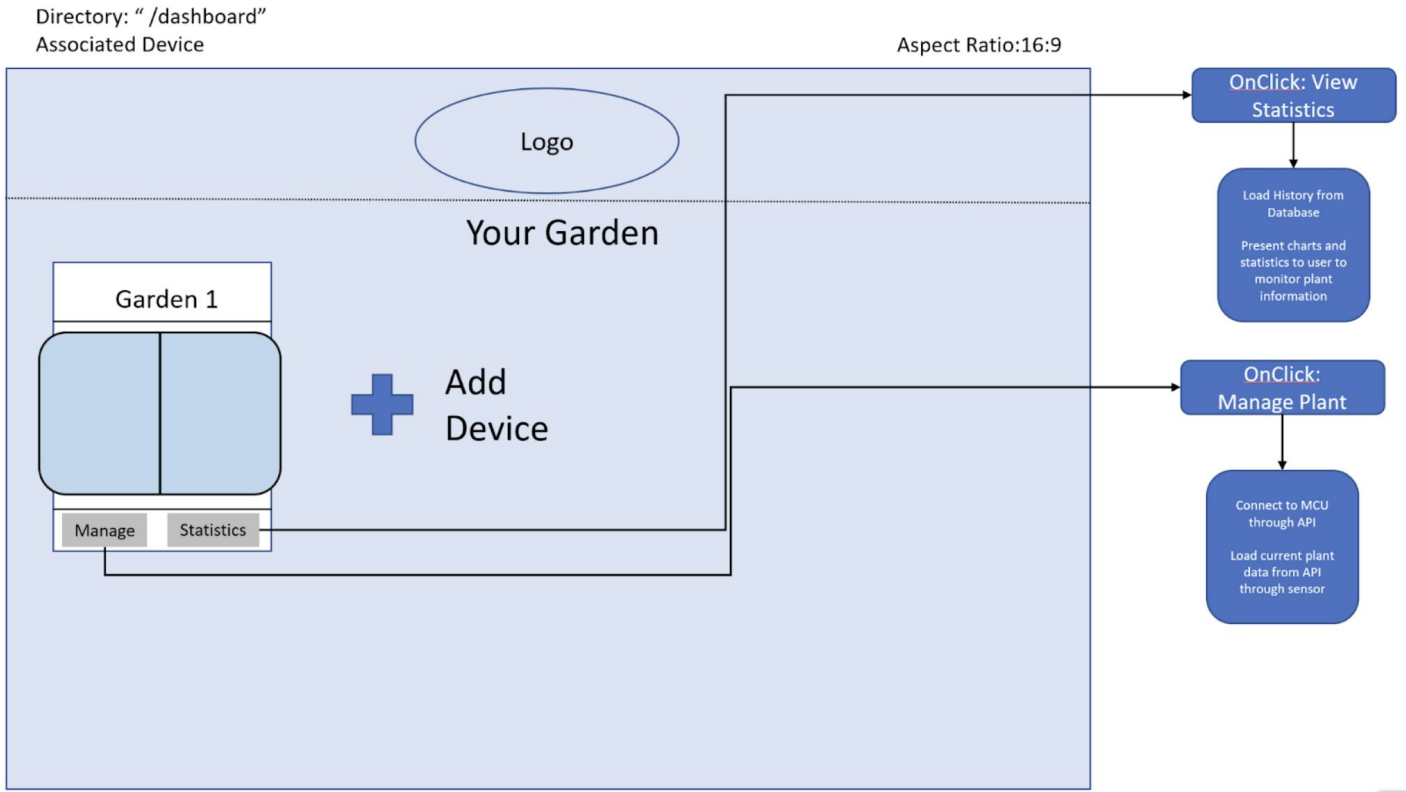
# Website Flow: Dashboard, Add Device



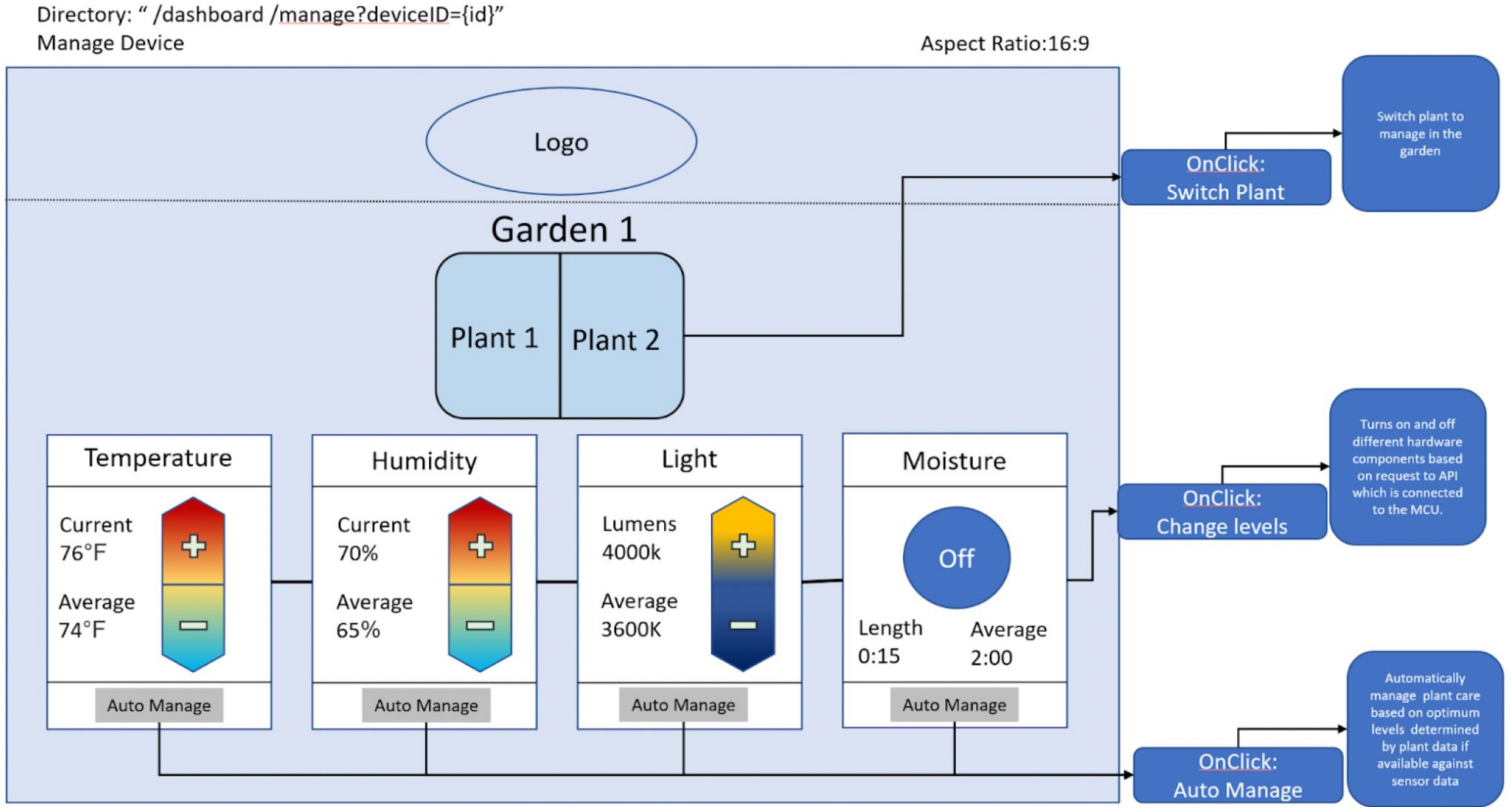
# Website Flow: Dashboard, Add Plant



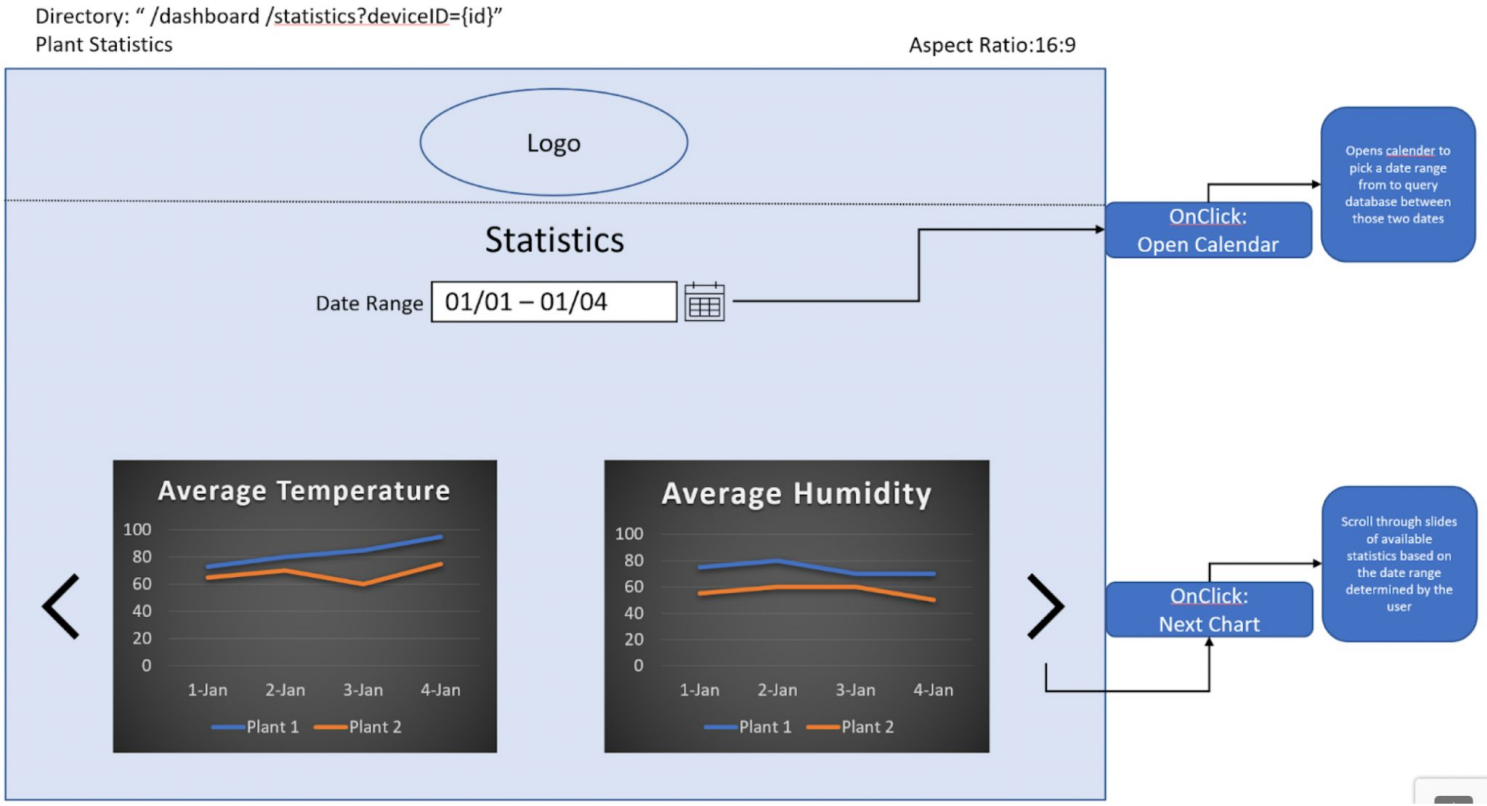
# Website Flow: Dashboard, Device Added



# Website Flow: Manage Plant



# Website Flow: Plant Statistic



# Administrative Content

# Work Distribution

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

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

# Budget

- ➔ Still subject to change
- ➔ May need to change components once PCB arrives
- ➔ Shows everything up to date as of September 12, 2022

| Item | Quantity | Price (estimated) | Column1 |
|---|---|---|---|
| Custom PCB | 3 | 60-90 | |
| Potting Mix | 1 | 5.97 | done |
| Planters | 6 | 20.88 | done |
| Miracle Grow Plant Food 2 lbs | 1 | 10.97 | done |
| Plant Seeds (Parsley and Basil) | 6 | 10.14 | done |
| PVC pipe + elbow fittings | 20 | 56.16 | done |
| Water Tank | 1 | 30.00 | |
| Reservoir | 1 | 30.00 | |
| Vinyl Tubing | 1 | 16.19 | done |
| Rayon (per yard) | 1 | $6-18 | |
| Light Sensor | 1 | $5-10 | |
| DC Dosing/Peristaltic Pump | 1 | 12.78 | done |
| CQRobot Liquid Sensor | 1 | 18.99 | done |
| Coospider UV Green Killer | 1 | 20.77 | done |
| PH Sensor | 1 | 35.88 | done |
| Weathershield (Humidity and Temp Sensor) | 1 | 74.91 | done |
| Wifi Module | 3 | 8.99 | done |
| LED Lights | 1 | 67.98 | done |
| Relay Module | 1 | $5-10 | |
| Arduino Mega R3 | 1 | 20.99 | |
| Nozzle | 1 | 15.00 | |
| DC Motor for Nozzle | 1 | $10-20 | |
| Circuit Elements (resistors, capacitors, etc.) | 1 | free | done |
| Camera | 1 | 25.99 | done |
| Pressure Diaphragm Pump | 1 | 20.95 | done |
| Barrel Plug (around 24 V) | 1 | 8.89 | done |
| | Total: | $600-700 | |

# Issues

- ➔ Board design: figuring out filters and component compatibility via pinouts
- ➔ Shortages
- ➔ Communication with components
- ➔ Acquiring datasheets for components

# Progress

- ➔ PCB designed
- ➔ GERBER file sent to supplier to be manufactured
- ➔ Basic skeletal design of enclosure done
- ➔ Website up and running but not yet completed
- ➔ Acquired all components for basic design (subject to change)

# Thanks!

Any questions?