GIRLS WITH ENERGY HACKATHON

GreenBot

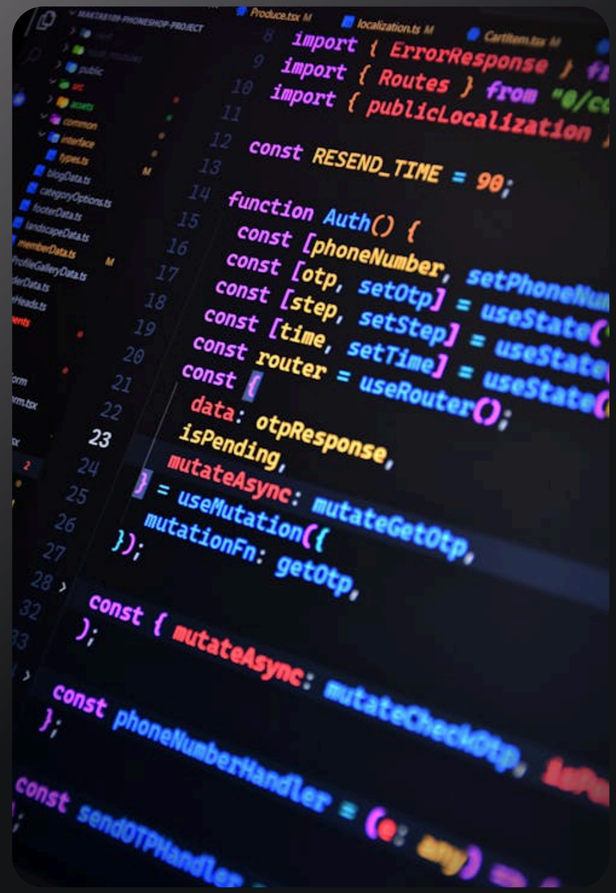
IMPROVING AIR QUALITY, ONE ROBOT AT A TIME.

JIZHEN HU, ZIQI TANG, ZI XIN ZHOU

GWE04

About Us

- Jizhen, Ziqi, Zi Xin
- 10th graders in Montreal, Canada
- Interested in computer science since a young age



Our Team

Jizhen Hu
Team Leader

Ziqi Tang
Team Member

Zi Xin Zhou
Team Member

Air Pollution: A Global Health Crisis

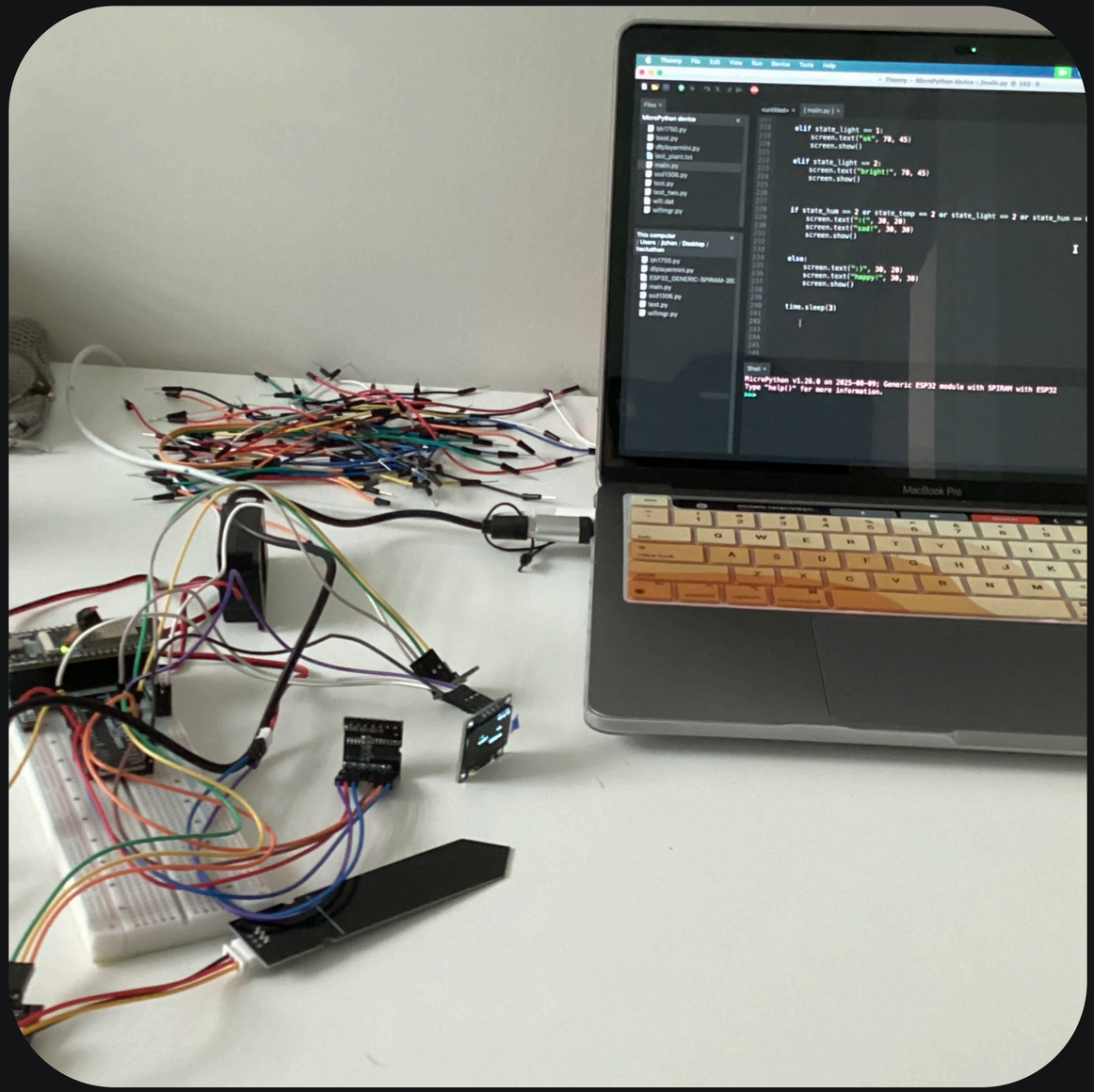
WHERE?
- Pollution: Chad, Bangladesh, Pakistan, India, etc.
- Cleanest: Australia, New Zealand, Finland, etc.

WHO'S AFFECTED?
- Citizens in countries with poor air quality
- Especially: Children, the elderly, people with existing health conditions (heart, lungs)

IMPACTS?
- Health: Chronic respiratory diseases, lung cancer, heart diseases, shorter life span
- Environmental: Acid rain formation, climate change acceleration, soil & water contamination

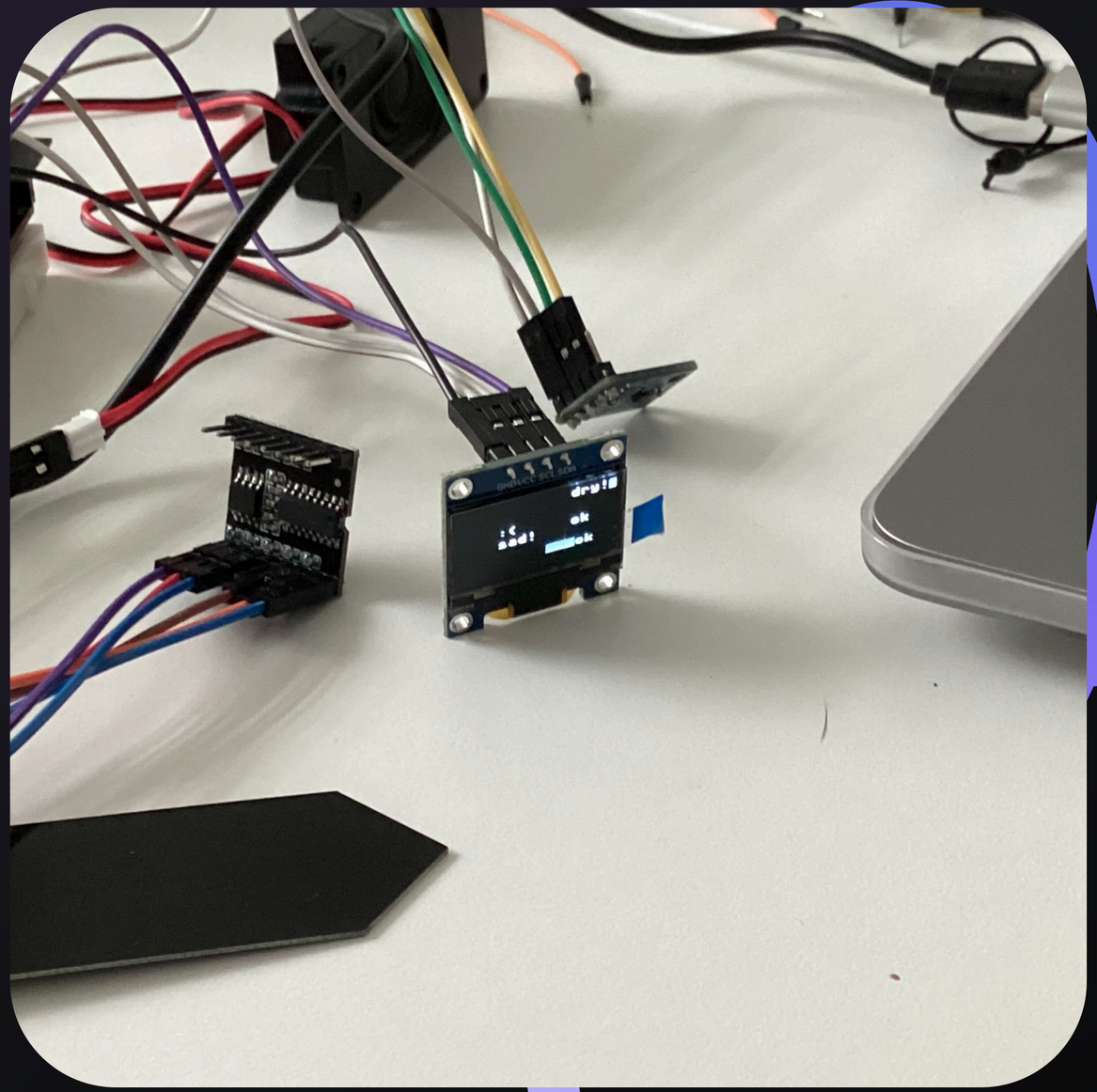
SOLUTION?
- Our robot!

A Solution for Tomorrow's Problems



What Does It Do?

- Teaches the user how to take care of any plant
- Displays any plant’s needs with an emoji
- Notifies the user when the plant needs water, or care (text and speaker)
  - User doesn’t need to remember about watering plants (user friendly)
- Plants stay healthy because of our bot
- User can access the web for more specifications



How It Works

- Detects a plant’s condition by running code stored internally
- Can connect to the Internet
  - It sends the plant’s data to the web by flask
    - The screen will display an emoji corresponding to the state (needs water, too much water, something else)
    - Pre-recorded message plays (DFPlayer)
    - The web will also display an emoji corresponding to the state (needs water, too much water, something else)
      - Plant state + corresponding emoji sent to the web, then generating a user-personalized message with ChatGPT API
        - Can analyze plant image to detect problems
- If it doesn’t, bot processes the data locally (wifi or web don’t answer)
  - Pre-recorded message plays (DFPlayer)
  - The screen will display an emoji corresponding to the state (needs water, too much water, something else)

PYTHON

```
        if value <= lum["low"]:
            return 0
        elif value >= lum["high"]:
            return 2
        elif lum["low"] < value < lum["high"]:
            return 1
        else:
            return 4


i2c = SoftI2C(scl=Pin(22), sda=Pin(21))
screen = ssd1306.SSD1306_I2C(128, 64, i2c)


state_hum = get_hum_soil()
state_temp = get_temp()
state_light = get_light()

if state_hum == 0:
    screen.text("dry!", 90, 0)
    screen.show()

elif state_hum == 1:
    screen.text("ok", 90, 0)
    screen.show()

elif state_hum == 2:
    screen.text("wet!", 90, 0)
    screen.show()
```

```
boot.py
dfplayermini.py
last_plant.txt
maiin.py
ssd1306.py
test.py
test_two.py
wifi.dat
wifimgr.py

This computer
/ Users / jizhen / Desktop / hackathon
bh1750.py
dfplayermini.py
ESP32_GENERIC-SPIRAM-20
main.py
ssd1306.py
test.py
wifimgr.py
```

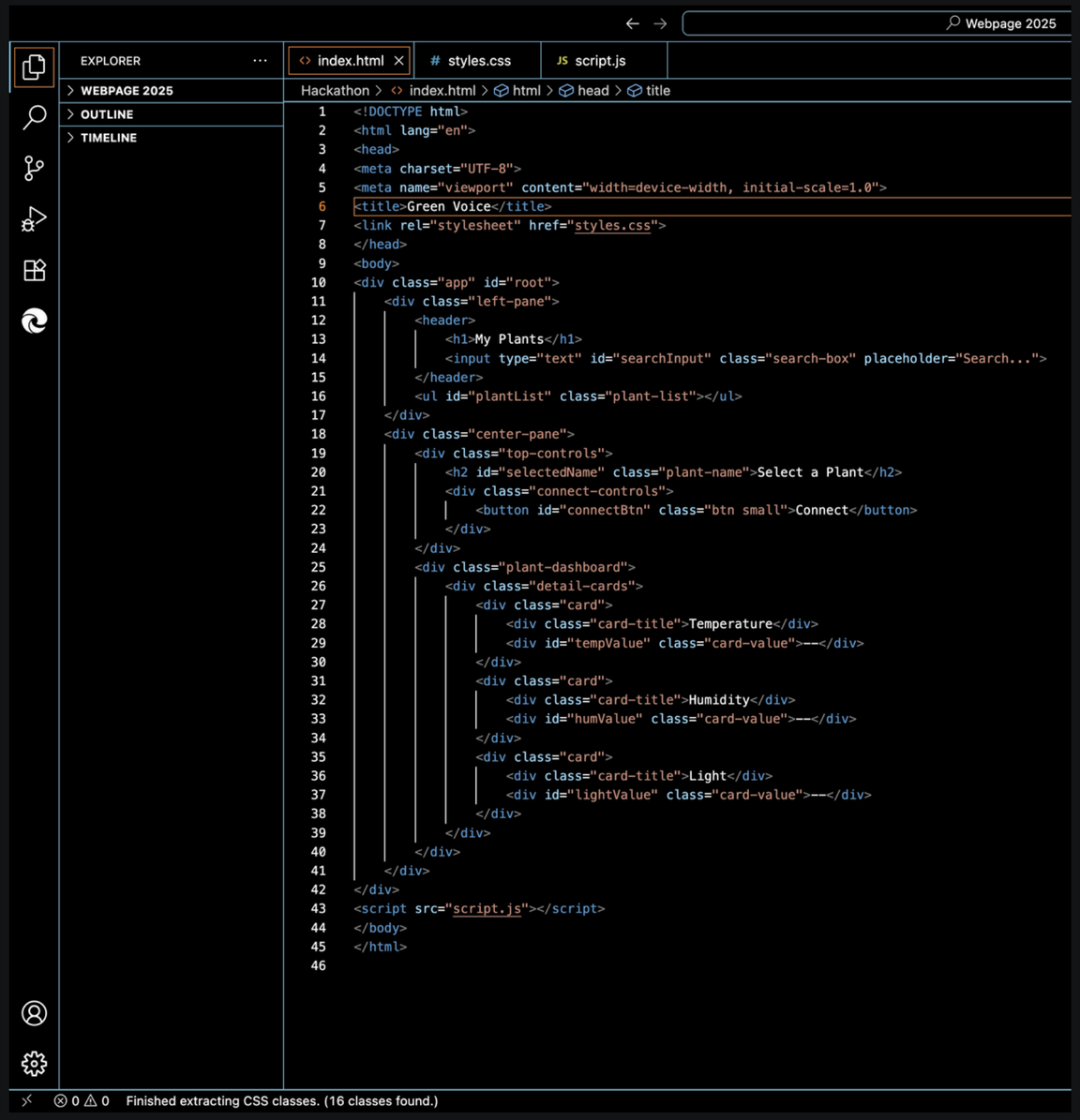
```
from ssd1306 import SSD1306_I2C
import ssd1306
import dht
from bh1750 import BH1750
import dfplayermini
import urequests
import wifimgr

plants_name = {
    "desert_cactus": {
        "soil": {"low": 3481, "high": 2457},
        "temp": {"low": 8, "high": 38},
        "light": {"low": 800, "high": 3000}
    },
    "succulent": {
        "soil": {"low": 3276, "high": 2048},
        "temp": {"low": 10, "high": 35},
        "light": {"low": 600, "high": 2500}
    },
    "tropical_fern": {
        "soil": {"low": 2048, "high": 410},
        "temp": {"low": 16, "high": 28},
        "light": {"low": 50, "high": 400}
    },
    "tropical_foliage": {
        "soil": {"low": 2457, "high": 819},
        "temp": {"low": 15, "high": 30},
        "light": {"low": 100, "high": 1500}
    },
    "mediterranean_herb": {
        "soil": {"low": 3071, "high": 1638},
        "temp": {"low": 12, "high": 32},
        "light": {"low": 800, "high": 2500}
    },
    "soft_herb": {
```

Impact

- Air quality will get better bit by bit
- Clean air = better health, less hospital visits, overall healthier population
- Better environment

HTML


Webpage 2025
EXPLORER
index.html
styles.css
script.js
WEBPAGE 2025
OUTLINE
TIMELINE
Hackathon > index.html > html > head > title
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Green Voice</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>

<header>
<h1>My Plants</h1>
<input type="text" id="searchInput" class="search-box" placeholder="Search...">
</header>
<ul id="plantList" class="plant-list">

<h2 id="selectedName" class="plant-name">Select a Plant</h2>

<button id="connectBtn" class="btn small">Connect</button>

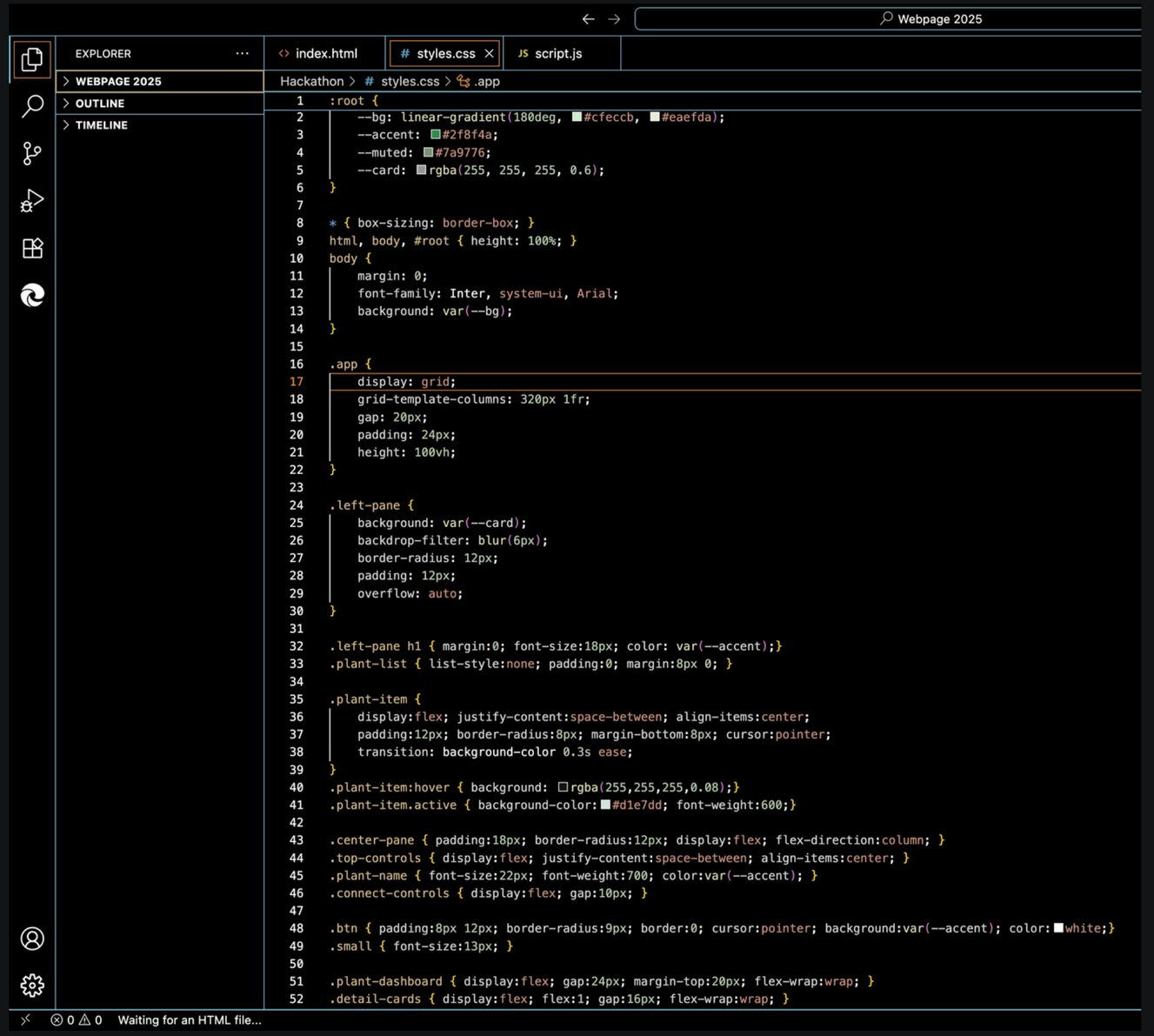
Temperature
--

Humidity
--

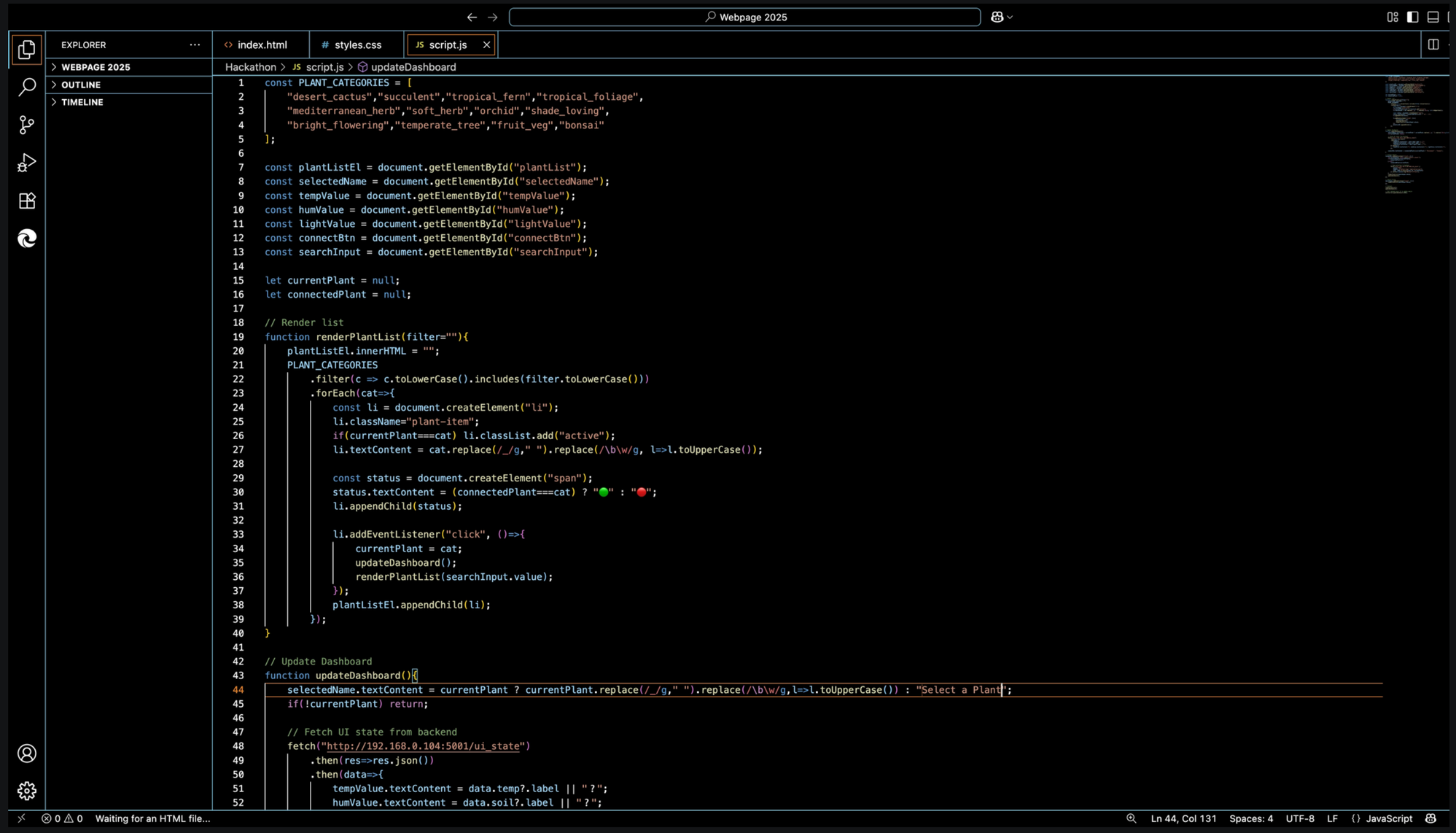
Light
--

<script src="script.js"></script>
</body>
</html>
⊗ 0 ⚠ 0 Finished extracting CSS classes. (16 classes found.)

CSS

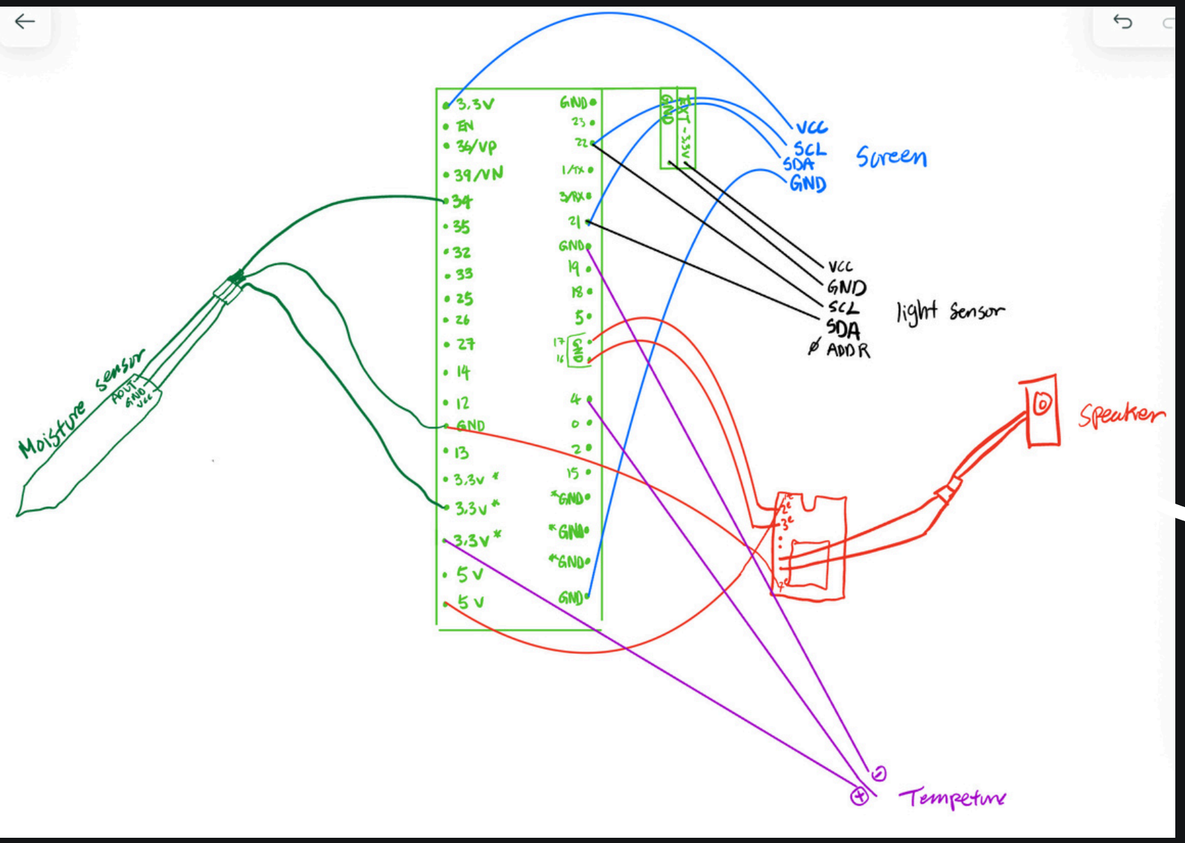

Webpage 2025
EXPLORER
index.html
styles.css
script.js
WEBPAGE 2025
OUTLINE
TIMELINE
Hackathon > styles.css > .app
:root {
--bg: linear-gradient(180deg, #cfeccb, #eaefda);
--accent: #2f8f4a;
--muted: #7a9776;
--card: rgba(255, 255, 255, 0.6);
}
* { box-sizing: border-box; }
html, body, #root { height: 100%; }
body {
margin: 0;
font-family: Inter, system-ui, Arial;
background: var(--bg);
}
.app {
display: grid;
grid-template-columns: 320px 1fr;
gap: 20px;
padding: 24px;
height: 100vh;
}
.left-pane {
background: var(--card);
backdrop-filter: blur(6px);
border-radius: 12px;
padding: 12px;
overflow: auto;
}
.left-pane h1 { margin:0; font-size:18px; color: var(--accent);}
.plant-list { list-style:none; padding:0; margin:8px 0; }
.plant-item {
display:flex; justify-content:space-between; align-items:center;
padding:12px; border-radius:8px; margin-bottom:8px; cursor:pointer;
transition: background-color 0.3s ease;
}
.plant-item:hover { background: rgba(255,255,255,0.08);}
.plant-item.active { background-color:#d1e7dd; font-weight:600;}
.center-pane { padding:18px; border-radius:12px; display:flex; flex-direction:column; }
.top-controls { display:flex; justify-content:space-between; align-items:center; }
.plant-name { font-size:22px; font-weight:700; color:var(--accent); }
.connect-controls { display:flex; gap:10px; }
.btn { padding:8px 12px; border-radius:9px; border:0; cursor:pointer; background:var(--accent); color:white;}
.small { font-size:13px; }
.plant-dashboard { display:flex; gap:24px; margin-top:20px; flex-wrap:wrap; }
.detail-cards { display:flex; flex:1; gap:16px; flex-wrap:wrap; }
⊗ 0 ⚠ 0 Waiting for an HTML file...

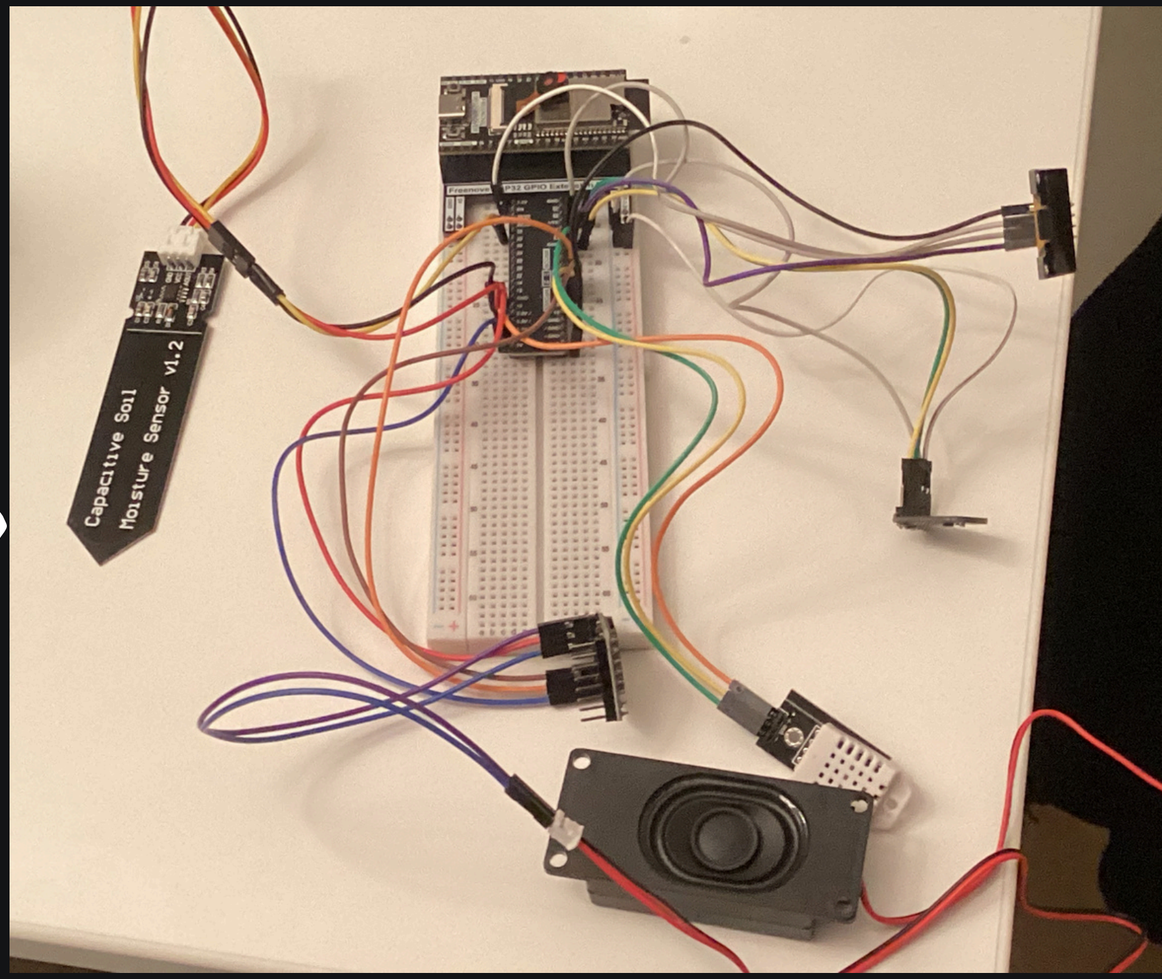
JAVASCRIPT


Webpage 2025
EXPLORER
index.html
styles.css
script.js
WEBPAGE 2025
OUTLINE
TIMELINE
Hackathon > script.js > updateDashboard
const PLANT_CATEGORIES = [
"desert_cactus","succulent","tropical_fern","tropical_foliage",
"mediterranean_herb","soft_herb","orchid","shade_loving",
"bright_flowering","temperate_tree","fruit_veg","bonsai"
];
const plantListEl = document.getElementById("plantList");
const selectedName = document.getElementById("selectedName");
const tempValue = document.getElementById("tempValue");
const humValue = document.getElementById("humValue");
const lightValue = document.getElementById("lightValue");
const connectBtn = document.getElementById("connectBtn");
const searchInput = document.getElementById("searchInput");
let currentPlant = null;
let connectedPlant = null;
// Render list
function renderPlantList(filter=""){
plantListEl.innerHTML = "";
PLANT_CATEGORIES
.filter(c => c.toLowerCase().includes(filter.toLowerCase()))
.forEach(cat=>{
const li = document.createElement("li");
li.className="plant-item";
if(currentPlant===cat) li.classList.add("active");
li.textContent = cat.replace(/_/g," ").replace(/\b\w/g, l=>l.toUpperCase());
const status = document.createElement("span");
status.textContent = (connectedPlant===cat) ? "🟢" : "🔴";
li.appendChild(status);
li.addEventListener("click", ()=>{
currentPlant = cat;
updateDashboard();
renderPlantList(searchInput.value);
});
plantListEl.appendChild(li);
});
}
// Update Dashboard
function updateDashboard(){
selectedName.textContent = currentPlant ? currentPlant.replace(/_/g," ").replace(/\b\w/g,l=>l.toUpperCase()) : "Select a Plant";
if(!currentPlant) return;
// Fetch UI state from backend
fetch("http://192.168.0.104:5001/ui_state")
.then(res=>res.json())
.then(data=>{
tempValue.textContent = data.temp?.label || "?";
humValue.textContent = data.soil?.label || "?";
Waiting for an HTML file...
Ln 44, Col 131
Spaces: 4
UTF-8
LF
JavaScript

Breadboard circuit:

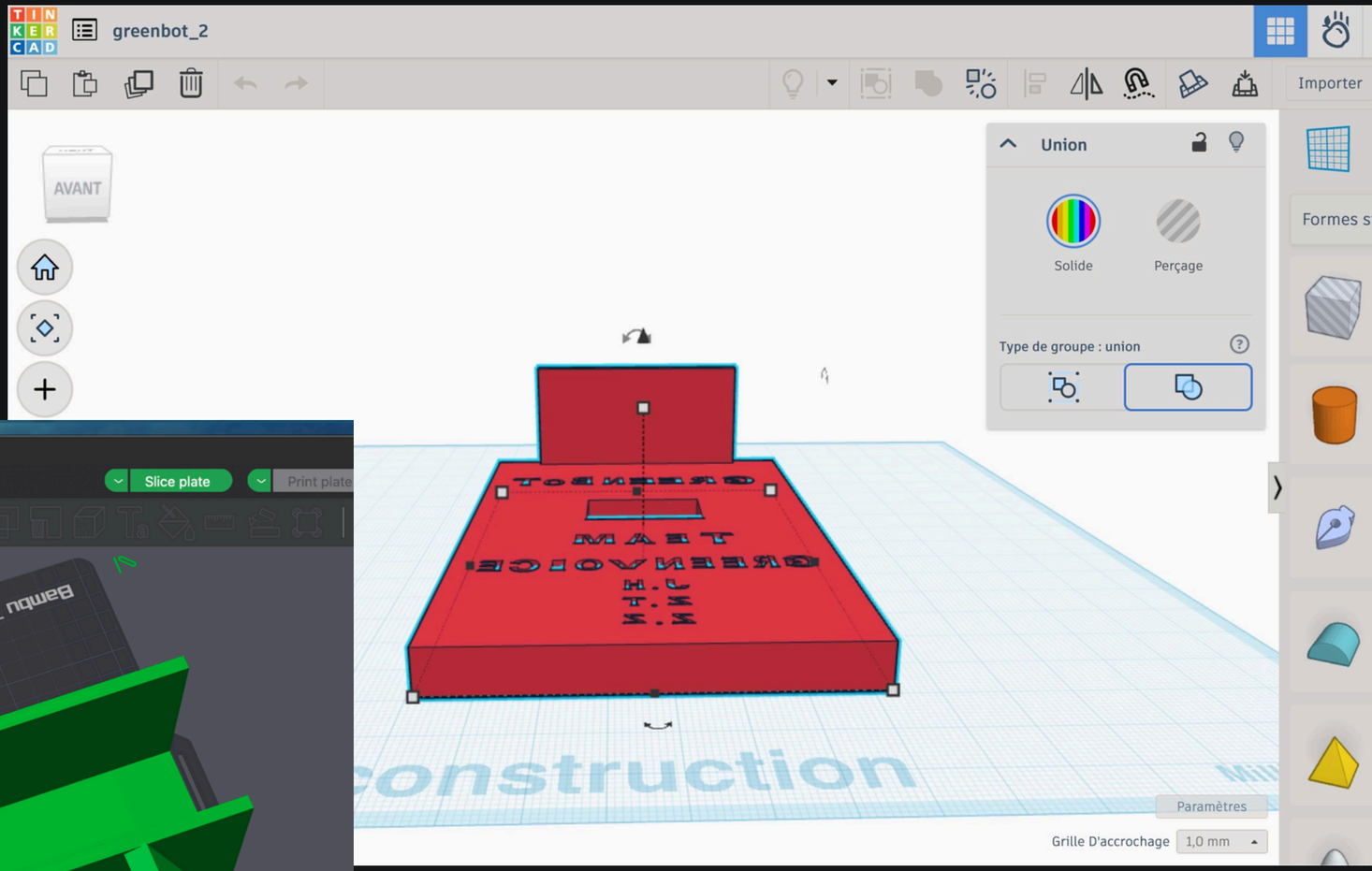
A microcontroller(ESP32), a soil moisture sensor, a temperature, humidity sensor and a small speaker by dfplayer

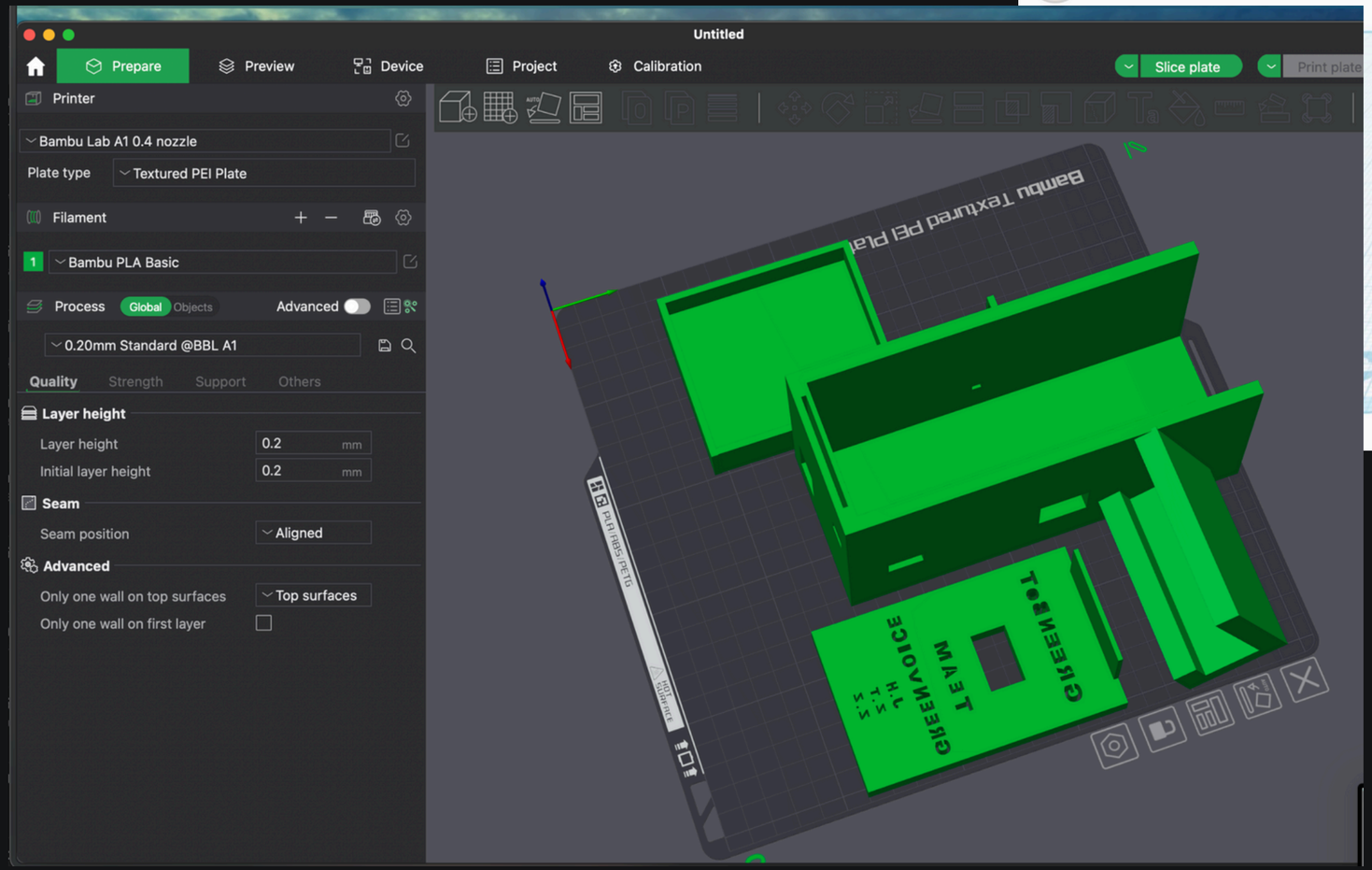




3D PRINTING

-Tinkercad





Thank You