Automation of Seed Planting and Crop Watering with Monitoring System Using Mobile Application

A Thesis
Presented to the Faculty of the
Department of Electronics and Communications Engineering Gokongwei College of Engineering
De La Salle University

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Engineering

by

ALBALADEJO, Ryan Jonathan D.
ESPIRITU, Danielle Marie V.
GO, Marien Allison M.

August 2020

De La Salle University

Thesis Approval Sheet

This thesis entitled **Automation of Seed Planting and Crop Watering with Monitoring System Using Mobile Application**, prepared and submitted by:

ALBALADEJO, Ryan Jonathan D.
ESPIRITU, Danielle Marie V.
GO, Marien Allison M.

with group number REST-1-1819-C1 in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Engineering** (**BS-CPE**) has been examined and is recommended for acceptance and approval.

PANEL OF EXAMINERS

_____
Engr. Antonette C. Roque
Chair

_____
Engr. Antoniette Ilahi
Member

_____
Engr. Armie E. Pakzad
Member

_____
Dr. Edwin Sybingco
Adviser

Date: _____

ACKNOWLEDGMENT

First and foremost, the researchers would like to thank the university for giving us the chance to implement this proposed system. Also, the group would like to give gratitude to Dr. Edwin Sybingco for guiding us for each process taken.

ABSTRACT

This system presents an electronic and mechanical platform that performs basic crop planting while considering the spacing between seeds with particular rows and specific columns. In this work, the size of the land is taken into consideration through which the machine can calculate its precise operations. All the operations are monitored and controlled by a microcontroller with the use of sensors and image processing. The driving force behind an automated machine is to reduce the human interference in the field of agriculture and to make it cost effective. The progress and status of the crops can be monitored through a mobile application provided by the researchers. The mobile application will aid the user in monitoring the crops by displaying the actual field, the progress of the crops in terms of growth, and the moisture of the soil using the camera and sensors attached to the robot.

Index Terms—xyz robot, indoor farming, wireless communication, mobile application, crop monitoring.

De La Salle University

TABLE OF CONTENTS

Thesis Approval Sheet ii

ACKNOWLEDGMENT iii

ABSTRACT iv

TABLE OF CONTENTS v

List of Figures xii

List of Tables xxiv

Chapter 1 INTRODUCTION 1

1.1 Background of the Study 2
1.2 Prior Studies 2
1.3 Problem Statement 4
1.4 Objectives 5
1.4.1 General Objective(s) 5
1.4.2 Specific Objectives 5
1.5 Significance of the Study 5
1.6 Assumptions, Scope and Delimitations 6
1.6.1 Assumptions 6
1.6.2 Scope 6
1.6.3 Delimitations 6
1.7 Description and Methodology of the Thesis 7
Overview 10

Chapter 2 LITERATURE REVIEW 11

2.1 Existing Work 13
2.1.1 Agricultural Robot 16
2.1.1.1 Multipurpose Autonomous Agricultural Robot 16
2.1.1.2 Indoor robot gardening: Design and implementation 17
2.1.1.3 Agricultural Robot for Automatic Ploughing and Seeding 18
2.1.1.4 Design of a Controlled Environment Agricultural Plant Inspection Robot 19
2.1.2 Smart Agricultural System 21
2.1.2.1 Design of Intelligent Plant Growth Cabinet Environment Monitoring and Control System 21
2.1.2.2. AgriSys: A Smart and Ubiquitous Controlled Environment Agriculture System 22
2.1.2.3 Plant Growth Using Automatic Control System under LED, Grow, and Natural Light 23
2.1.2.4 Smart Water Dripping System for Agriculture/Farming 25
2.1.3 Brassica Rapa 27
2.1.3.1 Growth and Yield of Pechay (Brassica Pekinesis) as Affected by Green Herds Organic Based Foliar Fertilizer 27
2.1.3.2 Response of Pechay (Brassica napus L.) to Different Levels of Compost Fertilizer 27
2.1.3.3 Pechay Production Guide 29
2.1.3.4 Pechay Open Field Production 30
2.1.3.5 Growth Response of Brassica Rapa on the Different Wavelength of Light 30
2.1.3.6 Retail Price 30
2.1.3.7 Characteristics 31
2.1.4 Grow Room 32
2.2 Lacking in the Approaches 33
2.3 Summary 33

Chapter 3 THEORETICAL CONSIDERATION 35

3.1 Soil Moisture Sensor 36
3.2 Temperature and Humidity Sensor 37
3.3 Stepper Motor 38
3.4 Crop 38
3.5 Summary 40

Chapter 4 DESIGN CONSIDERATION 41

4.1 Overview 44
4.2 Growth Chamber 45
4.2.1 Chamber Assembly 45
4.2.1.1 Cabinet 47
4.2.1.2 Aluminum Foil Insulation 48
4.2.2 Electronics System 49
4.2.2.1 System Circuit 50
4.2.2.1.1 Thermal Management 50
4.2.2.1.2 16 Channel Relay Module 50
4.2.2.1.3 Terminal Block 52
4.2.2.2 Power Supplies 52
4.2.2.3 Raspberry Pi 4B 53
4.2.2.3.1 Step-Down Voltage Regulator 54
4.2.2.3.2 Libraries 54
4.2.2.3.2.1 Pygame 54
4.2.2.3.2.2 Datetime 55
4.2.2.3.2.3 Numpy 55
4.2.2.3.2.4 Plantcv. 55
4.2.2.3.2.5 Cv2 55
4.2.2.3.2.6 Os 55
4.2.2.3.2.7 Sys 55
4.2.2.3.2.8 Rpi.gpio 56
4.2.3 Lighting System 56
4.2.3.1 White LED Light 56
4.2.3.2 Grow Lights 56
4.2.4 Air Circulation System 59
4.3 Cartesian Robot System 62
4.3.1 Cartesian Robot Assembly 63
4.3.1.1 System Components 63
4.3.1.1.1 Stepper Motors 63
4.3.1.1.2 DRV 8825 64

4.3.1.1.3 Limit Switches 65
4.3.1.2 Gantry 66
4.3.1.3 Z-Axis 68
4.3.1.4 Cross Slide 68
4.3.2 Plant Box 69
4.3.3 Seeding System 70
4.3.3.1 Components 70
4.3.3.2 Circuit 71
4.3.3.3 Software 72
4.3.3.4 Uniform and Variable Seed Distribution 78
4.3.4 Irrigation System 79
4.3.4.1 Components 80
4.3.4.2 Circuit 80
4.3.4.3 Software 81
4.3.4.3.1 Raspberry Pi Automatic Irrigation Function 82
4.3.4.3.2 Raspberry Pi Adaptive Irrigation Function 83
4.3.4.3.3 Arduino Irrigation System Function 83
4.4 Connectivity System 85
4.4.1 Django Framework 85
4.4.1.1 MVT Pattern 85
4.4.1.1.1 Models 86
4.4.1.1.2 View 87
4.4.1.1.3 Template 88
4.4.1.2 Database 88
4.4.1.3 Asynchronous Connectivity 90
4.4.2 Mobile Application 93
4.4.2.1 Dashboard Page 93
4.4.2.1.1 Side Navigation Bar 95
4.4.2.1.1.1 Mode Selector 96
4.4.2.1.1.2 GrowSmart Controller 99
4.4.2.1.2 Monitoring System 101
4.4.2.1.2.1 Sensors 101
4.4.2.1.2.2 Vision System 102
4.4.2.2 Database Page 103

4.4.2.3 Sensors Page 112
4.4.2.4 Development of Mobile Application 114
4.4.3 ZeroTier One 115
4.5 Monitoring System 116
4.5.1 Vision System 116
4.5.1.1 Logitech C310 Camera 116
4.5.1.2 Software 117
4.5.2 Sensors 122
4.5.2.1 Temperature and Humidity 125
4.5.2.2 Soil Moisture 126
4.5.2.3 Threshold Values 127
4.6 Summary 128

Chapter 5 METHODOLOGY 130

5.1 Cartesian Robot System 131
5.1.1 Seeding System 131
5.1.1.1 Uniform and Variable Seed Distribution 132
5.1.2 Irrigation System 134
5.1.2.1 Automatic Irrigation Function 136
5.1.2.2 Adaptive Irrigation Function 136
5.1.3 Robot Accuracy Testing 137
5.2 Monitoring System 143
5.2.1 Vision System 145
5.2.2 Sensors 152
5.2.2.1 Trial 1 154
5.2.2.2 Trial 2 157
5.3 Cost Analysis 161
5.4 Summary 166

Chapter 6 RESULTS AND DISCUSSION 169

6.1 Cartesian Robot System 170
6.1.1 Seeding Accuracy 170
6.1.2 Uniform and Variable Seed Distribution 177
6.1.3 Irrigation System 180
6.2 Mobile Application Monitoring System 180

6.2.1 Vision System 187
6.2.2.1 Trial 1 188
6.2.2.2 Trial 2 189
6.2.2 Sensors 190
6.2.2.1 Trial 1 190
6.2.2.2 Trial 2 193
6.3 Cost Analysis 194
6.4 Summary 195

Chapter 7 CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DIRECTIVES 196

7.1 Concluding Remarks 197
7.2 Contributions 197
7.3 Recommendations 199
7.4 Future Prospects 199

REFERENCES 201

Appendix A ANSWERS TO QUESTIONS TO THIS THESIS 206

Appendix B SPACER 210

Appendix C BEAM FITTING 211

Appendix D STEPPER MOTOR HOUSING 212

Appendix E BASE FITTING 213

Appendix F ALUMINUM EXTRUSION 214

Appendix G DJANGO VIEWS CODE 215

Appendix H BASE HTML 246

Appendix I DATABASE HTML 320

Appendix J ARDUINO CODES 325

Appendix K MONITORING SYSTEM DATA 342

Appendix L PRO1 PANEL COMMENTS AND REVISIONS 380

Appendix M PRO3 PANEL COMMENTS AND REVISIONS 381

List of Figures

Fig 1.1 Main Flow of the Gardening System 7
Fig 1.2 Concept Map 8
Fig 1.3 Flowchart of the System's General Algorithm 9
Fig. 2.1 Indoor Robot 18
Fig. 2.2 STM32F103 22
Fig. 2.3 System Flowchart 24
Fig 2.4 Room Setup 25
Fig. 2.5 Block Diagram 26
Fig. 2.6 Price of Pechay for 5 years 31
Fig. 3.1 Soil Moisture Sensor 36
Fig. 3.2 Schematic Diagram of the Soil Moisture Sensor Module 36
Fig. 3.4 DHT22 Sensor Module 37
Fig. 3.5 Stepper Motor Connection 38
Fig. 4.1 High-level Block Diagram of GrowSmart 44
Fig. 4.2 Growth Chamber Closed Front View 46
Fig. 4.3 Growth Chamber Open Front View 46
Fig. 4.4 Growth Chamber Side View 47
Fig. 4.5 Cabinet Sketch Structure 48
Fig. 4.6 Electronics System 49
Fig. 4.7 Relay Wiring to Raspberry Pi 51
Fig. 4.8 Relay Wiring of LED and Fan with Raspberry Pi 51

Fig. 4.9 Relay Wiring of Water and Vacuum Pump with Arduino Mega 52
Fig. 4.10 Power Supplies 53
Fig. 4.11 Raspberry Pi Connection with Submodules of System 54
Fig. 4.12 LED Light in Operation 56
Fig. 4.13 Grow Lights Distance Table 57
Fig. 4.14 Red:Blue Ratio for Plant 57
Fig. 4.15 Grow Lights 58
Fig. 4.16 Grow Lights in Operation 59
Fig. 4.17 Fan for Air Circulation 60
Fig. 4.18 Growth Chamber Ventilation 61
Fig. 4.19 Air Circulation System Sketch 61
Fig. 4.20 XYZ Movement 62
Fig. 4.21 Prototype 63
Fig. 4.22 CNC Shield 64
Fig. 4.23 DRV8825 Wiring Diagram 64
Fig. 4.24 Limit Switch Connection to CNC Shield 65
Fig. 4.25 Tuning the DRV8825 65
Fig. 4.26 Limit Switch Setup 66
Fig. 4.27 3D Printed Base Fitting 67
Fig. 4.28 3D Printed Beam Fitting 67
Fig. 4.29 3D Printed Stepper Motor Housing 68
Fig. 4.30 Z-Axis 68
Fig. 4.31 Plant Box Sketch 69
Fig. 4.32 Seed Container 70

Fig. 4.33 Seeder Setup 71
Fig. 4.34 Circuit Used for Seeding 72
Fig. 4.35 Commands Screenshot 73
Fig. 4.36 Mode Selection Screenshot 74
Fig. 4.37 Controls Screenshot 74
Fig. 4.38 Three Commands Available 76
Fig. 4.39 beginCalibration() skeleton 76
Fig. 4.40 Sample code for Individual axes in beginCalibration() 77
Fig. 4.41 beginSeedDistribution() code 78
Fig. 4.42 Declaration of Pins in Arduino IDE under PinConfiguration.h 79
Fig. 4.43 Water Nozzle 80
Fig. 4.44 Water Pump 80
Fig. 4.45 Rubber Tubing 80
Fig. 4.46 Irrigation System Circuit 81
Fig. 4.47 Flowchart of AJAX Function for the Automatic Irrigation System 82
Fig. 4.48 Robot Travel Direction on 5 Columns x 2 Rows Setting 84
Fig. 4.49 Robot Travel Direction on 5 Columns x 3 Rows Setting 84
Fig. 4.50 Django Framework Flowchart 85
Fig. 4.51 Snippet of the Models Class of GrowSmart's Application 86
Fig. 4.52 Snippet of Views Code 88
Fig. 4.53 MySQL Tables 89
Fig. 4.54 Sensor Status Table 90
Fig. 4.55 Device Status 90
Fig. 4.56 Plant Information Table 90

Fig. 4.57 Importing jQuery Library to Code 90
Fig. 4.58 Ajax - Django Communication flowchart 91
Fig. 4.59 Flowchart of AJAX function when the oneWhiteLed button is clicked 92
Fig. 4.60 Views function when the oneWhiteLed button is clicked 92
Fig. 4.61 Dashboard lights status 93
Fig. 4.62 GrowSmart Mobile Application Icons 93
Fig. 4.63 GrowSmart Mobile Application Dashboard Page 94
Fig. 4.64 Dashboard Page Mode Selector 95
Fig. 4.65 GrowSmart Mode 1 Activated 97
Fig. 4.66 GrowSmart Mode 2 Activated 98
Fig. 4.67 GrowSmart Mode 3 Activated 98
Fig. 4.68 GrowSmart Mode 4 Activated 99
Fig. 4.69 GrowSmart Controller Section 100
Fig. 4.70 Days Checker Function Snippet 101
Fig. 4.71 GrowSmart Application Days Checker Section 101
Fig. 4.72 GrowSmart Application Sensor Readings Section 101
Fig. 4.73 GrowSmart Application Vision System Section 103
Fig. 4.74 GrowSmart Application Database Page 104
Fig. 4.75 GrowSmart Application Sensor Charts Section 106
Fig. 4.76 GrowSmart Application Plant Growth Charts Section 107
Fig. 4.77 Temperature Chart (Mobile Application) 108
Fig. 4.78 Temperature Chart (Mobile Application) 108
Fig. 4.79 Soil Moisture Chart (Mobile Application) 109
Fig. 4.80 Plant 1 Chart (Mobile Application) 109

Fig. 4.81 Plant 2 Chart (Mobile Application) 109
Fig. 4.82 Plant 3 Chart (Mobile Application) 110
Fig. 4.83 Plant 4 Chart (Mobile Application) 110
Fig. 4.84 Plant 5 Chart (Mobile Application) 110
Fig. 4.85 Plant 6 Chart (Mobile Application) 111
Fig. 4.86 Plant 7 Chart (Mobile Application) 111
Fig. 4.87 Plant 8 Chart (Mobile Application) 111
Fig. 4.88 Plant 9 Chart (Mobile Application) 112
Fig. 4.89 Plant 10 Chart (Mobile Application) 112
Fig. 4.90 GrowSmart Application Sensors Page 113
Fig. 4.91 Internet Permission 114
Fig. 4.92 Loading URL 114
Fig. 4.94 ZeroTier Central Network Connected Devices Section 115
Fig. 4.95 ZeroTier App 116
Fig. 4.96 Camera Setup 117
Fig. 4.97 JavaScript Function that calls the different automated functions every minute 118
Fig. 4.98 Flowchart of AJAX function called at 7:50 AM and 8:00AM Everyday 118
Fig. 4.99 Flowchart of AJAX function called at 7:55AM Everyday 119
Fig. 4.100 RGB to LAB 120
Fig. 4.101 Threshold 120
Fig. 4.102 Detecting Objects 121
Fig. 4.103 Multiple ROI 121
Fig. 4.104 Detecting Area per ROI 122

Fig. 4.105 PlantCV Results 122
Fig. 4.106 Flowchart of Ajax Function Called Every Hour 123
Fig. 4.107 Code of Ajax Function Called Every Hour 123
Fig. 4.108 Views Function Snippet Called by AJAX 124
Fig. 4.109 DHT22 Placing in PlantBox 125
Fig. 4.110 Temperature Range 125
Fig. 4.111 Adaptive Irrigation System Conditional Code 126
Fig. 4.112 MCP3008 Pinout 126
Fig. 4.113 Python code for soil moisture sensor and MCP3008 127
Fig. 4.114 MCP3008 Connection to Raspberry Pi 127
Fig. 5.1 Seeder Reach 131
Fig. 5.2 Syringe Taped to Z-Axis with Tube 132
Fig. 5.3 Declaration of Specifications in Arduino IDE under PlantCoordinates.h 133
Fig. 5.4 Declaration of Mode Pins 133
Fig. 5.5 Mode 00 133
Fig. 5.6 Mode 01 134
Fig. 5.7 Mode 10 134
Fig. 5.8 Mode 11 134
Fig. 5.9 Irrigation 135
Fig. 5.10 Automatic and Adaptive Irrigation Flowchart 136
Fig. 5.11 Canvas for XYZ Accuracy Testing 138
Fig. 5.12 Desmos Plotting of Canvas 138
Fig. 5.13 Robot Accuracy Testing Setup 1 139
Fig. 5.14 Robot Accuracy Testing Setup 2 140

Fig. 5.15 Constants.h header file snippet 140
Fig. 5.16 Division and Numbering of Planting Grounds 141
Fig. 5.17 Set-up for Writing Material Along Seed Injector 141
Fig. 5.18 Result of Seeding After 5 Trials Indicated by Difference of Colors 142
Fig. 5.19 Distance Formula 142
Fig. 5.20 Mean Formula 143
Fig. 5.21 Standard Deviation Formula 143
Fig. 5.22 Monitoring System Flowchart 144
Fig. 5.23 Image Capturing and Processing Flowchart 145
Fig. 5.24 Day 1 145
Fig. 5.25 Day 6 146
Fig. 5.26 Day 11 146
Fig. 5.27 Day 16 146
Fig. 5.28 Day 21 146
Fig. 5.29 Day 26 147
Fig. 5.30 Day 31 147
Fig. 5.31 Day 36 147
Fig. 5.32 Drawing ROIs for Image Labelling 148
Fig. 5.33 Reading Pixel Image 149
Fig. 5.34 Category Results 149
Fig. 5.35 MATLAB Results 150
Fig. 5.36 PlantCV and MATLAB Growth Chart 151
Fig. 5.37 Monitoring System with Sensors Flowchart 152
Fig. 5.38 Pearson Correlation Coefficient Formula 153

Fig. 5.39 Scenario Soil Moisture Dry 1 154
Fig. 5.40 Scenario Soil Moisture Moist 2 154
Fig. 5.41 Scenario Soil Moisture Wet 3 155
Fig. 5.42 Scenario Low Temperature & Normal Humidity 4 155
Fig. 5.43 Scenario High Temperature & Normal Humidity 155
Fig. 5.44 DHT Left 156
Fig. 5.45 DHT Right 156
Fig. 5.46 Soil Moisture Sensor 157
Fig. 5.47 Day 1 8AM 157
Fig. 5.48 Day 1 12PM 157
Fig. 5.49 Day 1 6PM 157
Fig. 5.50 Day 2 8AM 157
Fig. 5.51 Day 2 12PM 157
Fig. 5.52 Day 2 6PM 157
Fig. 5.53 Day 3 8AM 158
Fig. 5.54 Day 3 12PM 158
Fig. 5.55 Day 3 6PM 158
Fig. 5.56 Day 4 8AM 158
Fig. 5.57 Day 4 12PM 158
Fig. 5.58 Day 4 6PM 158
Fig. 5.59 Day 5 8AM 158
Fig. 5.60 Day 5 12PM 158
Fig. 5.61 Day 5 6PM 158
Fig. 5.62 Day 6 8AM 158

Fig. 5.63 Day 6 12PM 158
Fig. 5.64 Day 6 6PM 158
Fig. 5.65 Day 7 8AM..... 159
Fig. 5.66 Day 7 12PM 159
Fig. 5.67 Day 7 6PM 159
Fig. 5.68 Day 8 8AM..... 159
Fig. 5.69 Day 8 12PM 159
Fig. 5.70 Day 8 6PM 159
Fig. 5.71 Day 9 8AM..... 159
Fig. 5.72 Day 9 12PM 159
Fig. 5.73 Day 9 6PM 159
Fig. 5.74 Day 10 8AM..... 159
Fig. 5.75 Day 10 12PM 159
Fig. 5.76 Day 10 6PM 159
Fig. 5.77 Day 11 8AM..... 160
Fig. 5.78 Day 11 12PM 160
Fig. 5.79 Day 11 6PM 160
Fig. 5.80 Day 12 8AM..... 160
Fig. 5.81 Day 12 12PM 160
Fig. 5.82 Day 12 6PM 160
Fig. 5.83 Day 13 8AM..... 160
Fig. 5.84 Day 13 12PM 160
Fig. 5.85 Day 13 6PM 160
Fig. 5.86 Day 14 8AM..... 160

Fig. 5.87 Day 14 12PM 160
Fig. 5.88 Day 14 6PM 160
Fig. 5.89 Plant Weight 164
Fig. 5.90 Plant Height 165
Fig. 5.91 Plant True Leaves 165
Fig. 6.1 Digitized Plotting of Seeding Results on Desmos 170
Fig. 6.2 Planting Ground No. 1 Centered at (25, 15) 171
Fig. 6.3 Planting Ground No. 2 Centered at (40, 15) 171
Fig. 6.4 Planting Ground No. 3 Centered at (55, 15) 172
Fig. 6.5 Planting Ground No. 4 Centered at (70, 15) 172
Fig. 6.6 Planting Ground No. 5 Centered at (85, 15) 173
Fig. 6.7 Planting Ground No. 6 Centered at (85, 35) 173
Fig. 6.8 Planting Ground No. 7 Centered at (70, 35) 174
Fig. 6.9 Planting Ground No. 8 Centered at (55, 35) 174
Fig. 6.10 Planting Ground No. 9 Centered at (40, 35) 175
Fig. 6.11 Planting Ground No. 10 Centered at (25, 35) 175
Fig. 6.12 Serial Monitor Output 177
Fig. 6.13 Serial Monitor Output 178
Fig. 6.14 Mode 00 178
Fig. 6.15 Mode 01 179
Fig. 6.16 Mode 10 179
Fig. 6.17 Mode 11 179
Fig. 6.18 Soil Moisture Data from Days 1-4 of Testing from the Database 180
Fig. 6.19 GrowSmart Mobile Application Dashboard 181

Fig. 6.20 Growth Monitor 182
Fig. 6.21 Sensor Monitor 182
Fig. 6.22 Day 1 183
Fig. 6.23 Day 2 183
Fig. 6.24 Day 3 183
Fig. 6.25 Day 4 184
Fig. 6.26 Day 5 184
Fig. 6.27 Day 6 184
Fig. 6.28 Day 7 185
Fig. 6.29 Day 8 185
Fig. 6.30 Day 9 185
Fig. 6.31 Day 10 186
Fig. 6.32 Day 11 186
Fig. 6.33 Day 12 186
Fig. 6.34 Day 13 187
Fig. 6.35 Day 14 187
Fig. 6.36 Plant Growth Database and Charts 348
Fig. 6.37 Sensors Database and Charts 355
Fig. 6.38 Plants 1-10 Growth Chart (Trial 1) 188
Fig. 6.39 Plant 1 Growth Rate Chart 356
Fig. 6.40 Plant 2 Growth Rate Chart 357
Fig. 6.41 Plant 3 Growth Rate Chart 357
Fig. 6.42 Plant 4 Growth Rate Chart 357
Fig. 6.43 Plant 5 Growth Rate Chart 358

Fig. 6.44 Plant 6 Growth Rate Chart 358
Fig. 6.45 Plant 7 Growth Rate Chart 358
Fig. 6.46 Plant 8 Growth Rate Chart 359
Fig. 6.47 Plant 9 Growth Rate Chart 359
Fig. 6.48 Plant 10 Growth Rate Chart 359
Fig. 6.49 Plants 1-10 Growth Chart (Trial 2) 189
Fig. 6.50 Plant 1 Growth Rate Chart 360
Fig. 6.51 Plant 2 Growth Rate Chart 360
Fig. 6.52 Plant 3 Growth Rate Chart 361
Fig. 6.53 Plant 4 Growth Rate Chart 361
Fig. 6.54 Plant 5 Growth Rate Chart 361
Fig. 6.55 Plant 6 Growth Rate Chart 362
Fig. 6.56 Plant 7 Growth Rate Chart 362
Fig. 6.57 Plant 8 Growth Rate Chart 362
Fig. 6.58 Plant 9 Growth Rate Chart 363
Fig. 6.59 Plant 10 Growth Rate Chart 363
Fig. 6.60 Correlation Graph Between Actual Response and Predicted Response of Soil Moisture Data 191
Fig. 6.61 Correlation Graph Between Actual Response and Predicted Response of Temperature Data 192
Fig. 6.62 Correlation Graph Between Actual Response and Predicted Response of Humidity Data 193

List of Tables

Table 1.1 Prior Studies 4
Table 2.1 Literature Summary 16
Table 3.1 Soil Moisture Sensor Intervals 37
Table 3.2 Relative Humidity Interval 37
Table 4.1 Components Range Interval 50
Table 4.2 Fan Specification 60
Table 4.3 DHT22 Temperature Threshold Values 128
Table 4.4 DHT22 Humidity Threshold Values 128
Table 4.5 Soil Moisture Sensor Threshold Values 128
Table 5.1 Mode Tabulation with Dimensions 133
Table 5.2 Adaptive Irrigation System Thresholds 137
Table 5.3 PlantCV and MATLAB Results 150
Table 5.4 PlantCV and MATLAB Percent Difference Table 151
Table 5.5 Sensors Comparison 161
Table 5.6 Cost Breakdown 162
Table 5.7 Plant Weight 165
Table 6.1 Radial Distances for Planting Ground No. 1 342
Table 6.2 Radial Distances for Planting Ground No. 2 342
Table 6.3 Radial Distances for Planting Ground No. 3 342
Table 6.4 Radial Distances for Planting Ground No. 4 343
Table 6.5 Radial Distances for Planting Ground No. 5 343
Table 6.6 Radial Distances for Planting Ground No. 6 343

Table 6.7 Radial Distances for Planting Ground No. 7 343
Table 6.8 Radial Distances for Planting Ground No. 8 344
Table 6.9 Radial Distances for Planting Ground No. 9 344
Table 6.10 Radial Distances for Planting Ground No. 10 344
Table 6.11 Tabulation of Standard Deviation per Trial 176
Table 6.12 Tabulated Area Occupied per Plant (Trial 1) 356
Table 6.13 Average % Increase for Plants 188
Table 6.14 Tabulated Area Occupied per Plant (Trial 2) 360
Table 6.15 Soil Moisture Sensor Comparison 364
Table 6.16 Temperature Sensor Comparison 364
Table 6.17 Humidity Sensor Comparison 365
Table 6.18 Soil Moisture Sensor Comparison for Trial 2 369
Table 6.19 Temperature Sensor Comparison for Trial 2 374
Table 6.20 Humidity Sensor Comparison for Trial 2 379

De La Salle University

Chapter 1 INTRODUCTION

CONTENTS

1.2 Prior Studies 2
1.3 Problem Statement 4
1.4 Objectives 5
1.4.1 General Objective(s) 5
1.4.2 Specific Objectives 5
1.5 Significance of the Study 5
1.6 Assumptions, Scope and Delimitations 6
1.6.1 Assumptions 6
1.6.2 Scope 6
1.6.3 Delimitations 6
1.7 Description and Methodology of the Thesis 7
Estimated Work Schedule and Budget ..... **Error! Bookmark not defined.**
Overview 10

1.1 Background of the Study

People continue to be drawn to cities by the economic, social, and creative opportunities it has to offer. Large cities have become more productive than rural areas, producing more patents and yielding higher returns on capital. In Metro Manila, it suffers effects such as poverty, congestion, and pollution daily (Schumacher, 2017). He has also stated that the new wave of urban farming can be seen as a new vertical green belt feeding the populous while also bringing nature to the city center.

One cannot discuss urban gardening without mentioning the plight of farmers or how pockets of greenery affect the environment. The long view based from those who fashion small farms in city homes is that urban gardening can hopefully raise awareness on salient issues in agriculture and ecology (Bueno, 2017).

The urban population in Metro Manila has significantly increased (Weller, 2016). This makes the city one of the busiest cities in the world. Since people living in the city mostly dwell on work, they do not have the luxury of time for other things. Not everyone has access to an easy process to start their garden at home. Through urban farming, it can encourage these people to plant their edible food at home.

1.2 Prior Studies

Method	Approach	Reference
Irrigation System	1. pH, temperature, moisture sensor, water pump, and servo motor that will communicate with the Arduino and software program.	1. Padalalu, P., Mahajan, S., Dabir, K., Mitkar, S., & Javale, D. (2017). Smart water dripping system for agriculture/farming. 2017 2nd International Conference for

	2. Comprehensive water delivering schedule using Arduino. Water pumps will be used as actuators along with soil moisture and liquid flow meters which will be used as sensors.	Convergence in Technology (I2CT) 2. Su, H., Chen, S., & Fatras, N. (n.d.). Smart Water Irrigation System : Civil and Environmental Engineering 186:Design of Cyber-physical System
Seed Planting	1. This is in a form of a disk provided with elongate perforations of a narrow width, which cooperate with a seed container and an air suction device. 2. Uses ATMega microcontroller to control the DC and servo motor which supervises the whole robot. 3. Seed metering mechanism is based upon a solenoid actuator assembly for an evenly spaced rows with each point where a seed is dropped equidistant from the successive point	1. Fiorido, A. F. (1994). U.S. Patent No. 5,325,801. Washington, DC: U.S. Patent and Trademark Office. 2. Amrita Sneha.A, Abirami.E, Ankita.A, Praveena, R., & Srimeena, R. (2015). Agricultural Robot for Automatic Ploughing and Seeding. 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR). 3. Srinivasan, N., Prabhu, P., Smruthi, S. S., Sivaraman, N. V., Gladwin, S. J., Rajavel, R., & Natarajan, A. R. (2016). Design of an autonomous seed planting robot. 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC).

Crop Monitoring	1. Mobile application interconnected using Bluetooth for handling communication between the mobile and the garden tools. 2. Internet of Things application that collectively combines the information from sensors that measure soil moisture, pH level, temperature, humidity, light dependent resistance, and water level.	1. Gowri, S., & Divya, G. (2015). Automation of garden tools monitored using mobile application. International Conference on Innovation Information in Computing Technologies 2. Pandithurai, O., Aishwarya, S., Aparna, B., & Kavitha, K. (2017). Agro-tech: A digital model for monitoring soil and crops using internet of things (IOT)

Table 1.1 Prior Studies

1.3 Problem Statement

With the growing trend of urban gardening and local produce, people from cities around the world grow produce from neighboring farms or even from their gardens. Unfortunately, there are a lot of challenges growers face including water availability, changes in climate, atmospheric conditions, and the manual labor that needs to be put in especially since a lot of growers who live in cities have office work. People living in the city have limited free time due to their jobs so maintaining an urban farm could be a challenge. Some beginner growers do not have sufficient skills and knowledge to determine the crops' state and condition. The seed planting process, since it is done by hand, might cause a problem in crop spacing. Overcrowding of crops may reduce yields,

lower the quality since they will be competing for light and soil nutrients, and give growers a hard time monitoring.

1.4 Objectives

1.4.1 General Objective(s)

To develop an automated system that performs basic gardening operations such as seeding, watering, and monitoring.

1.4.2 Specific Objectives

1. To develop a microcontroller-based system capable of monitoring and providing uniform and variable seed distribution within the field.
2. To develop a mobile application that will monitor the soil moisture, temperature, humidity, and plant growth.
3. To develop an adaptive automatic watering system for the crop based on soil moisture.
4. To create a monitoring system that consists of temperature, humidity, and soil moisture sensor that is at least 90 percent accurate.
5. To be able to plant the seeds with an accuracy of ±3 cm from the target.
6. To conduct cost analysis on the proposed system.

1.5 Significance of the Study

The findings of this study will redound to the benefit of society especially the urban community considering that urban gardening is becoming more and more popular today. The increasing prices and awareness of the quality of food people consume encourage do-it-yourself (DIY) gardening but people living in cities have limited time

due to a lot of factors such as work and deadlines. Thus, the automation of basic gardening processes will be able to be of great help to the urban community.

1.6 Assumptions, Scope and Delimitations

1.6.1 Assumptions

1. The soil on the planting ground is already conditioned, leveled, and ready for seed sowing.
2. Wi-Fi connection is available in the area.
3. The planting ground will be placed indoors, inside a room with a controlled environment.

1.6.2 Scope

1. This covers equal seed distribution making sure that seeds are evenly spaced throughout the land area.
2. The monitoring system will only cover 1m x 0.5m land area.
3. This machine will automate the process of seed planting and crop watering.
4. The soil moisture content will be monitored in a timely manner by the machine and this data will be available to the user through the mobile application.
5. The gripper will be used for seeding and watering only.
6. The project will make use of artificial lighting in replacement of natural light since it will be implemented indoors.

1.6.3 Delimitations

1. This does not cover other crops that have more than a months' time for it to fully grow.

2. The robot is designed for indoor use, therefore not applicable for use in outdoor spaces.
3. The camera will be placed on top of the center of the planting ground, not placed on the moving parts of the XYZ robot, therefore fixed and not free to move.

1.7 Description and Methodology of the Thesis

Fig 1.1 Main Flow of the Gardening System

The project performed three major gardening operations: seed planting, crop watering, and crop monitoring. The soil in the planting ground was manually conditioned by the user. Harvesting the crop was also done manually by the user.

The project consisted of the following: the cartesian robot that was responsible for the seeding and watering operation, the sensors that gathered data from the crops in the fixed-size planting ground, the camera that captured images which was used for image processing to determine whether the crop was ready to harvest or not, the lighting system that was used as an alternative to sunlight since the planting ground was made indoor, and the mobile application that was used to make the necessary adjustments in the settings of the system.

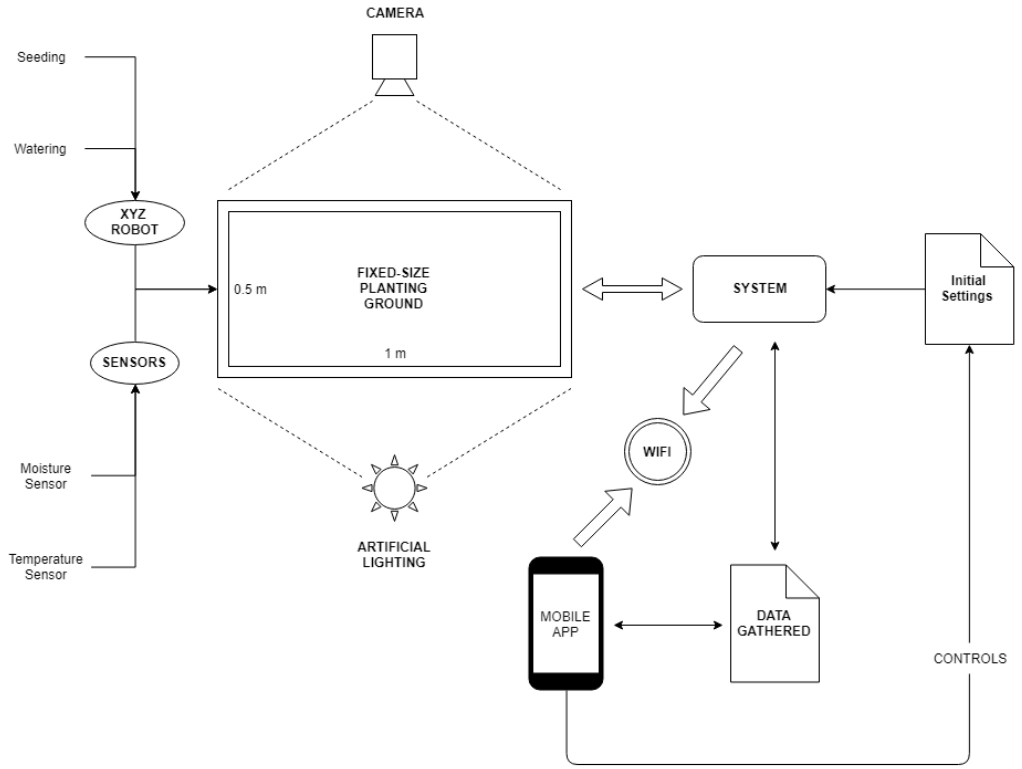


Fig 1.2 Concept Map

The system used the dimensions of the planting ground as a basis to calculate the necessary spaces for optimal placement of the seeds. It used the cartesian coordinate system to easily locate at which area it is currently at.

The cartesian robot was equipped with a seed planter. The seeding system used suction that picked up the seeds from a container and then travel to the specific coordinate. The seed planter penetrated the soil at a specific depth and released air to plant the seed at its location.

The soil's moisture content was measured in a timely manner using a soil moisture sensor module. The data obtained from this process was used for monitoring whether the crop needs to be watered and for the mobile application's data. The soil

moisture sensor was used to test and check the moisture content of the crops in a scheduled manner, and once the sensor has indicated the crops need for water, the system automatically dispenses water.

The group made use of a cartesian coordinate system that applies both seeding and watering. To validate whether the seeds were uniformly distributed, target locations were placed beforehand for the researchers to measure how far it is from the actual target.

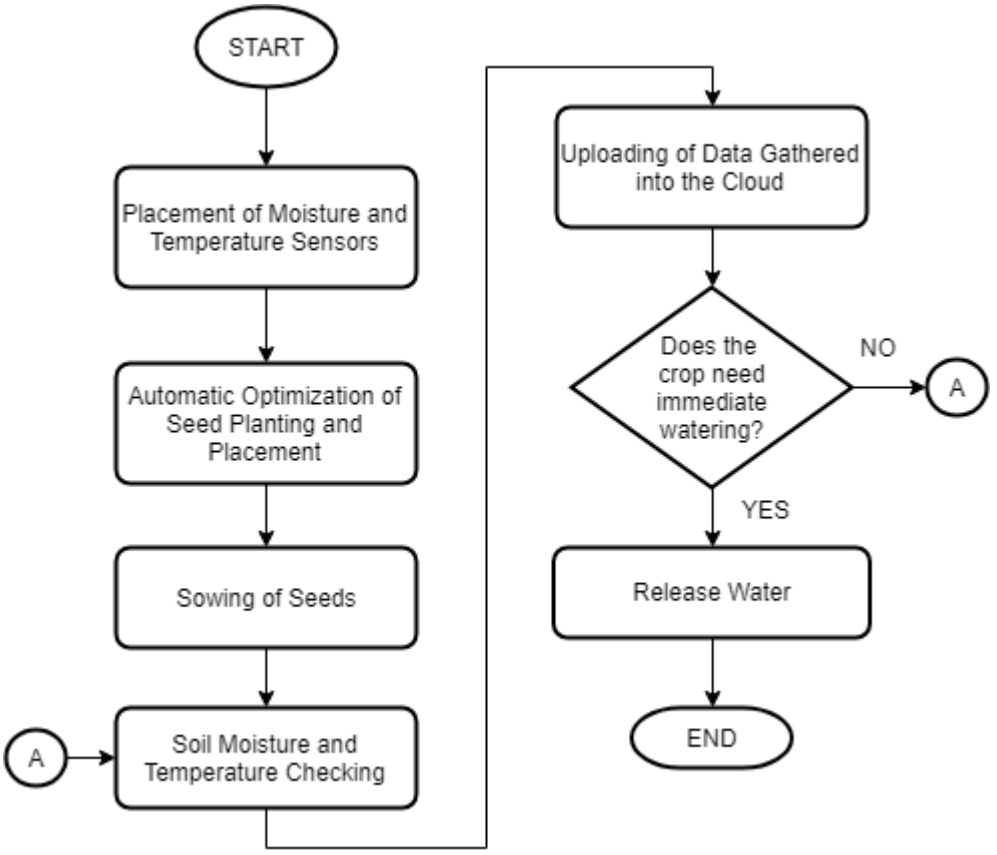


Fig 1.3 Flowchart of the System's General Algorithm

For the user to monitor the moisture level, temperature, humidity, and plant growth of the crop, the data obtained were displayed through the interface of the mobile

application which made it accessible for the user. Lastly, billing of the materials will be listed and tabulated to see the breakdown expense of the system.

Overview

This chapter focuses on the general view of how the proposed system works. Along with that, it discusses the main objectives to be achieved once implemented and tested. It tackles problems in relation to human labor that gives a solution in forms of automated robots. Through this robot, inventions can benefit humans to lessen the time consumed and accumulate more work. Works such as seed planting, watering, and monitoring are applied throughout the system.

Chapter 2 LITERATURE REVIEW

Contents

2.1 Existing Work 13
2.1.1 Agricultural Robot 16
2.1.1.1 Multipurpose Autonomous Agricultural Robot 16
2.1.1.2 Indoor robot gardening: Design and implementation 17
2.1.1.3 Agricultural Robot for Automatic Ploughing and Seeding 18
2.1.1.4 Design of a Controlled Environment Agricultural Plant Inspection Robot 19
2.1.2 Smart Agricultural System 21
2.1.2.1 Design of Intelligent Plant Growth Cabinet Environment Monitoring and Control System 21
2.1.2.2. AgriSys: A Smart and Ubiquitous Controlled Environment Agriculture System 22
2.1.2.3 Plant Growth Using Automatic Control System under LED, Grow, and Natural Light 23
2.1.2.4 Smart Water Dripping System for Agriculture/Farming 25
2.1.3 Brassica Rapa 27
2.1.3.1 Growth and Yield of Pechay (Brassica Pekinesis) as Affected by Green Herds Organic Based Foliar Fertilizer 27
2.1.3.2 Pechay Production Guide 29
2.1.3.3 Pechay Open Field Production 30
2.1.3.4 Growth Response of Brassica Rapa on the Different Wavelength of Light 30
2.1.3.5 Retail Price 30
2.1.3.6 Characteristics 31
2.1.4 Grow Room 32
2.2 Lacking in the Approaches 33
2.3 Summary 33

2.1 Existing Work

Source	Methodology	Findings
Gollakota, A., Srinivas, M. B. (2011). Agribot A multipurpose agricultural robot. 2011 Annual IEEE India Conference. Amritanshu Srivastava, Shubham Vijay, Alka Negi, Akash Singh, DTMF Based Intelligent Farming Robotic Vehicle, International Conference on Embedded Systems (ICES), IEEE 2014.	-Multipurpose agricultural robot that performed soil moisture testing, seeding, spraying pesticides, remove compost from field, obstacle avoidance, and metal detection -Controlled using Arduino Mega and Raspberry Pi -Communication between signals were done via Wi-Fi	-Application of a wireless network for efficiency and an overall effectivity system
Correll, N., Arechiga, N., Bolger, A., Bollini, M., Charrow, B., Clayton, A., Rus, D. (2010). Indoor robot gardening: design and implementation. Intelligent Service Robotics, 3(4), 219232.	-A mobile robot equipped with camera capable of locating and watering plants -Used networking to communicate servicing requests to the robot	-Mobile robot being capable of detecting plants and flowers and perform different operations
Chen, H. (2012) Design of a Controlled Environment Agricultural Plant Inspection Robot. MS) thesis, University of Iowa, 2012.	-Development of a fixed-position robot capable of providing a remote observer that assesses signs of plant stress -Incorporation of vision, detection, and identification systems	-Fixed-position systems have stationary base which provides precision movement and reliability -Usage of stepper or servo motors are ideal for linear movements
Lin, H., Rui-Qi, M., Shi-Gang, C., Yong-Li, Z., & Xmg-Li, W. (2018). Design of Intelligent Plant Growth Cabinet Environment Monitoring and Control System. 2018 Chinese	-Design of an effective growth cabinet that provides an optimal environment for plant production	-Significance of providing ideal light intensity for plants -Usage of fan for air circulation/exchange

Control And Decision Conference (CCDC).	-System that collects the environment parameter value in real-time -Used STM32 as the main controller	
Abdullah, A., Enazi, S. A., & Damaj, I. (2016). AgriSys: A smart and ubiquitous controlled-environment agriculture system. 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC). doi:10.1109/icbdsc.2016.7460386	-A smart agricultural system that analyzes an environment and intervene to maintain its adequacy such as dealing with temperature, humidity, pH, ,and nutrient support -Developed using LabVIEW that relied on using fuzzy logic	-Usage of many sensors to gather data and perform automated control and monitoring options
Limprasitwong, P.,Thongchaisuratkrul, C. Plant Growth Using Automatic Control System under LED, Grow, and Natural Light. 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), Krabi, 2018, pp. 192-195.	-An automatic control system that studied the effect usage of light for shallots in a smart farm -Used four air inlets for the cooling system -System was divided into two rooms, one for LED light and the other for grow lights -Two DHT22 sensors and two pumps were used for detection and for irrigation	-Humidity indoor was higher due to the low rate of evaporation -Indoor planting can reduce watering -Germination rate show that shallots grow faster under LED and grow lights as compared to natural light
Padalalu, P., Mahajan, S., Dabir, K., Mitkar, S., & Javale, D. (2017). Smart Water Dripping System for Agriculture/Farming. 2017 2nd International Conference for Convergence in Technology (I2CT).	-Utilization of various monitoring and controlled system to increase the yield with the help of automation -Used Arduino that remotely controlled the motor	-Usage of MySQL database server for data storage -Variable water distribution depending on the type of crop

Fernandez, A., Miñoza, E. (2015). Growth and Yield of Pechay (Brassica Pekinesis) as Affected by Green Herds Organic Based Foliar Fertilizer. Horticulturae	-Used fertilizer as a crucial factor for growth -Put two to three seedlings per hill -Effects of using fertilizer were evaluated through its leaf's height and width	-Fertilizer had significant effects on the growth hence providing the best growth and yield
Jimenez, E. F., Mariano, J. S., Ferrer, M. J., De Leon, V. (2000). Pechay Production Guide. Retrieved from http://bpi.da.gov.ph/bpi/images/Production_guide/pdf/PECHAY%20.pdf Woody, S., & Amasino, R. (2011). Instructions and Tips for Growing Rapid-cycling Brassica rapa variety "FPsc." University of Wisconsin-Madison. https://fpsc.wisc.edu/growguide/Instructions_and_Tips_for_Growing_FPsc.pdf Applied Horticultural Research Pty Ltd (2018) Open Field Production. Crop group: Brassica (cabbage family) – cabbage, broccoli, cauliflower. http://ahr.com.au/wp-content/uploads/2018/02/Philippines-Factsheet_Pechay_Field.pdf	-Explained different production guides of Brassica Rapa	-Soil moisture must be adequate to obtain maximum growth -Providing an ideal spacing and environment is crucial to growth -Grows best in clay loamy soil with pH ranging from 5.5 to 6.5 -Pechays grow all year round -This crop do not tolerate drought soil -Seedlings are planted directly or transplanted -Seeds should be covered to a depth of around 1cm -Spacing should be 10cm by 20cm between rows
Acero, L. (2013). Growth Response of Brassica Rapa	-Pechay was exposed to white, blue, red, green,	-Plants exposed to white color gave the highest

on the Different Wavelength of Light. International Journal of Chemical Engineering and Applications. 415-418. 10.7763/IJCEA.2013.V4.337.	and yellow colored lights to test its effect on the different wavelength of light	length, width, and average number of leaves after fourteen days. -White colored bulb is recommended as supplementary lighting to pechay
Espiritu, K., & Nielsen, L. (2020, January 24). Grow Room Ventilation 101. Epic Gardening. https://www.epicgardening.com/grow-room-ventilation-101/	-Discussed the concept of passive and active intake -Discussed effects when humidity is not properly controlled	-Emphasized the importance of air circulation in grow rooms -Plants in the vegetative growth thrive in 40%-70% humidity

Table 2.1 Literature Summary

2.1.1 Agricultural Robot

2.1.1.1 Multipurpose Autonomous Agricultural Robot

The robot which performs operation like soil, moisture testing, seeding, spraying pesticides, removes compost from the field, which also performs obstacles avoidance operation and metal detection in the path (Gollakota Srinivas, 2011). This robot is controlled using a mobile phone incorporating dual-tone multi-frequency (DTMF) technique that is used to overcome distance problem of using Bluetooth or radio frequency (RF) module. Srivastava, Vijay, Negi, Singh (2011) says that the Agribot integrated system which uses Wi-Fi to communicate between two robots which perform activities like seeding, weeding, spraying of fertilizers and insecticides. It is controlled using Arduino Atmega2560 controller and powerful Raspberry pi minicomputer to control and monitor working of robot. It has hexapod body which can move in any direction and uses an ultrasonic proximity sensor to avoid the obstacles in the path, and underbody sensor system to detect that seed is planted or not. In addition, this can dig a

hole in soil plant seed in it and cover the hole again with soil and necessary pre emergence fertilizers applies on it and move on along with communicating with other robot near to it using Wi-Fi. Dubey (2011) suggested a Wireless sensor network and Embedded based technique of DTMF (Dual Tone Multiple Frequency) signaling to control water flow for sectored, sprinkler or drip section irrigation. Two principles are utilized in their system, one is wireless sensor network, and another is of 'circuit switching' instead of 'packet switching' used by SMS controlled devices available currently in the market. Thus, such circumstance has certainly played the vital role for the purpose of the building up of the proposed system having into consideration the factors like conservation and effective working of system.

2.1.1.2 Indoor robot gardening: Design and implementation

The research tackles the implementation of an autonomous gardening robot with applications in urban or indoor precision agriculture Correll, N. et al. (2009). The garden is a mesh network of robots and plants. The gardening robots are mobile and equipped with an eye-in-hand camera capable of locating plants in the garden, watering them, and locating and grasping fruit. The plants are potted cherry tomatoes. These have sensors which will output data that will be used in monitoring the state of fruits and computing for soil humidity. This also has networking to communicate servicing requests to the robots. By embedding sensing, computation and communication into the pots, task allocation in the system is centrally coordinated, which makes the system scalable and robust against the failure of a centralized agent.

Fig. 2.1 Indoor Robot

2.1.1.3 Agricultural Robot for Automatic Ploughing and Seeding

This system develops a robot capable of performing operations like automatic ploughing, seed dispensing, fruit picking and pesticide spraying Amrita, Abirami et al. (2015). It also provides manual control when required and keeps tabs on the humidity with the help of humidity sensors. The main component here is the AVR ATMEGA microcontroller that supervises the entire process. Initially the robot tills the entire field and proceeds to ploughing, simultaneously dispensing seeds side by side. The device used for navigation is an ultrasonic sensor which continuously sends data to the microcontroller. On the field the robot operates on automated mode, but outside the field is strictly operated in manual mode. For manual control the robot uses the Bluetooth pairing app as control device and helps in the navigation of the robot outside the field. The field is fitted with humidity sensors placed at various spots that continuously monitor the environment for humidity levels. It checks these levels with the set point for humidity and alerts the farmer. This mechanism is a GSM module that sends a text message to the

farmer informing him about the breach in set point. The farmer then responds via SMS to either switch on the water sprinklers or ignore the alert. The water sprinklers, if on, bring down the humidity level thus providing an ideal growing environment to crop. Farmers today spend a lot of money on machines that help them decrease labor and increase yield of crops, but the profit and efficiency are very less. Hence automation is the ideal solution to overcome all the shortcomings by creating machines that perform one operation and automating it to increase yield on a large scale.

2.1.1.4 Design of a Controlled Environment Agricultural Plant Inspection Robot

This research conducted by Chen (2018) aims to develop a plant inspection robot for use in controlled environment agriculture. The inspection robot should be capable of providing a remote observer with multiple vantage points of each plant to visually assess for signs of plant stress. With that, vision and detection systems were applied in this research. This was to detect certain parameters and identify its size, shape, texture for classification. The selection of the image processing techniques and the classifier algorithm are both important in detection systems. The image processing techniques process the raw image to find features, such as shape outlines and color. The classifier algorithm uses these properties to determine the plant species or whether a particular region of the image represents a fruit or a leaf.

Plant detection and identification is an active and promising research area. The inspection system described could provide image input into such a system. This could provide a valuable aid to the human operator by automatically directing attention to unexpected growth in the greenhouse. Ultimately, such a system could perform weed surveillance autonomously.

The robotic platform used is a Fixed-Position Systems. This has a stationary base or frame, which provides the robot's end effector with a fixed reference frame. This precise and reliable reference is useful in allowing the robot to consistently reach positions within its work envelope. Fixed position robotic systems are widely used in the manufacturing environment because of their reliability.

Cartesian robotic systems have perpendicular linear actuators that naturally operate in a rectangular frame. Cartesian robots can have a work envelope encompassing acres. Cartesian robotic systems generally do not provide a large range of motion, although it can be coupled with a non-Cartesian robot. This type of robotic platform would be the most ideal for the plant inspection robot since plants are commonly placed in a rectangular area and the camera system does not require a large range of motion to sufficiently inspect the plants.

The gantry arrangement requires a guide system to support both static and dynamic loads, a motor to actuate the system, and a transmission system to convert the rotational motion of the motor into a linear motion. There are three types of guide systems commonly used in linear motion systems: round rail, profile rail, and v-style rollers. V-style rollers support the tangential load. The wheels ride on a hardened steel track with a complementary "V" profile. Although they are not as accurate as the round and profile guides, V-style rollers are easier to implement. Their simple design requires little or no maintenance while providing a long-life expectancy. They cost less than round or profile guides, are easier to install and can span long distances (Overby, 2010).

Stepper or servo motors can be used to provide the power for a linear motion system. Stepper motors are used in precision applications where a high rotational speed is not required. A stepper motor has a large number of magnetic poles in the stator

winding that enables the motor to achieve very small increments of rotational movement (Overby, 2010). Stepper motors are generally driven with an open-loop control structure, meaning the controller assumes that the motor responds as expected to all control signals. This reduces system complexity and the cost of the electronics. To ensure that small force deviations and less-than-ideal operating conditions do not cause the motor to misstep, stepper motors are generally selected to be significantly more powerful than necessary to prevent frequent mismatches between the expected and actual angular position of the stepper motor.

2.1.2 Smart Agricultural System

2.1.2.1 Design of Intelligent Plant Growth Cabinet Environment Monitoring and Control System

This research done by Lin et al. (2018) implanted a design scheme of intelligent plant growth cabinet monitoring and control system using a STM32 as the main controller. The plant growth cabinet can provide an optimal growth environment for plants which can be used as an alternative for the conventional plant production. The system environment parameters can be observed in real time and control the actuators in the system via touch screen.

The sensor initially collects the environment parameter value in real-time and the controller accepts the environment data measured by the sensor. After analyzing and comparing with the set value, the controller outputs the signal through the intelligent control algorithm. LED light source and hydroponic culture are used. Air temperature affects the photosynthesis and respiration of plants. Temperature can significantly promote photosynthesis, shorten the growth cycle of plants, and improve the final yield

of plants. Air humidity on the other hand affects transpiration therefore it must also be considered.

The most commonly used light intensity index is the Photosynthetic Photon Flux Density (PPFD) which refers to the quantum number of photons in the 400nm~700nm range. The system operation results show that the LED lights in the intelligent plant growth cabinet (1500mm x 1500mm x 2000mm) had a daily replenishment time of 8 am to 10 pm. Air temperature working range was 15C to 28C and humidity range 40%~80%. The fan works for 5 minutes per hour. Its purpose is to make the air exchange. The oxygen in goes outside while letting the carbon dioxide goes inside. This is to ensure the adequate supply of raw materials required by photosynthesis of the plants.

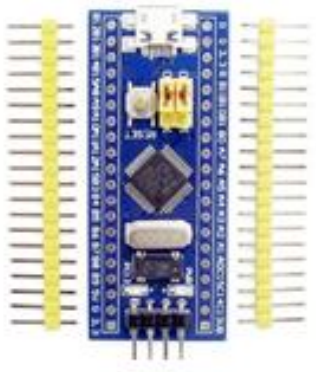


Fig. 2.2 STM32F103

2.1.2.2. AgriSys: A Smart and Ubiquitous Controlled Environment Agriculture System

This study by Abdullah et al. (2016) aims to present a smart Agricultural System (AgriSys) that analyzes an agriculture environment and intervene to maintain its adequacy such as dealing with its temperature, humidity, pH, and nutrient support. It

aims to increase productivity, enhanced safety, provide easier agriculture procedures, and immediate interventions.

This system is developed and implemented under LabVIEW and relies heavily on the concept of fuzzy logic to be able to create an operator that makes decisions similar to humans. Application of acquisition, monitoring, analysis, and recording data were used as well. The fuzzy controller uses simple linguistics which makes it easier to understand. AgriSys is made available for distant and remote access through a web server that a user can monitor in real-time the stats of its controlled agriculture environment.

Their system relies on the usage of many sensors to gather data and perform automated control and monitoring options. It records the response of sensor readings to turn the pump or the fan on or off. Monitoring and controlling the humidity and temperature of the environment is a must to save the plants from drought and extremes temperature. Moreso, this system also includes a pH and soil moisture sensor. The pump in AgriSys is used for irrigation and cooling system

2.1.2.3 Plant Growth Using Automatic Control System under LED, Grow, and Natural Light

This research conducted by Limprasitwong & Thongchaisuratkrul (2018) aims to study an effective way of light used for plant growth while implementing a smart farm. It presents an automatic control system for planting shallots. The light types applied here include LED, grow, and natural light.

The plant room used was 1.2x1.2x1.5m in dimension with four air inlets for the cooling system and water drain in towards the center. Room was divided into two – one for LED light and the other for grow lights. Two DHT22 sensors were used to detect

both temperature and humidity. Two pumps were used for water and cooling pad. The plant was watered automatically. The system was controlled by an Arduino Uno.

The temperature control consists of first detecting the DHT22 value, then evaluates whether the fan will be triggered by its threshold value. Water control on the other hand handles the pumping of water from the storage. Lastly, the light control focuses on turning on and off the LED and grow light at the same time by the help of relay.

Experimental results show that the humidity indoor was higher due to the low rate of evaporation. This indicates that indoor planting can reduce watering. Results for the germination rate show that shallots grow faster under LED and grow lights as compared to the natural light. Shallots under LED light had the fastest rate of germination. The followed by grow light and the natural light respectively. The plant under grow light is the most rapid growth.

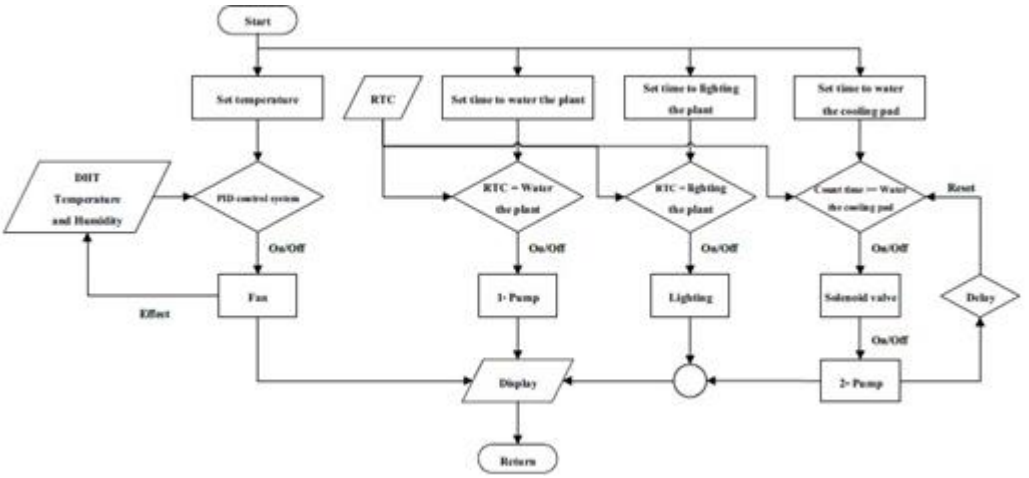


Fig. 2.3 System Flowchart

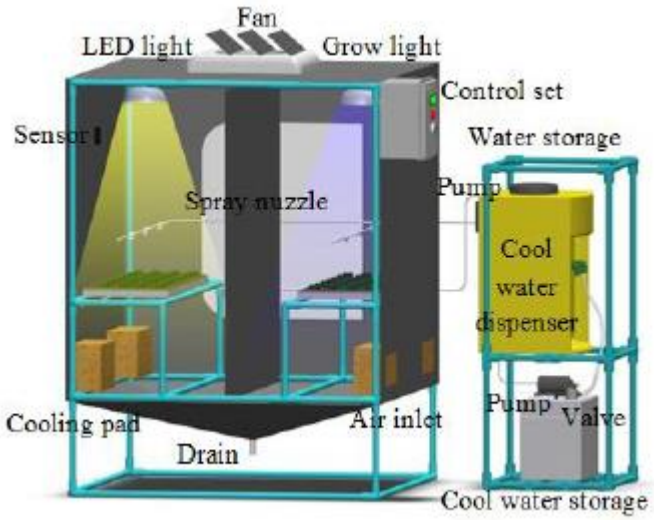


Fig 2.4 Room Setup

2.1.2.4 Smart Water Dripping System for Agriculture/Farming

The research performed by Padalalu et al. (2017) aims to utilize various monitoring and controlled system in order to increase the yield with help of automation of an agricultural parameter like temperature, humidity, soil moisture, carbon dioxide, light detection, and soil pH.

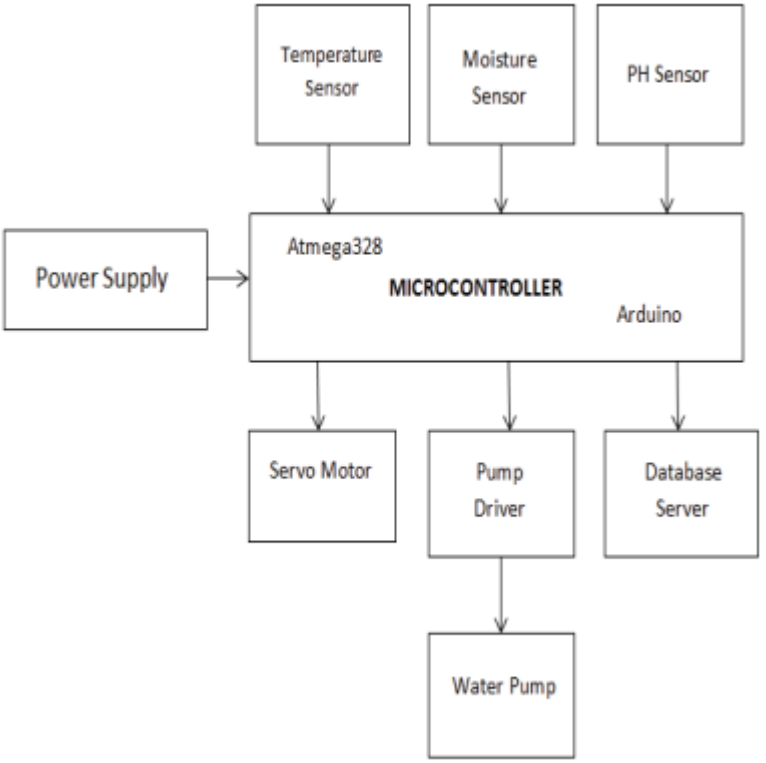


Fig. 2.5 Block Diagram

Based on these parameters, the actuators will decide movement of servo motor and provide accurate quantity of water to be supplied depending on the crops. In Arduino, the program is written in such a format that it denotes a specific range of resistance value in digital format (ranging from 0 to 1023) for the moisture, pH, and temperature.

Android application was developed that remotely controls the motor. Depending upon the input sensor values, a toast will be popped will tell whether or not the motor should be switched on. MySQL database server was also implemented. Predefined moisture, temperature and pH ranges are set. The software program will be executed accordingly and sends high signal to the relay to switch on the motor, thus performing

the irrigation process. Depending upon the type of crop, the amount of water is already set. A timer will be used that will be used to water the crops. Once, the timer reaches to 0, the water supply will be stopped. This smart irrigation system can benefit the environment by conserving water with minimal human intervention.

2.1.3 Brassica Rapa

2.1.3.1 Growth and Yield of Pechay (Brassica Pekinesis) as Affected by Green Herds Organic Based Foliar Fertilizer

In this study conducted by Fernandez and Miñoza (2015), both enhanced the growth and yield parameters of pechay. They have stated that the pechay production has a high potential, market-driven commodity that could provide income even during the abundant supply. The experiment used a fertilizer as a crucial factor for the growth. With the fertilizer, process was enhanced due to its natural benefits and less environment risks. This study mostly revolved around the use of Green Herds Organic Based Foliar Fertilizer, a newly developed mixed organic and inorganic fertilizer product that contains essential nutrients. The nutrients included elements that enhances the growth and protection from diseases.

The cultural management include a plowed field with two to three seedlings on each hill. After a week, one seedling was maintained for transplanting. The effects of the fertilizer were evaluated through its height, leaf width, and leaf length. Results have stated that application of the fertilizer had significant effect on the growth hence providing the best result on the growth and yield of the crop.

2.1.3.2 Response of Pechay (Brassica napus L.) to Different Levels of Compost Fertilizer

The study conducted by Gonzales et al. (2015) have tested the growth of pechay from different organic fertilizer such as goat manure, rice straw, and carbonized rice hull ash. Their process included covering the pile with plastic sheet completely to maintain the heat of decomposition and minimizes water evaporation. They even mixed the organic materials twice a week to avoid overheating. Irrigation was also done regularly to add moisture content in the composting materials.

Treatments	Mean Plant Height (cm)	Number of leaves/plant	Leaf Area
T1 = Pure Garden Soil (control)	25.57	5.17	21.81b
T2 = Pure Compost (OF)	30.77	5.67	40.69a
T3 = 50% Pure Garden Soil: 50% Pure Compost (OF)	29.1	5.17	42.6a
T4 = 75% Pure Garden Soil: 25% Pure Compost (OF)	32.5	5.83	42.17a
T5 = 85% Pure Garden Soil: 15% Pure Compost (OF)	31.7	5.33	39.97a

Treatments	Mean Fresh weight/plant (g)
T1 = Pure Garden Soil (control)	12.93b
T2 = Pure Compost (OF)	26.6a
T3 = 50% Pure Garden Soil:50% Pure Compost (OF)	23.5a
T4 = 75% Pure Garden Soil:25% Pure Compost (OF)	29.9a
T5 = 85% Pure Garden Soil: 15% Pure Compost (OF)	23.17a

Table source: Gonzales, L., Caralde, R. & Aban, M. (2015). Response of Pechay (Brassica napus L.) to Different Levels of Compost Fertilizer. International Journal of Scientific and Research Publications. 5. 1-2250.

Results have shown that the application of 75% pure garden soil and 25% pure compost provided the best growth and yield in terms of leaf area and fresh weight.

2.1.3.3 Pechay Production Guide

This study by Jimenez et al. (2000) focused on the production of pechay including its varieties, requirements, management, and harvesting. This research mentioned that pechay can be grown all year round but produces their best growth and quality during cool conditions. During the dry season, pechay can also be grown with good irrigation in a sandy to clay loam type of soil with a pH ranging from 5.5 to 6.5.

With regards to their cultural management, plowing the soil thoroughly enhances the growth of the crop. The use of organic and inorganic fertilizers is recommended. Urea can be used 2 tbsp/gal for the water solution. The pechay seeds can be planted directly or indirectly in the soil. Seeds are covered to a depth of about 1 cm by raking or spreading additional topsoil and applying water immediately afterwards. Spacing for each crop should be 10 cm between plants and 20 cm between rows. Researchers have also stated that to obtain the maximum growth, soil moisture must be adequate. Plants are watered every other day during dry season or when needed. A ready to harvest pechay can be done 30 to 40 days after sowing.

Issues such as pests can also occur due to different reasons. This may be due to soaked soil and excess water that can attack the seedlings. Cases such as wilting can also happen when not given proper attention.

Another study conducted by Woody and Amasino (2011) stated Brassica Rapa grows well in a temperature range of 15 – 25°C. At temperatures towards or below 15 shows that growth is notably slower, while signs of stress and wilting are evident at or

above 25. They have also mentioned that the soil must be kept moist but not saturated. This is because the roots must breathe to continuously supplement the nutrients needed.

2.1.3.4 Pechay Open Field Production

This factsheet conducted by Applied Horticultural Research Pty Ltd (AHR) (2018) mentioned that the optimum temperatures for growth and development are from 18 to 20°C. This is because pechay thrives in a well-drained, moisture-retentive, loamy soils well supplied with organic matter. It has also stated that pechay does not grow well in highly acidic soil. An ideal pH value should be around 5.5 to 6.5. Pechay does not tolerate drought conditions so adequate soil moisture must be maintained. The land preparation include a 1m wide raised beds in low to mid elevations and drainage canals on the side. Harvesting can be done 30 to 40 days after sowing.

2.1.3.5 Growth Response of Brassica Rapa on the Different Wavelength of Light

This experiment done by Acero (2013) determines what wavelengths of light affects the vegetative growth of Brassica rapa grown in pots in urban area. The experiment was implemented using 5 different treatments. Plants in all treatments were exposed to sunlight during daytime and to different colored fluorescent lights for eight hours during nighttime for two weeks. Pechay was exposed to white, blue, red, green, and yellow colored fluorescent lights. Results were evaluated by measuring the average length and width of the leaves, the number of leaves formed. Result of the study revealed that plants exposed to white color gave the highest length, width, and average number of leaves after fourteen days, followed by plants in green light, blue light, yellow light, and the least which is the red light. The study can conclude that white colored light bulb is recommended as supplementary lighting to Brassica rapa grown in pots in urban areas.

2.1.3.6 Retail Price

According to the Philippine Statistics Authority, data has shown that the retail price for native pechay in Baguio is 80 PHP/kg in Dec 2018. Average price seems to stay constant from 2013 to 2018 even with its all-time high and record low.

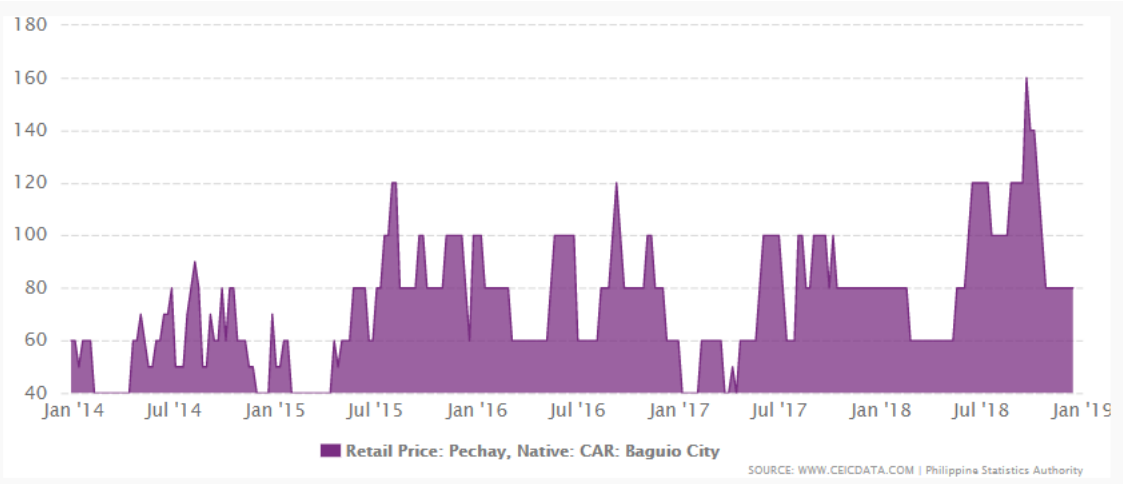


Fig. 2.6 Price of Pechay for 5 years

According to Department of Agriculture, a report was shown as of June 2020 that states the price of pechay in selected markets in Metro Manila. The native pechay retails 100.00 PHP/kg when high, and 60.00 PHP/kg.

2.1.3.7 Characteristics

According to the experiment done by Ultra (2018), the researchers mentioned that pechay weight is affected by different growing ways and fertilization. One of the results show that the average fresh weight of pechay was highest at 214.6g and lowest at 2.0g. Trials used with urea had a higher yield as compared to those grown in soil alone. Depending on which fertilizer used, the size of the pechay was growing in a different rate.

2.1.4 Grow Room

According to Espiritu & Nielsen (2020), plants grown outdoors enjoy the effects of light breezes, sunlight, evaporation, and precipitation. Both have emphasized that air circulations are necessary in grow room. These are used to remove excess heat, control humidity, prevent diseases, and control carbon dioxide. Espiritu and Nielsen stated that humidity is another factor that can contribute to problems with indoor growing systems.

Passive and active intake are also discussed here. These are two different ways to bring air into the grow room. The former one relies on passive airflow and negative pressure. With this ventilation system, it has a fan blowing air out but no fan blowing air in. Instead you have an intake hole that passively allows new air to enter. The latter one actively draws air into your grow room with a fan which means that one fan is blowing air out and another blowing air in.

Plants emit water vapor throughout the day which increases the humidity of your grow room. Uncontrolled humidity leads to poor growing conditions and can result in the development of pests and diseases. If humidity is too high, one can simply vent the humid air out using an exhaust system, this is to make the air from the intake less humid than the air inside the room. If humidity is too low, temporarily turning off exhaust fans will raise the humidity of the room quickly.

They have also mentioned that placing the fans at optimum location are also necessary since air must reach all areas of the room. Placing fans directly at the plant should be prevented as these can cause wind burns.

Another topic discussed by HTG Supply (2017) talked about ventilation in grow rooms. They have mentioned that air circulation fans are needed to move that fresh air

across the plant's leaves, to which one is creating an environment more aligned with what a plant would experience outside in the natural habitat it has evolved to thrive in. Plants in their vegetative growth phase need are thriving in the 40-70% range (50-55% is considered optimal). The air around plants can become extremely humid if the plant is transpiring at a healthy rate, but since the plants aren't outside, all that humidity does not just go up into the sky or blow away on the wind. Instead, it stays all around the plant.

In addition, Smsaleh (2019) mentioned that oscillating fans do the job of moving the air around the grow room. Whether inline fans are on or off, it is important to prevent CO2 or humidity buildup around the leaves. Therefore, oscillating fans are likely to be running for more hours comparing to inline fans which makes in most cases should be turned on 24 hours a day.

2.2 Lacking in the Approaches

These studies do not apply extreme image processing as detailed computer vision is a very large area of research. Since these deal with a lot of data and numbers, it can consume time and is prone to errors. The nature of image processing, most especially the 3D space, contains lots of processes to ensure that information received will be precise and correct.

2.3 Summary

The technology in the agriculture industry has been evolving rapidly. The basic procedures involved such as seed planting and watering are slowly becoming fully automated with the help of the latest systems being created. With this, the requirement of human interaction with their garden is minimized with the efficient use of technology.

The need for people to be diligent about having and maintaining a prosperous garden is lessened. The machine in turn is given the duties of a human gardener.

Chapter 3 THEORETICAL CONSIDERATION

CONTENTS

3.1 Soil Moisture Sensor.....36
3.2 Temperature and Humidity Sensor.....37
3.3 Stepper Motor.....38
3.4 Crop.....38
3.5 Summary.....40

3.1 Soil Moisture Sensor

Soil moisture sensors are heavily used in the automation of irrigation systems in gardens may it be indoors and outdoors. These sensors measure the volumetric water content of the soil in a specific region of interest. It uses capacitance to measure the water content through its dielectric permittivity attribute of the soil by just simply inserting the sensor into the area to be tested. The measured moisture level is reported in both analog and digital output.

Fig. 3.1 Soil Moisture Sensor

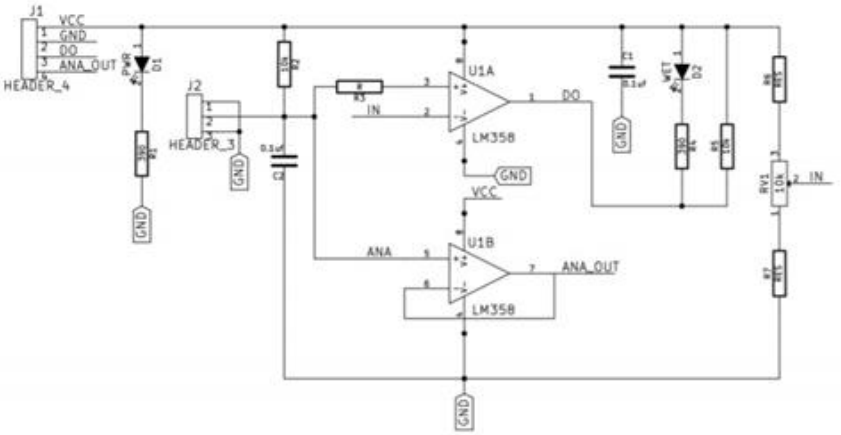


Fig. 3.2 Schematic Diagram of the Soil Moisture Sensor Module

The values returned by this sensor can be categorized into three main segments:

Range	Condition
0-300	Dry
301-700	Moist
701-1000	Wet

Table 3.1 Soil Moisture Sensor Intervals

3.2 Temperature and Humidity Sensor

This temperature and humidity sensor is also known as a hygrometer that measures and reports both moisture and air temperature. This module is made up of a capacitive humidity sensor and a thermistor. It also consists of a very basic chip that is responsible for analog to digital conversion.

Fig. 3.4 DHT22 Sensor Module

The values returned by this sensor can be categorized into three main segments:

Range	Condition
0%-30%	Dry
31%-60%	Moist
61%-100%	Wet

Table 3.2 Relative Humidity Interval

3.3 Stepper Motor

A stepper motor is a type of synchronous electric motor that converts digital pulses into mechanical shaft rotation. It can only take a step at a time and each step is identical. As the digital pulses increase in frequency the step movement changes into continuous rotation, with the speed of the rotation directly proportional to the frequency of the pulses. The control of these motors using the Arduino should be monitored and calculated so that the movement of the profiles along the y-axis will not go over the edge.

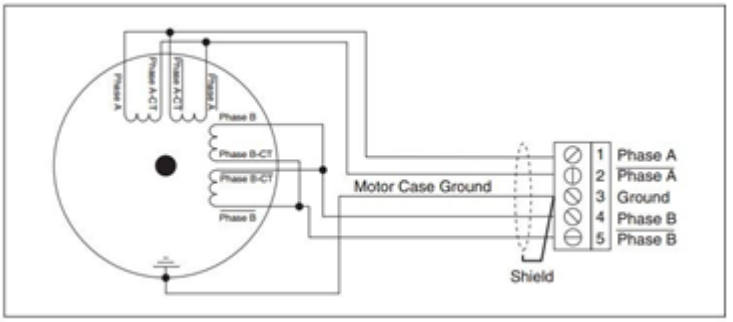


Fig. 3.5 Stepper Motor Connection

3.4 Crop

Considerations must be done to ensure that the cartesian robot can perform its seeding and for the controlled environment to provide its ideal environment. An option for this is to plant pechay. Pechay is about 15-30 cm tall in the vegetative stage. Its seeds are small and circular. The leaves are enlarged and grow upright. This crop can be easily harvested within 30-40 days when taken good care of (Jimenez, 2010). Along with its quick growth cycle, pechay does not require high maintenance, it grows all year round which makes it convenient for different environments and requires only a little space. It

is tolerant of many soil types and temperature zones. Its price average is around 80.00 PHP/kg depending on the region (Philippine Statistics Authority, 2020).

The second option is arugula. Similar to pechay, its seeds are a bit circular. This one, however, prefers a cooler temperature (10°C to 18°C) and partial sun. This is widely used in salads because of its packed nutrients. This plant prefers an evenly moist soil. The fastest-maturing varieties are ready for harvesting just 35 days after sowing; however, most are ready after 50 days (Melchor, 2020). Its space also requires attention depending on the current size. The price of organic arugula is 180.00 PHP/ 250g (Green Grocer Manila).

The third option is kangkong. This is a relatively low-maintenance plant. It just requires a lot of water and a decent amount of fertilizer (Kurtz, 2020). This can be grown from seeds or cuttings, then place the plants in either soil or a pot filled with water. This plant thrives in a temperature of 75 to 85 °F (24 to 29 °C). However, it can be damaged if the temperatures drop below 50 °F (10 °C). Similar to pechay, this should be irrigated every 1 to 2 days, or when needed. The leaves can be harvested in as early as 60 days from sowing. Kangkong retails for 65.00 PHP/ 250g (Green Grocer Manila).

Aside from the crops mentioned earlier, the website GrowVeg features a lot of guides on how to grow and take care of different types of crops. They also specify the difficulty of growing a specific crop, spacing requirements, as well as the environment it needs. The website also tells when is the ideal climate to grow one outdoors.

The group has chosen pechay as the crop mainly due to its fast-growing pace and low-maintenance requirements. Because of its characteristics, it is ideal for the researchers to use this to achieve the objectives flawlessly. The light, smooth, and round shape of the seed makes it easier for the syringe to pick it up. The green leaves also

provide a great contrast with the soil that aids with the image processing for monitoring the growth.

3.5 Summary

With the rise of people living in the city, most people have limited time, thus discourages them to do other things. With everything automated and easily accessible, this could be of a good catch to users most particularly those who are extremely busy. Through this technology, users would not need to worry about too many things. This prototype features a controlled environment that lessens the struggles of weather conditions in the metro. Data from the camera and sensors are available through the mobile application that can be monitored anytime. These sensor values obtained are collected and analyzed as operations are dependent on them.

Chapter 4 DESIGN CONSIDERATION

Contents

4.1 Overview 44
4.2 Growth Chamber 45
4.2.1 Chamber Assembly 45
4.2.1.1 Cabinet 47
4.2.1.2 Aluminum Foil Insulation 48
4.2.2 Electronics System 49
4.2.2.1 System Circuit 50
4.2.2.1.1 Thermal Management 50
4.2.2.1.2 16 Channel Relay Module 50
4.2.2.1.3 Terminal Block 52
4.2.2.2 Power Supplies 52
4.2.2.3 Raspberry Pi 4B 53
4.2.2.3.1 Step-Down Voltage Regulator 54
4.2.2.3.2 Libraries 54
4.2.2.3.2.1 Pygame 54
4.2.2.3.2.2 Datetime 55
4.2.2.3.2.3 Numpy 55
4.2.2.3.2.4 Plantcv. 55
4.2.2.3.2.5 Cv2 55
4.2.2.3.2.6 Os 55
4.2.2.3.2.7 Sys 55
4.2.2.3.2.8 Rpi.gpio 56
4.2.3 Lighting System 56
4.2.3.1 White LED Light 56
4.2.3.2 Grow Lights 56

4.2.4 Air Circulation System 59
4.3 Cartesian Robot System 62
4.3.1 Cartesian Robot Assembly 63
4.3.1.1 System Components 63
4.3.1.1.1 Stepper Motors 63
4.3.1.1.2 DRV 8825 64
4.3.1.1.3 Limit Switches 65
4.3.1.2 Gantry 66
4.3.1.3 Z-Axis 68
4.3.1.4 Cross Slide 68
4.3.2 Plant Box 69
4.3.3 Seeding System 70
4.3.3.1 Components 70
4.3.3.2 Circuit 71
4.3.3.3 Software 72
4.3.3.4 Uniform and Variable Seed Distribution 78
4.3.4 Irrigation System 79
4.3.4.1 Components 80
4.3.4.2 Circuit 80
4.3.4.3 Software 81
4.3.4.3.1 Raspberry Pi Automatic Irrigation Function 82
4.3.4.3.2 Raspberry Pi Adaptive Irrigation Function 83
4.3.4.3.3 Arduino Irrigation System Function 83
4.4 Connectivity System 85
4.4.1 Django Framework 85
4.4.1.1 MVT Pattern 85
4.4.1.1.1 Models 86
4.4.1.1.2 View 87
4.4.1.1.3 Template 88
4.4.1.2 Database 88
4.4.1.3 Asynchronous Connectivity 90
4.4.2 Mobile Application 93
4.4.2.1 Dashboard Page 93
4.4.2.1.1 Side Navigation Bar 95
4.4.2.1.1.1 Mode Selector 96

4.4.2.1.1.1.1 Mode 1 96
4.4.2.1.1.1.2 Mode 2 97
4.4.2.1.1.1.3 Mode 3 98
4.4.2.1.1.1.4 Mode 4 99
4.4.2.1.1.2 GrowSmart Controller 99
4.4.2.1.2 Monitoring System 101
4.4.2.1.2.1 Sensors 101
4.4.2.1.2.2 Vision System 102
4.4.2.2 Database Page 103
4.4.2.3 Sensors Page 112
4.4.2.4 Development of Mobile Application 114
4.4.3 ZeroTier One 115
4.5 Monitoring System 116
4.5.1 Vision System 116
4.5.1.1 Logitech C310 Camera 116
4.5.1.2 Software 117
4.5.2 Sensors 122
4.5.2.1 Temperature and Humidity 125
4.5.2.2 Soil Moisture 126
4.5.2.3 Threshold Values 127
4.6 Summary 128

4.1 Overview

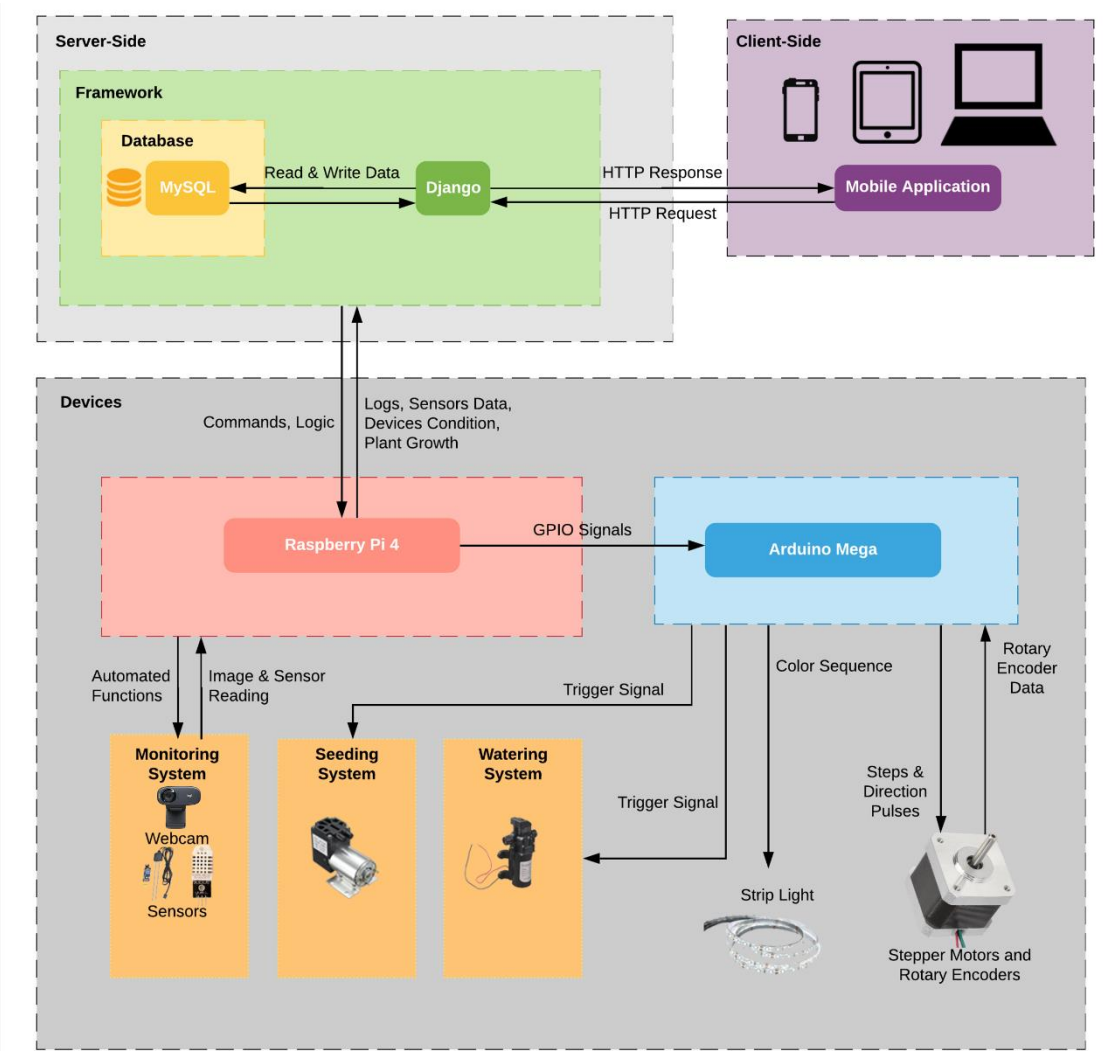


Fig. 4.1 High-level Block Diagram of GrowSmart

Figure 4.1 provides an overview of GrowSmart's system which is composed of a connectivity, monitoring, and cartesian robot system consisting of the seeding and irrigation system. The connectivity system, hosted by the Raspberry Pi, is shown on the upper half of the diagram. This is divided into two main parts - the client side and the

server side. The client side is represented by the mobile application that is controlled by the user while the server side is responsible for processing incoming and outgoing requests and then displays the output to the mobile application. The monitoring system consisted the camera, as part of the vision system, and sensors that are managed by the Raspberry Pi through automated functions. These functions include the automated time checker, temperature, humidity, and soil moisture sensor checker, and automated image capturing and processing. These interconnected systems are managed by the Raspberry Pi and are monitored by the user through GrowSmart's mobile application. The seeding and irrigation system consist of pumps and motors that controlled the movement of the cartesian robot managed by the Raspberry Pi and controlled by the Arduino Mega. The process consists of the Pi sending GPIO signals to the Mega through which the latter then analyzes its operation. This is mostly used for controlling the cartesian robot and lights since the Mega contained the functions that controlled the lights and actuators. Since the Mega controlled the pumps, it is in-charge of the two major operations – seeding and watering.

4.2 Growth Chamber

4.2.1 Chamber Assembly

The cabinet-type growth chamber is made of wood and its walls were lined with aluminum foil insulation. This was assembled by the researchers by measuring large planks of wood and then cutting them into pieces needed to build the chamber. After this, the researchers put together the pieces of wood using nails to make the cabinet structure.

Fig. 4.2 Growth Chamber Closed Front View

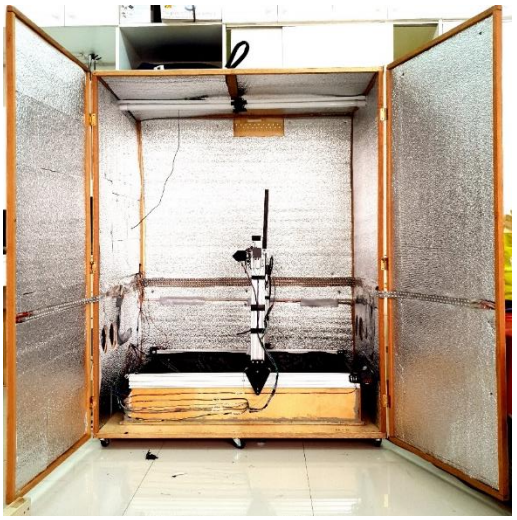


Fig. 4.3 Growth Chamber Open Front View

Fig. 4.4 Growth Chamber Side View

4.2.1.1 Cabinet

The cabinet enclosure is 48 inches in length, 30 inches in width, and 61 inches in height. Its material is made of a ¼ inch thick plywood for all sides excluding the bottom piece, which was 1 inch thick. The researchers designed the cabinet this size to make sure that the space for the plant box, cartesian robot, camera, lights, and wiring was enough to fit everything. Wheels were screwed on the four corners at the bottom of the cabinet and an additional three were screwed in the middle to provide better support and mobility. Small holes were drilled at the back for air circulation and medium-sized holes on the left and right side were drilled for the fans.

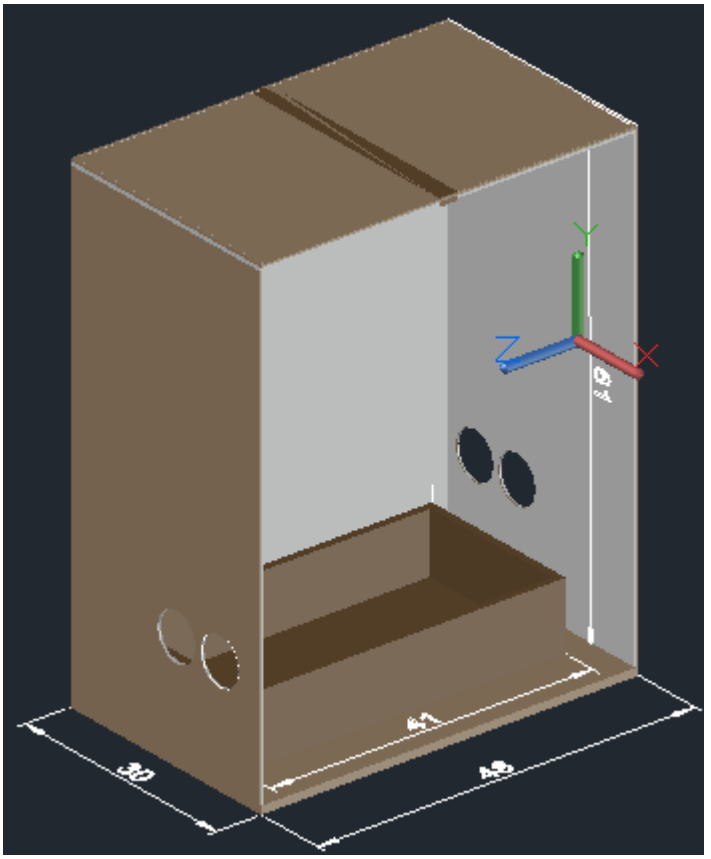


Fig. 4.5 Cabinet Sketch Structure

4.2.1.2 Aluminum Foil Insulation

Choosing the right surface for the walls of the cabinet enclosure was important. As artificial lighting diminishes exponentially with distance, it needs to contain as much of this light as possible. Aluminum foil insulations were a great option since those are highly reflective. The insulation was installed on four walls and ceiling with the shiny side facing outwards. The insulation was for the light to reflect properly enabling it to supply the sufficient amount to the crops for growth improvement (Williams, n.d.).

4.2.2 Electronics System

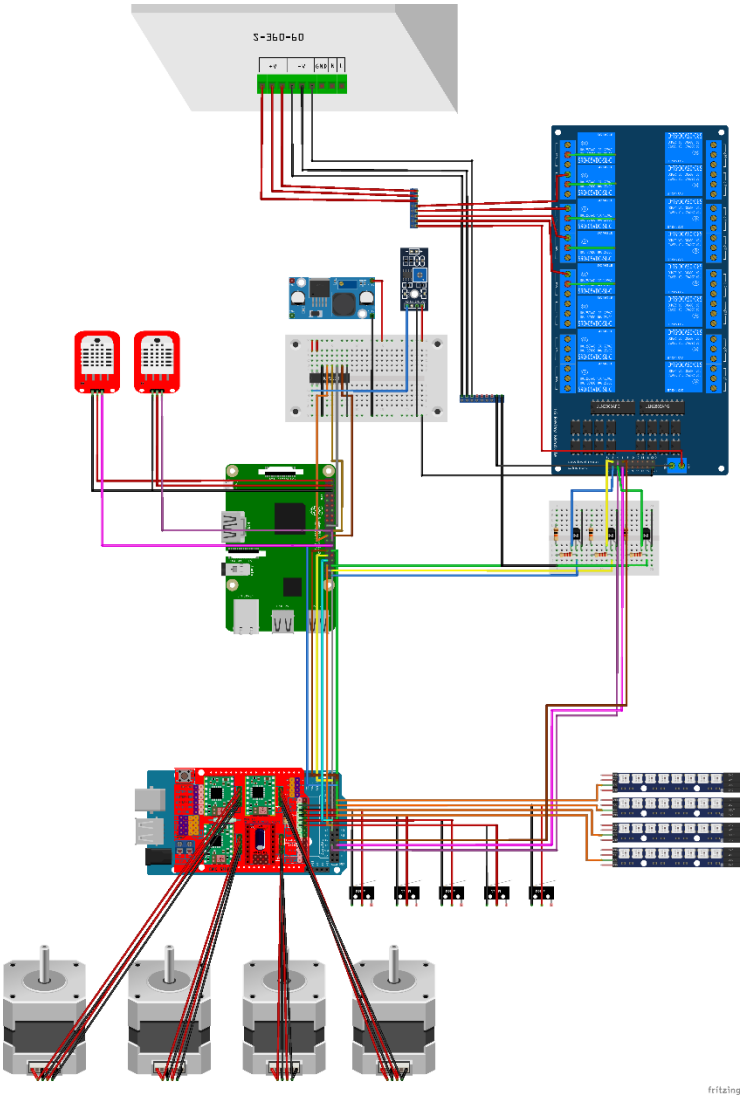


Fig. 4.6 Electronics System

Figure 4.6 provides an overview of the electronics system circuit of GrowSmart. As shown in the figure, the researchers used a Raspberry Pi and an Arduino Mega as the controllers of GrowSmart. The Pi controlled DHT22, soil moisture sensor, light, fans,

and sent signals to Mega. The Mega on the other hand controlled the light strips and utilized the CNC shield for the stepper motors.

4.2.2.1 System Circuit

4.2.2.1.1 Thermal Management

Heat sinks were used to absorb and disperse heat away from a high temperature object. This was added to prevent the electrical components such as the Raspberry Pi, stepper motors, and relays from overheating. Table 4.1 shows the temperature ranges of the listed components.

Components	Temperature Range
Raspberry Pi 4	0°C to 70°C
Arduino Mega	-25°C to 50°C
CNC Shield	-40°C to 85°C
Stepper Motors	-40°C to 80°C

Table 4.1 Components Range Interval

4.2.2.1.2 16 Channel Relay Module

The 16 channel relay module was powered by the 12V power supply. This was used as a switch for the components that needed to be turned on and off such as the fans, white LED light, vacuum pumps, and water pump. Each device connected to the channels used a normally open terminal so that they won't turn on unless triggered by the Raspberry Pi or Arduino. Additionally, the LED light and fans had their digital input pins connected to a circuit containing an NPN transistor to convert the relay pins since the relay board needed an active low signal to activate the channels and also because the Raspberry Pi cannot output an active low signal. On the other hand, the connections for the water and vacuum pump were direct since they are controlled by the Arduino which provided an active low signal. Based on Figure 4.8, the blue wire represented the LED light, yellow wire represented the left fan, and green wire represented the right fan. In

Figure 4.9, the pink and purple wires represented the vacuum pumps and the brown wire represented the water pump.

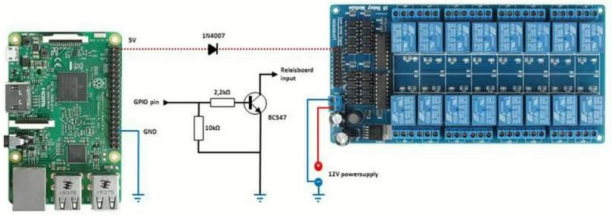


Fig. 4.7 Relay Wiring to Raspberry Pi

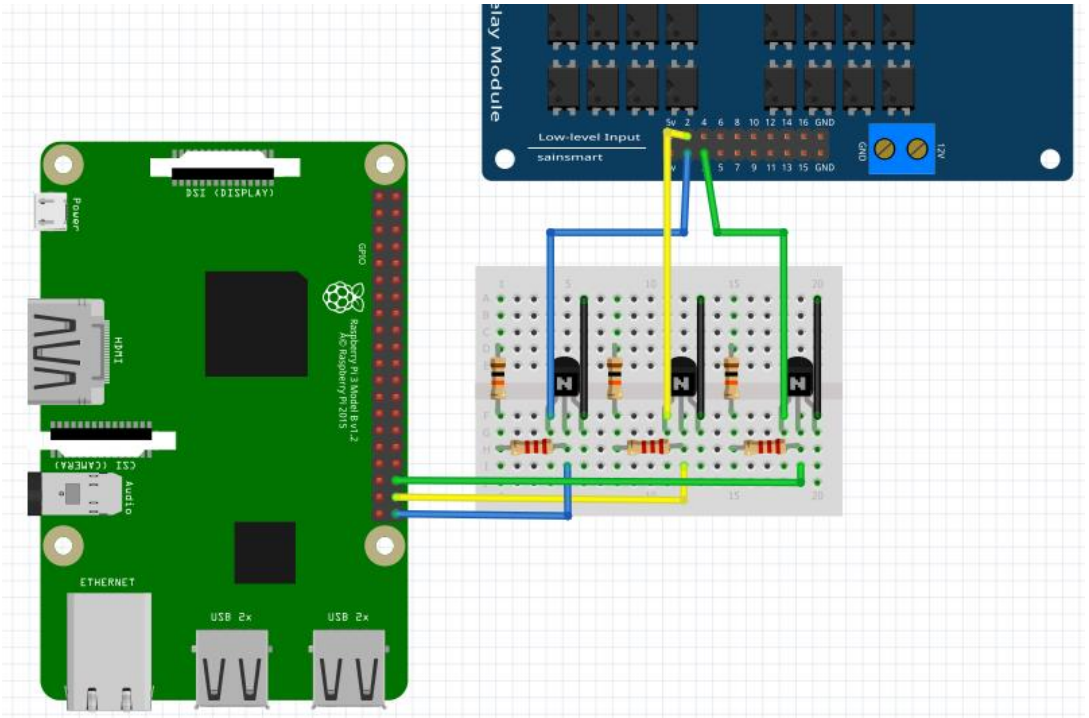


Fig. 4.8 Relay Wiring of LED and Fan with Raspberry Pi

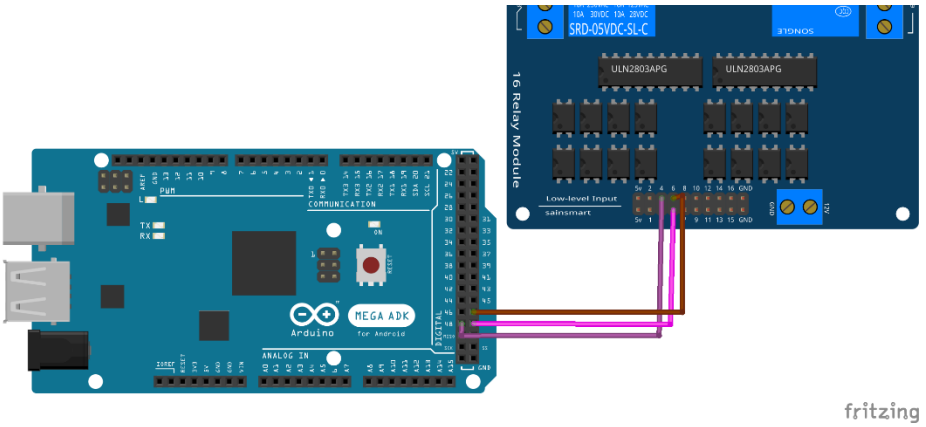


Fig. 4.9 Relay Wiring of Water and Vacuum Pump with Arduino Mega

4.2.2.1.3 Terminal Block

Terminal blocks were used for organization purposes since the prototype required lots of wires. The group used these to create a common terminal for the power supplies since the components used needed different amounts of voltage.

4.2.2.2 Power Supplies

A 12V 30A power supply was used for the vacuum pumps, water pump, relay, fans, and CNC Shield. One 5V 10A supply was used to power all the sensors (two DHT22, soil moisture sensor). Lastly, four 5V 10A supplies were used for the light strips.

To estimate the power supply needed for the grow lights, four strips containing 178 neopixels each were used. Each pixel used at maximum brightness may consume 60 milliamperes. This was to make sure that the power supplies used were enough to display the colors necessary for growing plants.

178 Neopixels x 60mA = 10.68 minimum Amps

Fig. 4.10 Power Supplies

4.2.2.3 Raspberry Pi 4B

The version 4 Model B 4GB ram of the Raspberry Pi series was used since it packed a powerful Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz processor and 4 gigabytes of ram enough to power the whole system. It acted as the main computer of the entire system because of its powerful tools and libraries that were aligned with what the researchers needed to develop the system. The Raspberry Pi 4B's main job was to host GrowSmart's server built using Django and Python 3. The Raspberry Pi 4 Model B required the latest Raspbian Operating system, Buster. Raspbian Buster is a free operating system available on the official website of Raspberry Pi. The researchers installed this operating system and peripherals such as a monitor, mouse, keyboard, and a 3A 5V USB-C power supply were used as part of the basic requirements for the Pi to function. At first boot, the researchers configured the installation of the operating system by setting the location, time zone, name of the Pi, and the Wi-Fi configuration. After the successful configuration, the Pi proceeded to the actual Desktop

GUI where the researchers installed the required packages and libraries to develop and run the system.

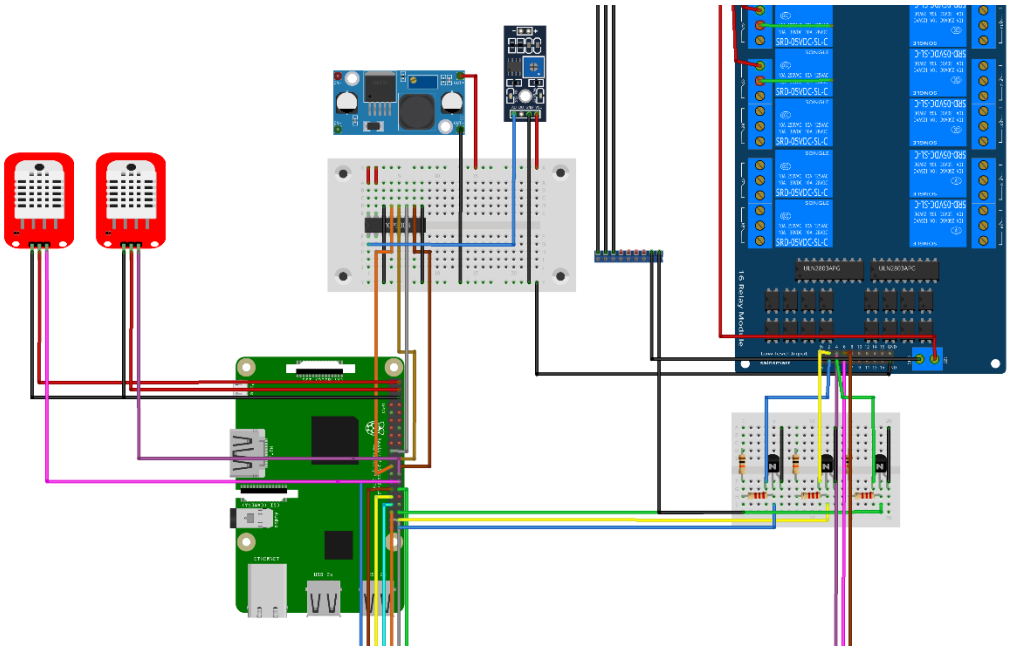


Fig. 4.11 Raspberry Pi Connection with Submodules of System

4.2.2.3.1 Step-Down Voltage Regulator

A voltage regulator was used to step down the voltage from 5V to 3.3V. This is used as a 3.3V supply for the analog to digital converter, MCP3008.

4.2.2.3.2 Libraries

4.2.2.3.2.1 Pygame

The Pygame library was used in this study to capture photos using the Raspberry Pi and Logitech C310 camera

4.2.2.3.2.2 Datetime

The datetime library was used in this study for manipulating dates and times to calculate the age of the gardening system based on the first every entry and the latest entry in the counters database.

4.2.2.3.2.3 Numpy

The Numpy library was used in this study for both the plantcv image processing and soil moisture sensor data acquisition. It adds support for multi-dimensional arrays and matrices.

4.2.2.3.2.4 Plantcv.

Plant growth was monitored by using PlantCV. This is an open-source image analysis software package particularly focusing on plant phenotyping. It is composed of modular functions on a variety of plant types to extract data from images. This was installed by using the command pip install plantcv on the Raspberry Pi

4.2.2.3.2.5 Cv2

The Cv2 library is the library of OpenCv2 which was used in this study as part of the required libraries in using PlantCV.

4.2.2.3.2.6 Os

The Os library was used in this study to allow the researchers to interface with the underlying operating system that Python is running on.

4.2.2.3.2.7 Sys

The Sys library was used in this study to provide information about constants, functions, and methods so that the researchers could access them anywhere in the code.

4.2.2.3.2.8 Rpi.gpio

The Rpi.gpio library was used in this study to control and set the modes of the gpio pins on the Raspberry Pi.

4.2.3 Lighting System

4.2.3.1 White LED Light

A 20W T8 LED tube box type fixture was attached at the topmost part of the cabinet that provided white light for better image capturing and processing since the camera does not take clear shots without proper lighting.

Fig. 4.12 LED Light in Operation

4.2.3.2 Grow Lights

LEDs offer high efficiency, small size, durability, long operating lifetime, and low thermal emission. They are cheaper and they last longer compared to fluorescent lights. Depending on the stage of the crop, the germination stage requires an average of 16 hours per day. For the growing stage, estimated hours of 12-16 are recommended for edible crops under artificial lighting (Harrington, 2016). These blue wavelengths encourage vegetative growth through strong root growth and intense photosynthesis while red wavelengths encourage stem growth. The ratio for red: blue used for the LED

strip lights is 3:1. This was used to replicate the light spectrum of the sunlight. It used specific wavelengths around 400 to 700nm similar to photosynthesis that provided energy needs of the plant (Watkins, 2018). For vegetative stages, 18-24 inches away from the LED grow light is recommended for feeding the plant without it burning.

The effect of intensity and spectrum of light on the efficiency of photosynthesis and productivity of various plants all differ. Light strips were lit with a 3:1 ratio of red (640-660 nm) and blue (433-450 nm) using an Arduino Mega.

LED Grow Lights Wattage	Distance From Plants
1 watt	16 inches or 30 cm from your plan
3-5watts	20 inches or 30 cm from your plants
6-10watts	21 inches or 53 cm away from your plants
200 watts	12 to 20 inches away from your plants
250- 400watts	16 to 30 inches away from your plants
450- 550 watts	20 to 30 inches away from your plants
600- 800 watts	28 to 42 inches away from your plants
900- 100 watts	36 to 46 inches away from your plants

Fig. 4.13 Grow Lights Distance Table

Photo Source: www.urbanorganicyield.com/grow-light-distance-from-plant/

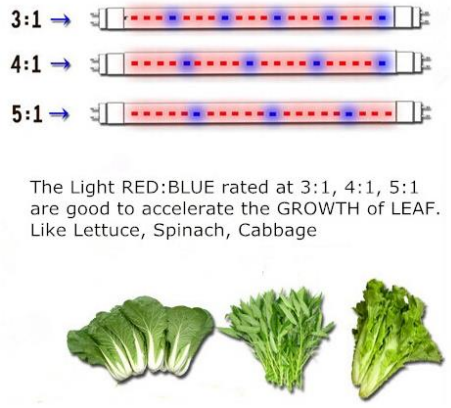


Fig. 4.14 Red:Blue Ratio for Plant

Photo Source: www.led-lights.com/different-red-blue-ratio-a-140

The plant box was kept in a controlled environment with artificial lighting using grow lights attached to the walls. Setting up the grow lights from a specific distance was important to ensure that the crops get the color intensity they need to grow. As height is dependent on the light wattage, the researchers used four 50-watt lights that require a 12 to 20-inch distance from the crop. In this case, the group placed the LED strips approximately twenty inches above the soil surface. Each strip was connected to their respective power supplies and each data out was connected to the digital pins of the Arduino.

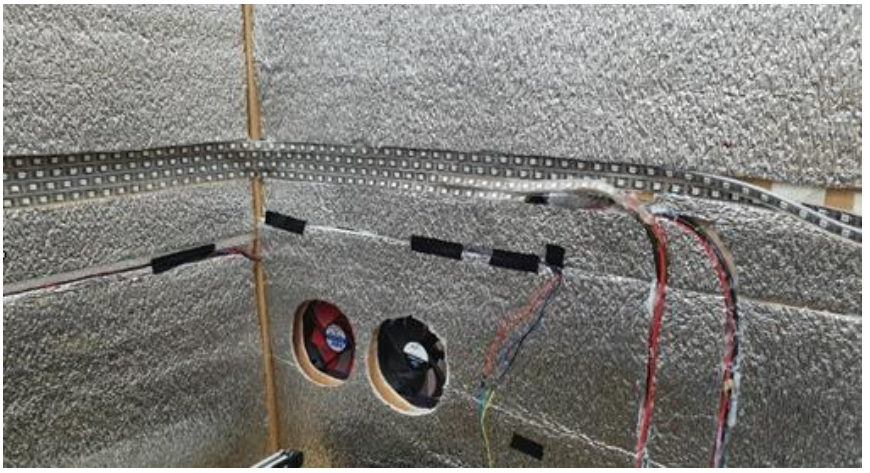


Fig. 4.15 Grow Lights

This was implemented using Neopixel strip lights and by including the <Adafruit_NeoPixel.h> library to light the digital RGB light strip. Since four strips were used in this design, each of their DO (data out) had their corresponding digital pin connected in the Arduino Mega. To provide the ideal 3:1 ratio, the group used a modulo function to alternate the values of red and blue along the whole strip.

Fig. 4.16 Grow Lights in Operation

4.2.4 Air Circulation System

A proper ventilation system is necessary as grow rooms require continuous air flow between the outside and indoor area. A grow room is about simulating the optimal outdoor conditions inside four walls, which also means maintaining the perfect carbon dioxide levels for plants (Max, 2019). Max has also mentioned that a ventilation system is crucial to keep the CO2 content inside the grow room at optimal levels for the plants. As the concentration of CO2 in the atmosphere is low, a constant flow of fresh air from outside is necessary.

Four 12V 1800 rpm fans were placed diagonally above the plant box. Two were used for air intake and another two as exhaust fans for proper air circulation. They were mounted approximately 5cm above the height of the plant box so that the air coming from the outside will not directly affect the crop. More so, they were placed diagonally to make sure that the air inside the cabinet reaches all corners of the box properly. Holes were also drilled at the back of the cabinet to ensure proper ventilation. Table 4.2 shows the fan specifications.

Item Type	**Cooling Fan**
Size	120 x 120 x 25mm
Rated Voltage	12V
Rated Current	1800RPM+-10%
Connector	2 Pin
Fan Blade	7 Blades

Table 4.2 Fan Specification

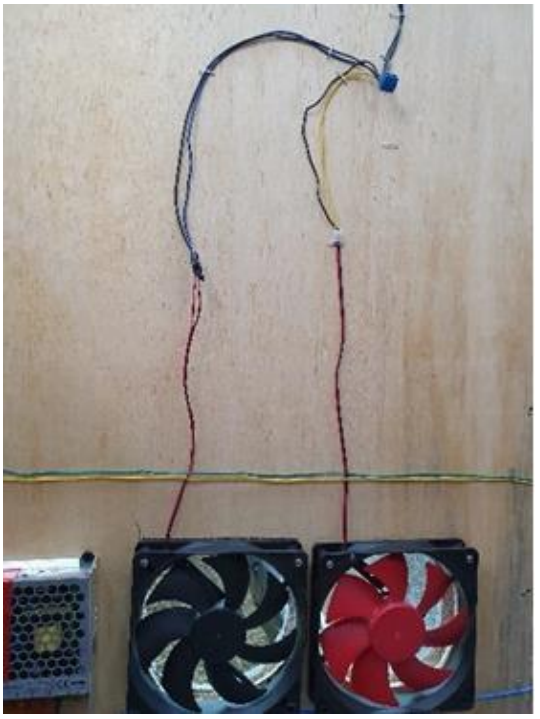


Fig. 4.17 Fan for Air Circulation

Fig. 4.18 Growth Chamber Ventilation

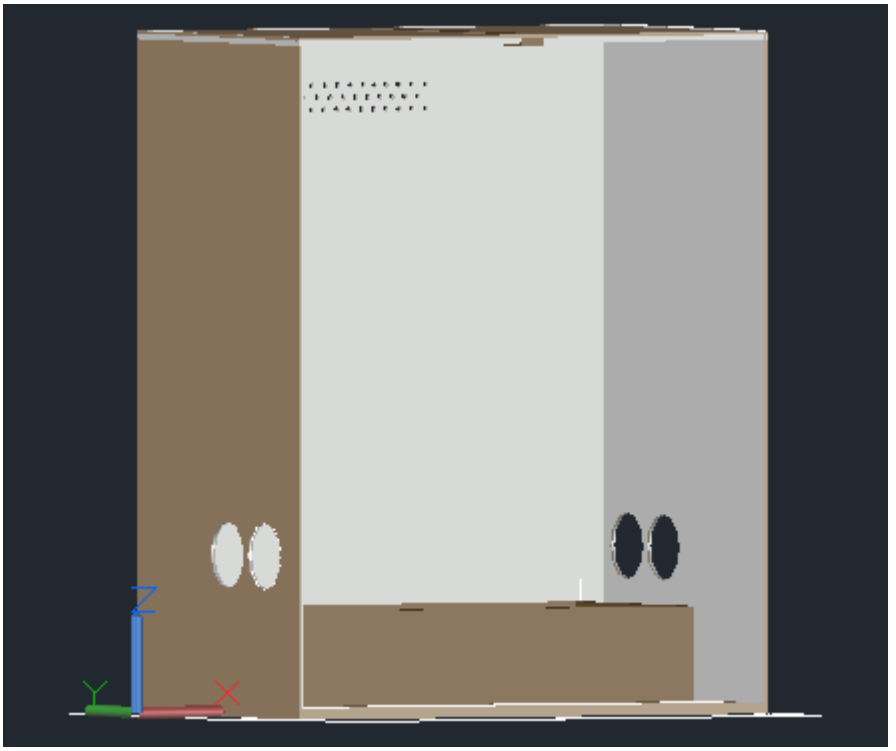


Fig. 4.19 Air Circulation System Sketch

4.3 Cartesian Robot System

The cartesian robot system was used for the seeding and watering operations of the proposed automated gardening system. The XYZ robot, more formally known as the cartesian coordinate robot, is a kind of robot with three principal axes of linear control as seen in Figure 4.20. It is commonly used in industrial operations that require a high level of precision and applied in 3D printers and other computer numerical control machines.

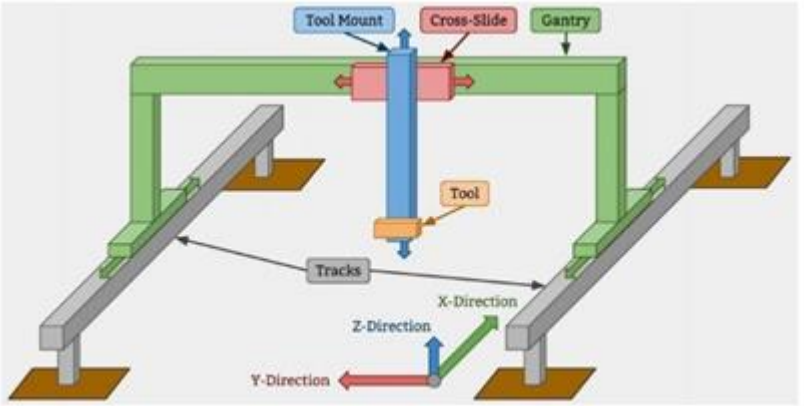


Fig. 4.20 XYZ Movement

Fig. 4.21 Prototype

4.3.1 Cartesian Robot Assembly

4.3.1.1 System Components

4.3.1.1.1 Stepper Motors

For the whole cartesian coordinate robot to move along each axis, four NEMA 17 stepper motors were used. The prototype utilized a CNC Shield V3 which enabled the Arduino board to perform CNC controls on the stepper motors. It controlled four stepper motors using four individual stepper motor drivers (DRV8825). The CNC shield moved the stepper motors in 3 different axes (XYZ), however, the fourth controller was used to replicate the movement along X-axis since the prototype used 2 stepper motors along the mentioned axis.

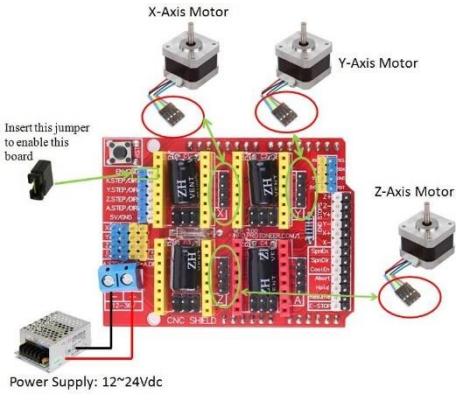


Fig. 4.22 CNC Shield

4.3.1.1.2 DRV 8825

The wiring diagram as seen in Figure 4.23 is a visualization of how each DRV8825 was connected terminal-by-terminal into the CNC Shield V3. Each DRV8825 was tuned to only allow a current value of 2A into each motor to prevent damage. This was done by using a screwdriver to turn the screw on each DRV8825 mounted on the CNC Shield V3.

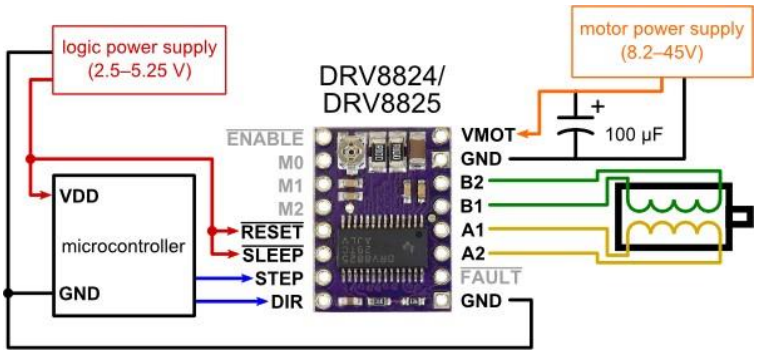


Fig. 4.23 DRV8825 Wiring Diagram

To determine or limit the current going into each stepper motor:

The positive/red probe was placed on top of the screw of the DRV8825 and the negative/black probe was placed on top of the screw on the Negative (-) Power Terminal of the board.

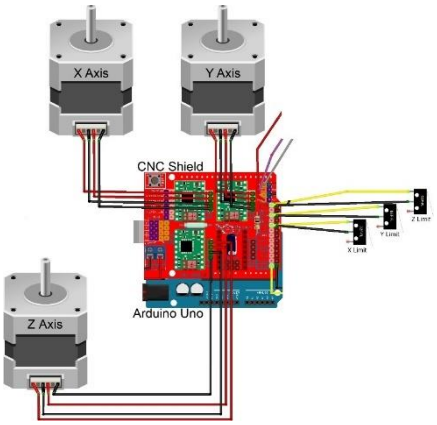


Fig. 4.24 Limit Switch Connection to CNC Shield

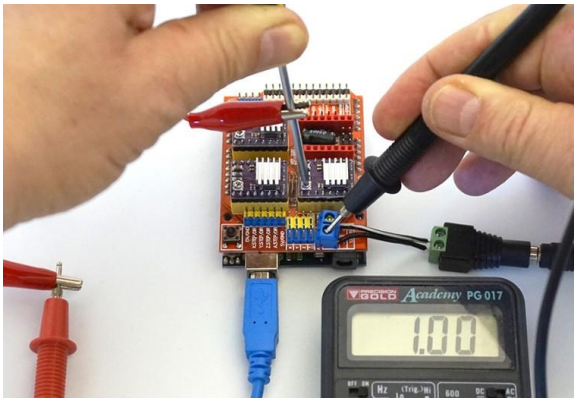


Fig. 4.25 Tuning the DRV8825

4.3.1.1.3 Limit Switches

Limit switches were attached to the aluminum profile to indicate that the robot has reached and touched the end of the aluminum profile. This was done by connecting the common pin and normally open switch to their respective axis. A limit switch was

placed for each axis to calibrate the movement of the cartesian robot for precision and accuracy.

Fig. 4.26 Limit Switch Setup

4.3.1.2 Gantry

The gantry is the structural component that bridged the two tracks and moves along the x-axis via a belt and pulley drive system. It also served as a linear guide for the cross-slide to move across the gantry along the y-axis. The gantry served as a base for mounting other components, devices, and sensors.

Two 500mm aluminum profiles were connected together on each side to form the base for the x-axis. Plates for the gantry were custom-designed and 3D printed. V-wheels were inserted with an M5 screw and then secured with locknuts. The process was repeated until the group achieved attaching two v-wheels on top and one on the bottom. The plates with the wheels were used to connect the extrusion for the x and y-axis. They were secured using nuts and tightened with the help of a wrench. A similar process was done to connect the y-axis and the profile that held the z-axis.

For the stepper motor, the researchers slid the motor into the motor housing, ensuring that the longer shaft of the motor was coming out of the housing and that the motor's wires were facing down through the open end of the housing. The researchers

also slid the gantry motor mount plate onto the shaft and fit it flush against the motor housing.

The timing belt was inserted in the gantry column, making sure that the orientation was correct so that the belt teeth engaged with the motor pulley properly. The belt was then tied into a knot in the plate as previously mentioned and secured with a clip ensuring that the flat side of the belt is in good contact with the v-wheels. Holes were drilled on the side of the plant box to give way for the x-axis. These were mounted using a long M5 screw and tightened with t-nuts to avoid getting wobbly and derailed.

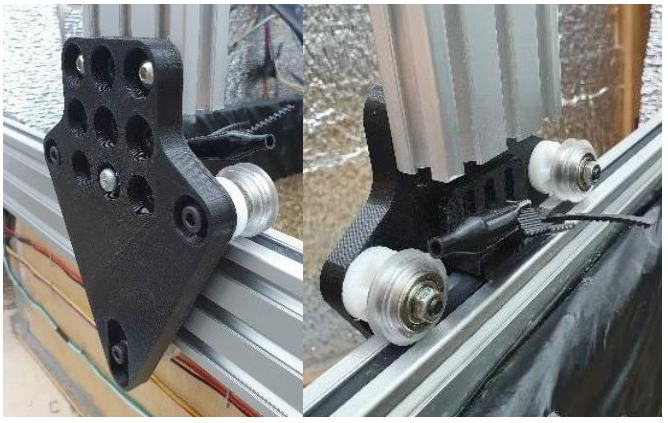


Fig. 4.27 3D Printed Base Fitting

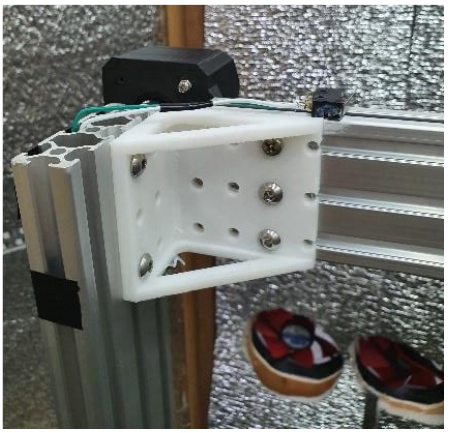


Fig. 4.28 3D Printed Beam Fitting

Fig. 4.29 3D Printed Stepper Motor Housing

4.3.1.3 Z-Axis

Like the other axes, the Z-axis also had a custom-designed plate. This one, however, used a 500mm lead screw to perform vertical motions for seeding and watering. With this design, the lead screw motion was directly in line with the motion of the syringe and dripper. A 1m aluminum extrusion was also attached to the 500mm profile for easier and longer reach of the seeder nozzle.

Fig. 4.30 Z-Axis

4.3.1.4 Cross Slide

The cross-slide moved across the gantry along the y-axis. This motion allowed operations such as seeding anywhere on the x and y-axis. The cross-slide was moved using a belt and pulley drive system and served as the base for the z-axis to attach to.

4.3.2 Plant Box

The plant box was 41x21x9 inches in size and its thickness was 1in on all sides. Small holes were drilled around the plant box and a plastic layer was placed to make sure that the wood would not be easily damaged by water over time.

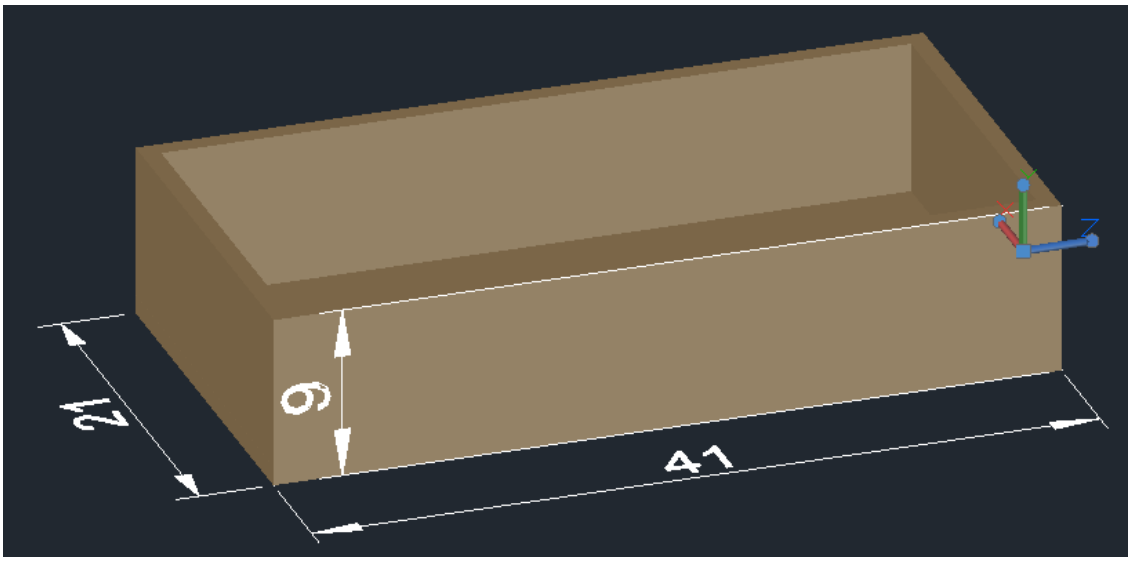


Fig. 4.31 Plant Box Sketch

Attached to the plant box was a cone-shaped container placed towards the edge of the plant box for seed picking. This shape was ideal as it allowed the seeds to fall into the center that provided convenience for the syringe to perform the seed picking easily.

Fig. 4.32 Seed Container

As the thesis only covers crops with a growth cycle of 30 days, the researchers used approximately 15kg of loam soil with 15 to 20 percent vermicast since vegetable crops grow best with these. Loam is the ideal soil for growing vegetables because it has an even balance of clay, sand, and organic matter therefore it holds sufficient moisture, oxygen, and nutrients for plants (Green, n.d.). As pechay belongs to the Brassicaceae family, these crops thrive best in loam soil.

4.3.3 Seeding System

GrowSmart's seeding system was composed of a syringe tip, a Y tube, and vacuum pumps. The characteristics such as the size, weight, and shape may have an impact on the effectiveness and reliability of the syringe tip connected to the vacuum pump. Seeds should not be too small or large, cannot be too heavy, and are ideally of a round, smooth shape. A larger diameter syringe tip should be used for larger, heavier seeds, while a smaller diameter syringe tip should be used for smaller, lighter seeds. If the syringe tip is too small, then the vacuum pump will not be able to create enough suction to consistently and reliably pick up a larger, heavier seed. In some cases, the seed may get picked up but then dropped when the robot is along the axes.

4.3.3.1 Components

The researchers used a syringe needle tip as its seeder nozzle. It acted as a suction which allowed it to pick up seeds from the container. Assuming the diameter of the seeder nozzle was 1 mm less than that of the seed, the small tip allowed the mechanism to suck one seed at a time which made the number of seeds planted precise and fixed.

This pump was powered by a 12V power supply with a 7mm tube connected to the syringe. This provided sufficient air pressure to pick up the seed. A y-tube was used to connect the 2 vacuum pumps and the syringe. One vacuum pump was used as a suction for the syringe to pick up a seed from the seed container while the other was used to release air pressure once it had reached approximately 1-3cm below the soil's surface.

Fig. 4.33 Seeder Setup

4.3.3.2 Circuit

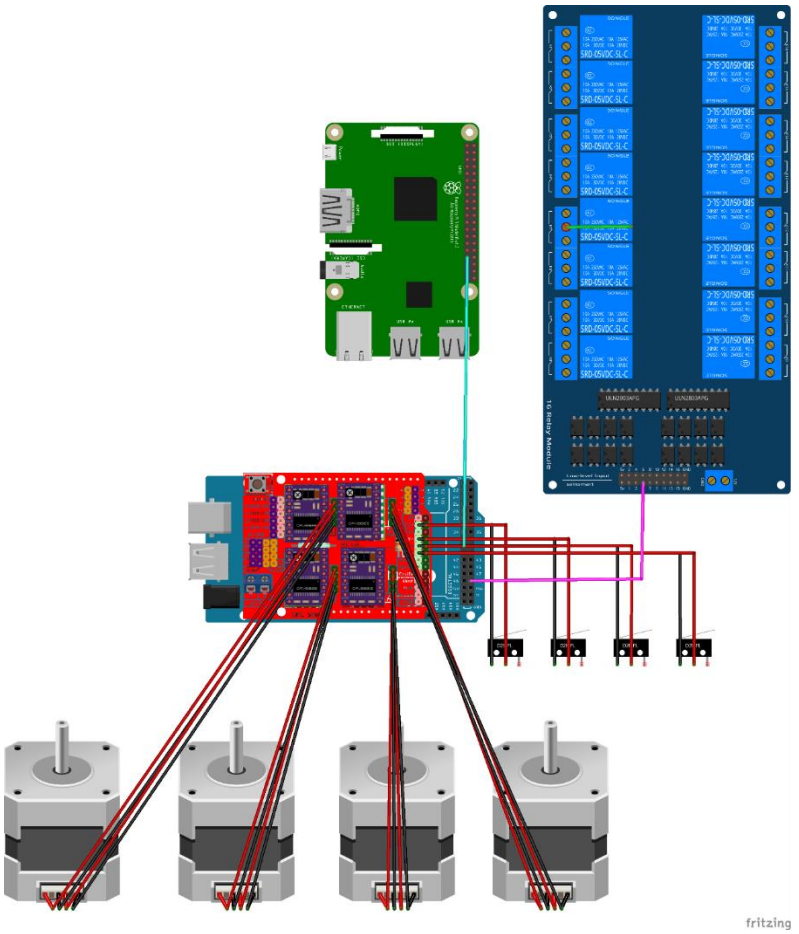


Fig. 4.34 Circuit Used for Seeding

The seeding process utilized the Raspberry Pi and Arduino Mega with the former acting as a trigger while the latter containing the actual code for the motors and vacuum pump. The process began with the Raspberry Pi sending a signal to the Arduino Mega though their GPIO pins represented as the cyan wire connecting the two devices as seen in Figure 4.34. The signal sent by the Pi was turned off after 1 second to avoid the seeding function on the Arduino to repeat. After a signal was sent to the Arduino, the Raspberry Pi records the seeding process on the database for monitoring purposes.

4.3.3.3 Software

Other components of the cartesian robot system included the stepper motor, along with the water and vacuum pumps, which were controlled by the Arduino Mega using its libraries and functions. These functions were put in several header files to fix the code in a modular manner an make it easily readable. This method also provided abstraction to the user which may help them understand the flow of the program better. Pin configurations, as well as different variables, were also put in separate files to provide easy access to editing.

The header files have .h as an extension and are mentioned below according to which type of code they contain as well as a brief explanation of the codes in it.

CLASSES:

1. Commands.h this contained the 3 main functions of the prototype namely: seeding, watering, and calibration.

Fig. 4.35 Commands Screenshot

2. ModeSelection.h this was responsible for the assignment of spacing and the dimensions of the planting ground in terms of rows and columns depending on the mode set which can be determined from the signal being sent from the RaspberryPi.

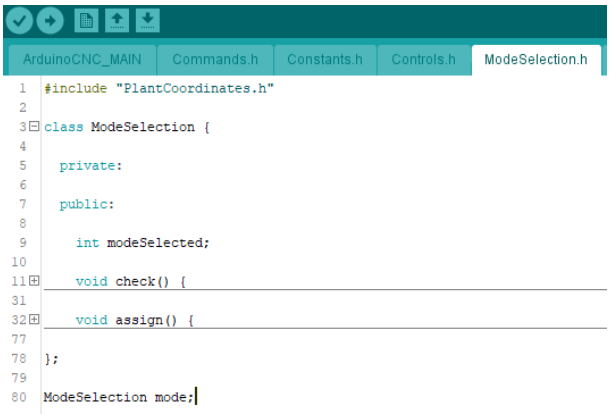


Fig. 4.36 Mode Selection Screenshot

3. Controls.h this was where the functions for the controls of the motors and the pumps were located.

Fig. 4.37 Controls Screenshot

DECLARATIONS:

1. Constants.h this file contained the checkers as well as indicators that were used for the operation of the robot in different scenarios

2. PinConfiguration.h this was where the pin assignments were assigned including the Arduino Mega pins that were connected to the CNC Shield V3 and to the RaspberryPi

3. PlantCoordinates.h this contained the values that dictated the spacing and dimensions of the planting ground based on the different modes available.

The commands –seeding, watering and axis calibration, were done by the Arduino microcontroller. The most important of the three commands was the axis calibration which ensured that the robot moved accurately and precisely in every task it needed to perform. These commands can be found in the Commands.h file.

ArduinoCNC_MAIN | Commands.h | Constants.h

```
class Commands {

  private:

  public:

    // 3-AXIS CALIBRATION
    void beginCalbiration() {


    // SEEDING
    void beginSeedDistribution() {

    // WATERING
    void beginWateringProcess() {

};

Commands AGE;
```

Fig. 4.38 Three Commands Available

Figure 4.38 shows the three commands – beginCalibration(), beginSeedDistribution(), and beginWateringProcess(). These commands were executed by the different motors and pumps. The code was implemented in such a way that it sends coordinates and signals for the motors to move to specific coordinates and signals that turns the pumps ON/OFF.

```
7     // 3-AXIS CALIBRATION
8     void beginCalbiration() {
9
10      // Z-AXIS CALIBRATION
11      if (axis == 'Z') {
26
27      // X-AXIS CALIBRATION
28      else if (axis == 'X') {
57
58      // Y-AXIS CALIBRATION
59      else if (axis == 'Y') {
99    }
```

Fig. 4.39 beginCalibration() skeleton

Figure 4.39 shows the major if-else statements for the beginCalibration() implementation in the Arduino IDE. The variable named *axis* served as the indicator for the current axis that was being moved or calibrated. The calibration algorithm was done in a way where each axis will move back and forth to both ends in each axis that are dictated by the limit switches placed.

```
// Z-AXIS CALIBRATION
if (axis == 'Z') {

  if (digitalRead(Z_LIM) == HIGH && Z_POS == false) {
    MPC.step(true, Z_DIR, Z_STP, stps);
    Serial.println("MOVING Z-AXIS UPWARD.");
  } else if (digitalRead(Z_LIM) == LOW && Z_POS == false) {
    Serial.println("Z_POS TOUCHED");
    Z_POS = true;
    Serial.println("MOTION ALONG Z-AXIS STOPPED.");
    Serial.println("Z-AXIS SUCCESSFULLY CALIBRATED.");
    delay(3000);
    axis = 'X';
  }

}
```

Fig. 4.40 Sample code for Individual axes in beginCalibration()

Figure 4.40 is a sample of the logic behind the limit switches and how they are used to indicate that a certain end has been touched or reached. These limit switches send signals to the microcontroller that the said axis has reached the end and then rotates it in the opposite direction until it reaches the other end. This logic behind this sample code is implemented on the rest of the remaining axes – *X*- and *Y-axis*.

```
// SEEDING
void beginSeedDistribution() {

  mode.check();
  mode.assign();

  // PRINT STARTING COORDINATES
  Serial.println((String)"START \n    --> X : " + X_VAL + (String)" CM , Y : " + Y_VAL +  (String)" CM , Z : " + Z_VAL + (String)" CM");

  for (int i = 0; i < noOfRows; i++) {
    for (int j = noOfCols; j > 0; j--) {
      for (int k = 0; k < noOfSeeds; k++) {
        while (digitalRead(X_LIM) == HIGH) {
          MPC.step(true, X_DIR, X_STP, stps);
        }
        while (digitalRead(Z_LIM) == HIGH) {
          MPC.step(false, Z_DIR, Z_STP, stps);
        }
        MPC.startVacuumPump();
        while (digitalRead(Z_LIM) == HIGH) {
          MPC.step(true, Z_DIR, Z_STP, stps);
        }
        if (j == noOfCols) {
          MPC.step(false, X_DIR, X_STP, stps * j * xSpace + xStart - X_MIN);
        }
        else {
          MPC.step(false, X_DIR, X_STP, stps * j * xSpace);
        }
        MPC.step(true, Y_DIR, Y_STP, stps * (ySeed - Y_MIN) * (i + 1) );
        while (digitalRead(Z_LIM) == HIGH) {
          MPC.step(false, Z_DIR, Z_STP, stps);
        }
        MPC.stopVacuumPump();
        delay(3000);
        while (digitalRead(Z_LIM) == HIGH) {
          MPC.step(true, Z_DIR, Z_STP, stps);
        }
      }
    }
  }

  Serial.println("Seeding process finished.");
  calibrate = 0;
  seed = 0;
  water = 0;

}
```

Fig. 4.41 beginSeedDistribution() code

Fig 4.41 displays the logic behind the seeding implementation in the Arduino IDE. This code used a lot of variables that were declared in a different header file – *PlantCoordinates.h*. This header file contained the specifications of each mode which permitted the user to several settings by which he/she wants to plant on the ground. Depending on the specific crop to be planted, the number of seeds per hole differs. Since the seeder can only obtain one seed at a time, it will travel back and forth from the seed tray to the specific coordinate. The code was implemented in such a way that it fills up the rows first before moving on to the next column.

4.3.3.4 Uniform and Variable Seed Distribution

With the spacious amount of area that can be used for planting different crops, the XYZ robot was programmed to be able to plant in four different modes. These modes differ by dimension and spacing which also set the XYZ robot to travel a particular amount of distance. The modes were limited to four due to space concerns. The modes were dictated by sets of bits that the Arduino Mega receives from the Raspberry Pi. Two digital Arduino Mega pins were allocated for this process and were named as modePin1 (most significant bit) and modePin0. These pins were declared in the PinConfiguration.h header file which contained all the pins that are being used by the Arduino Mega.

The dimensions, in terms of columns and rows, were specified in the PlantCoordinates.h header file. Each variable was used for the calculation of the succeeding coordinates for only the spacing, the number of rows and columns, and starting coordinates were declared.

Figure 4.42 below is the summarized and tabulated version of the code snippets from the PinConfiguration.h and PlantCoordinates.h files as well as the settings per mode.

```
// RPI-ARDUINO COMMUNICATION PINS
extern int modePin0   =  22;          // MODE SELECTION:    00 - 5 X 2;   10 - 6 X 3
extern int modePin1   =  23;          //                    01 - 4 X 2;   11 - 4 X 3
extern int seedPin    =  38;          // FOR SEEDING
extern int waterPin   =  40;          // FOR WATERING
extern int calibPin   =  42;          // FOR CALIBRATION
```

Fig. 4.42 Declaration of Pins in Arduino IDE under PinConfiguration.h

4.3.4 Irrigation System

GrowSmart was equipped with an automated and adaptive irrigation system. Unlike the seeding system, the irrigation system and monitoring system went hand in hand since the latter was responsible for the soil moisture reading. The system was

implemented using a water pump connected to a tube that is attached along the z-axis with a sprinkler at the end. It was placed behind the cabinet together with a water container used as the water source for the irrigation system. The pump provided enough pressure to make sure that once the water reached the sprinkler, it was dispersed evenly.

4.3.4.1 Components

The pump was powered by a 12V power supply that was also connected to the relay. The nozzle had an adjustable feature that allowed the user to control the spread of water for irrigation.

Fig. 4.43 Water Nozzle

Fig. 4.44 Water Pump

Fig. 4.45 Rubber Tubing

4.3.4.2 Circuit

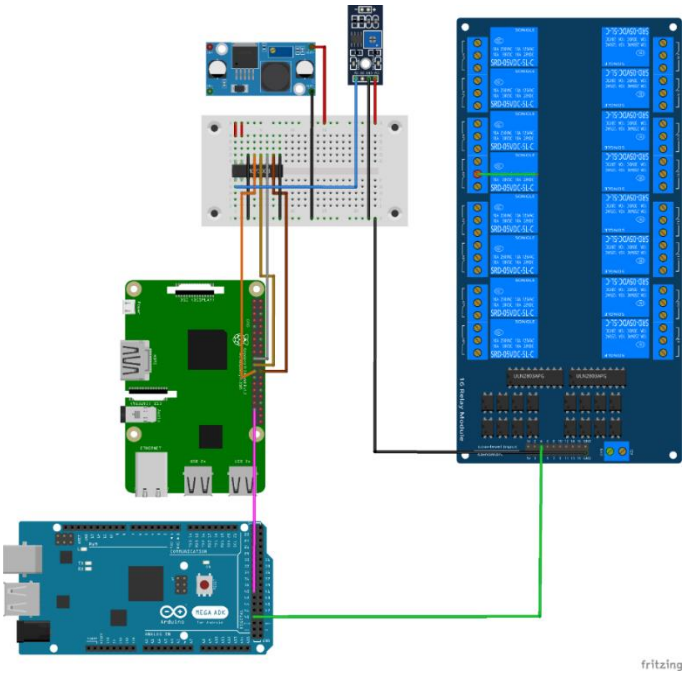


Fig. 4.46 Irrigation System Circuit

Similar to the seeding process, the irrigation system also utilized both the Raspberry Pi and Arduino Mega with the former acting as a trigger while the latter contains the actual code for the motors and water pump. The process begins with the Raspberry Pi reading the time and the soil moisture sensor data then decides whether it is time to water or not. If needed, then the Pi sends a signal to the Arduino Mega though their GPIO pins to start the irrigation system. This was represented by the pink wire connecting the two devices as seen in Figure 4.46. The signal sent by the Pi was turned off after 1 second to avoid the watering function of the Arduino Mega to repeat. After a signal was sent to the Arduino, the Raspberry Pi will record the irrigation process on the database for monitoring purposes.

4.3.4.3 Software

The researchers designed the algorithm for the irrigation system to have two functions. The first one functions as an automatic irrigation system which was the main function while the other as an adaptive watering system.

4.3.4.3.1 Raspberry Pi Automatic Irrigation Function

It was programmed in such a way that if it was around 12 noon, the Raspberry Pi reads the moisture value and checks whether it is below 50. If the condition was satisfied, the Pi then sends a signal to the Arduino Mega to trigger the irrigation system and record it in the database. This was done to ensure that the crops are watered once a day to maintain the moisture needed.

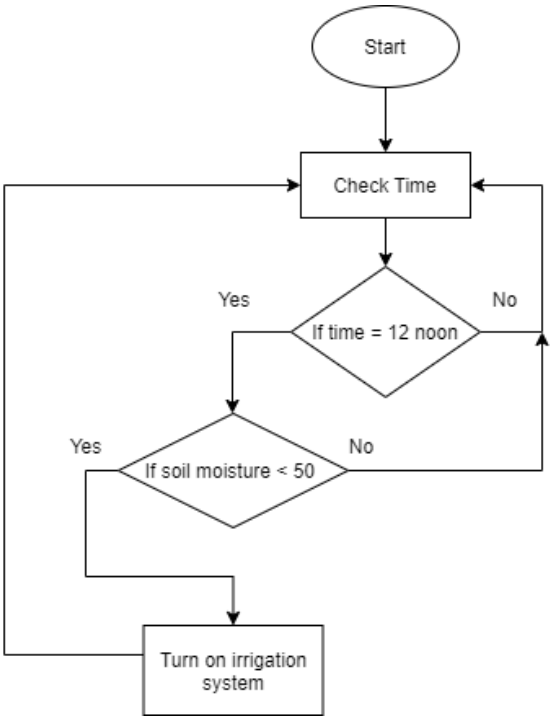


Fig. 4.47 Flowchart of AJAX Function for the Automatic Irrigation System

4.3.4.3.2 Raspberry Pi Adaptive Irrigation Function

This function was secondary to the main function of the irrigation system since it acted as a backup in case the crops need more watering apart from the scheduled 12 noon watering. Unlike the automatic watering function, this does not rely on a certain time. Instead, it was dependent on the soil moisture reading which was part of the monitoring system. If the soil moisture reading fell under the dry threshold then the Raspberry Pi triggers GrowSmart's irrigation system. However, if the reading fell under the moist or wet category, instead of triggering the irrigation system, the Raspberry Pi will display the status through the mobile application. The different thresholds for the soil moisture reading can be found on section 4.4.2.1.2.1.

4.3.4.3.3 Arduino Irrigation System Function

The Raspberry Pi only acted as a trigger while the Arduino Mega contained the actual code to run the irrigation system. For the Arduino Mega, the code for the watering process is similar to the seed distribution code wherein the main difference only was that the seed distribution code made use of the vacuum pump in its operation while the watering process made use of the watering pump. The seed distribution process commanded the robot to travel to specific coordinates for more than one time since some crops required several seeds per hole when planting. It was implemented in such a way that it first travelled along x-axis (first row) until it reached the last column similar to a snake pattern as shown in Figure 4.48 and Figure 4.49.

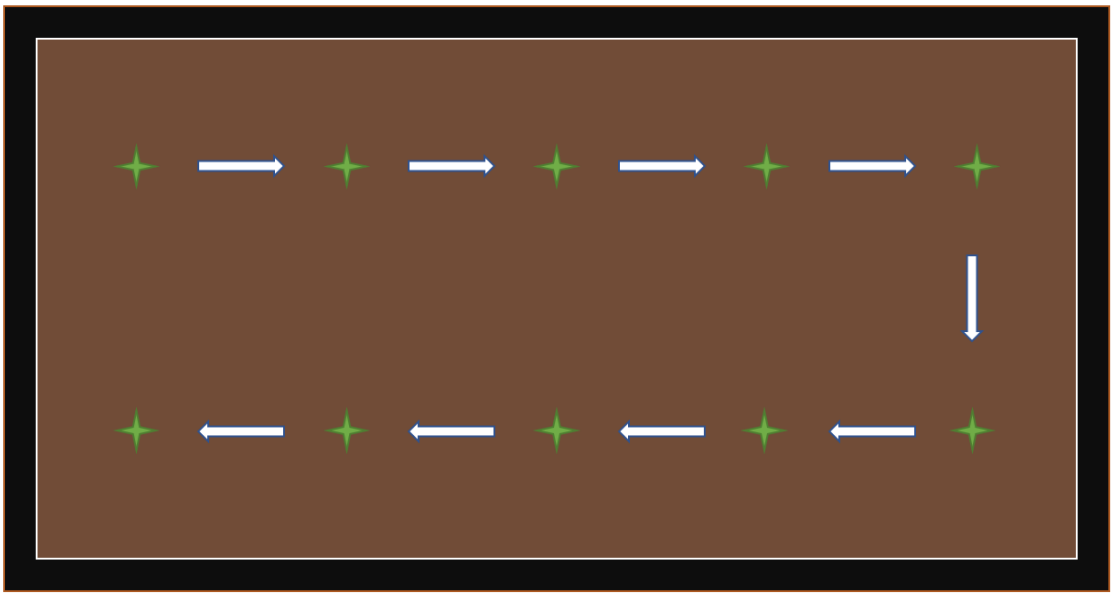


Fig. 4.48 Robot Travel Direction on 5 Columns x 2 Rows Setting

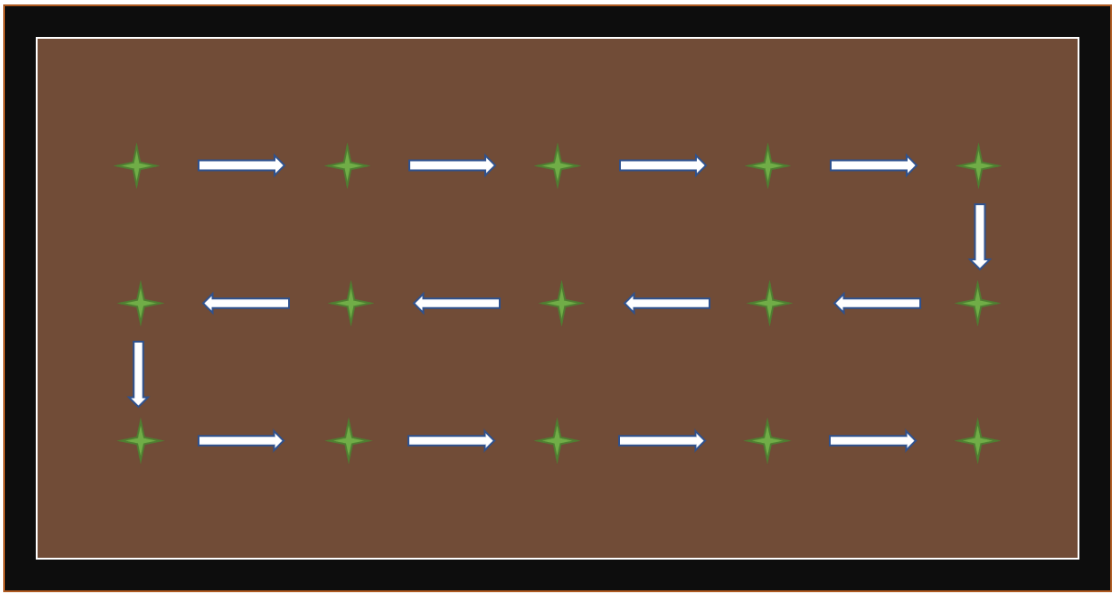


Fig. 4.49 Robot Travel Direction on 5 Columns x 3 Rows Setting

4.4 Connectivity System

4.4.1 Django Framework

The researchers developed the GrowSmart web application, both front-end and back-end, using the Django framework. It is a Python-based open-source web framework. Since Django was written in Python, it enabled the researchers to easily integrate all the sub-modules of GrowSmart such as the Raspberry Pi, the sensors, relay module, devices like the lights, pumps, fans, the XYZ robot, and the web application into one whole system. Figure 4.50 shows a high-level flowchart of the Django framework.

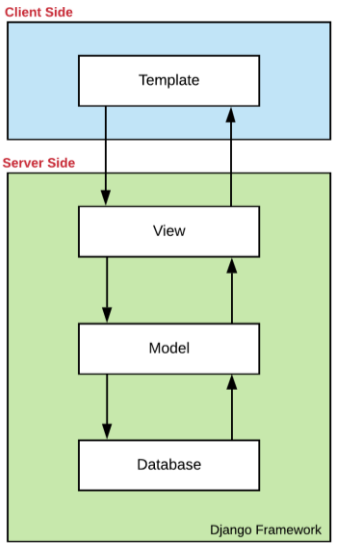


Fig. 4.50 Django Framework Flowchart

4.4.1.1 MVT Pattern

4.4.1.1.1 Models

Django applications access and manage databases through models. Models define the structure of data stored such as the size and type of field, label, default value, etc.

```
from django.db import models

class mode_selected(models.Model):
    daysCounter = models.IntegerField(default=0);
    grid = models.TextField(max_length=250, default="5x2");
    rows = models.IntegerField(default=5);
    columns = models.IntegerField(default=2);
    modeNumber = models.IntegerField(default=1);
    date = models.DateTimeField(auto_now=True);

class devices(models.Model):
    calibrationStatus = models.TextField(max_length=250, default="Off");
    fansStatus = models.TextField(max_length=250, default="Off");
    lightsStatus = models.TextField(max_length=250, default="Off");
    waterStatus = models.TextField(max_length=250, default="Off");
    seedStatus = models.TextField(max_length=250, default="Off");
    date = models.DateTimeField(auto_now=True);

class sensors(models.Model):
    temperature = models.FloatField(max_length=250, default=0.0);
    humidity = models.IntegerField(default=0);
    moisture = models.IntegerField(default=0);
    temperatureStatus = models.TextField(max_length=250, default="None");
    humidityStatus = models.TextField(max_length=250, default="None");
    soilMoistureStatus = models.TextField(max_length=250, default="None");
    date = models.DateTimeField(auto_now=True);

class model(models.Model):
    grid = models.TextField(max_length=250, default="5x2");
    rows = models.IntegerField(default=5);
    columns = models.IntegerField(default=2);
    modeNumber = models.IntegerField(default=1);
    date = models.DateTimeField(auto_now=True);
```

Fig. 4.51 Snippet of the Models Class of GrowSmart's Application

Figure 4.51 shows the tables that are migrated into GrowSmart's database. The researchers were able to migrate using the command *python 3 manage.py migrate*. After migration, the tables were now present inside the MySQL database of GrowSmart. The Django application developed by the researchers was a web application that serves as the graphical user interface, monitoring system interface, and controller of GrowSmart. The Raspberry Pi 4 computer was used to run the application since the GPIO pins acted as the main controller of GrowSmart. Therefore, the commands coming from the

application goes directly to the Pi. The same goes with the sensors and other components such as the fans and LEDs that are connected to the Pi, since the application is ran by the Pi, the data acquired by the sensors and the current status of the components are easily read by the application without needing any additional software to connect. With this, the user can easily monitor the status and control the entire garden locally or remotely.

4.4.1.1.2 View

A view is a Python function that takes a Web request and returns a Web response. It retrieves data from the database via the model, rearranges it, packages it up into an HTTP response object, and sends it to the client which will then be displayed using a template. The view part of GrowSmart's application contains all the functions that control the Pi's GPIO pins. These pins are responsible for controlling the different systems GrowSmart has. These include the monitoring, seeding, and irrigation system which are also previously discussed in sections 4.2 and 4.4. Figure 4.52 is a snippet code from the view of GrowSmart's application. What happens here is that whenever the view gets an "onFan" request (when the user turns the fans on via the application), this block of code inside the view will run. GPIO pins 20 and 16 will then output a high value signaling the relays connected to the fans to turn on. After triggering the GPIO pins, an on status will be recorded in the database as well as the current status of the lights, calibration, watering, and seeding system since they all belong to the same database table.

```
if response.POST.get('action') == 'onFan':

    print(" ")
    print("~Fans Activated~")
    print(" ")

    GPIO.output(20, GPIO.HIGH)
    GPIO.output(16, GPIO.HIGH)

    devices_.fansStatus = 'On'
    devices_.lightsStatus = devices_obj_global.lightsStatus
    devices_.calibrationStatus = devices_obj_global.calibrationStatus
    devices_.waterStatus = devices_obj_global.waterStatus
    devices_.seedStatus = devices_obj_global.seedStatus
    devices_.save()
```

Fig. 4.52 Snippet of Views Code

4.4.1.1.3 Template

The template was the HTML part of a Django application. It's what the client sees and interacts with using a browser to send HTTP request and response objects to a Django server. Django uses HTTP request and HTTP response to communicate between the client and the server as it is a web framework. GrowSmart's application used two templates which were the dashboard and database pages. These templates were html pages that were called depending on the request of the user.

4.4.1.2 Database

GrowSmart's database was developed using MySQL and its tables were already pre-defined in the models class of the Django application. The tables primarily used for monitoring were the devices table, mode table, mode vision system table, mode selected table, and the sensors table. These tables were enclosed in a red box as seen in Figure 4.53.

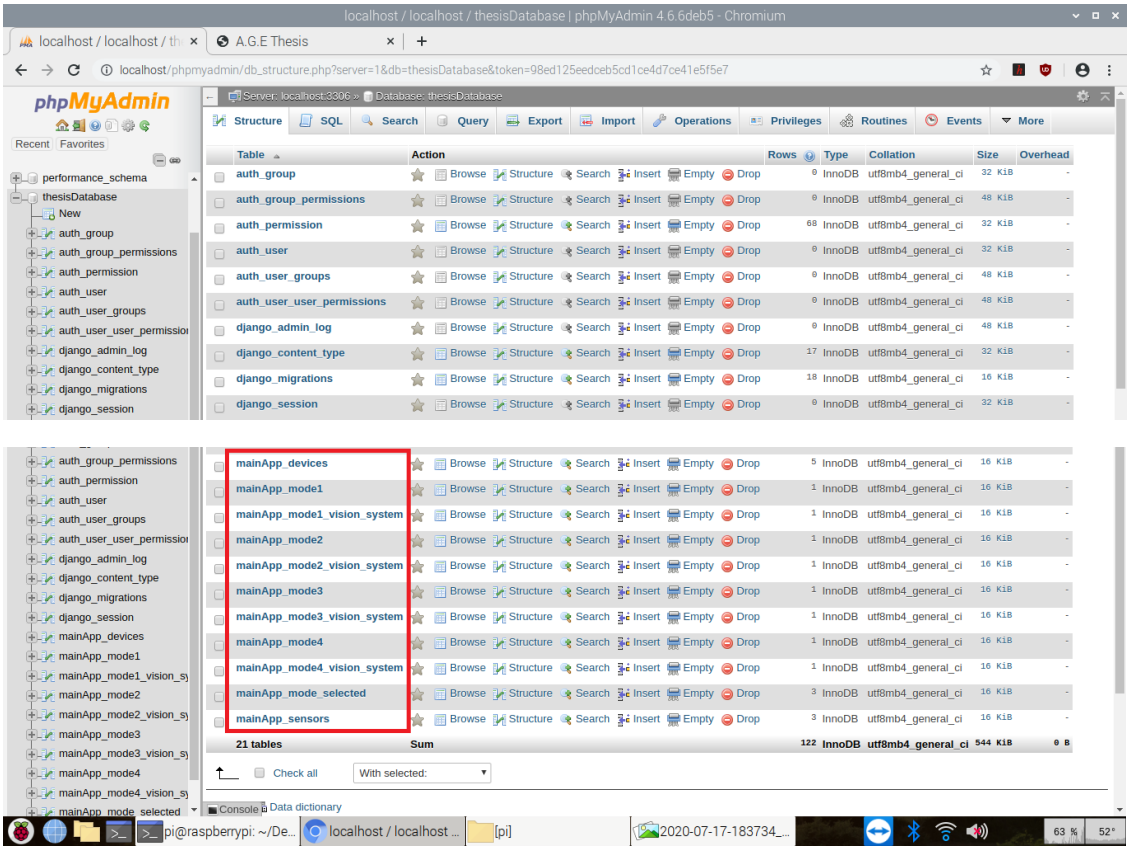


Fig. 4.53 MySQL Tables

The sensors table's columns included temperature, humidity, soil moisture values along with their interpreted status. The devices table, on the other hand, contained the current status of the robot calibration, fans, grow lights, white light, irrigation, and seeding process. The status of these devices was mostly stored as on or off. Four mode tables were also created that contain the grid which are either 2x5, 2x4, 3x6, and 2x3. The mode selected table showed the current mode the user has selected and counts the days of the system. Lastly, the mode vision system contained the pixel value of each plant received from the vision system.

id	temperature	humidity	moisture	temperatureStatus	humidityStatus	soilMoistureStatus	date
1	23.1	62	69	Good	Good	Moist	2020-01-15 00:00:00.000000

Fig. 4.54 Sensor Status Table

id	calibrationStatus	fansStatus	rgbLedStatus	whiteLedStatus	waterStatus	seedStatus	date
1	Off	On	Off	Off	Off	Off	2020-01-15 00:00:00.000000

Fig. 4.55 Device Status

id	grid	rows	columns	modeNumber	date
1	2x5	2	5	1	2020-01-15 00:00:00.000000

id	daysCounter	grid	rows	columns	modeNumber	image	date
1	1	2x5	2	5	1	2020-01-15-08_00_00	2020-01-15 08:00:00.000000

id	image	plant1	plant2	plant3	plant4	plant5	plant6	plant7	plant8	plant9	plant10	date
1	2020-01-15-08_00_00	0	0	0	0	0	0	0	0	0	0	2020-01-15 08:00:00.000000

Fig. 4.56 Plant Information Table

4.4.1.3 Asynchronous Connectivity

The jQuery library was used in this study specifically to utilize its Ajax Function to perform asynchronous requests.

```
<!-- Jquery for ajax calls -->
<script src="/assets/js/jquery-3.4.1.min.js"></script>
```

Fig. 4.57 Importing jQuery Library to Code

The Asynchronous JavaScript and XML, or AJAX was used in this study to update the application without reloading the whole page. It is composed of web development techniques on the client side to create asynchronous web applications. It enables GrowSmart's application to send requests from the client-side to the server asynchronously and vice versa making it fast and dynamic. Since the application has a dashboard and database system, data is often retrieved from the server-side to the client-

side and vice versa. With this, the user experience as well as the performance, speed, and efficiency of the application are improved.

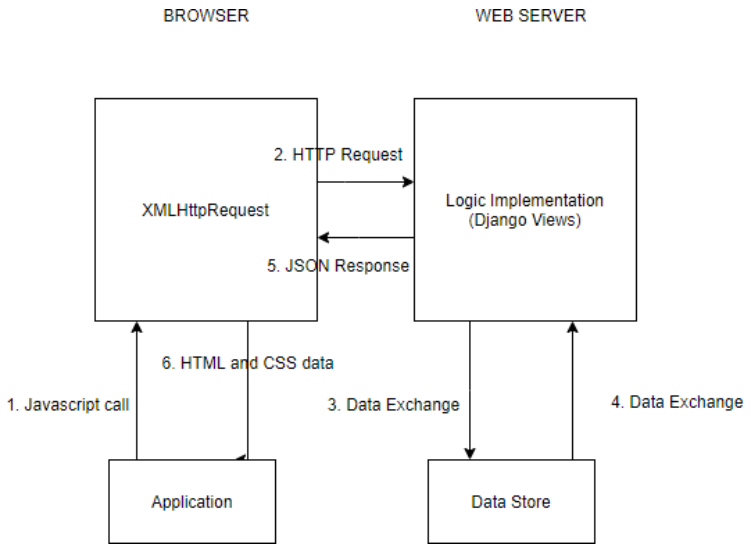


Fig. 4.58 Ajax - Django Communication flowchart

Figure 4.58 shows how AJAX is integrated into the GrowSmart application. Without it, the views part of Django will go straight to the client side (browser) and perform the necessary functions called but will not update the page unless refreshed by the user.

Whenever a user performs a request such as a click of a button or refreshing the page, the first process will be the JavaScript call to AJAX. In AJAX, it will check whether the request matches any of the AJAX requests in the code. It then proceeds to run that matching function and then matches the “action” from the AJAX function to the Django views function. The code inside the matching views function will run and then redirect back to AJAX. AJAX will then manipulate the front end of the system which will update the application without refreshing the entire system. For example, the

researchers want to turn on the lights of the system manually by clicking a button, AJAX will search for a matching request.

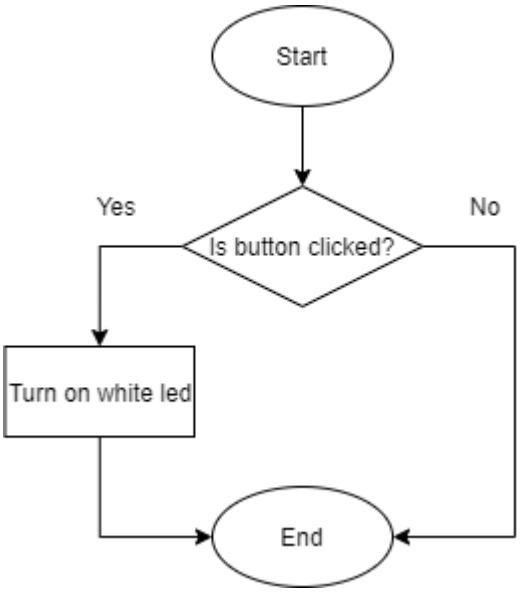


Fig. 4.59 Flowchart of AJAX function when the oneWhiteLed button is clicked

After this, AJAX will redirect the request to the views onWhiteLed function.

```
if response.POST.get('action') == 'onWhiteLed':

    print(" ")
    print("~White LED Activated~")
    print(" ")

    GPIO.output(21, GPIO.HIGH)

    devices_.fansStatus = devices_obj_global.fansStatus
    devices_.whiteLedStatus = 'On'
    devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
    devices_.calibrationStatus = devices_obj_global.calibrationStatus
    devices_.waterStatus = devices_obj_global.waterStatus
    devices_.seedStatus = devices_obj_global.seedStatus
    devices_.save()
```

Fig. 4.60 Views function when the oneWhiteLed button is clicked

Here, the GPIO pin connected to the actual lights will turn on and database insertion will occur that indicates the lights were turned on at that specific time.

Fig. 4.61 Dashboard lights status

Finally, the lights status changed from off to on without refreshing the page.

4.4.2 Mobile Application

Given that Django is a web framework, its front-end consists of a web application to send and receive requests from the user back to the server and vice versa. With this, the researchers developed a mobile-friendly web application to run smoothly using GrowSmart's mobile application. This enabled users to monitor GrowSmart using a variety of devices without having compatibility issues. The researchers utilized Bootstrap for easier application development. Bootstrap is a CSS framework used in web development to help programmers design websites easily using ready-made templates.

Fig. 4.62 GrowSmart Mobile Application Icons

4.4.2.1 Dashboard Page

This page allowed the user monitor and control GrowSmart locally and remotely. Bootstrap's grid system was used for the design of this page since a lot of data needs to be displayed in such a small screen so it has to be clean looking and organized.

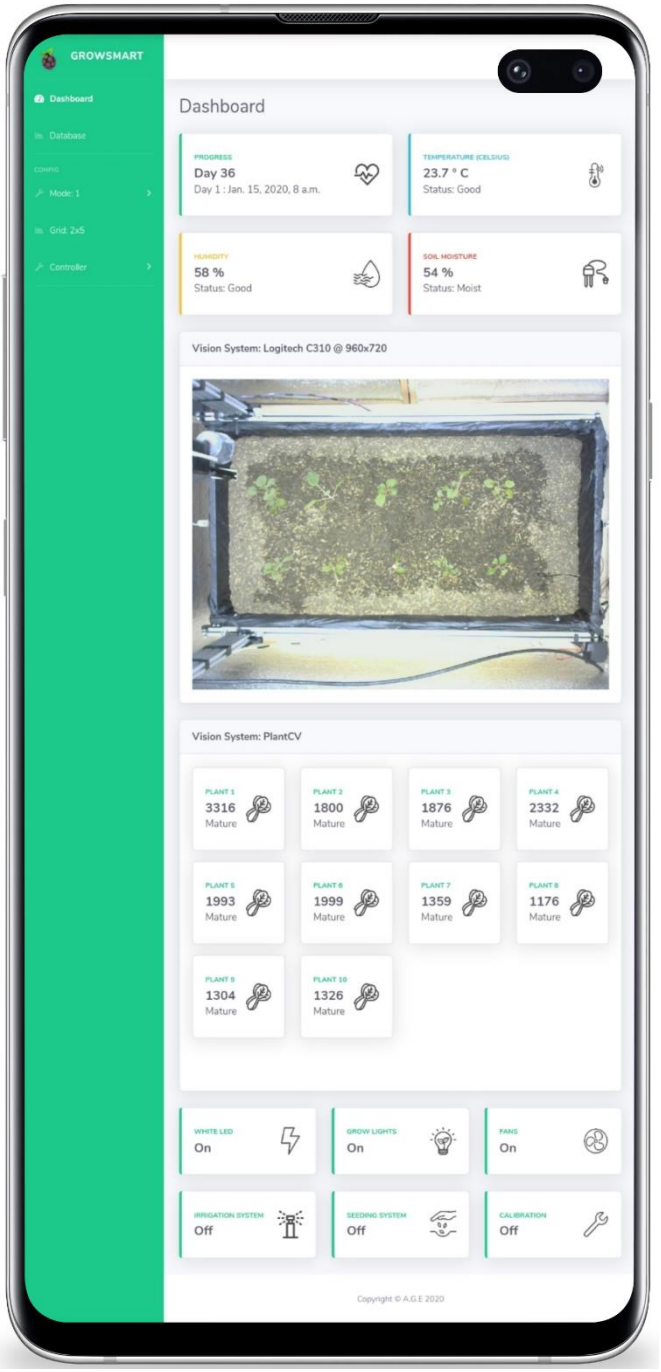


Fig. 4.63 GrowSmart Mobile Application Dashboard Page

For the first part of the dashboard, the progress, temperature, humidity, and soil moisture readings are displayed. These data are displayed on top so that the user could easily check the status of the garden.

The second part consists of the vision system. It is divided into 2 containers since the image has to be large enough for the user to see and the growth data contains a lot of information, so it also needs a big container. This design was done by creating 2 rows using bootstrap's grid system.

The last part contains information about the different devices and sub systems connected to the Raspberry Pi and Arduino. This section is placed at the bottom since among the other information displayed on the dashboard, these are the least important ones since GrowSmart is automated.

4.4.2.1.1 Side Navigation Bar

The dashboard contains a side navigation bar that consists of links such as the database page, the mode selector, and the GrowSmart controller.

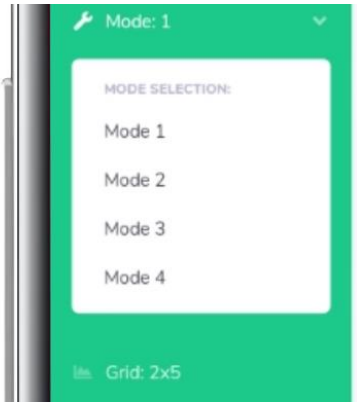


Fig. 4.64 Dashboard Page Mode Selector

4.4.2.1.1.1 Mode Selector

The mode selector allows the user to choose GrowSmart modes by clicking the mode number depending on the desired crop. The researchers decided to create a drop-down button for this part so that it would not take up a lot of space.

4.4.2.1.1.1.1 Mode 1

When clicked, the mode and grid displayed will be changed to mode:1 and grid: 2x5. The PlantCV monitor will also change to 10 plants since it is the designated number of plants for the said mode.

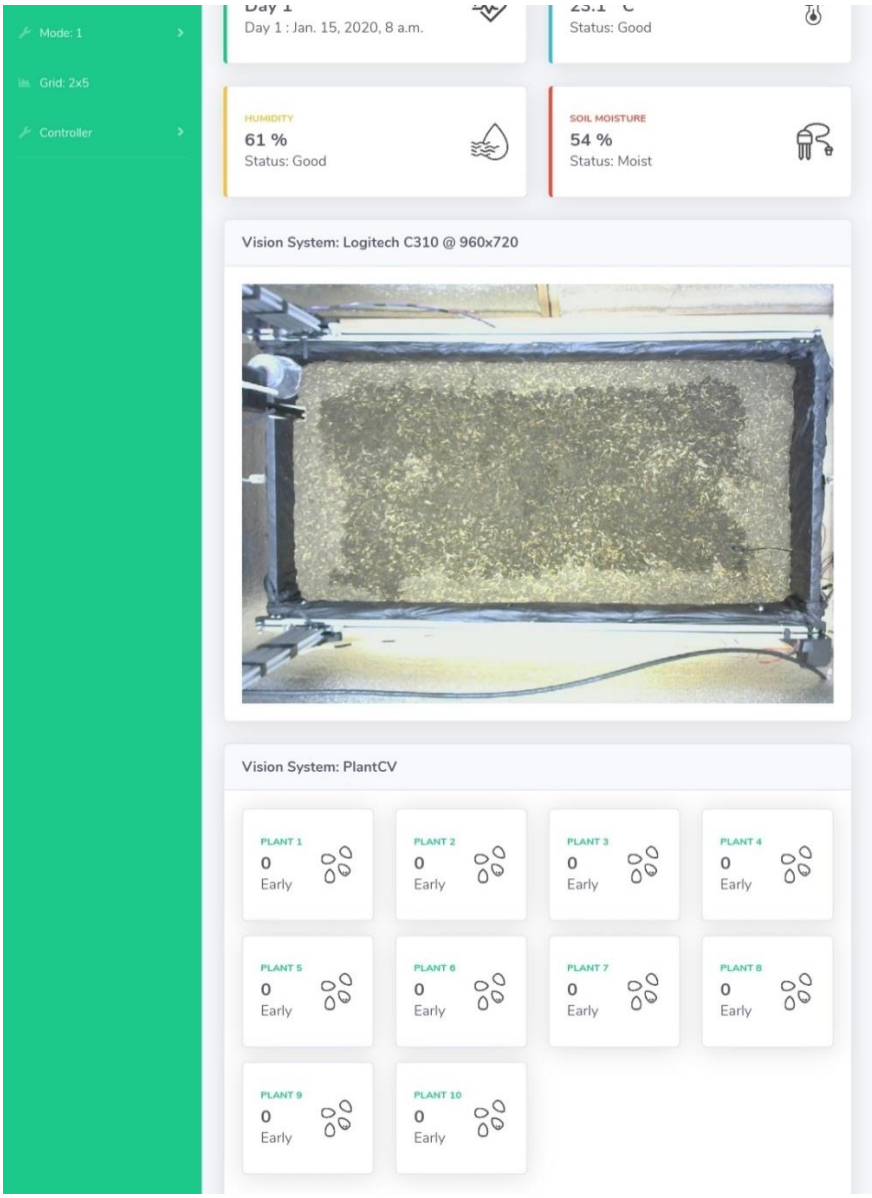


Fig. 4.65 GrowSmart Mode 1 Activated

4.4.2.1.1.1.2 Mode 2

When clicked, the mode and grid displayed will be changed to mode: 2 and grid: 2x4. The PlantCV monitor will also change to 8 plants since it is the designated number of plants for the said mode.

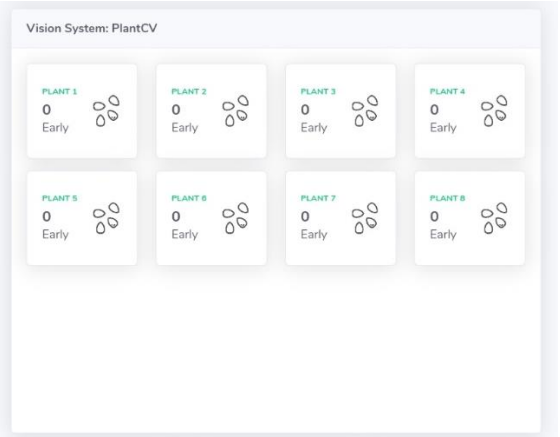


Fig. 4.66 GrowSmart Mode 2 Activated

4.4.2.1.1.1.3 Mode 3

When clicked, the mode and grid displayed will be changed to mode: 3 and grid: 3x6. The PlantCV monitor will also change to 18 plants since it is the designated number of plants for the said mode.

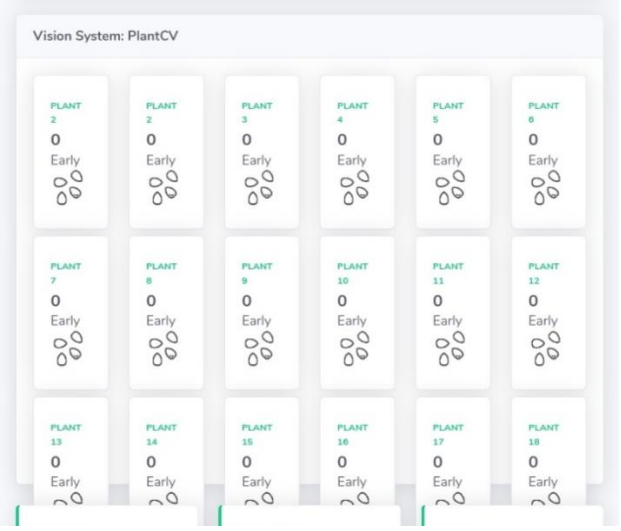


Fig. 4.67 GrowSmart Mode 3 Activated

4.4.2.1.1.1.4 Mode 4

When clicked, the mode and grid displayed will be changed to mode: 4 and grid: 3x4. The PlantCV monitor will also change to 12 plants since it is the designated number of plants for the said mode.

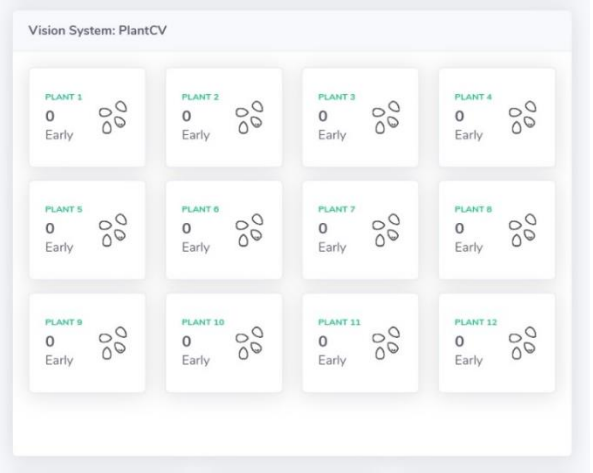


Fig. 4.68 GrowSmart Mode 4 Activated

4.4.2.1.1.2 GrowSmart Controller

This enabled the user to manually override GrowSmart's automated functions using the buttons. This was added by the researchers as a fail safe in case some automated functions do not work. However, the controller was mostly used for testing purposes. This part also utilized the drop down design to minimize the use of screen space.

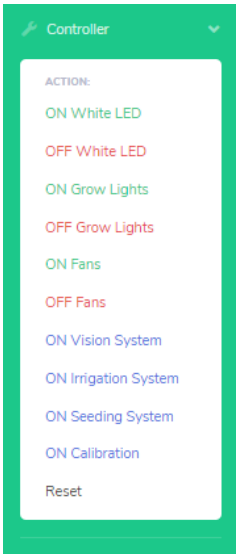


Fig. 4.69 GrowSmart Controller Section

The commands available are as follows:

1. Turn the white led on or off
2. Turn the grow lights on or off
3. Turn the fans on or off
4. Trigger the vision system
5. Trigger the irrigation system
6. Trigger the seeding system
7. Trigger the calibration function
8. Reset GrowSmart

   This command will wipe all data and will reset GrowSmart to its default settings such as: setting the mode to mode 1, deactivate all systems, and resetting the sensor readings to 0.

4.4.2.1.2 Monitoring System

Apart from the sensor readings and vision system, the age of the garden is also displayed on the application. Figure 4.71 shows the number of days GrowSmart is active. The number of days depends on the first and latest database update. A simple date subtraction Python function was ran to check this. Figure 4.70 was ran together with the sensor values checker function which means that the number of days are checked every hour.

```
date1 = mode_selected_obj_first.date
date2 = sensors_obj.date

def numOfDays(date1, date2):
    return (date2-date1).days
```

Fig. 4.70 Days Checker Function Snippet

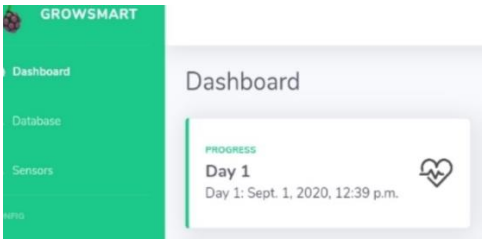


Fig. 4.71 GrowSmart Application Days Checker Section

4.4.2.1.2.1 Sensors

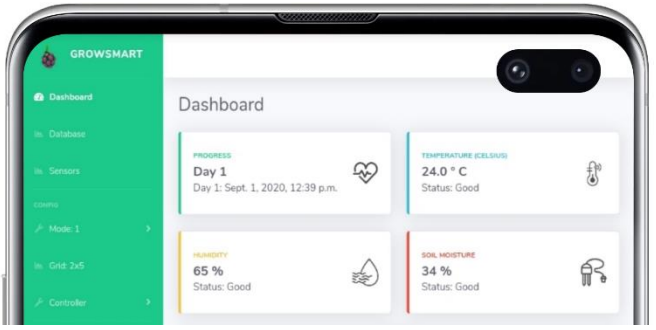


Fig. 4.72 GrowSmart Application Sensor Readings Section

The sensor readings were displayed real-time and were updated every hour automatically.

4.4.2.1.2.2 Vision System

The vision system part of the web application has two sections. The left part contained the actual picture of the garden and the right part contained the area per crop. Figure 4.73 was set to mode 1 so the number of plants is 10. The area per crop was acquired through image processing using PlantCV. The two sections were updated every day at exactly 8 AM.

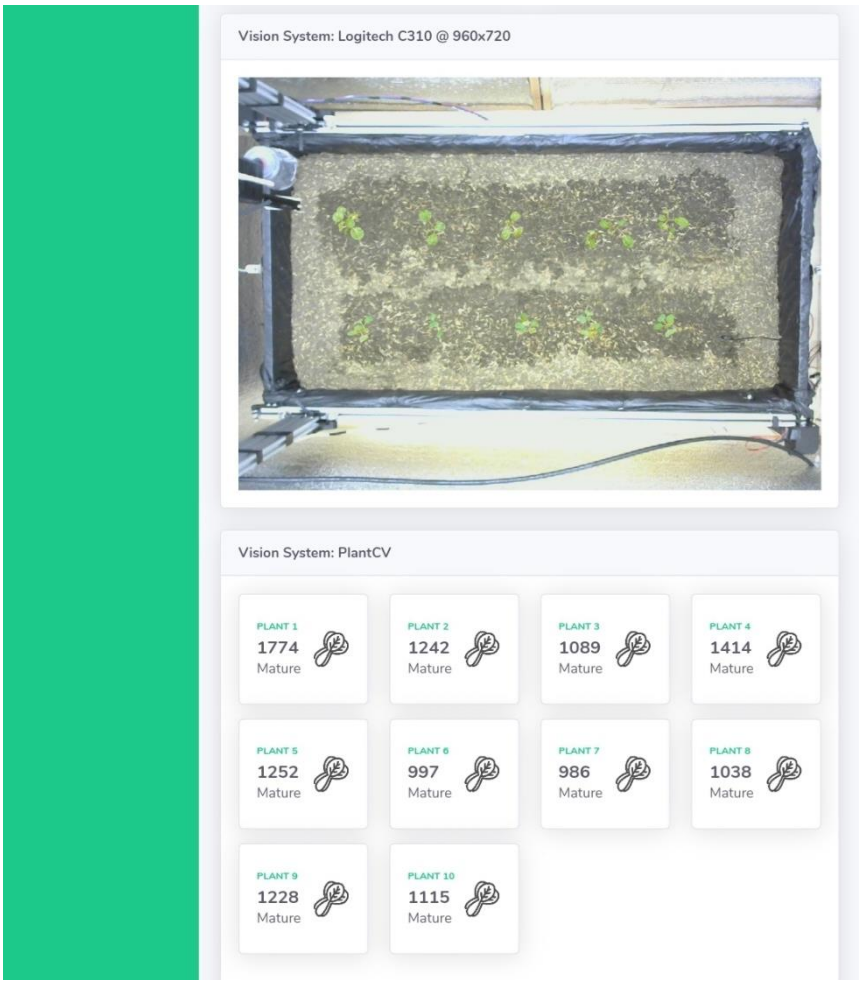


Fig. 4.73 GrowSmart Application Vision System Section

4.4.2.2 Database Page

This page displays the different tables of the database. Not all the tables are displayed since the web application should only contain what the user needs to maintain and monitor the garden. The tables displayed were the following: devices table and sensors table.

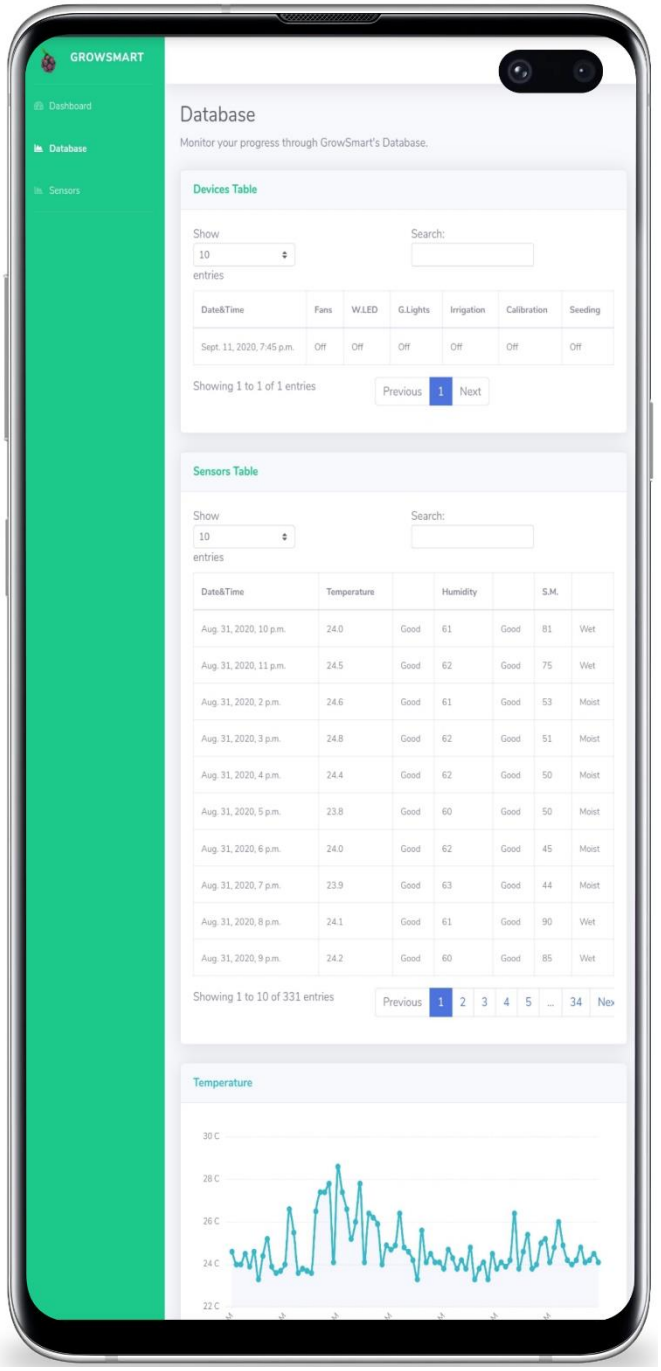


Fig. 4.74 GrowSmart Application Database Page

1. **Devices table**. This table contained the status of the different GrowSmart subcomponents such as the fans, White LED, grow lights, irrigation system, seeding system, and the calibration function as well as the dates when the devices were triggered.
2. **Sensors table**. This table contained the sensor readings from the DHT22 and soil moisture sensor as well as the date when the readings were acquired.

Aside from the tables, the charts version of the database is also displayed on this page. The purpose of this is to give the user a simplified monitoring experience compared to displaying data in table format. The Y-axis can display up to 14 days or 2 weeks' worth of data. The following charts are displayed in the database page: Temperature, Humidity, and Soil Moisture sensor readings as well as the growth of each plant.

For the sensor charts, since GrowSmart checks the sensor readings every hour, instead of displaying 24 readings per day (since 1 day has 24 hours), the researchers decided to divide the 24 readings. Instead of showing the values with 1 hour interval each, the interval was changed into 4 hours. Starting from 6AM, 10AM, 2PM, 6PM, 10PM and 2AM. This was done to optimize the use of charts and to avoid having an overcrowded look. Also, since mobile phones are compact, the labels on the Y-axis area were also simplified. To be able to see the exact value on a specific time and date, the user can click that specific dot on the chart to display the information.

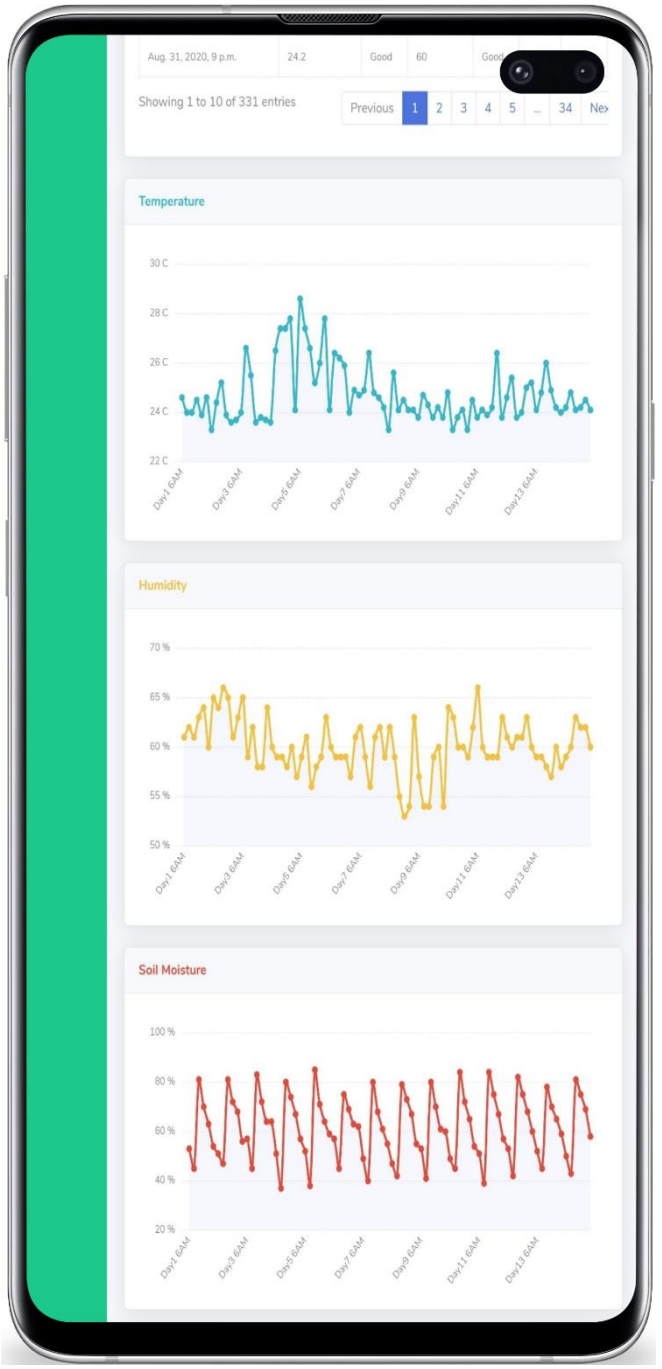


Fig. 4.75 GrowSmart Application Sensor Charts Section

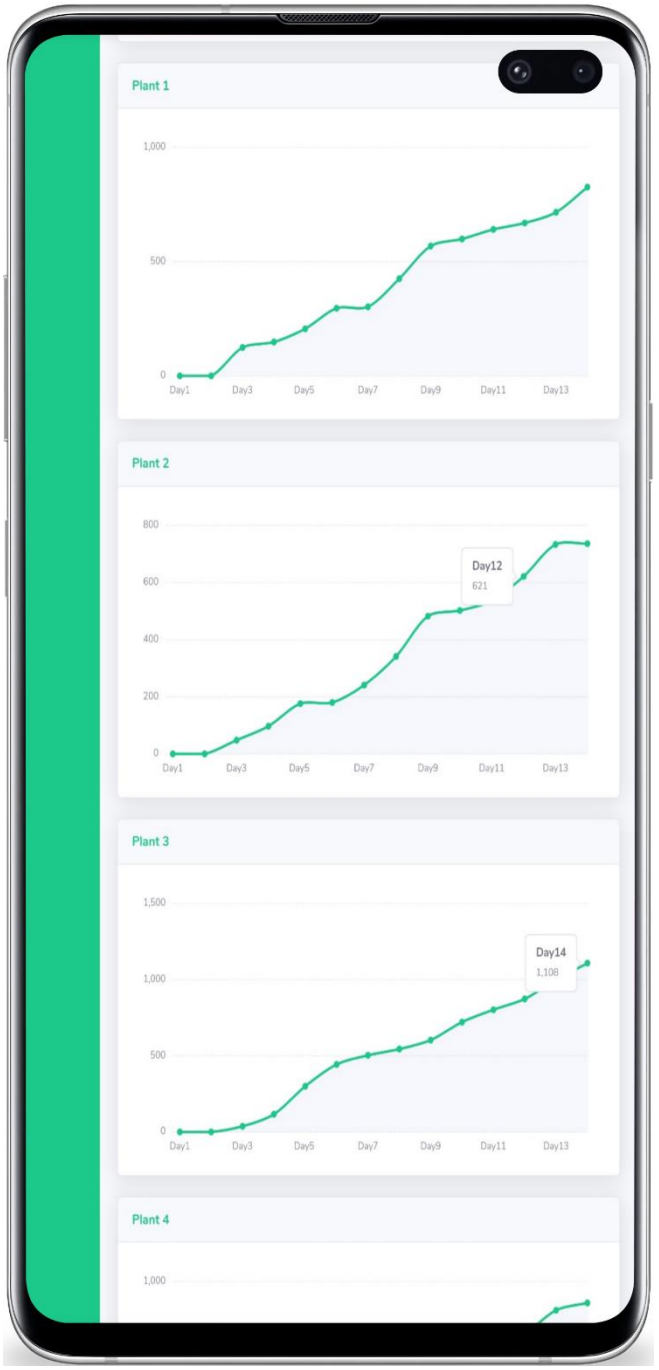


Fig. 4.76 GrowSmart Application Plant Growth Charts Section

Figures 4.77 – 4.89 below are individual snippets of the charts included in the database page.

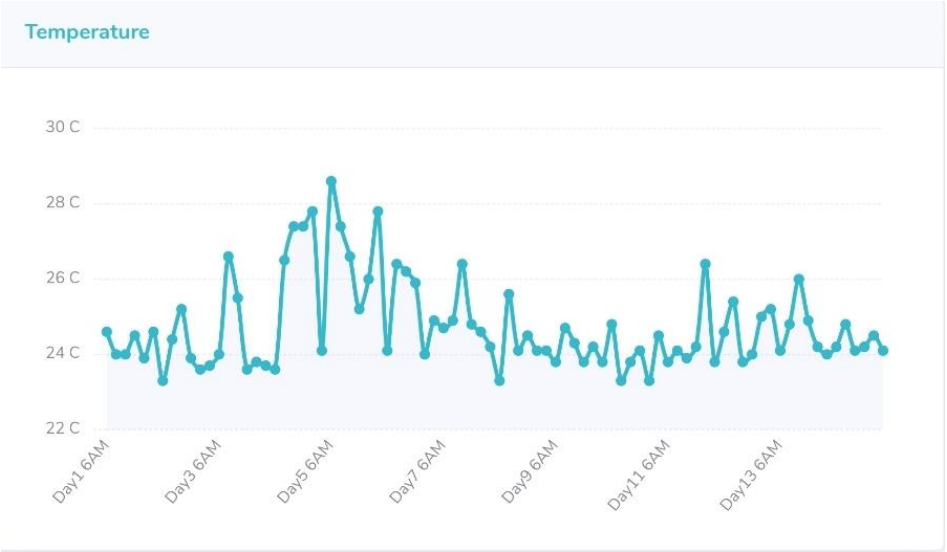


Fig. 4.77 Temperature Chart (Mobile Application)

Fig. 4.78 Temperature Chart (Mobile Application)

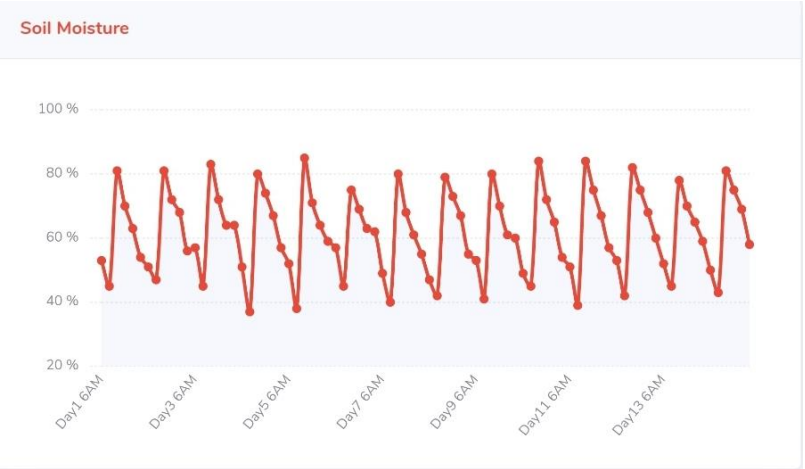


Fig. 4.79 Soil Moisture Chart (Mobile Application)

Fig. 4.80 Plant 1 Chart (Mobile Application)

Fig. 4.81 Plant 2 Chart (Mobile Application)

Fig. 4.82 Plant 3 Chart (Mobile Application)

Fig. 4.83 Plant 4 Chart (Mobile Application)

Fig. 4.84 Plant 5 Chart (Mobile Application)

Fig. 4.85 Plant 6 Chart (Mobile Application)

Fig. 4.86 Plant 7 Chart (Mobile Application)

Fig. 4.87 Plant 8 Chart (Mobile Application)

Fig. 4.88 Plant 9 Chart (Mobile Application)

Fig. 4.89 Plant 10 Chart (Mobile Application)

4.4.2.3 Sensors Page

This page displayed the current threshold set for the two DHT22 sensors as well as the soil moisture sensor. This also lets the user set a custom threshold for each since the default was set for pechay. With this, GrowSmart lets its users plant whatever crop they want without limiting the monitoring capabilities of the system.

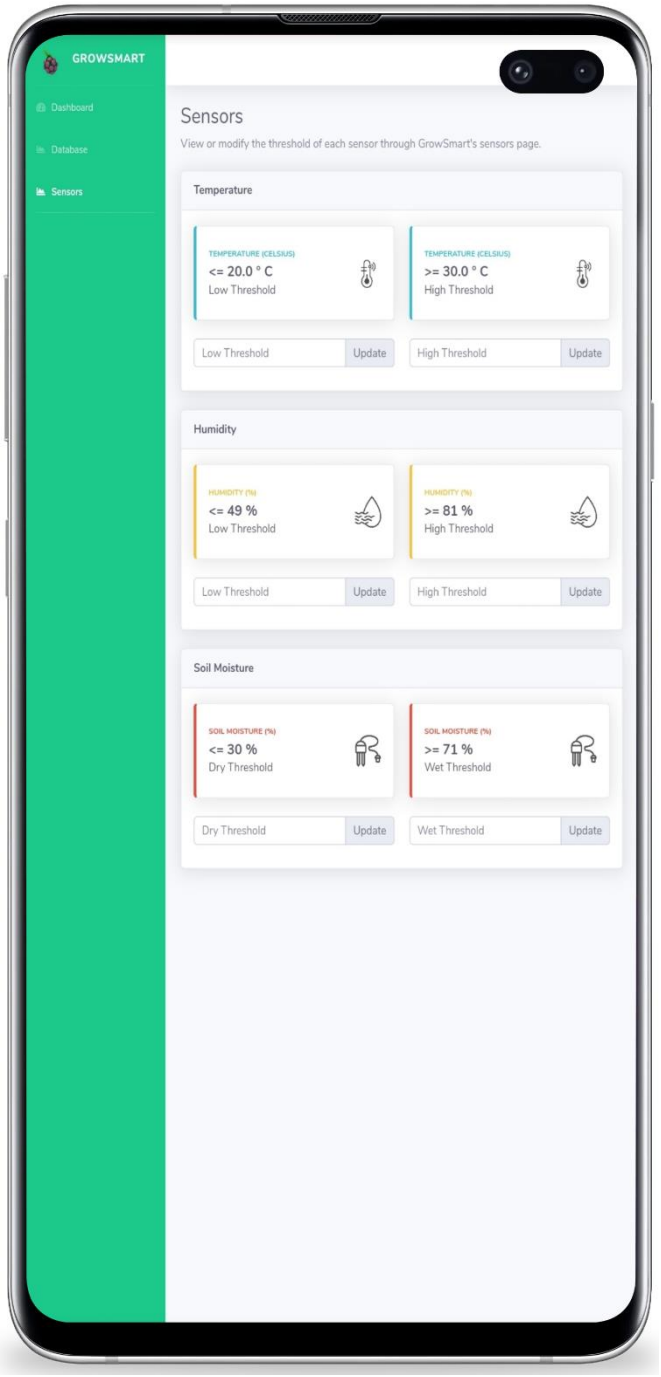


Fig. 4.90 GrowSmart Application Sensors Page

4.4.2.4 Development of Mobile Application

The mobile application was developed by the researchers to let its users run GrowSmart using their mobile devices since natively, it is a Django web application. This was developed on Android Studio using the Java language. WebView, a system component powered by Chrome that allows Android apps to display web content, was used to develop the application. This was implemented by including the <WebView> element in the main activity layout and by setting the entire MainActivity window as a WebView in onCreate().

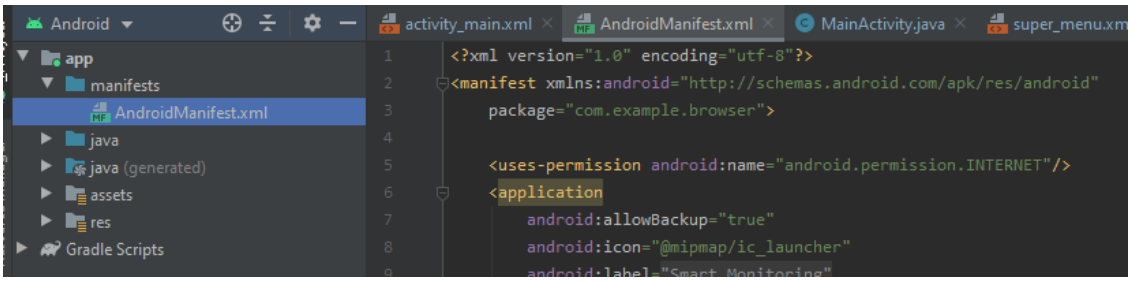


Fig. 4.91 Internet Permission

Another essential detail for the mobile application to work was to allow internet permission. This was done by inserting <uses-permission android:name="android.permission.INTERNET"/> in the AndroidManifest.xml.

```
superWebView.loadUrl(URL);
```

Fig. 4.92 Loading URL

The URL was stored in a string variable and was accessed using the .loadUrl function.

4.4.3 ZeroTier One

The researchers were able to provide connectivity between the Raspberry Pi and its users as if the whole world is one big cloud region using a network visualization platform called ZeroTier One. It is a client application that enables devices to securely connect to peer-to-peer virtual networks. It also allows networked devices, Virtual Machines, containers, and applications to communicate as if they all reside in the same network. The researchers began the set-up by creating a virtual network using ZeroTier Central, a website that lets its users create, manage, and delete virtual networks for free. After creating a virtual network, a device can now connect using the network ID provided. To do that, the user must install the ZeroTier App on his computer or phone, enter the network ID on the app, and wait for the connection status to change from offline to online. Once connected, any device inside the virtual network can now connect to another using their unique IP address. The mobile application developed by the researchers is already configured so the user does not need to enter the address of GrowSmart manually.

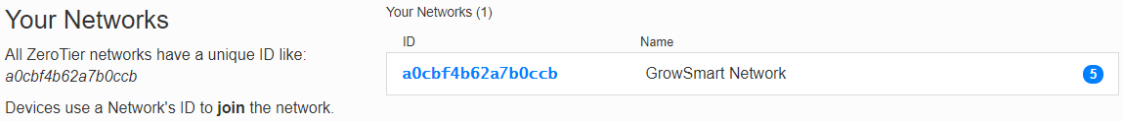


Fig. 4.93 ZeroTier Central Network ID section

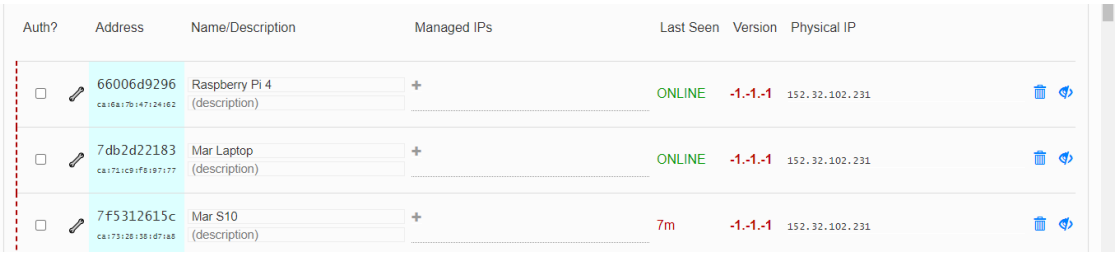


Fig. 4.94 ZeroTier Central Network Connected Devices Section

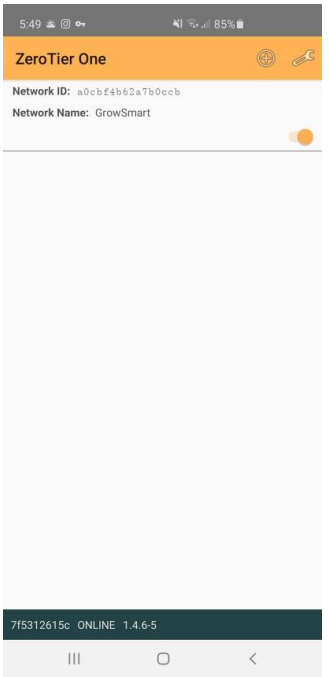


Fig. 4.95 ZeroTier App

4.5 Monitoring System

GrowSmart's monitoring system consisted of the vision system and the sensors used for monitoring the temperature, humidity, and soil moisture of GrowSmart. It was responsible for image processing and automated data acquisition. This goes hand in hand with GrowSmart's connectivity system since the mobile application serves as the graphical user interface of the monitoring system. With this, the user can connect to GrowSmart's network to monitor and control the system through the mobile application.

4.5.1 Vision System

4.5.1.1 Logitech C310 Camera

The Logitech C310 was used to capture images which will be then analyzed by the vision system's software. The camera was placed 60 inches above the ground of the

chamber considering the minimum height it needed to capture the whole plant box. A white led lamp was attached at the topmost portion since the camera does not have a night vision mode and is unable to capture clear images without proper lighting. The images captured were displayed on GrowSmart's mobile application. This allowed the user to check the current situation of the indoor gardening system.

Fig. 4.96 Camera Setup

4.5.1.2 Software

For the image capture and processing, this was done every at exactly 7:55 AM using the Logitech C310. This function also utilized AJAX since the system needs to constantly check for the time to verify if it was already 7:55 AM. The interval for checking the time is set to 60 seconds to make sure that the function won't be called twice for the time set. Before the vision system runs, the White led automatically turns on from 7:50 AM until 8AM to provide enough light to perform a clear shot.

```javascript
$(document).ready(function() {
  setInterval(automaticFunctions, 60000);
});

function automaticFunctions() {
  automaticRgbLed_on();
  automaticRgbLed_off();
  automaticWhiteLed_on();
  automaticWhiteLed_off();
  snapImage();
  automaticIrrigation();
  getSensorValues();
}
```

Fig. 4.97 JavaScript Function that calls the different automated functions every minute

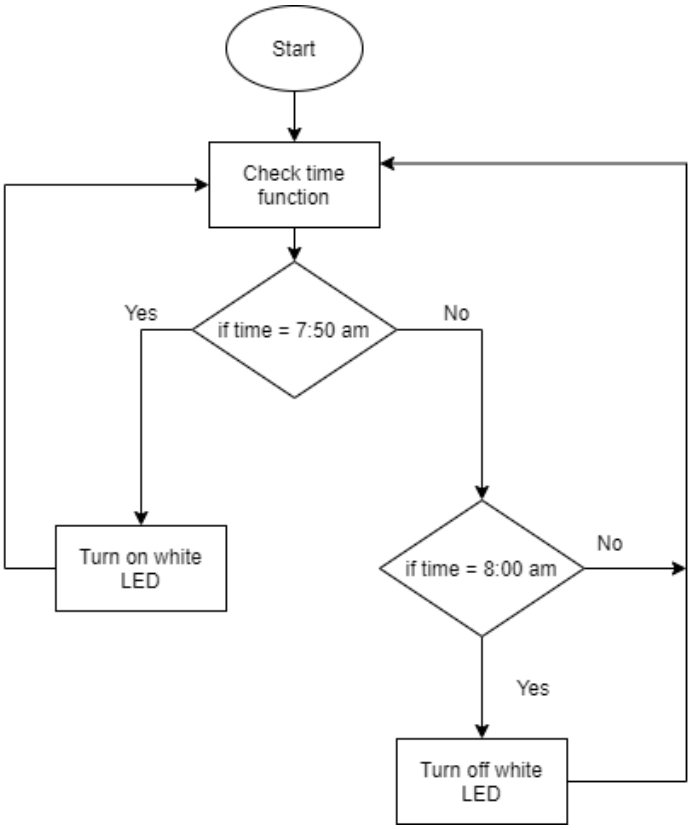


Fig. 4.98 Flowchart of AJAX function called at 7:50 AM and 8:00AM Everyday

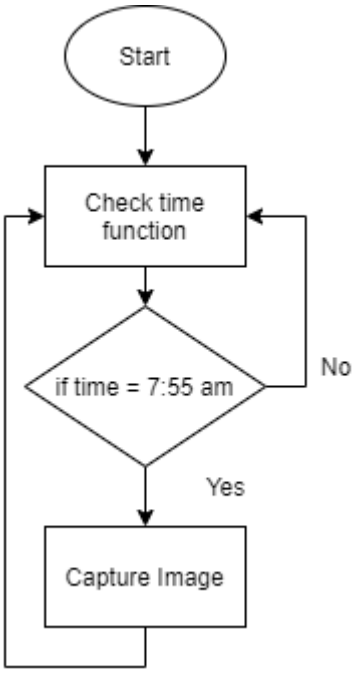
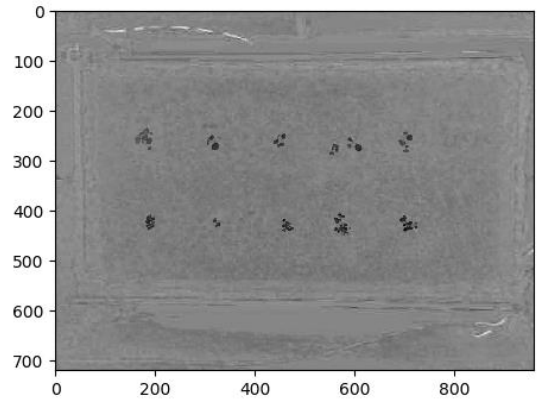
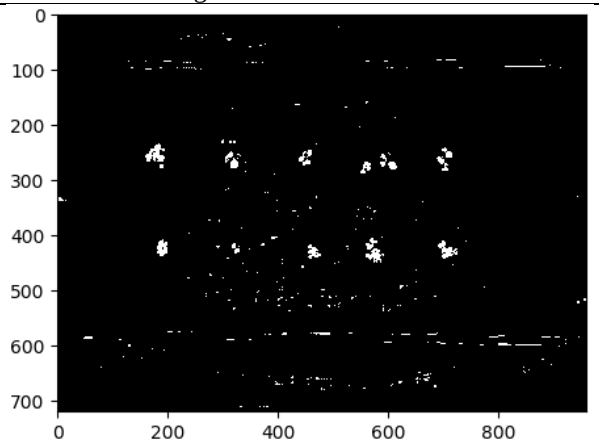


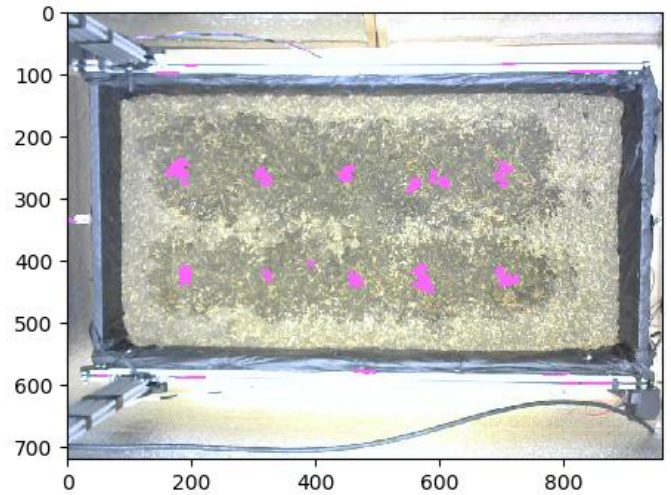
Fig. 4.99 Flowchart of AJAX function called at 7:55AM Everyday

The image processing of GrowSmart was developed using PlantCV and was based on the Multi Plant Workflow. PlantCV is an open-source image analysis software package particularly focusing on plant phenotyping. It is composed of modular functions on a variety of plant types to extract data from images.

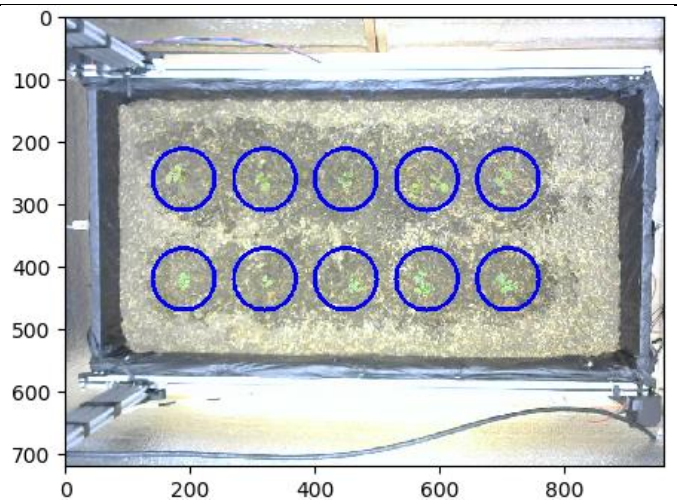
The process read by the code begins by converting the image from RGB to LAB color space and selects a single-color channel that has a contrast between the target object and background. In this case, the group has set the green magenta channel to achieve high contrast between the target objects and their background. It then uses a threshold value to convert the previous image to create a new binary image to specify which parts are light and dark.

a = pcv.rgb2gray_lab(rgb_img=img1, channel='a')	 Fig. 4.100 RGB to LAB
img_binary = pcv.threshold.binary(gray_img=a, threshold=119, max_value=255, object_type='dark')	 Fig. 4.101 Threshold

After filling up the unnecessary noises and assigning a region of interest for the whole plant box, the process then detects objects within the area.

id_objects, obj_hierarchy = pcv.find_objects(img=img1, mask=dilated)	 Fig. 4.102 Detecting Objects

To better monitor the growth of each plant, the process includes creating multiple regions of interest by assigning specific coordinates. The roi values differ for each mode since these contain a different number of plants every time. The coordinates for each mode were pre-defined as well. This was implemented by using if else statements (ex. if modeNumberSelected == 1).

roi1, roi_hier1 = pcv.roi.multi(img=img1, coord=(180,260), radius=50, spacing=(150, 200), nrows=2, ncols=5)	 Fig. 4.103 Multiple ROI

Then, it goes on a loop for each plant to extract its area and stores it in the database.

```
        for i in range(0, len(roi1)):
            roi = roi1[i]
            hierarchy = roi_hier1[i]
            # Find objects
            filtered_contours, filtered_hierarchy, filtered_mask, filtered_area = pcv.roi_objects(
                img=img, roi_type="partial", roi_contour=roi, roi_hierarchy=hierarchy,
object_contour=roi_objects,
                obj_hierarchy=roi_obj_hierarchy)
```

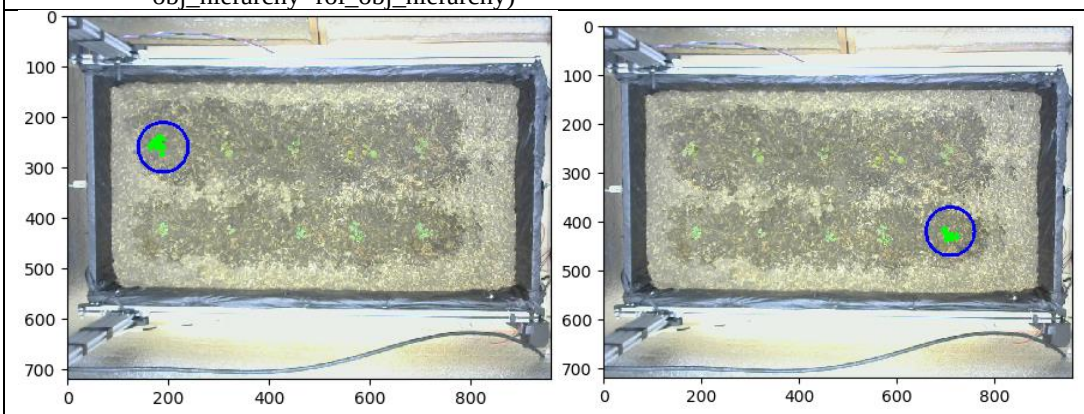


Fig. 4.104 Detecting Area per ROI

The terminal prints the value of the area in an array form.

```
[556, 194, 402, 184, 235, 346, 243, 167, 423, 498]
```

Fig. 4.105 PlantCV Results

4.5.2 Sensors

Data acquisition was done every hour to extract the temperature, humidity, and soil moisture readings. An AJAX-JavaScript function was developed to check the sensor readings every hour. Similar to the vision system, this also utilized the time checking function.

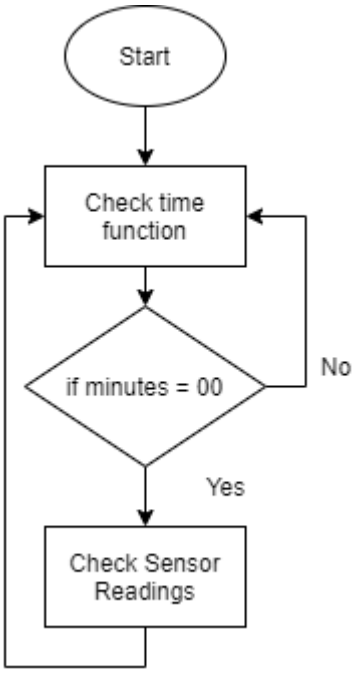


Fig. 4.106 Flowchart of Ajax Function Called Every Hour

```
function getSensorValues() {
  var date1 = new Date();
  if (date1.getMinutes() === 00) {
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {
        csrfmiddlewaretoken: $('input[name=csrfmiddlewaretoken]').val(),
        action: 'getSensorValues',
      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#daysID').text(json.daysCounter_json);
        $('#lastDayID').text('Age since: ' + json.date_json);
        $('#labelTemperature').text(json.temperature_json + " ° C");
        $('#labelHumidity').text(json.humidity_json + " %");
        $('#labelMoisture').text(json.soilMoisture_json + " %");
        $('#labelTemperatureStatus').text("Status: " + json.temperatureStatus_json);
        $('#labelHumidityStatus').text("Status: " + json.humidityStatus_json);
        $('#labelSoilMoistureStatus').text("Status: " + json.soilMoistureStatus_json);
      },
    });
  }
}
```

Fig. 4.107 Code of Ajax Function Called Every Hour

The command getMinutes got the current minute mark and checks whether it was equal to 00 which meant the code will only run every hour.

```
if response.POST.get('action') == 'getSensorValues':
    print(" ")
    print("~Sensor Values Updated~")
    print(" ")

    # Start SPI connection
    spi = spidev.SpiDev() # Created an object
    spi.open(0,0)

    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
    humidity2, temperature2 = Adafruit_DHT.read_retry(DHT_SENSOR2, DHT_PIN2)

    def analogInput(channel):
      spi.max_speed_hz = 1350000
      adc = spi.xfer2([1,(8+channel)<<4,0])
      data = ((adc[1]&3) << 8) + adc[2]
      return data

    output = analogInput(0) # Reading from CH0
    output = interp(output, [0, 1023], [100, 0])
    output = int(output)
    #print("Moistures", output)

    currentMoisture = output
    averageTemperature = (temperature + temperature2) / 2
    averageHumidity = (humidity + humidity2) / 2

    temperatureStatus = 'good'
    humidityStatus = 'good'
    soilMoistureStatus = 'good'

    temperatureStatusSummary = "Default"
    humidityStatusSummary = "Default"
    soilMoistureStatusSummary = "Default"

    if(averageTemperature > 20):
        temperatureStatus = 'low' # Too Low
    elif(averageTemperature > 30 ):
        temperatureStatus = 'high' # Too High
    else:
        temperatureStatus = 'good' # Good

    if (averageHumidity < 50):
        humidityStatus = 'low' # Too Low
    elif (averageHumidity > 80):
        humidityStatus = 'high' # Too High
```

Fig. 4.108 Views Function Snippet Called by AJAX

The views function was responsible for data acquisition and database insertion. When called, the function starts by getting the soil moisture reading from the sensor and converts the output into a percentage value. After that, it gets the readings from the two DHT22 sensors to calculate the average temperature and humidity reading. The values are then rounded up and inserted into the database.

4.5.2.1 Temperature and Humidity

Two DHT22 were used for GrowSmart. One was placed at the left side of the plant box while the other was placed at the right as seen in Figure 4.109. Two were used to collect two different data from both ends to achieve a more accurate result by getting their mean value.

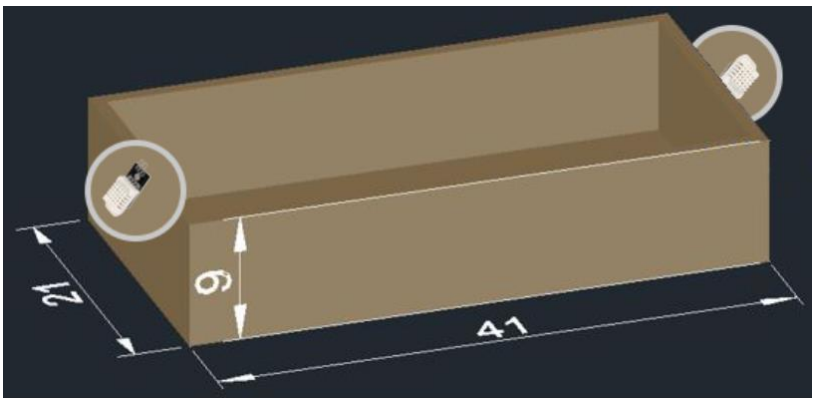


Fig. 4.109 DHT22 Placing in PlantBox

The team set the ideal temperature range between 20-30 degrees Celsius and humidity between 50%-80%.

```
if(averageTemperature < 20):
    temperatureStatus = 'low' # Too Low
elif(averageTemperature > 30 ):
    temperatureStatus = 'high' # Too High
else:
    temperatureStatus = 'good' # Good
```

Fig. 4.110 Temperature Range

4.5.2.2 Soil Moisture

The soil moisture sensor was responsible for monitoring the soil of GrowSmart. This was essential since the irrigation system was partly dependent on the soil moisture reading to perform the adaptive irrigation function. There were three possible readings namely: dry, moist, and wet. These readings were then checked by the Raspberry Pi to determine whether the garden needs watering or not.

Ranges for the output values of the probe were also pre-defined as seen in Figure 4.111.

```
if (currentMoisture <= 30):
    soilMoistureStatus = 'dry'; # Dry
elif (currentMoisture >= 31 and currentMoisture <= 70):
    soilMoistureStatus = 'moist'; # Moist
elif (currentMoisture >= 71):
    soilMoistureStatus = 'wet'; # Wet
```

Fig. 4.111 Adaptive Irrigation System Conditional Code

The MCP3008 is an 8-channel 10-bit analog to digital converter that was used for the soil moisture sensor since the Raspberry Pi cannot read analog values through its GPIO pins.

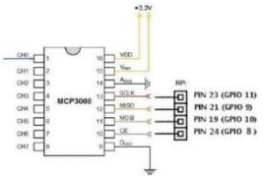


Fig. 4.112 MCP3008 Pinout

Adafruit MCP3008 Python 3 library was utilized by the Pi to be able to read the output from the MCP3008.

```python
# Start SPI connection
spi = spidev.SpiDev() # Created an object
spi.open(0,0)

humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
humidity2, temperature2 = Adafruit_DHT.read_retry(DHT_SENSOR2, DHT_PIN2)

def analogInput(channel):
  spi.max_speed_hz = 1350000
  adc = spi.xfer2([1,(8+channel)<<4,0])
  data = ((adc[1]&3) << 8) + adc[2]
  return data

output = analogInput(0) # Reading from CH0
output = interp(output, [0, 1023], [100, 0])
output = int(output)
#print("Moistures", output)

currentMoisture = output
averageTemperature = (temperature + temperature2) / 2
averageHumidity = (humidity + humidity2) / 2
```

Fig. 4.113 Python code for soil moisture sensor and MCP3008

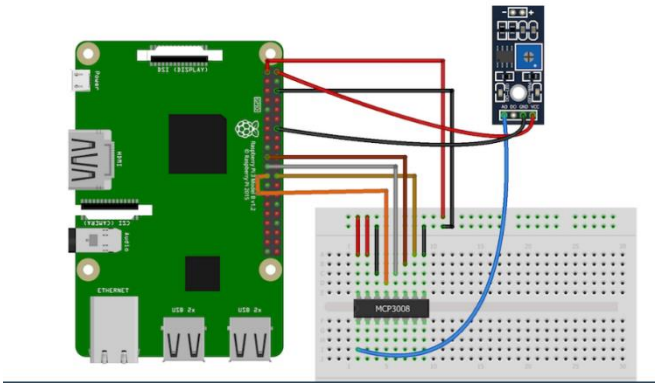


Fig. 4.114 MCP3008 Connection to Raspberry Pi

The soil moisture sensor probe was placed near the lower right of the plant box facing the door given that it is a controlled environment, the soil moisture is the same for all areas that are watered regularly.

4.5.2.3 Threshold Values

The default threshold values set for each sensor is based on the crop used by the researchers which is Pechay.

DHT22 Threshold Values

Sensor Reading (°C)	Status
0-20	Low
21-29	Good
30-100	High

Table 4.3 DHT22 Temperature Threshold Values

Sensor Reading (%)	Status
0-49	Dry
50-80	Moist
81-100	Wet

Table 4.4 DHT22 Humidity Threshold Values

Soil Moisture Sensor Threshold Values

Sensor Reading (%)	Status
0-30	Dry
31-70	Moist
71-100	Wet

Table 4.5 Soil Moisture Sensor Threshold Values

This limits GrowSmart's capability to grow other crops, so the researchers added a feature that lets its users customize the threshold values through the mobile application.

4.6 Summary

To present a functional prototype and mobile application, there are various things in both hardware and software that must be considered. With access to a 3D printer, the group was able to test each of the designs for modifications to improve the performance. The XYZ robot, which controls the seeding and watering, was assembled procedurally along with testing of its movements. The researchers made sure the components were placed correctly so that it glides softly along the tracks without having to worry about it getting derailed. The enclosure for the controlled environment also required multiple calculations such as the placing of each device to ensure that those materials are

performing with efficiency. In connection with the hardware peripherals to the software structure, A Raspberry Pi 4B and Arduino Mega together with multiple programming languages such as Python 3 and JavaSript were integrated to create a mobile application that effectively monitors the entire GrowSmart system.

Chapter 5 METHODOLOGY

Contents

5.1 Cartesian Robot System 131
5.1.1 Seeding System 131
5.1.1.1 Uniform and Variable Seed Distribution 132
5.1.2 Irrigation System 134
5.1.2.1 Automatic Irrigation Function 136
5.1.2.2 Adaptive Irrigation Function 136
5.1.3 Robot Accuracy Testing 137
5.2 Monitoring System 143
5.2.1 Vision System 145
5.2.2 Sensors 152
5.2.2.1 Trial 1 154
5.2.2.2 Trial 2 157
5.3 Cost Analysis 161
5.4 Summary 166

5.1 Cartesian Robot System

5.1.1 Seeding System

Three pechay seeds were placed on each of the ten coordinates for higher chances of growth. The seeds were distanced 15 cm horizontally and by 20 cm vertically for ideal spacing and easier monitoring.

Fig. 5.1 Seeder Reach

The seeding process began by calibrating each axis ensuring that the cartesian robot was at the edge of their respective profiles where the limit switches were attached to. This step was necessary to make sure that the movements along the specific coordinates assigned were precise. After calibration, the robot initially moved along the x-axis then the y-axis to reach the destination of the seed container. Once the desired position was reached, the pump was turned on and the z-axis moved downward and sucked the seed with the tip of the syringe. The robot then travelled to the first coordinate and once reached, the syringe forced the seed below soil surface before releasing the air pressure, hence planting the seed. After injecting the seed, the robot returned to the seed

container and does two more rounds of planting. This resulted in injecting three seedlings per coordinate before moving on to the succeeding coordinates.

This applied the concept of the injection method, thus, there was no need for human intervention to dig a hole and cover them manually. Irrigation was done immediately after planting the seeds to bond it with the soil.

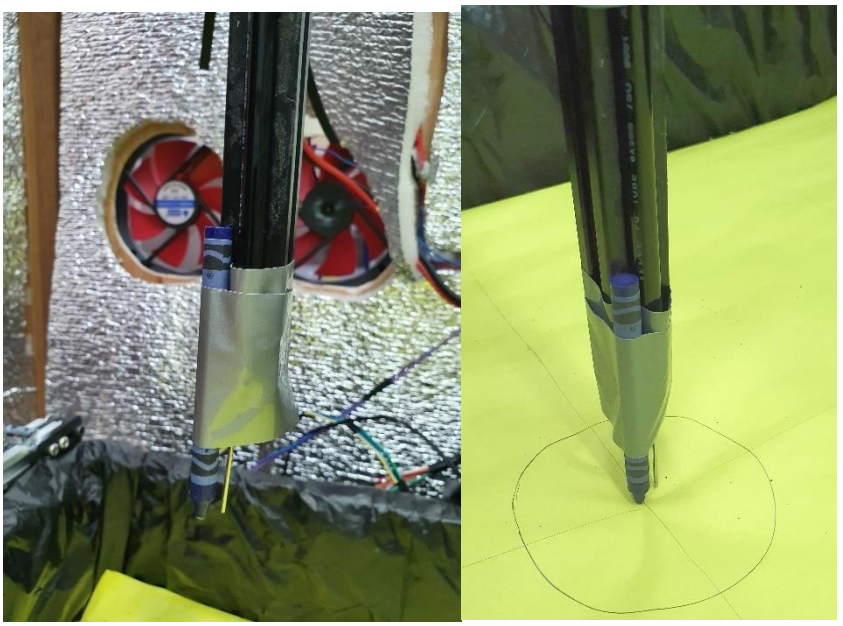


Fig. 5.2 Syringe Taped to Z-Axis with Tube

5.1.1.1 Uniform and Variable Seed Distribution

The travel distance of each axis was pre-defined and calculated as seen in Table 5.1. The cartesian robot was programmed to provide four variable seed distribution modes, each with uniform spacing horizontally and vertically depending on the mode selected. This was dictated by the signals sent from Raspberry Pi that the Arduino Mega receives from its two digital pins (modePin1 and modePin0). Declaration of variables for assigning pins was provided for a clearer understanding. Figures 5.5 to 8 on the other hand provides visualization of the four modes when used in image processing.

```
// SETTINGS FOR EACH MODE

externint xSpace    = 0;            // SPACING ALONG X
externint xStart    = 0;            // FIRST X-COORDINATE

externint ySpace    = 0;            // SPACING ALONG Y
externint ySeed     = 0;            // FIRST Y-COORDINATE FOR SEEDING
externint yWater    = 0;            // FIRST Y-COORDINATE FOR WATERING

externint noOfCols  = 0;            // NO. OF COLUMNS
externint noOfRows  = 0;            // NO. OF ROWS
```

Fig. 5.3 Declaration of Specifications in Arduino IDE under PlantCoordinates.h

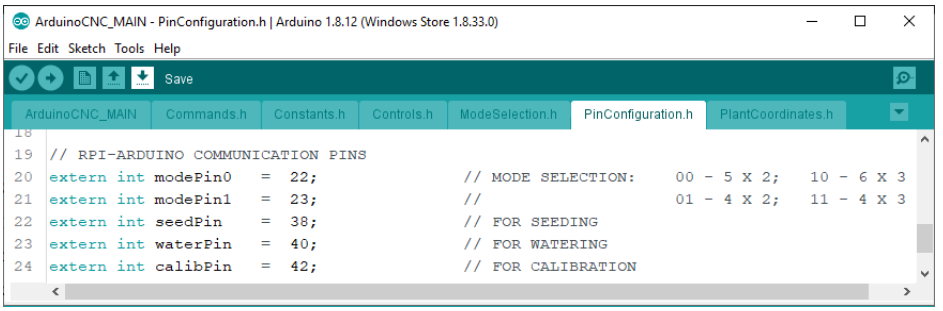


Fig. 5.4 Declaration of Mode Pins

modePin1	modePin0	Dimension (Column x Row)	X Spacing (cm)	Y Spacing (cm)	First Coordinate	
					X	Y
0	0	5 x 2	15	20	25	15
0	1	4 x 2	20	20	25	15
1	0	6 x 3	12	10	20	15
1	1	4 x 3	20	10	25	15

Table 5.1 Mode Tabulation with Dimensions

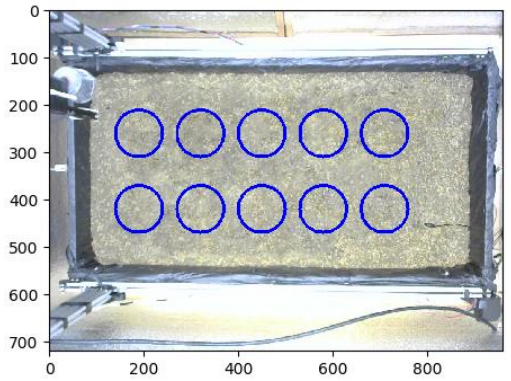


Fig. 5.5 Mode 00

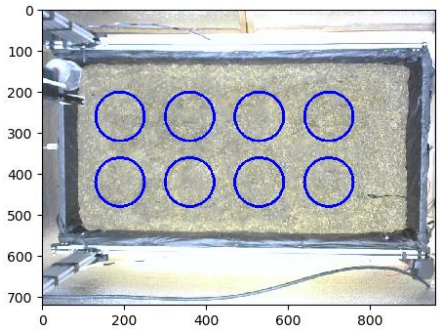


Fig. 5.6 Mode 01

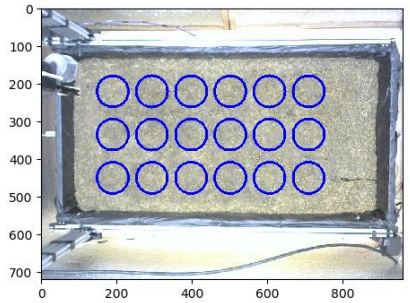


Fig. 5.7 Mode 10

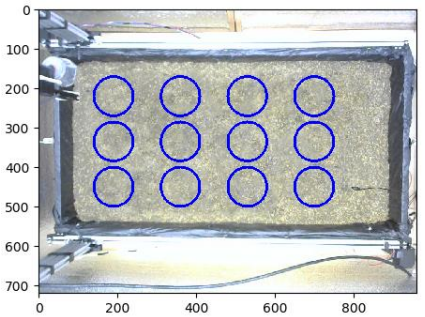


Fig. 5.8 Mode 11

5.1.2 Irrigation System

Unlike the seeding function, this one only utilized the x and y-axis. Similar with the seeding process, the irrigation process began with calibration to ensure that the cartesian robot was at the correct position for watering since its plant coordinates were the same as the seeding function. The x and y-axis moved to the first coordinate and

signaled the water pump to turn on. The dripper head then got water from the tube hose connected to the submersible water pump in the water container outside the enclosure. This allowed the nozzle to dispense water evenly for 3 seconds. Once finished, the water pump was turned off then the robot moved onto the next coordinate. The process was repeated until all coordinates were done.

Fig. 5.9 Irrigation

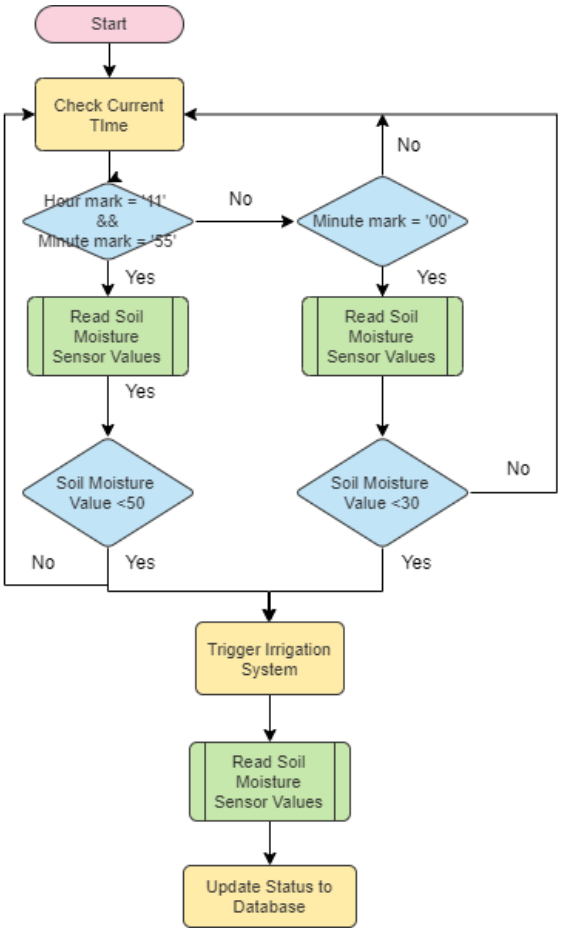


Fig. 5.10 Automatic and Adaptive Irrigation Flowchart

5.1.2.1 Automatic Irrigation Function

The time for watering was set at around 12 noon every day. But before it proceeds, GrowSmart checks whether the current soil moisture reading is below 50. If it was, then watering starts, if not, then watering does not proceed.

5.1.2.2 Adaptive Irrigation Function

The adaptive irrigation function operated together with the sensorReading function which was ran every hour at 00 minutes. Since part of the sensorReading function was to check the soil moisture reading, the adaptive irrigation function was

added here. If the soil moisture reading was under the dry threshold, then the Raspberry Pi triggers GrowSmart's irrigation system. However, if the reading was under the moist or wet category, watering does not proceed.

Sensor Reading	Soil Status	Action
0-30	Dry	On
31-70	Moist	Off
71-100	Wet	Off

Table 5.2 Adaptive Irrigation System Thresholds

5.1.3 Robot Accuracy Testing

To evaluate that both seeding and watering processes were accurate, the group had implemented a function that calibrates the cartesian robot before it performs the mentioned processes. With this method, the robot would always start on the edge beside the limit switch that signals the end of the extrusion. Before the XYZ robot performs seeding or watering, it first calibrates to guarantee that it is always in its starting position.

The calibration process can also be forced by using the controller which lets the Raspberry Pi send a signal to the Arduino Mega to initiate calibration. This was done to ensure that the robot can be calibrated in cases where there might be a power outage and the robot was not able to finish its process and had stopped somewhere along within the aluminum profile.

Fig. 5.11 Canvas for XYZ Accuracy Testing

Based on Figure 5.11, the team used a cartolina laid flat on the soil surface. The planting ground was separated into 10 circular areas, each having a radius of 4.5 cm as shown in Figure 5.13. The figure is a digitized version of the drawings on the cartolina which is plotted using Desmos. In this situation, the robot was using the default mode which sets the planting ground into 5 columns and 2 rows where it is spaced 15 cm horizontally and 20 cm vertically.

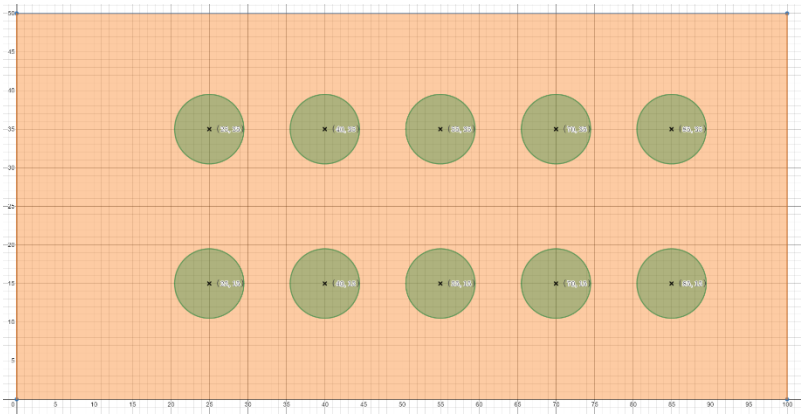


Fig. 5.12 Desmos Plotting of Canvas

The group had tested the accuracy of the seeding mechanism performed by the Z-axis by using a cartolina wherein one of its corners is set as the origin (0,0) and cut

into the size of the planting ground which is 100 cm by 50 cm. This was done to make sure that it can lay flat on the soil and can serve as a canvas for the cartesian robot to mark on using writing materials such as crayons, markers, and highlighters which were attached almost exactly at where the seed injector is located to simulate a seeding process. The motor travelled a distance of 1 cm for 50 micro steps per revolution. This ratio was determined by setting different values for the variable stps in the Arduino code and by manually measuring the start and end point of the cartesian robot. Upon determining this ratio, the stps variable was then set to 50 and was just being multiplied by the number of cm the user wants to traverse to make the code flexible.

This setting was hardcoded in the Constants.h header file so that the user can easily locate where to change the settings of the cartesian robot in the Arduino IDE. The same process was done for the Z-axis of the robot to make sure that each step was calibrated to move the lead screw or the linear drive a particular amount of distance only.

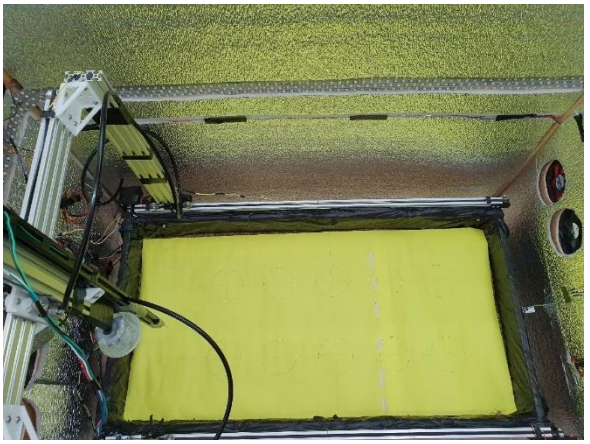


Fig. 5.13 Robot Accuracy Testing Setup 1

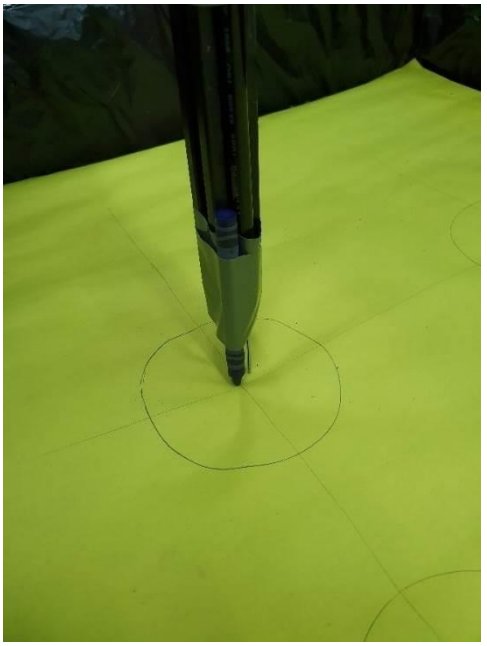


Fig. 5.14 Robot Accuracy Testing Setup 2

ArduinoCNC_MAIN § | Commands.h § | Constants.h § | Controls.h | ModeSelection.h | PinConfiguration.h § | PlantCoordinates.h

```
// STEPPER MOTOR SETTINGS

extern int delayTime = 1000;          //Delay between each pause (uS); The smaller the value, the faster the revolution.
extern long int stps = 50;            // microsteps per revolution with 1/32 microstepping    [50 steps = 1 cm travel]
```

Fig. 5.15 Constants.h header file snippet

The seeding process was done for a total of 5 times where each trial was indicated by the color of the writing material that was used. Each planting ground was marked by a number indicating the order by which the cartesian robot places a seed on that specific coordinate.

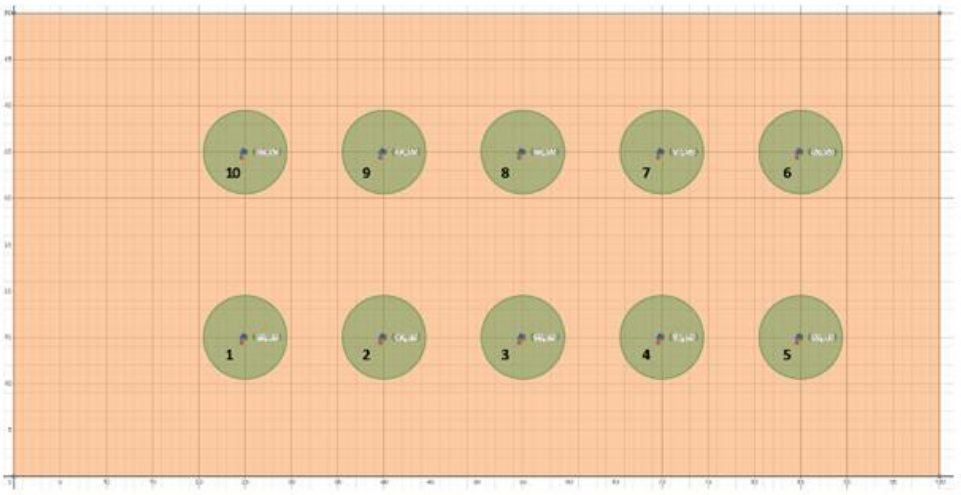


Fig. 5.16 Division and Numbering of Planting Grounds

Each time that the robot finished the execution of the process, the color was changed by replacing the writing material that provided easier distinction for each trial. A total of 5 different writing materials were used to mark the cartolina.

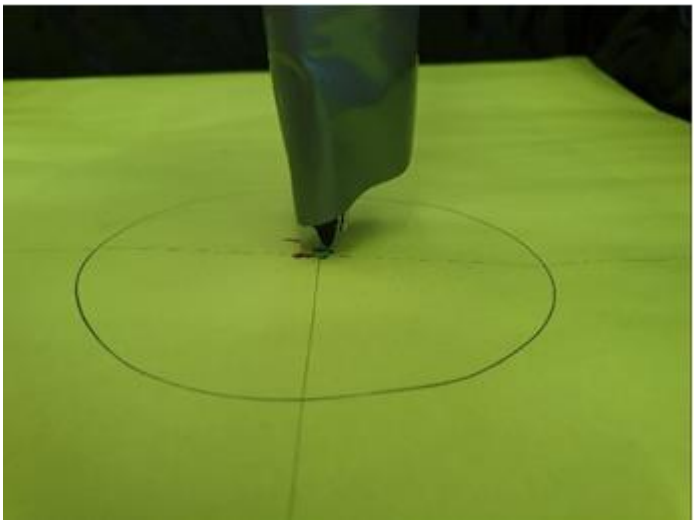


Fig. 5.17 Set-up for Writing Material Along Seed Injector

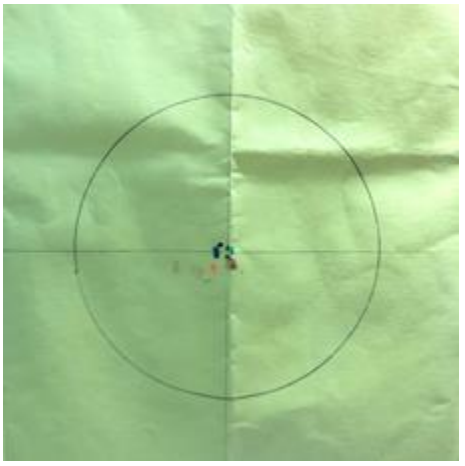


Fig. 5.18 Result of Seeding After 5 Trials Indicated by Difference of Colors

When the motors are in operation, they are put on a tight hold which meant that anyone cannot easily dislocate the robot in each axis. With this, the distances of the marks for each trial was the same for each area where it is only distanced at a particular value horizontally and vertically from the expected coordinates where each seed is to be planted.

To calculate the distance between the center of the circle and the mark for each trial, the distance formula was used:

$$distance = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

where: x_a = x-coordinate of circle's center

y_a = y-coordinate of circle's center

x_b = x-coordinate of trial mark

y_b = y-coordinate of trial mark

Fig. 5.19 Distance Formula

The distance calculated using the formula above is also known as the radial distance.

Once the distance was calculated or measured, the mean was obtained by getting the total of the individual radial distance from each trial and dividing it by the total number of trials.

$$\mu = \frac{total\ radial\ distance\ from\ each\ trial}{no.of\ trials}$$

where: μ = mean / average radial distance

Fig. 5.20 Mean Formula

To further describe the precision of the robot and how the measurements were spread out from the average or expected value, the standard deviation was calculated. The standard deviation was obtained by using the formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

where: σ = standard deviation

N = number of trials

x_i = value of the ith radial distance

μ = average radial distance

$x_i - \mu$ = deviation

$(x_i - \mu)^2$ = variance

Fig. 5.21 Standard Deviation Formula

5.2 Monitoring System

GrowSmart's monitoring system consisted of the vision system and the sensors. The researchers developed a mobile application that acts as the graphical user interface of GrowSmart so that the users can keep track of their garden's status anywhere easily. It contained a dashboard which displayed the status of the sensors, plant growth, status of the different sub systems such as the irrigation, seeding, lighting, air circulation, and calibration.

Shown along with the sensor values were its interpreted conditions. The process began by programming the pi to take pictures once a day and reading sensor values every hour. Once the plant growth and sensor values were obtained, these were immediately stored in the database. The stored data were instantly reflected in the mobile application as well.

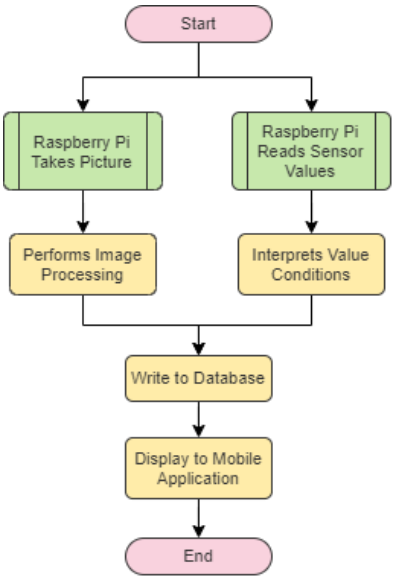


Fig. 5.22 Monitoring System Flowchart

5.2.1 Vision System

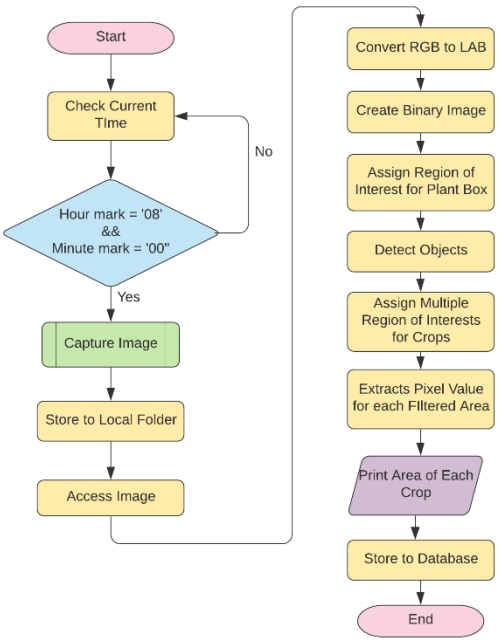


Fig. 5.23 Image Capturing and Processing Flowchart

Figure 5.23 shows a quick summary of how the areas of each plant are obtained. To confirm whether the vision system does monitor the growth of the plant, images every five days were used. As observed from Figures 5.24-31, the shaded green area portrayed the detected objects that gradually increased over time.

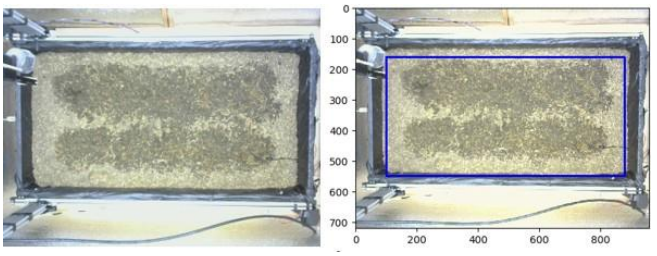


Fig. 5.24 Day 1

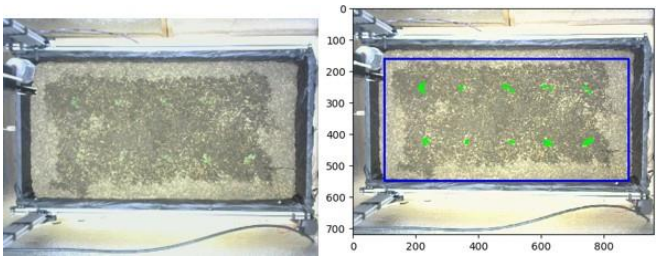


Fig. 5.25 Day 6

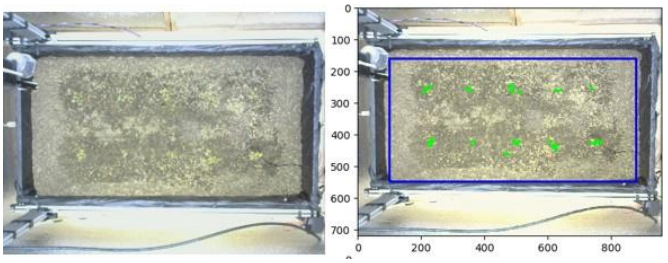


Fig. 5.26 Day 11

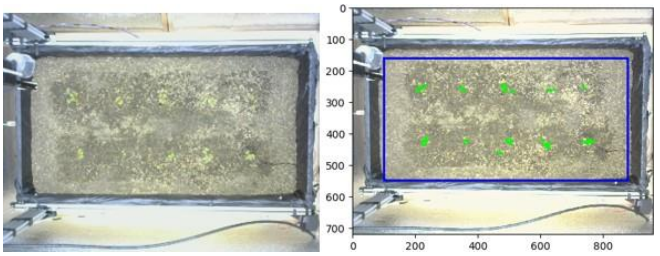


Fig. 5.27 Day 16

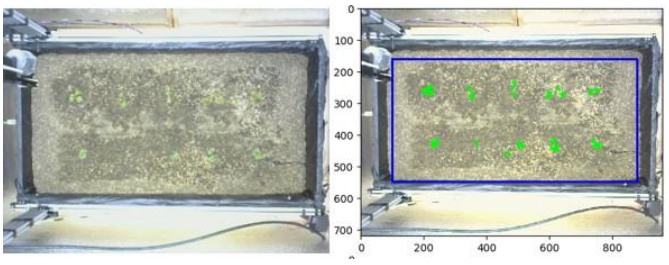


Fig. 5.28 Day 21

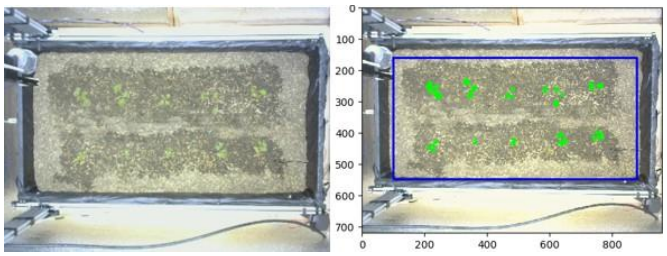


Fig. 5.29 Day 26

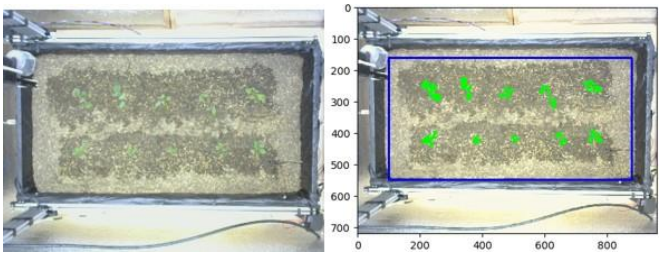


Fig. 5.30 Day 31

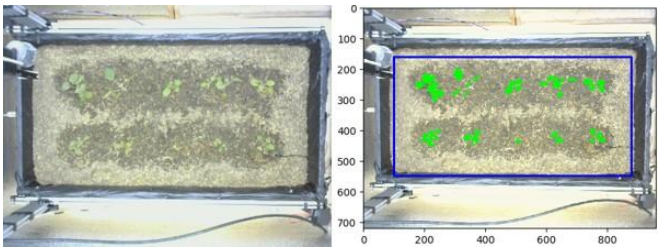


Fig. 5.31 Day 36

To evaluate whether the values were valid, another approach was used. The researchers applied the concept of labeling pixels for semantic segmentation using MATLAB. A label for the ROI of the plant was manually drawn by brush and extracted to the workspace as gTruth. Here, ten separate ROIs shown in ten different colors were used to identify the pixels of each plant.

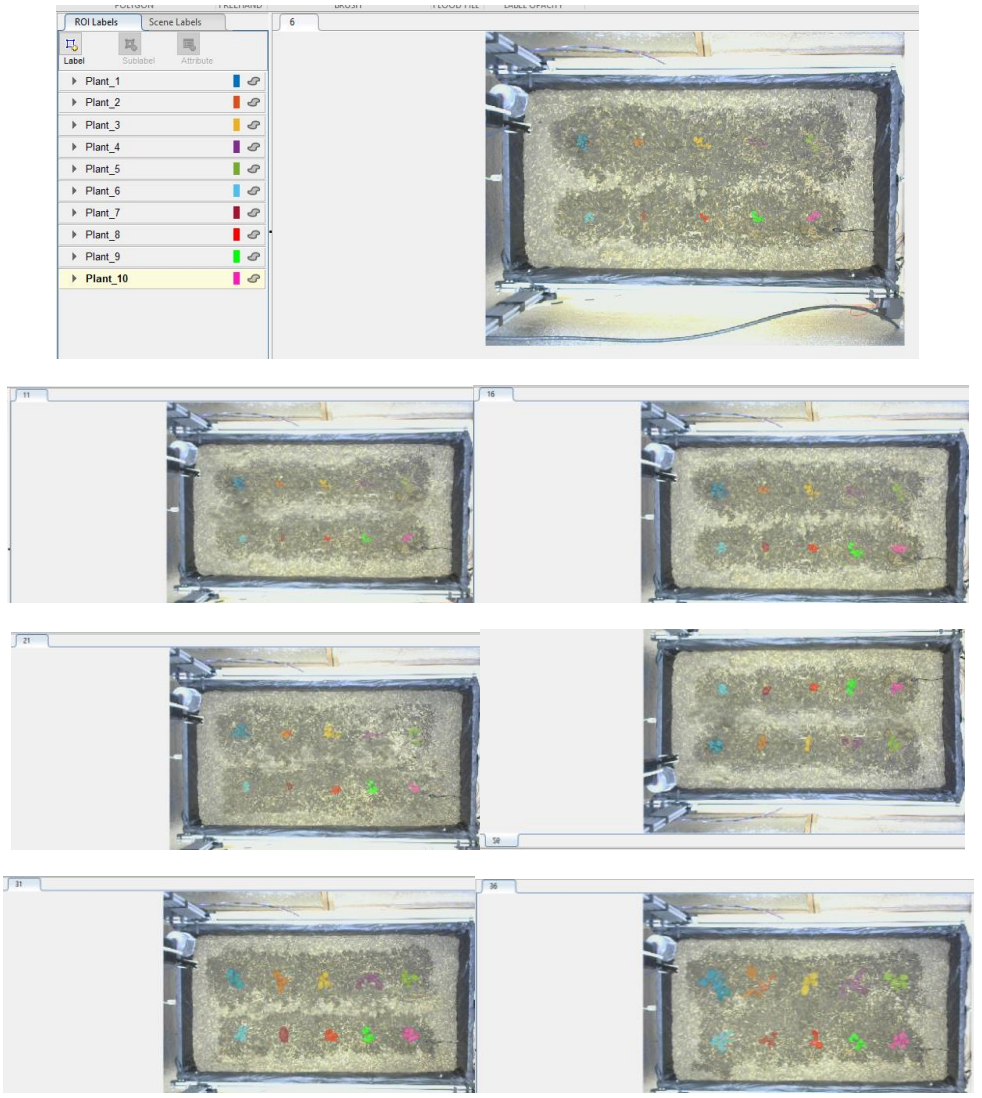


Fig. 5.32 Drawing ROIs for Image Labelling

The labeled pixel photo was stored as a categorical matrix and displayed in the form of sheets. The undefined and the shaded regions from the previous steps were printed and the user can print the output by using countcats, whose role is to return the number of elements in each category of the categorical array.

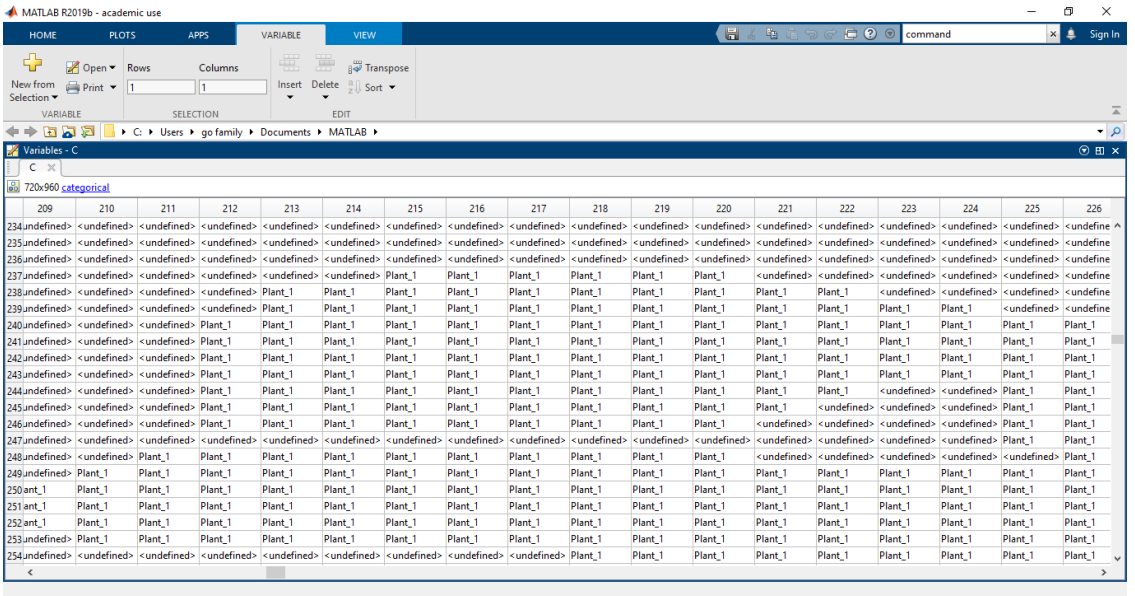


Fig. 5.33 Reading Pixel Image

By using countcats, the user can see the number of times the object appeared per column.

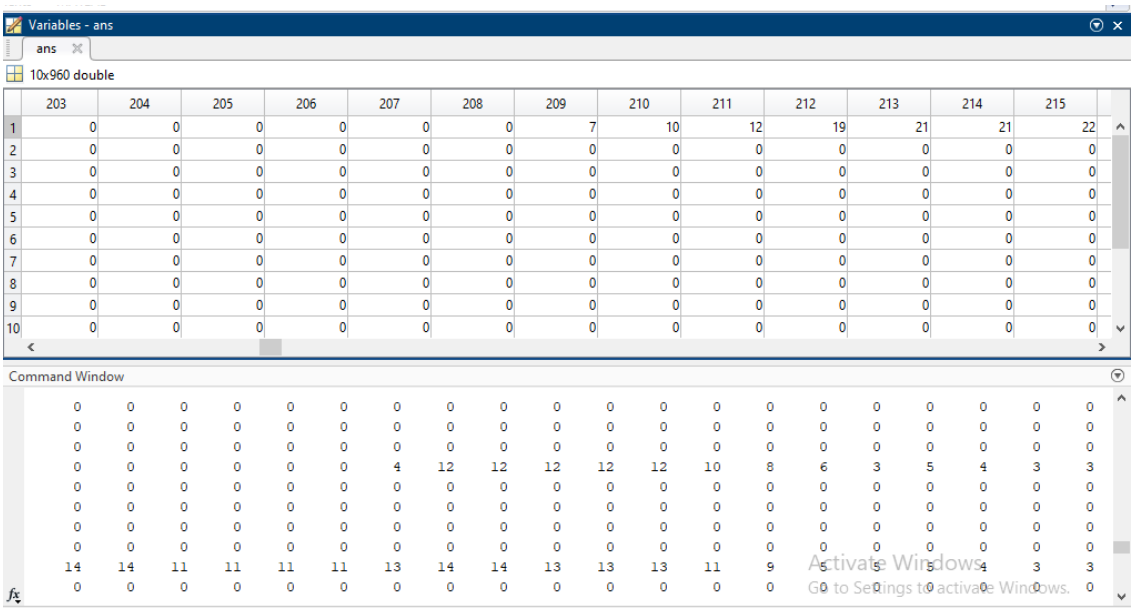


Fig. 5.34 Category Results

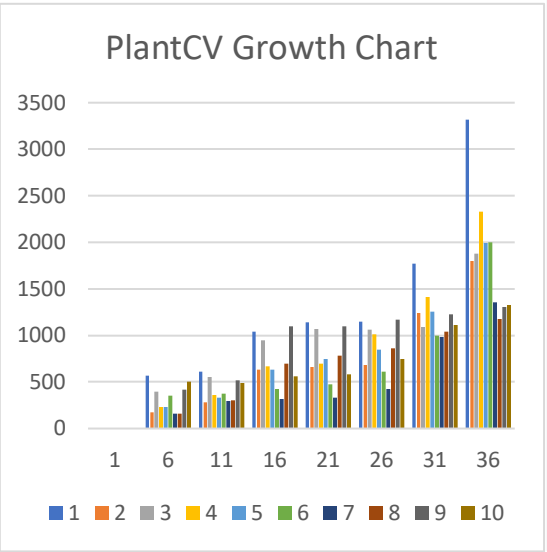
Summing every instance gave the result of the number of pixels it occupies.

```
>> S = sum(ans,2)

S =

   585
   245
   446
   299
   253
   300
   155
   170
   385
   415
```

Fig. 5.35 MATLAB Results

Plant	Day															
	1		6		11		16		21		26		31		36	
	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M
1	0	0	569	585	610	640	1043	1081	1140	1221	1146	1166	1774	1876	3316	3480
2	0	0	176	245	282	317	633	669	661	687	686	705	1242	1394	1800	1932
3	0	0	396	446	556	588	945	1014	1071	1110	1064	1123	1089	1132	1876	1919
4	0	0	230	299	360	393	666	740	696	727	1012	1079	1414	1557	2332	2467
5	0	0	233	253	334	380	635	693	744	761	845	935	1252	1350	1993	1974
6	0	0	351	300	374	401	424	473	478	500	609	629	997	1064	1999	2022
7	0	0	157	155	298	304	315	344	330	401	428	513	986	1112	1359	1399
8	0	0	163	170	301	321	697	707	782	804	864	901	1038	1118	1176	1220
9	0	0	421	385	518	535	1101	1115	1095	1207	1173	1204	1228	1283	1304	1395
10	0	0	501	415	492	531	559	605	581	603	748	766	1115	1179	1326	1524

Table 5.3 PlantCV and MATLAB Results



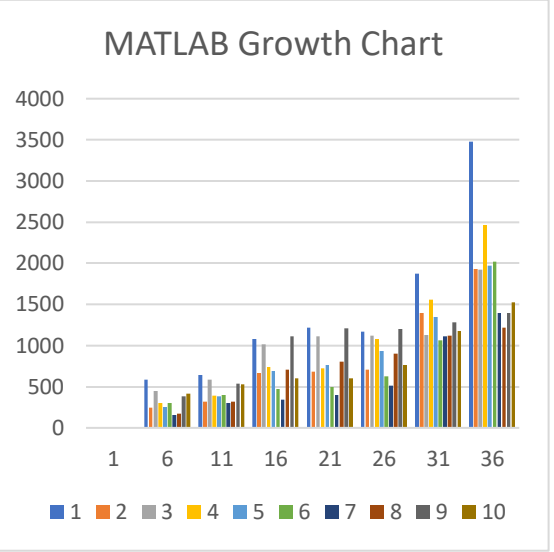


Fig. 5.36 PlantCV and MATLAB Growth Chart

The total average percent difference of the values returned from PlantCV and MATLAB was 6.34%, making the values from PlantCV reliable.

Plant	**Day**							
	1	**6**	**11**	**16**	**21**	**26**	**31**	**36**
1	0	2.77%	4.80%	3.58%	6.86%	1.73%	5.59%	4.83%
2	0	32.78%	11.69%	5.53%	3.86%	2.73%	11.53%	7.07%
3	0	11.88%	5.59%	7.04%	3.58%	5.40%	3.87%	2.27%
4	0	26.09%	8.76%	10.53%	4.36%	6.41%	9.63%	5.63%
5	0	8.23%	12.89%	8.73%	2.26%	10.11%	7.53%	0.96%
6	0	15.67%	6.97%	10.93%	4.50%	3.23%	6.50%	1.14%
7	0	1.28%	1.99%	8.80%	19.43%	18.07%	12.01%	2.90%
8	0	4.20%	6.43%	1.42%	2.77%	4.19%	7.42%	3.67%
9	0	8.93%	3.23%	1.26%	9.73%	2.61%	4.38%	6.74%
10	0	18.78%	7.62%	7.90%	3.72%	2.38%	5.58%	13.89%
							AVERAGE:	6.34%

Table 5.4 PlantCV and MATLAB Percent Difference Table

5.2.2 Sensors

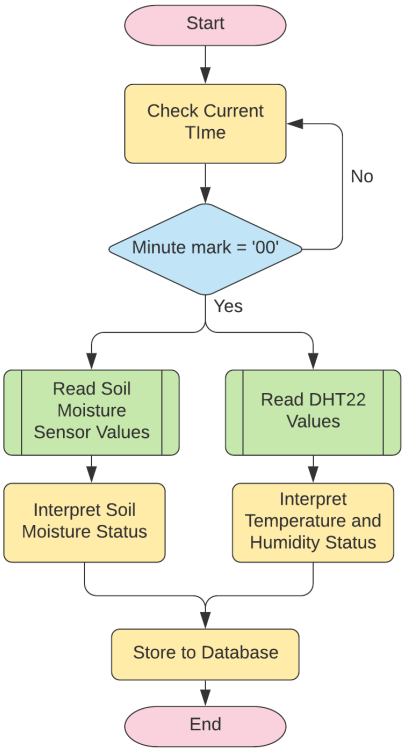


Fig. 5.37 Monitoring System with Sensors Flowchart

To validate the output of the digital sensors, these were compared with their manual or non-digital counterparts. The two kinds of sensors were put beside each other close enough to be measuring the parameter to be evaluated. Each sensor was left inside the cabinet for approximately 3 minutes that allows them to adjust its values before recording and comparing their values.

The outputs for each sensor were tabulated for easier comparison. Percentage error calculation was done to present the amount of difference between the actual and predicted values from the non-digital and digital sensors respectively. The percentage error was calculated using the formula:

$$percentage\ error\ = \left|\frac{absolute\ error}{theoretical\ value}\right| x\ 100\%$$

$$where: \qquad absolute\ error\ =\ actual\ value\ -\ theoretical\ value$$

The actual value being referred to in the formula was the digital sensor output while the theoretical value referred to the output of the non-digital sensor. The mean of the gathered percentage errors was calculated that determined the average percentage error which was also an indicator of how close the data sets are to each other.

In addition to the percentage error, the data gathered was evaluated using the Pearson correlation coefficient and coefficient of determination which was helpful to determine the relationship between two quantitative sets of data. The Pearson correlation coefficient was beneficial to conclude the amount of strength of the association between the two data sets from the digital and non-digital sensors. The Pearson correlation coefficient (r) was calculated using the formula:

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

$$where: \qquad \bar{X} = mean\ of\ x - variable$$

$$X_i = i^{th}\ item\ in\ the\ x - variable$$

$$\bar{Y} = mean\ of\ y - variable$$

$$Y_i = i^{th}\ item\ in\ the\ y - variable$$

$$n = total\ no.of\ data\ pairs$$

$$r_{XY} = Pearson\ Correlation\ Coefficient$$

Fig. 5.38 Pearson Correlation Coefficient Formula

Once the Pearson correlation coefficient has been calculated, this was squared to get the coefficient of determination which was used to predict future outcomes or the

goodness of fit. This served as a model for forecasts since the sample size for each parameter being measured by the sensor was limited only to 15. These calculations and graphical representations were done using the Microsoft Excel application.

5.2.2.1 Trial 1

These were tested in different conditions - dry soil, moist soil, wet soil, dry air, humid air, cold temperature, and warm temperatures.

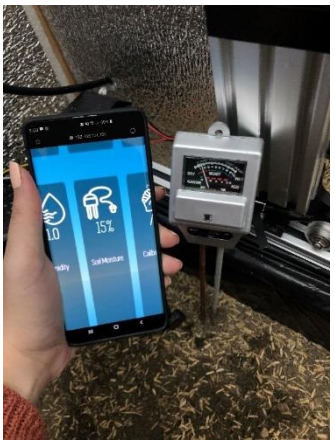


Fig. 5.39 Scenario Soil Moisture Dry 1

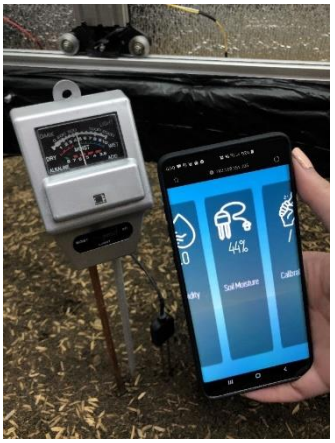


Fig. 5.40 Scenario Soil Moisture Moist 2

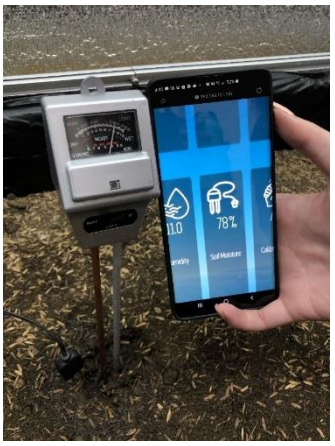


Fig. 5.41 Scenario Soil Moisture Wet 3

Fig. 5.42 Scenario Low Temperature & Normal Humidity 4

Fig. 5.43 Scenario High Temperature & Normal Humidity

For testing the temperature, fans were initially turned off. The two DHT22 were subjected to heat coming from a plugged soldering iron. After heating the two sensors, the average value was at 30 degrees Celsius and the fans were automatically activated.

Fig. 5.44 DHT Left

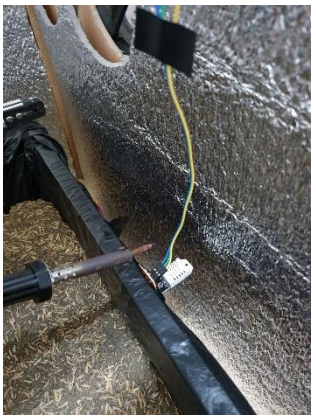


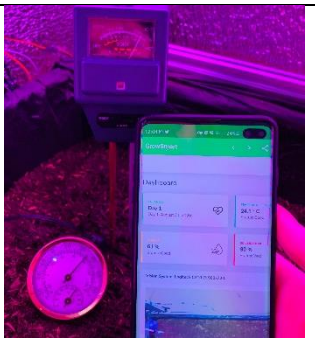
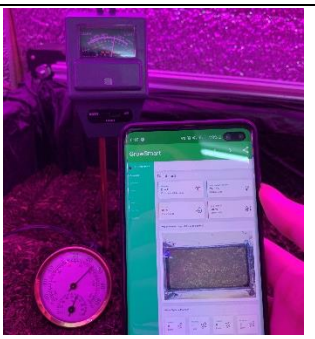
Fig. 5.45 DHT Right

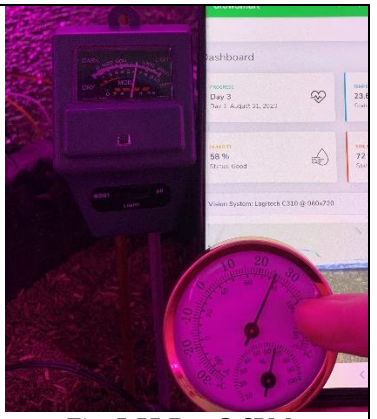
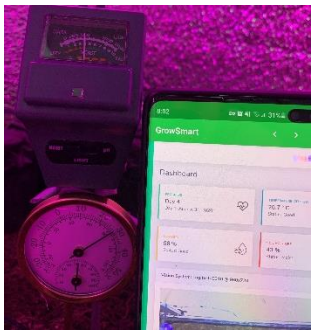
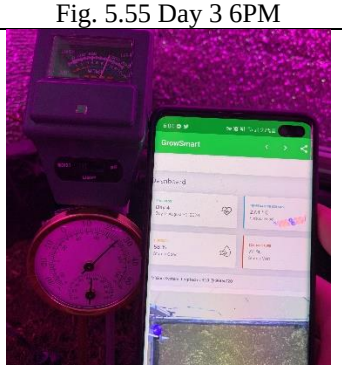
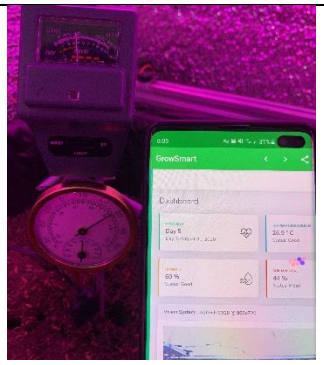
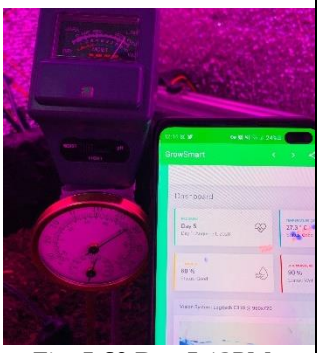
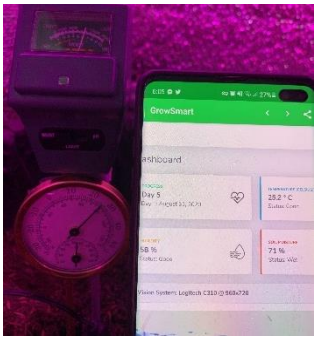
For testing the soil moisture sensor, the watering system was initially turned off. The probe was then placed on dry soil which returned a value of 12%, which was roughly the same with the analog sensor. The reading indicates that the soil was dry thus triggering the irrigation system to begin its process.

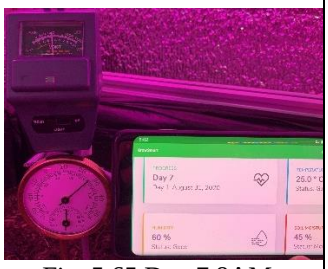
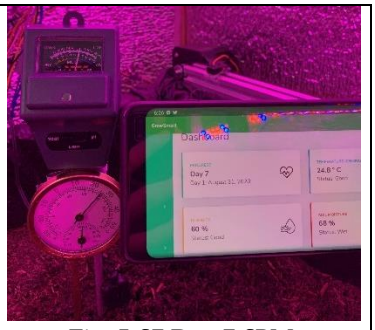
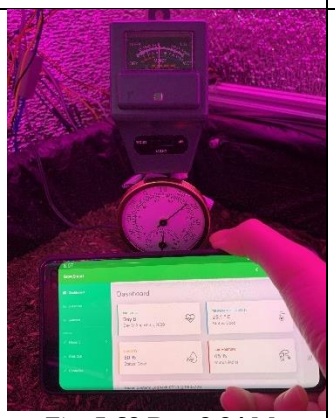
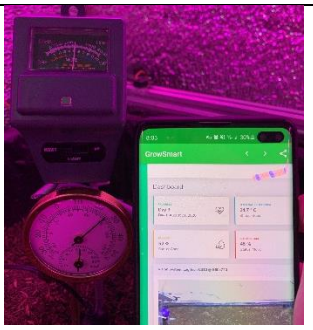
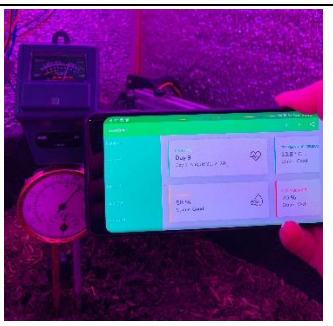
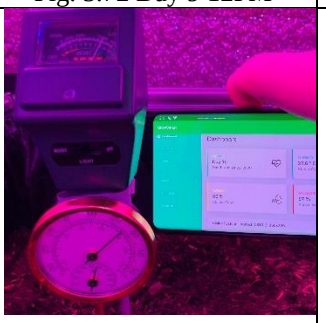
Fig. 5.46 Soil Moisture Sensor

5.2.2.2 Trial 2

A comparison between the actual sensors and commercially available ones was conducted during 8AM – 6PM every day for two weeks. Pictures for only 8AM, 12PM, and 6PM were presented while the rest were tabulated for easier visualization.

Day	8AM	12PM	6PM
1	Fig. 5.47 Day 1 8AM	 Fig. 5.48 Day 1 12PM	 Fig. 5.49 Day 1 6PM
2	Fig. 5.50 Day 2 8AM	Fig. 5.51 Day 2 12PM	Fig. 5.52 Day 2 6PM

3	Fig. 5.53 Day 3 8AM	Fig. 5.54 Day 3 12PM	 Fig. 5.55 Day 3 6PM
4	 Fig. 5.56 Day 4 8AM	Fig. 5.57 Day 4 12PM	 Fig. 5.58 Day 4 6PM
5	 Fig. 5.59 Day 5 8AM	 Fig. 5.60 Day 5 12PM	 Fig. 5.61 Day 5 6PM
6	Fig. 5.62 Day 6 8AM	Fig. 5.63 Day 6 12PM	Fig. 5.64 Day 6 6PM

7	 Fig. 5.65 Day 7 8AM	Fig. 5.66 Day 7 12PM	 Fig. 5.67 Day 7 6PM
8	 Fig. 5.68 Day 8 8AM	Fig. 5.69 Day 8 12PM	Fig. 5.70 Day 8 6PM
9	Fig. 5.71 Day 9 8AM	 Fig. 5.72 Day 9 12PM	 Fig. 5.73 Day 9 6PM
10	Fig. 5.74 Day 10 8AM	 Fig. 5.75 Day 10 12PM	Fig. 5.76 Day 10 6PM

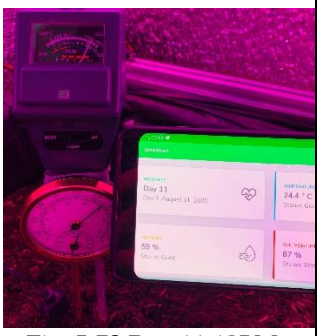
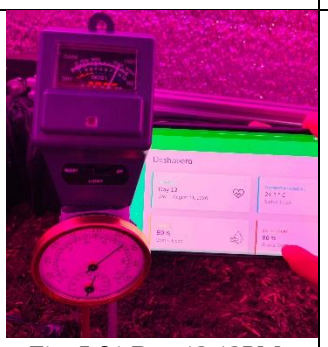
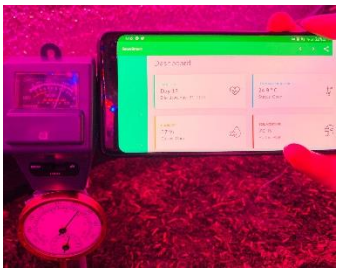
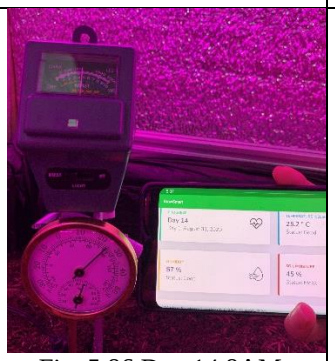
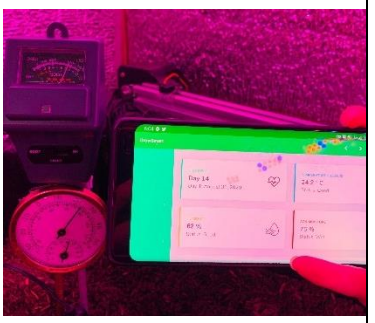
11	Fig. 5.77 Day 11 8AM	 Fig. 5.78 Day 11 12PM	Fig. 5.79 Day 11 6PM
12	Fig. 5.80 Day 12 8AM	 Fig. 5.81 Day 12 12PM	Fig. 5.82 Day 12 6PM
13	Fig. 5.83 Day 13 8AM	Fig. 5.84 Day 13 12PM	 Fig. 5.85 Day 13 6PM
14	 Fig. 5.86 Day 14 8AM	Fig. 5.87 Day 14 12PM	 Fig. 5.88 Day 14 6PM

Table 5.5 Sensors Comparison

5.3 Cost Analysis

The following is a breakdown of all the costs incurred in the creation of the prototype:

Item Name	Unit	Price	Total Price
Water Pump	1	200	**200**
16 Channel 5V Relay Module	1	414	**414**
Raspberry Pi 4	1	4,000	**4000**
50pcs T Sliding Nut	1	175	**175**
4x Stepper Motor Driver DRV8825	1	277	**277**
500mm Lead Screw with Nut	1	381	**381**
Polycarbonate wheels (5 pieces)	2	244	**488**
Nema17 Stepper Motor	4	312	**1248**
2x GT2 Pulley + 2x 6mm Timing Belt	2	211	**422**
500mm 2060 T-Slot Aluminum Profile	8	695	**5560**
Logitech C310 HD Webcam	1	1550	**1550**
5mm Ball Bearing	20	40	**800**
5mm Locknut	20	1.08	**21.6**
Bolt	40	0.68	**27.2**
Screws	20	1.47	**29.4**
Allen Cap M5x30	6	5	**30**
5mm Ball Bearing	6	60	**360**
Pechay Seeds	1	74	**74**
Plant Box	1	475	**475**
Cabinet Enclosure	1	2825	**2825**
LED Light	1	300	**300**
Neopixel Light Strip	4	500	**2000**
12V Vacuum Pump	1	688	**688**
3m Double Sided Aluminum Foil Insulation	1	370	**370**
5V 10A Power Supply	5	500	**2500**
12V 30A Power Supply	1	1000	**1000**
Voltage Regulator 3.3V EZ1084CT	1	55	**55**
50pcs Transistor NPN	1	72	**72**

Stepper Motor DRV8825	1	139	**139**
Fan 12V DC	4	109	**436**
Loam Soil	10	20	**200**
Fertilizer	1	200	**200**
MCP3008	1	275	**275**
Drip Irrigation	1	202	**202**
Syringe Needle Tip	1	106	**106**
Electrical Tape	1	120	**120**
Soil Moisture Meter	1	226	**226**
Indoor Analog Humidity Temperature Thermometer	1	123	**123**
2020 Aluminum Extrusion	1	502	**502**
Limit Switches	5	29	**145**
Soil Moisture Detection Corrosion Probe	1	306	**306**
DHT22	2	200	**400**
Trash Bag	1	165	**165**
Terminal Block	1	89	**89**
Wires	1	400	**400**
		Total Amount	**Php 30376.2**

Table 5.6 Cost Breakdown

The total cost for the creation of one fully functional prototype was Php 30,376.20. For the experimentation and testing of the functionality of the prototype, seeds of Pechay were used as subject. Pechay was a relatively easy to grow crop since it does not need much maintenance as compared to other crops.

As stated from previous chapters, pechay has a selling price between Php 60/kilo (lowest) and Php 100/kilo as of June 18, 2020 according to the Bantay Presyo Prices Monitoring of the Department of Agriculture. It provided a comprehensive report of the retail prices of selected agri-fishery commodities in selected markets in Metro Manila.

Assuming that the fresh weight of mature pechay from each plot is 210g, the 5 x 2 plotting mode produces 2100g or 2.1kg of mature pechay ready for harvesting after

each cycle of 30 days. If 2.1kg of pechay was produced per 30 days or exactly a month, it can be sold from Php 126 (based on Php 60/kilo) to Php 210 (based on Php 100/kilo). In a year, income would be around Php 1,512 to Php 2,520.

By dividing the total cost of the prototype which is Php 30,376.20 by the yearly income of Php 1,512 to Php 2,520, the minimum and maximum number of years before the return of investment will be obtained.

$$ROI_{min} = \frac{Php\ 30{,}376.20}{Php\ 2{,}520\ per\ year} = 12.054\ years$$

$$ROI_{max} = \frac{Php\ 30{,}376.20}{Php\ 1{,}512\ per\ year} = 20.090\ years$$

This project however is also capable of growing other crops and for computation purposes, the vegetable Arugula will be used is a kind of vegetable belonging to the family of broccolis, kales, and Brussel sprouts. Arugulas are relatively hard to grow since it is a cool-season crop and would require a tremendous amount of care and monitoring to produce it. This is where our project comes in handy. According to The Green Grocer Manila website, 250 grams of Organic Arugula costs Php 180 and it also requires the similar if not same planting distance from the pechay crop. This means that the 5 x 2 mode can be used again and thus 10 Arugulas can be harvested every cycle. They are typically ready to harvest after at least 35 days.

If it takes 35 days for Arugula seeds to be mature and ready harvesting, that would mean you can only produce 10.428 or approximately 10 cycle-amount of Arugulas. Assuming that the fresh weight of mature Arugula is around 200 grams, the plotting mode 5 x 2 produces 2kg of Arugula per 35-day cycle. If 2kg of Arugula is

produced per 35-day cycle, it means that a person can sell 20kg per year. The yearly income would then be Php 14,400.

$$\frac{180\ Php}{250\ g} x \left(20\ \frac{kg}{year}\right)\left(1000\frac{g}{kg}\right) = Php\ 14{,}400$$

By dividing the total cost of the prototype which is Php 30,376.20 by the yearly income of Php 14,400, the number of years before the return of investment will be obtained.

$$ROI = \frac{Php\ 30{,}376.20}{Php\ 14{,}400\ per\ year} = 2.109\ years$$

Fig. 5.89 Plant Weight

Fig. 5.90 Plant Height

Fig. 5.91 Plant True Leaves

Plant	Ave. Weight (g)	Ave. Height (cm)	Ave. Number of Leaves
	0.9g	6.5cm	4

Table 5.7 Plant Weight

The figures above show how the Pechay looks 14 days after they have been planted. In this trial, the height and the weight of the plant were measured by removing them from the soil and then using a ruler, and a digital weighing scale. Table 5.6 shows the average weight, height, and number of leaves of the 10 plots of Pechay planted in this trial. The group used soil that was mixed with vermicast, a type of organic fertilizer that provided the best growth and yield performance among different varieties of pechay in a study conducted by Saanoding A. Balayo.

The average number of leaves that sprouted from each plot after 2 weeks were 4 which is close to the number of leaves when it is fully grown according to a study that focused on the response of Pechay to different compost fertilizers. Gonzales et. al observed that the average number of leaves of a fully grown pechay is around 5.434 however, since this trial was cut short in 14 days, the group's average number of leaves can further increase as it has not fully matured yet. As seen in Fig. 5.48, the leaves are still small which explains why the average weight is also significantly small. However, these are also bound to increase over time as the number of leaves and the leaf area are still bound to change and expand more.

5.4 Summary

The researchers considered the implementation to be successful when the cartesian robot, the Django server together with the sensors, and the Arduino Mega work simultaneously without failing. The cartesian robot was evaluated through thorough testing and analyzing its performance for both seeding and watering operations. The Django server controlled the devices attached to it well, displayed the picture, monitored plant growth, sensor values, and device statuses without errors. The Arduino Mega successfully followed the commands given by the Raspberry Pi and performed the tasks assigned. For the growth status, several pictures were tested to check the accuracy of the image processing code. For the sensors, these were tested under extreme conditions and the different automated responses of the system were observed and analyzed. For the Arduino Mega, the test consisted of it being subjected to different inputs from the Raspberry Pi. The correctness and its ability to follow the commands given were evaluated as well.

First, the mechanical-electrical structure of the robot was successfully built. Second, electronic components, firmware, robot work tools, and artificial vision were installed properly. Third, the programs were designed to ensure efficient work and without robot collisions. Lastly, all modules were integrated and tested altogether.

The researchers were able to conceptualize, develop, and construct an indoor gardening system capable of seeding, planting, and monitoring using a Raspberry Pi as its main computer and server host, Logitech C310 and PlantCV as its vision system, 3D printed parts, Nema 17 stepper motors, and aluminum profiles as the foundation of the cartesian robot, Django as its server, and Arduino as the robot controller. The system had 4 modes of operation - mode 1, mode 2, mode 3, and mode 4. Each mode has different specifications.

The researchers also positioned the devices such as the lights, fans, camera, and sensors strategically to properly set up the controlled environment. The Django server was also developed in such a way that it would automate the entire system with ease with the help of Asynchronous JavaScript and XML. With this, the entire system updates asynchronously by exchanging small amounts of data with the server behind the scenes that made it responsive, dynamic, and fast. With the use of Python's PlantCV library, a Logitech C310 webcam, and Raspberry Pi, the researchers were able to develop an image processing algorithm that acts as an accurate growth tracking system for the crops grown inside the system. With this, a collection of pictures and growth data was also produced providing significant information about the efficiency and effectiveness of the indoor gardening system. The researchers conducted several tests for the different functionalities of the system. There were numerous problems encountered by the researchers such as wrong placements of the sensors, fans, and lights making them

inefficient as well as the cartesian robot not functioning properly. These problems were addressed as the research went on, trial and errors were done to solve the issues, solutions were made and the outcome is a working indoor gardening system capable of doing basic gardening tasks such as seeding and watering as well as monitoring and communicating with the Django server through the internet connection.

Chapter 6 RESULTS AND DISCUSSION

Contents

6.1 Cartesian Robot System 170
6.1.1 Seeding Accuracy 170
6.1.2 Uniform and Variable Seed Distribution 177
6.1.3 Irrigation System 180
6.2 Mobile Application Monitoring System 180
6.2.1 Vision System 187
6.2.2.1 Trial 1 188
6.2.2.2 Trial 2 189
6.2.2 Sensors 190
6.2.2.1 Trial 1 190
6.2.2.2 Trial 2 193
6.3 Cost Analysis 194
6.4 Summary 195

6.1 Cartesian Robot System

6.1.1 Seeding Accuracy

Figure 6.1 shows the digitized plotting of seeds with the trials on the plant box using Desmos. The circles signify the area that is within the acceptable range for the seeder. Each circle has a radius of 3 cm each. The trials are differentiated from one another by using different number of colors in the plot.

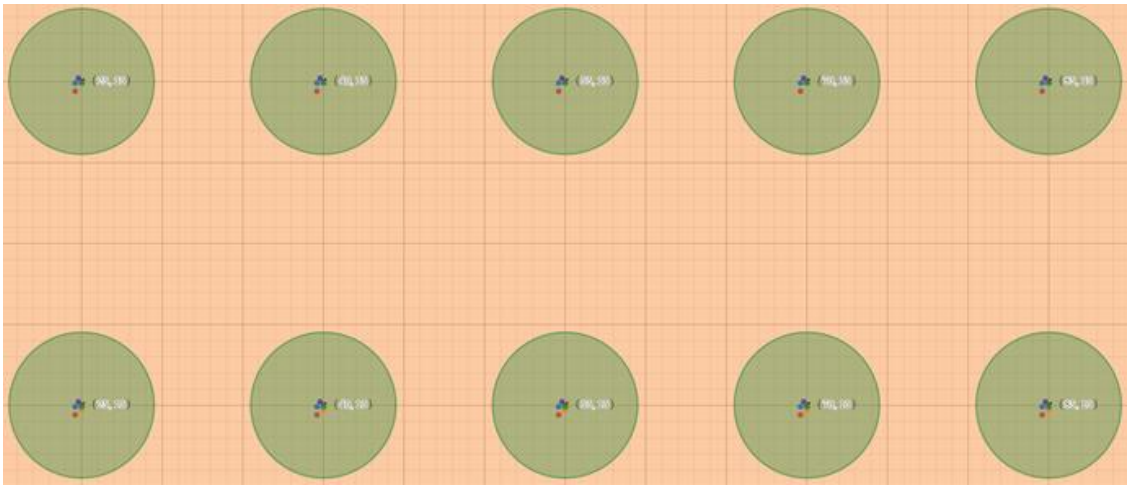


Fig. 6.1 Digitized Plotting of Seeding Results on Desmos

The following figures and tables below present the plotted results and a tabulated data containing the trial number, location, and radial distance on each coordinate. The distances between each coordinate and the target coordinate were calculated using the distance formula. The mean radial distance as well is calculated which is used for the calculation for the standard deviation in evaluating the seeder's performance.

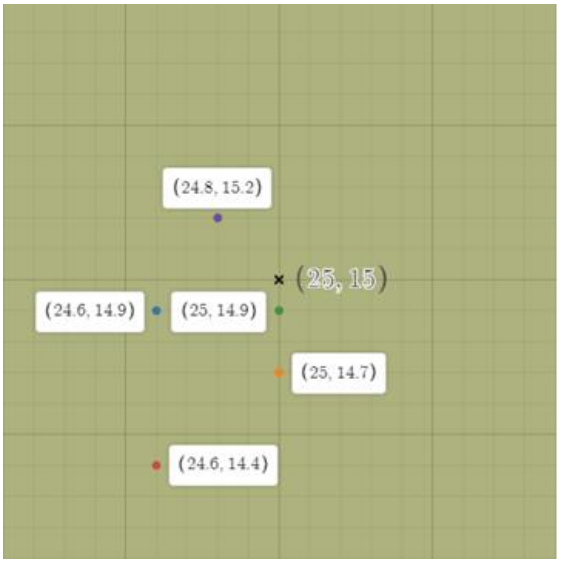


Fig. 6.2 Planting Ground No. 1 Centered at (25, 15)

Table 6.1 shows the actual location of the seeder and the computed average radial distance for planting ground no. 1 is 0.3633 cm from the target location (25,15) as seen in Fig. 6.2.

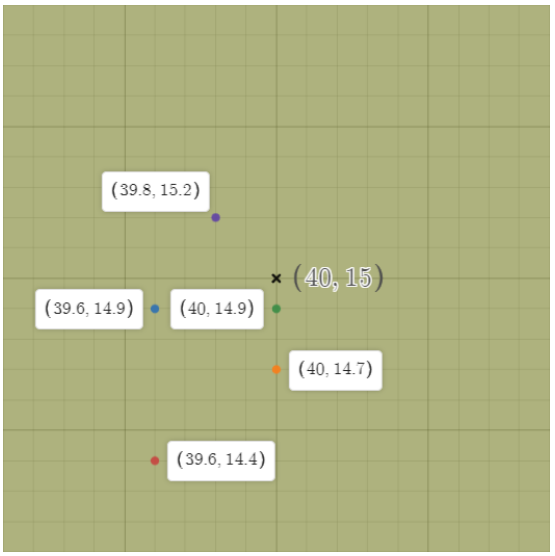


Fig. 6.3 Planting Ground No. 2 Centered at (40, 15)

Table 6.2 shows the actual location of the seeder and the computed average radial distance for planting ground no. 2 is 0.3633 cm from the target location (40,15) as seen in Fig. 6.3.

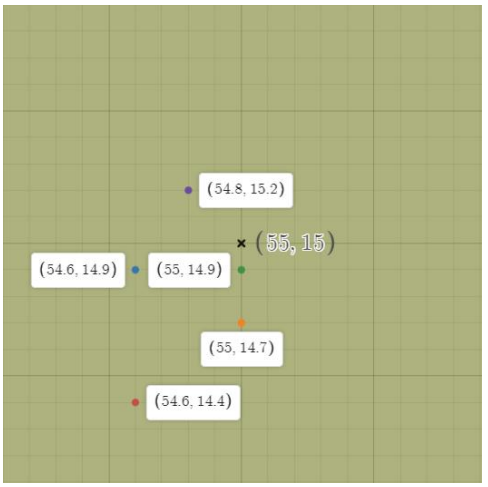


Fig. 6.4 Planting Ground No. 3 Centered at (55, 15)

Table 6.3 shows the actual location of the seeder and the computed average radial distance for planting ground no. 3 is 0.3633 cm from the target location (55,15) as seen in Fig. 6.4.

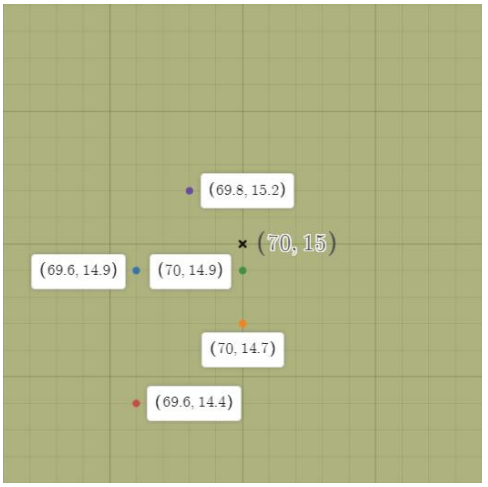


Fig. 6.5 Planting Ground No. 4 Centered at (70, 15)

Table 6.4 shows the actual location of the seeder and the computed average radial distance for planting ground no. 4 is 0.3633 cm from the target location (70,15) as seen in Fig. 6.5.

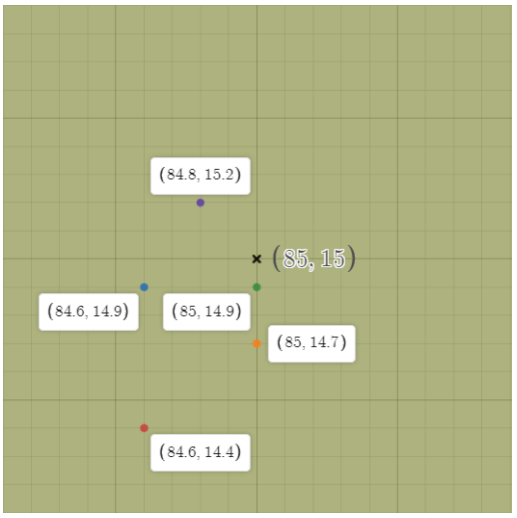


Fig. 6.6 Planting Ground No. 5 Centered at (85, 15)

Table 6.5 shows the actual location of the seeder and the computed average radial distance for planting ground no. 5 is 0.3633 cm from the target location (85,15) as seen in Fig. 6.6.

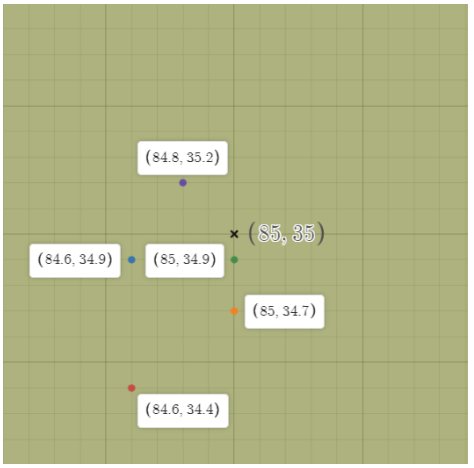


Fig. 6.7 Planting Ground No. 6 Centered at (85, 35)

Table 6.6 shows the actual location of the seeder and the computed average radial distance for planting ground no. 6 is 0.3633 cm from the target location (85,35) as seen in Fig. 6.7.

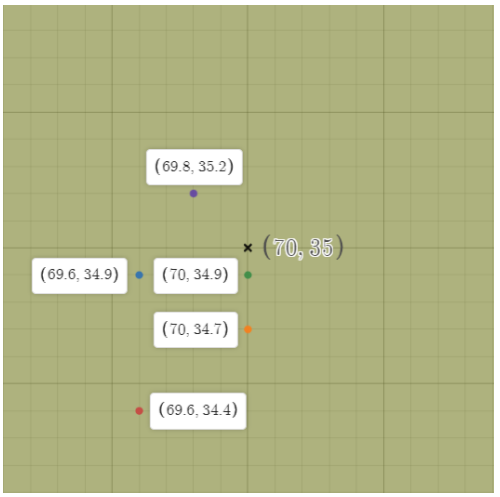


Fig. 6.8 Planting Ground No. 7 Centered at (70, 35)

Table 6.7 shows the actual location of the seeder and the computed average radial distance for planting ground no. 7 is 0.3633 cm from the target location (70,35) as seen in Fig. 6.8.

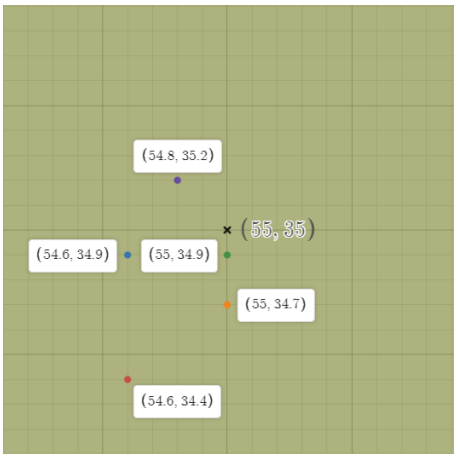


Fig. 6.9 Planting Ground No. 8 Centered at (55, 35)

Table 6.8 shows the actual location of the seeder and the computed average radial distance for planting ground no. 8 is 0.3633 cm from the target location (55,35) as seen in Fig. 6.9.

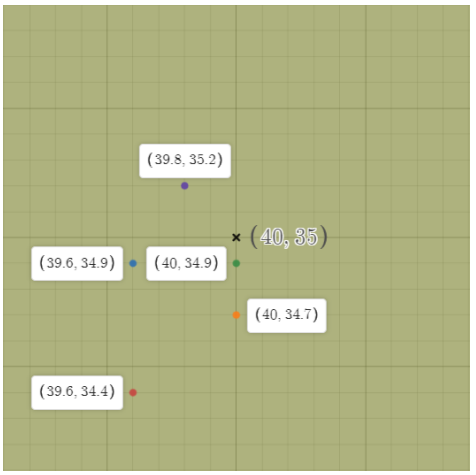


Fig. 6.10 Planting Ground No. 9 Centered at (40, 35)

Table 6.9 shows the actual location of the seeder and the computed average radial distance for planting ground no. 9 is 0.3633 cm from the target location (40,35) as seen in Fig. 6.10.

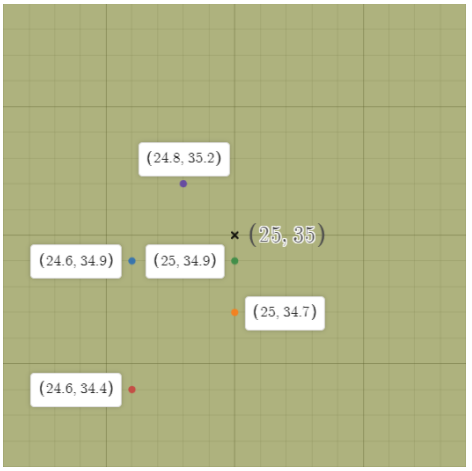


Fig. 6.11 Planting Ground No. 10 Centered at (25, 35)

Table 6.10 shows the actual location of the seeder and the computed average radial distance for planting ground no. 9 is 0.3633 cm from the target location (25,35) as seen in Fig. 6.11.

Trial No.	Radial Distance (cm)	Mean (cm)	Deviation (cm)	Variance (cm)
1	0.7211	0.3633	0.3579	0.128062025
2	0.4123	0.3633	0.0491	0.002406673
3	0.3000	0.3633	-0.0633	0.004000905
4	0.1000	0.3633	-0.2633	0.069301987
5	0.2828	0.3633	-0.0804	0.006465767

Table 6.11 Tabulation of Standard Deviation per Trial

$$\sigma = \sqrt{\frac{0.128062025 + 0.002406673 + 0.004000905 + 0.069301987 + 0.006465767}{5}}$$

$$\sigma = 0.205054801$$

As seen in Table 6.11, the computation for the standard deviation for each trial was the same and yielded a low value of 0.205054801. With the value for the standard deviation being small, it can be inferred that the coordinates that were measured in each trial were close to the mean of the data set. This meant that most of the seeds would be planted in the right locations. The values obtained were made possible because of the lock mechanism of the motors when plugged into power.

Values were accurate because of the calibration of the machine before it performs a specific set of processes. The errors or differences in the marks as compared to the expected location were mainly produced by human error. The manual removal and putting off writing materials close to the seeder affected the actual points by which the robot goes to. Differences between sizes or writing materials have also contributed to the production of minimal errors.

6.1.2 Uniform and Variable Seed Distribution

The serial monitor displayed the status and the coordinates where the seeds are currently placed. Coordinates were printed in a x:distance, y:distance, and z:distance pattern as printed in the serial monitor. This also showed the status of the seeder whether it has already planted the seed in the correct coordinate, or in the process of getting seeds from the seed tray. These can be seen in the serial monitor outputs in Fig 6.12 and 6.13.

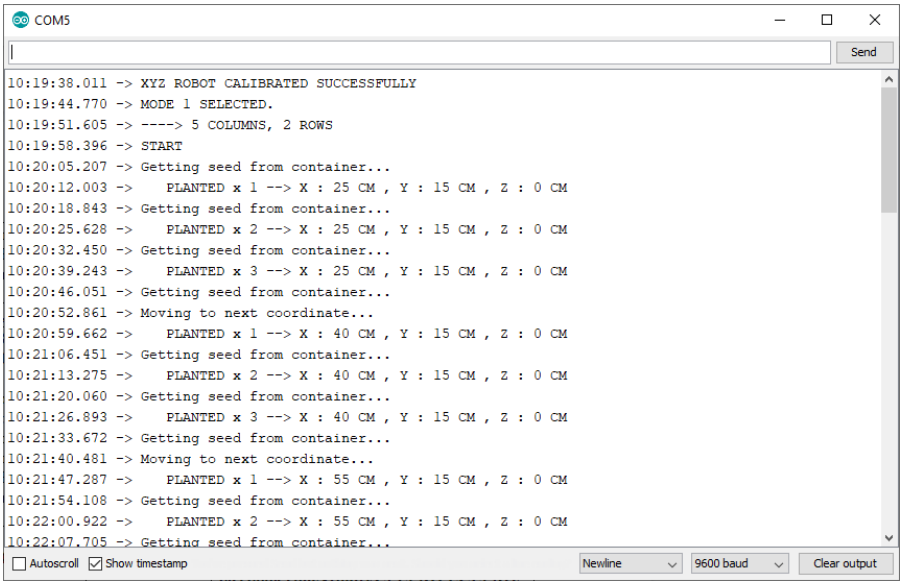


Fig. 6.12 Serial Monitor Output

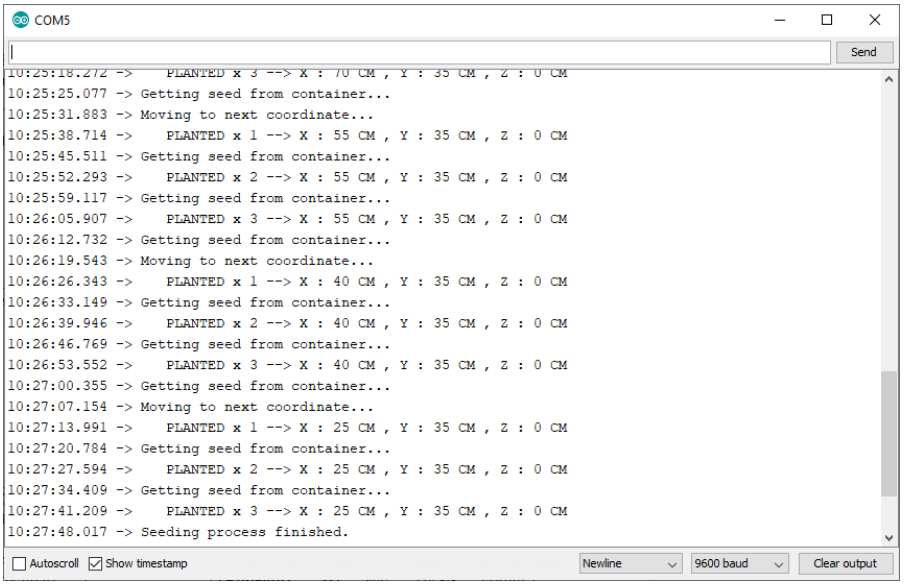


Fig. 6.13 Serial Monitor Output

Fig 6.14 to Fig 6.17 shows the plots that the seeder imparts on the planting ground by impaling the soil. These match the corresponding settings that are set in the program.

Fig. 6.14 Mode 00

Fig. 6.15 Mode 01

Fig. 6.16 Mode 10

Fig. 6.17 Mode 11

6.1.3 Irrigation System

Figure 6.18 shows the graphical representation of data gathered during the testing phase. The automatic programmed system caused the peak in soil moisture value at the said hour from day 1 to day 4. However, during the whole testing phase, it was observed that the scheduled watering of the crops was sufficient since the soil moisture of the system did not go below 30 during the whole cycle of testing.

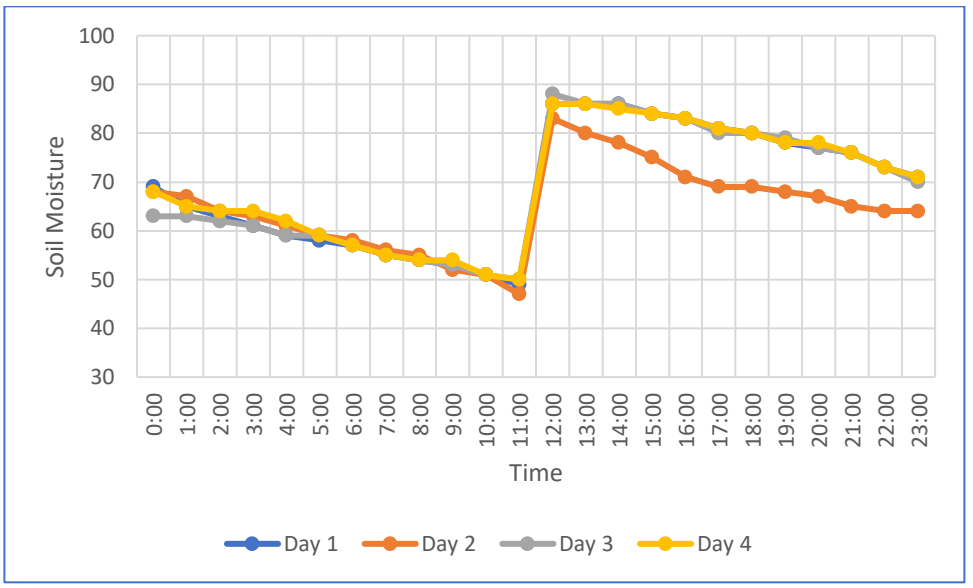


Fig. 6.18 Soil Moisture Data from Days 1-4 of Testing from the Database

6.2 Mobile Application Monitoring System

The mobile application of the system provides the user a dashboard that shows the different parameters that the system is capable of monitoring. Fig 6.19 shows the interface of the application upon startup.

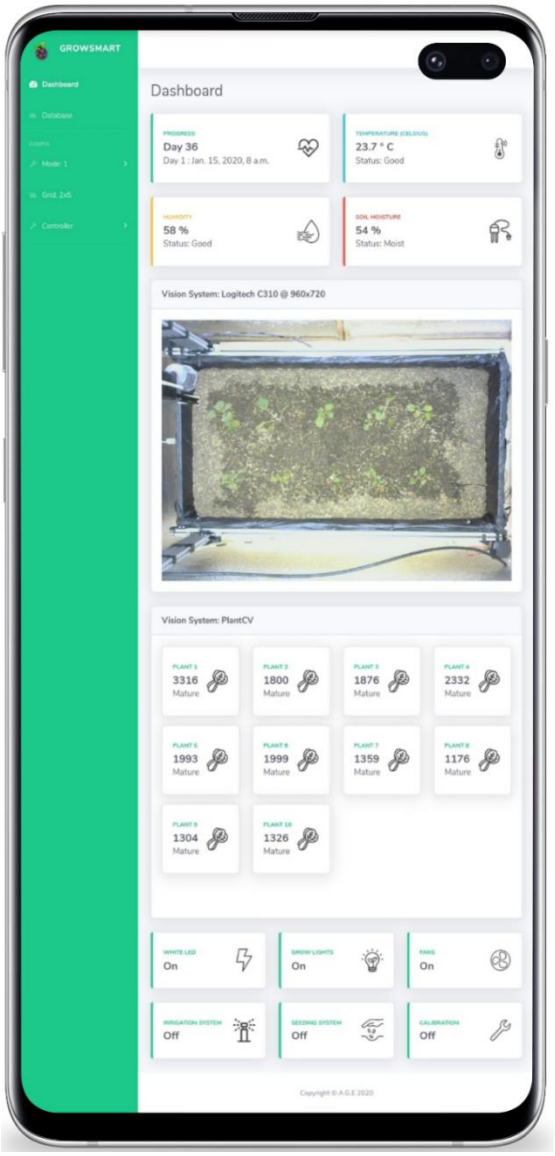


Fig. 6.19 GrowSmart Mobile Application Dashboard

The screenshot of the mobile application shows the growth monitor that prints the pixel value of each plant coordinate and displays the parameters read by the system's sensors as seen in Fig 6.20 and Fig 6.21 below.

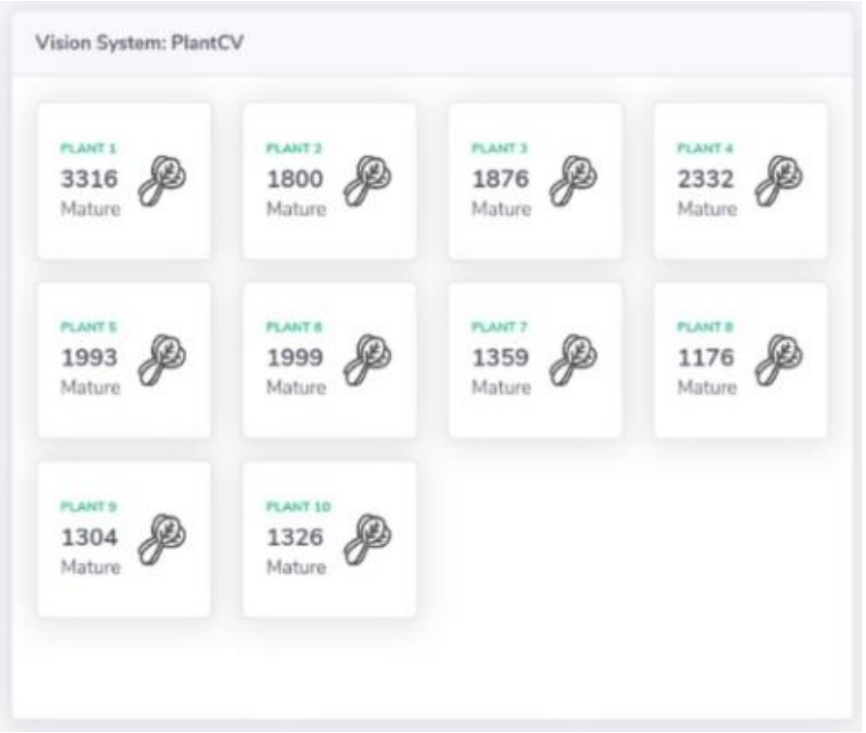


Fig. 6.20 Growth Monitor

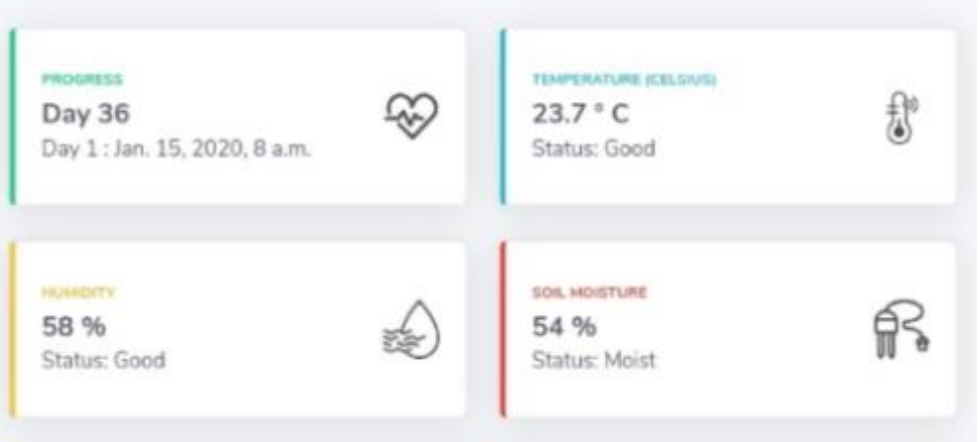


Fig. 6.21 Sensor Monitor

Fig. 6.22 Day 1

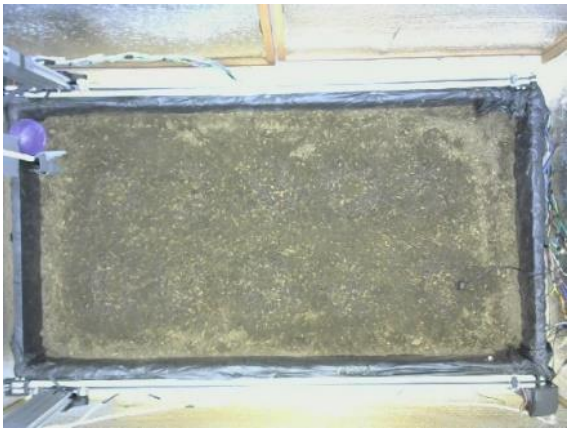


Fig. 6.23 Day 2

Fig. 6.24 Day 3

Fig. 6.25 Day 4

Fig. 6.26 Day 5

Fig. 6.27 Day 6

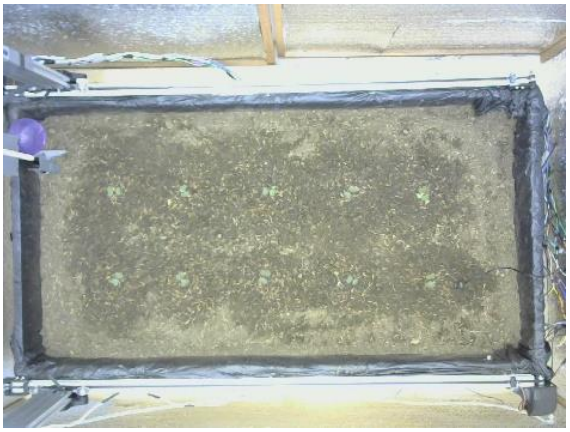


Fig. 6.28 Day 7

Fig. 6.29 Day 8

Fig. 6.30 Day 9

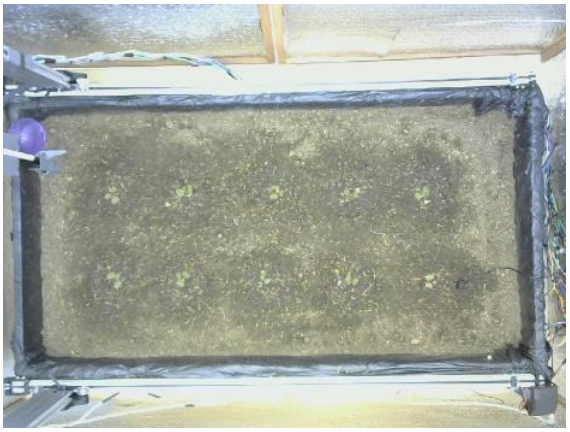


Fig. 6.31 Day 10

Fig. 6.32 Day 11

Fig. 6.33 Day 12

Fig. 6.34 Day 13

Fig. 6.35 Day 14

Fig.6.22 to Fig. 6.35 shows the photos that were taken for each day during the 14-day trial of the prototype. The system takes a photo of the ground at the end of each day with the grow lights closed and the white LED open for clarity.

6.2.1 Vision System

Values from both trials were tabulated and graphically presented. The green bars represented the pixel value the crop occupied while the orange line represented the average value of all the crop on that day.

According to data, the pixel data for each plant continuously increased. Some decrease may be due to the plant's growth direction that made the leaves bend sideways and influence its overall position. Since the camera was placed on the top eye position, the image taken processed only those seen from the view. High chances of leaves overlapping occurred that can contribute to sudden fluctuations of values. Regardless, the average percent change from the days were still positive, showing that plants were continuously growing.

6.2.2.1 Trial 1

Figure 6.38 presents the growth chart of each plant through the days.

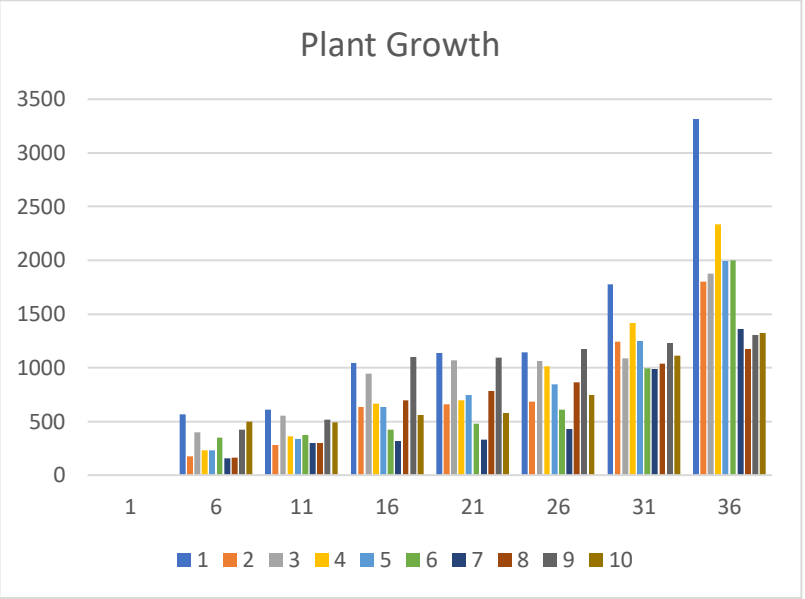


Fig. 6.38 Plants 1-10 Growth Chart (Trial 1)

Day 6 to 11	Day 11 to 16	Day 16 to 21	Day 21 to 26	Day 26 to 31	Day 31 to 36
41%	72%	8%	17%	49%	50%

Table 6.13 Average % Increase for Plants

Table 6.13 shows the average percentage increase in pixel value for each plant in 36 days. From the values shown and the chart above, it can be inferred that the plant

grows significantly in the during the early and late stages. These pixel areas occupied are then tabulated from Fig 6.39 to 6.48. The green bars represent the pixel values on each specific day while the orange line represents the average pixel value for all the plants. This is to show whether the growth per plot is below or above average.

6.2.2.2 Trial 2

A second trial, which lasted for 14 days only, was done to further test the image processing algorithm of the system to verify whether growth can be measured accurately with the algorithm. Table 6.14 shows the data for this 14-day course and these are tabulated daily.

The values from Table 6.12 are then represented in a bar chart in Fig 6.49 for easier visualization.

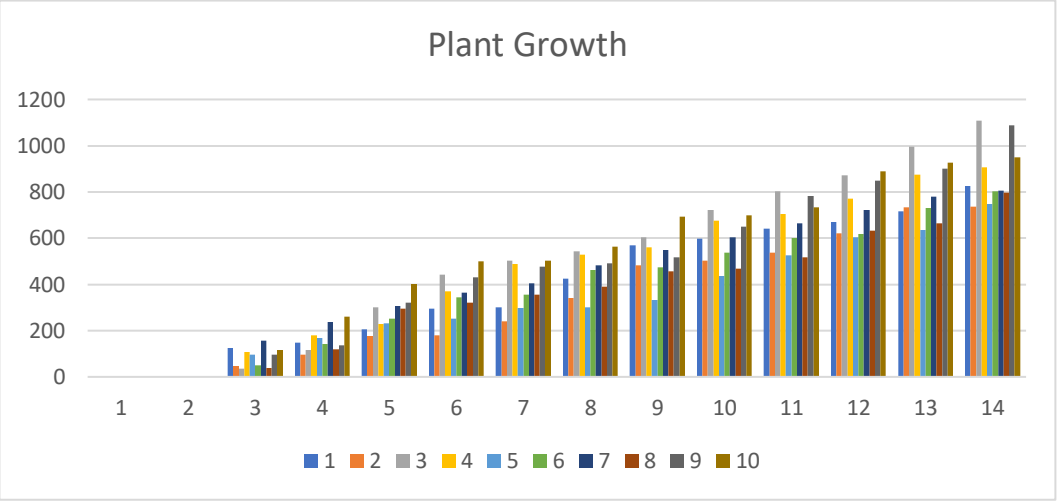


Fig. 6.49 Plants 1-10 Growth Chart (Trial 2)

From the chart above, it can be inferred that the plants are indeed growing in size in terms of pixel area occupied. These pixel areas occupied are then tabulated from Fig 6.50 to 6.59. The orange bars represent the pixel values on each specific day while the

green line represents the average pixel value for all the plants. This is to show whether the growth per plot is below or above average.

6.2.2 Sensors

The system recorded three sensor values every hour and stored it in the database. These data were used on the website to display the numerical values of the sensors. Through this, the user was able to monitor the conditions.

6.2.2.1 Trial 1

Table 6.15 and Fig 6.60 contain information regarding the soil moisture data. As seen, the highest percentage error was 7.143% and the average percentage error was 3.099% which was relatively low. The computed coefficient of determination which was 0.9977. This meant that the actual and predicted response of soil moisture values were highly reliable for future outcomes thus, describes that the digital soil moisture sensor is accurate and provides acceptable output.

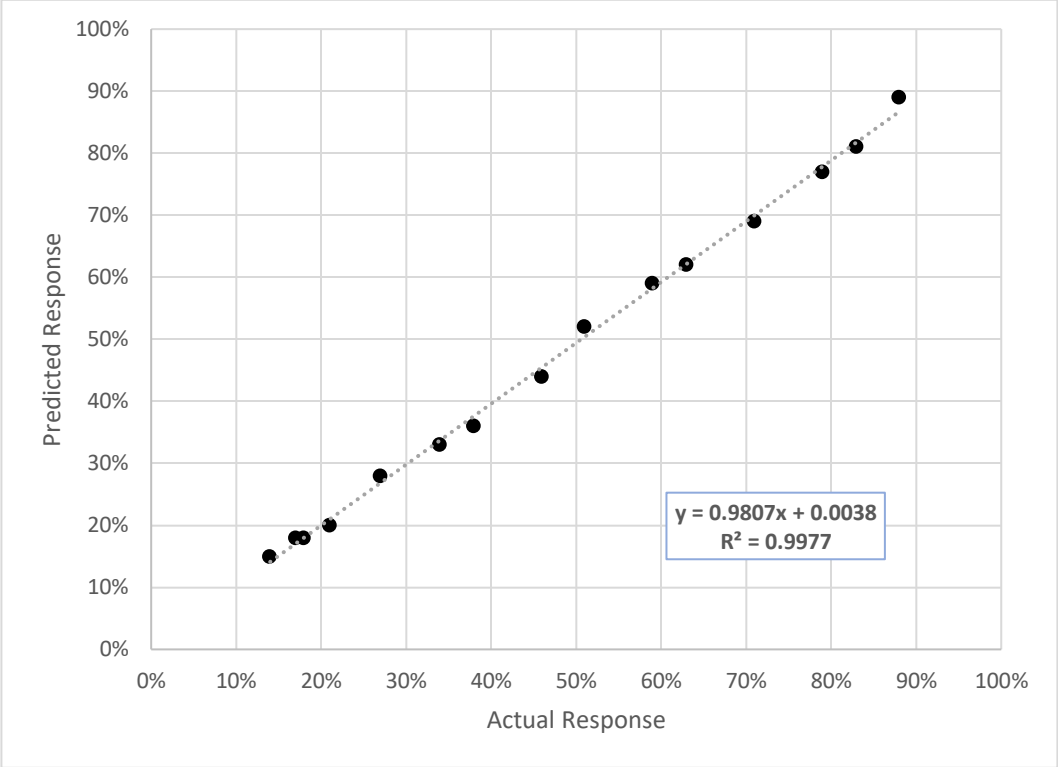


Fig. 6.60 Correlation Graph Between Actual Response and Predicted Response of Soil Moisture Data

Table 6.16 and Fig 6.61 contain information regarding the temperature data. As seen above, the highest percentage error is 5.625% and the average percentage error is 3.388% which is relatively low. The computed coefficient of determination which is 0.9946 infers that the actual and predicted response of temperature values are highly reliable for future outcomes thus describes that the DHT22 temperature sensor is accurate and provides acceptable output.

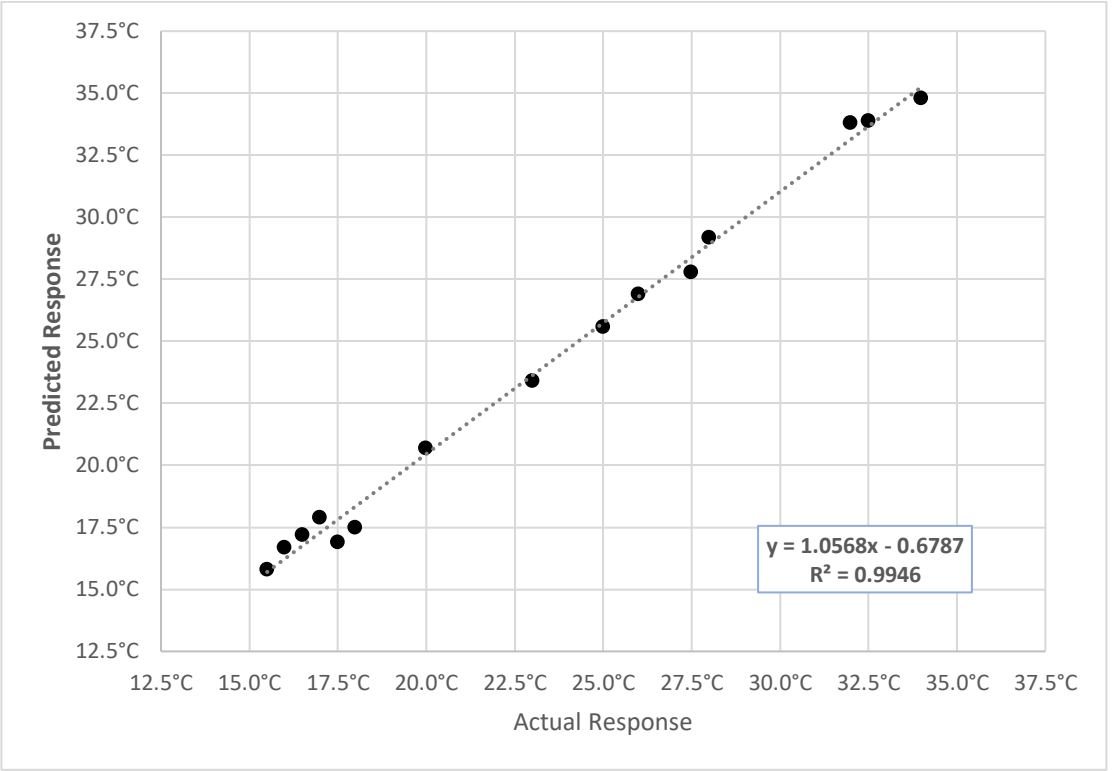


Fig. 6.61 Correlation Graph Between Actual Response and Predicted Response of Temperature Data

Table 6.17 and Fig 6.62 contain information regarding the humidity data. As seen above, the highest percentage error is 3.860% and the average percentage error is 1.656% which is relatively low. The computed coefficient of determination which is 0.973 infers that the actual and predicted response of humidity values are highly reliable for future outcomes thus, describes that the DHT22 humidity sensor is accurate and provides acceptable output.

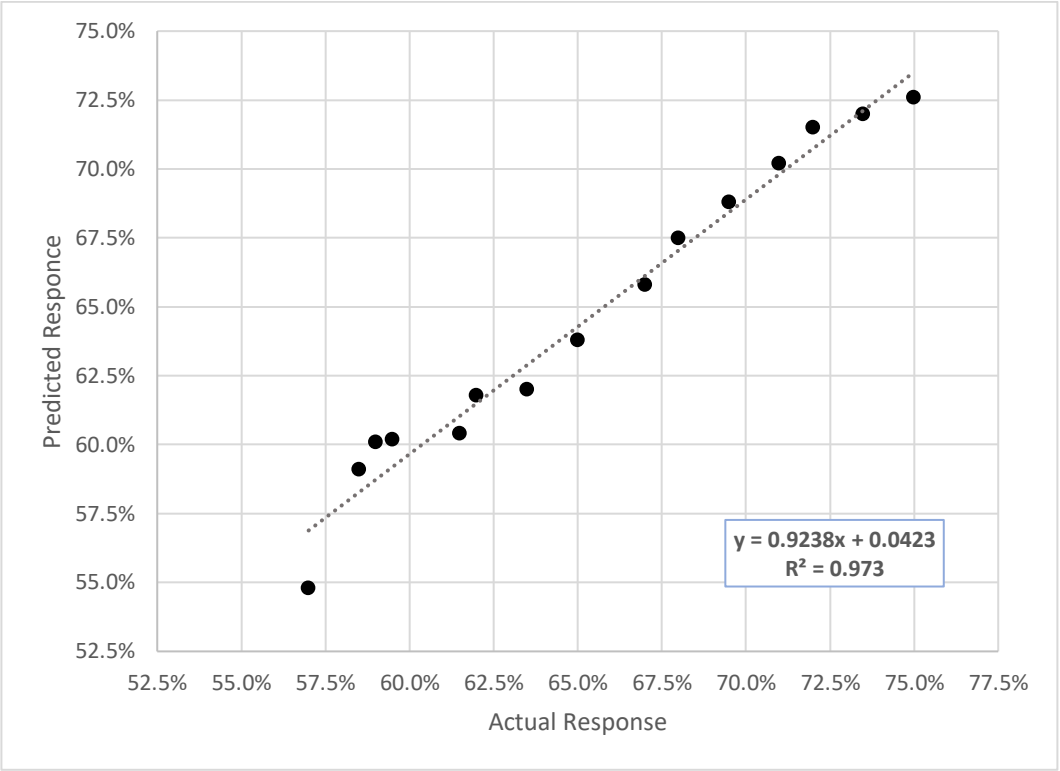


Fig. 6.62 Correlation Graph Between Actual Response and Predicted Response of Humidity Data

6.2.2.2 Trial 2

For the 2nd Trial, the sensor values from both the manual and the digital sensors were gathered 10 times a day on an hourly basis from 8AM to 6 PM. The corresponding data gathered from both the manual sensor and its digital counterpart were used for calculating the percentage error to determine the reliability and accuracy of the sensors being used by the system. The daily average percentage errors were also calculated to easily notice how much the outputs differ from each other per day.

Sensor	Average % Error
Soil Moisture	4.40%
Temperature	4.03%
Humidity	4.61%

Table 6.18 Sensors Summary T

Table 6.19 shows the tabulated values gathered from Day 1 to Day 14. The accumulative % error is 4.40% and is quite low. With this, it can be inferred that the digital soil moisture sensor being used by the system is reliable.

Table 6.20 shows the tabulated values gathered from Day 1 to Day 14. The accumulative % error is 4.03% and is quite low. With this, it can be inferred that the DHT22 Temperature sensor being used by the system is reliable.

Table 6.21 shows the tabulated values gathered from Day 1 to Day 14. The accumulative % error is 4.61% and is quite low. With this, it can be inferred that the DHT22 Humidity sensor being used by the system is reliable.

6.3 Cost Analysis

Based on the pechay ROI_{min} and ROI_{max} calculation, it would take 12.054 to 20.090 years before the return of investment is achieved assuming that the price range would remain the same and that crops would only be pechay. The number of years before the return of investment is quite high but this is only because it was calculated using the price range of a relatively cheap variety of vegetable.

Based on the arugula ROI calculation, it would take only 2.109 years before the return of investment is achieved assuming that the price would remain the same and that crops grown would only be Arugula. There is a 10 times increase in speed of ROI and a 570% increase in profit in comparison between Pechay and Arugula. The number of years before the return of investment became low which shows that if the right crop is selected, the project would be beneficial for its potential owner.

This can also be utilized in times where the outside environment is no longer safe or ideal for the growing of different agriculture. A good example would be in this

pandemic where mobility of each person is limited and access to groceries and supermarkets are regulated, this is when this project can be beneficial for the majority of people residing in urbanized areas.

6.4 Summary

Given the objectives and the testing, the group was able to achieve the results by performing multiple trials and comparing the actual and theoretical values. Values from different data from testing were tabulated to further conclude the outcome. In addition, statistical analysis was also performed to achieve accurate values. The group was able to perform the seeding process within the +-3cm distance from the target. It also provided an option for the different modes for the plant box. All these can be monitored through the application that also gave the user an option to manually control the devices for testing purposes. Lastly, a breakdown of all the materials used was calculated to monitor the overall expense.

Chapter 7 CONCLUSIONS, RECOMMENDATIONS, AND FUTURE DIRECTIVES

Contents

7.1 Concluding Remarks 197
7.2 Contributions 197
7.3 Recommendations 199
7.4 Future Prospects 199

7.1 Concluding Remarks

After several considerations as to how each module would integrate as a whole system, the researchers were able to meet their general and specific objectives. The team was able to carry out the main goal which was to perform the three basic gardening operations – seeding, watering, and monitoring. The group was able to meet the first objective by developing a Raspberry Pi based system capable of monitoring through the system through mobile application and controlling the cartesian robot that provided uniform and variable seed distribution. For the second objective, the group was able to develop the GrowSmart mobile application that monitored the soil moisture, temperature, humidity, and plant growth. For the third objective, GrowSmart was able to achieve an adaptive automatic irrigation system based on time and its moisture content. For the fourth objective, the group was able to determine from results that average percentage errors were relatively low, inferring that the digital sensors being used were reliable and accurate. For the fifth objective, target accuracy was achieved by testing the cartesian robot going to targets that fitted within the 10 circles which were 6 cm in diameter. Lastly, the cost analysis conducted showed that choosing the right crop multiplies the speed of ROI and greatly increase the profit by taking into account the market value of each crop, how relatively easy some are to grow and how long until they can be sold into the market.

7.2 Contributions

The interrelated contributions and supplements that have been developed in this thesis are listed as follows.

Ryan Jonathan Albaladejo

- In charge of the movement of XYZ robot
- Annotated most data for the final paper
- Programmed code for the Arduino motor movements
- Helped in data gathering for documentation
- Made sure that wire connections were safe and secure
- Provided statistical analysis for the data recorded

Danielle Marie Espiritu

- Finalized the design on how both software and hardware will be integrated
- Developed the Django based application (front-end and back-end)
- Developed the Python code that controls the entire system using Raspberry Pi
- Developed the connectivity system of GrowSmart
- Responsible for Raspberry Pi and Django server integration
- Responsible for software and hardware integration of the entire system
- Responsible for the wiring of electronic modules required to create the entire system
- Made sure that the proposed system was working efficiently

Marien Allison Go

- Responsible for the structure of the growth chamber
- Provided materials for the plant box
- Documented the processes for the final paper
- Programmed code using Python for image processing

- Helped in collecting data for tabulation and for testing
- Helped in designing the website

7.3 Recommendations

The researchers propose an addition of a pH sensor to monitor the health condition of the soil. This is because the pH level could highly influence the growth of each crop. Also, future researchers should consider using a camera with a higher resolution and a different approach for plant detection since the method used for this research covered only photos taken from the top view. Another good option for multiple usage long term is by mounting the cartesian robot to a steel frame. This is to be applied so that the plant box can easily be replaced without the need to unmount the robot and other devices. A replacement of slimmer aluminum extrusion for the z axis would be a good alternative to lessen the overall weight. Along with the replacement is the reconsideration of the design for the cartesian robot. This should be modified so that the z axis can reach the edges of the box, thus maximizing the space. Adjustable height for the grow lights is also ideal to cater to the different distances needed of the plant from the light. A nice addition to the mobile application is to provide an option for the user to input the desired type of crop, ideal temperature, and humidity. Lastly, oscillating fans could be used as this affects the overall air circulation of the grow room.

7.4 Future Prospects

The researchers suggest that this system can also be used for:

1. Vertical Farming
2. Hydroponics

REFERENCES

Abdullah, A., Enazi, S. A., & Damaj, I. (2016). *AgriSys: A smart and ubiquitous controlled-environment agriculture system*. 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC). doi:10.1109/icbdsc.2016.7460386

Abhinav Rajpal, Sumit Jain, Nistha Khare and Anil Kumar Shukla. *Microcontroller based Automatic Irrigation System with Moisture Sensors*. Proc. of the International Conference on Science and Engineering. ICSE 2011.

Acero, L. (2013). *Growth Response of Brassica Rapa on the Different Wavelength of Light.* International Journal of Chemical Engineering and Applications. 415-418. 10.7763/IJCEA.2013.V4.337.

Albert, S. (2019, December 30). *How to Grow Arugula*. Retrieved from https://harvesttotable.com/how-to-grow-arugula/

Amrita Sneha.A, Abirami.E, Ankita.A, R. Praveena and R. Srimeena, *Agricultural Robot for automatic ploughing and seeding*, 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR)

Amritanshu Srivastava, Shubham Vijay, Alka Negi, Akash Singh, *DTMF Based Intelligent Farming Robotic Vehicle*, International Conference on Embedded Systems (ICES), IEEE 2014.

Applied Horticultural Research Pty Ltd (2018) *Open Field Production. Crop group: Brassica (cabbage family) – cabbage, broccoli, cauliflower*. http://ahr.com.au/wp-content/uploads/2018/02/Philippines-Factsheet_Pechay_Field.pdf

Bueno, A. (2017, September 28). *Can urban gardening flourish in Manila?* Retrieved from http://cnnphilippines.com/life/leisure/food/2017/09/28/urban-farming-gardeners-manila.html

Balayo, S. (n.d.). Growth and Yield Performance of the Different Varieties of Pechay (Brassica chinensis) as Affected by the Different Organic Fertilizers under MSU-Marawi Condition. International Journal of Humanities and Social Sciences.

Chen, H. (2012) *Design of a Controlled Environment Agricultural Plant Inspection Robot.* MS) thesis, University of Iowa, 2012.

Cocetta, G., Casciani, D., Bulgari, R. et al. *Light use efficiency for vegetables production in protected and indoor environments.* Eur. Phys. J. Plus 132, 43 (2017).

Cong Ming, *in Ligang and Fag Bo Intelligent robot Mowers: A review*, Robot, Vol. 29, no 4, 2007, in press

Correll, N., Arechiga, N., Bolger, A., Bollini, M., Charrow, B., Clayton, A., Rus, D. (2010). *Indoor robot gardening: design and implementation. Intelligent Service Robotics*, 3(4), 219232.

De Guzman, S. S. (2018, June 17). *Agriculture is dying in the Philippines*. Retrieved from https://www.philstar.com/opinion/2018/06/18/1825542/agriculture-dying-philippines

Department of Agriculture (2020) *Retail Prices of Selected Agri-fishery Commodities in Selected Markets in Metro Manila*. Retrieved from https://www.da.gov.ph/wp-content/uploads/2020/06/Price-Monitoring-June-18-2020.pdf

DIFFERENT RED: BLUE RATIO. (n.d.). Led-Lights. https://www.led-lights.com/different-red-blue-ratio-a-140

Dubey, V. Dubey, N., Chouhan, S.S. *Wireless Sensor Network Based Remote Irrigation Control System and Automation Using DTMF Code*, IEEE Conference proceedings Communication Systems and Network Technologies (CSNT), pp- 34 - 37, 2011.

Espiritu, K., & Nielsen, L. (2020, January 24). *Grow Room Ventilation 101*. Epic Gardening. https://www.epicgardening.com/grow-room-ventilation-101/

Fernandez, A., Miñoza, E. (2015). *Growth and Yield of Pechay (Brassica Pekinesis) as Affected by Green Herds Organic Based Foliar Fertilizer*. Horticulturae

Gollakota, A., Srinivas, M. B. (2011). *Agribot A multipurpose agricultural robot*. 2011 Annual IEEE India Conference.

Gonzales, L., Caralde, R. & Aban, M. (2015). *Response of Pechay (Brassica napus L.) to Different Levels of Compost Fertilizer*. International Journal of Scientific and Research Publications,, 5(2), 2250-3153.

Green, J. (n.d.). *Can I Use Loam to Fill My Raised Vegetable Bed?* SFGATE. Retrieved from https://homeguides.sfgate.com/can-use-loam-fill-raised-vegetable-bed-93292.html#:~:text=Most%20garden%20soils%20are%20either,oxygen%20and%20nutrients%20for%20plants.

Gulam Amer, S.M.M. Mudassir, M.A. Malik, Design and operation of Wi-Fi *Agribot Integrated system,* International Conference on Industrial Instrumentation and control (ICIC), IEEE, 2015

Harrington, J. (2016, October 07). *How Many Hours a Day Are Needed for Indoor Plant Grow Lights?* Retrieved from https://homeguides.sfgate.com/many-hours-day-needed-indoor-plant-grow-lights-100245.html

HTG Supply. (2017, October 27). *Ventilation & Circulation In The Grow Space.* https://www.htgsupply.com/informationcenter/talking-shop/grow-room-air-circulation-ventilation/

Hyland, L. (2020, May 05). *How Far Should LED Grow Lights Be From Plants? Learn How High To Hang A Grow Light.* Urban Organic Yield. Retrieved from https://urbanorganicyield.com/grow-light-distance-from-plant/urbanorganicyield.com/grow-light-distance-from-plant/

How to Grow Arugula. (2015, December 02). Retrieved from https://www.diynetwork.com/how-to/outdoors/gardening/how-to-grow-arugula

Jimenez, E. F., Mariano, J. S., Ferrer, M. J., De Leon, V. (2000). *Pechay Production Guide.* Retrieved from http://bpi.da.gov.ph/bpi/images/Production_guide/pdf/PECHAY%20.pdf

Kurtz, L. (2020, April 29). *How to Plant Kangkong.* Retrieved from https://www.wikihow.com/Plant-Kangkong

Limprasitwong, P.,Thongchaisuratkrul, C. *Plant Growth Using Automatic Control System under LED, Grow, and Natural Light.* 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), Krabi, 2018, pp. 192-195.

Lin, H., Rui-Qi, M., Shi-Gang, C., Yong-Li, Z., & Xmg-Li, W. (2018). *Design of Intelligent Plant Growth Cabinet Environment Monitoring and Control System.* 2018 Chinese Control And Decision Conference (CCDC).

Max. (2019, January 5). *Grow Room Ventilation Systems - A Beginner's Guide.* Green and Vibrant. Retrieved from https://www.greenandvibrant.com/grow-room-ventilation

Melchor, L. (2020, March 09). *How and When to Harvest Arugula.* Retrieved from https://gardenerspath.com/plants/vegetables/harvest-arugula/

Organic Arugula (250 grams). (n.d.). Retrieved from https://thegreengrocermanila.com/product/organic-arugula/

Organic Water Spinach (250 grams). (n.d.). Retrieved from https://thegreengrocermanila.com/product/organic-water-spinach-250g/

Overby, A. (2010). *CNC Machining Handbook*. New York: McGraw-Hill.

Padalalu, P., Mahajan, S., Dabir, K., Mitkar, S., & Javale, D. (2017). *Smart Water Dripping System for Agriculture/Farming*. 2017 2nd International Conference for Convergence in Technology (I2CT).

Philippine Statistics Authority (2018) *Philippines Retail Price: Pechay, Native: CAR: Baguio City.* Retrieved from https://www.ceicdata.com/en/philippines/retail-price-selected-agricultural-commodities/retail-price-pechay-native-car-baguio-city#:~:text=Philippines's%20Retail%20Price%3A%20Pechay%2C%20Native,PHP%2Fkg%20for%20Dec%202018

Schumacher, H. J. (2017, June 11). *One of the top 10 urban innovations: Urban farming*. Retrieved from https://businessmirror.com.ph/one-of-the-top-10-urban-innovations-urban- farming/

Smsaleh. (2017, October 20). *How Long to Keep Fans on in Grow Room?* What For Me. https://whatforme.com/how-long-to-keep-fans-on-in-grow-room

Sowjanya, K. D., Sindhu, R., Parijatham, M., Srikanth, K., Bhargav, P. (2017). *Multipurpose autonomous agricultural robot.* 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA).

Ultra, V., Sotto, J., Punzalan M. (2018). *Nutrient Supplying Potential of Different Spent Mushroom Substrate Preparations as Soil Amendment in a Potting Media.* IOP Conference Series: Earth and Environmental Science. 120. 012010. 10.1088/1755-1315/120/1/012010.

Watkins, D. (2018, December 15). *The Best Plant Light Spectrum for Growing Flowering Plants.* Retrieved from https://homeguides.sfgate.com/plant-light-spectrum-growing-flowering-plants-72801.html

Weller, C. (2016, August 04). *Manila is the most crowded city in the world - here's what life is like.* Retrieved from https://www.businessinsider.com/manila-worlds-most-crowded- city-2016-8

Williams, D. (n.d.). *List of Reflective Materials for Indoor Growing*. SFGATE. Retrieved from https://homeguides.sfgate.com/list-reflective-materials-indoor-growing-92288.html

Woody, S., & Amasino, R. (2011). Instructions and Tips for Growing Rapid-cycling Brassica rapa variety "FPsc." University of Wisconsin-Madison. https://fpsc.wisc.edu/growguide/Instructions_and_Tips_for_Growing_FPsc.pdf

Appendix A ANSWERS TO QUESTIONS TO THIS THESIS

A1 How important is the problem to practice?

The availability of land and resources for agriculture is depleting due to urbanization and an increase in population. Agricultural crops are also susceptible to pests and diseases. In addition to that pest infestation and diseases have increased in intensity along with the unpredictable rise of infection by certain chemical resistant pathogens and pests. The rise is due to climate-induced pests or diseases, inappropriate use of chemical pesticides, fungicides, and other agents. The availability of land is also under severe exploitation due to the rise in the global population, urbanization and rapid industrial growth. These above factors are becoming a serious threat to the usage of lands in conventional agricultural practices and will result in a shortage of food supply in urban areas. One of the possible solutions is to grow crops in an indoor facility with tightly packed spaces which increases crop density.

A2 How will you know if the solution/s that you will achieve would be better than existing ones?

The existing project was implemented outdoor. This can be affected due to different weather conditions. With the concept of having an indoor garden, this eliminates the worry about it getting excessively dry and wet. Another concept introduced is its smart automated system that also gives the user a chance to monitor the mobile application.

A2.1 How will you measure the improvement/s?

This will be measured by testing its efficiency, convenience, and the growth of the vegetable crop.

A2.1.1 What is/are your basis/bases for the improvement/s?

Proposing an indoor environment without giving the same effect as compared to the outdoor planting will be useless. This can be a solution for some countries with extreme weather conditions and limited daylight hours.

A2.1.2 Why did you choose that/those basis/bases?

The researchers chose this as the group thinks that having an automated indoor gardening can be a solution and answer to sustainability as the population rapidly increases. The bases chosen by the group explicitly show the differences between the existing labor-intensive methods done by local farmers.

A2.1.3 How significant are your measure/s of the improvement/s?

This was to check whether or not the act of automating the gardening process will lessen the burden to people.

A3 What is the difference of the solution/s from existing ones?

This system is implemented in an enclosed chamber with a controlled environment. As compared to the previous projects, those mainly focused on the cartesian robot with agricultural applications.

A3.1 How is it different from previous and existing ones?

The previous ones only either have automated the gardening processes or a gardening system that was grown in a controlled environment.

A4 What are the assumptions made (that are behind for your proposed solution to work)?

The researchers always assume that the soil in the plantbox is always sowed and ready for planting. Second is internet, this is because the whole system can only be programmed and controlled using the web application, meaning an active internet connection is required to send and receive data.

A4.1 Will your proposed solution/s be sensitive to these assumptions?

The proposed solutions will be sensitive to the assumptions since they are factors for the prototype to work, plant growth and its conditions.

A4.2 Can your proposed solution/s be applied to more general cases when some of the assumptions are eliminated? If so, how?

No, since the prototype specifically focuses on basic gardening operations which are seeding, watering, and monitoring.

A5 What is the necessity of your approach / proposed solution/s?

The researchers proposed solution is necessary so that more vegetable crop will be harvested with the less hassle. At the same time, it gives people living in the city a chance to grow plants indoors. The current method being used, manual planting, is a tedious job which is mostly prominent in rural areas. Since urbanization has evolved, proposing this solution can eliminate the problem of scarcity.

A5.1 What will be the limits of applicability of your proposed solution/s?

This only covers crops grown with seed. Along with that, only crops with the same ideal conditions can be planted at a time. This is because the controlled environment has limited applications.

A5.2 What will be the message of the proposed solution to technical people? How about to non-technical managers and business men?

Technology can go beyond our imagination. The message to the technical people is to say that is it possible for a technology to integrate multiple concepts and merged all into one remote-controlled garden. For the non-technical people, the message will be saying that incorporating a small garden at your own home is possible and feasible.

A6 How will you know if your proposed solution/s is/are correct?

Data collection through multiple trials should be made to see if the results are accurate.

A6.1 Will your results warrant the level of mathematics used (i.e., will the end justify the means)?

Yes, by comparing the objectives to the results.

A7 Is/are there an/ alternative way/s to get to the same solution/s?

There are no alternative ways to get the same solution.

A7.1 Can you come up with illustrating examples, or even better, counter examples to your proposed solution/s?

A good improvement for the cartesian robot is by using steel frames for the plant box. By using steel frames, it applies a concept similar to a drawer frame which is better to use so that the plant box can easily be replaced without unmounting everything else.

A7.2 Is there an approximation that can arrive at the essentially the same proposed solution/s more easily?

Better processing of images can be done for better results.

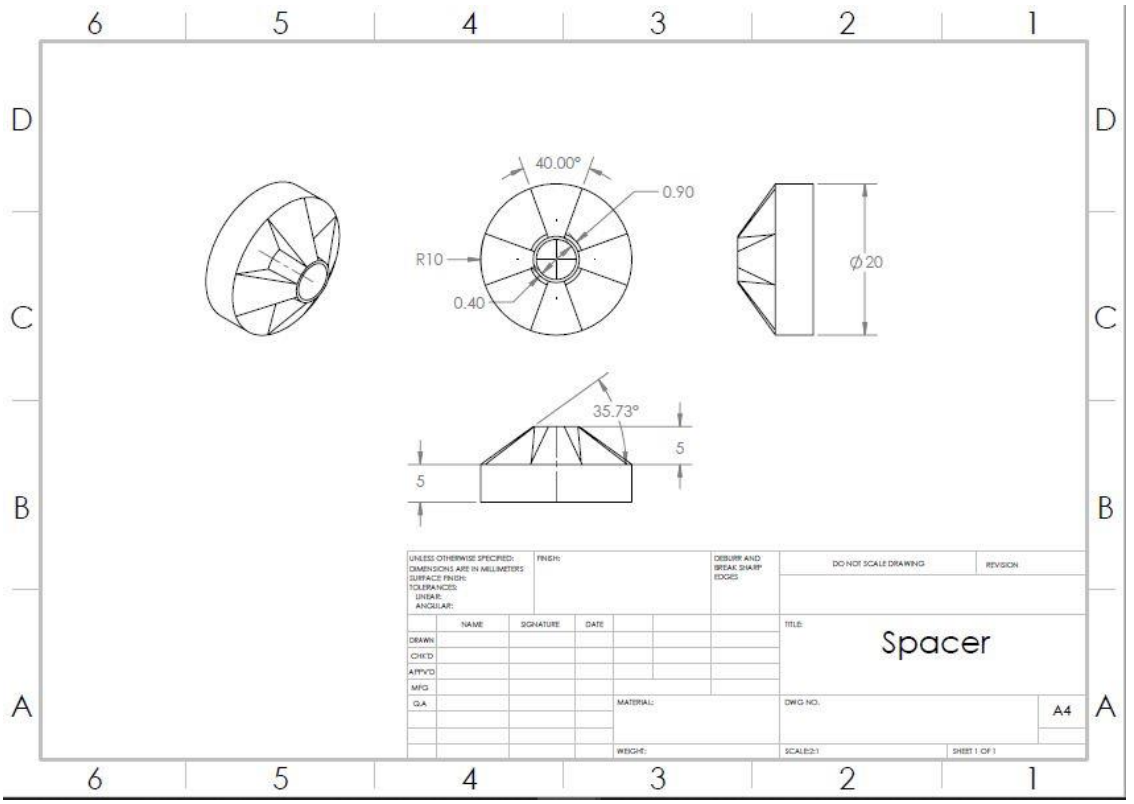
A8 If you were the examiner of your Thesis, how would you present the Thesis in another way? Give your remarks, especially for your methodology and the results and discussions.

The researchers would present this as an innovative way to grow plants for research and for commercial use. This can be used for precise growing experiments with multiple variations. Also, in your own home, this can be done without the need to take care of it most of the time. Through this, it saves labor costs while improving efficiency. Imagine having your garden with barely doing anything is something technology could do for you. This is a project that can run at all hours of the day, non-stop, to ensure your plants are taken care of in the most optimal way.

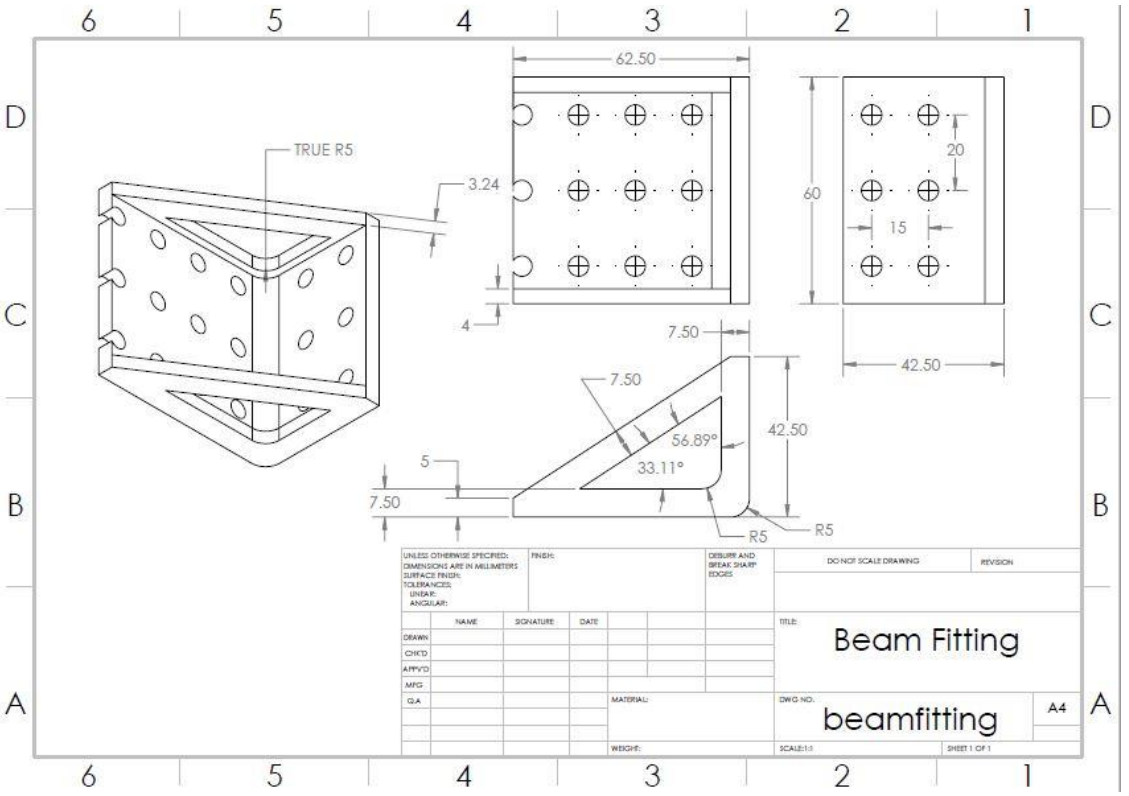
A8.1 What are the weaknesses of your Thesis, specifically your methodology and the results and discussions?

Some weaknesses that present a major factor in the overall functionality is stable Wi-Fi and occasional power outage. Another is the growth of the plant with overlapping leaves. Due to camera only capturing images from a 2D view, image processing is not completely accurate thus resulting to a value far from the expected values.

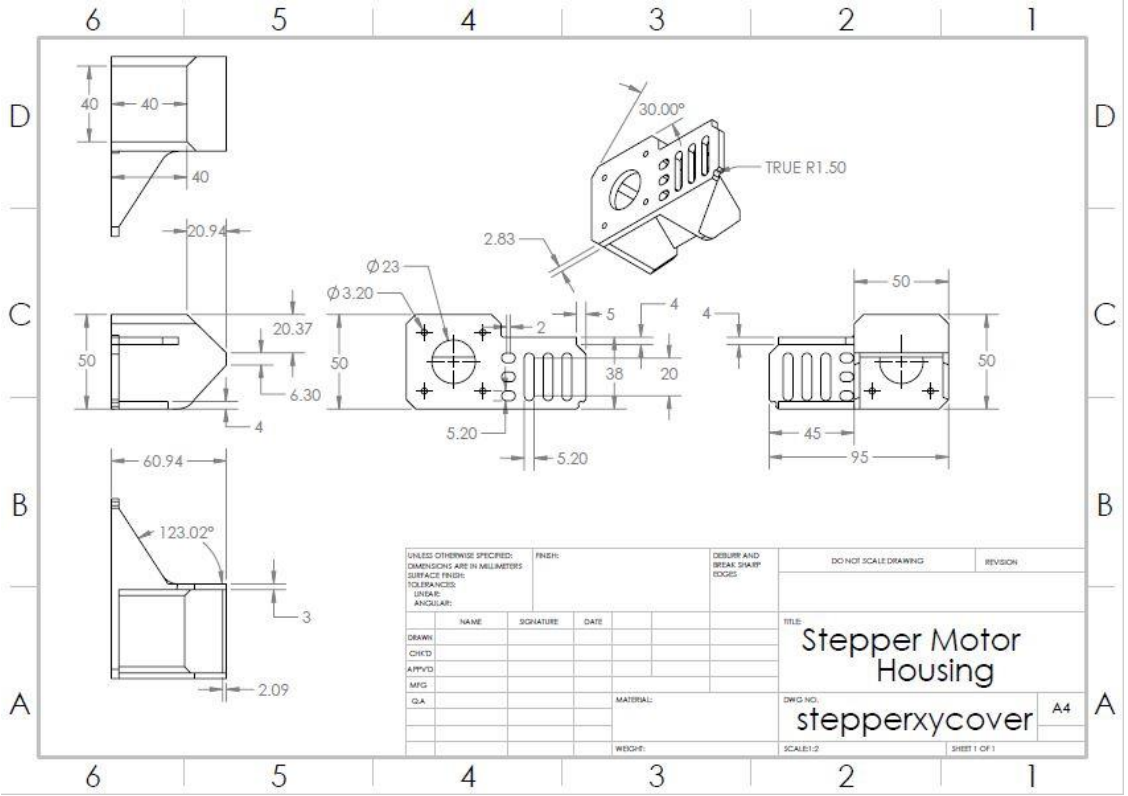
Appendix B SPACER



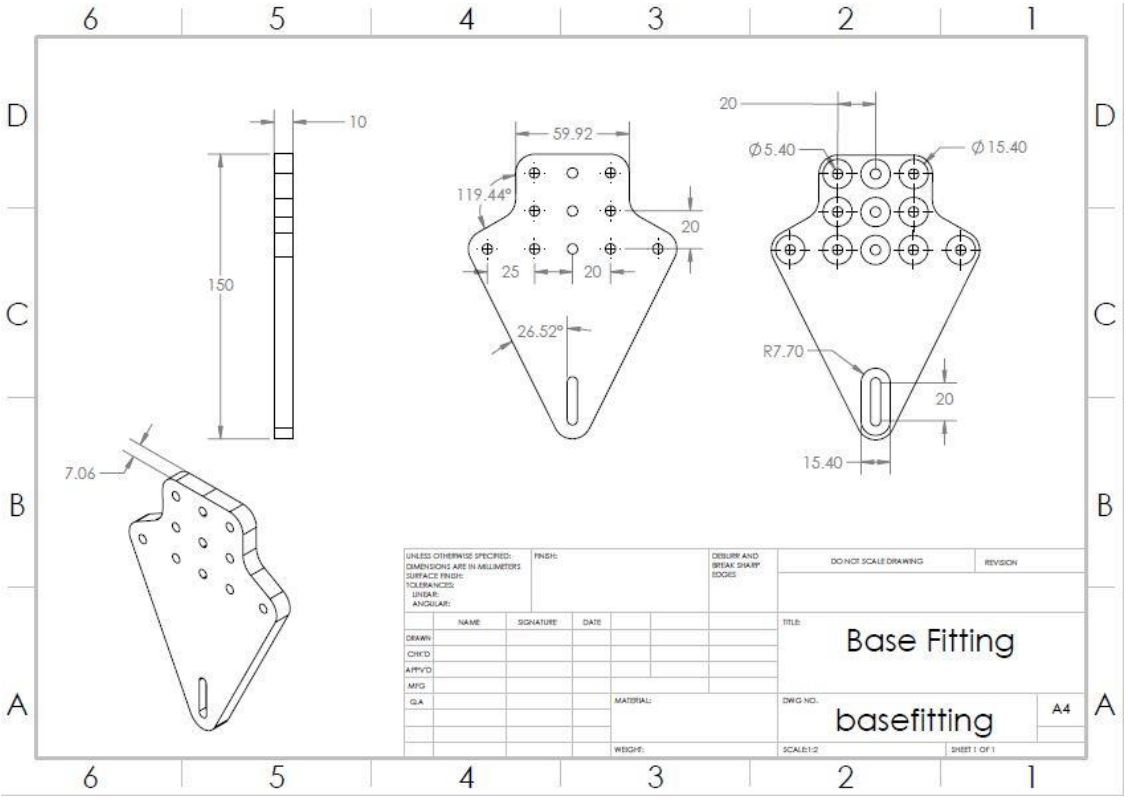
Appendix C BEAM FITTING



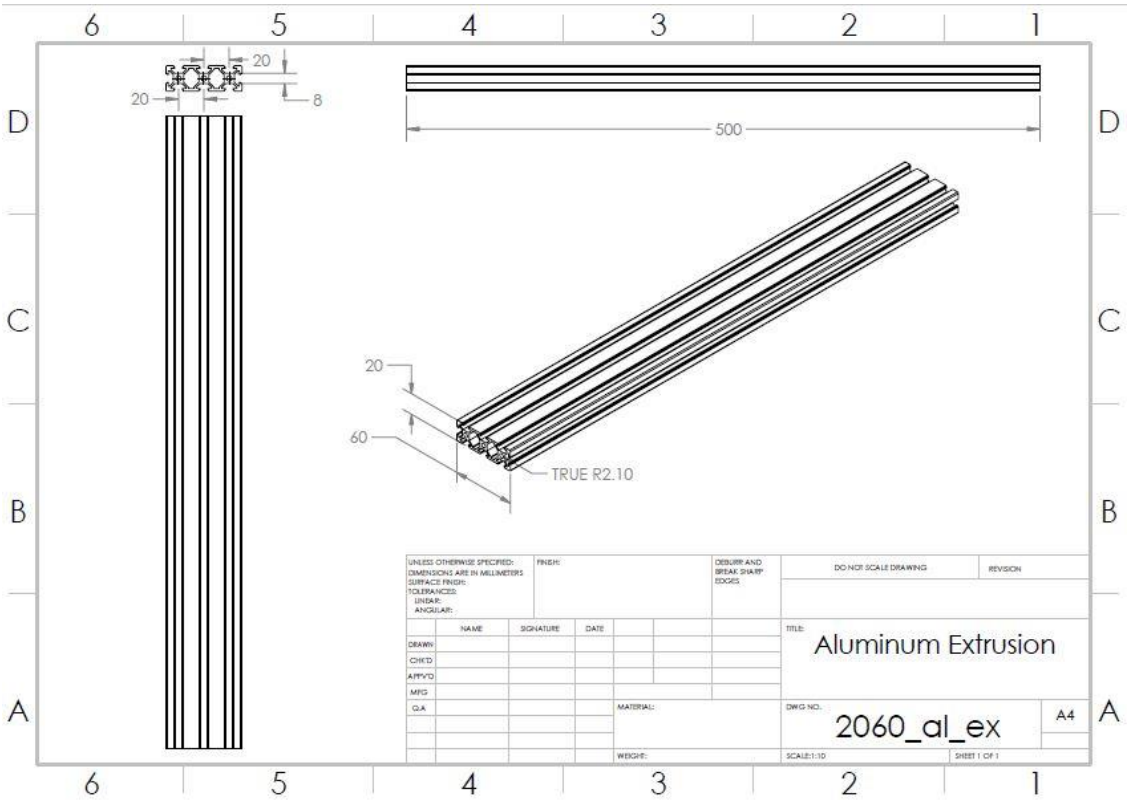
Appendix D STEPPER MOTOR HOUSING



Appendix E BASE FITTING



Appendix F ALUMINUM EXTRUSION



Appendix G DJANGO VIEWS CODE

```
from django.shortcuts import render
from django.http import HttpResponse
from django.http import JsonResponse
from .models import mode_selected
from .models import devices
from .models import sensors
from .models import threshold
from .models import mode1
from .models import mode2
from .models import mode3
from .models import mode4
from .models import mode1_vision_system
from .models import mode2_vision_system
from .models import mode3_vision_system
from .models import mode4_vision_system

from pygame.locals import *
from datetime import datetime
from datetime import date
from numpy import interp  # To scale values
from time import sleep  # To add delay
from plantcv import plantcv as pcv

import os
import sys
import RPi.GPIO as GPIO
#Soil Moisture Sensor
import spidev # To communicate with SPI devices
#Camera
import pygame
import pygame.camera
#DHT22
import Adafruit_DHT
DHT_SENSOR = Adafruit_DHT.DHT22
DHT_SENSOR2 = Adafruit_DHT.DHT22
#Image Processing
import numpy as np
import cv2
import re

GPIO.setmode(GPIO.BCM) #Read GPIO# and not pin #!
#RIGHT TERMINAL
GPIO.setup(21, GPIO.OUT)  # White LED, PIN 40 (Right)
GPIO.setup(20, GPIO.OUT)  # Fan1, PIN 38 (Right)
GPIO.setup(16, GPIO.OUT)  # Fan2, PIN 36 (Right)
GPIO.setup(12, GPIO.OUT)  # RGB LED, PIN 12 (Right)
```

```
#LEFT TERMINAL
GPIO.setup(26, GPIO.OUT)  # CalibrationXYZ, PIN 37 (Left)

GPIO.setup(19, GPIO.OUT)  # WaterXYZ, PIN 35 (Left)
GPIO.setup(13, GPIO.OUT)  # SeederXYZ, PIN 33 (Left)

GPIO.setup(6, GPIO.OUT)  # Mode_1, PIN 31 (Left)
GPIO.setup(5, GPIO.OUT)  # Mode_2, PIN 29 (Left)
GPIO.setup(0, GPIO.OUT)  # GrowLights, PIN 27 (Left)

DHT_PIN = 1 # PIN 28 (Right)
DHT_PIN2 = 7 # PIN 26 (Right)


def mainPage(response):

    mode_selected_obj_global = mode_selected.objects.latest('date')
    devices_obj_global = devices.objects.latest('date')

    mode1_obj_global = mode1.objects.latest('date')
    mode2_obj_global = mode2.objects.latest('date')
    mode3_obj_global = mode3.objects.latest('date')
    mode4_obj_global = mode4.objects.latest('date')

    # Create instances so you can insert into the database
    mode_selected_ = mode_selected()
    devices_ = devices()
    devices_2 = devices()
    sensors_ = sensors()
    threshold_ = threshold()

    mode1_ = mode1()
    mode2_ = mode2()
    mode3_ = mode3()
    mode4_ = mode4()

    mode1_vision_system_ = mode1_vision_system()
    mode2_vision_system_ = mode2_vision_system()
    mode3_vision_system_ = mode3_vision_system()
    mode4_vision_system_ = mode4_vision_system()

    if response.POST.get('action') == 'setup':

        print(" ")
        print("--------------------------- GrowSmart Initializing --
-------------------------")
        print("Time:     "     +     datetime.now().strftime('%Y-%m-%d
%H:%M:%S'))
```

```
        print("Mode: " + str(mode_selected_obj_global.modeNumber))
        print("Grid: " + mode_selected_obj_global.grid)
        print(" ")
        print(" ")

        json = {
        'modeNumber' :mode_selected_obj_global.modeNumber
        }
        return JsonResponse(json)


    if response.POST.get('action') == 'getSensorValues':
        print(" ")
        print("------------------------- Checking Sensor Readings -
-------------------------")
        print("Time:     "     +     datetime.now().strftime('%Y-%m-%d
%H:%M:%S'))

        # Start SPI connection
        spi = spidev.SpiDev() # Created an object
        spi.open(0,0)

        humidity,  temperature  =  Adafruit_DHT.read_retry(DHT_SENSOR,
DHT_PIN)
        humidity2,                   temperature2                   =
Adafruit_DHT.read_retry(DHT_SENSOR2, DHT_PIN2)

        def analogInput(channel):
          spi.max_speed_hz = 1350000
          adc = spi.xfer2([1,(8+channel)<<4,0])
          data = ((adc[1]&3) << 8) + adc[2]
          return data

        output = analogInput(0) # Reading from CH0
        output = interp(output, [0, 1023], [100, 0])
        output = int(output)
        #print("Moistures", output)

        currentMoisture = output
        averageTemperature = (temperature + temperature2) / 2
        averageHumidity = (humidity + humidity2) / 2

        temperatureStatus = 'good'
        humidityStatus = 'good'
        soilMoistureStatus = 'good'

        temperatureStatusSummary = "Default"
        humidityStatusSummary = "Default"
        soilMoistureStatusSummary = "Default"
```

```
        if(averageTemperature                                        <=
threshold_obj_global.temperature_low):
            temperatureStatus = 'low' # Too Low
        elif(averageTemperature                                      >=
threshold_obj_global.temperature_high):
            temperatureStatus = 'high' # Too High
        else:
            temperatureStatus = 'good' # Good

        if (averageHumidity <= threshold_obj_global.humidity_low):
            humidityStatus = 'low' # Too Low
        elif (averageHumidity >= threshold_obj_global.humidity_high):
            humidityStatus = 'high' # Too High
        else:
            humidityStatus = 'good' # Good

        if (currentMoisture <= threshold_obj_global.moisture_dry):
            soilMoistureStatus = 'dry'; # Dry
        elif  (currentMoisture  >=  threshold_obj_global.moisture_dry
and currentMoisture <= threshold_obj_global.moisture_wet):
            soilMoistureStatus = 'moist'; # Moist
        elif (currentMoisture >= threshold_obj_global.moisture_wet):
            soilMoistureStatus = 'wet'; # Wet

        if(temperatureStatus == 'high'):
            temperatureStatusSummary = 'Too High!'
        elif (temperatureStatus == 'low'):
            temperatureStatusSummary = 'Too Low!'
        else:
            temperatureStatusSummary = 'Good'

        if (humidityStatus == 'high'):
            humidityStatusSummary = 'Too High!'
        elif (humidityStatus == 'low'):
            humidityStatusSummary = 'Too Low!'
        else:
            humidityStatusSummary = 'Good'

        print("Soil Moisture: "+ str(currentMoisture))
        print("Temperature Left: " + str(temperature))
        print("Temperature Right: "+ str(temperature2))
        print("Average Temperature: "+ str(round(averageTemperature,
2)))
        print("Humidity Left: "+ str(humidity))
        print("Humidity Rght: "+ str(humidity2))
        print("Average Humidity: "+ str(round(averageHumidity, 0)))

        # Adaptive Irrigation System Code
```

```
        if (soilMoistureStatus == 'dry'):
            soilMoistureStatus = 'Dry!'
            print(" ")
            print("~ Soil moisture is dry, adaptive irrigation system
activated~")
            print(" ")
            devices_.fansStatus = devices_obj_global.fansStatus
            devices_.whiteLedStatus                                =
devices_obj_global.whiteLedStatus
            devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
            devices_.calibrationStatus                             =
devices_obj_global.calibrationStatus
            devices_.waterStatus = 'On'
            devices_.seedStatus = devices_obj_global.seedStatus
            devices_.save()
            GPIO.output(19, GPIO.HIGH)
            sleep(1)
            GPIO.output(19, GPIO.LOW)
            print(" ")
            print("~Adaptive irrigation system deactivated~")
            print(" ")
            devices_2.fansStatus = devices_obj_global.fansStatus
            devices_2.whiteLedStatus                               =
devices_obj_global.whiteLedStatus
            devices_2.rgbLedStatus = devices_obj_global.rgbLedStatus
            devices_2.calibrationStatus                            =
devices_obj_global.calibrationStatus
            devices_2.waterStatus = 'Off'
            devices_2.seedStatus = devices_obj_global.seedStatus
            devices_2.save()

        elif (soilMoistureStatus == 'moist'):
            soilMoistureStatus = 'Moist'
        elif (soilMoistureStatus == 'wet'):
            soilMoistureStatus = 'Wet!'

        if(temperatureStatus == 'low'):
            print(" ")
            print("~Temperatures is low, fans deactivated~")
            print(" ")
            GPIO.output(20, GPIO.LOW)
            GPIO.output(16, GPIO.LOW)
            devices_.fansStatus = 'Off'
            devices_.whiteLedStatus                                =
devices_obj_global.whiteLedStatus
            devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
            devices_.calibrationStatus                             =
devices_obj_global.calibrationStatus
            devices_.waterStatus = devices_obj_global.waterStatus
```

```
            devices_.seedStatus = devices_obj_global.seedStatus
            devices_.save()
        elif(temperatureStatus == 'high'):
            print(" ")
            print("~Temperatures is high, fans activated~")
            print(" ")
            GPIO.output(20, GPIO.HIGH)
            GPIO.output(16, GPIO.HIGH)
            devices_.fansStatus = 'On'
            devices_.whiteLedStatus                                =
devices_obj_global.whiteLedStatus
            devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
            devices_.calibrationStatus                             =
devices_obj_global.calibrationStatus
            devices_.waterStatus = devices_obj_global.waterStatus
            devices_.seedStatus = devices_obj_global.seedStatus
            devices_.save()
        elif(temperatureStatus  !=  'low'  and  humidityStatus  ==
'high'):
            print(" ")
            print("~Temperatures is not low and humidity is high, fans
activated~")
            print(" ")
            GPIO.output(20, GPIO.HIGH)
            GPIO.output(16, GPIO.HIGH)
            devices_.fansStatus = 'On'
            devices_.whiteLedStatus                                =
devices_obj_global.whiteLedStatus
            devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
            devices_.calibrationStatus                             =
devices_obj_global.calibrationStatus
            devices_.waterStatus = devices_obj_global.waterStatus
            devices_.seedStatus = devices_obj_global.seedStatus
            devices_.save()

        sensors_.temperature = round(averageTemperature, 2)
        sensors_.humidity = round(averageHumidity, 0)
        sensors_.moisture = currentMoisture
        sensors_.temperatureStatus = temperatureStatusSummary
        sensors_.humidityStatus = humidityStatusSummary
        sensors_.soilMoistureStatus = soilMoistureStatus
        sensors_.save()

        sensors_obj = sensors.objects.latest('date')
        mode_selected_obj_first = mode_selected.objects.first()
        mode_selected_obj = mode_selected.objects.latest('date')

        date1 = mode_selected_obj_first.date
        date2 = sensors_obj.date
```

```
        def numOfDays(date1, date2):
            return (date2-date1).days

        mode_selected_.daysCounter = numOfDays(date1, date2) + 1
        mode_selected_.grid = mode_selected_obj.grid
        mode_selected_.rows = mode_selected_obj.rows
        mode_selected_.columns = mode_selected_obj.columns
        mode_selected_.modeNumber = mode_selected_obj.modeNumber
        mode_selected_.image = mode_selected_obj.image
        mode_selected_.save()

        mode_selected_obj_2 = mode_selected.objects.latest('date')

        print("Day "+ str(mode_selected_obj_2.daysCounter))
        print(" ")
        print(" ")

        json = {
        'daysCounter_json' : str(mode_selected_obj_2.daysCounter),
        'date_json': str(datetime.now().strftime('%b. %d, %Y, %-I:%M
%p')),
        'temperature_json': sensors_obj.temperature,
        'humidity_json': sensors_obj.humidity,
        'soilMoisture_json': sensors_obj.moisture,
        'temperatureStatus_json' : sensors_obj.temperatureStatus,
        'humidityStatus_json' : sensors_obj.humidityStatus,
        'soilMoistureStatus_json' : sensors_obj.soilMoistureStatus,
        }

        return JsonResponse(json)

    if response.POST.get('action') == 'snapImage':
        mode_selected_obj = mode_selected.objects.latest('date')
        if(mode_selected_obj.modeNumber == 1):
            print(" ")
            print("~[ Mode 1 ] Vision System Starting~")
            print(" ")
            print(" ")

            #pygame.init()
            pygame.font.init()
            pygame.camera.init()
            cam = pygame.camera.Camera("/dev/video0", (960, 720))
            cam.start()
            image = cam.get_image()
            getTime = datetime.now().strftime('%Y-%m-%d-%H:%M:%S')
```

```
            pygame.image.save(image,
'/home/pi/Desktop/thesis/thesis/assets/gardenPics/'   +   getTime   +
'.jpg')
            cam.stop()

            class options:
                def __init__(self):
                    self.debug = "plot"
                    self.outdir = "./assets/gardenPics/"


            args = options()
            #pcv.params.debug = args.debug

            plant_area_list = [] #Plant area array for storage

            img,             path,             filename            =
pcv.readimage(filename='./assets/gardenPics/'  +  getTime  +  '.jpg',
mode="native") # Read image to be used
            #img,    path,    filename    =    pcv.readimage(filename=
'./assets/gardenPics/test.jpg', mode="native") # Read image to be used

            #     START     of          Multi     Plant     Workflow
https://plantcv.readthedocs.io/en/stable/multi-plant_tutorial/

            # STEP 1: Check if this is a night image
            # STEP 2: Normalize the white color so you can later
            img1 = pcv.white_balance(img, roi = (600,70,20,20))
            # STEP 3: Rotate the image so that plants line up with
grid
            # STEP 4: Shift image
            # STEP 5: Convert image from RGB colorspace to LAB
colorspace Keep only the green-magenta channel (grayscale)
            a = pcv.rgb2gray_lab(rgb_img=img1, channel='a')
            # STEP 6: Set a binary threshold on the saturation channel
image
            img_binary      =      pcv.threshold.binary(gray_img=a,
threshold=119, max_value=255, object_type='dark')
            # STEP 7: Fill in small objects (speckles)
            fill_image = pcv.fill(bin_img=img_binary, size=100)
            # STEP 8: Dilate so that you don't lose leaves (just in
case)
            dilated = pcv.dilate(gray_img=fill_image, ksize=2, i=1)
            # STEP 9: Find objects (contours: black-white boundaries)
            id_objects,  obj_hierarchy  =  pcv.find_objects(img=img1,
mask=dilated)
            # STEP 10: Define region of interest (ROI)
            roi_contour, roi_hierarchy = pcv.roi.rectangle(img=img1,
x=100, y=160, h=390, w=780)
```

```
            # STEP 11: Keep objects that overlap with the ROI
            roi_objects,  roi_obj_hierarchy,  kept_mask,  obj_area  =
pcv.roi_objects(img=img1, roi_contour=roi_contour,

roi_hierarchy=roi_hierarchy,

object_contour=id_objects,

obj_hierarchy=obj_hierarchy,

roi_type='partial')

            # END of Multi Plant Workflow

            #  START  of  Create  Multiple  Regions  of  Interest  (ROI)
https://plantcv.readthedocs.io/en/stable/roi_multi/

            # Make a grid of ROIs
            roi1,     roi_hier1         =     pcv.roi.multi(img=img1,
coord=(180,260), radius=50, spacing=(150, 200), nrows=2, ncols=5)


            # Loop through and filter each plant, record the area
            for i in range(0, len(roi1)):
                roi = roi1[i]
                hierarchy = roi_hier1[i]
                # Find objects
                filtered_contours,               filtered_hierarchy,
filtered_mask, filtered_area = pcv.roi_objects(
                    img=img,   roi_type="partial",   roi_contour=roi,
roi_hierarchy=hierarchy, object_contour=roi_objects,
                    obj_hierarchy=roi_obj_hierarchy)

                # Record the area
                plant_area_list.append(filtered_area)

                if(i<10):
                    print(plant_area_list[i])

            # END of Create Multiple Regions of Interest (ROI)

            # Label area by plant ID, leftmost plant has id=0
            plant_area_labels    =    [i    for    i    in    range(0,
len(plant_area_list))]

            #out = args.outdir
            # Create a new measurement
            pcv.outputs.add_observation(variable='plant_area',
trait='plant area ',
```

```
method='plantcv.plantcv.roi_objects', scale='pixels', datatype=list,
                                        value=plant_area_list,
label=plant_area_labels)

            # Print areas to XML

#pcv.print_results(filename="./assets/gardenPics/plant_area_results.
xml")


            mode1_vision_system_.image  =  '../assets/gardenPics/'  +
getTime + '.jpg'
            mode1_vision_system_.plant1 = plant_area_list[0]
            mode1_vision_system_.plant2 = plant_area_list[1]
            mode1_vision_system_.plant3 = plant_area_list[2]
            mode1_vision_system_.plant4 = plant_area_list[3]
            mode1_vision_system_.plant5 = plant_area_list[4]
            mode1_vision_system_.plant6 = plant_area_list[5]
            mode1_vision_system_.plant7 = plant_area_list[6]
            mode1_vision_system_.plant8 = plant_area_list[7]
            mode1_vision_system_.plant9 = plant_area_list[8]
            mode1_vision_system_.plant10 = plant_area_list[9]

            if(plant_area_list[0] < 721):
                status1 = 'Early'
                image1 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[0] > 720) and (plant_area_list[0] <
861)):
                status1 = 'Dvlping'
                image1 = '../assets/icons/plantIcon.png'
            else:
                status1 = 'Mature'
                image1 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[1] < 721):
                status2 = 'Early'
                image2 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[1] > 720) and (plant_area_list[1] <
861)):
                status2 = 'Dvlping'
                image2 = '../assets/icons/plantIcon.png'
            else:
                status2 = 'Mature'
                image2 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[2] < 721):
                status3 = 'Early'
                image3 = '../assets/icons/seedIcon.png'
```

```
            elif((plant_area_list[2] > 720) and (plant_area_list[2] <
861)):
                status3 = 'Dvlping'
                image3 = '../assets/icons/plantIcon.png'
            else:
                status3 = 'Mature'
                image3 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[3] < 721):
                status4 = 'Early'
                image4 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[3] > 720) and (plant_area_list[3] <
861)):
                status4 = 'Dvlping'
                image4 = '../assets/icons/plantIcon.png'
            else:
                status4 = 'Mature'
                image4 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[4] < 721):
                status5 = 'Early'
                image5 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[4] > 720) and (plant_area_list[0] <
861)):
                status5 = 'Dvlping'
                image5 = '../assets/icons/plantIcon.png'
            else:
                status5 = 'Mature'
                image5 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[5] < 721):
                status6 = 'Early'
                image6 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[5] > 720) and (plant_area_list[5] <
861)):
                status6 = 'Dvlping'
                image6 = '../assets/icons/plantIcon.png'
            else:
                status6 = 'Mature'
                image6 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[6] < 721):
                status7 = 'Early'
                image7 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[6] > 720) and (plant_area_list[6] <
861)):
                status7 = 'Dvlping'
                image7 = '../assets/icons/plantIcon.png'
            else:
```

```
                status7 = 'Mature'
                image7 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[7] < 721):
                status8 = 'Early'
                image8 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[7] > 720) and (plant_area_list[7] <
861)):
                status8 = 'Dvlping'
                image8 = '../assets/icons/plantIcon.png'
            else:
                status8 = 'Mature'
                image8 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[8] < 721):
                status9 = 'Early'
                image9 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[8] > 720) and (plant_area_list[8] <
861)):
                status9 = 'Dvlping'
                image9 = '../assets/icons/plantIcon.png'
            else:
                status9 = 'Mature'
                image9 = '../assets/icons/pechayIcon.png'

            if(plant_area_list[9] < 721):
                status10 = 'Early'
                image10 = '../assets/icons/seedIcon.png'
            elif((plant_area_list[9] > 720) and (plant_area_list[9] <
861)):
                status10 = 'Dvlping'
                image10 = '../assets/icons/plantIcon.png'
            else:
                status10 = 'Mature'
                image10 = '../assets/icons/pechayIcon.png'



            mode1_vision_system_.status1  = status1
            mode1_vision_system_.status2  = status2
            mode1_vision_system_.status3  = status3
            mode1_vision_system_.status4  = status4
            mode1_vision_system_.status5  = status5
            mode1_vision_system_.status6  = status6
            mode1_vision_system_.status7  = status7
            mode1_vision_system_.status8  = status8
            mode1_vision_system_.status9  = status9
            mode1_vision_system_.status10  = status10
```

```
            mode1_vision_system_.image1 = image1
            mode1_vision_system_.image2 = image2
            mode1_vision_system_.image3 = image3
            mode1_vision_system_.image4 = image4
            mode1_vision_system_.image5 = image5
            mode1_vision_system_.image6 = image6
            mode1_vision_system_.image7 = image7
            mode1_vision_system_.image8 = image8
            mode1_vision_system_.image9 = image9
            mode1_vision_system_.image10 = image10

            mode1_vision_system_.save()


            mode1_visionSystem_obj_afterInsertion                  =
mode1_vision_system.objects.latest('date')
            mode_selected_obj_first = mode_selected.objects.first()
            mode_selected_obj = mode_selected.objects.latest('date')

            date1 = mode_selected_obj_first.date
            date2 = mode1_visionSystem_obj_afterInsertion.date

            def numOfDays(date1, date2):
                return (date2-date1).days

            mode_selected_.daysCounter = numOfDays(date1, date2) + 1
            mode_selected_.grid = mode_selected_obj.grid
            mode_selected_.rows = mode_selected_obj.rows
            mode_selected_.columns = mode_selected_obj.columns
            mode_selected_.modeNumber = mode_selected_obj.modeNumber
            mode_selected_.image = '../assets/gardenPics/' + getTime
+ '.jpg'
            mode_selected_.save()

            json = {
            'image_json': str(mode1_vision_system_.image),
            'cameraDateJSON':  str(datetime.now().strftime('%b.  %d,
%Y, %-I:%M %p')),
            'daysCounter_json' : str(numOfDays(date1, date2) + 1),

            'plant1_json': mode1_vision_system_.plant1,
            'plant2_json': mode1_vision_system_.plant2,
            'plant3_json': mode1_vision_system_.plant3,
            'plant4_json': mode1_vision_system_.plant4,
            'plant5_json': mode1_vision_system_.plant5,
            'plant6_json': mode1_vision_system_.plant6,
            'plant7_json': mode1_vision_system_.plant7,
            'plant8_json': mode1_vision_system_.plant8,
            'plant9_json': mode1_vision_system_.plant9,
```

```
            'plant10_json': mode1_vision_system_.plant10,

            'status1_json': mode1_vision_system_.status1,
            'status2_json': mode1_vision_system_.status2,
            'status3_json': mode1_vision_system_.status3,
            'status4_json': mode1_vision_system_.status4,
            'status5_json': mode1_vision_system_.status5,
            'status6_json': mode1_vision_system_.status6,
            'status7_json': mode1_vision_system_.status7,
            'status8_json': mode1_vision_system_.status8,
            'status9_json': mode1_vision_system_.status9,
            'status10_json': mode1_vision_system_.status10,

            'image1_json': mode1_vision_system_.image1,
            'image2_json': mode1_vision_system_.image2,
            'image3_json': mode1_vision_system_.image3,
            'image4_json': mode1_vision_system_.image4,
            'image5_json': mode1_vision_system_.image5,
            'image6_json': mode1_vision_system_.image6,
            'image7_json': mode1_vision_system_.image7,
            'image8_json': mode1_vision_system_.image8,
            'image9_json': mode1_vision_system_.image9,
            'image10_json': mode1_vision_system_.image10,
            }

            return JsonResponse(json)

        if(mode_selected_obj.modeNumber == 2):
            print(" ")
            print("~[ Mode 2 ] Vision System Starting~")
            print(" ")
            print(" ")

        if(mode_selected_obj.modeNumber == 3):
            print(" ")
            print("~[ Mode 3 ] Vision System Starting~")
            print(" ")
            print(" ")

        if(mode_selected_obj.modeNumber == 4):
            print(" ")
            print("~[ Mode 4 ] Vision System Starting~")
            print(" ")
            print(" ")

    if response.POST.get('action') == 'onMode1':

        print(" ")
        print("~Mode 1 Activated~")
```

```
        print(" ")

        GPIO.output(6, GPIO.LOW)
        GPIO.output(5, GPIO.LOW)

        mode_selected_.grid = mode1_obj_global.grid
        mode_selected_.rows = mode1_obj_global.rows
        mode_selected_.columns = mode1_obj_global.columns
        mode_selected_.modeNumber = mode1_obj_global.modeNumber
        mode_selected_.image = '../assets/background/rpiBG.gif'
        mode_selected_.save()

        mode_selected_obj = mode_selected.objects.latest('date')

        json = {
        'image_json': mode_selected_obj.image,
        'grid_json': mode_selected_obj.grid,
        'mode_json': mode_selected_obj.modeNumber,
        }

        return JsonResponse(json)

    if response.POST.get('action') == 'onMode2':

        print(" ")
        print("~Mode 2 Activated~")
        print(" ")

        GPIO.output(6, GPIO.LOW)
        GPIO.output(5, GPIO.HIGH)

        mode_selected_.grid = mode2_obj_global.grid
        mode_selected_.rows = mode2_obj_global.rows
        mode_selected_.columns = mode2_obj_global.columns
        mode_selected_.modeNumber = mode2_obj_global.modeNumber
        mode_selected_.image = '../assets/background/rpiBG.gif'
        mode_selected_.save()

        mode_selected_obj = mode_selected.objects.latest('date')

        json = {
        'image_json': mode_selected_obj.image,
        'grid_json': mode_selected_obj.grid,
        'mode_json': mode_selected_obj.modeNumber,
        }

        return JsonResponse(json)

    if response.POST.get('action') == 'onMode3':
```

```
        print(" ")
        print("~Mode 3 Activated~")
        print(" ")

        GPIO.output(6, GPIO.HIGH)
        GPIO.output(5, GPIO.LOW)

        mode_selected_.grid = mode3_obj_global.grid
        mode_selected_.rows = mode3_obj_global.rows
        mode_selected_.columns = mode3_obj_global.columns
        mode_selected_.modeNumber = mode3_obj_global.modeNumber
        mode_selected_.image = '../assets/background/rpiBG.gif'
        mode_selected_.save()

        mode_selected_obj = mode_selected.objects.latest('date')

        json = {
        'image_json': mode_selected_obj.image,
        'grid_json': mode_selected_obj.grid,
        'mode_json': mode_selected_obj.modeNumber,
        }

        return JsonResponse(json)

    if response.POST.get('action') == 'onMode4':

        print(" ")
        print("~Mode 4 Activated~")
        print(" ")

        GPIO.output(6, GPIO.HIGH)
        GPIO.output(5, GPIO.HIGH)

        mode_selected_.grid = mode4_obj_global.grid
        mode_selected_.rows = mode4_obj_global.rows
        mode_selected_.columns = mode4_obj_global.columns
        mode_selected_.modeNumber = mode4_obj_global.modeNumber
        mode_selected_.image = '../assets/background/rpiBG.gif'
        mode_selected_.save()

        mode_selected_obj = mode_selected.objects.latest('date')

        json = {
        'image_json': mode_selected_obj.image,
        'grid_json': mode_selected_obj.grid,
        'mode_json': mode_selected_obj.modeNumber,
        }
```

```
        return JsonResponse(json)

    if response.POST.get('action') == 'onFan':

        print(" ")
        print("~Air Circulation System Activated~")
        print(" ")

        GPIO.output(20, GPIO.HIGH)
        GPIO.output(16, GPIO.HIGH)

        devices_.fansStatus = 'On'
        devices_.whiteLedStatus = devices_obj_global.whiteLedStatus
        devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
        devices_.calibrationStatus                                  =
devices_obj_global.calibrationStatus
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = devices_obj_global.seedStatus
        devices_.save()

    if response.POST.get('action') == 'offFan':

        print(" ")
        print("~Air Circulation System Deactivated~")
        print(" ")

        GPIO.output(20, GPIO.LOW)
        GPIO.output(16, GPIO.LOW)

        devices_.fansStatus = 'Off'
        devices_.whiteLedStatus = devices_obj_global.whiteLedStatus
        devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
        devices_.calibrationStatus                                  =
devices_obj_global.calibrationStatus
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = devices_obj_global.seedStatus
        devices_.save()

    if response.POST.get('action') == 'onWhiteLed':

        print(" ")
        print("~White LED Activated~")
        print(" ")

        GPIO.output(21, GPIO.HIGH)

        devices_.fansStatus = devices_obj_global.fansStatus
        devices_.whiteLedStatus = 'On'
        devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
```

```
        devices_.calibrationStatus                                  =
devices_obj_global.calibrationStatus
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = devices_obj_global.seedStatus
        devices_.save()


    if response.POST.get('action') == 'offWhiteLed':

        print(" ")
        print("~White LED Deactivated~")
        print(" ")

        GPIO.output(21, GPIO.LOW)

        devices_.fansStatus = devices_obj_global.fansStatus
        devices_.whiteLedStatus = 'Off'
        devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
        devices_.calibrationStatus                                  =
devices_obj_global.calibrationStatus
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = devices_obj_global.seedStatus
        devices_.save()

    if response.POST.get('action') == 'onRgbLed':

        print(" ")
        print("~Grow Lights Activated~")
        print(" ")

        GPIO.output(12, GPIO.HIGH)

        devices_.fansStatus = devices_obj_global.fansStatus
        devices_.whiteLedStatus = devices_obj_global.whiteLedStatus
        devices_.rgbLedStatus = 'On'
        devices_.calibrationStatus                                  =
devices_obj_global.calibrationStatus
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = devices_obj_global.seedStatus
        devices_.save()


    if response.POST.get('action') == 'offRgbLed':

        print(" ")
        print("~Grow Lights Deactivated~")
        print(" ")

        GPIO.output(12, GPIO.LOW)
```

```
        devices_.fansStatus = devices_obj_global.fansStatus
        devices_.whiteLedStatus =  devices_obj_global.whiteLedStatus
        devices_.rgbLedStatus = 'Off'
        devices_.calibrationStatus                                =
devices_obj_global.calibrationStatus
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = devices_obj_global.seedStatus
        devices_.save()


    if response.POST.get('action') == 'onWater':

        sensors_water = sensors.objects.latest('date')

        if (sensors_water.moisture < 50):
            print(" ")
            print("-------------------- Automatic Irrigation System
Activated --------------------")
            print("Time:   "   +   datetime.now().strftime('%Y-%m-%d
%H:%M:%S'))
            print("Soil Moisture: " + str(sensors_water.moisture))

            devices_.fansStatus = devices_obj_global.fansStatus
            devices_.whiteLedStatus                               =
devices_obj_global.whiteLedStatus
            devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
            devices_.calibrationStatus                            =
devices_obj_global.calibrationStatus
            devices_.waterStatus = 'On'
            devices_.seedStatus = devices_obj_global.seedStatus
            devices_.save()

            GPIO.output(19, GPIO.HIGH)
            sleep(1)
            GPIO.output(19, GPIO.LOW)

            print("Trigger Deactivated...")
            print(" ")
            print(" ")

            devices_2.fansStatus = devices_obj_global.fansStatus
            devices_2.whiteLedStatus                              =
devices_obj_global.whiteLedStatus
            devices_2.rgbLedStatus = devices_obj_global.rgbLedStatus
            devices_2.calibrationStatus                           =
devices_obj_global.calibrationStatus
            devices_2.waterStatus = 'Off'
            devices_2.seedStatus = devices_obj_global.seedStatus
```

```
            devices_2.save()

        else:
            print(" ")
            print("Soil Moisture: " + str(sensors_water.moisture))
            print("~Automatic Irrigation System Can't Activate~")
            print(" ")


    if response.POST.get('action') == 'onCalibration':

        print(" ")
        print("~ (PIN 26) Calibration Activated~")
        print(" ")

        devices_.fansStatus = devices_obj_global.fansStatus
        devices_.whiteLedStatus = devices_obj_global.whiteLedStatus
        devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
        devices_.calibrationStatus = 'On'
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = devices_obj_global.seedStatus
        devices_.save()

        GPIO.output(26, GPIO.HIGH)
        sleep(1)
        GPIO.output(26, GPIO.LOW)

        print(" ")
        print("~ (PIN 26) Calibration Deactivated~")
        print(" ")

        devices_2.fansStatus = devices_obj_global.fansStatus
        devices_2.whiteLedStatus = devices_obj_global.whiteLedStatus
        devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
        devices_2.calibrationStatus = 'Off'
        devices_2.waterStatus = devices_obj_global.waterStatus
        devices_2.seedStatus = devices_obj_global.seedStatus
        devices_2.save()

    if response.POST.get('action') == 'onSeed':

        print(" ")
        print("~Seeding System Activated~")
        print(" ")

        devices_.fansStatus = devices_obj_global.fansStatus
        devices_.whiteLedStatus = devices_obj_global.whiteLedStatus
        devices_.rgbLedStatus = devices_obj_global.rgbLedStatus
```

```
        devices_.calibrationStatus                                  =
devices_obj_global.calibrationStatus
        devices_.waterStatus = devices_obj_global.waterStatus
        devices_.seedStatus = 'On'
        devices_.save()

        GPIO.output(13, GPIO.HIGH)
        sleep(1)
        GPIO.output(13, GPIO.LOW)

        print(" ")
        print("~Seeding System Deactivated~")
        print(" ")

        devices_2.fansStatus = devices_obj_global.fansStatus
        devices_2.whiteLedStatus = devices_obj_global.whiteLedStatus
        devices_2.rgbLedStatus = devices_obj_global.rgbLedStatus
        devices_2.calibrationStatus                                 =
devices_obj_global.calibrationStatus
        devices_2.waterStatus = devices_obj_global.waterStatus
        devices_2.seedStatus = 'Off'
        devices_2.save()

    if response.POST.get('action') == 'fullReset':

        print(" ")
        print("~Formatting Database, Turning Off Devices...~")
        print(" ")
        print(" ")
        print(" ")

        # Deactivate Fans
        GPIO.output(20, GPIO.LOW)
        GPIO.output(16, GPIO.LOW)
        # Deactivate White LED
        GPIO.output(21, GPIO.LOW)
        # Deactivate RGB LED
        GPIO.output(12, GPIO.LOW)

        mode_selected.objects.all().delete()
        mode_selected_.daysCounter = 1
        mode_selected_.grid = mode1_obj_global.grid
        mode_selected_.rows = mode1_obj_global.rows
        mode_selected_.columns = mode1_obj_global.columns
        mode_selected_.modeNumber = mode1_obj_global.modeNumber
        mode_selected_.image = '../assets/background/rpiBG.gif'
        mode_selected_.save()

        devices.objects.all().delete()
```

```
        devices_.calibrationStatus = 'Off'
        devices_.fansStatus = 'Off'
        devices_.whiteLedStatus = 'Off'
        devices_.rgbLedStatus = 'Off'
        devices_.waterStatus = 'Off'
        devices_.seedStatus = 'Off'
        devices_.save()

        sensors.objects.all().delete()
        sensors_.temperature = 24
        sensors_.humidity = 65
        sensors_.moisture = 34
        sensors_.temperatureStatus = "Good"
        sensors_.humidityStatus = "Good"
        sensors_.soilMoistureStatus = "Moist"
        sensors_.save()

        threshold.objects.all().delete()
        threshold_.temperature_low = 20
        threshold_.temperature_high = 30
        threshold_.humidity_low = 49
        threshold_.humidity_high = 81
        threshold_.moisture_dry = 30
        threshold_.moisture_moist = 70
        threshold_.moisture_wet = 71
        threshold_.save()

        mode1.objects.all().delete()
        mode1_.grid = '2x5'
        mode1_.rows = 2
        mode1_.columns = 5
        mode1_.modeNumber = 1
        mode1_.save()

        mode2.objects.all().delete()
        mode2_.grid = '2x4'
        mode2_.rows = 2
        mode2_.columns = 4
        mode2_.modeNumber = 2
        mode2_.save()

        mode3.objects.all().delete()
        mode3_.grid = '3x6'
        mode3_.rows = 3
        mode3_.columns = 6
        mode3_.modeNumber = 3
        mode3_.save()

        mode4.objects.all().delete()
```

```
        mode4_.grid = '3x4'
        mode4_.rows = 3
        mode4_.columns = 4
        mode4_.modeNumber = 4
        mode4_.save()

        mode1_vision_system.objects.all().delete()
        mode1_vision_system_.image                                  =
'../assets/background/rpiBG.gif'
        mode1_vision_system_.plant1 = 0
        mode1_vision_system_.plant2 = 0
        mode1_vision_system_.plant3 = 0
        mode1_vision_system_.plant4 = 0
        mode1_vision_system_.plant5 = 0
        mode1_vision_system_.plant6 = 0
        mode1_vision_system_.plant7 = 0
        mode1_vision_system_.plant8 = 0
        mode1_vision_system_.plant9 = 0
        mode1_vision_system_.plant10 = 0
        mode1_vision_system_.status1 = 'Early'
        mode1_vision_system_.status2 = 'Early'
        mode1_vision_system_.status3 = 'Early'
        mode1_vision_system_.status4 = 'Early'
        mode1_vision_system_.status5 = 'Early'
        mode1_vision_system_.status6 = 'Early'
        mode1_vision_system_.status7 = 'Early'
        mode1_vision_system_.status8 = 'Early'
        mode1_vision_system_.status9 = 'Early'
        mode1_vision_system_.status10 = 'Early'
        mode1_vision_system_.image1 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image2 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image3 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image4 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image5 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image6 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image7 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image8 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image9 = '../assets/icons/seedIcon.png'
        mode1_vision_system_.image10                                =
'../assets/icons/seedIcon.png'
        mode1_vision_system_.save()

        mode2_vision_system.objects.all().delete()
        mode2_vision_system_.image                                  =
'../assets/background/rpiBG.gif'
        mode2_vision_system_.plant1 = 0
        mode2_vision_system_.plant2 = 0
        mode2_vision_system_.plant3 = 0
        mode2_vision_system_.plant4 = 0
```

```
        mode2_vision_system_.plant5 = 0
        mode2_vision_system_.plant6 = 0
        mode2_vision_system_.plant7 = 0
        mode2_vision_system_.plant8 = 8
        mode2_vision_system_.plant10 = 0
        mode2_vision_system_.status1 = 'Early'
        mode2_vision_system_.status2 = 'Early'
        mode2_vision_system_.status3 = 'Early'
        mode2_vision_system_.status4 = 'Early'
        mode2_vision_system_.status5 = 'Early'
        mode2_vision_system_.status6 = 'Early'
        mode2_vision_system_.status7 = 'Early'
        mode2_vision_system_.status8 = 'Early'
        mode2_vision_system_.image1 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.image2 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.image3 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.image4 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.image5 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.image6 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.image7 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.image8 = '../assets/icons/seedIcon.png'
        mode2_vision_system_.save()

        mode3_vision_system.objects.all().delete()
        mode3_vision_system_.image                                  =
'../assets/background/rpiBG.gif'
        mode3_vision_system_.plant1 = 0
        mode3_vision_system_.plant2 = 0
        mode3_vision_system_.plant3 = 0
        mode3_vision_system_.plant4 = 0
        mode3_vision_system_.plant5 = 0
        mode3_vision_system_.plant6 = 0
        mode3_vision_system_.plant7 = 0
        mode3_vision_system_.plant8 = 0
        mode3_vision_system_.plant9 = 0
        mode3_vision_system_.plant10 = 0
        mode3_vision_system_.plant11 = 0
        mode3_vision_system_.plant12 = 0
        mode3_vision_system_.plant13 = 0
        mode3_vision_system_.plant14 = 0
        mode3_vision_system_.plant15 = 0
        mode3_vision_system_.plant16 = 0
        mode3_vision_system_.plant17 = 0
        mode3_vision_system_.plant18 = 18
        mode3_vision_system_.status1 = 'Early'
        mode3_vision_system_.status2 = 'Early'
        mode3_vision_system_.status3 = 'Early'
        mode3_vision_system_.status4 = 'Early'
        mode3_vision_system_.status5 = 'Early'
```

```
        mode3_vision_system_.status6 = 'Early'
        mode3_vision_system_.status7 = 'Early'
        mode3_vision_system_.status8 = 'Early'
        mode3_vision_system_.status9 = 'Early'
        mode3_vision_system_.status10 = 'Early'
        mode3_vision_system_.status11 = 'Early'
        mode3_vision_system_.status12 = 'Early'
        mode3_vision_system_.status13 = 'Early'
        mode3_vision_system_.status14 = 'Early'
        mode3_vision_system_.status15 = 'Early'
        mode3_vision_system_.status16 = 'Early'
        mode3_vision_system_.status17 = 'Early'
        mode3_vision_system_.status18 = 'Early'
        mode3_vision_system_.image1 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image2 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image3 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image4 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image5 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image6 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image7 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image8 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image9 = '../assets/icons/seedIcon.png'
        mode3_vision_system_.image10                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image11                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image12                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image13                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image14                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image15                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image16                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image17                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.image18                                =
'../assets/icons/seedIcon.png'
        mode3_vision_system_.save()

        mode4_vision_system.objects.all().delete()
        mode4_vision_system_.image                                  =
'../assets/background/rpiBG.gif'
        mode4_vision_system_.plant1 = 0
        mode4_vision_system_.plant2 = 0
        mode4_vision_system_.plant3 = 0
        mode4_vision_system_.plant4 = 0
```

```
        mode4_vision_system_.plant5 = 0
        mode4_vision_system_.plant6 = 0
        mode4_vision_system_.plant7 = 0
        mode4_vision_system_.plant8 = 0
        mode4_vision_system_.plant9 = 0
        mode4_vision_system_.plant10 = 0
        mode4_vision_system_.plant11 = 0
        mode4_vision_system_.plant12 = 12
        mode4_vision_system_.status1 = 'Early'
        mode4_vision_system_.status2 = 'Early'
        mode4_vision_system_.status3 = 'Early'
        mode4_vision_system_.status4 = 'Early'
        mode4_vision_system_.status5 = 'Early'
        mode4_vision_system_.status6 = 'Early'
        mode4_vision_system_.status7 = 'Early'
        mode4_vision_system_.status8 = 'Early'
        mode4_vision_system_.status9 = 'Early'
        mode4_vision_system_.status10 = 'Early'
        mode4_vision_system_.status11 = 'Early'
        mode4_vision_system_.status12 = 'Early'
        mode4_vision_system_.image1 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image2 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image3 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image4 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image5 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image6 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image7 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image8 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image9 = '../assets/icons/seedIcon.png'
        mode4_vision_system_.image10                                =
'../assets/icons/seedIcon.png'
        mode4_vision_system_.image11                                =
'../assets/icons/seedIcon.png'
        mode4_vision_system_.image12                                =
'../assets/icons/seedIcon.png'
        mode4_vision_system_.save()

        mode_selected_obj = mode_selected.objects.latest('date')
        mode1_visionSystem_obj                                      =
mode1_vision_system.objects.latest('date')
        sensors_obj = sensors.objects.latest('date')
        devices_obj = devices.objects.latest('date')

        json = {
        'mode_json': mode_selected_obj.modeNumber,
        'grid_json': mode_selected_obj.grid,
        'startDate_json': str(datetime.now().strftime('%b. %d, %Y, %-
I:%M %p')),
        'daysCounter_json' : str(mode_selected_obj.daysCounter),
```

```
        'calibration_json' : devices_obj.calibrationStatus,
        'fans_json' : devices_obj.fansStatus,
        'whiteLed_json' : devices_obj.whiteLedStatus,
        'rgbLed_json' : devices_obj.rgbLedStatus,
        'water_json' : devices_obj.waterStatus,
        'seeder_json' : devices_obj.seedStatus,

        'temperature_json': sensors_obj.temperature,
        'humidity_json': sensors_obj.humidity,
        'soilMoisture_json': sensors_obj.moisture,
        'temperatureStatus_json': sensors_obj.temperatureStatus,
        'humidityStatus_json': sensors_obj.humidityStatus,
        'soilMoistureStatus_json': sensors_obj.soilMoistureStatus,

        'image_json' : str(mode1_visionSystem_obj.image),
        }

        print("---------------------------- GrowSmart Initializing --
----------------------------")
        print("Time:     "     +     datetime.now().strftime('%Y-%m-%d
%H:%M:%S'))
        print("Mode: " + str(mode_selected_obj.modeNumber))
        print("Grid: " + mode_selected_obj.grid)
        print(" ")
        print(" ")

        return JsonResponse(json)

    sensors_obj_global = sensors.objects.latest('date')
    mode1_vision_system_obj_global                                    =
mode1_vision_system.objects.latest('date')
    mode2_vision_system_obj_global                                    =
mode2_vision_system.objects.latest('date')
    mode3_vision_system_obj_global                                    =
mode3_vision_system.objects.latest('date')
    mode4_vision_system_obj_global                                    =
mode4_vision_system.objects.latest('date')
    mode_selected_obj_global_first = mode_selected.objects.first()
    mode_selected_obj_global_2                                        =
mode_selected.objects.latest('date')

    myObj            =            {'mode_selected_obj_global_first':
mode_selected_obj_global_first,         'mode_selected_obj_global_2':
mode_selected_obj_global_2,                     'devices_obj_global':
devices_obj_global,
                'sensors_obj_global':            sensors_obj_global,
'mode1_vision_system_obj_global':    mode1_vision_system_obj_global,
'mode2_vision_system_obj_global': mode2_vision_system_obj_global
```

```
                ,                    'mode3_vision_system_obj_global':
mode3_vision_system_obj_global,     'mode4_vision_system_obj_global':
mode4_vision_system_obj_global }

    return render(response, 'main.html', context=myObj)

def databasePage(response):

    devices_obj_global = devices.objects.all()
    sensors_obj_global = sensors.objects.all()
    mode1_vision_system_obj_global                                  =
mode1_vision_system.objects.all()
    mode_selected_obj_global = mode_selected.objects.all()
    mode_selected_obj_global_2                                      =
mode_selected.objects.latest('date')

    myObj  =  {'mode_selected_obj_global':  mode_selected_obj_global,
'mode_selected_obj_global_2':            mode_selected_obj_global_2,
'devices_obj_global': devices_obj_global,
                'sensors_obj_global':            sensors_obj_global,
'mode1_vision_system_obj_global': mode1_vision_system_obj_global}

    return render(response, 'database.html', context=myObj)

def sensorsPage(response):

    threshold_ = threshold()

    if response.POST.get('action') == 'temperature_low_set':

        threshold_obj0 = threshold.objects.latest('date')

        threshold_.temperature_low                                  =
response.POST.get('temperature_low_data')
        threshold_.temperature_high                                 =
threshold_obj0.temperature_high
        threshold_.humidity_low = threshold_obj0.humidity_low
        threshold_.humidity_high = threshold_obj0.humidity_high
        threshold_.moisture_dry = threshold_obj0.moisture_dry
        threshold_.moisture_wet = threshold_obj0.moisture_wet
        threshold_.save()

        threshold_obj = threshold.objects.latest('date')

        json = {
        'temperature_low' : threshold_obj.temperature_low
        }
        return JsonResponse(json)
```

```
    if response.POST.get('action') == 'temperature_high_set':

        threshold_obj0 = threshold.objects.latest('date')

        threshold_.temperature_low = threshold_obj0.temperature_low
        threshold_.temperature_high                                 =
response.POST.get('temperature_high_data')
        threshold_.humidity_low = threshold_obj0.humidity_low
        threshold_.humidity_high = threshold_obj0.humidity_high
        threshold_.moisture_dry = threshold_obj0.moisture_dry
        threshold_.moisture_wet = threshold_obj0.moisture_wet
        threshold_.save()

        threshold_obj = threshold.objects.latest('date')

        json = {
        'temperature_high' : threshold_obj.temperature_high
        }
        return JsonResponse(json)

    if response.POST.get('action') == 'humidity_low_set':

        threshold_obj0 = threshold.objects.latest('date')

        threshold_.temperature_low = threshold_obj0.temperature_low
        threshold_.temperature_high                                 =
threshold_obj0.temperature_high
        threshold_.humidity_low                                     =
response.POST.get('humidity_low_data')
        threshold_.humidity_high = threshold_obj0.humidity_high
        threshold_.moisture_dry = threshold_obj0.moisture_dry
        threshold_.moisture_wet = threshold_obj0.moisture_wet
        threshold_.save()

        threshold_obj = threshold.objects.latest('date')

        json = {
        'humidity_low' : threshold_obj.humidity_low
        }
        return JsonResponse(json)

    if response.POST.get('action') == 'humidity_high_set':

        threshold_obj0 = threshold.objects.latest('date')

        threshold_.temperature_low = threshold_obj0.temperature_low
        threshold_.temperature_high                                 =
threshold_obj0.temperature_high
        threshold_.humidity_low = threshold_obj0.humidity_low
```

```
        threshold_.humidity_high                                    =
response.POST.get('humidity_high_data')
        threshold_.moisture_dry = threshold_obj0.moisture_dry
        threshold_.moisture_wet = threshold_obj0.moisture_wet
        threshold_.save()

        threshold_obj = threshold.objects.latest('date')

        json = {
        'humidity_high' : threshold_obj.humidity_high
        }
        return JsonResponse(json)

    if response.POST.get('action') == 'moisture_dry_set':

        threshold_obj0 = threshold.objects.latest('date')

        threshold_.temperature_low = threshold_obj0.temperature_low
        threshold_.temperature_high                                 =
threshold_obj0.temperature_high
        threshold_.humidity_low = threshold_obj0.humidity_low
        threshold_.humidity_high = threshold_obj0.humidity_high
        threshold_.moisture_dry                                     =
response.POST.get('moisture_dry_data')
        threshold_.moisture_wet = threshold_obj0.moisture_wet
        threshold_.save()

        threshold_obj = threshold.objects.latest('date')

        json = {
        'moisture_dry' : threshold_obj.moisture_dry
        }
        return JsonResponse(json)

    if response.POST.get('action') == 'moisture_wet_set':

        threshold_obj0 = threshold.objects.latest('date')

        threshold_.temperature_low = threshold_obj0.temperature_low
        threshold_.temperature_high                                 =
threshold_obj0.temperature_high
        threshold_.humidity_low = threshold_obj0.humidity_low
        threshold_.humidity_high = threshold_obj0.humidity_high
        threshold_.moisture_dry = threshold_obj0.moisture_dry
        threshold_.moisture_wet                                     =
response.POST.get('moisture_wet_data')
        threshold_.save()

        threshold_obj = threshold.objects.latest('date')
```

```
        json = {
        'moisture_wet' : threshold_obj.moisture_wet
        }
        return JsonResponse(json)

    threshold_obj_global = threshold.objects.latest('date')

    myObj = {'threshold_obj_global': threshold_obj_global}

    return render(response, 'sensors.html', context=myObj)
```

Appendix H BASE HTML

```
{% load static %}

<!DOCTYPE html>
<html>

<head>

  <title>GrowSmart</title>

  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <!-- <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no"> -->


  <!-- <link rel="stylesheet" href="{% static 'styles.css' %}"> -->

  <link rel="stylesheet" href="/assets/css/styles.css">

  <!-- Jquery for ajax calls -->
  <script src="/assets/js/jquery-3.4.1.min.js"></script>

  <!-- Alertify for confirmation prompts -->
  <script                                    src="/assets/alertify.js-
0.3.11/lib/alertify.min.js"></script>
  <link          rel="stylesheet"          href="/assets/alertify.js-
0.3.11/themes//alertify.core.css" />
  <link          rel="stylesheet"          href="/assets/alertify.js-
0.3.11/themes/alertify.default.css" id="toggleCSS" />

  <!-- Custom fonts for this template-->
  <link       href="/assets/vendor/fontawesome-free/css/all.min.css"
rel="stylesheet" type="text/css">
  <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,30
0i,400,400i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

  <!-- Custom styles for this template-->
  <link href="/assets/css/sb-admin-2.min.css" rel="stylesheet">

  <!-- Bootstrap core JavaScript-->
  <script src="/assets/vendor/jquery/jquery.min.js"></script>
  <script
src="/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

  <!-- Core plugin JavaScript-->
```

```
  <script                                    src="/assets/vendor/jquery-
easing/jquery.easing.min.js"></script>

  <!-- Custom scripts for all pages-->
  <script src="/assets/js/sb-admin-2.min.js"></script>

  <!-- Page level plugins -->
  <script
src="/assets/vendor/datatables/jquery.dataTables.js"></script>
  <script
src="/assets/vendor/datatables/jquery.sensorsTable.js"></script>
  <script
src="/assets/vendor/datatables/dataTables.bootstrap4.js"></script>
  <script
src="/assets/vendor/datatables/sensorsTables.bootstrap4.js"></script
>

  <!-- Page level custom scripts -->
  <script src="/assets/js/demo/datatables-demo.js"></script>

</head>


<body id="page-top">

  <!-- Page Wrapper -->
  <div id="wrapper">

    <!-- Sidebar -->
    <ul  class="navbar-nav  bg-gradient-primary  sidebar  sidebar-dark
accordion" id="accordionSidebar" style="background: #1cc88a">

      <!-- Sidebar - Brand -->
      <a  class="sidebar-brand  d-flex  align-items-center  justify-
content-center" href="{% url 'mainPage' %}">
        <div class="sidebar-brand-icon rotate-n-15">
          <i><img                  src="../assets/icons/rpiGif2.gif"
class="singleContainer_icon" /></i>
        </div>
        <div class="sidebar-brand-text mx-3">GrowSmart</div>
      </a>

      <!-- Divider -->
      <hr class="sidebar-divider my-0">



      {% block content %}
      {% endblock %}
```

```
  </div>
  <!-- /.container-fluid -->

  </div>
  <!-- End of Main Content -->

  <!-- Footer -->
  <footer class="sticky-footer bg-white">
    <div class="container my-auto">
      <div class="copyright text-center my-auto">
        <span>Copyright &copy; A.G.E 2020</span>
      </div>
    </div>
  </footer>
  <!-- End of Footer -->

  </div>
  <!-- End of Content Wrapper -->

  </div>
  <!-- End of Page Wrapper -->

</body>

<script>
  setup()

  function confirmFullReset() {
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'fullReset',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();

        $('#labelMode').text("Mode: " + json.mode_json);
        $('#labelGrid').text("Grid: " + json.grid_json);
        $('#daysID').text("Day " + json.daysCounter_json);
        $('#firstDayID').text("Day 1: " + json.startDate_json);

        $('#labelCalibration').text(json.calibration_json);
        $('#labelFans').text(json.fans_json);
```

```
        $('#labelWhiteLed').text(json.whiteLed_json);
        $('#labelRgbLed').text(json.rgbLed_json);
        $('#labelWater').text(json.water_json);
        $('#labelSeed').text(json.seeder_json);

        $('#labelTemperature').text(json.temperature_json + " ° C");
        $('#labelHumidity').text(json.humidity_json + " %");
        $('#labelMoisture').text(json.soilMoisture_json + " %");
        $('#labelTemperatureStatus').text("Status:        "        +
json.temperatureStatus_json);
        $('#labelHumidityStatus').text("Status:         "         +
json.humidityStatus_json);
        $('#labelSoilMoistureStatus').text("Status:       "       +
json.soilMoistureStatus_json);

        $('#cameraID').attr('src', json.image_json);


        $('#growthContainer').remove()
        $('#growthContainer_permanent').append('<div
id="growthContainer">' +

          '<div class="row">' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          //'<div class="card border-left-success shadow h-100 py-2"
style="background: white;">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 1</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                  src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
```

```
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 2</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                  src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 3</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                  src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
```

```
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 4</div>' +
          '<div id="labelPlant1Growth" class="h5 mb-0 font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div class="card shadow h-100 py-2" style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 5</div>' +
          '<div id="labelPlant1Growth" class="h5 mb-0 font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div class="card shadow h-100 py-2" style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 6</div>' +
          '<div id="labelPlant1Growth" class="h5 mb-0 font-weight-
bold text-gray-800">0</div>' +
```

```
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                  src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 7</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                  src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 8</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
```

```
          '<i><img                    src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 9</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                    src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 10</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold text-gray-800">0</div>' +
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal text-gray-800">Early</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                    src=../assets/icons/seedIcon.png
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
```

```
          '</div>' +
          '</div>' +

          '</div>' +

          '</div>')
      },
    });

    alertify.custom = alertify.extend("custom");
    alertify.custom("Database Cleared!");

  }

  function automaticRgbLed_on() {
    var date1 = new Date();
    if (date1.getHours() === 13 && date1.getMinutes() === 00) {
      $.ajax({
        type: 'POST',
        url: '{% url "mainPage" %}',
        data: {

          csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
          action: 'onRgbLed',

        },
        success: function(json) {
          document.getElementById("mainForm").reset();
          $('#labelRgbLed').text('On');
        },

      });
      alertify.custom = alertify.extend("custom");
      alertify.custom("Grown Lights Activated");
    }
  }

  function automaticRgbLed_off() {
    var date1 = new Date();
    if (date1.getHours() === 17 && date1.getMinutes() === 00) {
      $.ajax({
        type: 'POST',
        url: '{% url "mainPage" %}',
        data: {

          csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
          action: 'offRgbLed',
```

```
        },
        success: function(json) {
          document.getElementById("mainForm").reset();
          $('#labelRgbLed').text('Off');
        },

      });
      alertify.custom = alertify.extend("custom");
      alertify.custom("Grown Lights Deactivated");
    }
  }


  function automaticWhiteLed_on() {
    var date1 = new Date();
    if (date1.getHours() === 7 && date1.getMinutes() === 55) {
      $.ajax({
        type: 'POST',
        url: '{% url "mainPage" %}',
        data: {

          csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
          action: 'onWhiteLed',

        },
        success: function(json) {
          document.getElementById("mainForm").reset();
          $('#labelWhiteLed').text('On');
        },

      });
      alertify.custom = alertify.extend("custom");
      alertify.custom("White Leds Activated");
    }
  }

  function automaticWhiteLed_off() {
    var date1 = new Date();
    if (date1.getHours() === 8 && date1.getMinutes() === 05) {
      $.ajax({
        type: 'POST',
        url: '{% url "mainPage" %}',
        data: {

          csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
          action: 'offWhiteLed',
```

```
        },
        success: function(json) {
          document.getElementById("mainForm").reset();
          $('#labelWhiteLed').text('Off');
        },

      });
      alertify.custom = alertify.extend("custom");
      alertify.custom("White Leds Deactivated");
    }
  }

  function snapImage() {
    var date1 = new Date();
    if (date1.getHours() === 8 && date1.getMinutes() === 00) {
      //Change getHours and getMinutes depending on platform used
      //Rpi 24 hr format
      //Ubuntu 12 hr format
      //Windows 12 hr format
      //alert(date1);
      $.ajax({
        type: 'POST',
        url: '{% url "mainPage" %}',
        data: {
          csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
          action: 'snapImage',
        },
        success: function(json) {
          $('#cameraID').attr('src', json.image_json);

          $('#labelPlant1Growth').text(json.plant1_json);
          $('#labelPlant2Growth').text(json.plant2_json);
          $('#labelPlant3Growth').text(json.plant3_json);
          $('#labelPlant4Growth').text(json.plant4_json);
          $('#labelPlant5Growth').text(json.plant5_json);
          $('#labelPlant6Growth').text(json.plant6_json);
          $('#labelPlant7Growth').text(json.plant7_json);
          $('#labelPlant8Growth').text(json.plant8_json);
          $('#labelPlant9Growth').text(json.plant9_json);
          $('#labelPlant10Growth').text(json.plant10_json);

          $('#labelPlant1GrowthStatus').text(json.status1_json);
          $('#labelPlant2GrowthStatus').text(json.status2_json);
          $('#labelPlant3GrowthStatus').text(json.status3_json);
          $('#labelPlant4GrowthStatus').text(json.status4_json);
          $('#labelPlant5GrowthStatus').text(json.status5_json);
          $('#labelPlant6GrowthStatus').text(json.status6_json);
```

```
            $('#labelPlant7GrowthStatus').text(json.status7_json);
            $('#labelPlant8GrowthStatus').text(json.status8_json);
            $('#labelPlant9GrowthStatus').text(json.status9_json);
            $('#labelPlant10GrowthStatus').text(json.status10_json);

            $('#image1ID').attr('src', json.image1_json);
            $('#image2ID').attr('src', json.image2_json);
            $('#image3ID').attr('src', json.image3_json);
            $('#image4ID').attr('src', json.image4_json);
            $('#image5ID').attr('src', json.image5_json);
            $('#image6ID').attr('src', json.image6_json);
            $('#image7ID').attr('src', json.image7_json);
            $('#image8ID').attr('src', json.image8_json);
            $('#image9ID').attr('src', json.image9_json);
            $('#image10ID').attr('src', json.image10_json);

            $('#daysID').text("Day " + json.daysCounter_json);
          },
        });
        alertify.custom = alertify.extend("custom");
        alertify.custom("Vision System Activated");
      }
    }


    function automaticIrrigation() {
      var date1 = new Date();
      if (date1.getHours() === 12 && date1.getMinutes() === 00) {
        $('#labelWater').text('On');
        $.ajax({
          type: 'POST',
          url: '{% url "mainPage" %}',
          data: {

            csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
            action: 'onWater',

          },
          success: function(json) {
            document.getElementById("mainForm").reset();
            $('#labelWater').text('Off');
          },
        });
        alertify.custom = alertify.extend("custom");
        alertify.custom("Irrigation System Activated");
      }
    }
```

```
  function getSensorValues() {
    var date1 = new Date();
    if (date1.getMinutes() === 00) {
      $.ajax({
        type: 'POST',
        url: '{% url "mainPage" %}',
        data: {
          csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
          action: 'getSensorValues',
        },
        success: function(json) {
          document.getElementById("mainForm").reset();
          $('#daysID').text("Day " + json.daysCounter_json);
          $('#lastDayID').text('Age since: ' + json.date_json);
          $('#labelTemperature').text(json.temperature_json  +  "  °
C");
          $('#labelHumidity').text(json.humidity_json + " %");
          $('#labelMoisture').text(json.soilMoisture_json + " %");
          $('#labelTemperatureStatus').text("Status:       "       +
json.temperatureStatus_json);
          $('#labelHumidityStatus').text("Status:         "         +
json.humidityStatus_json);
          $('#labelSoilMoistureStatus').text("Status:       "       +
json.soilMoistureStatus_json);
        },
      });
      alertify.custom = alertify.extend("custom");
      alertify.custom("Sensor Values Updated");
    }
  }

  $(document).ready(function() {
    setInterval(automaticFunctions, 60000);
  });

  function automaticFunctions() {
    automaticRgbLed_on();
    automaticRgbLed_off();
    automaticWhiteLed_on();
    automaticWhiteLed_off();
    snapImage();
    automaticIrrigation();
    getSensorValues();
  }

  $("#processImage").click(function(e) {
    e.preventDefault();
```

```
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {
        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'snapImage',
      },
      success: function(json) {
        $('#cameraID').attr('src', json.image_json);

        $('#labelPlant1Growth').text(json.plant1_json);
        $('#labelPlant2Growth').text(json.plant2_json);
        $('#labelPlant3Growth').text(json.plant3_json);
        $('#labelPlant4Growth').text(json.plant4_json);
        $('#labelPlant5Growth').text(json.plant5_json);
        $('#labelPlant6Growth').text(json.plant6_json);
        $('#labelPlant7Growth').text(json.plant7_json);
        $('#labelPlant8Growth').text(json.plant8_json);
        $('#labelPlant9Growth').text(json.plant9_json);
        $('#labelPlant10Growth').text(json.plant10_json);

        $('#labelPlant1GrowthStatus').text(json.status1_json);
        $('#labelPlant2GrowthStatus').text(json.status2_json);
        $('#labelPlant3GrowthStatus').text(json.status3_json);
        $('#labelPlant4GrowthStatus').text(json.status4_json);
        $('#labelPlant5GrowthStatus').text(json.status5_json);
        $('#labelPlant6GrowthStatus').text(json.status6_json);
        $('#labelPlant7GrowthStatus').text(json.status7_json);
        $('#labelPlant8GrowthStatus').text(json.status8_json);
        $('#labelPlant9GrowthStatus').text(json.status9_json);
        $('#labelPlant10GrowthStatus').text(json.status10_json);

        $('#image1ID').attr('src', json.image1_json);
        $('#image2ID').attr('src', json.image2_json);
        $('#image3ID').attr('src', json.image3_json);
        $('#image4ID').attr('src', json.image4_json);
        $('#image5ID').attr('src', json.image5_json);
        $('#image6ID').attr('src', json.image6_json);
        $('#image7ID').attr('src', json.image7_json);
        $('#image8ID').attr('src', json.image8_json);
        $('#image9ID').attr('src', json.image9_json);
        $('#image10ID').attr('src', json.image10_json);

        $('#daysID').text("Day " + json.daysCounter_json);
      },
    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Vision System Activated");
```

```
  });

  $("#onFan").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onFan',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelFans').text('On');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Fans Activated");
  });

  $("#offFan").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'offFan',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelFans').text('Off');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Fans Deactivated");
  });
```

```
  $("#onWhiteLed").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onWhiteLed',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelWhiteLed').text('On');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("White Leds Activated");
  });

  $("#offWhiteLed").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'offWhiteLed',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelWhiteLed').text('Off');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("White Leds Deactivated");
  });


  $("#onRgbLed").click(function(e) {
```

```
    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onRgbLed',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelRgbLed').text('On');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Grown Lights Activated");
  });

  $("#offRgbLed").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'offRgbLed',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelRgbLed').text('Off');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Grow Lights Deactivated");
  });

  $("#onWater").click(function(e) {

    e.preventDefault();
    $('#labelWater').text('On');
    $.ajax({
```

```
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onWater',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelWater').text('Off');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Irrigation System Activated");
  });

  $("#onSeed").click(function(e) {

    e.preventDefault();
    $('#labelSeed').text('On');
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onSeed',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelSeed').text('Off');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Seeding System Activated");
  });

  $("#onCalibration").click(function(e) {

    e.preventDefault();
    $('#labelCalibration').text('On');
    $.ajax({
      type: 'POST',
```

```
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onCalibration',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#labelCalibration').text('Off');
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Calibration Activated");
  });

  $("#temperature_low_btn").click(function(e) {
    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "sensorsPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'temperature_low_set',
        temperature_low_data:
document.getElementById("temperature_low_input").value

      },
      success: function(json) {
        document.getElementById("sensorsForm").reset();
        $('#temperature_low_label').text("<= " + json.temperature_low
+ " ° C");
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Temperature low threshold updated");
  });

  $("#temperature_high_btn").click(function(e) {
    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "sensorsPage" %}',
      data: {
```

```
        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'temperature_high_set',
        temperature_high_data:
document.getElementById("temperature_high_input").value

      },
      success: function(json) {
        document.getElementById("sensorsForm").reset();
        $('#temperature_high_label').text(">=           "           +
json.temperature_high + " ° C");
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Temperature high threshold updated");
  });

  $("#humidity_low_btn").click(function(e) {
    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "sensorsPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'humidity_low_set',
        humidity_low_data:
document.getElementById("humidity_low_input").value

      },
      success: function(json) {
        document.getElementById("sensorsForm").reset();
        $('#humidity_low_label').text("<= " + json.humidity_low + "
%");
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Humidity low threshold updated");
  });

  $("#humidity_high_btn").click(function(e) {
    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "sensorsPage" %}',
```

```
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'humidity_high_set',
        humidity_high_data:
document.getElementById("humidity_high_input").value

      },
      success: function(json) {
        document.getElementById("sensorsForm").reset();
        $('#humidity_high_label').text(">= " + json.humidity_high + "
%");
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Humidity high threshold updated");
  });

  $("#moisture_dry_btn").click(function(e) {
    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "sensorsPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'moisture_dry_set',
        moisture_dry_data:
document.getElementById("moisture_dry_input").value

      },
      success: function(json) {
        document.getElementById("sensorsForm").reset();
        $('#moisture_dry_label').text("<= " + json.moisture_dry + "
%");
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Moisture dry threshold updated");
  });

  $("#moisture_wet_btn").click(function(e) {
    e.preventDefault();
    $.ajax({
      type: 'POST',
```

```
      url: '{% url "sensorsPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'moisture_wet_set',
        moisture_wet_data:
document.getElementById("moisture_wet_input").value

      },
      success: function(json) {
        document.getElementById("sensorsForm").reset();
        $('#moisture_wet_label').text(">= " + json.moisture_wet + "
%");
      },

    });
    alertify.custom = alertify.extend("custom");
    alertify.custom("Moisture wet threshold updated");
  });



  $("#onMode1").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onMode1',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#cameraID').attr('src', json.image_json);
        $('#labelMode').text("Mode: " + json.mode_json);
        $('#labelGrid').text("Grid: " + json.grid_json);
        $('#growthContainer').remove()
        $('#growthContainer_permanent').append('<div
id="growthContainer">' +

          '<div class="row">' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
```

```
          //'<div class="card border-left-success shadow h-100 py-2"
style="background: white;">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 1</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant1}}</div>'
+
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal                                            text-gray-
800">{{model_vision_system_obj_global.status1}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                      id="image1ID"
src={{model_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 2</div>' +
          '<div  id="labelPlant2Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant2}}</div>'
+
          '<div id="labelPlant2GrowthStatus" class="text-s mb-0 font-
weight-normal                                            text-gray-
800">{{model_vision_system_obj_global.status2}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                      id="image2ID"
src={{model_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
```

```
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 3</div>' +
          '<div  id="labelPlant3Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant3}}</div>'
+
          '<div id="labelPlant3GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{model_vision_system_obj_global.status3}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                        id="image3ID"
src={{model_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 4</div>' +
          '<div  id="labelPlant4Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant4}}</div>'
+
          '<div id="labelPlant4GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{model_vision_system_obj_global.status4}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                        id="image4ID"
src={{model_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
```

```
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 5</div>' +
          '<div  id="labelPlant5Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant5}}</div>'
+
          '<div id="labelPlant5GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{model_vision_system_obj_global.status5}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                      id="image5ID"
src={{model_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 6</div>' +
          '<div  id="labelPlant6Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant6}}</div>'
+
          '<div id="labelPlant6GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{model_vision_system_obj_global.status6}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                      id="image6ID"
src={{model_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
```

```
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 7</div>' +
          '<div  id="labelPlant7Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant7}}</div>'
+
          '<div id="labelPlant7GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{model_vision_system_obj_global.status7}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                      id="image7ID"
src={{model_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 8</div>' +
          '<div  id="labelPlant8Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant8}}</div>'
+
          '<div id="labelPlant8GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{model_vision_system_obj_global.status8}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                      id="image8ID"
src={{model_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
          '</div>' +
```

```
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 9</div>' +
          '<div  id="labelPlant9Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{model_vision_system_obj_global.plant9}}</div>'
+
          '<div id="labelPlant9GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{model_vision_system_obj_global.status9}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                      id="image9ID"
src={{model_vision_system_obj_global.image9}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 10</div>' +
          '<div  id="labelPlant10Growth"  class="h5  mb-0  font-weight-
bold                                                          text-gray-
800">{{model_vision_system_obj_global.plant10}}</div>' +
          '<div   id="labelPlant10GrowthStatus"   class="text-s   mb-0
font-weight-normal                                            text-gray-
800">{{model_vision_system_obj_global.status10}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img                                     id="image10ID"
src={{model_vision_system_obj_global.image10}}
class="singleContainer_icon" /></i>' +
```

```
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '</div>' +

          '</div>')
      },

    });

    alertify.custom = alertify.extend("custom");
    alertify.custom("Mode 1 Activated");
  });

  $("#onMode2").click(function(e) {

    e.preventDefault();

    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onMode2',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#cameraID').attr('src', json.image_json);
        $('#labelMode').text("Mode: " + json.mode_json);
        $('#labelGrid').text("Grid: " + json.grid_json);
        $('#growthContainer').remove()
        $('#growthContainer_permanent').append('<div
id="growthContainer">' +

          '<div class="row">' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          //'<div class="card border-left-success shadow h-100 py-2"
style="background: white;">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
```

```
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 1</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode2_vision_system_obj_global.plant1}}</div>'
+
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode2_vision_system_obj_global.status1}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 2</div>' +
          '<div  id="labelPlant2Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode2_vision_system_obj_global.plant2}}</div>'
+
          '<div id="labelPlant2GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode2_vision_system_obj_global.status2}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
```

```
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 3</div>' +
          '<div id="labelPlant3Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant3}}</div>'
+
          '<div id="labelPlant3GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode2_vision_system_obj_global.status3}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div class="card shadow h-100 py-2" style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 4</div>' +
          '<div id="labelPlant4Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant4}}</div>'
+
          '<div id="labelPlant4GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode2_vision_system_obj_global.status4}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div class="card shadow h-100 py-2" style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
```

```
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 5</div>' +
          '<div id="labelPlant5Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant5}}</div>'
+
          '<div id="labelPlant5GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode2_vision_system_obj_global.status5}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div class="card shadow h-100 py-2" style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 6</div>' +
          '<div id="labelPlant6Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant6}}</div>'
+
          '<div id="labelPlant6GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode2_vision_system_obj_global.status6}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div class="card shadow h-100 py-2" style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
```

```
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 7</div>' +
          '<div id="labelPlant7Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant7}}</div>'
+
          '<div id="labelPlant7GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode2_vision_system_obj_global.status7}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div class="card shadow h-100 py-2" style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 8</div>' +
          '<div id="labelPlant8Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant8}}</div>'
+
          '<div id="labelPlant8GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode2_vision_system_obj_global.status8}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode2_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '</div>' +

          '</div>')
      },

    });
```

```
    alertify.custom = alertify.extend("custom");
    alertify.custom("Mode 2 Activated");
  });

  $("#onMode3").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onMode3',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
        $('#cameraID').attr('src', json.image_json);
        $('#labelMode').text("Mode: " + json.mode_json);
        $('#labelGrid').text("Grid: " + json.grid_json);
        $('#growthContainer').remove()
        $('#growthContainer_permanent').append('<div
id="growthContainer">' +

          '<div class="row">' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 2</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant1}}</div>'
+
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status1}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
```

```
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 2</div>' +
          '<div  id="labelPlant2Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant2}}</div>'
+
          '<div id="labelPlant2GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode3_vision_system_obj_global.status2}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 3</div>' +
          '<div  id="labelPlant3Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant3}}</div>'
+
          '<div id="labelPlant3GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode3_vision_system_obj_global.status3}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
```

```
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 4</div>' +
          '<div  id="labelPlant4Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant4}}</div>'
+
          '<div id="labelPlant4GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode3_vision_system_obj_global.status4}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 5</div>' +
          '<div  id="labelPlant5Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant5}}</div>'
+
          '<div id="labelPlant5GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode3_vision_system_obj_global.status5}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
```

```
          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 6</div>' +
          '<div  id="labelPlant6Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant6}}</div>'
+
          '<div id="labelPlant6GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode3_vision_system_obj_global.status6}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 7</div>' +
          '<div  id="labelPlant7Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant7}}</div>'
+
          '<div id="labelPlant7GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode3_vision_system_obj_global.status7}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image7}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
```

```
          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 8</div>' +
          '<div  id="labelPlant8Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant8}}</div>'
+
          '<div id="labelPlant8GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode3_vision_system_obj_global.status8}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 9</div>' +
          '<div  id="labelPlant9Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode3_vision_system_obj_global.plant9}}</div>'
+
          '<div id="labelPlant9GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode3_vision_system_obj_global.status9}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode3_vision_system_obj_global.image9}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
```

```
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 10</div>' +
          '<div  id="labelPlant10Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant10}}</div>' +
          '<div   id="labelPlant10GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status10}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image10}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 11</div>' +
          '<div  id="labelPlant11Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant11}}</div>' +
          '<div   id="labelPlant11GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status11}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image11}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
```

```
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 12</div>' +
          '<div  id="labelPlant12Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant12}}</div>' +
          '<div   id="labelPlant12GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status12}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image12}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 13</div>' +
          '<div  id="labelPlant13Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant13}}</div>' +
          '<div   id="labelPlant13GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status13}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image13}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
```

```
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 14</div>' +
          '<div  id="labelPlant14Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant14}}</div>' +
          '<div   id="labelPlant14GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status14}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image14}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 15</div>' +
          '<div  id="labelPlant15Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant15}}</div>' +
          '<div   id="labelPlant15GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status15}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image15}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
```

```
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 16</div>' +
          '<div  id="labelPlant16Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant16}}</div>' +
          '<div   id="labelPlant16GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status16}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image16}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 17</div>' +
          '<div  id="labelPlant17Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode3_vision_system_obj_global.plant17}}</div>' +
          '<div   id="labelPlant17GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status17}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image17}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-2 col-md-2 mb-2">' +
```

```
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 18</div>' +
          '<div  id="labelPlant18Growth"  class="h5  mb-0  font-weight-
bold                                                          text-gray-
800">{{mode3_vision_system_obj_global.plant18}}</div>' +
          '<div   id="labelPlant18GrowthStatus"   class="text-s   mb-0
font-weight-normal                                            text-gray-
800">{{mode3_vision_system_obj_global.status18}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode3_vision_system_obj_global.image18}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '</div>' +

          '</div>')
      },

    });

    alertify.custom = alertify.extend("custom");
    alertify.custom("Mode 3 Activated");
  });

  $("#onMode4").click(function(e) {

    e.preventDefault();
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {

        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'onMode4',

      },
      success: function(json) {
        document.getElementById("mainForm").reset();
```

```
        $('#cameraID').attr('src', json.image_json);
        $('#labelMode').text("Mode: " + json.mode_json);
        $('#labelGrid').text("Grid: " + json.grid_json);
        $('#growthContainer').remove()
        $('#growthContainer_permanent').append('<div
id="growthContainer">' +

          '<div class="row">' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          //'<div class="card border-left-success shadow h-100 py-2"
style="background: white;">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 1</div>' +
          '<div  id="labelPlant1Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant1}}</div>'
+
          '<div id="labelPlant1GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode4_vision_system_obj_global.status1}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode4_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 2</div>' +
          '<div  id="labelPlant2Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant2}}</div>'
+
          '<div id="labelPlant2GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode4_vision_system_obj_global.status2}}</div>' +
```

```
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode4_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 3</div>' +
          '<div  id="labelPlant3Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant3}}</div>'
+
          '<div id="labelPlant3GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode4_vision_system_obj_global.status3}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode4_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 4</div>' +
          '<div  id="labelPlant4Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant4}}</div>'
+
          '<div id="labelPlant4GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode4_vision_system_obj_global.status4}}</div>' +
          '</div>' +
```

```
          '<div class="col-auto">' +
          '<i><img     src={{mode4_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 5</div>' +
          '<div  id="labelPlant5Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant5}}</div>'
+
          '<div id="labelPlant5GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode4_vision_system_obj_global.status5}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode4_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 6</div>' +
          '<div  id="labelPlant6Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant6}}</div>'
+
          '<div id="labelPlant6GrowthStatus" class="text-s mb-0 font-
weight-normal                                              text-gray-
800">{{mode4_vision_system_obj_global.status6}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
```

```
          '<i><img     src={{mode4_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 7</div>' +
          '<div  id="labelPlant7Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant7}}</div>'
+
          '<div id="labelPlant7GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode4_vision_system_obj_global.status7}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode4_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 8</div>' +
          '<div  id="labelPlant8Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant8}}</div>'
+
          '<div id="labelPlant8GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode4_vision_system_obj_global.status8}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
```

```
          '<i><img     src={{mode4_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 9</div>' +
          '<div  id="labelPlant9Growth"  class="h5  mb-0  font-weight-
bold  text-gray-800">{{mode4_vision_system_obj_global.plant9}}</div>'
+
          '<div id="labelPlant9GrowthStatus" class="text-s mb-0 font-
weight-normal                                                text-gray-
800">{{mode4_vision_system_obj_global.status9}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img     src={{mode4_vision_system_obj_global.image9}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 10</div>' +
          '<div  id="labelPlant10Growth" class="h5  mb-0  font-weight-
bold                                                          text-gray-
800">{{mode4_vision_system_obj_global.plant10}}</div>' +
          '<div   id="labelPlant10GrowthStatus"   class="text-s   mb-0
font-weight-normal                                            text-gray-
800">{{mode4_vision_system_obj_global.status10}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
```

```
          '<i><img    src={{mode4_vision_system_obj_global.image10}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 11</div>' +
          '<div  id="labelPlant10Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode4_vision_system_obj_global.plant11}}</div>' +
          '<div   id="labelPlant10GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode4_vision_system_obj_global.status11}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
          '<i><img    src={{mode4_vision_system_obj_global.image11}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '<div class="col-xl-3 col-md-3 mb-3">' +
          '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
          '<div class=" card-body">' +
          '<div class="row no-gutters align-items-center">' +
          '<div class="col mr-2">' +
          '<div  class="text-xs  font-weight-bold  text-primary  text-
uppercase mb-1">Plant 12</div>' +
          '<div  id="labelPlant10Growth"  class="h5  mb-0  font-weight-
bold                                                        text-gray-
800">{{mode4_vision_system_obj_global.plant12}}</div>' +
          '<div   id="labelPlant10GrowthStatus"   class="text-s   mb-0
font-weight-normal                                          text-gray-
800">{{mode4_vision_system_obj_global.status12}}</div>' +
          '</div>' +
          '<div class="col-auto">' +
```

```
          '<i><img    src={{mode4_vision_system_obj_global.image12}}
class="singleContainer_icon" /></i>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +
          '</div>' +

          '</div>' +

          '</div>')
      },

    });

    alertify.custom = alertify.extend("custom");
    alertify.custom("Mode 4 Activated");
  });

  function setup() {
    $.ajax({
      type: 'POST',
      url: '{% url "mainPage" %}',
      data: {
        csrfmiddlewaretoken:
$('input[name=csrfmiddlewaretoken]').val(),
        action: 'setup',
      },
      success: function(json) {
        if (json.modeNumber === 1) {
          $('#growthContainer').remove()
          $('#growthContainer_permanent').append('<div
id="growthContainer">' +

            '<div class="row">' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            //'<div class="card border-left-success shadow h-100 py-
2" style="background: white;">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 1</div>' +
            '<div id="labelPlant1Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode1_vision_system_obj_global.plant1}}</div>'
+
```

```
            '<div  id="labelPlant1GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{model_vision_system_obj_global.status1}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                     id="image1ID"
src={{model_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 2</div>' +
            '<div id="labelPlant2Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant2}}</div>'
+
            '<div  id="labelPlant2GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{model_vision_system_obj_global.status2}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                     id="image2ID"
src={{model_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 3</div>' +
```

```
            '<div id="labelPlant3Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant3}}</div>'
+
            '<div  id="labelPlant3GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{model_vision_system_obj_global.status3}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                     id="image3ID"
src={{model_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 4</div>' +
            '<div id="labelPlant4Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant4}}</div>'
+
            '<div  id="labelPlant4GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{model_vision_system_obj_global.status4}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                     id="image4ID"
src={{model_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 5</div>' +
            '<div id="labelPlant5Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant5}}</div>'
+
            '<div  id="labelPlant5GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{model_vision_system_obj_global.status5}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                    id="image5ID"
src={{model_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 6</div>' +
            '<div id="labelPlant6Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant6}}</div>'
+
            '<div  id="labelPlant6GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{model_vision_system_obj_global.status6}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                    id="image6ID"
src={{model_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
```

```
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 7</div>' +
            '<div id="labelPlant7Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant7}}</div>'
+
            '<div  id="labelPlant7GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{model_vision_system_obj_global.status7}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                     id="image7ID"
src={{model_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 8</div>' +
            '<div id="labelPlant8Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant8}}</div>'
+
            '<div  id="labelPlant8GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{model_vision_system_obj_global.status8}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                     id="image8ID"
src={{model_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
```

```
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 9</div>' +
            '<div id="labelPlant9Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{model_vision_system_obj_global.plant9}}</div>'
+
            '<div  id="labelPlant9GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{model_vision_system_obj_global.status9}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                   id="image9ID"
src={{model_vision_system_obj_global.image9}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div  class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 10</div>' +
            '<div id="labelPlant10Growth" class="h5 mb-0 font-weight-
bold                                                        text-gray-
800">{{model_vision_system_obj_global.plant10}}</div>' +
            '<div  id="labelPlant10GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{model_vision_system_obj_global.status10}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img                                  id="image10ID"
src={{model_vision_system_obj_global.image10}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '</div>' +

            '</div>')
```

```
        } else if (json.modeNumber === 2) {
          $('#growthContainer').remove()
          $('#growthContainer_permanent').append('<div
id="growthContainer">' +

            '<div class="row">' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            //'<div class="card border-left-success shadow h-100 py-
2" style="background: white;">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 1</div>' +
            '<div id="labelPlant1Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant1}}</div>'
+
            '<div  id="labelPlant1GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status1}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode2_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 2</div>' +
            '<div id="labelPlant2Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant2}}</div>'
+
            '<div  id="labelPlant2GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status2}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
```

```
            '<i><img   src={{mode2_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 3</div>' +
            '<div id="labelPlant3Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant3}}</div>'
+
            '<div  id="labelPlant3GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status3}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode2_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 4</div>' +
            '<div id="labelPlant4Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant4}}</div>'
+
            '<div  id="labelPlant4GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status4}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
```

```
            '<i><img   src={{mode2_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 5</div>' +
            '<div id="labelPlant5Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant5}}</div>'
+
            '<div  id="labelPlant5GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status5}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode2_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 6</div>' +
            '<div id="labelPlant6Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant6}}</div>'
+
            '<div  id="labelPlant6GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status6}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
```

```
            '<i><img   src={{mode2_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 7</div>' +
            '<div id="labelPlant7Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant7}}</div>'
+
            '<div  id="labelPlant7GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status7}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode2_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 8</div>' +
            '<div id="labelPlant8Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode2_vision_system_obj_global.plant8}}</div>'
+
            '<div  id="labelPlant8GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode2_vision_system_obj_global.status8}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
```

```
            '<i><img   src={{mode2_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '</div>' +

            '</div>')
        } else if (json.modeNumber === 3) {
          $('#growthContainer').remove()
          $('#growthContainer_permanent').append('<div
id="growthContainer">' +

            '<div class="row">' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 2</div>' +
            '<div id="labelPlant1Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant1}}</div>'
+
            '<div  id="labelPlant1GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode3_vision_system_obj_global.status1}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 2</div>' +
            '<div id="labelPlant2Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant2}}</div>'
+
            '<div  id="labelPlant2GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode3_vision_system_obj_global.status2}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 3</div>' +
            '<div id="labelPlant3Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant3}}</div>'
+
            '<div  id="labelPlant3GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode3_vision_system_obj_global.status3}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 4</div>' +
            '<div id="labelPlant4Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant4}}</div>'
+
            '<div  id="labelPlant4GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode3_vision_system_obj_global.status4}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 5</div>' +
            '<div id="labelPlant5Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant5}}</div>'
+
            '<div  id="labelPlant5GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode3_vision_system_obj_global.status5}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 6</div>' +
            '<div id="labelPlant6Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant6}}</div>'
+
            '<div  id="labelPlant6GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode3_vision_system_obj_global.status6}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 7</div>' +
            '<div id="labelPlant7Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant7}}</div>'
+
            '<div  id="labelPlant7GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode3_vision_system_obj_global.status7}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image7}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 8</div>' +
            '<div id="labelPlant8Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant8}}</div>'
+
            '<div  id="labelPlant8GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status8}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 9</div>' +
            '<div id="labelPlant9Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode3_vision_system_obj_global.plant9}}</div>'
+
            '<div  id="labelPlant9GrowthStatus"  class="text-s  mb-0
font-weight-normal                                          text-gray-
800">{{mode3_vision_system_obj_global.status9}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode3_vision_system_obj_global.image9}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 10</div>' +
            '<div id="labelPlant10Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant10}}</div>' +
            '<div  id="labelPlant10GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status10}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image10}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 11</div>' +
            '<div id="labelPlant11Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant11}}</div>' +
            '<div  id="labelPlant11GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status11}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image11}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 12</div>' +
            '<div id="labelPlant12Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant12}}</div>' +
            '<div  id="labelPlant12GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status12}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image12}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 13</div>' +
            '<div id="labelPlant13Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant13}}</div>' +
            '<div  id="labelPlant13GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status13}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image13}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 14</div>' +
            '<div id="labelPlant14Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant14}}</div>' +
            '<div  id="labelPlant14GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status14}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image14}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 15</div>' +
            '<div id="labelPlant15Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant15}}</div>' +
            '<div  id="labelPlant15GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status15}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image15}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 16</div>' +
            '<div id="labelPlant16Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant16}}</div>' +
            '<div  id="labelPlant16GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status16}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image16}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 17</div>' +
            '<div id="labelPlant17Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant17}}</div>' +
            '<div  id="labelPlant17GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status17}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image17}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-2 col-md-2 mb-2">' +
            '<div  class="card  shadow  h-100  py-2"  style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
```

```
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 18</div>' +
            '<div id="labelPlant18Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode3_vision_system_obj_global.plant18}}</div>' +
            '<div  id="labelPlant18GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode3_vision_system_obj_global.status18}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode3_vision_system_obj_global.image18}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '</div>' +

            '</div>')
        } else if (json.modeNumber === 4) {
          $('#growthContainer').remove()
          $('#growthContainer_permanent').append('<div
id="growthContainer">' +


            '<div class="row">' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            //'<div class="card border-left-success shadow h-100 py-
2" style="background: white;">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 1</div>' +
            '<div id="labelPlant1Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant1}}</div>'
+
            '<div  id="labelPlant1GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode4_vision_system_obj_global.status1}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode4_vision_system_obj_global.image1}}
class="singleContainer_icon" /></i>' +
```

```
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 2</div>' +
            '<div id="labelPlant2Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant2}}</div>'
+
            '<div id="labelPlant2GrowthStatus" class="text-s mb-0
font-weight-normal                                    text-gray-
800">{{mode4_vision_system_obj_global.status2}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img src={{mode4_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 3</div>' +
            '<div id="labelPlant3Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant3}}</div>'
+
            '<div id="labelPlant3GrowthStatus" class="text-s mb-0
font-weight-normal                                    text-gray-
800">{{mode4_vision_system_obj_global.status3}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img src={{mode4_vision_system_obj_global.image3}}
class="singleContainer_icon" /></i>' +
            '</div>' +
```

```
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 4</div>' +
            '<div id="labelPlant4Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant4}}</div>'
+
            '<div  id="labelPlant4GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode4_vision_system_obj_global.status4}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode4_vision_system_obj_global.image4}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 5</div>' +
            '<div id="labelPlant5Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant5}}</div>'
+
            '<div  id="labelPlant5GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode4_vision_system_obj_global.status5}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode4_vision_system_obj_global.image5}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
```

```
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 6</div>' +
            '<div id="labelPlant6Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant6}}</div>'
+
            '<div  id="labelPlant6GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode4_vision_system_obj_global.status6}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode4_vision_system_obj_global.image6}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 7</div>' +
            '<div id="labelPlant7Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant7}}</div>'
+
            '<div  id="labelPlant7GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode4_vision_system_obj_global.status7}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode4_vision_system_obj_global.image2}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
```

```
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 8</div>' +
            '<div id="labelPlant8Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant8}}</div>'
+
            '<div  id="labelPlant8GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode4_vision_system_obj_global.status8}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode4_vision_system_obj_global.image8}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 9</div>' +
            '<div id="labelPlant9Growth" class="h5 mb-0 font-weight-
bold text-gray-800">{{mode4_vision_system_obj_global.plant9}}</div>'
+
            '<div  id="labelPlant9GrowthStatus"  class="text-s  mb-0
font-weight-normal                                        text-gray-
800">{{mode4_vision_system_obj_global.status9}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img   src={{mode4_vision_system_obj_global.image9}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
```

```
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 10</div>' +
            '<div id="labelPlant10Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode4_vision_system_obj_global.plant10}}</div>' +
            '<div  id="labelPlant10GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode4_vision_system_obj_global.status10}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode4_vision_system_obj_global.image10}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 11</div>' +
            '<div id="labelPlant10Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode4_vision_system_obj_global.plant11}}</div>' +
            '<div  id="labelPlant10GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode4_vision_system_obj_global.status11}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode4_vision_system_obj_global.image11}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
```

```
            '<div class="col-xl-3 col-md-3 mb-3">' +
            '<div class="card shadow h-100 py-2" style="background:
white;">' +
            '<div class=" card-body">' +
            '<div class="row no-gutters align-items-center">' +
            '<div class="col mr-2">' +
            '<div class="text-xs font-weight-bold text-primary text-
uppercase mb-1">Plant 12</div>' +
            '<div id="labelPlant10Growth" class="h5 mb-0 font-weight-
bold                                                    text-gray-
800">{{mode4_vision_system_obj_global.plant12}}</div>' +
            '<div  id="labelPlant10GrowthStatus"  class="text-s  mb-0
font-weight-normal                                      text-gray-
800">{{mode4_vision_system_obj_global.status12}}</div>' +
            '</div>' +
            '<div class="col-auto">' +
            '<i><img  src={{mode4_vision_system_obj_global.image12}}
class="singleContainer_icon" /></i>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>' +

            '</div>' +

            '</div>')
        }
      },
    });
  }
</script>

</html>
```

Appendix I DATABASE HTML

```
{% extends 'baseLayout.html' %}

{% block content %}


<!-- Nav Item - Dashboard      -->
<li class="nav-item">
  <a class="nav-link" href="{% url 'mainPage' %}">
    <i class="fas fa-fw fa-tachometer-alt"></i>
    <span>Dashboard</span></a>
</li>
<li class="nav-item active">
  <a class="nav-link" href="{% url 'databasePage' %}">
    <i class="fas fa-fw fa-chart-area"></i>
    <span>Database</span>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link" href="{% url 'sensorsPage' %}">
    <i class="fas fa-fw fa-chart-area"></i>
    <span>Sensors</span>
  </a>
</li>

<!-- Divider -->
<hr class="sidebar-divider d-none d-md-block">


</ul>
<!-- End of Sidebar -->

<!-- Content Wrapper -->
<div id="content-wrapper" class="d-flex flex-column">

  <!-- Main Content -->
  <div id="content">

    <!-- Topbar -->
    <nav class="navbar navbar-expand navbar-light bg-white topbar mb-
4 static-top shadow">



      <!-- Topbar Navbar -->
      <ul class="navbar-nav ml-auto">
      </ul>
```

```
    </nav>


    <!-- Begin Page Content -->
    <div class="container-fluid">
      <!-- Page Heading -->
      <h1 class="h3 mb-2 text-gray-800">Database</h1>
      <p  class="mb-4">Monitor  your  progress  through  GrowSmart's
Database</a>.</p>

      <form method='GET' id="databaseForm">
        {% csrf_token %}

        <!--Devices Table-->
        <div class="card shadow mb-4">
          <div class="card-header py-3">
            <h6  class="m-0  font-weight-bold  text-primary">Devices
Table</h6>
          </div>
          <div class="card-body">
            <div class="table-responsive" style="overflow-x: hidden">
              <table  class="table  table-bordered"  id="dataTable"
width="100%" cellspacing="0">
                <thead>
                  <tr>
                    <th>
                      <font size="2">Date&Time</font>
                    </th>
                    <th>
                      <font size="2">Fans</font>
                    </th>
                    <th>
                      <font size="2">W.LED</font>
                    </th>
                    <th>
                      <font size="2">G.Lights</font>
                    </th>
                    <th>
                      <font size="2">Irrigation</font>
                    </th>
                    <th>
                      <font size="2">Calibration</font>
                    </th>
                    <th>
                      <font size="2">Seeding</font>
                    </th>
                  </tr>
                </thead>
```

```
                <tbody>
                  {% for values in devices_obj_global %}
                  <tr>
                    <td>
                      <font size="2">{{values.date}}</font>
                    </td>
                    <td>
                      <font size="2">{{values.fansStatus}}</font>
                    </td>
                    <td>
                      <font
size="2">{{values.whiteLedStatus}}</font>
                    </td>
                    <td>
                      <font size="2">{{values.rgbLedStatus}}</font>
                    </td>
                    <td>
                      <font size="2">{{values.waterStatus}}</font>
                    </td>
                    <td>
                      <font
size="2">{{values.calibrationStatus}}</font>
                    </td>
                    <td>
                      <font size="2">{{values.seedStatus}}</font>
                    </td>
                  </tr>
                  {% endfor %}
                </tbody>
              </table>
            </div>
          </div>
        </div>
        <!--Sensors Table-->
        <div class="card shadow mb-4">
          <div class="card-header py-3">
            <h6  class="m-0  font-weight-bold  text-primary">Sensors
Table</h6>
          </div>
          <div class="card-body">
            <div class="table-responsive" style="overflow-x: hidden">
              <table  class="table  table-bordered"  id="sensorsTable"
width="100%" cellspacing="0">
                <thead>
                  <tr>
                    <th>
                      <font size="2">Date&Time
                    </th>
```

```
                    <th>
                      <font size="2">Temperature
                    </th>
                    <th>
                      <font size="2">
                    </th>
                    <th>
                      <font size="2">Humidity
                    </th>
                    <th>
                      <font size="2">
                    </th>
                    <th>
                      <font size="2">S.M.
                    </th>
                    <th>
                      <font size="2">
                    </th>
                  </tr>
                </thead>
                <tbody>
                  {% for values in sensors_obj_global %}
                  <tr>
                    <td>
                      <font size="2">{{values.date}}</font>
                    </td>
                    <td>
                      <font size="2">{{values.temperature}}</font>
                    </td>
                    <td>
                      <font
size="2">{{values.temperatureStatus}}</font>
                    </td>
                    <td>
                      <font size="2">{{values.humidity}}</font>
                    </td>
                    <td>
                      <font
size="2">{{values.humidityStatus}}</font>
                    </td>
                    <td>
                      <font size="2">{{values.moisture}}</font>
                    </td>
                    <td>
                      <font
size="2">{{values.soilMoistureStatus}}</font>
                    </td>
                  </tr>
                  {% endfor %}
```

```
                </tbody>
              </table>
            </div>
          </div>
        </div>



      </form>

      {% endblock %}
```

Appendix J ARDUINO CODES

ArduinoCNC_MAIN.ino

```
#include "Controls.h"
#include "ModeSelection.h"
#include "Commands.h"
#include <Adafruit_NeoPixel.h>

Adafruit_NeoPixel pixels1(NUMPIXELS, NeoPin_1, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel pixels2(NUMPIXELS, NeoPin_2, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel pixels3(NUMPIXELS, NeoPin_3, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel pixels4(NUMPIXELS, NeoPin_4, NEO_GRB + NEO_KHZ800);

void setup() {

  Serial.begin(9600);

  pinMode(X_DIR, OUTPUT);
  pinMode(Y_DIR, OUTPUT);
  pinMode(Z_DIR, OUTPUT);

  pinMode(X_STP, OUTPUT);
  pinMode(Y_STP, OUTPUT);
  pinMode(Z_STP, OUTPUT);

  pinMode(seedPin, INPUT);
  pinMode(waterPin, INPUT);
  pinMode(calibPin, INPUT);

  pinMode(waterPump, OUTPUT);
  pinMode(vacuumPump, OUTPUT);

  pinMode(EN, OUTPUT);

  pinMode(X_LIM, INPUT_PULLUP);
  pinMode(Y_LIM, INPUT_PULLUP);
  pinMode(Z_LIM, INPUT_PULLUP);

  digitalWrite(EN, LOW);
  digitalWrite(waterPump, HIGH);    // WATER PUMP : OFF
  digitalWrite(vacuumPump, HIGH);   // VACUUM PUMP : OFF

  pixels1.begin();
  pixels2.begin();
  pixels3.begin();
  pixels4.begin();
}
```

```
void loop() {

  if (idle == 0) {
    if (calibrate == 1 && water == 0 && seed == 0) {
      AGE.beginCalbiration();
      //Serial.println("Calibrate: " + String(calibrate) + " Water: "
+ String(water) + " Seed: " + String(seed));    //FOR DEBUGGING
    } else if (calibrate == 0 && water == 0 && seed == 1) {
      AGE.beginSeedDistribution();
      //Serial.println("Calibrate: " + String(calibrate) + " Water: "
+ String(water) + " Seed: " + String(seed));    //FOR DEBUGGING
    } else if (calibrate == 0 && water == 1 && seed == 0) {
      AGE.beginWateringProcess();
      //Serial.println("Calibrate: " + String(calibrate) + " Water: "
+ String(water) + " Seed: " + String(seed));    //FOR DEBUGGING
    } else if (calibrate == 0 && water == 0 && seed == 0) {
      idle = 1;
    }  else {
      Serial.println("ERROR. Can only execute one command per run.");
    }
  } else if (DONE == 0 && idle == 1) {
    if (digitalRead(lightPin) == HIGH) {
      // TURN ON GROW LIGHTS
      for (int i = 0; i < NUMPIXELS; i++) {
        if (i % 4 == 0) {
          pixels1.setPixelColor(i, pixels1.Color(0, 0, 255));
          pixels2.setPixelColor(i, pixels2.Color(0, 0, 255));
          pixels3.setPixelColor(i, pixels3.Color(0, 0, 255));
          pixels4.setPixelColor(i, pixels4.Color(0, 0, 255));
          pixels1.show();
          pixels2.show();
          pixels3.show();
          pixels4.show();
        }
        else {
          pixels1.setPixelColor(i, pixels1.Color(255, 0, 0));
          pixels2.setPixelColor(i, pixels2.Color(255, 0, 0));
          pixels3.setPixelColor(i, pixels3.Color(255, 0, 0));
          pixels4.setPixelColor(i, pixels4.Color(255, 0, 0));
          pixels1.show();
          pixels2.show();
          pixels3.show();
          pixels4.show();
        }
      }
    } else {
      // TURN OFF GROW LIGHTS
      for (int i = 0; i < NUMPIXELS; i++) {
```

```
        pixels1.setPixelColor(i, pixels1.Color(0, 0, 0));
        pixels2.setPixelColor(i, pixels2.Color(0, 0, 0));
        pixels3.setPixelColor(i, pixels3.Color(0, 0, 0));
        pixels4.setPixelColor(i, pixels4.Color(0, 0, 0));
        pixels1.show();
        pixels2.show();
        pixels3.show();
        pixels4.show();
      }
    }

    Serial.println("Nothing to do.");
    DONE = 1;
    delay(3000);

  } else if (DONE == 1 && idle == 1) {

    if (digitalRead(seedPin) == HIGH && digitalRead(waterPin) == LOW
&& digitalRead(calibPin) == LOW) {
      Serial.println("Seed pin set to HIGH.");
      calibrate = 0;
      seed = 1;
      water = 0;
      idle = 0;
      DONE = 0;
      delay(1000);
    } else if (digitalRead(seedPin) == LOW && digitalRead(waterPin)
== HIGH && digitalRead(calibPin) == LOW) {
      Serial.println("Water pin set to HIGH.");
      calibrate = 0;
      seed = 0;
      water = 1;
      idle = 0;
      DONE = 0;
      delay(1000);
    } else if (digitalRead(seedPin) == LOW && digitalRead(waterPin)
== LOW && digitalRead(calibPin) == HIGH) {
      Serial.println("Calibrate pin set to HIGH.");
      calibrate = 1;
      seed = 0;
      water = 0;
      idle = 0;
      DONE = 0;
      delay(1000);
    }
    else {
      Serial.println("No tasks assigned.");
      delay(1000);
      calibrate = 1;
```

```
      water = 0;
      seed = 0;
      DONE = 0;
      idle = 0;
    }
  }

}
```

Commands.h

```
class Commands {

  private:

  public:

    // 3-AXIS CALIBRATION
    void beginCalbiration() {

      // Z-AXIS CALIBRATION
      if (axis == 'Z') {

        if (digitalRead(Z_LIM) == HIGH && Z_POS == false) {
          MPC.step(true, Z_DIR, Z_STP, stps);
          Serial.println("MOVING Z-AXIS UPWARD.");
        } else if (digitalRead(Z_LIM) == LOW && Z_POS == false) {
          Serial.println("Z_POS TOUCHED");
          Z_POS = true;
          Serial.println("MOTION ALONG Z-AXIS STOPPED.");
          Serial.println("Z-AXIS SUCCESSFULLY CALIBRATED.");
          delay(3000);
          axis = 'X';
        }

      }

      // X-AXIS CALIBRATION
      else if (axis == 'X') {

        if (digitalRead(X_LIM) == HIGH && X_POS == false) {
          MPC.step(true, X_DIR, X_STP, stps);
          Serial.println("MOVING X-AXIS TO THE RIGHT.");
        } else if (digitalRead(X_LIM) == LOW && X_POS == false) {
          Serial.println("X_POS TOUCHED");
          X_POS = true;
          if (X_POS == true) {
            MPC.step(false, X_DIR, X_STP, stps);     // GO BACK ONE
STEP
          }
          delay(1000);
```

```
        } else if (digitalRead(X_LIM) == HIGH && X_NEG == false) {
          MPC.step(false, X_DIR, X_STP, stps);
          Serial.println("MOVING X-AXIS TO THE LEFT.");
        } else if (digitalRead(X_LIM) == LOW && X_NEG == false) {
          Serial.println("X_NEG TOUCHED");
          delay(1000);
          Serial.println("Now stopping motion along X-AXIS.");
          X_NEG = true;
          delay(1000);
        } else if (X_POS == true && X_NEG == true) {
          Serial.println("MOTION ALONG X-AXIS STOPPED.");
          Serial.println("X-AXIS SUCCESSFULLY CALIBRATED.");
          delay(3000);
          axis = 'Y';
        }

      }

      // Y-AXIS CALIBRATION
      else if (axis == 'Y') {

        if (digitalRead(Y_LIM) == HIGH && Y_POS == false) {
          MPC.step(true, Y_DIR, Y_STP, stps);
          Serial.println("MOVING Y-AXIS UPWARD.");
        } else if (digitalRead(Y_LIM) == LOW && Y_POS == false) {
          Serial.println("Y_POS TOUCHED");
          Y_POS = true;
          if (Y_POS == true) {
            MPC.step(false, Y_DIR, Y_STP, stps);     // GO BACK ONE
STEP
          }
          delay(1000);
        } else if (digitalRead(Y_LIM) == HIGH && Y_NEG == false) {
          MPC.step(false, Y_DIR, Y_STP, stps);
          Serial.println("MOVING Y-AXIS DOWNWARD.");
        } else if (digitalRead(Y_LIM) == LOW && Y_NEG == false) {
          Serial.println("Y_NEG TOUCHED");
          delay(1000);
          Serial.println("Now stopping motion along Y-AXIS.");
          Y_NEG = true;
          delay(1000);
        } else if (Y_POS == true && Y_NEG == true) {
          Serial.println("MOTION ALONG Y-AXIS STOPPED.");
          Serial.println("Y-AXIS SUCCESSFULLY CALIBRATED.");
          delay(3000);

          // CALIBRATION FINISHED
          calibrate = 0;
          seed = 1;
```

```
          Serial.println("XYZ ROBOT CALIBRATED SUCCESSFULLY");
          water = 0;

          // RESET LIMIT SWTICH CHECKERS
          X_POS = false;
          X_NEG = false;
          Y_POS = false;
          Y_NEG = false;
          Z_POS = false;
        }
      }
    }


    // SEEDING
    void beginSeedDistribution() {

      mode.check();
      mode.assign();

      // PRINT STARTING COORDINATES
      Serial.println((String)"START \n   --> X : " + X_VAL + (String)"
CM , Y : " + Y_VAL +  (String)" CM , Z : " + Z_VAL + (String)" CM");

      // MOVE TO LOCATION OF SEED TRAY
      MPC.step(true, X_DIR, X_STP, stps * (xTray - X_MIN));
      Serial.println((String)"X MOVED " + (xTray - X_MIN) + (String)"
CM.");
      MPC.step(true, Y_DIR, Y_STP, stps * (yTray - Y_MIN));
      Serial.println((String)"Y MOVED " + (yTray - Y_MIN) + (String)"
CM.");

      // GET SEED
      MPC.startVacuumPump();
      MPC.step(false, Z_DIR, Z_STP, 1000);          // Z - DOWN
      delay(1000);
      MPC.step(true, Z_DIR, Z_STP, 1000);           // Z - UP

      for (int i = 0; i < noOfRows; i++) {
        for (int j = 0; j < noOfCols; j++) {
          for (int k = 0; k < noOfSeeds; k++) {

            // MOVE TO iTH ROW (ySeed)
            MPC.step(false, Y_DIR, Y_STP, stps * (yTray - ySeed - (i
* ySpace)));
            Serial.println((String)"Y MOVED " + (yTray - ySeed - (i *
ySpace)) + (String)" CM.");
            delay(500);
```

```
            // MOVE TO jTH COLUMN
            MPC.step(false, X_DIR, X_STP, stps * (xTray - xStart - (j
* xSpace)));
            Serial.println((String)"X MOVED " + (xTray - xStart - (j
* xSpace)) + (String)" CM.");
            delay(2000);
            MPC.step(false, Z_DIR, Z_STP, 1000);    // Z - DOWN
            delay(1000);
            MPC.stopVacuumPump();
            delay(1000);
            MPC.step(true, Z_DIR, Z_STP, 1000);     // Z - UP
            delay(1000);

            if (i == noOfRows - 1 && j == noOfCols - 1 && k ==
noOfSeeds - 1) {

              // MOVE TOWARDS LEFTMOST END
              MPC.step(false, X_DIR, X_STP, stps * (xStart + (j *
xSpace)));
              Serial.println((String)"X MOVED " + (xStart + (j *
xSpace)) + (String)" CM.");

              // MOVE TO ORIGIN
              MPC.step(false, Y_DIR, Y_STP, stps * (ySeed + (i *
ySpace)));
              Serial.println((String)"Y MOVED " + (ySeed + (i *
ySpace)) + (String)" CM.");

            } else {

              // MOVE BACK TO SEED TRAY
              MPC.step(true, X_DIR, X_STP, stps * (xTray - xStart -
(j * xSpace)));
              Serial.println((String)"X MOVED " + (xTray - xStart -
(j * xSpace)) + (String)" CM.");
              do {
                MPC.step(true, X_DIR, X_STP, stps);
              } while (digitalRead(Y_LIM) == HIGH);
              delay(2000);

            }

          }
        }
      }

      Serial.println("Seeding process finished.");
      calibrate = 0;
      seed = 0;
```

```
      water = 0;

    }

    // WATERING

    //      For visualization:      X - starting point
    //
    //          modePin0 = 0:                    |       modePin = 1;
    //                                           |
    //                              <<<<<<<<<<<<<<^                  |
^>>>>>>>>>>>>>>>>
    //                                ^          |             ^
    //                          ^         |           ^<<<<<<<<<<<<<<<<^
    //                            ^         |                          ^
    //              X       >>>>>>>>>>>>>>>^              |          X
>>>>>>>>>>>>>>>>^
    //                                           |

    void beginWateringProcess() {

      mode.check();
      mode.assign();

      // PRINT STARTING COORDINATES
      Serial.println((String)"START \n   --> X : " + X_VAL + (String)"
CM , Y : " + Y_VAL +  (String)" CM , Z : " + Z_VAL + (String)" CM");

      // MOVE TO 1ST ROW AND 1ST COLUMN
      MPC.step(true, X_DIR, X_STP, stps * (xStart - X_MIN));
      Serial.println((String)"X MOVED " + (xStart - X_MIN) + (String)"
CM.");
      MPC.step(true, Y_DIR, Y_STP, stps * (yWater - Y_MIN));
      Serial.println((String)"Y MOVED " + (yWater - Y_MIN) + (String)"
CM.");
      delay(2000);
      MPC.startWaterPump();
      delay(3000);
      MPC.stopWaterPump();

      // TRAVEL TO REMAINING COLUMNS OF 1ST ROW  [L -> R]
      for (int i = 1; i < noOfCols; i++) {
        MPC.step(true, X_DIR, X_STP, stps * xSpace);
        Serial.println((String)"X MOVED " + xSpace + (String)" CM.");
        delay(2000);
        MPC.startWaterPump();
        delay(3000);
        MPC.stopWaterPump();
      }
```

```
      // MOVE TO 2ND ROW
      MPC.step(true, Y_DIR, Y_STP, stps * ySpace);
      Serial.println((String)"Y MOVED " + ySpace + (String)" CM.");
      delay(2000);
      MPC.startWaterPump();
      delay(3000);
      MPC.stopWaterPump();

      // TRAVEL TO REMAINING COLUMNS OF 2ND ROW  [L <- R]
      for (int i = 1; i < noOfCols; i++) {
        MPC.step(false, X_DIR, X_STP, stps * xSpace);
        Serial.println((String)"X MOVED " + xSpace + (String)" CM.");
        delay(2000);
        MPC.startWaterPump();
        delay(3000);
        MPC.stopWaterPump();
      }

      // [FOR TESTING IN WATERING] IF MODE 3 OR MODE 4 IS SELECTED,
CHANGE ARGUMENT.
      if        (digitalRead(modePin1)        ==        1)        {
// READS MODE PIN FROM RPI : if modePin1 = 0; then noOfCols = 2
        //            MOVE            TO            3RD            ROW
modePin1 = 1; then noOfCols = 3
        MPC.step(true, Y_DIR, Y_STP, stps * ySpace);
        Serial.println((String)"Y MOVED " + ySpace + (String)" CM.");
        delay(2000);
        MPC.startWaterPump();
        delay(3000);
        MPC.stopWaterPump();

        // TRAVEL TO REMAINING COLUMNS OF 3RD ROW  [L -> R]
        for (int i = 1; i < noOfCols; i++) {
          MPC.step(true, X_DIR, X_STP, stps * xSpace);
          Serial.println((String)"X  MOVED  "  +  xSpace  +  (String)"
CM.");
          delay(2000);
          MPC.startWaterPump();
          delay(3000);
          MPC.stopWaterPump();
        }

        // MOVE TO 1ST X-COORDINATE (3RD ROW VERSION)
        for (int j = noOfCols; j > 0; j--) {
          MPC.step(false, X_DIR, X_STP, stps * xSpace);
          Serial.println((String)"X  MOVED  "  +  xSpace  +  (String)"
CM.");
        }
```

```
        // MOVE TO LEFTMOST END ALONG X
        MPC.step(false,  X_DIR,  X_STP,  stps  *  (xStart  -  X_MIN  +
noOfCols * xSpace));
        Serial.println((String)"X MOVED " + (xStart - X_MIN + noOfCols
* xSpace) + (String)" CM.");

      } else {
        // MOVE TO LEFTMOST END ALONG X
        MPC.step(false, X_DIR, X_STP, stps * (xStart - X_MIN));
        Serial.println((String)"X  MOVED  "  +  (xStart  -  X_MIN)  +
(String)" CM.");
      }

      // MOVE BACK TO ORIGIN
      for (int i = 1; i < noOfRows; i++) {
        MPC.step(false, Y_DIR, Y_STP, stps * ySpace);
        Serial.println((String)"Y MOVED " + ySpace + (String)" CM.");
      }

      MPC.step(false, Y_DIR, Y_STP, stps * (yWater - Y_MIN));
      Serial.println((String)"Y MOVED " + (yWater - Y_MIN) + (String)"
CM.");

      Serial.println("Watering process finished.");
      calibrate = 0;
      water = 0;
      seed = 0;

    }

};

Commands AGE;
```

Constants.h

```
// STEPPER MOTOR SETTINGS

extern int delayTime = 1000;        // Delay between each pause (uS);
The smaller the value, the faster the revolution.
extern long int stps = 50;          // microsteps per revolution with
1/32 microstepping    [50 steps = 1 cm travel]

// GROW LIGHT SETTINGS

extern int NUMPIXELS = 178;          // No. of LEDs to be opened


// LIMIT SWITCH CHECKERS
```

```
extern boolean X_POS = false;    // X RIGHT LIMIT SWITCH
extern boolean X_NEG = false;    // X LEFT LIMIT SWITCH

extern boolean Y_POS = false;    // Y DOWN LIMIT SWITCH
extern boolean Y_NEG = false;    // Y UP LIMIT SWITCH

extern boolean Z_POS = false;    // Z LIMIT SWITCH

// MIN / MAX REACH FOR EACH AXIS

extern long int X_MIN = 15;
extern long int X_MAX = 95;

extern long int Y_MIN = 12;
extern long int Y_MAX = 43;

extern long int Z_MIN = 0;        // PLEASE CHECK PROTOTYPE
extern long int Z_MAX = 0;

// MOTOR COORDINATES AFTER SUCCESSFUL CALIBRATION

extern long int X_VAL = X_MIN;
extern long int Y_VAL = Y_MIN;
extern long int Z_VAL = Z_MIN;

// AXIS

extern char axis = 'Z';

// RPI HARDCODED COMMANDS        <--  NOTE: CHANGE FROM 0 -> 1 IF
TESTING (See ArduinoCNC_MAIN for instructions.)

extern int calibrate = 0;
extern int water     = 0;
extern int seed      = 0;

// IDLE

extern boolean idle = 1;
extern boolean DONE = 0;

// Z SEEDING DEPTH

extern boolean Z_SEED = 10;
```

Controls.h

```
#include "PinConfiguration.h"
#include "Constants.h"
```

```
class Controls {

  private:

    boolean Dir;
    byte DirPin;
    byte StepperPin;
    long int Steps;

  public:

    // ACTIVATE STEPPER MOTOR

    void step(boolean dir, byte dirPin, byte stepperPin, long int
steps) {

      Dir = dir;
      DirPin = dirPin;
      StepperPin = stepperPin;
      Steps = steps;

      digitalWrite(dirPin, dir);

      for (long int i = 0; i < steps; i++) {
        digitalWrite(stepperPin, HIGH);
        delayMicroseconds(delayTime);
        digitalWrite(stepperPin, LOW);
        delayMicroseconds(delayTime);
      }

      if (calibrate == 1) {
        // DO NOT PRINT COORDINATES DURING CALIBRATION
        //Serial.println((String)"X : " + X_VAL + (String)" CM , Y :
" + Y_VAL +  (String)" CM , Z : " + Z_VAL + (String)" CM");
      } else {
        if (dir) {
          if (dirPin == X_DIR) {
            X_VAL = X_VAL + (steps / 50);
          } else if (dirPin == Y_DIR) {
            Y_VAL = Y_VAL + (steps / 50);
          } else if (dirPin == Z_DIR) {
            Z_VAL = Z_VAL + (steps / 50);
          }
        } else {
          if (dirPin == X_DIR) {
            X_VAL = X_VAL - (steps / 50);
          } else if (dirPin == Y_DIR) {
            Y_VAL = Y_VAL - (steps / 50);
          } else if (dirPin == Z_DIR) {
```

```
            Z_VAL = Z_VAL - (steps / 50);
          }
        }
        Serial.println((String)"X : " + X_VAL + (String)" CM , Y : "
+ Y_VAL +  (String)" CM , Z : " + Z_VAL + (String)" CM");
      }

    }

    // ACTIVATE WATER PUMP

    void startWaterPump() {

      digitalWrite(waterPump, LOW);
      Serial.println("ON: Water Pump");

    }

    void stopWaterPump() {

      digitalWrite(waterPump, HIGH);
      Serial.println("OFF: Water Pump");

    }

    // ACTIVATE VACUUM PUMP

    void startVacuumPump() {

      digitalWrite(vacuumPump, LOW);
      Serial.println("ON: Vacuum Pump");

    }

    void stopVacuumPump() {

      digitalWrite(vacuumPump, HIGH);
      Serial.println("OFF: Vacuum Pump");

    }

};

Controls MPC;
```

ModeSelection.h

```
#include "PlantCoordinates.h"

class ModeSelection {
```

```
  private:

  public:

    int modeSelected;

    void check() {

      // NOTE: CHANGE ARGUMENT AND MODESELECTED VALUE IF TESTING (See
ArduinoCNC_MAIN for instructions.)

      if (digitalRead(modePin1) == 0 && digitalRead(modePin0) == 0) {
// <---- CHANGE ARGUMENT TO THE PROCESS BEING TESTED.
        modeSelected                      =                      1;
// <---- CHANGE THIS VALUE TO 1/2/3/4 WHEN TESTING DIFFERENT MODES.
        Serial.println("MODE 1 SELECTED.");
      } else if (digitalRead(modePin1) == 0 && digitalRead(modePin0)
== 1) {
        modeSelected = 2;
        Serial.println("MODE 2 SELECTED.");
      } else if (digitalRead(modePin1) == 1 && digitalRead(modePin0)
== 0) {
        modeSelected = 3;
        Serial.println("MODE 3 SELECTED.");
      } else if (digitalRead(modePin1) == 1 && digitalRead(modePin0)
== 1) {
        modeSelected = 4;
        Serial.println("MODE 4 SELECTED.");
      } else {
        Serial.println("INVALID MODE SELECTION.");
      }

    }

    void assign() {

      switch (modeSelected) {
        default:
          Serial.println("INVALID!");
          break;
        case 1:
          xSpace    = MODE00_xSpace;
          xStart    = MODE00_xStart;
          ySpace    = MODE00_ySpace;
          ySeed     = MODE00_ySeed;
          yWater    = MODE00_yWater;
          noOfCols  = MODE00_noOfCols;
          noOfRows  = MODE00_noOfRows;
```

```
          break;
        case 2:
          xSpace    = MODE01_xSpace;
          xStart    = MODE01_xStart;
          ySpace    = MODE01_ySpace;
          ySeed     = MODE01_ySeed;
          yWater    = MODE01_yWater;
          noOfCols  = MODE01_noOfCols;
          noOfRows  = MODE01_noOfRows;
          break;
        case 3:
          xSpace    = MODE10_xSpace;
          xStart    = MODE10_xStart;
          ySpace    = MODE10_ySpace;
          ySeed     = MODE10_ySeed;
          yWater    = MODE10_yWater;
          noOfCols  = MODE10_noOfCols;
          noOfRows  = MODE10_noOfRows;
          break;
        case 4:
          xSpace    = MODE11_xSpace;
          xStart    = MODE11_xStart;
          ySpace    = MODE11_ySpace;
          ySeed     = MODE11_ySeed;
          yWater    = MODE11_yWater;
          noOfCols  = MODE11_noOfCols;
          noOfRows  = MODE11_noOfRows;
          break;
      }

    }

};

ModeSelection mode;
```

PinConfiguration.h

```
// STEP PINS
extern int X_STP      = 2;
extern int Y_STP      = 3;
extern int Z_STP      = 4;

// DIRECTION PINS
extern int X_DIR      = 5;
extern int Y_DIR      = 6;
extern int Z_DIR      = 7;

// ENABLE PIN
extern int EN         = 8;
```

```
// LIMIT PINS
extern int X_LIM      = 9;
extern int Y_LIM      = 10;
extern int Z_LIM      = 11;

// GROW LIGHTS
extern int NeoPin_1   = 31;
extern int NeoPin_2   = 33;
extern int NeoPin_3   = 35;
extern int NeoPin_4   = 37;

// RPI-ARDUINO COMMUNICATION PINS
extern int modePin0   =  22;               // MODE SELECTION:    00 -
5 X 2;   10 - 6 X 3
extern int modePin1   =  23;               //                    01 -
4 X 2;   11 - 4 X 3
extern int lightPin   =  39;               // FOR GROWLIGHTS
extern int seedPin    =  38;               // FOR SEEDING
extern int waterPin   =  40;               // FOR WATERING
extern int calibPin   =  42;               // FOR CALIBRATION

// PUMPS
extern int waterPump  =  48;
extern int vacuumPump =  49;
```

PlantCoordinates.h

```
// SETTINGS FOR EACH MODE

extern int xSpace     = 0;              // SPACING ALONG X
extern int xStart     = 0;              // FIRST X-COORDINATE

extern int ySpace     = 0;              // SPACING ALONG Y
extern int ySeed       = 0;                // FIRST Y-COORDINATE FOR
SEEDING
extern int yWater      = 0;                // FIRST Y-COORDINATE FOR
WATERING

extern int zDepth     = 25;             // SEEDING DEPTH

extern int noOfCols   = 0;              // NO. OF COLUMNS
extern int noOfRows   = 0;              // NO. OF ROWS

extern int noOfSeeds   = 3;                // 3 SEEDS PER COORDINATE
<--- CHANGE VALUE IF PLANT REQUIRES DIFFERENT NO. OF SEEDS PER PLOT.

// SEED TRAY COORDINATE

extern int xTray      = X_MAX;
```

```
extern int yTray      = Y_MAX;

// MODE 00: 5 X 2
extern int  MODE00_xSpace   = 15;
extern int  MODE00_xStart   = 25;

extern int  MODE00_ySpace   = 20;
extern int  MODE00_ySeed    = 15;
extern int  MODE00_yWater   = 22;

extern int  MODE00_noOfCols = 5;
extern int  MODE00_noOfRows = 2;

// MODE 01: 4 X 2
extern int  MODE01_xSpace   = 20;
extern int  MODE01_xStart   = 25;

extern int  MODE01_ySpace   = 20;
extern int  MODE01_ySeed    = 15;
extern int  MODE01_yWater   = 22;

extern int  MODE01_noOfCols = 4;
extern int  MODE01_noOfRows = 2;

// MODE 10: 6 x 3
extern int  MODE10_xSpace   = 12;
extern int  MODE10_xStart   = 20;

extern int  MODE10_ySpace   = 10;
extern int  MODE10_ySeed    = 15;
extern int  MODE10_yWater   = 22;

extern int  MODE10_noOfCols = 6;
extern int  MODE10_noOfRows = 3;

// MODE 11: 4 X 3
extern int  MODE11_xSpace   = 20;
extern int  MODE11_xStart   = 25;

extern int  MODE11_ySpace   = 10;
extern int  MODE11_ySeed    = 15;
extern int  MODE11_yWater   = 22;

extern int  MODE11_noOfCols = 4;
extern int  MODE11_noOfRows = 3;
```

Appendix K MONITORING SYSTEM DATA

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	24.6	14.4	0.7211
2	24.6	14.9	0.4123
3	25	14.7	0.3000
4	25	14.9	0.1000
5	24.8	15.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.1 Radial Distances for Planting Ground No. 1

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	39.6	14.4	0.7211
2	39.6	14.9	0.4123
3	40	14.7	0.3000
4	40	14.9	0.1000
5	39.8	15.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.2 Radial Distances for Planting Ground No. 2

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	54.6	14.4	0.7211
2	54.6	14.9	0.4123
3	55	14.7	0.3000
4	55	14.9	0.1000
5	54.8	15.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.3 Radial Distances for Planting Ground No. 3

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	69.6	14.4	0.7211
2	69.6	14.9	0.4123

3	70	14.7	0.3000
4	70	14.9	0.1000
5	69.8	15.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.4 Radial Distances for Planting Ground No. 4

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	84.6	14.4	0.7211
2	84.6	14.9	0.4123
3	85	14.7	0.3000
4	85	14.9	0.1000
5	84.8	15.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.5 Radial Distances for Planting Ground No. 5

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	84.6	34.4	0.7211
2	84.6	34.9	0.4123
3	85	34.7	0.3000
4	85	34.9	0.1000
5	84.8	35.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.6 Radial Distances for Planting Ground No. 6

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	69.6	34.4	0.7211
2	69.6	34.9	0.4123
3	70	34.7	0.3000
4	70	34.9	0.1000
5	69.8	35.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.7 Radial Distances for Planting Ground No. 7

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	54.6	34.4	0.7211
2	54.6	34.9	0.4123
3	55	34.7	0.3000
4	55	34.9	0.1000
5	54.8	35.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.8 Radial Distances for Planting Ground No. 8

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	39.6	34.4	0.7211
2	39.6	34.9	0.4123
3	40	34.7	0.3000
4	40	34.9	0.1000
5	39.8	35.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.9 Radial Distances for Planting Ground No. 9

Trial No.	Location (cm)		Radial Distance (cm)
	X	Y	
1	24.6	34.4	0.7211
2	24.6	34.9	0.4123
3	25	34.7	0.3000
4	25	34.9	0.1000
5	24.8	35.2	0.2828
AVERAGE RADIAL DISTANCE			**0.3633**

Table 6.10 Radial Distances for Planting Ground No. 10

Server: localhost:3306 » Database: thesisDatabase » Table: mainApp_mode1_vision_system

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

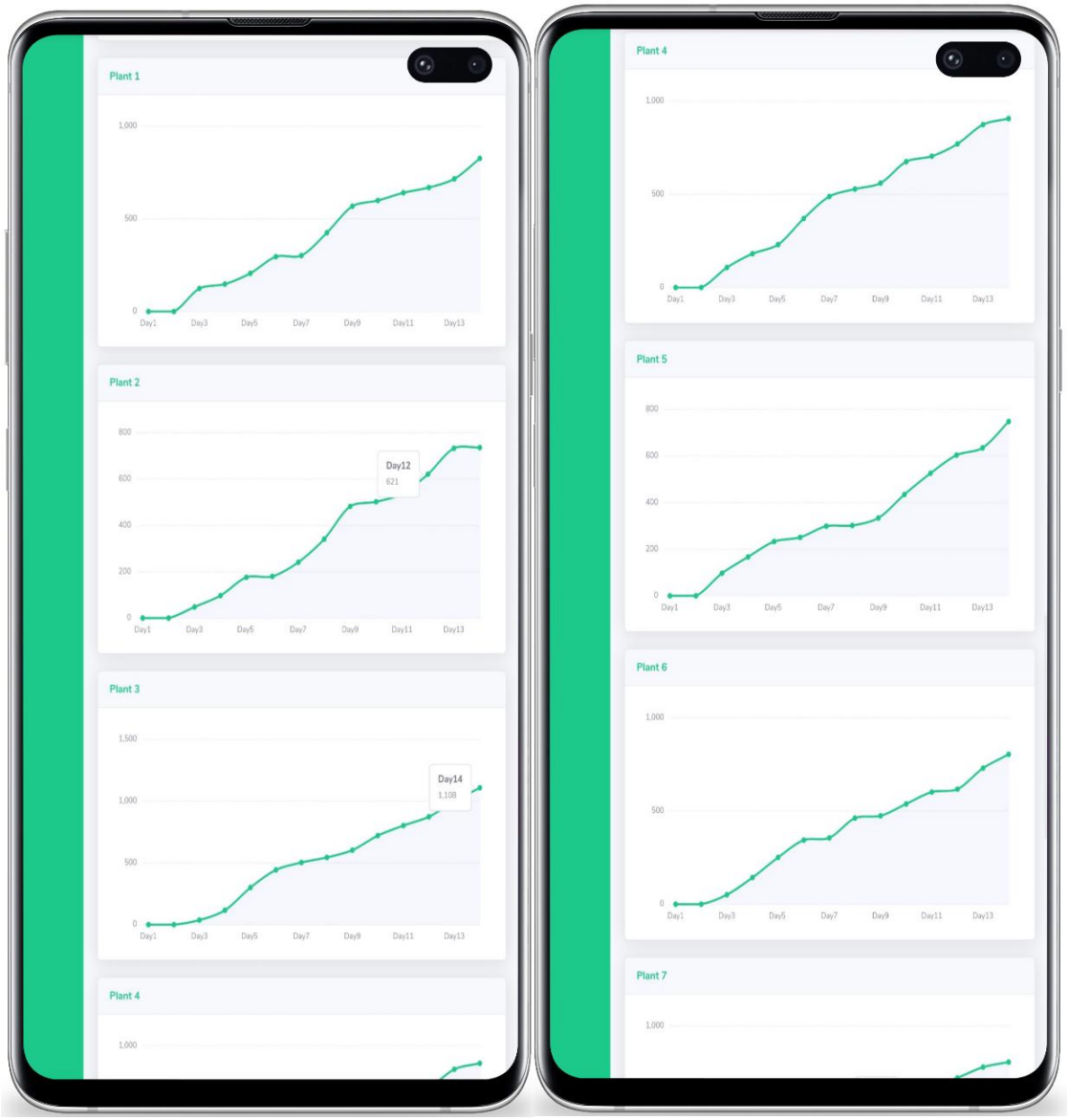
Options

	id	image	plant1	status1	plant2	status2	plant3	status3	plant4	status4	plant5	status5	plant6
Edit Copy Delete	1	../assets/gardenPics/2020-08-31-08_01_32.jpg	0	Early	0	Early	0	Early	0	Early	0	Early	0
Edit Copy Delete	2	../assets/gardenPics/2020-09-01-08_00_40.jpg	0	Early	0	Early	0	Early	0	Early	0	Early	0
Edit Copy Delete	3	../assets/gardenPics/2020-09-02-08_00_22.jpg	124	Early	48	Early	37	Early	107	Early	97	Early	51
Edit Copy Delete	4	../assets/gardenPics/2020-09-03-08_00_13.jpg	148	Early	97	Early	116	Early	181	Early	167	Early	143
Edit Copy Delete	5	../assets/gardenPics/2020-09-04-08_00_05.jpg	206	Early	176	Early	300	Early	230	Early	233	Early	251
Edit Copy Delete	6	../assets/gardenPics/2020-09-05-08_00_39.jpg	296	Early	180	Early	443	Early	370	Early	251	Early	344
Edit Copy Delete	7	../assets/gardenPics/2020-09-06-08_00_54.jpg	302	Early	241	Early	503	Dvlping	488	Dvlping	299	Early	356
Edit Copy Delete	8	../assets/gardenPics/2020-09-07-08_00_12.jpg	425	Early	341	Early	544	Dvlping	528	Dvlping	302	Early	462
Edit Copy Delete	9	../assets/gardenPics/2020-09-08-08_00_27.jpg	568	Dvlping	482	Dvlping	603	Dvlping	560	Dvlping	334	Early	474
Edit Copy Delete	10	../assets/gardenPics/2020-09-09-08_00_09.jpg	599	Dvlping	502	Dvlping	721	Dvlping	675	Dvlping	435	Early	538
Edit Copy Delete	11	../assets/gardenPics/2020-09-10-08_00_37.jpg	641	Dvlping	536	Dvlping	802	Dvlping	704	Dvlping	526	Dvlping	602
Edit Copy Delete	12	../assets/gardenPics/2020-09-11-08_00_32.jpg	669	Dvlping	621	Dvlping	873	Dvlping	770	Dvlping	604	Dvlping	617
Edit Copy Delete	13	../assets/gardenPics/2020-09-12-08_00_15.jpg	716	Dvlping	733	Dvlping	997	Dvlping	874	Dvlping	635	Dvlping	730
Edit Copy Delete	14	../assets/gardenPics/2020-09-13-08_00_29.jpg	826	Dvlping	736	Dvlping	1108	Dvlping	906	Dvlping	748	Dvlping	804

status6	plant7	status7	plant8	status8	plant9	status9	plant10	status10	image1	image2	i
Early	0	Early	0	Early	0	Early	0	Early	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Early	0	Early	0	Early	0	Early	0	Early	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Early	158	Early	39	Early	96	Early	115	Early	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Early	237	Early	119	Early	137	Early	261	Early	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Early	307	Early	296	Early	321	Early	401	Early	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Early	364	Early	320	Early	430	Early	499	Dvlping	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Early	405	Early	356	Early	478	Dvlping	503	Dvlping	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Dvlping	482	Dvlping	389	Early	491	Dvlping	564	Dvlping	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	..
Dvlping	549	Dvlping	456	Dvlping	518	Dvlping	692	Dvlping	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	..
Dvlping	603	Dvlping	468	Dvlping	649	Dvlping	700	Dvlping	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	..
Dvlping	664	Dvlping	517	Dvlping	782	Dvlping	734	Dvlping	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	..
Dvlping	721	Dvlping	632	Dvlping	848	Dvlping	888	Dvlping	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	..
Dvlping	779	Dvlping	664	Dvlping	901	Dvlping	927	Dvlping	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	..
Dvlping	806	Dvlping	798	Dvlping	1088	Dvlping	951	Dvlping	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	..

image4	image5	image6	image7	image8	i
../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/plantIcon.png	../assets/icons/seedIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/seedIcon.png	.
../assets/icons/plantIcon.png	../assets/icons/seedIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	.
../assets/icons/plantIcon.png	../assets/icons/seedIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	.
../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	.
../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	.
../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	.
../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	.

	image9	image10	date
g	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	2020-08-31 08:01:32.871503
g	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	2020-09-01 08:00:40.343019
g	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	2020-09-02 08:00:22.656583
g	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	2020-09-03 08:00:13.267035
g	../assets/icons/seedIcon.png	../assets/icons/seedIcon.png	2020-09-04 08:00:05.594733
g	../assets/icons/seedIcon.png	../assets/icons/plantIcon.png	2020-09-05 08:00:39.300828
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-06 08:00:54.448592
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-07 08:00:12.335560
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-08 08:00:27.410094
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-09 08:00:09.556638
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-10 08:00:37.797899
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-11 08:00:32.960436
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-12 08:00:15.886021
g	../assets/icons/plantIcon.png	../assets/icons/plantIcon.png	2020-09-13 08:00:29.680178


Plant 1
1,000
500
0
Day1
Day3
Day5
Day7
Day9
Day11
Day13
Plant 2
800
600
400
200
0
Day12
621
Plant 3
1,500
1,000
500
0
Day14
1,108
Plant 4
1,000
Plant 4
1,000
500
0
Plant 5
800
600
400
200
0
Plant 6
1,000
500
0
Plant 7
1,000

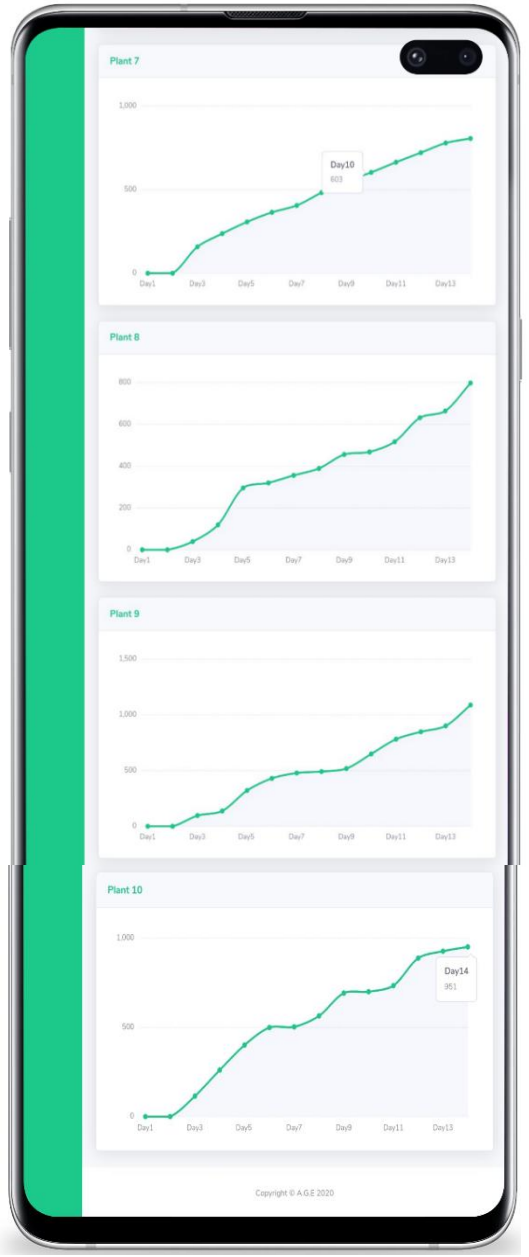


Fig. 6.36 Plant Growth Database and Charts

Server: localhost:3306 » Database: thesisDatabase » Table: mainApp_sensors

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

+ Options

	id	temperature	humidity	moisture	temperatureStatus	humidityStatus	soilMoistureStatus	date
Edit Copy Delete	1	24.6	60	54	Good	Good	Moist	2020-08-31 02:00:05.349241
Edit Copy Delete	2	24.3	61	53	Good	Good	Moist	2020-08-31 03:00:04.349242
Edit Copy Delete	3	24.5	61	53	Good	Good	Moist	2020-08-31 04:00:01.349243
Edit Copy Delete	4	24.3	61	52	Good	Good	Moist	2020-08-31 05:00:07.349244
Edit Copy Delete	5	24.9	60	53	Good	Good	Moist	2020-08-31 06:00:05.349241
Edit Copy Delete	6	24.8	61	51	Good	Good	Moist	2020-08-31 07:00:04.057842
Edit Copy Delete	7	24.4	62	50	Good	Good	Moist	2020-08-31 08:00:02.801342
Edit Copy Delete	8	23.8	60	50	Good	Good	Moist	2020-08-31 09:00:01.592240
Edit Copy Delete	9	24	62	45	Good	Good	Moist	2020-08-31 10:00:00.383742
Edit Copy Delete	10	23.9	63	44	Good	Good	Moist	2020-08-31 11:00:03.778951
Edit Copy Delete	11	24.1	61	90	Good	Good	Wet	2020-08-31 12:00:01.873167
Edit Copy Delete	12	24.2	60	85	Good	Good	Wet	2020-08-31 13:00:00.149221
Edit Copy Delete	13	24	61	81	Good	Good	Wet	2020-08-31 14:00:01.796112
Edit Copy Delete	14	24.5	62	75	Good	Good	Wet	2020-08-31 15:00:03.191266
Edit Copy Delete	15	24.2	61	73	Good	Good	Wet	2020-08-31 16:00:03.138598
Edit Copy Delete	16	24.7	63	71	Good	Good	Wet	2020-08-31 17:00:02.478688
Edit Copy Delete	17	24.5	63	70	Good	Good	Moist	2020-08-31 18:00:00.258587
Edit Copy Delete	18	23.8	65	68	Good	Good	Moist	2020-08-31 19:00:01.647867
Edit Copy Delete	19	24	65	67	Good	Good	Moist	2020-08-31 20:00:03.324479
Edit Copy Delete	20	24.1	65	64	Good	Good	Moist	2020-08-31 21:00:03.760762
Edit Copy Delete	21	23.9	64	63	Good	Good	Moist	2020-08-31 22:00:02.336539
Edit Copy Delete	22	24.1	64	61	Good	Good	Moist	2020-08-31 23:00:00.210020
Edit Copy Delete	23	23.8	61	59	Good	Good	Moist	2020-09-01 00:00:04.107250
Edit Copy Delete	24	23.6	63	58	Good	Good	Moist	2020-09-01 01:00:03.861795
Edit Copy Delete	25	23.7	63	56	Good	Good	Moist	2020-09-01 02:00:02.702513
Edit Copy Delete	26	23.9	63	55	Good	Good	Moist	2020-09-01 03:00:02.356313
Edit Copy Delete	27	23.2	64	53	Good	Good	Moist	2020-09-01 04:00:02.920334
Edit Copy Delete	28	23.5	64	52	Good	Good	Moist	2020-09-01 05:00:02.464906
Edit Copy Delete	29	23.3	65	51	Good	Good	Moist	2020-09-01 06:00:02.988233
Edit Copy Delete	30	23.6	66	51	Good	Good	Moist	2020-09-01 07:00:02.161892
Edit Copy Delete	31	24.1	63	50	Good	Good	Moist	2020-09-01 08:00:02.179158
Edit Copy Delete	32	24.3	64	50	Good	Good	Moist	2020-09-01 09:00:02.272683
Edit Copy Delete	33	24.4	64	47	Good	Good	Moist	2020-09-01 10:00:02.458406
Edit Copy Delete	34	24.8	65	43	Good	Good	Moist	2020-09-01 11:00:02.719136
Edit Copy Delete	35	25	66	89	Good	Good	Wet	2020-09-01 12:00:02.878194
Edit Copy Delete	36	24.7	65	85	Good	Good	Wet	2020-09-01 13:00:02.954274
Edit Copy Delete	37	25.2	66	81	Good	Good	Wet	2020-09-01 14:00:01.521679
Edit Copy Delete	38	24.6	65	80	Good	Good	Wet	2020-09-01 15:00:07.233642
Edit Copy Delete	39	24.2	65	76	Good	Good	Wet	2020-09-01 16:00:07.929330
Edit Copy Delete	40	24.5	65	74	Good	Good	Wet	2020-09-01 17:00:07.990822
Edit Copy Delete	41	23.9	65	72	Good	Good	Wet	2020-09-01 18:00:07.525688
Edit Copy Delete	42	24.2	62	71	Good	Good	Wet	2020-09-01 19:00:07.624913
Edit Copy Delete	43	24.1	66	69	Good	Good	Moist	2020-09-01 20:00:07.166390
Edit Copy Delete	44	23.8	64	69	Good	Good	Moist	2020-09-01 21:00:07.702132
Edit Copy Delete	45	23.6	61	68	Good	Good	Moist	2020-09-01 22:00:03.627566
Edit Copy Delete	46	23.7	63	67	Good	Good	Moist	2020-09-01 23:00:03.731217
Edit Copy Delete	47	23.8	63	65	Good	Good	Moist	2020-09-02 00:00:05.180762
Edit Copy Delete	48	23.3	63	64	Good	Good	Moist	2020-09-02 01:00:05.597070
Edit Copy Delete	49	23.7	64	64	Good	Good	Moist	2020-09-02 02:00:04.159409
Edit Copy Delete	50	24.2	64	61	Good	Good	Moist	2020-09-02 03:00:05.635128
Edit Copy Delete	51	23.9	65	60	Good	Good	Moist	2020-09-02 04:00:05.742737
Edit Copy Delete	52	23.8	63	59	Good	Good	Moist	2020-09-02 05:00:05.585990
Edit Copy Delete	53	24	65	57	Good	Good	Moist	2020-09-02 06:00:05.458447
Edit Copy Delete	54	24.1	65	55	Good	Good	Moist	2020-09-02 07:00:05.812395

Edit	Copy	Delete	55	24	64	53	Good	Good	Moist	2020-09-02 08:00:05.460553
Edit	Copy	Delete	56	26.3	64	48	Good	Good	Moist	2020-09-02 09:00:05.577858
Edit	Copy	Delete	57	26.6	59	45	Good	Good	Moist	2020-09-02 10:00:05.563890
Edit	Copy	Delete	58	24.8	61	39	Good	Good	Moist	2020-09-02 11:00:02.529698
Edit	Copy	Delete	59	25.1	62	86	Good	Good	Wet	2020-09-02 12:00:03.537308
Edit	Copy	Delete	60	25.3	60	84	Good	Good	Wet	2020-09-02 13:00:00.811202
Edit	Copy	Delete	61	25.5	62	83	Good	Good	Wet	2020-09-02 14:00:00.808546
Edit	Copy	Delete	62	25.7	62	77	Good	Good	Wet	2020-09-02 15:00:00.227063
Edit	Copy	Delete	63	25.4	61	75	Good	Good	Wet	2020-09-02 16:00:01.729338
Edit	Copy	Delete	64	25.1	58	74	Good	Good	Wet	2020-09-02 17:00:02.686145
Edit	Copy	Delete	65	23.6	58	72	Good	Good	Wet	2020-09-02 18:00:02.456870
Edit	Copy	Delete	66	23.7	59	70	Good	Good	Moist	2020-09-02 19:00:02.538509
Edit	Copy	Delete	67	24.2	59	68	Good	Good	Moist	2020-09-02 20:00:02.756318
Edit	Copy	Delete	68	23.9	58	65	Good	Good	Moist	2020-09-02 21:00:02.206542
Edit	Copy	Delete	69	23.8	58	64	Good	Good	Moist	2020-09-02 22:00:02.821680
Edit	Copy	Delete	70	24	58	64	Good	Good	Moist	2020-09-02 23:00:03.310389
Edit	Copy	Delete	71	24.1	56	62	Good	Good	Moist	2020-09-03 00:00:07.429680
Edit	Copy	Delete	72	24	56	59	Good	Good	Moist	2020-09-03 01:00:05.295725
Edit	Copy	Delete	73	24.1	57	57	Good	Good	Moist	2020-09-03 02:00:04.757056
Edit	Copy	Delete	74	23.9	61	55	Good	Good	Moist	2020-09-03 03:00:02.705936
Edit	Copy	Delete	75	23.2	61	54	Good	Good	Moist	2020-09-03 04:00:02.697263
Edit	Copy	Delete	76	23.3	60	54	Good	Good	Moist	2020-09-03 05:00:01.238390
Edit	Copy	Delete	77	23.6	60	51	Good	Good	Moist	2020-09-03 06:00:01.477723
Edit	Copy	Delete	78	24.1	59	46	Good	Good	Moist	2020-09-03 07:00:01.639153
Edit	Copy	Delete	79	26.7	59	43	Good	Good	Moist	2020-09-03 08:00:01.669292
Edit	Copy	Delete	80	26.1	59	40	Good	Good	Moist	2020-09-03 09:00:01.194269
Edit	Copy	Delete	81	26.5	59	37	Good	Good	Moist	2020-09-03 10:00:01.479908
Edit	Copy	Delete	82	27.1	60	35	Good	Good	Moist	2020-09-03 11:00:01.685183
Edit	Copy	Delete	83	28.2	59	84	Good	Good	Wet	2020-09-03 12:00:03.676174
Edit	Copy	Delete	84	28.3	59	83	Good	Good	Wet	2020-09-03 13:00:02.767780
Edit	Copy	Delete	85	27.4	59	80	Good	Good	Wet	2020-09-03 14:00:02.805240
Edit	Copy	Delete	86	27.5	59	77	Good	Good	Wet	2020-09-03 15:00:02.140583
Edit	Copy	Delete	87	28.2	60	75	Good	Good	Wet	2020-09-03 16:00:02.576803
Edit	Copy	Delete	88	27.7	60	74	Good	Good	Wet	2020-09-03 17:00:02.164708
Edit	Copy	Delete	89	27.4	58	74	Good	Good	Wet	2020-09-03 18:00:02.805490
Edit	Copy	Delete	90	28.1	56	73	Good	Good	Wet	2020-09-03 19:00:02.194690
Edit	Copy	Delete	91	28	56	71	Good	Good	Wet	2020-09-03 20:00:04.711379
Edit	Copy	Delete	92	27.9	59	69	Good	Good	Moist	2020-09-03 21:00:04.842552
Edit	Copy	Delete	93	27.8	60	67	Good	Good	Moist	2020-09-03 22:00:04.871027
Edit	Copy	Delete	94	27.7	60	66	Good	Good	Moist	2020-09-03 23:00:04.514443
Edit	Copy	Delete	95	27.6	63	63	Good	Good	Moist	2020-09-04 00:00:04.448257
Edit	Copy	Delete	96	27.8	61	60	Good	Good	Moist	2020-09-04 01:00:04.345246
Edit	Copy	Delete	97	27.8	63	59	Good	Good	Moist	2020-09-04 02:00:04.347672
Edit	Copy	Delete	98	28.1	61	58	Good	Good	Moist	2020-09-04 03:00:04.433799
Edit	Copy	Delete	99	28.2	60	56	Good	Good	Moist	2020-09-04 04:00:04.869246
Edit	Copy	Delete	100	28.2	60	53	Good	Good	Moist	2020-09-04 05:00:04.400488
			id	temperature	humidity	moisture	temperatureStatus	humidityStatus	soilMoistureStatus	date
Edit	Copy	Delete	101	28.6	59	52	Good	Good	Moist	2020-09-04 06:00:04.691935
Edit	Copy	Delete	102	27.4	60	50	Good	Good	Moist	2020-09-04 07:00:04.469246
Edit	Copy	Delete	103	26.9	60	44	Good	Good	Moist	2020-09-04 08:00:02.785836
Edit	Copy	Delete	104	26.5	61	41	Good	Good	Moist	2020-09-04 09:00:00.737854
Edit	Copy	Delete	105	27.4	61	38	Good	Good	Moist	2020-09-04 10:00:01.373994
Edit	Copy	Delete	106	27	61	35	Good	Good	Moist	2020-09-04 11:00:06.537623
Edit	Copy	Delete	107	27.3	61	90	Good	Good	Wet	2020-09-04 12:00:06.233258

Console

Edit	Copy	Delete	109	26.6	56	85	Good	Good	Wet	2020-09-04 14:00:06.418868
Edit	Copy	Delete	110	24.8	57	80	Good	Good	Wet	2020-09-04 15:00:06.343462
Edit	Copy	Delete	111	24.6	57	79	Good	Good	Wet	2020-09-04 16:00:06.864789
Edit	Copy	Delete	112	25.4	58	78	Good	Good	Wet	2020-09-04 17:00:06.308529
Edit	Copy	Delete	113	25.2	58	71	Good	Good	Wet	2020-09-04 18:00:06.803275
Edit	Copy	Delete	114	25.1	58	67	Good	Good	Moist	2020-09-04 19:00:03.717545
Edit	Copy	Delete	115	24.5	59	67	Good	Good	Moist	2020-09-04 20:00:04.407243
Edit	Copy	Delete	116	25.8	59	65	Good	Good	Moist	2020-09-04 21:00:02.352593
Edit	Copy	Delete	117	26	59	64	Good	Good	Moist	2020-09-04 22:00:00.981310
Edit	Copy	Delete	118	26.4	58	64	Good	Good	Moist	2020-09-04 23:00:00.370989
Edit	Copy	Delete	119	25.7	59	63	Good	Good	Moist	2020-09-05 00:00:01.620439
Edit	Copy	Delete	120	25.1	61	63	Good	Good	Moist	2020-09-05 01:00:01.936917
Edit	Copy	Delete	121	24.9	61	62	Good	Good	Moist	2020-09-05 02:00:01.699143
Edit	Copy	Delete	122	25.2	60	62	Good	Good	Moist	2020-09-05 03:00:01.161917
Edit	Copy	Delete	123	25	60	60	Good	Good	Moist	2020-09-05 04:00:01.871503
Edit	Copy	Delete	124	24.8	60	59	Good	Good	Moist	2020-09-05 05:00:01.343019
Edit	Copy	Delete	125	24.1	60	57	Good	Good	Moist	2020-09-05 06:00:01.656583
Edit	Copy	Delete	126	24.6	61	55	Good	Good	Moist	2020-09-05 07:00:01.267035
Edit	Copy	Delete	127	24.2	58	49	Good	Good	Moist	2020-09-05 08:00:01.290531
Edit	Copy	Delete	128	25.9	59	47	Good	Good	Moist	2020-09-05 09:00:01.594733
Edit	Copy	Delete	129	26.4	59	45	Good	Good	Moist	2020-09-05 10:00:01.300828
Edit	Copy	Delete	130	26.8	59	43	Good	Good	Moist	2020-09-05 11:00:00.448592
Edit	Copy	Delete	131	25.1	60	82	Good	Good	Wet	2020-09-05 12:00:06.335560
Edit	Copy	Delete	132	26.9	59	80	Good	Good	Wet	2020-09-05 13:00:03.410094
Edit	Copy	Delete	133	26.2	59	75	Good	Good	Wet	2020-09-05 14:00:05.640445
Edit	Copy	Delete	134	26.8	58	73	Good	Good	Wet	2020-09-05 15:00:03.262548
Edit	Copy	Delete	135	26	58	72	Good	Good	Wet	2020-09-05 16:00:03.556638
Edit	Copy	Delete	136	25.6	58	70	Good	Good	Moist	2020-09-05 17:00:03.797899

Console

Edit	Copy	Delete	137	25.9	59	69	Good	Good	Moist	2020-09-05 18:00:03.960436
Edit	Copy	Delete	138	24.5	58	68	Good	Good	Moist	2020-09-05 19:00:03.886021
Edit	Copy	Delete	139	24.7	57	65	Good	Good	Moist	2020-09-05 20:00:03.680178
Edit	Copy	Delete	140	23.8	59	64	Good	Good	Moist	2020-09-05 21:00:00.971767
Edit	Copy	Delete	141	24	57	63	Good	Good	Moist	2020-09-05 22:00:04.652858
Edit	Copy	Delete	142	24.1	58	60	Good	Good	Moist	2020-09-05 23:00:02.523159
Edit	Copy	Delete	143	24.5	59	58	Good	Good	Moist	2020-09-06 00:00:02.838809
Edit	Copy	Delete	144	24.2	60	57	Good	Good	Moist	2020-09-06 01:00:02.228298
Edit	Copy	Delete	145	24.2	59	55	Good	Good	Moist	2020-09-06 02:00:02.965102
Edit	Copy	Delete	146	24.5	59	55	Good	Good	Moist	2020-09-06 03:00:02.420605
Edit	Copy	Delete	147	24.7	60	53	Good	Good	Moist	2020-09-06 04:00:02.721207
Edit	Copy	Delete	148	24.9	63	51	Good	Good	Moist	2020-09-06 05:00:02.884101
Edit	Copy	Delete	149	24.7	62	49	Good	Good	Moist	2020-09-06 06:00:02.428254
Edit	Copy	Delete	150	25.2	62	48	Good	Good	Moist	2020-09-06 07:00:02.475850
Edit	Copy	Delete	151	25	60	45	Good	Good	Moist	2020-09-06 08:00:02.471153
Edit	Copy	Delete	152	24.3	59	43	Good	Good	Moist	2020-09-06 09:00:02.649560
Edit	Copy	Delete	153	24.9	59	40	Good	Good	Moist	2020-09-06 10:00:08.935970
Edit	Copy	Delete	154	25.3	58	39	Good	Good	Moist	2020-09-06 11:00:06.836728
Edit	Copy	Delete	155	25.7	58	85	Good	Good	Wet	2020-09-06 12:00:08.997595
Edit	Copy	Delete	156	25.9	58	83	Good	Good	Wet	2020-09-06 13:00:08.176168
Edit	Copy	Delete	157	26.4	56	80	Good	Good	Wet	2020-09-06 14:00:08.327943
Edit	Copy	Delete	158	26.1	56	77	Good	Good	Wet	2020-09-06 15:00:08.984463
Edit	Copy	Delete	159	26.8	57	73	Good	Good	Wet	2020-09-06 16:00:03.754972
Edit	Copy	Delete	160	25.1	61	71	Good	Good	Wet	2020-09-06 17:00:03.530212
Edit	Copy	Delete	161	24.8	61	68	Good	Good	Moist	2020-09-06 18:00:03.319205
Edit	Copy	Delete	162	24.3	60	66	Good	Good	Moist	2020-09-06 19:00:03.212998

Edit	Copy	Delete	163	24.5	61	64	Good	Good	Moist	2020-09-06 20:00:03.443500
Edit	Copy	Delete	164	24.8	62	63	Good	Good	Moist	2020-09-06 21:00:03.893399
Edit	Copy	Delete	165	24.6	62	61	Good	Good	Moist	2020-09-06 22:00:03.818060
Edit	Copy	Delete	166	24.5	63	58	Good	Good	Moist	2020-09-06 23:00:03.405415
Edit	Copy	Delete	167	24.8	63	58	Good	Good	Moist	2020-09-07 00:00:03.527049
Edit	Copy	Delete	168	24.4	63	57	Good	Good	Moist	2020-09-07 01:00:03.349571
Edit	Copy	Delete	169	24.1	63	55	Good	Good	Moist	2020-09-07 02:00:03.479105
Edit	Copy	Delete	170	24.2	62	52	Good	Good	Moist	2020-09-07 03:00:03.256983
Edit	Copy	Delete	171	23.8	62	51	Good	Good	Moist	2020-09-07 04:00:03.574951
Edit	Copy	Delete	172	24	62	48	Good	Good	Moist	2020-09-07 05:00:03.576377
Edit	Copy	Delete	173	23.3	62	47	Good	Good	Moist	2020-09-07 06:00:03.736256
Edit	Copy	Delete	174	24.2	62	45	Good	Good	Moist	2020-09-07 07:00:03.441831
Edit	Copy	Delete	175	25.1	60	45	Good	Good	Moist	2020-09-07 08:00:03.283319
Edit	Copy	Delete	176	25.8	60	44	Good	Good	Moist	2020-09-07 09:00:03.422085
Edit	Copy	Delete	177	25.6	59	42	Good	Good	Moist	2020-09-07 10:00:00.484269
Edit	Copy	Delete	178	25.7	60	40	Good	Good	Moist	2020-09-07 11:00:00.171271
Edit	Copy	Delete	179	25.8	57	85	Good	Good	Wet	2020-09-07 12:00:00.909855
Edit	Copy	Delete	180	24	55	84	Good	Good	Wet	2020-09-07 13:00:00.123682
Edit	Copy	Delete	181	24.1	55	79	Good	Good	Wet	2020-09-07 14:00:00.814617
Edit	Copy	Delete	182	24.5	55	78	Good	Good	Wet	2020-09-07 15:00:00.234786
Edit	Copy	Delete	183	24.2	55	75	Good	Good	Wet	2020-09-07 16:00:04.892674
Edit	Copy	Delete	184	24.2	55	74	Good	Good	Wet	2020-09-07 17:00:03.806232
Edit	Copy	Delete	185	24.5	53	73	Good	Good	Wet	2020-09-07 18:00:05.226887
Edit	Copy	Delete	186	24	53	72	Good	Good	Wet	2020-09-07 19:00:06.151254
Edit	Copy	Delete	187	24.1	52	69	Good	Good	Moist	2020-09-07 20:00:06.312072
Edit	Copy	Delete	188	24.2	53	68	Good	Good	Moist	2020-09-07 21:00:06.868635
Edit	Copy	Delete	189	24.1	54	67	Good	Good	Moist	2020-09-07 22:00:06.990942
Edit	Copy	Delete	190	23.8	55	65	Good	Good	Moist	2020-09-07 23:00:08.268307
Edit	Copy	Delete	191	23.6	54	64	Good	Good	Moist	2020-09-08 00:00:02.884061
Edit	Copy	Delete	192	23.7	55	63	Good	Good	Moist	2020-09-08 01:00:02.254007
Edit	Copy	Delete	193	23.8	54	60	Good	Good	Moist	2020-09-08 02:00:02.993306
Edit	Copy	Delete	194	23.8	56	59	Good	Good	Moist	2020-09-08 03:00:02.668183
Edit	Copy	Delete	195	23.6	58	55	Good	Good	Moist	2020-09-08 04:00:02.930290
Edit	Copy	Delete	196	23.7	58	54	Good	Good	Moist	2020-09-08 05:00:02.401546
Edit	Copy	Delete	197	23.8	57	53	Good	Good	Moist	2020-09-08 06:00:02.868506
Edit	Copy	Delete	198	23.3	57	50	Good	Good	Moist	2020-09-08 07:00:02.926349
Edit	Copy	Delete	199	24.7	52	45	Good	Good	Moist	2020-09-08 08:00:02.461797
Edit	Copy	Delete	200	24.9	53	44	Good	Good	Moist	2020-09-08 09:00:02.859760
			id	temperature	humidity	moisture	temperatureStatus	humidityStatus	soilMoistureStatus	date
Edit	Copy	Delete	201	24.7	54	41	Good	Good	Moist	2020-09-08 10:00:04.801342
Edit	Copy	Delete	202	24.4	55	39	Good	Good	Moist	2020-09-08 11:00:04.592240
Edit	Copy	Delete	203	24.2	54	83	Good	Good	Wet	2020-09-08 12:00:04.383742
Edit	Copy	Delete	204	24.1	55	82	Good	Good	Wet	2020-09-08 13:00:04.778951
Edit	Copy	Delete	205	24.3	54	80	Good	Good	Wet	2020-09-08 14:00:04.873167
Edit	Copy	Delete	206	23.9	56	78	Good	Good	Wet	2020-09-08 15:00:04.149221
Edit	Copy	Delete	207	23.9	56	74	Good	Good	Wet	2020-09-08 16:00:04.796112
Edit	Copy	Delete	208	24	56	73	Good	Good	Wet	2020-09-08 17:00:04.191266
Edit	Copy	Delete	209	23.8	59	70	Good	Good	Moist	2020-09-08 18:00:07.138598
Edit	Copy	Delete	210	23.8	60	67	Good	Good	Moist	2020-09-08 19:00:09.478688
Edit	Copy	Delete	211	23.3	60	65	Good	Good	Moist	2020-09-08 20:00:02.258587
Edit	Copy	Delete	212	23.7	59	63	Good	Good	Moist	2020-09-08 21:00:01.647867
Edit	Copy	Delete	213	24.2	60	61	Good	Good	Moist	2020-09-08 22:00:01.324479
Edit	Copy	Delete	214	23.9	60	58	Good	Good	Moist	2020-09-08 23:00:01.760762
Edit	Copy	Delete	215	24	61	57	Good	Good	Moist	2020-09-09 00:00:01.336539

Edit	Copy	Delete	216	24.3	62	55	Good	Good	Moist	2020-09-09 01:00:01.210020
Edit	Copy	Delete	217	24.5	62	54	Good	Good	Moist	2020-09-09 02:00:01.107250
Edit	Copy	Delete	218	24.8	63	52	Good	Good	Moist	2020-09-09 03:00:01.861795
Edit	Copy	Delete	219	24.6	63	52	Good	Good	Moist	2020-09-09 04:00:01.702513
Edit	Copy	Delete	220	24.5	63	50	Good	Good	Moist	2020-09-09 05:00:01.356313
Edit	Copy	Delete	221	24.8	64	49	Good	Good	Moist	2020-09-09 06:00:01.920334
Edit	Copy	Delete	222	24.4	64	48	Good	Good	Moist	2020-09-09 07:00:01.464906
Edit	Copy	Delete	223	24.2	60	46	Good	Good	Moist	2020-09-09 08:00:01.988233
Edit	Copy	Delete	224	23.8	60	46	Good	Good	Moist	2020-09-09 09:00:01.161892
Edit	Copy	Delete	225	23.3	63	45	Good	Good	Moist	2020-09-09 10:00:01.179158
Edit	Copy	Delete	226	23.8	61	44	Good	Good	Moist	2020-09-09 11:00:04.272683
Edit	Copy	Delete	227	23.6	61	87	Good	Good	Wet	2020-09-09 12:00:03.458406
Edit	Copy	Delete	228	23.7	61	86	Good	Good	Wet	2020-09-09 13:00:03.719136
Edit	Copy	Delete	229	23.8	60	84	Good	Good	Wet	2020-09-09 14:00:03.878194
Edit	Copy	Delete	230	23.8	60	82	Good	Good	Wet	2020-09-09 15:00:03.954274
Edit	Copy	Delete	231	23.6	59	78	Good	Good	Wet	2020-09-09 16:00:03.521679
Edit	Copy	Delete	232	23.8	60	73	Good	Good	Wet	2020-09-09 17:00:03.233642
Edit	Copy	Delete	233	24.1	60	72	Good	Good	Wet	2020-09-09 18:00:03.836728
Edit	Copy	Delete	234	23.6	62	69	Good	Good	Moist	2020-09-09 19:00:03.997595
Edit	Copy	Delete	235	23.7	62	68	Good	Good	Moist	2020-09-09 20:00:04.176168
Edit	Copy	Delete	236	23.8	59	67	Good	Good	Moist	2020-09-09 21:00:05.327943
Edit	Copy	Delete	237	23.3	59	65	Good	Good	Moist	2020-09-09 22:00:05.984463
Edit	Copy	Delete	238	23.7	60	64	Good	Good	Moist	2020-09-09 23:00:05.754972
Edit	Copy	Delete	239	24.2	62	63	Good	Good	Moist	2020-09-10 00:00:05.530212
Edit	Copy	Delete	240	24.1	62	60	Good	Good	Moist	2020-09-10 01:00:05.319205
Edit	Copy	Delete	241	23.8	63	57	Good	Good	Moist	2020-09-10 02:00:05.212998
Edit	Copy	Delete	242	23.6	64	55	Good	Good	Moist	2020-09-10 03:00:05.443500
Edit	Copy	Delete	243	23.7	65	54	Good	Good	Moist	2020-09-10 04:00:05.893399
Edit	Copy	Delete	244	23.8	65	53	Good	Good	Moist	2020-09-10 05:00:05.818060
Edit	Copy	Delete	245	23.8	66	51	Good	Good	Moist	2020-09-10 06:00:05.405415
Edit	Copy	Delete	246	23.6	65	49	Good	Good	Moist	2020-09-10 07:00:05.527049
Edit	Copy	Delete	247	24.2	60	44	Good	Good	Moist	2020-09-10 08:00:05.349571
Edit	Copy	Delete	248	24	60	43	Good	Good	Moist	2020-09-10 09:00:05.479105
Edit	Copy	Delete	249	24.1	60	39	Good	Good	Moist	2020-09-10 10:00:11.256983
Edit	Copy	Delete	250	24.5	59	37	Good	Good	Moist	2020-09-10 11:00:06.574951
Edit	Copy	Delete	251	24.4	60	87	Good	Good	Wet	2020-09-10 12:00:07.576377
Edit	Copy	Delete	252	24.8	58	85	Good	Good	Wet	2020-09-10 13:00:07.736256
Edit	Copy	Delete	253	23.9	59	84	Good	Good	Wet	2020-09-10 14:00:07.441831
Edit	Copy	Delete	254	23.2	59	83	Good	Good	Wet	2020-09-10 15:00:07.283319
Edit	Copy	Delete	255	23.3	59	78	Good	Good	Wet	2020-09-10 16:00:07.422085
Edit	Copy	Delete	256	23.6	58	77	Good	Good	Wet	2020-09-10 17:00:07.484269
Edit	Copy	Delete	257	24.2	59	75	Good	Good	Wet	2020-09-10 18:00:03.171271
Edit	Copy	Delete	258	24.5	59	73	Good	Good	Wet	2020-09-10 19:00:02.909855
Edit	Copy	Delete	259	25.8	59	71	Good	Good	Wet	2020-09-10 20:00:03.123682
Edit	Copy	Delete	260	26	59	69	Good	Good	Moist	2020-09-10 21:00:03.814617
Edit	Copy	Delete	261	26.4	59	67	Good	Good	Moist	2020-09-10 22:00:03.234786
Edit	Copy	Delete	262	25.7	59	66	Good	Good	Moist	2020-09-10 23:00:03.962979
Edit	Copy	Delete	263	25.1	59	65	Good	Good	Moist	2020-09-11 00:00:03.179069
Edit	Copy	Delete	264	24.9	60	63	Good	Good	Moist	2020-09-11 01:00:03.452238
Edit	Copy	Delete	265	25.2	60	60	Good	Good	Moist	2020-09-11 02:00:03.938852
Edit	Copy	Delete	266	25	61	58	Good	Good	Moist	2020-09-11 03:00:03.792736
Edit	Copy	Delete	267	24.8	60	56	Good	Good	Moist	2020-09-11 04:00:03.305435
Edit	Copy	Delete	268	24.1	61	55	Good	Good	Moist	2020-09-11 05:00:03.708164

Console

| | | |

Edit Copy Delete	318	24.1	60	48	Good	Good	Moist	2020-09-13 07:00:02.981310
Edit Copy Delete	319	25.2	58	45	Good	Good	Moist	2020-09-13 08:00:02.370989
Edit Copy Delete	320	25	59	44	Good	Good	Moist	2020-09-13 09:00:06.115792
Edit Copy Delete	321	24.8	60	43	Good	Good	Moist	2020-09-13 10:00:06.364347
Edit Copy Delete	322	24.1	59	40	Good	Good	Moist	2020-09-13 11:00:06.715409
Edit Copy Delete	323	24.6	59	84	Good	Good	Wet	2020-09-13 12:00:06.565403
Edit Copy Delete	324	23.9	60	82	Good	Good	Wet	2020-09-13 13:00:06.675450
Edit Copy Delete	325	24.1	63	81	Good	Good	Wet	2020-09-13 14:00:03.308443

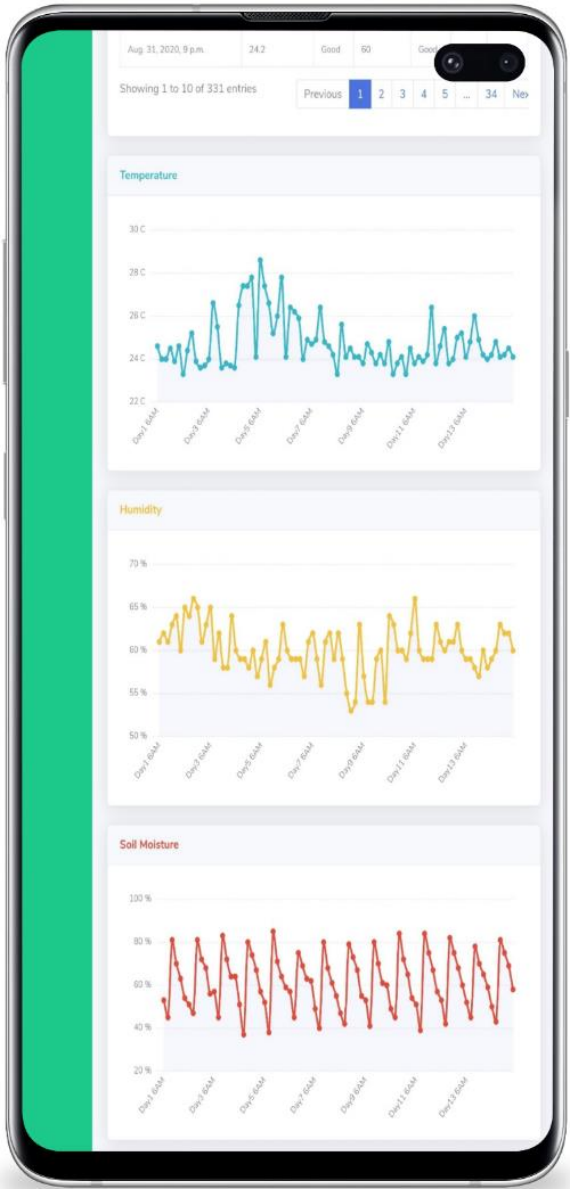


Fig. 6.37 Sensors Database and Charts

Plant	Day							
	1	6	11	16	21	26	31	36
1	0	569	610	1043	1140	1146	1774	3316
2	0	176	282	633	661	686	1242	1800
3	0	396	556	945	1071	1064	1089	1876
4	0	230	360	666	696	1012	1414	2332
5	0	233	334	635	744	845	1252	1993
6	0	351	374	424	478	609	997	1999
7	0	157	298	315	330	428	986	1359
8	0	163	301	697	782	864	1038	1176
9	0	421	518	1101	1095	1173	1228	1304
10	0	501	492	559	581	748	1115	1326

Table 6.12 Tabulated Area Occupied per Plant (Trial 1)

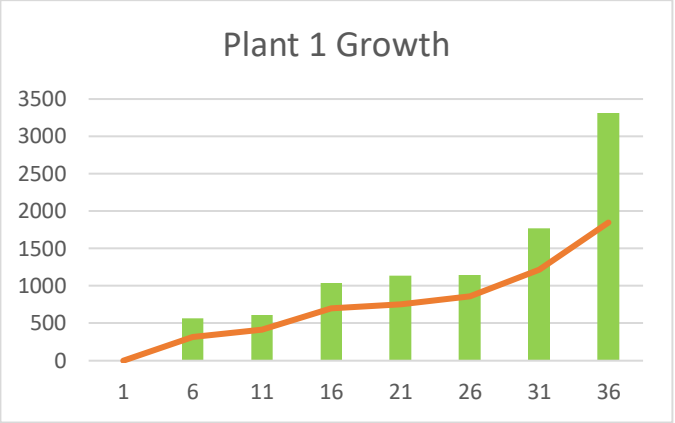


Fig. 6.39 Plant 1 Growth Rate Chart

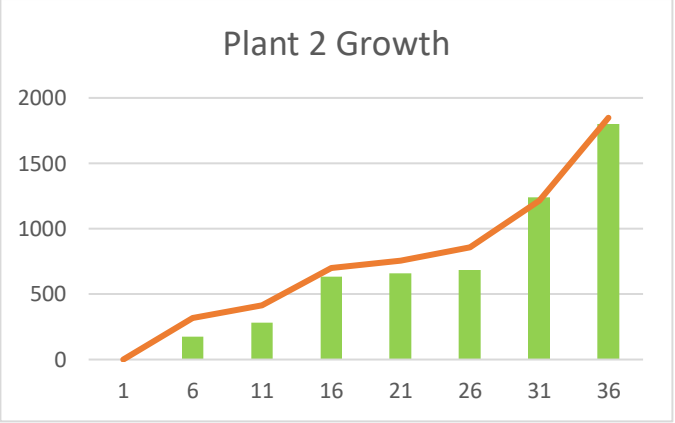


Fig. 6.40 Plant 2 Growth Rate Chart

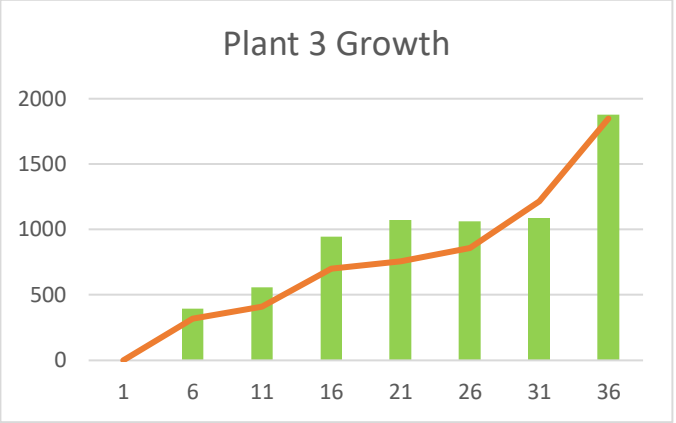


Fig. 6.41 Plant 3 Growth Rate Chart

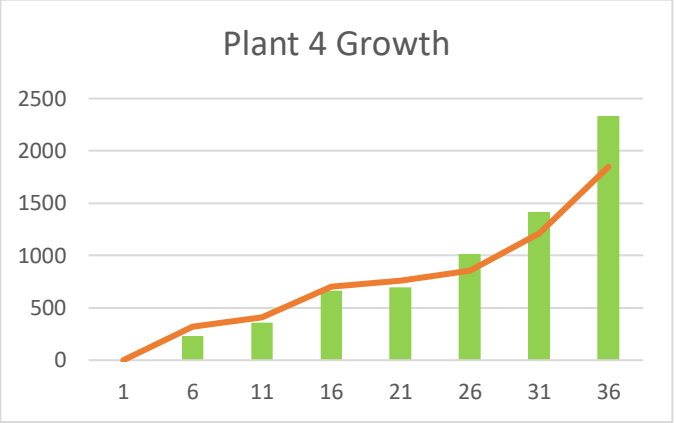


Fig. 6.42 Plant 4 Growth Rate Chart

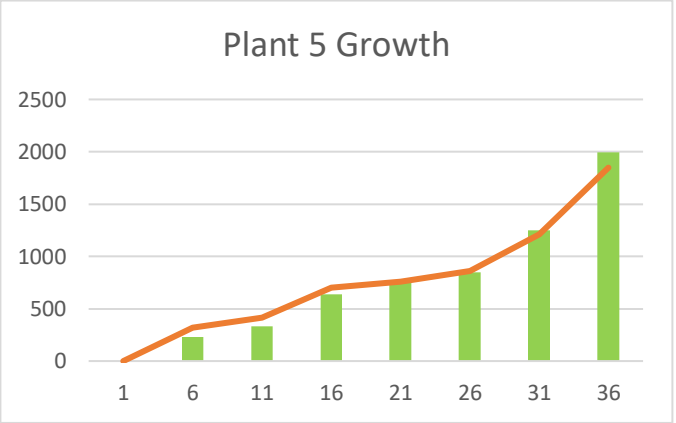


Fig. 6.43 Plant 5 Growth Rate Chart

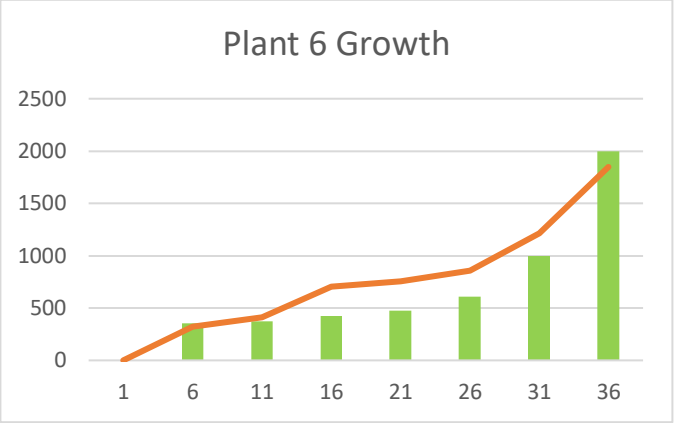


Fig. 6.44 Plant 6 Growth Rate Chart

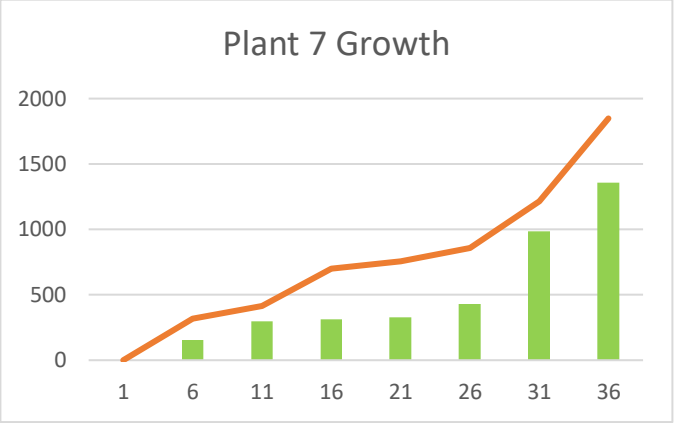


Fig. 6.45 Plant 7 Growth Rate Chart

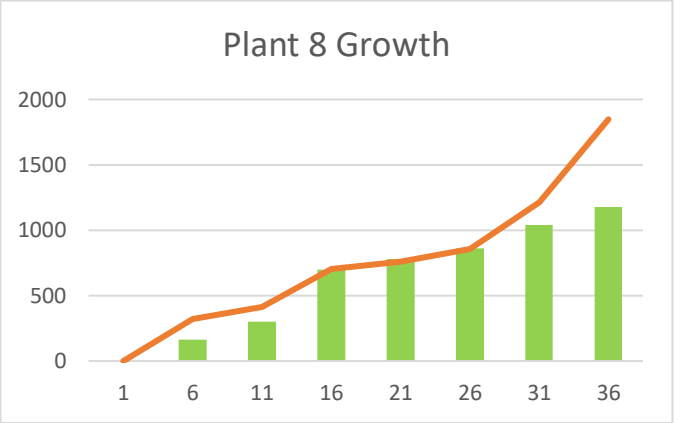


Fig. 6.46 Plant 8 Growth Rate Chart

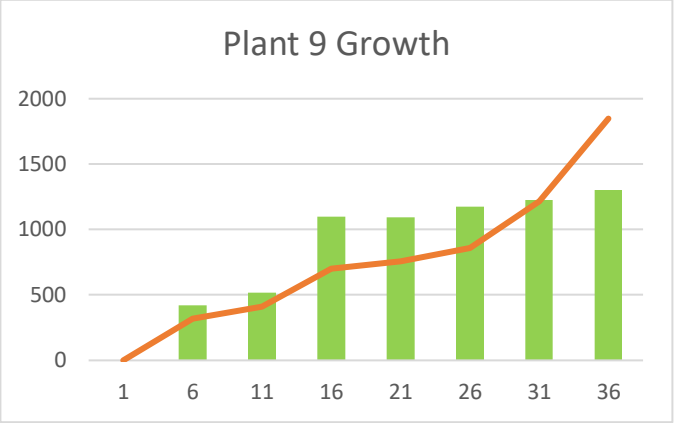


Fig. 6.47 Plant 9 Growth Rate Chart

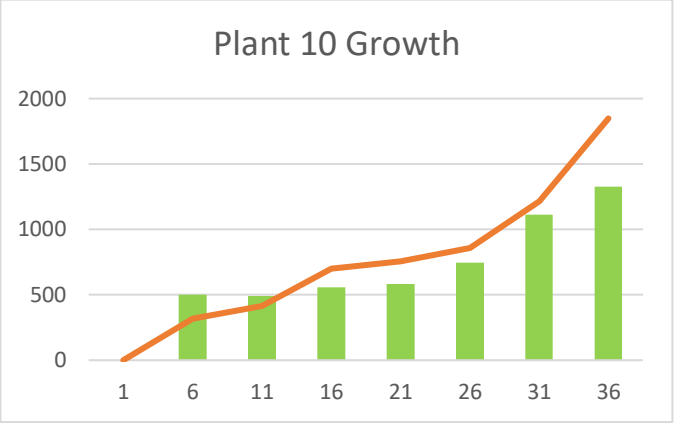


Fig. 6.48 Plant 10 Growth Rate Chart

Plant	**Day**													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	0	0	124	148	206	296	302	425	568	599	641	669	716	826
2	0	0	48	97	176	180	241	341	482	502	536	621	733	736
3	0	0	37	116	300	443	503	544	603	721	802	873	997	1108
4	0	0	107	181	230	370	488	528	560	675	704	770	874	906
5	0	0	97	167	233	251	299	302	334	435	526	604	635	748
6	0	0	51	143	251	344	356	462	474	538	602	617	730	804
7	0	0	158	237	307	364	405	482	549	603	664	721	779	806
8	0	0	39	119	296	320	356	389	456	468	517	632	664	798

9	0	0	96	137	321	430	478	491	518	649	782	848	901	1088
10	0	0	115	261	401	499	503	564	692	700	734	888	927	951

Table 6.14 Tabulated Area Occupied per Plant (Trial 2)

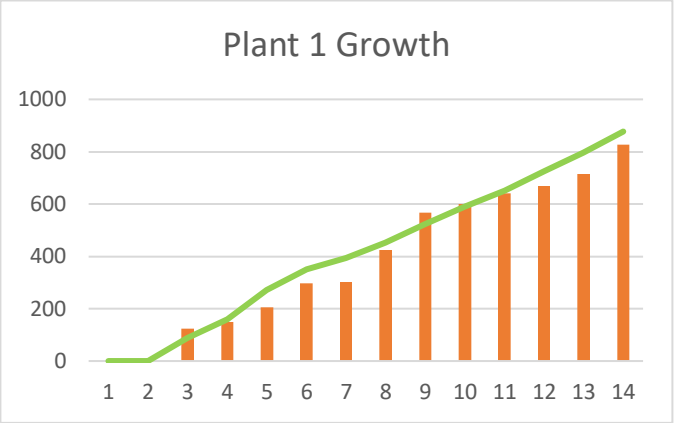


Fig. 6.50 Plant 1 Growth Rate Chart

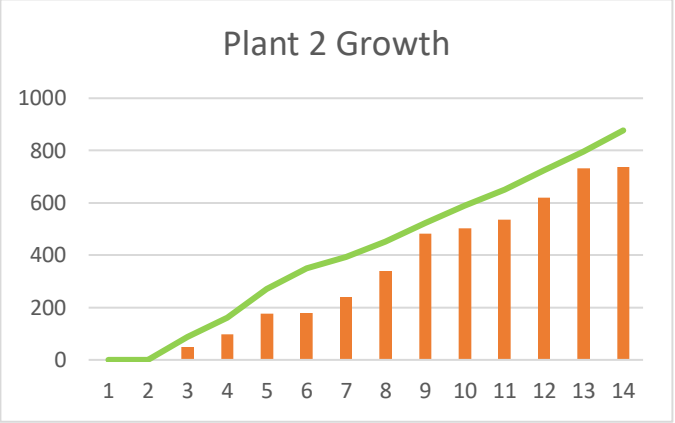


Fig. 6.51 Plant 2 Growth Rate Chart

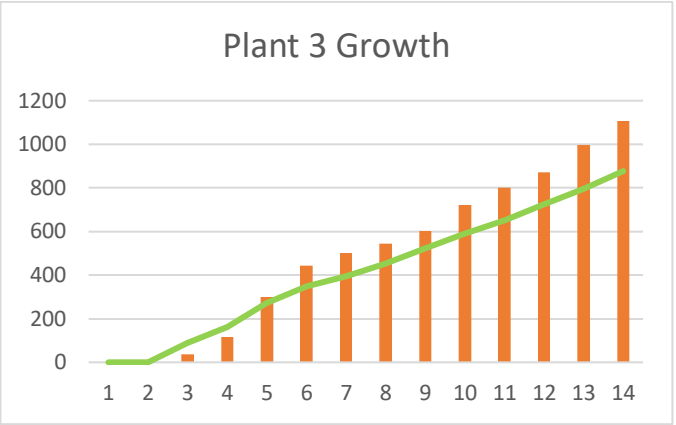


Fig. 6.52 Plant 3 Growth Rate Chart

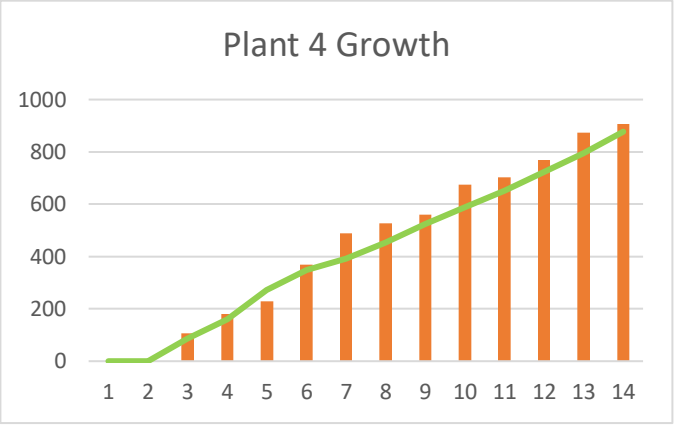


Fig. 6.53 Plant 4 Growth Rate Chart

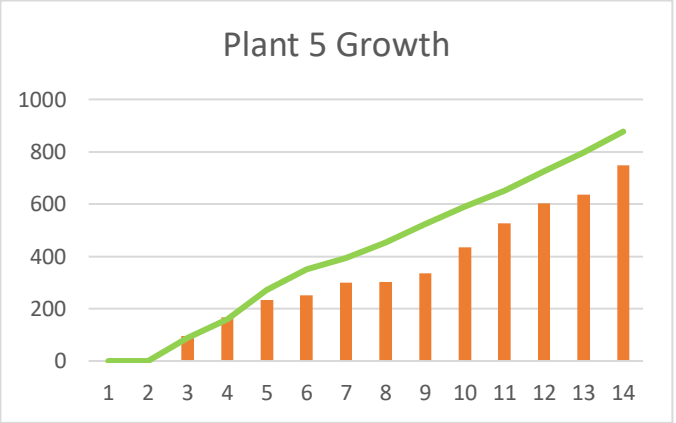


Fig. 6.54 Plant 5 Growth Rate Chart

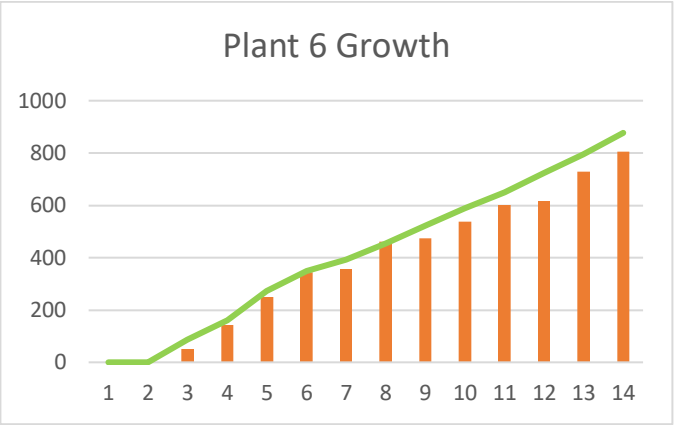


Fig. 6.55 Plant 6 Growth Rate Chart

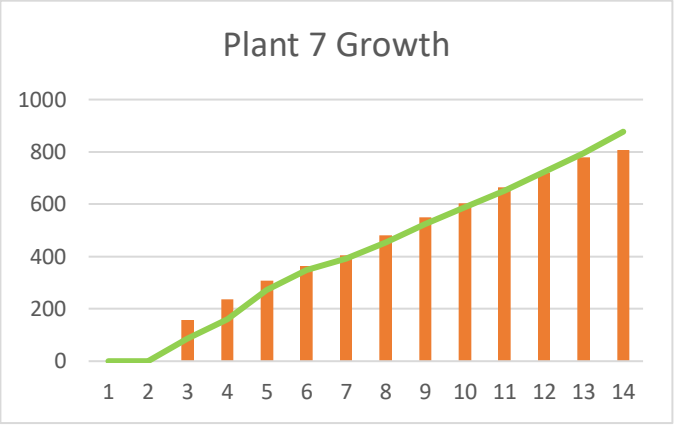


Fig. 6.56 Plant 7 Growth Rate Chart

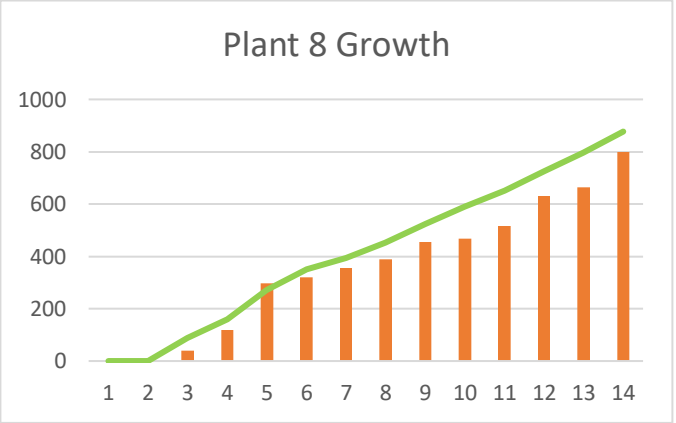


Fig. 6.57 Plant 8 Growth Rate Chart

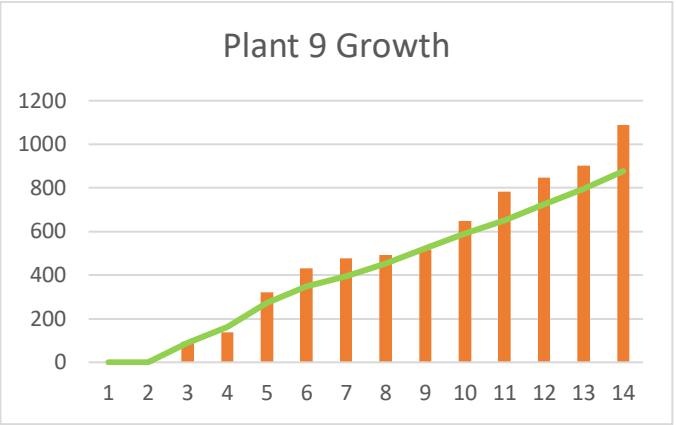


Fig. 6.58 Plant 9 Growth Rate Chart

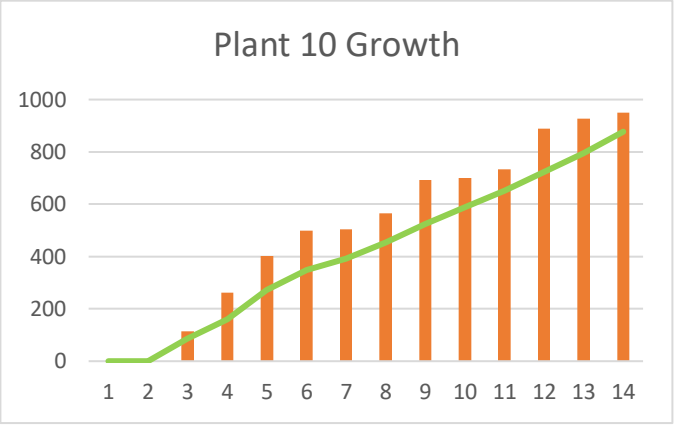


Fig. 6.59 Plant 10 Growth Rate Chart

	Trial	Actual Soil Moisture	Soil Moisture Sensor Output	Percentage Error
Dry	**1**	14%	15%	7.143%
	2	17%	18%	5.882%
	3	18%	18%	0.000%
	4	21%	20%	4.762%
	5	27%	28%	3.704%
Moist	**6**	34%	33%	2.941%
	7	38%	36%	5.263%
	8	46%	44%	4.348%
	9	51%	52%	1.961%

	10	59%	59%	0.000%
	11	63%	62%	1.587%
Wet	**12**	71%	69%	2.817%
	13	79%	77%	2.532%
	14	83%	81%	2.410%
	15	88%	89%	1.136%

Table 6.15 Soil Moisture Sensor Comparison

Trial	Actual Temperature	DHT22 Temperature Sensor Output	Percent Error
1	15.5°C	15.8°C	1.935%
2	16.0°C	16.7°C	4.375%
3	16.5°C	17.2°C	4.242%
4	17.0°C	17.9°C	5.294%
5	17.5°C	16.9°C	3.429%
6	18.0°C	17.5°C	2.778%
7	20.0°C	20.7°C	3.500%
8	23.0°C	23.4°C	1.739%
9	25.0°C	25.6°C	2.400%
10	26.0°C	26.9°C	3.462%
11	27.5°C	27.8°C	1.091%
12	28.0°C	29.2°C	4.286%
13	32.0°C	33.8°C	5.625%
14	32.5°C	33.9°C	4.308%
15	34.0°C	34.8°C	2.353%

Table 6.16 Temperature Sensor Comparison

Trial	Actual Humidity	DHT22 Humidity Output	Percent Error
1	57.0%	54.8%	3.860%
2	58.5%	59.1%	1.026%
3	59.0%	60.1%	1.864%
4	59.5%	60.2%	1.176%
5	61.5%	60.4%	1.789%

6	62.0%	61.8%	0.323%
7	63.5%	62.0%	2.362%
8	65.0%	63.8%	1.846%
9	67.0%	65.8%	1.791%
10	68.0%	67.5%	0.735%
11	69.5%	68.8%	1.007%
12	71.0%	70.2%	1.127%
13	72.0%	71.5%	0.694%

Table 6.17 Humidity Sensor Comparison

Day	Time	Actual Soil Moisture (%)	Soil Moisture Sensor Output (%)	% Error	Daily Average % Error
1	8:00	52	50	3.85%	2.671%
	9:00	49	50	2.04%	
	10:00	46	45	2.17%	
	11:00	42	44	4.76%	
	12:00	88	90	2.27%	
	13:00	84	85	1.19%	
	14:00	81	81	0.00%	
	15:00	76	75	1.32%	
	16:00	71	73	2.82%	
	17:00	69	71	2.90%	
	18:00	66	70	6.06%	
2	8:00	51	50	1.96%	2.696%
	9:00	49	50	2.04%	
	10:00	45	47	4.44%	
	11:00	43	43	0.00%	
	12:00	90	89	1.11%	
	13:00	88	85	3.41%	
	14:00	83	81	2.41%	
	15:00	79	80	1.27%	
	16:00	75	76	1.33%	
	17:00	71	74	4.23%	
	18:00	67	72	7.46%	
3	8:00	50	53	6.00%	3.789%

	9:00	45	48	6.67%	
	10:00	44	45	2.27%	
	11:00	36	39	8.33%	
	12:00	88	86	2.27%	
	13:00	87	84	3.45%	
	14:00	83	83	0.00%	
	15:00	78	77	1.28%	
	16:00	74	75	1.35%	
	17:00	70	74	5.71%	
	18:00	69	72	4.35%	
4	8:00	40	43	7.50%	4.175%
	9:00	39	40	2.56%	
	10:00	35	37	5.71%	
	11:00	33	35	6.06%	
	12:00	82	84	2.44%	
	13:00	80	83	3.75%	
	14:00	78	80	2.56%	
	15:00	74	77	4.05%	
	16:00	74	75	1.35%	
	17:00	71	74	4.23%	
	18:00	70	74	5.71%	
5	8:00	41	44	7.32%	6.456%
	9:00	38	41	7.89%	
	10:00	37	38	2.70%	
	11:00	37	35	5.41%	
	12:00	88	90	2.27%	
	13:00	85	88	3.53%	
	14:00	78	85	8.97%	
	15:00	77	80	3.90%	
	16:00	74	79	6.76%	
	17:00	69	78	13.04%	
	18:00	65	71	9.23%	
6	8:00	50	49	2.00%	2.794%
	9:00	48	47	2.08%	

	10:00	42	45	7.14%	
	11:00	41	43	4.88%	
	12:00	80	82	2.50%	
	13:00	77	80	3.90%	
	14:00	74	75	1.35%	
	15:00	75	73	2.67%	
	16:00	73	72	1.37%	
	17:00	71	70	1.41%	
	18:00	70	69	1.43%	
7	8:00	42	45	7.14%	5.308%
	9:00	42	43	2.38%	
	10:00	41	40	2.44%	
	11:00	40	39	2.50%	
	12:00	82	85	3.66%	
	13:00	78	83	6.41%	
	14:00	77	80	3.90%	
	15:00	73	77	5.48%	
	16:00	67	73	8.96%	
	17:00	66	71	7.58%	
	18:00	63	68	7.94%	
8	8:00	47	45	4.26%	3.255%
	9:00	43	44	2.33%	
	10:00	42	42	0.00%	
	11:00	37	40	8.11%	
	12:00	83	85	2.41%	
	13:00	81	84	3.70%	
	14:00	77	79	2.60%	
	15:00	75	78	4.00%	
	16:00	73	75	2.74%	
	17:00	73	74	1.37%	
	18:00	70	73	4.29%	
9	8:00	51	45	11.76%	7.772%
	9:00	50	44	12.00%	
	10:00	48	41	14.58%	

	11:00	48	39	18.75%	
	12:00	81	83	2.47%	
	13:00	79	82	3.80%	
	14:00	75	80	6.67%	
	15:00	75	78	4.00%	
	16:00	71	74	4.23%	
	17:00	70	73	4.29%	
	18:00	68	70	2.94%	
10	8:00	48	46	4.17%	2.142%
	9:00	45	46	2.22%	
	10:00	45	45	0.00%	
	11:00	43	44	2.33%	
	12:00	90	87	3.33%	
	13:00	88	86	2.27%	
	14:00	85	84	1.18%	
	15:00	81	82	1.23%	
	16:00	77	78	1.30%	
	17:00	75	73	2.67%	
	18:00	70	72	2.86%	
11	8:00	49	44	10.20%	5.785%
	9:00	47	43	8.51%	
	10:00	45	39	13.33%	
	11:00	45	37	17.78%	
	12:00	89	87	2.25%	
	13:00	85	85	0.00%	
	14:00	85	84	1.18%	
	15:00	84	83	1.19%	
	16:00	81	78	3.70%	
	17:00	76	77	1.32%	
	18:00	72	75	4.17%	
12	8:00	42	45	7.14%	5.636%
	9:00	40	43	7.50%	
	10:00	38	42	10.53%	
	11:00	37	39	5.41%	

	12:00	83	86	3.61%	
	13:00	80	85	6.25%	
	14:00	79	82	3.80%	
	15:00	77	80	3.90%	
	16:00	75	77	2.67%	
	17:00	72	76	5.56%	
	18:00	71	75	5.63%	
13	8:00	55	49	10.91%	5.713%
	9:00	54	48	11.11%	
	10:00	51	45	11.76%	
	11:00	48	44	8.33%	
	12:00	82	85	3.66%	
	13:00	82	82	0.00%	
	14:00	80	78	2.50%	
	15:00	77	76	1.30%	
	16:00	77	75	2.60%	
	17:00	76	72	5.26%	
	18:00	74	70	5.41%	
14	8:00	47	45	4.26%	3.412%
	9:00	45	44	2.22%	
	10:00	45	43	4.44%	
	11:00	43	40	6.98%	
	12:00	85	84	1.18%	
	13:00	84	82	2.38%	
	14:00	80	81	1.25%	
	15:00	79	80	1.27%	
	16:00	75	79	5.33%	
	17:00	74	77	4.05%	
	18:00	72	75	4.17%	

Table 6.19 Soil Moisture Sensor Comparison for Trial 2

Day	Time	Actual Temperature (°C)	DHT22 Temperature Sensor Output (°C)	% Error	Daily Average % Error
1	8:00	26	24.4	6.15%	5.527%
	9:00	26	23.8	8.46%	

	10:00	26	24	7.69%	
	11:00	26	23.9	8.08%	
	12:00	25	24.1	3.60%	
	13:00	26	24.2	6.92%	
	14:00	25	24	4.00%	
	15:00	26	24.5	5.77%	
	16:00	26	24.2	6.92%	
	17:00	25	24.7	1.20%	
	18:00	25	24.5	2.00%	
2	8:00	26	24.1	7.31%	3.279%
	9:00	25.5	24.3	4.71%	
	10:00	25	24.4	2.40%	
	11:00	25	24.8	0.80%	
	12:00	26	25	3.85%	
	13:00	25.5	24.7	3.14%	
	14:00	25	25.2	0.80%	
	15:00	25	24.6	1.60%	
	16:00	25	24.2	3.20%	
	17:00	25	24.5	2.00%	
	18:00	25.5	23.9	6.27%	
3	8:00	27	24	11.11%	4.518%
	9:00	28	26.3	6.07%	
	10:00	27	26.6	1.48%	
	11:00	25.5	24.8	2.75%	
	12:00	26.5	25.1	5.28%	
	13:00	26	25.3	2.69%	
	14:00	26.5	25.5	3.77%	
	15:00	27.5	25.7	6.55%	
	16:00	27	25.4	5.93%	
	17:00	25	25.1	0.40%	
	18:00	24.5	23.6	3.67%	
4	8:00	28	26.7	4.64%	3.887%
	9:00	28	26.1	6.79%	
	10:00	26	26.5	1.92%	

	11:00	26	27.1	4.23%	
	12:00	27.5	28.2	2.55%	
	13:00	26.5	28.3	6.79%	
	14:00	27	27.4	1.48%	
	15:00	27.5	27.5	0.00%	
	16:00	27	28.2	4.44%	
	17:00	26.5	27.7	4.53%	
	18:00	26	27.4	5.38%	
5	8:00	27	26.9	0.37%	3.550%
	9:00	27	26.5	1.85%	
	10:00	26	27.4	5.38%	
	11:00	26	27	3.85%	
	12:00	26.5	27.3	3.02%	
	13:00	26	26.6	2.31%	
	14:00	27.5	26.6	3.27%	
	15:00	26	24.8	4.62%	
	16:00	26	24.6	5.38%	
	17:00	27	25.4	5.93%	
	18:00	26	25.2	3.08%	
6	8:00	28	24.2	13.57%	4.632%
	9:00	28	25.9	7.50%	
	10:00	27	26.4	2.22%	
	11:00	28	26.8	4.29%	
	12:00	27	25.1	7.04%	
	13:00	27.5	26.9	2.18%	
	14:00	27	26.2	2.96%	
	15:00	27.5	26.8	2.55%	
	16:00	26.5	26	1.89%	
	17:00	27	25.6	5.19%	
	18:00	25.5	25.9	1.57%	
7	8:00	27	25	7.41%	4.812%
	9:00	26	24.3	6.54%	
	10:00	27.5	24.9	9.45%	
	11:00	28	25.3	9.64%	

	12:00	26	25.7	1.15%	
	13:00	26.5	25.9	2.26%	
	14:00	27	26.4	2.22%	
	15:00	27.5	26.1	5.09%	
	16:00	26	26.8	3.08%	
	17:00	26.5	25.1	5.28%	
	18:00	25	24.8	0.80%	
8	8:00	27	25.1	7.04%	3.717%
	9:00	26.5	25.8	2.64%	
	10:00	26	25.6	1.54%	
	11:00	27	25.7	4.81%	
	12:00	25	25.8	3.20%	
	13:00	25.5	24	5.88%	
	14:00	25	24.1	3.60%	
	15:00	26	24.5	5.77%	
	16:00	25	24.2	3.20%	
	17:00	25	24.2	3.20%	
	18:00	24.5	24.5	0.00%	
9	8:00	26	24.7	5.00%	3.684%
	9:00	26.5	24.9	6.04%	
	10:00	25	24.7	1.20%	
	11:00	24	24.4	1.67%	
	12:00	26.5	24.2	8.68%	
	13:00	25	24.1	3.60%	
	14:00	25.5	24.3	4.71%	
	15:00	24	23.9	0.42%	
	16:00	24	23.9	0.42%	
	17:00	25	24	4.00%	
	18:00	25	23.8	4.80%	
10	8:00	25	24.2	3.20%	4.417%
	9:00	24	23.8	0.83%	
	10:00	24.5	23.3	4.90%	
	11:00	25	23.8	4.80%	
	12:00	25.5	23.6	7.45%	

	13:00	25	23.7	5.20%	
	14:00	26	23.8	8.46%	
	15:00	25.5	23.8	6.67%	
	16:00	24	23.6	1.67%	
	17:00	24.5	23.8	2.86%	
	18:00	23.5	24.1	2.55%	
11	8:00	25	24.2	3.20%	4.066%
	9:00	26	24	7.69%	
	10:00	24	24.1	0.42%	
	11:00	24.5	24.5	0.00%	
	12:00	25	24.4	2.40%	
	13:00	26.5	24.8	6.42%	
	14:00	25	23.9	4.40%	
	15:00	24	23.2	3.33%	
	16:00	25	23.3	6.80%	
	17:00	26	23.6	9.23%	
	18:00	24	24.2	0.83%	
12	8:00	24	24.6	2.50%	2.497%
	9:00	25	24.7	1.20%	
	10:00	26	25.4	2.31%	
	11:00	26.5	26.1	1.51%	
	12:00	25	24.1	3.60%	
	13:00	25.5	23.6	7.45%	
	14:00	25	23.8	4.80%	
	15:00	24	24.1	0.42%	
	16:00	24.5	24.2	1.22%	
	17:00	24	24.1	0.42%	
	18:00	24.5	24	2.04%	
13	8:00	24	24.1	0.42%	3.666%
	9:00	25	23.9	4.40%	
	10:00	25.5	24.8	2.75%	
	11:00	25	25.1	0.40%	
	12:00	24	24.5	2.08%	
	13:00	26	25.8	0.77%	

	14:00	24.5	26	6.12%	
	15:00	25	26.4	5.60%	
	16:00	24	25.7	7.08%	
	17:00	24.5	25.1	2.45%	
	18:00	23	24.9	8.26%	
14	8:00	25	25.2	0.80%	4.153%
	9:00	27	25	7.41%	
	10:00	26.5	24.8	6.42%	
	11:00	26	24.1	7.31%	
	12:00	25	24.6	1.60%	
	13:00	24.5	23.9	2.45%	
	14:00	25	24.1	3.60%	
	15:00	26.5	24.4	7.92%	
	16:00	24	24	0.00%	
	17:00	25	23.7	5.20%	
	18:00	23.5	24.2	2.98%	

Table 6.20 Temperature Sensor Comparison for Trial 2

Day	Time	Actual Humidity (%)	DHT22 Humidity Output (%)	% Error	Daily Average % Error
1	8:00	67	61	8.96%	8.456%
	9:00	66	60	9.09%	
	10:00	67	62	7.46%	
	11:00	67.5	63	6.67%	
	12:00	68	61	10.29%	
	13:00	67	60	10.45%	
	14:00	68	60	11.76%	
	15:00	68	61	10.29%	
	16:00	65	60	7.69%	
	17:00	65.5	62	5.34%	
	18:00	60	63	5.00%	
2	8:00	66	63	4.55%	3.247%
	9:00	65	63	3.08%	
	10:00	68.5	64	6.57%	
	11:00	66	65	1.52%	

	12:00	69	66	4.35%	
	13:00	68.5	65	5.11%	
	14:00	66	65	1.52%	
	15:00	64	64	0.00%	
	16:00	65.5	64	2.29%	
	17:00	67.5	64	5.19%	
	18:00	64	65	1.56%	
3	8:00	61	64	4.92%	7.893%
	9:00	60	63	5.00%	
	10:00	63.5	59	7.09%	
	11:00	65	61	6.15%	
	12:00	68	62	8.82%	
	13:00	66	59	10.61%	
	14:00	67	62	7.46%	
	15:00	67.5	61	9.63%	
	16:00	65	60	7.69%	
	17:00	64.5	58	10.08%	
	18:00	64	58	9.38%	
4	8:00	51	58	13.73%	6.102%
	9:00	54	59	9.26%	
	10:00	54.5	59	8.26%	
	11:00	55	60	9.09%	
	12:00	59	59	0.00%	
	13:00	57.5	59	2.61%	
	14:00	55	58	5.45%	
	15:00	56	58	3.57%	
	16:00	56.5	60	6.19%	
	17:00	57	59	3.51%	
	18:00	55	58	5.45%	
5	8:00	53	60	13.21%	5.927%
	9:00	55	60	9.09%	
	10:00	55.5	61	9.91%	
	11:00	56	60	7.14%	
	12:00	57	60	5.26%	

	13:00	58	60	3.45%	
	14:00	57	56	1.75%	
	15:00	57.5	56	2.61%	
	16:00	58	57	1.72%	
	17:00	55	57	3.64%	
	18:00	54	58	7.41%	
6	8:00	52	58	11.54%	6.988%
	9:00	53	58	9.43%	
	10:00	56.5	59	4.42%	
	11:00	56	59	5.36%	
	12:00	54	60	11.11%	
	13:00	55	59	7.27%	
	14:00	55.5	59	6.31%	
	15:00	53	57	7.55%	
	16:00	57.5	58	0.87%	
	17:00	56	58	3.57%	
	18:00	53	58	9.43%	
7	8:00	52	60	15.38%	5.592%
	9:00	55	59	7.27%	
	10:00	56	59	5.36%	
	11:00	56.5	57	0.88%	
	12:00	56	58	3.57%	
	13:00	54	58	7.41%	
	14:00	55.5	56	0.90%	
	15:00	56	55	1.79%	
	16:00	57	56	1.75%	
	17:00	57.5	61	6.09%	
	18:00	54	60	11.11%	
8	8:00	54	60	11.11%	3.587%
	9:00	56	60	7.14%	
	10:00	57	59	3.51%	
	11:00	57.5	59	2.61%	
	12:00	57	57	0.00%	
	13:00	58	55	5.17%	

	14:00	55	55	0.00%	
	15:00	55.5	54	2.70%	
	16:00	56	54	3.57%	
	17:00	55	55	0.00%	
	18:00	55	53	3.64%	
9	8:00	54	52	3.70%	1.976%
	9:00	54.5	53	2.75%	
	10:00	54	53	1.85%	
	11:00	55	54	1.82%	
	12:00	57	54	5.26%	
	13:00	55	54	1.82%	
	14:00	54	53	1.85%	
	15:00	54.5	55	0.92%	
	16:00	56	56	0.00%	
	17:00	57	56	1.75%	
	18:00	58	58	0.00%	
10	8:00	57	59	3.51%	3.196%
	9:00	57	60	5.26%	
	10:00	58.5	62	5.98%	
	11:00	60	61	1.67%	
	12:00	57	60	5.26%	
	13:00	58.5	61	4.27%	
	14:00	59	60	1.69%	
	15:00	60	59	1.67%	
	16:00	60.5	59	2.48%	
	17:00	60	59	1.67%	
	18:00	59	60	1.69%	
11	8:00	57	60	5.26%	2.889%
	9:00	61	60	1.64%	
	10:00	60.5	60	0.83%	
	11:00	62	58	6.45%	
	12:00	58	59	1.72%	
	13:00	60	58	3.33%	
	14:00	60.5	59	2.48%	

	15:00	61	58	4.92%	
	16:00	58	59	1.72%	
	17:00	57	58	1.75%	
	18:00	60	59	1.67%	
12	8:00	59	60	1.69%	2.181%
	9:00	59.5	61	2.52%	
	10:00	59	60	1.69%	
	11:00	62	60	3.23%	
	12:00	58	59	1.72%	
	13:00	61.5	59	4.07%	
	14:00	60	61	1.67%	
	15:00	61	59	3.28%	
	16:00	60.5	59	2.48%	
	17:00	60	60	0.00%	
	18:00	61	60	1.64%	
13	8:00	59	59	0.00%	3.614%
	9:00	60	59	1.67%	
	10:00	61	59	3.28%	
	11:00	60.5	57	5.79%	
	12:00	59	57	3.39%	
	13:00	58.5	57	2.56%	
	14:00	59	57	3.39%	
	15:00	59	58	1.69%	
	16:00	60	57	5.00%	
	17:00	61	58	4.92%	
	18:00	62	57	8.06%	
14	8:00	59	57	3.39%	2.899%
	9:00	60	58	3.33%	
	10:00	60.5	59	2.48%	
	11:00	61	59	3.28%	
	12:00	59	59	0.00%	
	13:00	58.5	60	2.56%	
	14:00	58	62	6.90%	
	15:00	59	62	5.08%	

	16:00	62.5	61	2.40%	
	17:00	60.5	61	0.83%	
	18:00	61	62	1.64%	

Table 6.21 Humidity Sensor Comparison for Trial 2

Appendix L PRO1 PANEL COMMENTS AND REVISIONS

PRO1 Panel Comments and Revisions

List Comments and Revisions here

1. Consider humidity
2.
3.

1. Revise general objective to become: To develop an automated system that performs basic gardening operations such as seeding, watering, & monitoring.
2. Include in the methodology how to assess the attainment of the objectives.
3. Revise the specific objectives to become
   1) To develop a system capable of providing uniform and variable seed distribution within the field.
   2) To develop a mobile application that will monitor the soil moisture, temperature, & the humidity. plant growth
   3) To develop an adaptive automatic watering system for the crop.
   4) To conduct cost analysis on the proposed system.

Noted by:

Dr. Gerino P. Mappatao
Chair of Panel

Date: October 25, 2018

Note: Keep a copy of this Appendix. It is a requirement that has to be submitted in order to qualify for PRO3 Defense.

Appendix M PRO3 PANEL COMMENTS AND REVISIONS

Panelist	Comment	Summary of how the comment was addressed	Location
Engr. Antonette C. Roque	1. Record all information in the continuous monitoring for 2 weeks and compare the growth to existing literatures. 2. Compare hourly retrieved data output to commercially available ones 3. Record weight of the 2-week plant and predict its output weight after a month and compare it with existing literature to have a basis for the cost analysis. 4. Provide an option for user to set the threshold values. 5. Plot the data of each parameter in a graphical form. 6. Add/remove some information in design	-Documented the processes and results for the two-week trial -Compared used sensors to commercially available ones -Weighed pechay at two weeks and performed analysis on the results -Added sensor threshold feature and graphs in the mobile application	-Section4.1 Overview -Section4.2.1 Chamber Assembly -Section4.2.2 Electronics System -Section4.2.2.3.2 Libraries -Section4.3.1 Cartesian Robot Assembly -Section4.4.2.2 Database Page -Section 4.4.2.3 Sensors Page -Section4.4.2.4 Development of Mobile Application -Section5.2.2.2 Trial 2

	consideration and methodology 7. Add explanations to tables and move selected ones to appendix 8. Explain in conclusion how each objective was achieved 9. Remove estimated work schedule and budget 10. Discuss the relationship of the different parts of your system 11. Discuss the different packages and libraries and its purpose in the design 12. Present the flowchart instead of the codes		-Section5.3 Cost Analysis -Section 6.2.2.2 Trial 2 -Section 7.1 Concluding Remarks -SectionAppendix K MONITORING SYSTEM DATA
Engr. Armie E. Pakzad	1. Corrections in objectives 2. Use appropriate tenses in the highlighted paragraphs 3. Insert a table of related works for quick reference 4. Include in the recommendations the inclusion of crops to be planted and the ideal temperature and humidity for the crops,	-Edited the requested grammatical revisions -Included a table in the literature for reference and additional feature in the recommendations	-Section 1.4.2 Specific Objectives -Section 2.1 Existing Work -Section4.4.2.2 Database Page -Section 4.4.2.3 Sensors Page

	to the user interface as the input parameters 5. Include the monitoring table of the temperature, humidity, and plant growth 6. Record the weight of the produce and compare it to the values mentioned in the existing literature 7. Make a table of revision which includes the following : Panelist, comment, summary of how the comment was addressed, and clickable location		-Section5.2.2.2 Trial 2 -Section5.3 Cost Analysis -Section 6.2.2.2 Trial 2 -Section 7.3 Recommendations
Engr. Ana Antoniette C. Illahi	1. Please plant Pechay for 2 weeks, make a journal for monitoring the whole process. 2. Don't forget to gather the data for 14 days. ("Continuous Monitoring") 3. Process the data as a form of testing to verify the functionality of all the sensors' "Accuracy Rate". 4. Please weigh in your produce and compare it to the other reliable source "literature" if it's in ideal size, and weight of the Pechay. 5. Testing of all sensors must be visible in the documentation.	-Planted pechay and gathered data for two weeks -Observed the retrieved data from sensors and performed a comparison between the ones used and commercially bought to see if its reliability -Plotted the data in a graphical form that can be seen through the application	-Section4.4.2.2 Database Page -Section 4.4.2.3 Sensors Page -Section5.2.2.2 Trial 2 -Section5.3 Cost Analysis -Section 6.2.2.2 Trial 2

	6. Graph all the data you gather from the system. 7. All revisions must be visible in the documentation.		