K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

University of Mumbai

Automated Plant Monitoring and Watering System

Submitted at the end of semester IV in partial fulfillment of requirements

Bachelors of Technology in Computer Engineering

sprout

by

Aditya Pawar
Roll No: 16010120184

Aryaman Gandhi
Roll No: 16010120185

Harshit Wadhavkar
Roll No: 16010120186

Guide

Prof. Uday Joshi

Department of Computer Engineering

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch 2020-2024

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the MINIPROJECT report entitled **Supervised Learning Techniques for Text Classification** submitted by Aditya Pawar, Aryaman Gandhi, Harshit Wadhavkar, at the end of semester IV of SY B. Tech are bona fide record for partial fulfillment of requirements for the degree of Bachelors in Computer Engineering of University of Mumbai.

Guide

Head of the Department

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this MiniProject report entitled **Automated Plant Monitoring and Watering System** is a bona fide record of Mini project work done by Aditya Pawar, Aryaman Gandhi and Harshit Wadhavkar during semester IV.

This Mini project work is submitted at the end of semester IV in partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

____

Internal Examiner 1

____

Internal Examiner 2

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student **Roll No.**	**Signature of the Student** **Roll No.**
Signature of the Student **Roll No.**	**Signature of the Student** **Roll No.**

Date:

Place: Mumbai-77

Chapter 1: Introduction

1.1 Background

Home gardens grant access to cheaper, fresher and more sustainable produce. Owning a home garden is one of the simplest things that one can do to move towards self-sufficiency and reducing the load on the environment. Besides this, one can control what fertilizers and pesticides go into the plant. Apart from this, it adds to the aesthetic value and provides a positive environment at home. However, in today's world, people live busy lives. They have many different tasks to focus on, like their jobs, education, and side projects. Most people do not have the time or resources available to make or maintain a home garden.

Hydroponics is the study of growing plants in a liquid nutrient medium, without the need for soil. The roots of the plant are immersed in the nutrient solution, and the plants receive their nourishment from there.

Hydroponics has become more popular over the years, with more and more people adapting hydroponics into their home gardens. It provides an additional convenience over traditional soil setups, which is that soil is messy and hard to clean or work with. Apart from these, there is basically no risk of overwatering or underwatering. The root growth can be monitored easily, and it is effortless to change the container that the plant is growing in. As such, these hydroponic setups have many advantages over conventional soil setups, but they also need to be monitored more closely. Routine measurements of solution concentration and pH are necessary, which may not be possible for someone with a busy life living in an urban area.

1.2 Problem Statement

Lack of time and energy to maintain and monitor and maintain a home hydroponics setup.

1.3 Objective

To design and build a system that automates the process of monitoring the plant's health and watering/fertilizing it. The relevant information about the nutrient levels, pH, humidity and temperature should also be available to the user.

1.4 Scope

- This project specifically targets hydroponic farming. The reason for this is that hydroponic farms are more cost effective and easier to control with the help of pumps.

- This product only targets the people living in urban or suburban areas, who want to have an indoor farm without the hassle of having to maintain it.

- Although the target audience is limited, even actual hydroponic farmers working on larger scales may make use of this system by scaling it up.

- This project is intended to be a prototype and a proof of concept.

- A successful prototype would be one that would be able to fully regulate the nutrient concentration by making use of pumps. It should also be able to report the current nutrient concentration in the medium, along with the current pH and temperature through a webpage.

- While this project is intended to be as energy efficient as possible, there will be no special provisions for powering it using renewable energy. It will be powered directly through a standard 220V power outlet.

- The project will make use of a Raspberry Pi board as its main component. This will be programmed using Python. It will make use of several sensors and pumps connected to the Raspberry Pi to read the data about the plant, and thus maintain a fixed environment for the plant to grow in.

Chapter 2: Literature Survey

2.1 Literature Survey

Before starting any work, it is necessary to gain an understanding of the topic at hand. Hydroponics is a relatively new technique of horticulture. For this project, Kratky Hydroponics was chosen. The Kratky Method is detailed in [1], which is a research paper written by B. A. Kratky. In the Kratky Method, the plant's roots are immersed in a nutrient solution.

We also used the internet to search for similar projects that had been done before. [2] describes a similar project that was built using the ESP8266 board and an ATMega microcontroller. It made use of a pH sensor and a DHT11 temperature sensor, along with an ultrasonic sensor to detect water level. It makes use of a Mobile app to deliver the data to the user.

[3] describes another project, which was made using 3 Arduino boards and a Raspberry Pi to act as a web server. Two Arduinos act as "Nodes", which serve to get data from the sensors and send the data to a third Arduino using a radio signal. The third Arduino acts as a gateway, and sends this data to the Raspberry Pi which acts as a web server, running a monitoring and control program. The system delivers data through an Android app. Upon comparison, the plants were shown to grow better indoors while using this system, as compared to the same plant growing outdoors.

[4] explores a similar system, making use of a NodeMCU ESP8266. This board publishes sensor data to a Raspberry Pi 3 Model B using the MQTT protocol, which is a protocol used

for IoT devices to communicate with each other. The Raspberry Pi communicates with a Node Red server, which is used for automation purposes and also serves to deliver the sensor data to the user.

After learning what hydroponics is, and exploring the Kratky Method of hydroponics, looking at the different approaches that may be used to create a system to manage a hydroponic system, we may proceed onto the next step.

Chapter 3: Project Design

3.1 Proposed Plan

After reviewing the different approaches in the literature survey stage, we decided to use a Raspberry Pi as the main component of the system. The system would feature 4 sensors, a TDS sensor, a pH sensor, a temperature and humidity sensor, and a water level sensor.

Apart from this, the Raspberry Pi would also be connected to pumps, which would be able to pump water or nutrients into the solution, or pump the solution out if needed.

The Raspberry Pi would monitor the sensor data, and upon noticing a deviation from the expected nutrient concentration value or the expected pH value, it would trigger the pumps and correct it.

The Raspberry Pi should also be able to provide the sensor data onto a webpage, from where the sensor values can be read by the user.

The project would use Python and Flask for the backend, while HTML, CSS and JS would be appropriate for the frontend.

3.2 Choice of Components

Once a rough plan is in place, components can be chosen that fit this plan.

3.2.1 Raspberry Pi

First a Raspberry Pi model needs to be picked. It is a microprocessor which is sold along with a PCB that includes power inputs, USB and HDMI ports, and GPIO pins, along with other useful components. It is completely programmable.

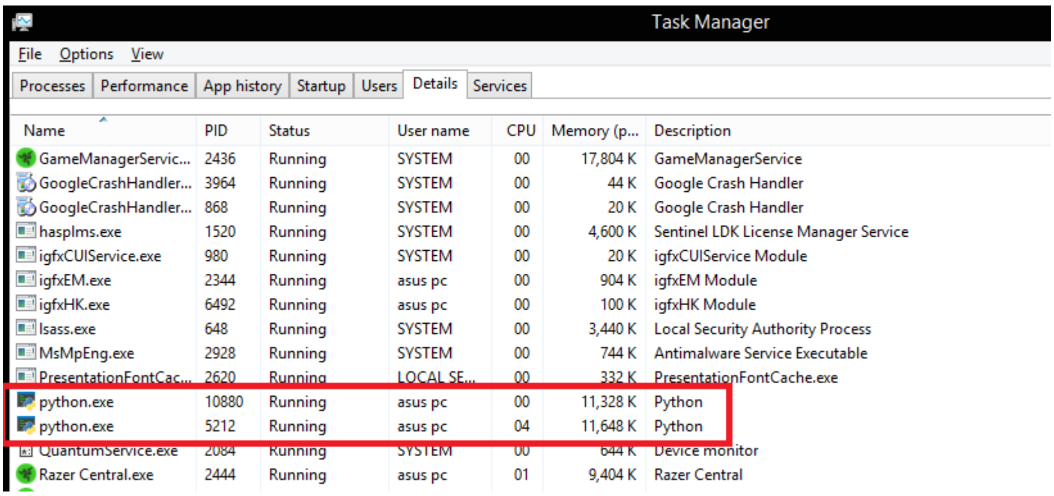
The Raspberry Pi is sold in many different models, all with different features.

The two major models considered for the project are: **Raspberry Pi Zero W** and the **Raspberry Pi 3 Model B**. Only boards that feature Wi-Fi connectivity have been chosen to facilitate portability and easy testing of the prototype.

The two models were compared against each other.

Raspberry Pi Zero W	Raspberry Pi 3 Model B
Cheaper in cost.	More expensive.
512MB of RAM.	1GB of RAM
Smaller in size.	Larger in size.
Very light	5 times as heavy

When considering all these factors, the deciding points came down to the price and the RAM. When tested and benchmarked, a python flask app only used about ~23MB of RAM.



Task Manager

File Options View

Processes Performance App history Startup Users Details Services

Name	PID	Status	User name	CPU	Memory (p...	Description
GameManagerServic...	2436	Running	SYSTEM	00	17,804 K	GameManagerService
GoogleCrashHandler...	3964	Running	SYSTEM	00	44 K	Google Crash Handler
GoogleCrashHandler...	868	Running	SYSTEM	00	20 K	Google Crash Handler
hasplms.exe	1520	Running	SYSTEM	00	4,600 K	Sentinel LDK License Manager Service
igfxCUIService.exe	980	Running	SYSTEM	00	20 K	igfxCUIService Module
igfxEM.exe	2344	Running	asus pc	00	904 K	igfxEM Module
igfxHK.exe	6492	Running	asus pc	00	100 K	igfxHK Module
lsass.exe	648	Running	SYSTEM	00	3,440 K	Local Security Authority Process
MsMpEng.exe	2928	Running	SYSTEM	00	744 K	Antimalware Service Executable
PresentationFontCac...	2620	Running	LOCAL SE...	00	332 K	PresentationFontCache.exe
python.exe	10880	Running	asus pc	00	11,328 K	Python
python.exe	5212	Running	asus pc	04	11,648 K	Python
QuantumService.exe	2084	Running	SYSTEM	00	644 K	Device monitor
Razer Central.exe	2444	Running	asus pc	01	9,404 K	Razer Central

Peak usage was found to be about ~28MB of RAM.

The finished product is supposed to contain 2 such flask apps. Even allowing for a 300% margin of error, the final app should use only around ~170MB of RAM.

This would make the Zero W the ideal board for developing this project. However, it should be considered that external dependencies and high quality images may easily inflate the RAM usage of the apps. The Raspberry Pi will also be running its own OS, which will likely use a sizable amount of available memory.

Keeping all these factors in mind, we decided to use the Raspberry Pi 3 Model B for the purpose of this project. Since the project is in a prototyping phase, there may not be enough room for optimization of the code. The ability to write code without having to worry about the small memory constraint is more valuable at this stage than a more economical product.

However, a finished and polished version of the code could possibly run on a Zero W, and if this project ever were to be mass-manufactured (or sold as a kit), the Zero or Zero W would be the board of choice.

3.2.2 Sensors

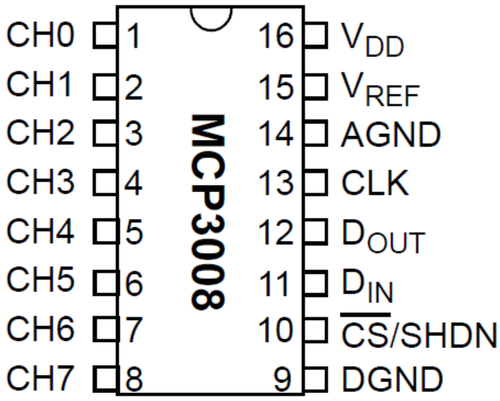
Many different sensors were considered for this project. However, the sensors picked for the project are listed below.

- Temperature and Humidity Sensor: DHT22
- pH Sensor: KPE-03 probe + pH-4502C module
- TDS Sensor: DFRobotics TDS Meter v1.0
- Water Level Sensor: Funduino Water Level Sensor

3.2.2 Analog to Digital Converter

The pH sensor, the TDS sensor, and the Water Level Sensor are all analog sensors. The Raspberry Pi, however, can only read digital signals. As such, an Analog to Digital Converter (ADC) is needed to convert the analog signal to digital. The ADC chosen for this project is the MCP3008.

The MCP3008 is available in a 16-pin DIP package. It has 8 channels (meaning it can take 8 separate analog signal inputs), which are labeled CH0 to CH7. Here is the pin layout for MCP3008.

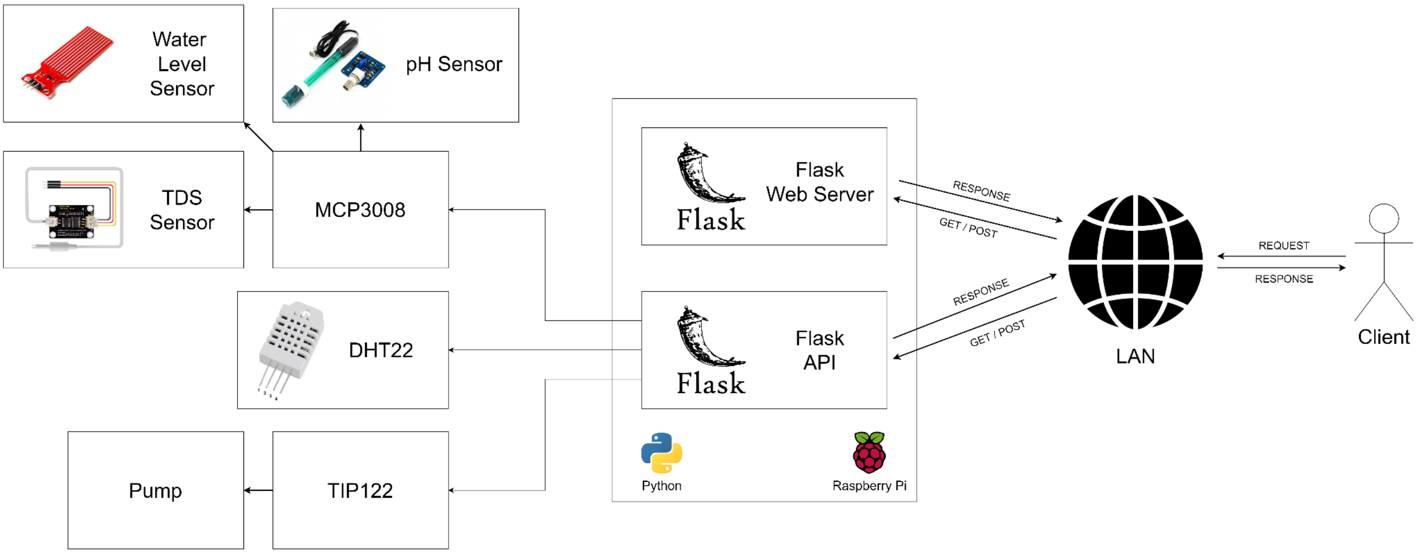


3.2.3 TIP122

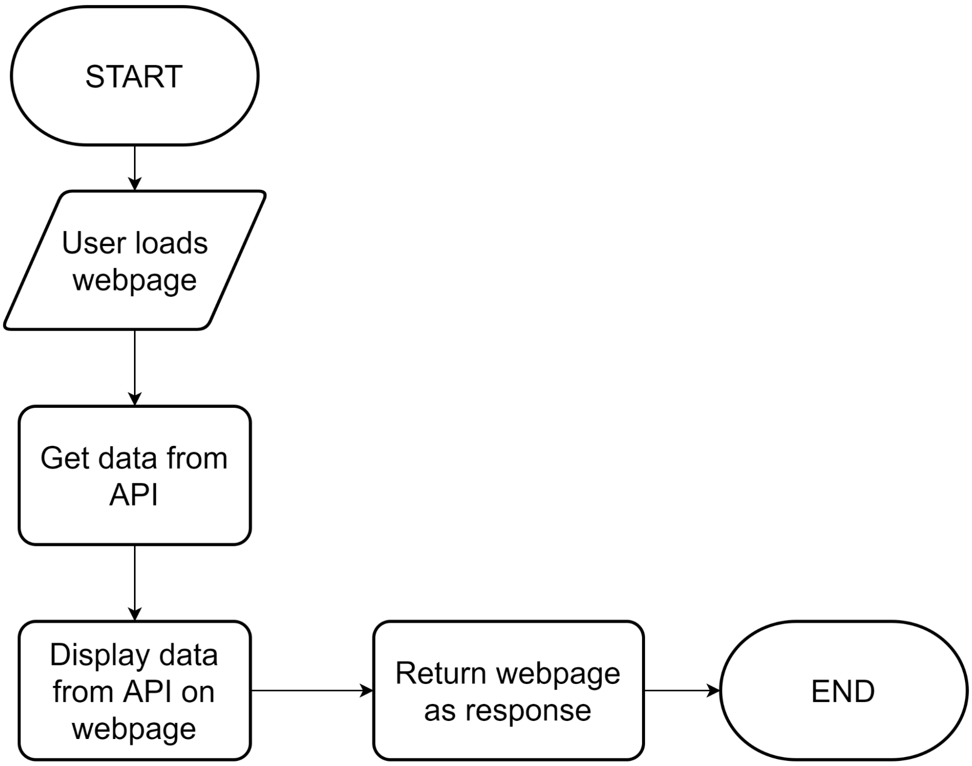
The pumps used for this project are generic DC pumps that operate between 3-6V. These pumps will be powered directly from the Raspberry Pi's 5V output. To turn these pumps on and off, many options are available. One could use a relay module, or a simple transistor. However, upon testing, a transistor caused the voltage to drop way below 5V. A relay was not even considered, as it was way too complex and expensive. A middle ground between these two was the TIP122, which is a **darlington transistor pair.** The TIP122 allows one to switch the pump on and off using a signal from the Raspberry Pi GPIO pins. The voltage drop was minimal too. Since these factors made it ideal for this project, the TIP122 was used.

3.3 System Architecture Diagram

A system architecture diagram was made using draw.io.



3.4 Module Wise Flow Diagram



3.5 Programming Languages & Standards

3.5.1 Python

The programming language to be used for this project is primarily **Python.** Python will be used to read the data from sensors, actuate the motors, and will also serve as the backend for both the API and the website.

The reasons for using Python are listed below.

- Readability
- Ease of writing code
- Availability of a large number of libraries and examples
- Supported on Raspberry Pi by default

The standards set for the Python code are as follows.

- Code should be readable and well documented.
- OOP programming paradigms should be adhered to as far as possible.
- Python's PEP8 style guide should be followed.
- All python functions and classes should be grouped in separate py files, as is appropriate.
- In python, one has the option to use either tabs or spaces to indent. Spaces should be used to indent code in all files.
- Proper comments should be used everywhere, and docstrings must be included for all functions and classes.

3.5.1 HTML / CSS / JS

HTML, CSS and JS will be used to program the frontend of the website. They were chosen over something more complicated like React.js or Angular for the main reason that the level of complexity provided by those frameworks is not required for this project.

The standards for the frontend code are as follows.

- HTML tags used should be semantic.
- HTML should be properly formatted and indented.
- The use of code in style and script tags to be avoided. All the CSS and JS must be external.
- The website should be accessible by people with disabilities.
- The HTML and CSS should be validated according to W3C's standards.
- CSS classes should be appropriately named.
- The **!important** keyword in CSS is to **NOT** be used at all.

3.5 Responsibility Matrix

A responsibility matrix was created to display the responsibilities for each member of the group.

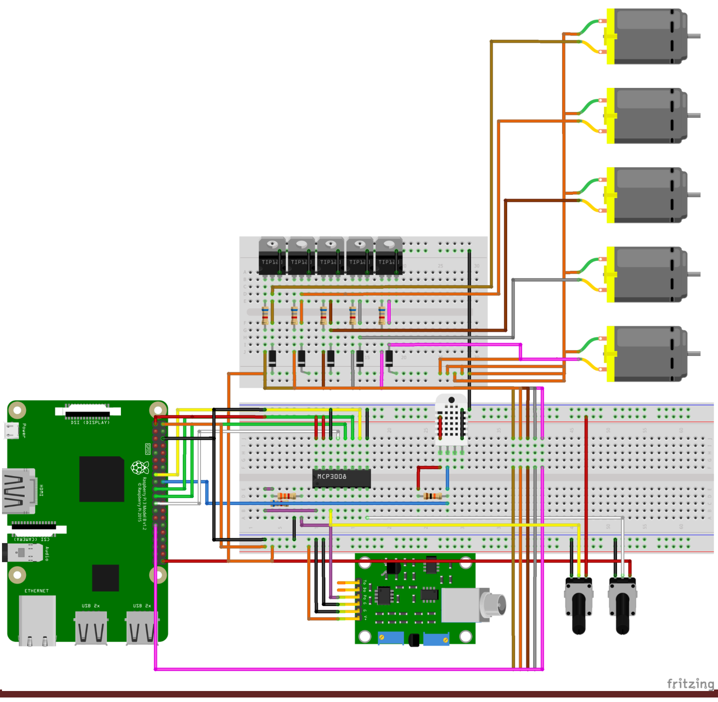
Task	Aditya Pawar	Aryaman Gandhi	Harshit Wadhavkar
UI / Website frontend			
Full UI Design	X		
Coding	X		
Circuit			
Circuit design		X	
Circuit assembly		X	
Library for sensors and motor		X	

Backend			
API design			X
Backend coding			X
Plant			
Acquiring supplies			X
Planting		X	
Testing			
Testing approach	X		
Test Cases	X	X	X
UI / Website Tests	X		
Circuit Tests		X	
Backend Tests			X
Presentation			
Powerpoint	X	X	X
Report	X	X	X

Chapter 4: Implementation & Testing

4.1 Breadboard Circuit Diagram

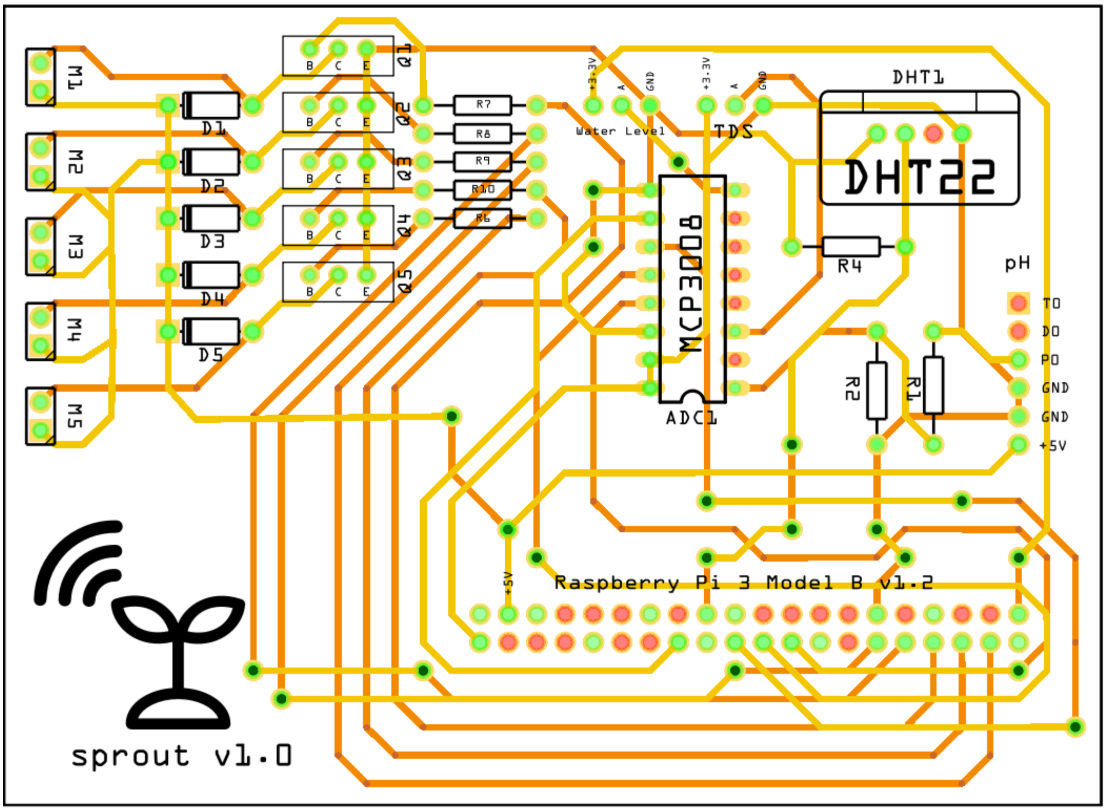
Originally, a standard breadboard was used to test this project. Here is a diagram of the original breadboard design that was created using **Fritzing Software.**



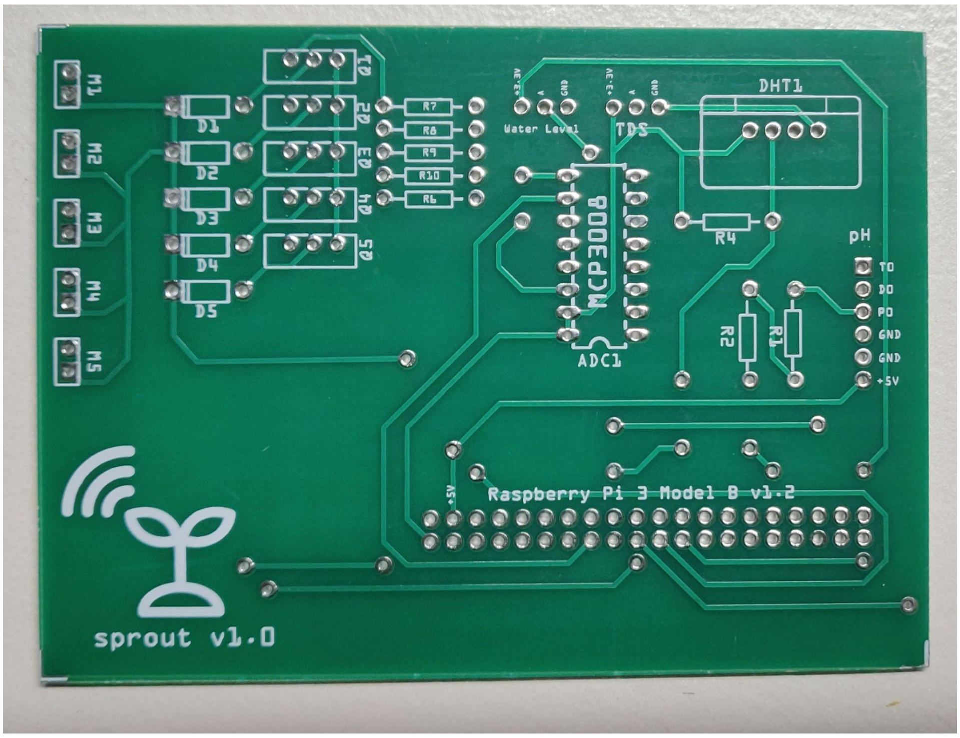
4.2 PCB

The major problem with the breadboard was that it involved using a lot of wires. This made it complex to assemble, and also very difficult to carry. It did not look appealing or presentable. Also, the breadboard connections are somewhat loose, due to which there would be chances of components getting unplugged during operation. This could cause damage to the components.

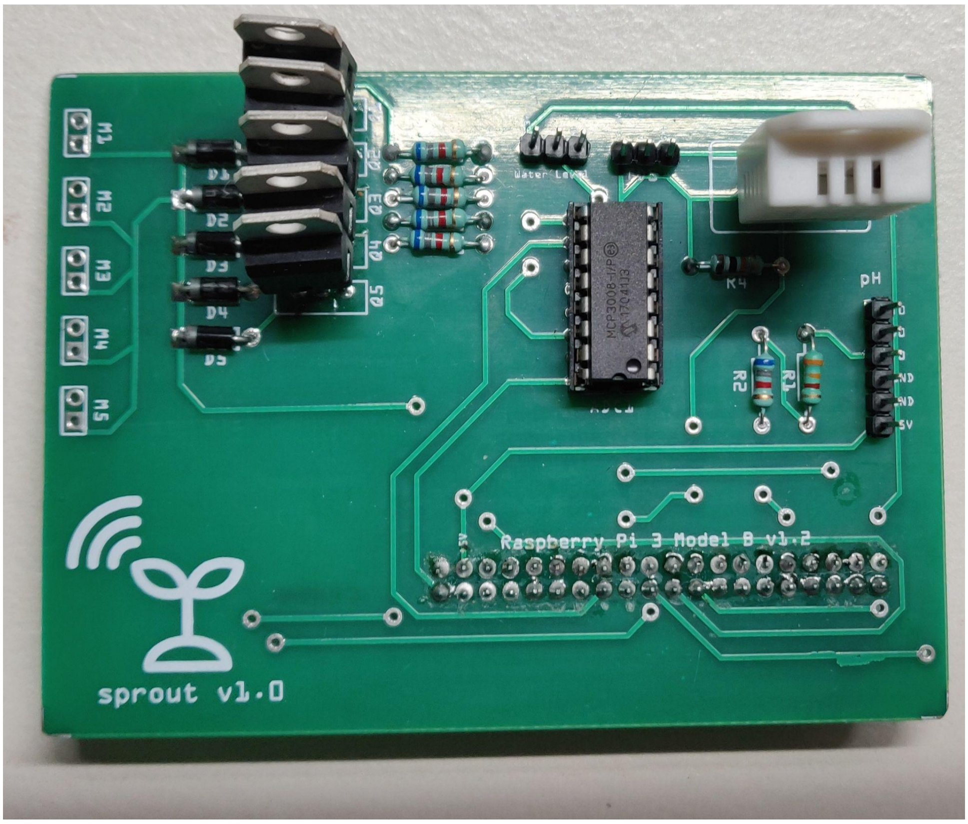
As a result, the decision was made to redo the circuit on a PCB.
The PCB was first designed on Fritzing, then the design files were exported and sent to a PCB manufacturer. The components were then soldered onto the prepared PCB.
Here is a screenshot of the PCB design.



Here is the actual PCB.



Here is the PCB after soldering components.



4.3 Implemented Functions

Currently, the system can perform the following functions.

- Read sensor data and provide it over an API.
- Read sensor data and provide it on the webpage.
- If the nutrient content is low, it can actuate the pumps to deliver nutrients to the plant.
- If the water level is low, it can actuate the pumps to deliver water to the plant.
- It also has one pump designated to drain the plant tank.
- There are a total of 5 pumps, of which two are currently not used. However, they may be incorporated into the system to pump solutions to increase or decrease the pH of the solution.

4.3 Test Cases

Test case	**Description**	**Intended result**	**Actual result**	**Completed by**
pH sensor test	The pH sensor is immersed into different solutions. The pH is noted.	The pH should be below 7 for acidic substances, and above 7 for basic substances.	The pH levels were below 7 for lemon juice, and above 7 for soap.	Aryaman Gandhi
TDS sensor test	The TDS sensor is immersed into different solutions. The TDS reading is noted.	The TDS reading is 0 in plain air, and is close to 0 for distilled water. It increases in salt water.	The result matched the expected result exactly. The TDS sensor is more sensitive than expected.	Aryaman Gandhi

Temperature sensor test	The ambient temperature and humidity level is measured using the DHT22	The temperature and humidity levels should match those from a weather app.	The temperature was found to be almost 1 degree below ambient, and the humidity was slightly higher.	Aryaman Gandhi
Pump controller test	The pump control system is connected to the Pi and the pump is turned on.	The pump is expected to start pumping water.	The water is pumped.	Aryaman Gandhi
Pump voltage test	The potential difference across the pump is measured in the “on” state.	The potential difference is expected to be around 4.8V	The potential difference is higher than 4.8V, and is fluctuating. This is probably due to back emf from the pump, so it can be ignored.	Aryaman Gandhi
Overheating test	The sensors and pumps are connected to the Pi. Readings are taken every second and the pumps are left on for a total of 30 minutes.	The system is expected to not heat up too much.	Most of the components did not rise significantly in temperature. The MCP3008, however, was warm to the touch.	Aryaman Gandhi
API testing	The API is queried for sensor data and	The response should come immediately,	The response is received in 11 milliseconds,	Harshit Wadhavkar

	the response from the API is checked.	containing valid sensor data.	containing valid and plausible readings.	

Chapter 5: Conclusion & Further Work

5.1 Conclusion

The final product obtained was a product that was capable of maintaining a hydroponic garden on its own for a very long time. As long as the reservoirs are full and the system is connected to power, it will continue to function. Overall, the results were satisfactory and aesthetically pleasing. However, a much longer testing period is required to gain insight into the efficiency and effectiveness of the system.

5.2 Further Work

There is still room for improvement in this project. An actual, 3D printed enclosure for both the circuits and a custom 3D printed planter to hold the sensors in place would benefit the system a lot.

The system may be improved by adding data storage capabilities to it. So, the sensors would be able to record data, and the user can gain access to the raw data or plot graphs using it.

The accuracy of the sensors may be improved by further calibration.

Lastly, it would be very interesting to incorporate machine learning or artificial intelligence into this project. A machine learning algorithm could monitor the plants behaviours and make predictions about it. A camera could be connected to the plant, and the machine learning algorithm would be able to detect plant diseases or other deficiencies and notify the user before they have a chance to hurt the growth of the plant.

Chapter 6: Bibliography

[1] Kratky, B. A. (2009). Three non-circulating hydroponic methods for growing lettuce. *Acta Horticulturae*, (843), 65–72. https://doi.org/10.17660/actahortic.2009.843.6

[2] Dr. Asawari Dudwadkar. (2020). Automated hydroponics with remote monitoring and control using IOT. *International Journal of Engineering Research And*, *V9*(06). https://doi.org/10.17577/ijertv9is060677

[3] Palande, V., Zaheer, A., & George, K. (2018). Fully Automated Hydroponic System for Indoor Plant Growth. Procedia Computer Science, 129, 482–488. https://doi.org/10.1016/j.procs.2018.03.028

[4] Lakshmanan, R., Djama, M., Perumal, S., & Abdulla, R. (2020). Automated smart hydroponics system using the internet of things. International Journal of Electrical and Computer Engineering (IJECE), 10(6), 6389. https://doi.org/10.11591/ijece.v10i6.pp6389-6398