Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

Report

Engineering Reports
Open Access

Automatic Plant Watering System

Chandler McGehee

Mechanical, Aerospace, and Biomedical Engineering Dept.,
Tickle College of Engineering – University of Tennessee,
Tennessee, United States

Correspondence
Chandler McGehee,
Email: chagmcge@vols.utk.edu

Funding Information
Ultrasonic Radar Project
Grant/Award Number: XXXXXXXX

Abbreviations

Direct Current – DC

Soil Moisture Sensor - SMS

Automatic Watering System – AWS

Ground Wire – GND

Submitted to TRACE for potential publication

Abstract

The Automatic Plant Watering System was created to simplify plant care and the work needed to maintain plant health by automating irrigation based on soil moisture level. Using an Arduino Uno microcontroller as the nucleus for the system, this project integrates a soil moisture sensor, a relay module, and a miniature water pump to monitor soil conditions and supply water when needed. The system was constructed using available components from an online-purchased kit and Arduino code found online.
The primary objective was to create a cost-effective and easily assembled solution to reduce the need for human intervention in irrigation and prevent under or over-irrigation. By analyzing soil moisture in real-time, the system activates the water pump only when necessary, ensuring optimal water usage. During development, challenges such as sensor calibration, pump control, and circuit design were addressed through trial and error, as well as additional online guides. The results demonstrated that the system could operate reliably, responding accurately to varying moisture conditions. Potential improvements would include a system that would replace the need to be plugged into a computer. Overall, this system shows itself to be a valuable tool.

KEYWORDS

Automated, Arduino-Uno, irrigation, DIY, moisture sensor, water conservation, gardening

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

I. INTRODUCTION

When talking to many plant owners, the same worries are often brought up: "I forgot to water the plants," "I hope I didn't under/overwater the plants," "If I'm gone for an extended time, I will need someone to water them." These are some of the very reasons why some people say that they do not want plants or think that they are too needy. The reality is that people are very forgetful, and much of the time, plants end up paying the price. Beyond the forgetfulness, an additional issue in plant ownership is that many consumers are ignorant of how much water their plant actually needs. Because of this, many people end up over or underwatering their plants and once again, they pay the price.
The Automatic Watering System (AWS) aims to solve this issue by automating irrigation, ensuring that plants receive the optimal amount of water based on soil moisture levels.
The project uses an Arduin-Uno microcontroller as the central control to a system consisting of a soil moisture sensor, a relay module, and a miniature water pump. When the soil moisture falls below a predetermined level, the system activates the pump, supplying water to the plant. All the materials needed to build this system, excluding the Arduino, came in a kit that was approximately $10. This wound up being a very cost-effective fix that could save a lot of time in the future.
This report outlines the development of the system, including its design, assembly, programming, and testing. It also discusses the challenges faced during the development process and the results achieved during the testing. With potential applications aimed mostly at single-potted plants, the system also showed the possibility of being applied to small indoor gardens or precision agriculture systems. Future improvements would look to eliminate the need for the Arduino to be plugged into a computer to work. Initial thinking has a wireless application as the driver behind the Arduino, in which a user could select a plant type, and the system automatically chooses how often to irrigate.

Document Structure:

II. Materials & Methods

II.1. ARDUINO-UNO

The first and most obvious material for this system is the Arduino-Uno and the wires that come with it. In the UT Parts store, the whole kit costs around $40. The Arduino Uno is the brain of the automatic plant watering system. It is an open-source microcontroller board based on the ATmega328P chip. The Arduino Uno acts as the central processing unit, reading sensor inputs and executing programmed instructions to control the system. In this project, the Arduino Uno processes data from the soil

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

moisture sensor, compares it to a predefined threshold, and triggers the relay module to activate the pump when necessary.

Key Features

- 14 digital input/output pins and 6 analog inputs.
- Operates at 5V and is powered via USB or an external adapter.
- Easy-to-use programming environment via the Arduino IDE.

Figure 1 - Arduino Kit

II.2. SOIL MOISTURE SENSOR (SMS)

Purchased in the AWS kit online, this sensor can detect moisture and can have its sensitivity adjusted to prevent over-watering. It consists of two probes that are inserted into the soil, which detect moisture levels based on electrical resistance. According to Circuit Digest, where the instructions to build the system are from, “This resistance is inversely proportional to the soil moisture which means that higher water in the soil means better conductivity and hence a lower resistance. While the lower water in the soil means poor conductivity and will

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

result in higher resistance. The sensor produces an analog voltage output according to the resistance. It uses this data to decide whether the pump should turn on or off.

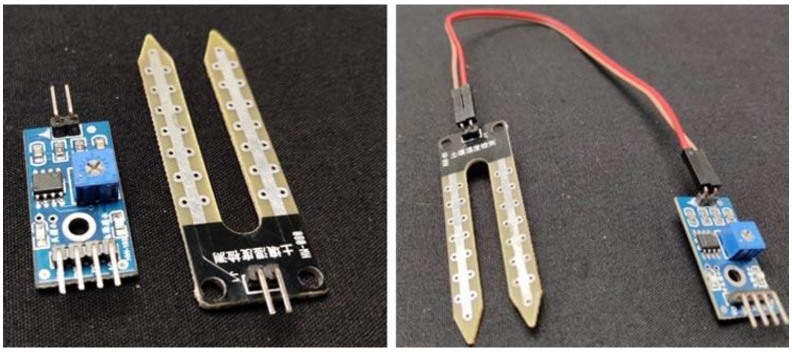
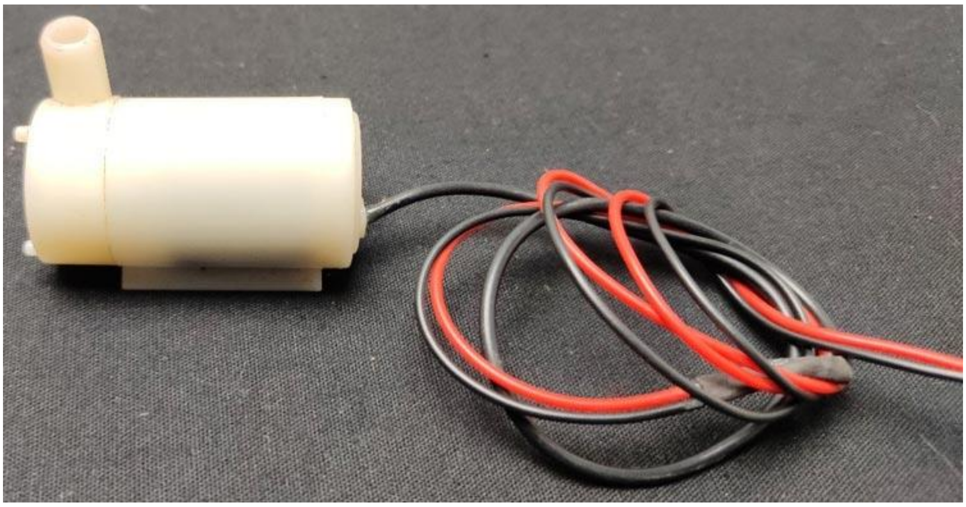


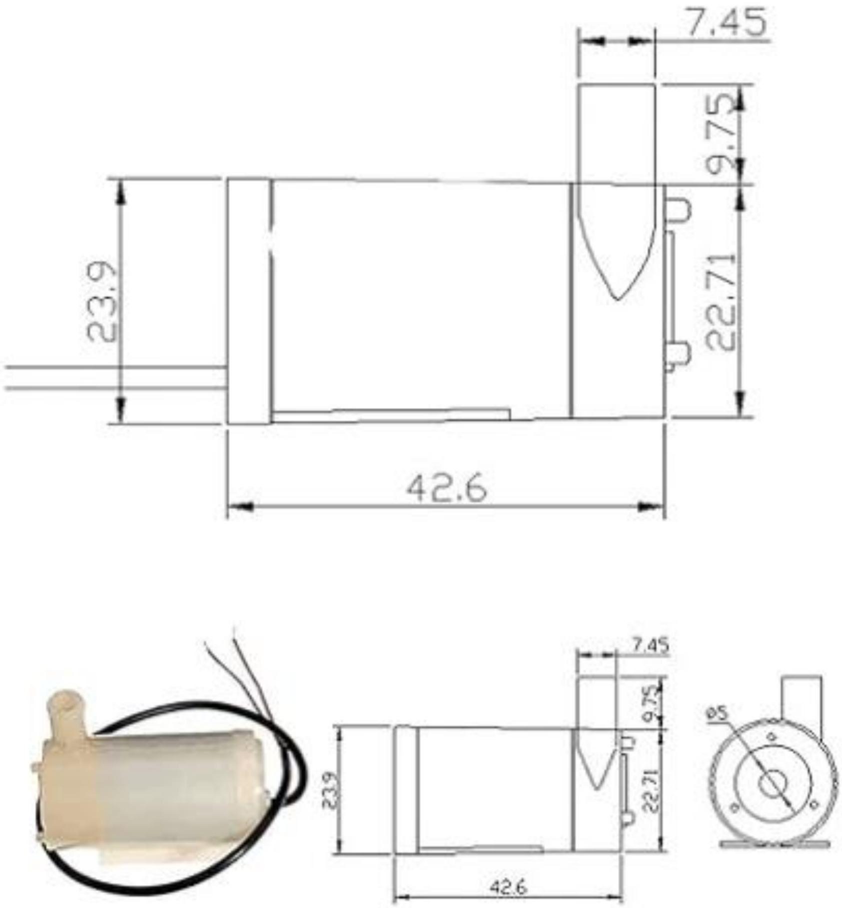
Figure 2 - Soil Moisture Sensor

II.3. MINI PUMP

This small 6-12V pump is activated by the signal sent from the SMS through the relay module. This system uses a small, low-power DC pump capable of lifting water from a reservoir to the plants. Its miniature size makes it ideal for small-scale irrigation. Despite its small size, this pump is capable of up to 100 L/hr.



Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx



Model	Voltage Scope (DC)	Current (A)	Power (W)	Max Water Head (M)	Max Flow Rate (L/H)	Starting Voltage	waterproofing grade
JT-DC3W-3	3V	0.12	0.36	0.35	80	1	IP68
JT-DC3W-4.5	4.5V	0.18	0.91	0.55	100	1	IP68

Figure 3 - 6-12V Pump

II.4. RELAY MODULE

Also from the pump kit is the 5 V relay module used between the SMS and the pump. The relay module acts as a switch to control high-current devices like the water pump. The Arduino sends a low-power signal to the relay, which completes the circuit to power the pump. It provides electrical isolation to protect the Arduino from high voltages. For this system, the relay ensures the pump operates safely and only when it has the (literal) green light from the Arduino and therefore SMS. As seen in the picture below there is a red light

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

on. This is just to show that it is receiving power via the Arduino. When the pump turns on, the green light will illuminate.

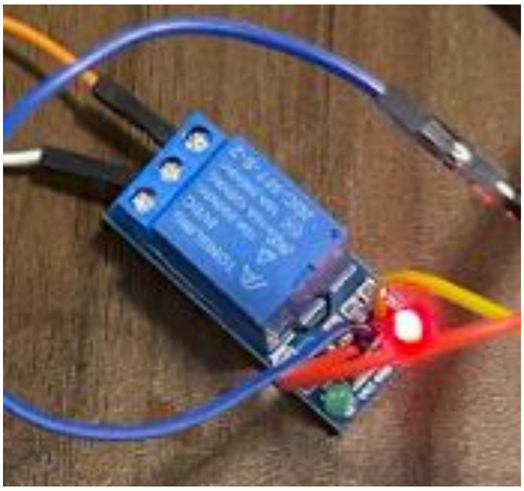


Figure 4 - 5 V Relay Module

II.5. BATTERY PACK

The next important item from this kit that was necessary to build the system is the battery pack. The pump needed its own power supply so the battery pack was needed to operate. The battery pack holds a 9 V battery inside so if an operator did not want to buy the pack, they could use the 9 V battery included in the Arduino kit. Having the pack also helps provide portability and ensure uninterrupted operation of the system. This also shows that the system can function in locations without direct power, like a garden or a corner without the need for an outlet.

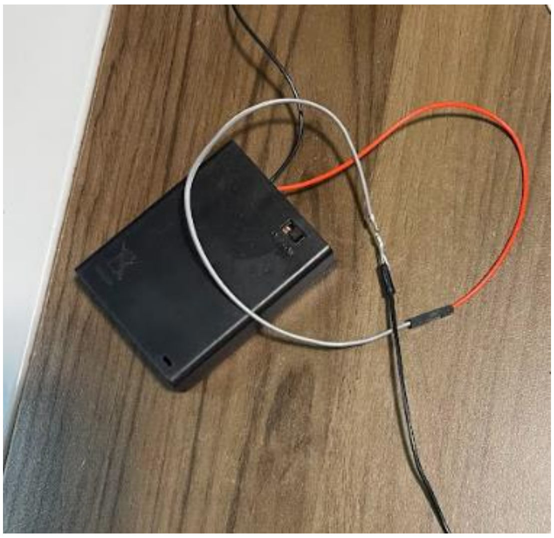


Figure 5 - Battery Pack

II.6. VINYL TUBING

An item on the list that could be easily forgettable is the piping to get the water from the pump to the plant. It acts as a conduit, ensuring an efficient and mess-free water delivery system. Flexible, durable tubing is preferred for ease of setup and to prevent leaks. This is why it’s preferable to use the tubing provided in the pump kit as it has a tight, waterproof fit onto the pump and is not easily broken. The tube provided is made of vinyl, meaning it is

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

durable but malleable for any use the operator requires. At one point, there was a thought of being able to put the pump under the plant and therefore pump water directly to the roots, but for the simplicity of the prototype the piping was deemed easier. Through this action, the system was more capable of avoiding waste and enhancing overall efficiency.

Figure 6 - Vinyl Tubing

II.7. WATER RESERVIOR

The next material on the list is something that everyone already has, a bowl. This part of the system does not have to get fancy; a simple bowl will suffice. In the testing phase, the operator does not even have to have a plant necessarily. While testing, creating a circular system with the pump and tube will make it more neat and easier to deal with. This can be done by placing the pump in the bowl of water and making the tubing stay in the water. This is beneficial while making sure that the system works. When ready to adjust SMS sensitivity, a plant in a pot is needed to make it as accurate as possible.

Figure 7 - Circular System

II.9. ARDUINO IDE CODE

The final "material" in this system is not really a material, but is still very important. The code is the backbone of the system, defining how the components interact. It is written in the Arduino programming language (similar to C++). The code reads the analog signal from the

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

soil moisture sensor, compares it to a threshold value, and activates the relay module to power the pump.

```
int percentage=0;

void setup() {

  pinMode(3,OUTPUT);

  Serial.begin(9600);

}

void loop() {

soilMoistureValue = analogRead(A0);

Serial.println(percentage);

percentage = map(soilMoistureValue, 490, 1023, 0, 100);

if(percentage < 10)

{

  Serial.println(" pump on");

  digitalWrite(3,LOW);

}

if(percentage >80)

{

  Serial.println("pump off");

  digitalWrite(3,HIGH);
```

Figure 8 - Arduino Code

II.10. METHODS

Once all the materials were collected and ready to be put together, the article by Circuit Digest and its corresponding circuit diagram, as well as additional help from creators on YouTube were helpful in building the AWS. I started building the system at the SMS. The only wires to attach to the SMS were the positive and the ground, which was attached to the sensor module that is used to adjust the sensitivity. From there I connected the sensor module's VCC, GND, and DO terminals to the Arduino's Vin, GND, and digital 6 terminals respectively. The next step was to connect the 5 V relay module. I connected the relay module from its IN, GND, and VCC terminals to the Arduino's digital 4, GND, and the 5 V terminals. Next, we connected the pump by attaching the positive wire to the relay modules NO terminal. The pump's negative terminal was connected directly to the positive wire of the battery pack. The battery packs ground was connected to the relay module's NC terminal. Finally, the Arduino was connected to the laptop and the code was running. Now that the AWS was running, it was possible to determine the correct level of moisture the sensor needed, and adjustments were made to the sensor module.

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

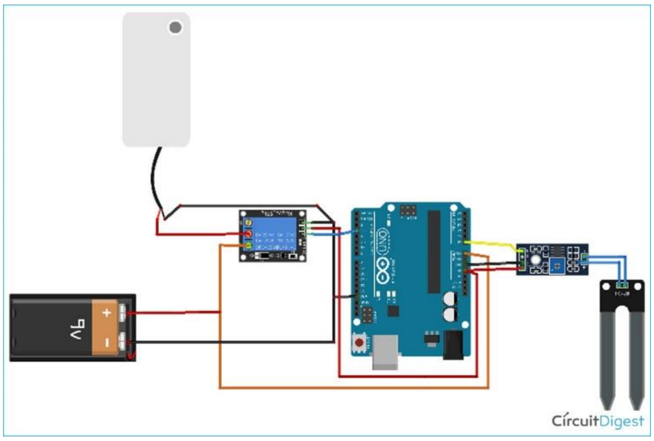


Figure 9 - Circuit Diagram from Circuit Digest

III. Challenges Encountered

During the development of the Automatic Plant Watering System, several challenges were encountered that required troubleshooting and iterative problem-solving. The first problem was that the pump was not turning on at all. I checked all of the wires to make sure they were in the right place. On this inspection, I found that VCC and IN wires were flipped on the wrong sides of the relay module. From there, the next biggest challenge in the development of the project was that the pump would not shut off when the SMS was wet. The first check was to see if the SMS was working accurately. To do this, the screw on the sensor module was turned to adjust the sensitivity until it was accurate. The sensitivity was correct when the green light turned on when the SMS was removed from the water. The problem continued, however, even when the sensor said it should be shut off. After searching additional resources online, I discovered that in the initial creation of the AWS, we had connected the grounds together and positives together between the pump and the battery pack. This created a nonstop stream of power to the pump, overriding the relay module. To fix this, I took the advice of Techie Lagan on YouTube and connected the negative of the pump to the positive of the battery back and the positive of the pump to the relay module. Although the answer is not quite clear on how it worked correctly after that, results cannot be argued with.

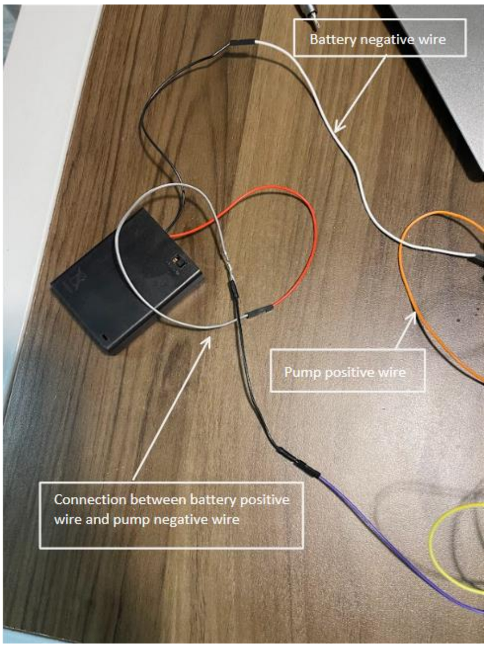


Figure 10 - Pump Troubleshooting

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

IV. Results & Discussion for Improvement

The AWS successfully achieved its primary objective of automating irrigation based on soil moisture levels. The system consistently activated the water pump when the SMS was dry and stopped watering when sufficient moisture was detected. This demonstrated its capability to conserve water and reduce the manual effort required for plant care. However, the project also revealed areas for improvement. One such improvement that came to mind during the development was that while the sensor would know if the soil was moist or not, it did not know how much water the plant needed daily. This could be solved through some kind of wireless application that, based on the plant type, could override the SMS and would irrigate the plant by the predetermined amount that would be in the database. When it came time for the plant to be watered, the SMS would be used as a kind of shut-off detector. EX. According to the database plant x needs to be watered once every 3 days, and on the third day the AWS waters the plant until the SMS shuts the pump off.

Another big improvement would be developing a way to have the system work without being directly connected to a computer feeding its code. The way to do this is not exactly clear and would require more research to determine the best way forward, but this improvement would make the system more ergonomic. A final improvement would also be some kind of water level sensor to let the user know when the reservoir is low. This would ensure that the plant would never go thirsty.

V. Conclusion

The development of the AWS demonstrated the effectiveness of combining simple electronics and programming to create a functional and practical solution for automating irrigation. By combining an Arduino Uno, a soil moisture sensor, a relay module, and a water pump, the system successfully monitored soil moisture levels and watered plants as needed, reducing manual effort and conserving water. This project highlights the potential for such systems to benefit home gardeners and possibly, small-scale agricultural operations. Despite the potential of the system, there are some limitations at the moment. From these limitations, there are opportunities to improve the AWS through things like a mobile app and additional sensors for more in-depth analysis. These enhancements could make the system more efficient and scalable for diverse applications. Overall, this project serves as a cost-effective solution to modern home irrigation challenges and provides a foundation for future advancements in automated plant care systems.

VI. Data Availability Statement

- The code that supports the system is openly available in the description at ["ROBOMAC" on YouTube]
  at [https://www.youtube.com/watch?v=UinHdYklRkw] , (Figure 8)
- The step-by-step guide for building the system is available at ["Techie Lagan" on YouTube] at [https://www.youtube.com/watch?v=iwkE_HWU-6M&t=223s].
- The hardware specifications and components used in this project are listed at [Amazon] at [https://www.amazon.com/dp/B07VRMRQWR?ref=ppx_yo2ov_dt_b_fed_asin_title], [Figure 3].

Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

- The circuit diagrams and hardware setup detail is openly available on the Circuit Digest website at [https://circuitdigest.com/microcontroller-projects/automatic-irrigation-system-using-arduino-uno], (Figure 9).
- The Arduino kit used in this project are listed in [Vilros] at [https://vilros.com/products/arduino-ultimate-starter-kit], [Figure 1].

VII. Conflict of Interest

The author declares no financial, professional, or personal conflict of interest related to the creation of this project. The concept for the Automatic Plant Watering System originated independently as part of previous project ideas and personal interest in an automated system. This project builds on those initial ideas and does not involve any external influences or affiliations that could affect objectivity.

VIII. Acknowledgements

This project was fully self-funded, with no external financial contributions. I would like to acknowledge the technical support and guidance provided by online resources, including Circuit Digest as well as instructional videos on YouTube creators “Techie Lagan” and “ROBOMAC.” I would also like to share an additional acknowledgement of Connor Ottinger in helping to construct and troubleshoot the AWS physically. I also acknowledge ChatGPT for assistance in organizing and refining the written report. All of these resources played a significant role in the creation of the AWS and the coinciding report.

IX. References

[1] “Arduino Uno 3 Ultimate Starter Kit: Vilros.Com: .” *Vilros.Com*, vilros.com/products/arduino-ultimate-starter-kit . Accessed 4 Dec. 2024.

[2] AITRIP. “Soil Moisture Sensor Kit Automatic Watering System Manager with Mini Water Pump for Arduino DIY Kit .” *Amazon*, www.amazon.com/dp/B07VRMRQWR?ref=ppx_yo2ov_dt_b_fed_asin_title . Accessed 4 Dec. 2024.

[3] Rajesh. “Automatic Irrigation System Using an Arduino Uno.” *Circuit Digest - Electronics Engineering News, Latest Products, Articles and Projects*, Circuit Digest, 5 Oct. 2021, circuitdigest.com/microcontroller-projects/automatic-irrigation-system-using-arduino-uno .

[4] ROBOMAC. “DIY Automatic Plant Watering System Using Arduino .” *YouTube*, YouTube, 28 July 2022, www.youtube.com/watch?v=UinHdYklRkw.

[5] Techie Lagan. “How to Make Automatic Plant Watering System Using Arduino UNO and Soil Sensor.” *YouTube*, YouTube, 29 July 2021, www.youtube.com/watch?v=iwkE_HWU-6M&t=223s .

[6] OpenAI. “Writing Assistant.” *ChatGPT*, chatgpt.com/ . Accessed 4 Dec. 2024.