

AN-NAJAH NATIONAL
UNIVERSITY PALESTINE



A PROJECT WORK REPORT ON
Green House Machine

By

Lama Kharaz¹ and Seema Ziyad²

¹kharrazlama@gmail.com

²seemaabubaker6@gmail.com

Under the supervision of
Dr.Hanal Abu Zant

A report submitted in partial fulfillment of the requirements for Bachelor
degree in Computer Engineering in the Faculty of Engineering & Information
Technology - Hardware Project

AUG 2022

Acknowledgements

Firstly, we would like to thank Dr. Hanal Abu Zant, our project supervisor for Green house. Dr. Hanal Abu Zant always kept his office door open when we encountered problems or had questions about our project.

Thank you to An-Najah National University, especially the Computer Engineering Department of the College of Engineering, for providing us with the skills and knowledge we needed to complete this work.

Our families have provided us with continued support and encouragement throughout our school years and through our work and writing processes. Without them, this achievement would not have been possible. Thank you.

Seema & Lama

Disclaimer

This report was written by means of students at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the students.

An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

Acknowledgment.....	1
Disclaimer	2
Table of Content.....	3
Table of Figure	4
Abstract.....	5
Chapter 1 : Introduction	6
1.1 introduction	6
1.2 Objective of the project	7
1.3 Scope of the project.....	7
1.4 Importance of the project	7
Chapter 2 : Constraints, Standards/ Codes and Earlier course work	8
2.1 Constraints and Standards	8
2.2 Symbols and Earlier course work.....	8
Chapter 3: Literature Review.....	9
Chapter 4: Design and Implementation.....	9
4.1 Hardware and Software Specifications	10
4.1.1 Hardware Components	10
4.1.2 Software Components	21
Chapter 5 : Results and Analysis.....	4
5.1 Nextion Code	21
5.2 Green House Arduino.....	31
5.3 Circuit Layout	57
Chapter 6 : Discussion	58
Chapter 7 : Conclusions and Recommendation.....	59
7.1 Conclusions	59
7.2 Recommendations	59

List of Figure

4.1.1: Arduino mega.....	10
4.1.2: Nextion basic display	11
4.1.3: Nextion basic HMI display	12
4.1.4: RFID-RC522 Sensor.....	13
4.1.5: Global System for mobile (GSM A6)	14
4.1.6: DHT11–Temperature and Humidity Sensor	14
4.1.7: Light Dependent Resistor	15
4.1.8: Fan 12V.....	16
4.1.9: IR Flame Sensor.....	16
4.1.10: Ultrasonic Fogger Maker	17
4.1.11: Ultrasonic sensor (HC-SR04)	18
4.1.12: Relay 12V.....	19
4.1.13: Buck Convertor	19
4.1.14: Servo motor	20
4.1.15: tungsten lump.....	20
5.1.1: Enter card GUI.....	22
5.1.2: Customer Info GUI.....	23
5.1.3: Types of plant Info GUI	23
5.1.4: Admin page GUI.....	24
5.1.5: add new customer GUI.....	25
5.1.6: view sensor value GUI.....	26
5.1.7: store after adding new plant GUI	27

Abstract

We made a machine that includes a group of seedlings and their types, and provides the seedlings with the appropriate conditions in terms of measuring temperature and humidity in addition to the lighting system and the fire protection system.

And we did a store that initially relied on a touch screen (nextion) and a login card, so the user entered his card via RFID, and then his data appeared on the screen, and through it he entered the number of seedlings to be purchased and the type after making sure that the order was accepted, the order will be successfully finished and the door will be opened, as well as The system is based on the principle of using humidity and temperature sensors to check the appropriate conditions for the seedlings, when the temperature exceeds the required value and the humidity is low, the fan and humidifier are turned on until the required value is reached. In addition to the fire protection system, if there is a fire on the farm, the system will call the administrator and activate the night lighting system, so that in the dark this idea is not available in Palestine, it is similar and close to the idea of a vending machine

We were able to set up the system almost successfully, but we encountered difficulties in the mechanical part and everything that was mentioned above was implemented.

Chapter 1

Introduction

1.1 Background

As technology advances in all aspects of people's lives, developers seek to improve working life and save effort by using machine technology and making it available to them at lower costs. Direct selling plant seedlings is one such technology movement that promises to make life simpler and more abundant for the workforce.

The process of direct electronic sales through the use of the touch screen (Nexion) that enables you to view the store and what it contains of the types of seedlings and the price of each seedling by entering the store using an (id) card. The subscriber in the green house transfers the user to an interface that shows his information in terms of his name, unique number (identifier) and his balance in the card, then he transfers him to the interface of the home that shows the species. From seedlings in the green house, when he chooses the type of seedlings to be purchased, he will be transferred to an interface showing the type of seedlings and the number of available seedlings of that type he chooses according to what is inside the green house and the price according to the type of seedling ` who chose it. If the user's balance is less than the amount required to purchase, he will not be able to purchase, and an interface will appear to him that he cannot buy because of his financial balance.

In addition to the store owner (supervisor) card, which enables the store owner to add new types of seedlings or change the price of any seedlings who wants to change or modify the number of seedlings or add a new type of seedling (expansion) through a specific interface (update) or it is possible to modify information users.

And the presence of a feature that enables the supervisor to set the temperature and humidity required for the green house.

Another function is to monitor the health of the seedlings, such as monitoring and controlling the temperature and humidity in the greenhouse. For example, when the temperature and humidity rise, the system turns on the fans and fog machines, and when it falls, the system turns on (Tunstun blocks) to raise them.

It also provides an automatic function for the lighting system so that the system turns on the lights when it gets dark.

In addition to another feature, it's a fire sensor, so if there's a burning smell, the system will call the administrator to be careful.

1.2 Objective of the project

Initially saving energy for the greenhouse owner and reducing their presence in it by intelligently controlling some things, such as what we will be doing in this project, namely purchasing, heat, humidity and sales.

1.3 Scope of the project

A smart greenhouse can actually span multiple areas. Smart shopping, smart lighting, smart protection design, smart temperature and humidity, greenhouse monitoring and smart control. In our project, we wanted to develop a plan to help control the greenhouse in a time-saving and labor-saving way, which depends on smart things, sensors and smart technology.

1.4 Importance of the project

Seedlings play an important role as a source of environmental production.

Seedlings occupy a leading position it can be used for food and decoration.

Seedlings play an important role as a source of ecological balance, maintaining soil moisture by storing water in roots and stems, protecting the land from the threat of desertification, and helping to remove toxic gases and dust.

Relative to humans, it is in the lead. It is used in the manufacture of perfumes, paints, poultry feed and cosmetics.

This is why they are very important for a person.

Chapter 2

Constraints and Standards / Codes and Earlier course work

2.1 Constraints and Standards

The combination of technology and human efforts improves and promotes the quality of life, saving time and energy for everyone involved, but we may encounter many obstacles and are in a dilemma at the beginning. In addition, we also lack advanced communication networks such as 4th and 5th generation and the problem of internet speed. In fact, the occupation of Palestine is a major obstacle to many projects.

One of the biggest challenges in this project was the mechanical part to get the seedlings out of the greenhouse, and we initially resorted to using the spring, but the mechanical properties of the spring were not suitable in terms of the lack of a compressible spring and the displacement of the spring to the right side when moving it, and also in the case The project has been expanded, it may need a spring of large sizes and this is not considered easy and flexible, so the idea of spring for seedlings is not good

In the end, we were able to turn on the mechanical part and complete the process of removing the seedlings from the greenhouse by moving them on an alternating inclined surface and passing them with moving bar.

2.2 Symbols and Earlier course work

This project was carried out by Arduino and Nextion to design a smart touch screen and some sensors such as light sensor to control lighting time and intensity, temperature and humidity sensor and fire sensor to issue an alarm in case of fire and a GSM system will be implemented to manage the communication procedures necessary to perform.

In terms of courses, we took an Arduino course at the IEEE Engineers Society. This made working on our project easier as we completed it faster than expected. We didn't take any classes on the Nextion screen and learned how to design a screen interface on it while we working , how to code the special programs that come with it, and how to connect it to an Arduino. In this case we had to use Google as this is the first time we've looked into this.

Chapter 3

Literature Review

The idea to sell came from machines in the market. These vending machines have a great history, the first commercial coin-operated vending machine was exhibited in London, England in the early 1880s.

Vending machines are defined as self-contained automatic machines that dispense goods (such as candy or soft drinks) or provide services (such as a weighing scale) when coins are inserted. [1]

Today, the vending machine industry generates approximately \$20 billion in revenue each year in the United States. A standard vending machine costs about \$3,400 on Amazon, and there are an estimated eight million vending machines in America, according to Vending Market Watch. [2] Now it is widely spread to sell chocolate, Coca-Cola, drinks, candy and toys. And now we used it to sell seedlings, and inside it we used a method to create the right conditions for a plant to be inside the machine.

The idea to sell the plant machine came from a company (Beijing Sino-Horizon Agriculture Science and Technology Co., Ltd) located in China, the plant vending machine can be used independently or can be combined with a plant factory which is specialized for live plants such as fruits, vegetables and flowers to provide them with good environmental conditions, and these conditions are necessary for the growth of Plants are kept alive and fresh. It should also be noted that this vending machine can be placed in shopping centers (commercial malls) and flower markets for the purpose of selling fruits, vegetables and potted plants, and therefore, it is very useful in cities that lack vegetable growing areas or in places that are not available to the farmers market.[3]

Chapter 4

Design and Implementation

Green House is a platform that uses the Arduino hardware platform designed, for the seedling procurement process and to control the appropriate conditions for the seedling.

The Arduino board works as a central processing unit and a storage device, and there is many ways of input: the Nextion touch screen.

And the presence of (RFID) that reads the information of the customer who wants to buy from the green house.

➤ **(GSM)** : is used to send warning messages to the mobile phone.

- **(DHT 11)** : is also used to check humidity and temperature.
- **(LDR)** : It uses to measure the intensity of illumination and distinguish between day and night.
- **(Ultrasonic Fogger Maker)** : Increases humidity.
- **(IR Flame Sensor)** : senses the presence of fire.
- **(Servo motor)** : opens the store door.
- **(Power supply)** : provide 12v.
- **(Fan)** : Cooling green house.
- **(Relay)** : Switches for electronic parts.
- **(Buck)** : Convert from 12v to 5v.
- **(Tungsten lamp)** : heating green house.
- **(Moving bar)** : to moving seedlings
- **(Ultrasonic distance sensor)** : to measure distance of seedlings
- **(Solenoid)** : device that convert energy into linear motion.

4.1 Hardware and Software Specifications

4.1.1 Hardware Components

For this, we have used an Arduino Mega (2560 Rev3), Nextion Basic Display , GSM,DHT11, RFID, LDR , Fan , IR Flame sensor , Relay ,Ultrasonic Fogger Maker ,Buck , Ultrasonic distance sensor , solenoid , tungsten lamp and Power supply.

➤ Arduino Mega (2560 Rev3)

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator. It also has a 256 KB flash memory. It has a Mini-B USB port and a reset button.

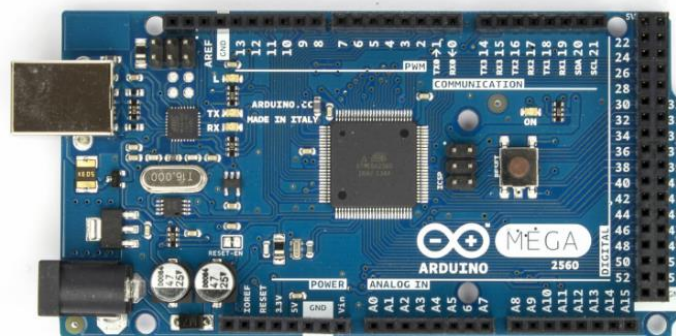


Figure 4.1.1: Arduino mega

➤ Nextion Basic Display

Nextion is a Human Machine Interface (HMI) solution that combines an onboard processor and a memory touch display with Nextion Editor Software for developing HMI GUI projects.

Nextion Editor Software allows you to quickly develop HMI GUIs by dragging and dropping components (graphics, text, buttons, sliders, etc.) and coding component interactions on the displayed page with ASCII text-based instructions.

The Nextion HMI display is connected to the peripheral MCU via TTL Serial (5V, TX, RX, GND), providing event notification for the peripheral MCU to respond, the peripheral MCU can easily update the progress and status under Send instructions back to the Nextion using plain ASCII text show.

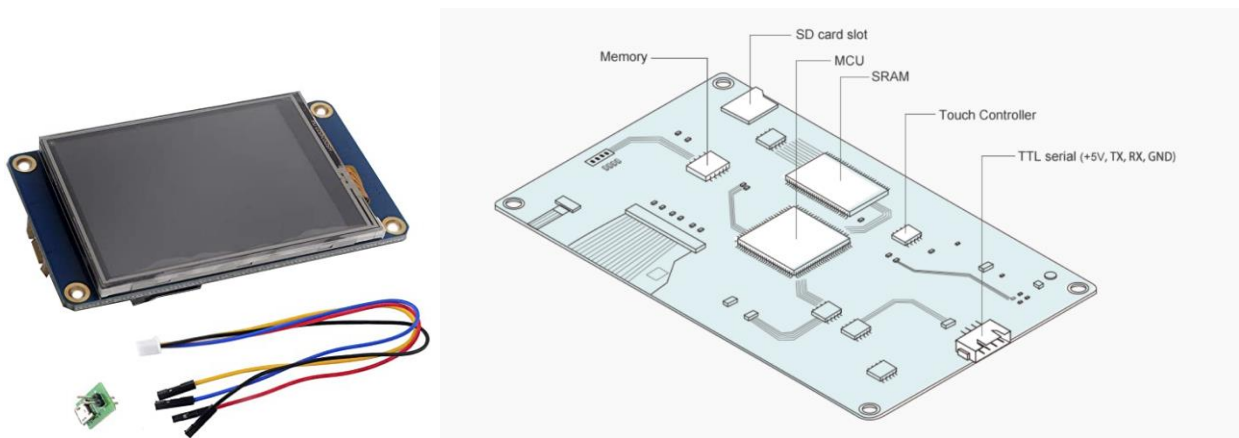


Figure 4.1.2: Nextion basic display

MODEL NAME	NX4024T032
Size	3.2"
Resolution	400*240
Touch Panel	RTP
Color	65536
Flash	4 MB
RAM	3584 Byte
MCU	48 MHz

Figure 4.1.3: Nextion basic HMI display

➤ **RFID technology**

What is RFID (radio frequency identification) ?

RFID (radio frequency identification) is a wireless communication technology that uses electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum in order to uniquely identify a person.

RFID consists of three main part :

1. The card that contains the transmitter and information
2. Reading and transmitting device
3. Computer programs and databases Radio Frequency Identification (RFID)

How does RFID work?

When a scanning antenna and transceiver are combined, they are called an RFID reader or interrogator. There are two types of RFID readers - stationary readers and mobile readers. An RFID reader is a portable or permanently attached network-connected device. It uses radio waves to transmit a signal that activates the tag. When activated, the tag sends the wave back to the antenna, where it is converted into data.

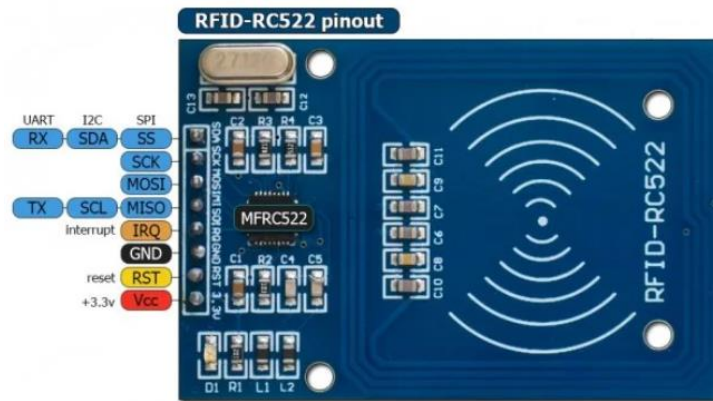


Figure 4.1.4: RFID-RC522 Sensor

➤ GSM

The Global System for Mobile Communications (GSM) is a standard used in cellular communication networks. Of all the cellular technologies in use today, GSM is one of the most widely used.

The GSM module provides connectivity as well as wireless data transfer. Lightweight and easy to use, these devices have surprisingly low power consumption for the amount of work they can get done. They can be used to track communication items, connect remote site monitoring systems to your LAN, and many other applications.

A GSM module is a special type of device that accepts a SIM card and operates through a subscribing cellular service provider, just like a cell phone or pager. From a cellular carrier's perspective, a GSM modem looks like a phone. The difference between a phone and a mod is the flexibility of the application.

A cellular connection is a quick and efficient way to start using SMS alerts in a network monitoring solution, as it does not require a special subscription to a separate service provider. In most parts of the world, a GSM module is a cost-effective solution for sending and receiving alarm notifications, because it uses the existing network and does not have to add hardware to your infrastructure to connect directly to your remote location.



Figure 4.1.5: Global System for mobile (GSM A6)

➤ Humidity Temperature Sensor (DHT11)

The **DHT11 temperature and humidity sensor** has a temperature and humidity sensor complex with a calibrated digital signal output. Adopt unique digital signal detection technology and temperature and humidity detection technology to ensure high reliability and excellent long-term stability. The sensor consists of a resistive humidity sensing component and an NTC temperature sensing component, and interfaces with a powerful 8-bit microcontroller that provides superior quality, fast response, noise immunity, and cost-effectiveness.

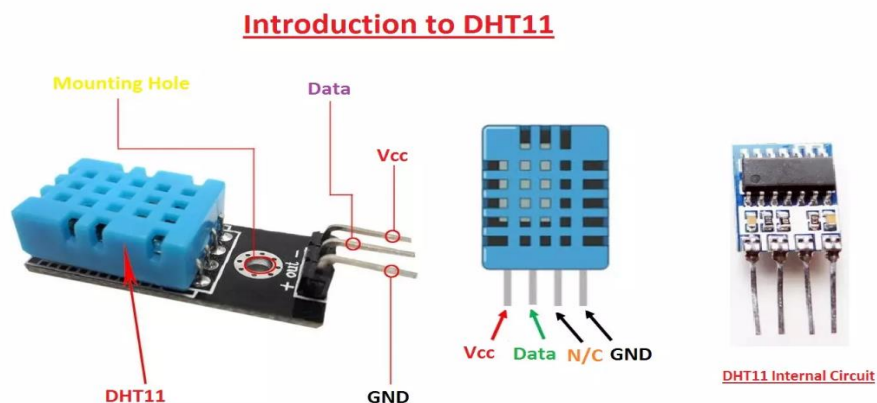


Figure 4.1.6: DHT11–Temperature and Humidity Sensor

➤ **Light Dependent Resistor (LDR)**

Light Dependent Resistor is a resistor whose resistance depends on the amount of light falling on its surface, called an LDR or photoresistor. It can also be called a photoresistor, photocell or photoconductor. Light falling on a resistor changes the resistance of the resistor, making these resistors widely used in many circuits that require light sensing.

These resistors have a variety of functions and resistances such as: B. LDR In the dark, turn on the light in the dark or turn off the light when it is bright. Typically, a photoresistor has a resistance of 1 MOhm in the dark and a few KOhm in the light.

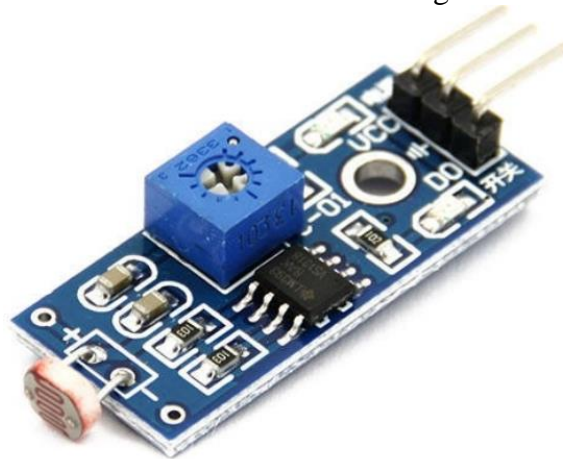


Figure 4.1.7: Light Dependent Resistor

➤ **FAN**

DC fans also known as DC Brushless Fans are basically cooling fans engineered to meet requirements such as higher air flows at high static pressures.

The direct current fans, or DC fans, are powered with a potential of fixed value such as the voltage of a battery. Typical voltage values for DC fans are, 5V, 12V, 24V, and 48V, we use 12V. It has motors that rely on permanent magnets in order to attract and repel a rotor around the axis using electronic switching.

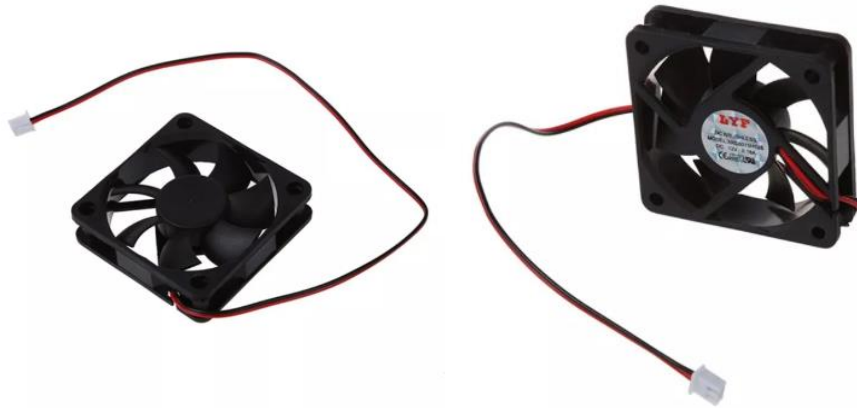


Figure 4.1.8: Fan 12V

➤ IR Flame Sensor

A Flame Sensor module or Fire Sensor module is a small size electronics device that can detect a fire source or any other bright light source. This sensor basically detects IR (Infrared) light wavelength between 760 nm – 1100 nm that is emitted from the fire flame or light source. The flame sensor comes with a YG1006 Phototransistor sensor which is a high speed and high sensitivity. Two types of IR Infrared Flame Sensor Module available in the market one has three pins (D0, GND, VCC) and another one has four pins (A0, D0, Gnd, Vcc) both are can be easily used with Arduino and other microcontroller boards.



Figure 4.1.9: IR Flame Sensor

➤ Ultrasonic Humidifier Maker Fogger

Ultrasonic Humidifier Maker Fogger using high frequency oscillation. The water spray device, the connecting piece and the circuit, high frequency oscillatory fog atomizing force will be produced under the. Containing air ion in thick fog can increase the air humidity and fresh air. This product uses the advanced large room, atomized mist is generated and the ideal of good decorative effect, in the rockery, fountain, potted landscape.



Figure 4.1.10: Ultrasonic Fogger Maker

➤ Ultrasonic Distance Sensor (HC SR04)

The HC SR04 ultrasonic sensor is a popular sensor for measuring distance and sensing the area of an object. On its front, the sensor has two eyes that resemble the item. These two eyes act as ultrasound receivers and transmitters. The working of this ultrasonic sensor assumes that distance is the product of velocity and time.

When an ultrasonic transmitter emits ultrasonic waves, the waves travel through the air and bounce off the sensor after encountering obstacles. The ultrasonic receiver then observes the reflected waves. The module has a built-in looping system that calculates the time it takes for a wave to return and become an echo. The distance is then calculated using a microprocessor or

microcontroller.



Figure 4.1.11: Ultrasonic sensor (HC-SR04)

➤ Relay 12V

An electrical relay is an electromechanical switch used mainly for switching application either high voltage or low voltage switching, and they can be also used with microcontrollers like the Arduino for controlling high voltage devices.

It is a type of heavy-duty, remote-control switch able to handle high-current accessories, yet capable of being actuated by substantially less current. Relays install between the power source and the electrical accessory requiring on/off power. When the relay is energized, the high current to operate the accessory flows from the power source, through the relay, and directly to the part.

Latching relays require only a single pulse of control power to operate the switch persistently. Another pulse applied to a second set of control terminals, or a pulse with opposite polarity, resets the switch, while repeated pulses of the same kind have no effects. Magnetic latching relays are useful in applications when interrupted power should not affect

the circuits that the relay is controlling.

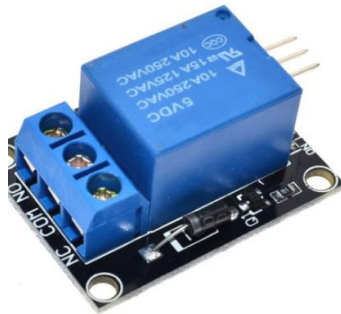


Figure 4.1.12: Relay 12V

➤ Power supply

This 12V DC power supply can provide a power source for any electronic device that has an M-type barrel and requires 12 volts and up to 2 amps of power.

➤ Buck

A buck converter is a step-down DC to DC converter. For a DC–DC converter, input and output voltages are both DC. It uses a power semiconductor device as a switch to turn on and off the DC supply to the load. The switching action can be implemented by a BJT, a MOSFET, or an IGBT. Figure 1 shows a simplified block diagram of a buck converter that accepts a DC input and uses pulse-width modulation (PWM) of switching frequency to control the switch. An external diode, together with external inductor and output capacitor, produces the regulated dc output. Buck, or step down converters produce an average output voltage lower than the input source voltage.

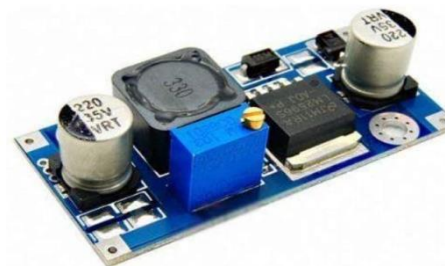


Figure 4.1.13: Buck Convertor

➤ Servo motor

A servo motor is a self-contained electrical device that rotates parts of a machine with high efficiency and precision.

The output shaft of this motor can be moved to a specific angle, position and speed that a normal motor does not have.

Servo motors use a normal motor and couple it with a position feedback sensor.

The controller is the most important part of a servo motor designed and used specifically for this purpose.

we used it to open the door by giving the servo motor an angle, an angle of 0 to close the door, and an angle of 90 to open the door.



Figure 4.1.14: Servo motor

➤ Tungsten lump

A **Tungsten** lamp is an electric lamp that emits light from a tungsten metal filament. Current flows through the thin wire, raising its temperature until it glows. The filament has a high resistance and is protected from oxidation by a vacuum glass bulb, or sometimes contains an inert gas that protects the filament from evaporation, such as B. A mixture of argon and nitrogen. We use it to increase the temperature of the house by heating the surrounding air.



Figure 4.1.15: tungsten lump

4.1.2 Software Components

To make the system works as Green house or to make components work together, we must program the Arduino and the Application.

➤ Arduino IDE

The Arduino Integrated Development Environment (IDE) is a piece of software that allows for enhanced and aided code editing, compilation, and debugging. The Arduino IDE is based on the Java Platform. It's used to write and upload code to Arduino-compatible boards. The Green House are programmed using the Arduino.

➤ Nxtion Editor

The GUI interfaces are designed using Nxtion Editor so that the program sends data from the screen to the Arduino.

Chapter 5

Results and Analysis

5.1 Nxtion Code

We designed the touch screen interface to send commands to the Arduino board, so the project consists of different types of seedlings you can buy, the admin adds each type of seedlings, the number of seedlings available in green can change the house, or change the price of the seedlings.

It consists of two main ideas:

1. Customer card entry:

Each customer can enter the Green House store via a card so they can browse the store and select seed to buy. Before making a purchase, he checks the card to see if there is enough credit to buy the seedlings or he select any type of their.

p0



t6

Figure 5.1.1: Enter card GUI

In the above image, the system is waiting for the user to enter his card to go to the store to buy, but if the card is unknown in this system and not entered into the database by the administrator, he can't enter this and tells him he didn't recognize the card.

if the card is unknown then this page in the picture below will be appear



But if he successfully entered the store then this page in the picture below will be shown and it shows him his name, the amount available to him, and his unique ID number



Figure 5.1.2: Customer Info GUI

Then when he click in to store button ,this page appear in the image below , his name and number appear at the top of the page, as well as the types of seedlings available inside the machine

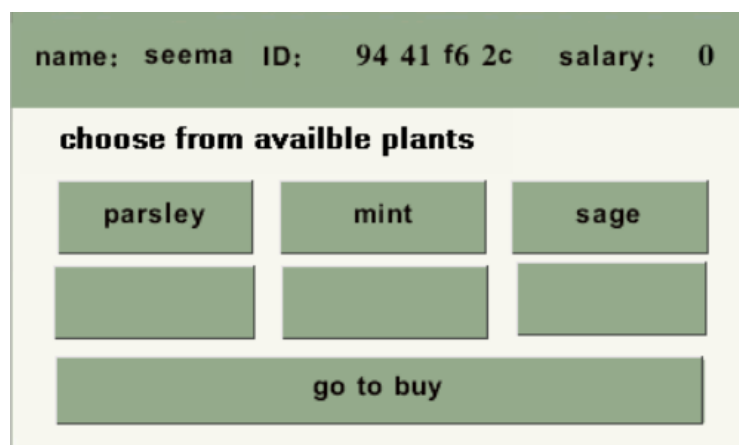


Figure 5.1.3: Types of plant Info GUI

2. Admin card entry:

The admin has a special card that distinguishes him from the client, so he can change the number of seedlings. The number of them can be decreased or increased (expanded).

It can also change the price of seedlings or the name of them, and administrators can also add new customers or change his information with new cards. New types of seedlings can also be added to the Greenhouse Store.

From this page, he can also control the proper temperature and humidity required for that house and view these values through his sensors.

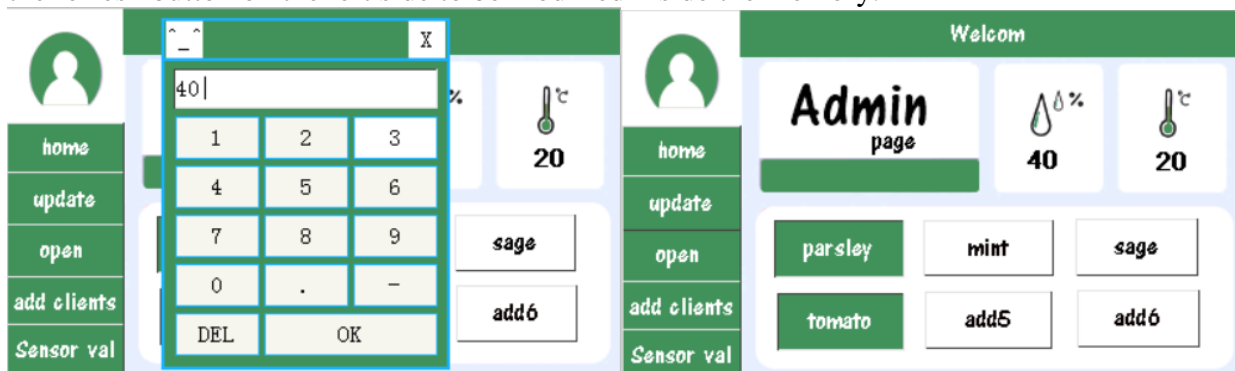
So if the admin entered his card this page in picture below will be shown



Figure 5.1.4: Admin page GUI

On this page that appears in the images above , when he click in the open button the door will be opened .

On this page that appears in the images below, it is possible to control the temperature and humidity to be available inside the machine by pressing it and changing the value, then pressing the refresh button on the left side to be modified inside the memory.



In the figure 5.1.4 , if the admin wants to add new customer, he presses the add clients button the, then the following page shown in the figure 5.1.5 appears for him , so when he enter the card , his ID appears to him, then he presses the name and account to enter the values, and then presses the update button to be added to the database as shown un figure

The figure consists of three screenshots of a web application interface for managing customer information.

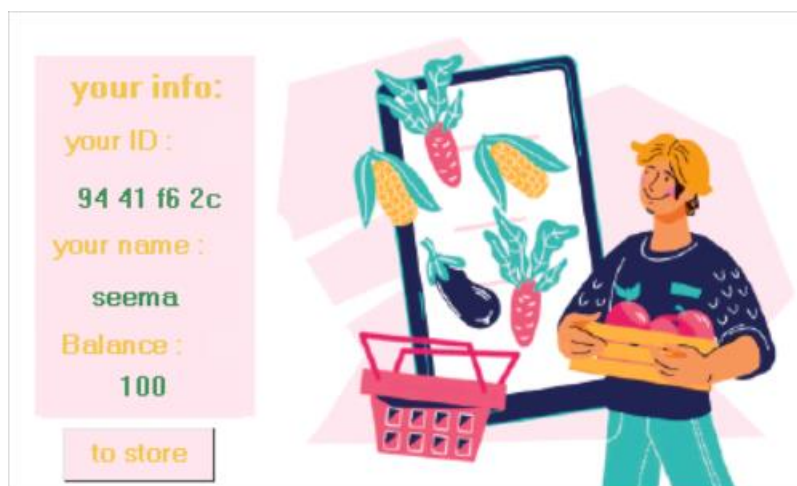
Screenshot 1 (Top Left): The page has a green header with the text "please enter the card". On the left is a sidebar with a "home" button and an "Add/update" button. The main form contains three input fields: "Name :", "ID:", and "Balance :". The "Balance :" field contains the value "0".

Screenshot 2 (Top Right): The page has a green header with the text "Update completed successfully". The "ID:" field now contains the value "94 41 f6 2c".

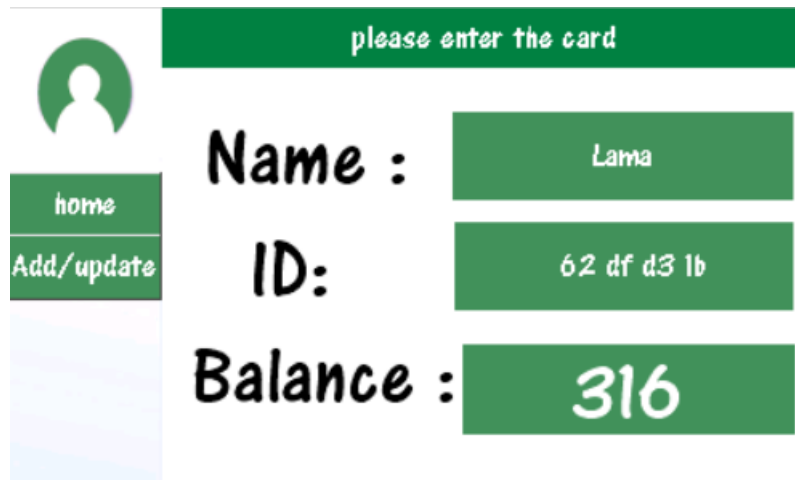
Screenshot 3 (Bottom): The page has a green header with the text "The customer has been added successfully". The "Name :" field contains "seema", the "ID:" field contains "94 41 f6 2c", and the "Balance :" field contains "100".

Figure 5.1.5: add new customer GUI

After adding the card to the system, its owner was able to enter the store as shown in the image below



But if the admin wants to update the card values that are already stored in the database , he inserts the card and then the values appear on the page as shown in the image below



please enter the card

Name : Lama

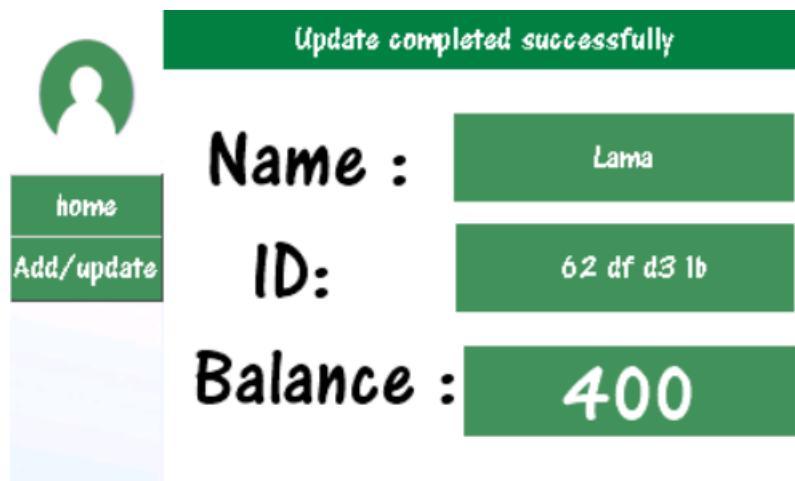
ID: 62 df d3 1b

Balance : 316

home

Add/update

Here in the image below, the balance has been modified



Update completed successfully

Name : Lama

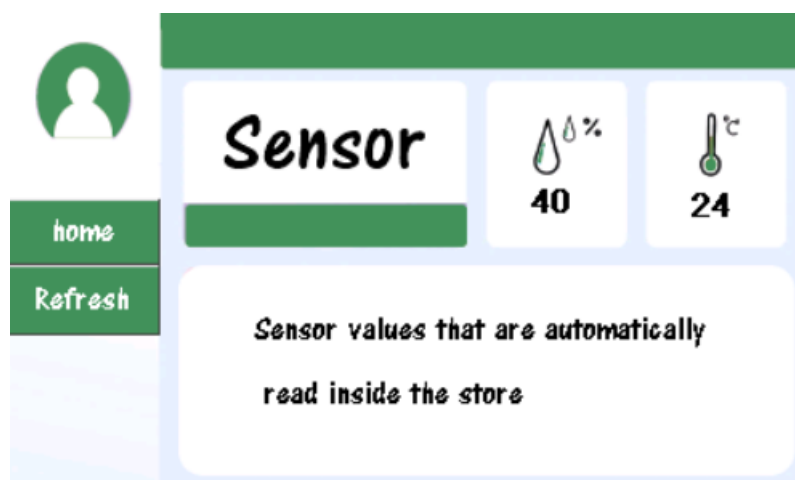
ID: 62 df d3 1b

Balance : 400

home

Add/update

The admin can also know the temperature and humidity inside the machine that is being read from the sensor and shown on this page shown in the image below and by pressing the Refresh button, the values are updated if the sensor reading changes



Sensor

40%

24°C

home

Refresh

Sensor values that are automatically read inside the store

Figure 5.1.6: view sensor value GUI

In the figure 5.1.4 when you click on one of the buttons that show the types of available seedlings , it goes to the page shown in the image below, and then displays the information about it , and through it can modify the price or the existing quantity through the admin only.



You can also add a new type of seedling through the add button on the home page of admin, and then it is stored in the database and added to the seedlings available in the store as shown in the figure 5.1.7

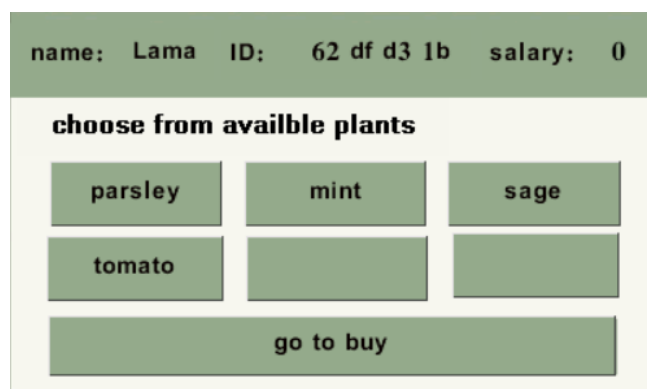
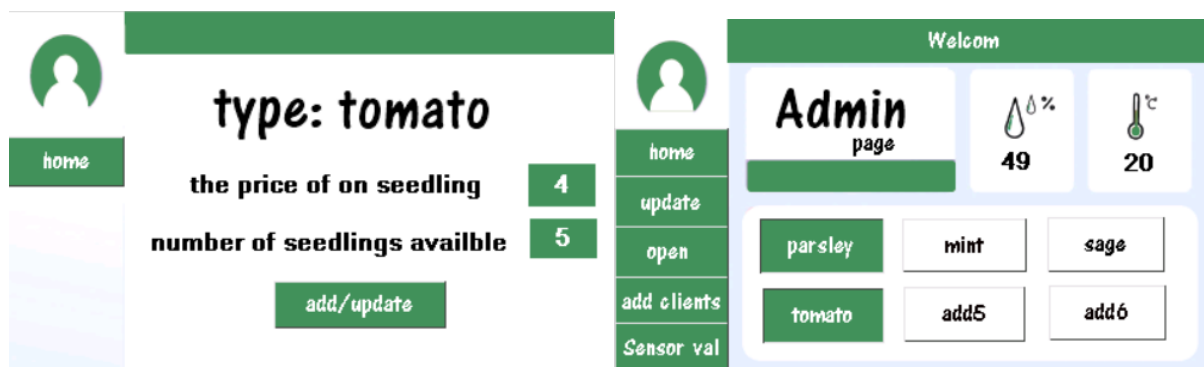
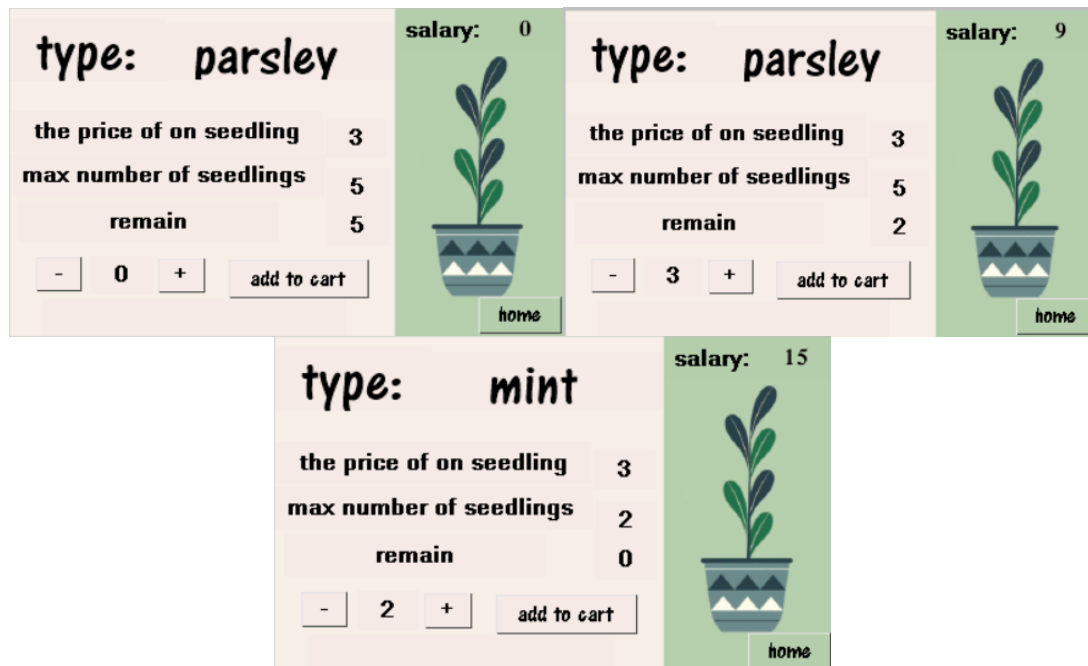


Figure 5.1.7: store after adding new plant GUI

When entering the store page and clicking on one of the seedlings buttons, for example, parsley, its information appears to him, and the number to be purchased is added through the click of a (add to cart) button, then it calculates the total price and shows it at the top of the page as shown in the pictures below.



In figure 5.1.4 when you press the button (go to buy) , you will go to the final page to display the items that have been selected with information about them in terms of the price of one item, the remaining quantity and the selected quantity, in addition to the final price as shown in the picture below.

type	selected	edit		remain	price
parsley	3	-	+	2	3
mint	2	-	+	0	3
	0	-	+	0	0
	0	-	+	0	0
	0	-	+	0	0
	0	-	+	0	0
total salary : 15		remain:	385	buy	back

In the event that the customer decides to buy another variety after moving to the final page, he can press the back button and add the item he wants, for example, a tomato then go to the final page as shown in the pictures below.

type: tomato

the price of on seedling 4

max number of seedlings 5

remain 2


-

3

+

add to cart

salary: 27



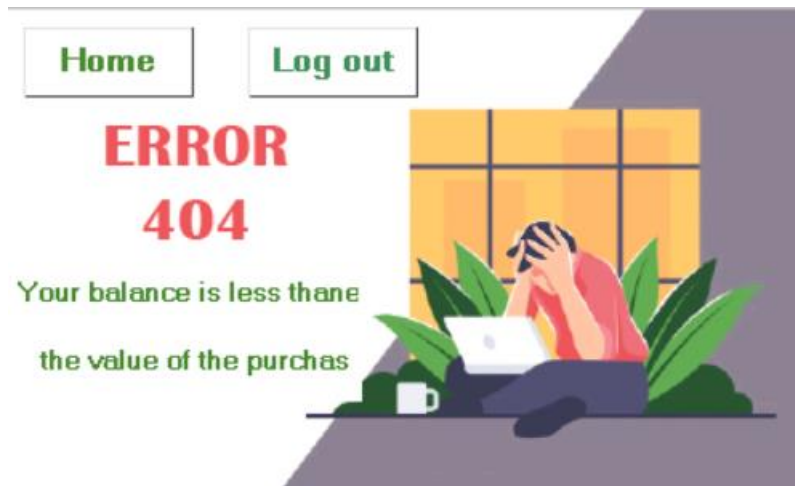
home

type	selected	edit		remain	price
parsley	3	-	+	2	3
mint	2	-	+	0	3
	0	-	+	0	0
tomato	3	-	+	2	4
	0	-	+	0	0
	0	-	+	0	0
total salary : 27		remain: 358		buy	back

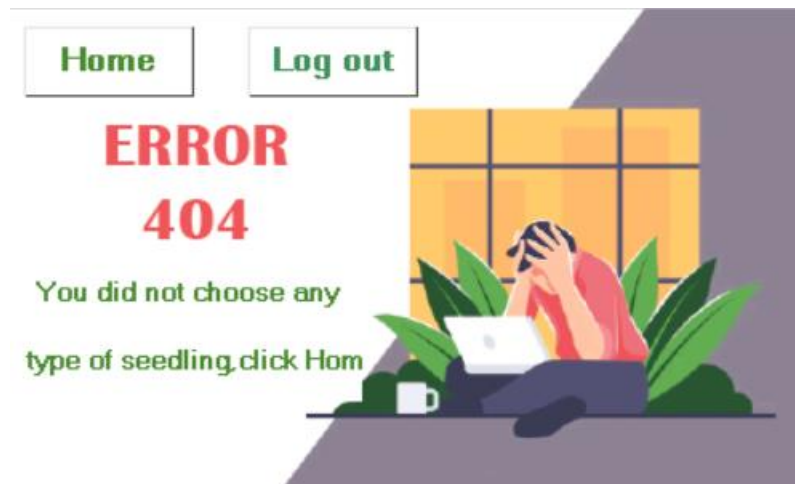
If the total amount purchased is less than the amount available to the customer, the purchase process will be completed successfully and they will be taken to the last page and warned them not to click the "logout" button as the purchase will not be completed successfully as shown in the picture below.



But if it is greater than it, it goes to the page shown below, and the purchase is not completed.



But if no category is selected, go to the page below



5.2 Green House Arduino

➤ Read Card by RFID

GreenStore | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help



//reading card by RFID

```
if (mfrc522.PICC_IsNewCardPresent() ){
  if (mfrc522.PICC_ReadCardSerial()) {

    content = " ";
    for (byte i = 0; i < mfrc522.uid.size; i++) {

      content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
      content.concat(String(mfrc522.uid.uidByte[i], HEX));

    }
    // set information about customer when he entered
    if ( !content.compareTo(id1) || (!content.compareTo(id2))|| (!content.compareTo(id4))|| (!content.compareTo(id5)))
    {
      //this flag is set here to enter the system one time and when another card set to read by rfid ,
      //system can't do anything because he already entered but it allow another when the
      //the customer who entered logout the system -> flagstart set true otherwise flagstart=false
      if(flagstart){
        Serial.print("page 2");
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);
        flagstart=false;
      }
      Serial.print("id.txt=");
      Serial.print("\n");
      Serial.print(content);
      Serial.print("\n");
      Serial.write(0xff);
      Serial.write(0xff);
      Serial.write(0xff);
    }

    // This sentence is executed when a card that is not stored in the database is inserted
    else {
      Serial.print("t6.txt=");
      Serial.print("\n");
      Serial.print("your card is not accepted ");
      Serial.print("\n");
      Serial.write(0xff);
      Serial.write(0xff);
      Serial.write(0xff);

      //when this new card entered in admin page to adding it
      Serial.print("id.txt=");
      Serial.print("\n");
      Serial.print(content);
      Serial.print("\n");
      Serial.write(0xff);
      Serial.write(0xff);
      Serial.write(0xff);

      Serial.print("name.txt=");
      Serial.print("\n");
      Serial.print("");
      Serial.print("\n");
      Serial.write(0xff);
      Serial.write(0xff);
      Serial.write(0xff);
    }
  }
}
```



```
Serial.print("name.txt=");  
Serial.print("\n");  
Serial.print("");  
Serial.print("\n");  
Serial.write(0xff);  
Serial.write(0xff);  
Serial.write(0xff);  
  
Serial.print("n0.val=");  
Serial.print(0);  
Serial.write(0xff);  
Serial.write(0xff);  
Serial.write(0xff);  
}
```

- Show information about each customer by **showInformation** function



```
//show information about each customer

    if (!content.compareTo(id1)) {
        balance1 = readStringFromEEPROM(0).toInt();
        salary = balance1;
        name1=readStringFromEEPROM(220);
        showInformation(name1 ,balance1 );

    }
    else if (!content.compareTo(id2)) {
        balance2 = readStringFromEEPROM(4).toInt();
        name2=readStringFromEEPROM(260);
        salary = balance2;
        showInformation(name2 ,balance2 );

    }
    else if (!content.compareTo(id4)) {
        balance4 = readStringFromEEPROM(380).toInt();
        salary = balance4;
        name4=readStringFromEEPROM(340);
        showInformation(name4 ,balance4 );

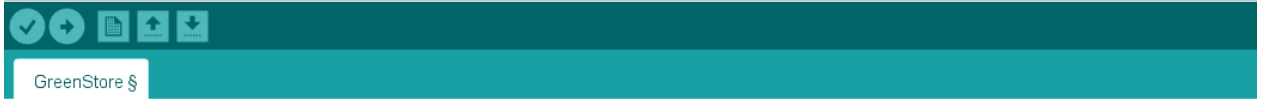
    }
    else if (!content.compareTo(id5)) {
        balance5 = readStringFromEEPROM(400).toInt();
        salary = balance5;
        name5=readStringFromEEPROM(340);
        showInformation(name5 ,balance5 );

    }
}
```



```
void showInformation(String Name ,int Balance){  
    Serial.print("name.txt=");  
    Serial.print("\n");  
    Serial.print(Name);  
    Serial.print("\n");  
    Serial.write(0xff);  
    Serial.write(0xff);  
    Serial.write(0xff);  
  
    Serial.print("\n0.val=");  
    Serial.print(Balance);  
    Serial.write(0xff);  
    Serial.write(0xff);  
    Serial.write(0xff);  
}
```

➤ Read temperature & humidity sensor



```

}
}

////////////////////////////////////////
//this part about humidity and temperture senor
DHT.readll(dht_pin);
humidty = readStringFromEEPROM(130).toInt(); // humidtyv alue that setting by admin
temp = readStringFromEEPROM(150).toInt(); // temperture value that setting by admin

//DHT.humidity ->humidty value read by sensor
// DHT.temperature -> temperature value read by sensor

if ((int (DHT.humidity) < humidty)&&(int (DHT.temperature) > temp)) {
    digitalWrite( humditypin, HIGH);
    digitalWrite(fanpin,HIGH);
    digitalWrite(tengston,LOW);
}
else if ((int (DHT.humidity) == humidty)|| (int (DHT.temperature) == temp)) {
    digitalWrite( humditypin, LOW);
    digitalWrite(fanpin, LOW);
    digitalWrite(tengston,LOW);
}

else {
    digitalWrite( humditypin, LOW);
    digitalWrite(fanpin, LOW);
    digitalWrite(tengston, HIGH);
}

```

➤ Read flame sensor



GreenStore \$

```
//this part for flame sensor when it detecte fire then gsm will callling the number
const int sensorMin = 0;
const int sensorMax = 1024;

int sensorReading = analogRead(A2);
int range = map(sensorReading, sensorMin, sensorMax, 0, 3);

// range value:
switch (range) {
case 0:

digitalWrite(gsm, HIGH);
    break;
case 1:

    digitalWrite(gsm, LOW);
    break;
case 2:
    // Serial.println("***no fire***");
    digitalWrite(gsm, LOW);
    break;
}
delay(1000); // delay between reads
```

➤ Read LDR sensor

```

GreenStore $

// this part for LDR sensor
int sensorValue = analogRead(A1);
// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
float voltage = sensorValue * (5.0 / 1024.0);

if ((voltage >= 0) && (voltage <= 0.4)) {
    // dark this means that it is night
    digitalWrite(LED, HIGH);
    delay(500); //Wait .5 seconds before accessing sensor again.
}
else if ((voltage > 0.4) && (voltage <= 2)) {
    // dark this means that it is light
    digitalWrite(LED, LOW);
    delay(500); //Wait .5 seconds before accessing sensor again.
}
else if (voltage >= 3) {

    digitalWrite(LED, HIGH);

    //delay(500); //Wait .5 seconds before accessing sensor again.
}

```

➤ Read ultrasonic sensor

To maintain safety, we have created a protection system so that if one of the thieves opens the door without the purchase process, a voice call is made with the admin to warn of the presence of a theft

```

GreenStore $

////////////////////////////////////
if(alarmsensor){
    int distanceToAlarm;
    distanceToAlarm=calculateDistance();
    if(distanceToAlarm>=1 && distanceToAlarm<=30){
        digitalWrite(30,HIGH);
    }
    else{
        digitalWrite(30,LOW);
    }
}

```

➤ Show information about each seedling



```
// to showing information about seedling number4
    else if ( int(rec[0]) == 0) {

        flagupdate4 = true;

        type = readStringFromEEPROM(120);
        max4 = readStringFromEEPROM(60).toInt();
        price4 = readStringFromEEPROM(65).toInt();
        rem4 = readStringFromEEPROM(190).toInt();
        ShowInformationSeedlings (type,max4,price4,rem4 );

    }
// to showing information about seedling number5
    else if ( int(rec[0]) == 1) {

        flagupdate5 = true;
        type = readStringFromEEPROM(70);
        max5 = readStringFromEEPROM(85).toInt();
        price5 = readStringFromEEPROM(90).toInt();
        rem5 = readStringFromEEPROM(200).toInt();
        ShowInformationSeedlings (type,max5,price5,rem5 );

    }
}
```



GreenStore \$

```
// to showing information about seedling number6
else if ( int(rec[0]) == 2) {
    flagupdate6 = true;

    type = readStringFromEEPROM(95);
    max6 = readStringFromEEPROM(110).toInt();
    price6 = readStringFromEEPROM(115).toInt();
    rem6 = readStringFromEEPROM(210).toInt();
    ShowInformationSeedlings(type,max6,price6,rem6);
}
// to showing information about seedling (mint)
else if ( int(rec[0]) == 8) {

    flagupdatemint = true;
    type = readStringFromEEPROM(8);
    maxmint = readStringFromEEPROM(13).toInt();
    pricemint = readStringFromEEPROM(16).toInt();
    remmint = readStringFromEEPROM(170).toInt();
    ShowInformationSeedlings(type,maxmint,pricemint,remmint);
}
// to showing information about seedling (parsley)
else if ( int(rec[0]) == 9) {

    flagupdatepars = true;
    type = readStringFromEEPROM(32);
    maxparsley = readStringFromEEPROM(40).toInt();
    priceparsley = readStringFromEEPROM(44).toInt();
    remparsley = readStringFromEEPROM(160).toInt();
    ShowInformationSeedlings(type,maxparsley,priceparsley,remparsley);
}

// to showing information about seedling (sage)
else if ( int(rec[0]) == 10) {
    flagupdatesage = true;
    type = readStringFromEEPROM(20);
    maxsage = readStringFromEEPROM(25).toInt();
    pricesage = readStringFromEEPROM(28).toInt();
    remsage = readStringFromEEPROM(180).toInt();
    ShowInformationSeedlings(type,maxsage,pricesage,remage);
}
```


➤ Show information seedlings function

GreenStore | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help



GreenStore \$

```
void ShowInformationSeedlings(String Type ,int maxaddress,int priceaddress ,int remaddress ){
    Serial.print("page4.t4.txt=");
    Serial.print("\n");
    Serial.print(Type);
    Serial.print("\n");
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);

    Serial.print("page4.max.val=");
    Serial.print(maxaddress);
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);

    Serial.print("page4.n0.val=");
    Serial.print(priceaddress);
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);

    Serial.print("page4.rem.val=");
    Serial.print(remaddress);
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);
}
```

➤ When click buy button

GreenStore | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help



GreenStore \$

```
else if (int(rec[0]) == 4) { //when click buy button
  //read the salary
  alarmsensor=false;
  finalsal = Mynex.readNumber("page7.salary2.val");
  parsleyfinal=Mynex.readNumber("page7.nparsley.val");
  mintfinal=Mynex.readNumber("page7.nmint.val");
  sagefinal=Mynex.readNumber("page7.nsage.val");
  type4final=Mynex.readNumber("page7.n4.val");
  type5final=Mynex.readNumber("page7.n5.val");
  type5final=Mynex.readNumber("page7.n6.val");

  if (flag1 == true) {
    remmint = Mynex.readNumber("page7.remm.val");
    writeStringToEEPROM(13, String( remmint));
    writeStringToEEPROM(170, String(remmint));
    flag1 = false;
  }
  if (flag2 == true) {
    remparsley = Mynex.readNumber("page7.remp.val");
    writeStringToEEPROM(40, String(remparsley));
    writeStringToEEPROM(160, String(remparsley));
    flag2 = false;
  }
  if (flag3 == true) {
    remsage = Mynex.readNumber("page7.rems.val");
    writeStringToEEPROM(25, String( remsage));
    writeStringToEEPROM(180, String(remsage));
    flag3 = false;
  }
}
```

```

    if (flag5 == true) {
        rem5 = Mynex.readNumber("page7.rem5.val");
        writeStringToEEPROM(85, String(rem5));
        writeStringToEEPROM(200, String(rem5));
        flag5 = false;
    }
    if (flag6 == true) {
        rem6 = Mynex.readNumber("page7.rem6.val");
        writeStringToEEPROM(110, String(rem6));
        writeStringToEEPROM(210, String(rem6));
        flag6 = false;
    }

    if (!content.compareTo(id1)) {

        //can buy
        balance1 = balance1 - finalsal;
        writeStringToEEPROM(0, String(balance1));

    }
    else if (!content.compareTo(id2)) {

        balance2 = balance2 - finalsal;
        writeStringToEEPROM(4, String(balance2));
    }

    else if (!content.compareTo(id4)) {

        balance4 = balance4 - finalsal;
        writeStringToEEPROM(380, String(balance4));
    }

    else if (!content.compareTo(id5)) {

        balance5 = balance5 - finalsal;
        writeStringToEEPROM(400, String(balance5));
    }

}
// trun off each fan,fogger and tungeston when buy
digitalWrite(humditypin,LOW);
digitalWrite(fanpin,LOW);
digitalWrite(tengston,LOW);

if( mintfinal!=0){
    for(int i=0;i<mintfinal;i++){
        StartBuy(Am1 ,Am2);
    }
    delay(5000);
    digitalWrite(motor,LOW);
}

```



GreenStore \$

```
if( parsleyfinal!=0){

    for(int i=0;i<parsleyfinal;i++){
        StartBuy(Bm1 ,B2);
    }
    delay(5000);
    digitalWrite(motor,LOW);
}

if( sagefinal!=0){
    for(int i=0;i<sagefinal;i++){
        StartBuy(C1 ,C2);
    }
    delay(5000);
    digitalWrite(motor,LOW);
}

myservo.write(0);
alarmsensor=true;
}
```

➤ StartBuy function



GreenStore \$

```
void StartBuy(const int solnoid1 ,const int solnoid2){
    myservo.write(90);
    delay(1000);
    digitalWrite(motor,HIGH);
    digitalWrite(solnoid2,HIGH);
    delay(1000);
    digitalWrite(solnoid2,LOW);
    delay(2900);
    digitalWrite(motor,LOW);

    while (true){
        distance= calculateDistance();
        if(distance>=1 && distance<=30){
            digitalWrite(motor,LOW);
        }
        else
            break;
    }
    delay(50);
    digitalWrite(solnoid1,HIGH);
    delay(230);
    digitalWrite(solnoid1,LOW);
}
```

➤ **Show information in admin page**



```
// show information in admin page

else if ( int(rec[0]) == 14) {

    type1 = readStringFromEEPROM(32);
    maxparsley = readStringFromEEPROM(40).toInt();
    priceparsley = readStringFromEEPROM(44).toInt();
    type = Mynex.readStr("page11.t4.txt");
    ShowSeedlingInfoAdmin(type1 ,maxparsley ,priceparsley);

}
else if ( int(rec[0]) == 15) {

    type2 = readStringFromEEPROM(8);
    pricemint = readStringFromEEPROM(16).toInt();
    type = Mynex.readStr("page11.t4.txt");
    maxmint = readStringFromEEPROM(13).toInt();
    ShowSeedlingInfoAdmin(type2 ,maxmint ,pricemint);

}

else if ( int(rec[0]) == 16) {

    type3 = readStringFromEEPROM(20);
    maxsage = readStringFromEEPROM(25).toInt();
    pricesage = readStringFromEEPROM(28).toInt();
    type = Mynex.readStr("page11.t4.txt");
    ShowSeedlingInfoAdmin(type3 ,maxsage ,pricesage);

}
```

```

else if ( int(rec[0]) == 17) {

    type4 = readStringFromEEPROM(120);
    max4 = readStringFromEEPROM(60).toInt();
    price4 = readStringFromEEPROM(65).toInt();
    type = Mynex.readString("page11.t4.txt");
    ShowSeedlingInfoAdmin(type4 ,max4 ,price4);

}
else if ( int(rec[0]) == 18) {

    type5 = readStringFromEEPROM(70);
    max5 = readStringFromEEPROM(85).toInt();
    price5 = readStringFromEEPROM(90).toInt();
    type = Mynex.readString("page11.t4.txt");
    ShowSeedlingInfoAdmin(type5 ,max5 ,price5);

}

else if ( int(rec[0]) == 19) {

    type6 = readStringFromEEPROM(95);
    max6 = readStringFromEEPROM(110).toInt();
    price6 = readStringFromEEPROM(115).toInt();
    type = Mynex.readString("page11.t4.txt");
    ShowSeedlingInfoAdmin(type6 ,max6 ,price6);

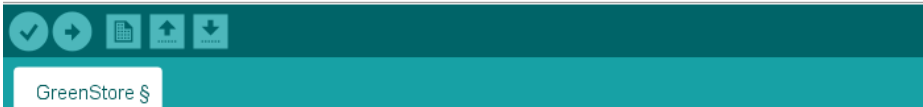
}

```

➤ ShowSeedlingInfoAdmin function

 GreenStore | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help



```

void ShowSeedlingInfoAdmin(String msg ,int maxval , int priceval){
    Serial.print("page11.t4.txt=");
    Serial.print("\n");
    Serial.print(msg);
    Serial.print("\n");
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);

    Serial.print("page11.max.val=");
    Serial.print(maxval);
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);

    Serial.print("page11.n0.val=");
    Serial.print(priceval);
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);
}

```

➤ Update or add seedling



GreenStore \$

```
// when click any button in admin page to get type of seedlings#
// when click in update button in admin page

else if ( int(rec[0]) == 5) {

    if (type == type1) {
        flagupdatepars = false;
        type = Mynex.readString("page11.t4.txt");
        //when add new seedling that already exist
        if (type == type2 || type == type3 || type == type4 || type == type5 || type == type6)
            AddSeedlingExist();

        else {

            priceparsley = Mynex.readNumber("page11.n0.val");
            maxparsley = Mynex.readNumber("page11.max.val");
            writeStringToEEPROM(44, String(priceparsley));
            writeStringToEEPROM(40, String(maxparsley));
            writeStringToEEPROM(32, type);
            String s0="page3.b0.txt=";
            String s1="page10.b1.txt=";
            AddSeedlingNotExistOrUpdate( type,s0,s1);

        }
    }
}
```

➤ AddSeedlingNotExistOrUpdate function



```
void AddSeedlingNotExistOrUpdate(String typeofseedling,String s14,String s15){
    Serial.print(s14);
    Serial.print("\n");
    Serial.print(typeofseedling);
    Serial.print("\n");
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);

    Serial.print(s15);
    Serial.print("\n");
    Serial.print(typeofseedling);
    Serial.print("\n");
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);

    Serial.print("page 8");
    Serial.write(0xff);
    Serial.write(0xff);
    Serial.write(0xff);
}
```

➤ AddSeedlingNoExistOrUpdate function



GreenStore \$

}

```
void AddSeedlingExist() {
```

```
    Serial.print("page11.t0.txt=");  
    Serial.print("\n");  
    Serial.print("ERROR !! this type is already exist");  
    Serial.print("\n");  
    Serial.write(0xff);  
    Serial.write(0xff);  
    Serial.write(0xff);
```

```
    Serial.print("page 10");  
    Serial.write(0xff);  
    Serial.write(0xff);  
    Serial.write(0xff);
```

}

➤ Show all information in admin page



GreenStore \$

```
// to print type for each button in admin page . humiditet and tempreature

else if ( int(rec[0]) == 7) {
  humidity = readStringFromEEPROM(130).toInt();
  Serial.print("page10.hu.val=");
  Serial.print(humidity);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);

  temp = readStringFromEEPROM(150).toInt();
  Serial.print("page10.temp.val=");
  Serial.print(temp);
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);

  type1 = readStringFromEEPROM(32);
  Serial.print("page10.b1.txt=");
  Serial.print("\n");
  Serial.print(type1);
  Serial.print("\n");
  Serial.write(0xff);
  Serial.write(0xff);
  Serial.write(0xff);

  type2 = readStringFromEEPROM(8);
  Serial.print("page10.b2.txt=");
  Serial.print("\n");
  Serial.print(type2);
  Serial.print("\n");
  Serial.write(0xff);
}
```

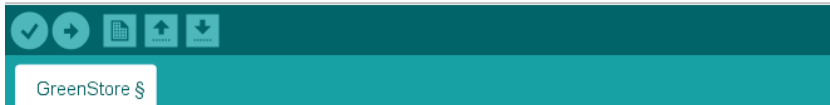
```
Serial.write(0xff);
Serial.write(0xff);

type3 = readStringFromEEPROM(20);
Serial.print("page10.b3.txt=");
Serial.print("\n");
Serial.print(type3);
Serial.print("\n");
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);

type4 = readStringFromEEPROM(120);
Serial.print("page10.b4.txt=");
Serial.print("\n");
Serial.print(type4);
Serial.print("\n");
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);

type5 = readStringFromEEPROM(70);
Serial.print("page10.b5.txt=");
Serial.print("\n");
Serial.print(type5);
Serial.print("\n");
Serial.write(0xff);
Serial.write(0xff);
Serial.write(0xff);
```

➤ When click to store button for show information



```

        type6 = readStringFromEEPROM(95);
        Serial.print("page10.b6.txt=");
        Serial.print("\n");
        Serial.print(type6);
        Serial.print("\n");
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);

    }

    else if ( int(rec[0]) == 24) {

        type1 = readStringFromEEPROM(32);
        Serial.print("page3.b0.txt=");
        Serial.print("\n");
        Serial.print(type1);
        Serial.print("\n");
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);

        type2 = readStringFromEEPROM(8);
        Serial.print("page3.b1.txt=");
        Serial.print("\n");
        Serial.print(type2);
        Serial.print("\n");
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);

        type3 = readStringFromEEPROM(20);
        Serial.print("page3.b2.txt=");
        Serial.print("\n");
        Serial.print(type3);
        Serial.print("\n");
        Serial.write(0xff);
        Serial.write(0xff);
        Serial.write(0xff);

        type4 = readStringFromEEPROM(120);
        if (type4 == "add4") {
            Serial.print("page3.b3.txt=");
            Serial.print("\n");
            Serial.print("");
            Serial.print("\n");
            Serial.write(0xff);
            Serial.write(0xff);
            Serial.write(0xff);
        }
        else {

            Serial.print("page3.b3.txt=");
            Serial.print("\n");
            Serial.print(type4);
            Serial.print("\n");
            Serial.write(0xff);
            Serial.write(0xff);
            Serial.write(0xff);
        }
    }

```

➤ **When click update button in admin page to update temperature & humidity**

GreenStore | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

GreenStore \$

```
}  
  
else if ( int(rec[0]) == 25) {  
  
    humidity = Mynex.readNumber("page10.hu.val");  
  
    writeStringToEEPROM(130, String(humidity));  
  
    temp = Mynex.readNumber("page10.temp.val");  
  
    writeStringToEEPROM(150, String(temp));  
}
```

➤ **When customer click buy but something wrong happened**



```
else if ( int(rec[0]) == 26) {  
    String remmmint = readStringFromEEPROM(13);  
    writeStringToEEPROM(170, String( remmmint));  
  
    String remparsley = readStringFromEEPROM(40);  
    writeStringToEEPROM(160, String( remparsley));  
  
    String remsage = readStringFromEEPROM(25);  
    writeStringToEEPROM(180, String( remsage));  
  
    String rem4 = readStringFromEEPROM(60);  
    writeStringToEEPROM(190, String( rem4));  
  
    String rem5 = readStringFromEEPROM(85);  
    writeStringToEEPROM(200, String( rem5));  
  
    String rem6 = readStringFromEEPROM(110);  
    writeStringToEEPROM(210, String( rem6));  
}
```

➤ When update or add customer



GreenStore \$

```

else if ( int(rec[0]) == 29) {
    String addID= Mynex.readStr("id.txt");
    String addName= Mynex.readStr("name.txt");
    int addBalance= Mynex.readNumber("n0.val");
    id4=readStringFromEEPROM(420);
    id5=readStringFromEEPROM(440);

    if(!addID.compareTo(id1)) {
        writeStringToEEPROM(220, String( addName));
        updateIDText();
    }
    else if(!addID.compareTo(id2)) {
        writeStringToEEPROM(260, String( addName));
        writeStringToEEPROM(4, String( addBalance));
        updateIDText();
    }

    else{

        if((id4=="")){
            writeStringToEEPROM(300, String( addName));
            writeStringToEEPROM(420, String( addID));
            writeStringToEEPROM(380, String( addBalance));
            updateCustomerText();
        }
        else{
            writeStringToEEPROM(300, String( addName));
            writeStringToEEPROM(380, String( addBalance));
            updateIDText();
        }
    }

    if(id5==""){
        writeStringToEEPROM(340, String( addName));
        writeStringToEEPROM(440, String( addID));
        writeStringToEEPROM(400, String( addBalance));
        updateCustomerText();
    }
    else{
        writeStringToEEPROM(340, String( addName));
        writeStringToEEPROM(400, String( addBalance));
        updateIDText();
    }
}
}

```


➤ **When admin click open button in his page**

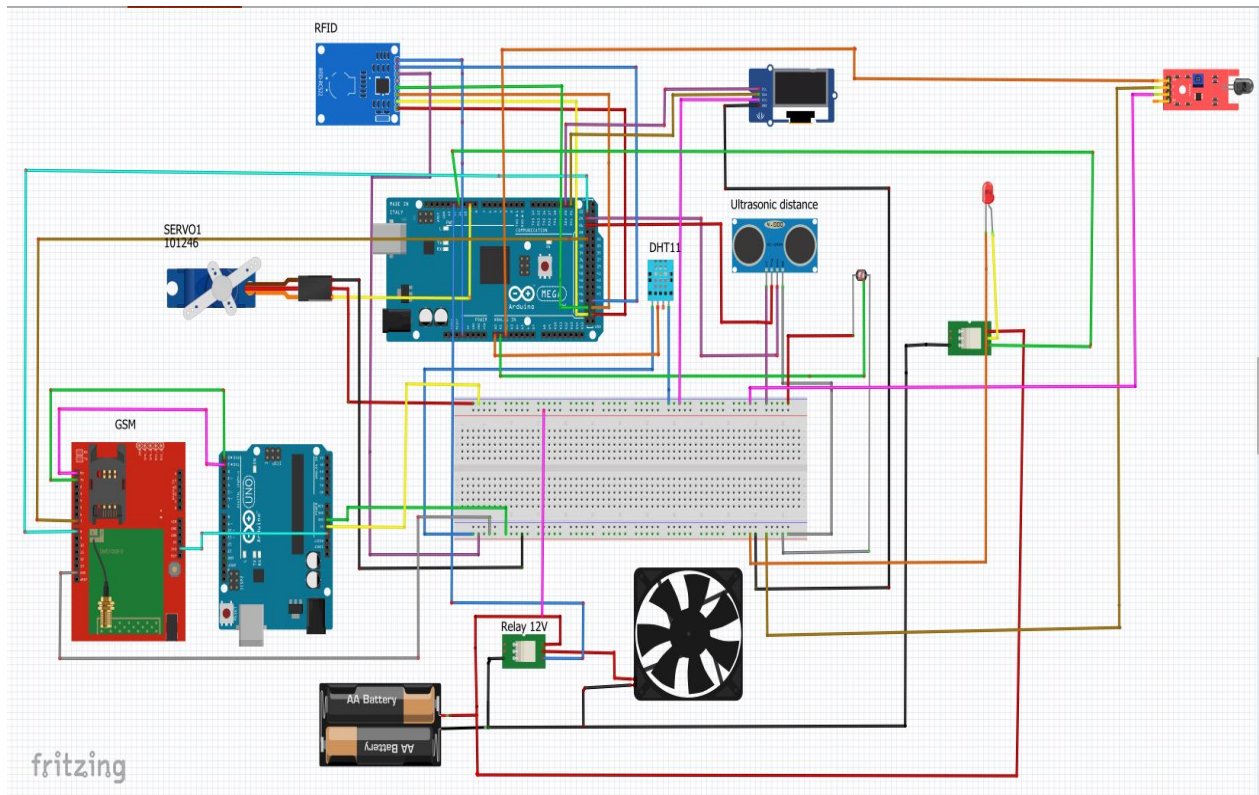
```
else if ( int(rec[0]) == 30) {  
    myservo.write(90);  
    delay(60000);  
    myservo.write(0);  
}
```

➤ **When admin click sensor Val button in his admin**



```
else if ( int(rec[0]) == 31) {  
  
    Serial.print("page13.humidity.val=");  
    Serial.print(int(DHT.humidity));  
    Serial.write(0xff);  
    Serial.write(0xff);  
    Serial.write(0xff);  
  
    Serial.print("page13.temprture.val=");  
    Serial.print(int(DHT.temperature));  
    Serial.write(0xff);  
    Serial.write(0xff);  
    Serial.write(0xff);  
  
}
```

➤ 5.3 Circuit Layout



Chapter 6

Discussion

This project achieved its objectives in that it is a direct electronic sale process available anywhere in the malls, and at any time so that this sale process does not require the presence of an employee.

Some types of seedlings are not at hand and are found in nurseries very far from the city and in remote areas, so those seedlings or plants can be inside a greenhouse machine, so that often these seedling vending machines are not available, they can be used in plants Fuel near rural areas or in large nurseries, they reduce the work of employees.

Also, some seedlings suffer from the problem of extinction as a result of hot and cold weather, as well as changes in the wind level, so that the machine provides them with positive environmental conditions in terms of humidity and temperature, so that when the person adjusts the specific temperature and humidity according to the type of seedling inside the green house. These conditions are necessary for its growth and maintenance, and from here we can reduce the idea of the extinction of some species of seedlings and the preservation of certain species.

Chapter 7

Conclusions and recommendations

7.1 Conclusion

Shopping without walking a few miles is made possible thanks to technology. Over the past few years, e-commerce has reached a whole new level and it is constantly growing. Undoubtedly, if technology had not advanced, the easy online shopping experience you see today would not have existed.

Thanks to a variety of advanced electronics, hardware, and software, remote businesses have been able to reach global customers and increase their sales. Likewise, shoppers benefit by saving the time and energy they waste when visiting physical stores.

In addition, technology has made it simple and safe, which has encouraged more companies and customers to use this technology and machines.

With technology, the process of remote control of warehouses and machines without the need for a supervisor has become a simple and easy process.

7.2 Recommendations

And in the end we found that these machines must be located in remote areas, so in the future can use the automatic mowing process and automatic watering process, and the process of identifying damaged seedlings so that if the seedlings are damaged, a message can be sent to the official store or can be banned from the store so that the customer from the purchase.

For example, if it is possible to add manure to the damaged seedlings automatically and after adding the manure, it can be re-checked if it is damaged, the ban should be sent to the store manager.

It is possible in the future to use the idea of a QR code so that this code is sent to the website of the online store, and the store is a website instead of using a touch screen.

Or it is possible to add the idea of digital processing of images so that if one of the seedlings is damaged, the type of virus is analyzed through the data..

References

- [1] <https://rarehistoricalphotos.com/vending-machines-vintage-pictures/>
- [2] <https://www.atlasobscura.com/articles/the-history-of-vending-machines-goes-back-to-the-1st-century>
- [3] <http://smartplantfactory.asia/3-plant-vending-machine.html>
- [4] <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>
- [5] <https://www.allaboutcircuits.com/technical-articles/buck-converters-and-their-cool-applications/>
- [6] <https://www.datasheetarchive.com/relay%2012VDC%20datasheet%20omron-datasheet.html>
- [7] <https://components101.com/sensors/dht11-temperature-sensor>
- [8] <https://www.geeksforgeeks.org/gsm-in-wireless-communication/>
- [9] <https://store.arduino.cc/products/arduino-mega-2560-rev3>
- [10] <https://nextion.tech/instruction-set/>
- [11] <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>

