



CS330–Computer Networks Project

Client/Server Application

Students Names:

Lubna Alrabiah – Id: _____

Reem Alothman – Id: _____

Aljowharah Aldawood – _____

Teef Alosan – _____

Section: 371

Submission Date: 19/5/2022

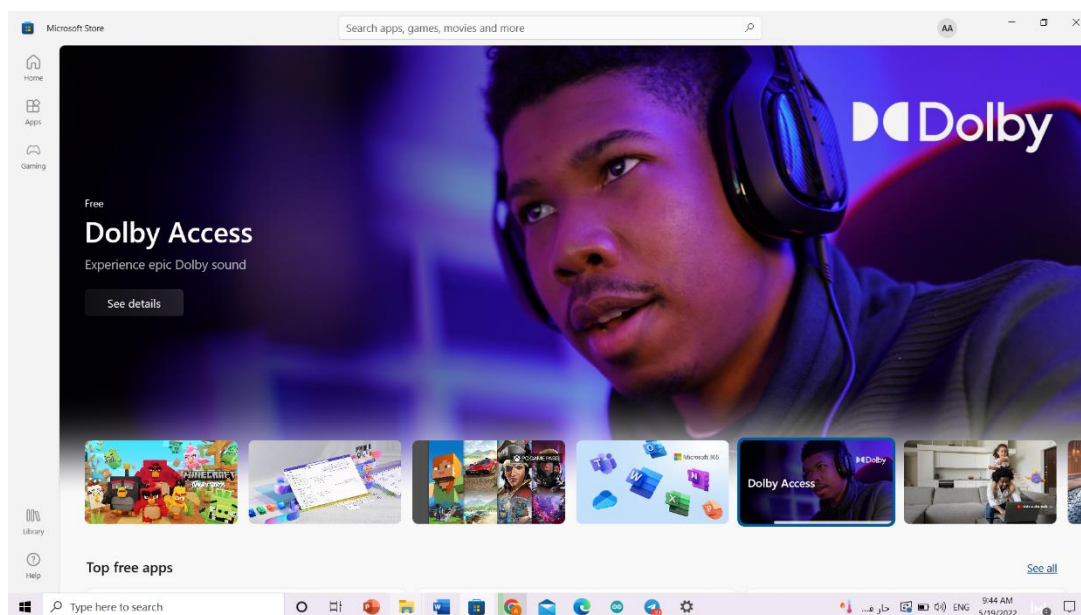
1- Setting up the Programming Environment

In this section of the report explain your choice of programming language and why you chose it?

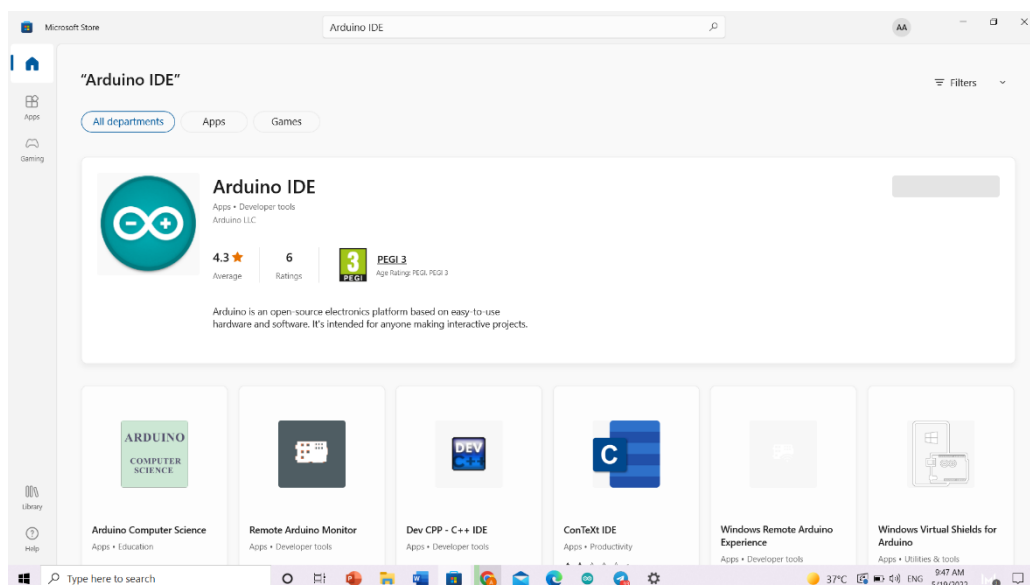
The programming language is Arduino C, since the language is open source and it can be modified and added to suit the project, which made it the best choice to use.

Provide screenshots of the programming environment and a detailed description of how you installed the environment ?

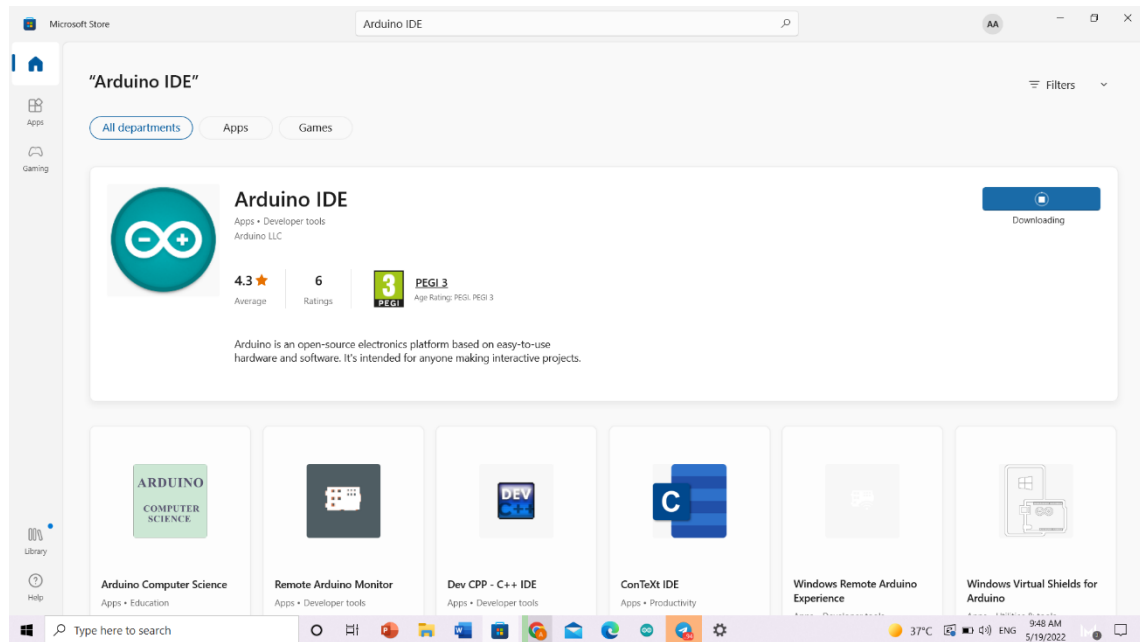
1- Open Microsoft Store.



2- Search Arduino IDE.



3- Install Arduino IDE.



and how you compile and run a program

1- Startup the Arduino IDE.



2- Write the code.

```
sketch_may16a | Arduino 1.8.16
File Edit Sketch Tools Help

sketch_may16a
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

Servo myservo;

#define ir_enter 2
#define ir_back 4

#define ir_car1 5
#define ir_car2 6

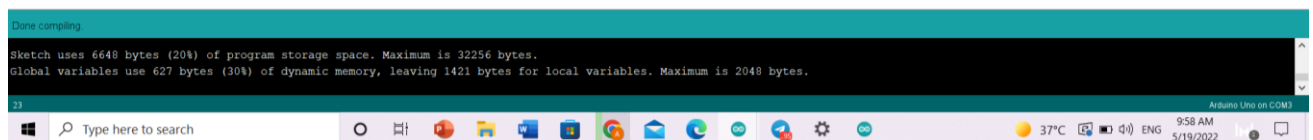
int s1 = 0, s2 = 0;
int flag1 = 0, flag2 = 0;
int slot = 2;
int redPin = 8;
int greenPin = 7;

void setup() {
  Serial.begin(9600);

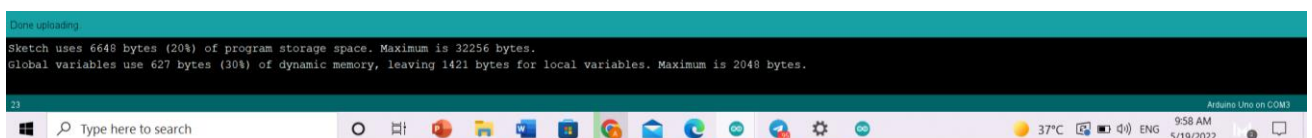
  pinMode(ir_car1, INPUT);
  pinMode(ir_car2, INPUT);

  pinMode(ir_enter, INPUT);
  pinMode(ir_back, INPUT);
}
```

3- Verify: this help to verify the code that has been written and check for any errors..



4- Upload: Uploading the code to the Arduino device (Arduino UNO).



Don't forget to describe any libraries you needed to install and why you needed to install them

- 1- **LiquidCrystal_I2C:** for coding, editing and displaying the LCD screen.
- 2- **NewliquidCrystal:** for coding, editing and displaying the LCD screen.
- 3- **Wire:** for simplifying the communication between devices.

2-Steps for TCP/UDP socket programming for client-server connection

We use write the name of programming language socket programming to implement the client- server communication over TCP/UDP protocol [cite the resource that you adapted to write the code, if you do not cite the resource then you are suggesting that you wrote the code from scratch and you risk the grade of your project if the plagiarism tool indicates resemblance to other code].

We use Arduino C language , you can show code here:

1- <https://youtu.be/5Qi93MjlqzE>

2- <https://www.prateeks.in/2021/11/car-parking-system-using-arduino.html?m=1>]

Suggested format (your code may use a different sequence of steps)

Some lines of the code have been changed to reach the required output:

- 1- We change the number of sensors and the number of slots to fit the car parks that were made.
- 2- Typing print sentences have been added to explain how the server works.
- 3- Typing print sentences have been added to show different format of output in the LSD screen.
- 4- Code of the first resource has been merged with the second resource after applying the changes.

CODE:

```
sketch_may19a$
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

Servo myservo;

#define ir_enter 2 // The IR sensor in port 2 that is located at the gate.
#define ir_back 4 // The IR sensor in port 4 that is located at the gate.

#define ir_car1 5 // The IR sensor in port 5 that is located at the first parking slot.
#define ir_car2 6 // The IR sensor in port 6 that is located at the second parking slot.

int S1 = 0, S2 = 0; // Initial values for Slot 1 and Slot 2.
int flag1 = 0, flag2 = 0; // signaling for the wires.
int slot = 2; // Available number of slots. (This project only has 2 slots)
int redPin = 8; // The pin(port) number that is connecting the Arduino UNO to RGB light Red.
int greenPin = 7; // The pin(port) number that is connecting the Arduino UNO to RGB light Green.

void setup() { // Settingup the Arduino UNO.
  Serial.begin(9600); //The beginning point.
  pinMode(ir_car1, INPUT); // Taking the input for the IR sensor of the car slot 1.
  pinMode(ir_car2, INPUT); // Taking the input for the IR sensor of the car slot 2.
  pinMode(ir_enter, INPUT); // Taking the input for the IR sensor of the entering the gate.
  pinMode(ir_back, INPUT); // Taking the input for the IR sensor for exiting the gate.
  myservo.attach(3); // The servo motor that used for the gate.
  myservo.write(90); // The beginning point of the servo motor.

  lcd.init(); // Initializing the LCD Screen.
  lcd.backlight(); // Printing a message to the LCD Screen.
  lcd.setCursor(0, 1); // The LCD screen coordinates.
  lcd.print(" Hi Welcome To "); // The text that will be displayed on the LCD screen.
  lcd.setCursor(0, 2); // The LCD screen coordinates.
  lcd.print(" -"); // the text that will display on LDC screen -
  delay(5000); // Delay time by 5000.
  lcd.clear(); // To clear the LCD screen.
  lcd.setCursor(0, 0); // The screen coordinates.
  lcd.print("NETWORK Project"); // the text that will display on the LCD screen.
  lcd.setCursor(0, 1); // The LCD screen coordinates.
  lcd.print(" Car Parking system"); // The text that will be displayed on the LCD screen.
  lcd.setCursor(0, 2); // The LCD screen coordinates.
  lcd.print(" System "); // The text that will be displayed on the LCD screen.
  delay(5000); // Delay time by 5000.
  lcd.clear(); // To clear the LCD screen.
  pinMode(redPin, OUTPUT); // Taking the output for the RGB light Red.
  pinMode(greenPin, OUTPUT); // Taking the output for the RGB light Green.

  Read_Sensor(); // Reading sensor info.

  int total = S1 + S2; // To calculate the total parking spaces.
  slot = slot - total; // To calculate a single parking space.
}
```

```

void loop() { // Loop for the devices

  Read_Sensor(); // Reading sensor info.

  lcd.setCursor(0, 0); // The LCD screen coordinates.
  lcd.print(" Avail Slot: "); // The text that will be displayed on the LCD screen.
  lcd.print(slot); // Printing number of available slots.

  Serial.println(" Avail Slot: "); // The text that will be displays it on the server laptop screen.
  Serial.println(slot); // Printing number of available slots.
  lcd.print(" "); // The text that will be displayed on the LCD screen.
  Serial.println(" "); // The text that will be displays it on the server laptop screen.

  lcd.setCursor(0, 1); // The LCD screen coordinates.
  if (S1 == 1) { // If statment if the slot 1 was not available.
    lcd.print("S1:Full "); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Full "); // The text that will be displays it on the server laptop screen if the slot is full.
    digitalWrite(greenPin, LOW); // Not show the RGB Green light (set green light as LOW).
    digitalWrite(redPin, HIGH); // Show the RGB Red light (set red light as HIGH).
  }
  else { // Else for the if statement when slot 1 is available.
    lcd.print("S1:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Empty"); // The text that will be displays it on the server laptop screen if the slot is Empty.
    digitalWrite(redPin, LOW); // Not show the RGB Red light (set red light as LOW).
    digitalWrite(greenPin, HIGH); // Show the RGB Green light (set green light as HIGH).
  }

  lcd.setCursor(8, 1); // The LCD screen coordinates.
  if (S2 == 1) { // If statment if the slot 2 was not available.
    lcd.print("S2:Full "); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S2:Full "); // The text that will be displays it on the server laptop screen if the slot is full.
    // there is suppose to be a code here for the RGB lights but the Light has burnedup and does not work.
  }
  else { // Else for the if statement when slot 2 is available.
    lcd.print("S2:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S2:Empty"); // The text that will be displays it on the server laptop screen if the slot is Empty.
    // there is suppose to be a code here for the RGB lights but the Light has burnedup and does not work.
  }

  if (digitalRead(ir_enter) == 0 && flag1 == 0) { //If statment for the gate if the car parking is full.
    if (slot > 0) {
      flag1 = 1;
      if (flag2 == 0) {
        myservo.write(180);
        slot = slot - 1;
      }
    }
    else {
      lcd.setCursor(0, 0); // The LCD screen coordinates.
      lcd.print("Parking Full"); // The text that will be displayed on the LCD screen.
      Serial.println("Parking Full"); // The text that will be displays it on the server laptop screen.
      delay(1500); // Delay by 1500.
    }
  }

  if (digitalRead(ir_back) == 0 && flag2 == 0) { //If statment for the gate if the car parking has available slots.
    flag2 = 1;
    if (flag1 == 0) {
      myservo.write(180);
      slot = slot + 1;
    }
  }

  if (flag1 == 1 && flag2 == 1)
    delay(1000);
  myservo.write(90);
  flag1 = 0, flag2 = 0;
}

delay(1); //Delay by 1.

}

void Read_Sensor() {
  S1 = 0, S2 = 0; //Initiation for both sensors.

  if (digitalRead(ir_car1) == 0) { // if slot one = 1 it means its full
    S1 = 1;
  }
  if (digitalRead(ir_car2) == 0) { // if slot two = 1 it means its full
    S2 = 1;
  }
}

```

1- The server will create the socket using:

For the server here we initialized all LCD display screen. First initialized the screen it self then the backlight which it prints the messages to the LCD screen. As well, the set cursor (0,1) that the LCD coordinates the phrases that will be printed. More forward printing the phrase “ Hi Welcome” to appear on the LCD display Screen. For the delay(5000) it delays the time by 5000. Lcd.clear it justifies it in order to clear the Lcd Display screen. It continues the same steps and methods throw the rest. PinMode job is we have to initialize the input and it takes output for the sensor for this it pinmode the red light and green light in RGB Light. Read sensor reads the sensor information.

```
lcd.init(); // Initializing the LCD Screen.
lcd.backlight(); // Printing a message to the LCD Screen.
lcd.setCursor(0, 1); // The LCD screen coordinates.
lcd.print(" Hi Welcome To "); // The text that will be displayed on the LCD screen.
lcd.setCursor(0, 2); // The LCD screen coordinates.
lcd.print(" -"); // the text that will display on LDC screen -
delay(5000); // Delay time by 5000.
lcd.clear(); // To clear the LCD screen.
lcd.setCursor(0, 0); // The screen coordinates.
lcd.print("NETWORK Project"); // the text that will display on the LCD screen.
lcd.setCursor(0, 1); // The LCD screen coordinates.
lcd.print(" Car Parking system"); // The text that will be displayed on the LCD screen.
lcd.setCursor(0, 2); // The LCD screen coordinates.
lcd.print("      System      "); // The text that will be displayed on the LCD screen.
delay(5000); // Delay time by 5000.
lcd.clear(); // To clear the LCD screen.
pinMode(redPin, OUTPUT); // Taking the output for the RGB light Red.
pinMode(greenPin, OUTPUT); // Taking the output for the RGB light Green.

Read_Sensor(); // Reading sensor info.

int total = S1 + S2; // To calculate the total parking spaces.
slot = slot - total; // To calculate a single parking space.
}
```

void loop() it loops through the devices. Then reads the information in each sensor. For it to print the number of available slots. All lcd.print means it prints on the LCD Display screen. But the Serial.println it prints all the output on the server which is on the laptop screen. Serial.println(“Avai slot: “) prints the output on the laptop screen (server) and the same for the rest prints the s1 is full or s1 is empty and same for s2 if full or empty.

```
void loop() { // Loop for the devices

  Read_Sensor(); // Reading sensor info.

  lcd.setCursor(0, 0); // The LCD screen coordinates.
  lcd.print("  Avail Slot: "); // The text that will be displayed on the LCD screen.
  lcd.print(slot); // Printing number of available slots.

  Serial.println("Avail Slot: "); // The text that will be displays it on the server laptop screen
  Serial.println(slot); // The text that will be displayed on the LCD screen.
  Serial.println(" "); // The text that will be displays it on the server laptop screen.

  lcd.setCursor(0, 1); // The LCD screen coordinates.
  if (S1 == 1) { // If statment if the slot 1 was not available.
    lcd.print("Slot1 Full "); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Full "); // The text that will be displays it on the server laptop screen if the slot is full.
    digitalWrite(greenPin, LOW); // Not show the RGB Green light (set green light as LOW).
    digitalWrite(redPin, HIGH); // Show the RGB Red light (set red light as HIGH).
  }
  else { // Else for the if statment when slot 1 is available.
    lcd.print("S1:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Empty"); // The text that will be displays it on the server laptop screen if the slot is Empty.
    digitalWrite(redPin, LOW); // Not show the RGB Red light (set red light as LOW).
    digitalWrite(greenPin, HIGH); // Show the RGB Green light (set Green light as HIGH).
  }
}
```

the if statement for digital read for the gate that justifies if the car parking slots are full, and it does not allow the more than two cars enter. If one of the cars exited the parking lot in updates it and allows one more car to enter and so on.

```
lcd.setCursor(0, 1); // The LCD screen coordinates.
if (S2 == 1) { // If statment if the slot 2 was not available.
  lcd.print("S2:Full "); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Full "); // The text that will be displays it on the server laptop screen if the slot is full.
  // there is suppose to be a code here for the red lights but the light has burnsup and does not work.
}
else { // Else for the if statment when slot 2 is available.
  lcd.print("S2:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Empty"); // The text that will be displays it on the server laptop screen if the slot is Empty.
  // there is suppose to be a code here for the red lights but the light has burnsup and does not work.
}

if (digitalRead(ir_enter) == 0 && flag1 == 0) { //If statment for the gate if the car parking is full.
  if (slot > 0) {
    flag1 = 1;
    if (flag2 == 0) {
      myservo.write(180);
      slot = slot - 1;
    }
  }
  else {
    lcd.setCursor(0, 0); // The LCD screen coordinates.
    lcd.print("Parking Full"); // The text that will be displayed on the LCD screen.
    Serial.println("Parking Full"); // The text that will be displays it on the server laptop screen.
    delay(1500); // Delay by 1500.
  }
}
```


2- The client will:

For the client we working on setting up the Arduino Uno by the method void setup(). Pinmode initializes the input to get the desired output, here we implied it on the parking slots, entering and exiting the gate by the IR sensor. Myservo.attach which is used for the gate and the write is for the beginning point.

```
void setup() { // Settingup the Arduino UNO.
  Serial.begin(9600); //The beginning point.
  pinMode(ir_car1, INPUT); // Taking the input for the IR sensor of the car slot 1.
  pinMode(ir_car2, INPUT); // Taking the input for the IR sensor of the car slot 2.
  pinMode(ir_enter, INPUT); // Taking the input for the IR sensor of the entering the gate.
  pinMode(ir_back, INPUT); // Taking the input for the IR sensor for exiting the gate.
  myservo.attach(3); // The servo moter that used for the gate.
  myservo.write(90); // The beginning point of the servo moter.
}
```

the difference in this loop is that we have a new initialization which is the digital write (digitalWrite(greenpin, LOW)) which means that if the parking slot is full meaning there is a car parked then set green light to low and set red light to high. Else, if the slot is empty then set green light to high and red light to low.

```
void loop() { // Loop for the devices

  Read_Sensor(); // Reading sensor info.

  lcd.setCursor(0, 0); // The LCD screen coordinates.
  lcd.print(" Avai Slot: "); // The text that will be displayed on the LCD screen.
  lcd.print(slot); // Printing number of avalibale slots.

  Serial.println(" Avai Slot: "); // The text that will be displays it on the server laptop screen.
  Serial.println(slot); // Printing number of avalibale slots.
  lcd.print(" "); // The text that will be displayed on the LCD screen.
  Serial.println(" "); // The text that will be displays it on the server laptop screen.

  lcd.setCursor(0, 1); // The LCD screen coordinates.
  if (S1 == 1) { // If statment if the slot 1 was not avalibale.
    lcd.print("S1:Full "); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Full "); // The text that will be displays it on the server laptop screen if the slot is full.
    digitalWrite(greenPin, LOW); // Not show the RGB Green light (set green light as LOW).
    digitalWrite(redPin, HIGH); // Show the RGB Red light (set red light as HIGH).
  }
  else { // Else for the if statement when slot 1 is avalibale.
    lcd.print("S1:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Empty"); // The text that will be displays it on the server laptop screen if the slot is Empty.
    digitalWrite(redPin, LOW); // Not show the RGB Red light (set red light as LOW).
    digitalWrite(greenPin, HIGH); // Show the RGB Green light (set green light as HIGH).
  }
}
```

Within the black box it uses more of the digital read the first if statement is for the gate if the car parking is full. And the second is also for the gate if there is available slots and how many. So as the last two if statements informing if slot one equals to 1 then its full and same for slot two. Between these if statements there is commands that I have explained previously.

```

lcd.setCursor(8, 1); // The LCD screen coordinates.
if (S2 == 1) { // If statement if the slot 2 was not available.
  lcd.print("S2:Full "); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Full "); // The text that will be displayed on the server laptop screen if the slot is full.
  // there is supposed to be a code here for the RGB lights but the light has burned up and does not work.
}
else { // Else for the if statement when slot 2 is available.
  lcd.print("S2:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Empty"); // The text that will be displayed on the server laptop screen if the slot is empty.
  // there is supposed to be a code here for the RGB lights but the light has burned up and does not work.
}

if (digitalRead(ir_enter) == 0 && flag1 == 0) { // If statement for the gate if the car parking is full.
  if (slot > 0) {
    flag1 = 1;
    if (flag2 == 0) {
      myservo.write(180);
      slot = slot - 1;
    }
  } else {
    lcd.setCursor(0, 0); // The LCD screen coordinates.
    lcd.print("Parking Full"); // The text that will be displayed on the LCD screen.
    Serial.print("Parking Full"); // The text that will be displayed on the server laptop screen.
    delay(1500); // Delay by 1500.
  }
}

if (digitalRead(ir_back) == 0 && flag2 == 0) { // If statement for the gate if the car parking has available slots.
  flag2 = 1;
  if (flag1 == 0) {
    myservo.write(180);
    slot = slot + 1;
  }
}

if (flag1 == 1 && flag2 == 1)
  delay(1000);
myservo.write(90);
flag1 = 0, flag2 = 0;

delay(1); // Delay by 1.
}

void Read_Sensor() {
  S1 = 0, S2 = 0; // Initiation for both sensors.

  if (digitalRead(ir_car1) == 0) { // If slot one = 1 it means its full
    S1 = 1;
  }
  if (digitalRead(ir_car2) == 0) { // If slot two = 1 it means its full
    S2 = 1;
  }
}

```

3-Steps for setting up the network

In this section of the report explain how you set up two hosts to communicate over a network. Describe any hardware or software you used to connect the hosts. Clearly show if you used wired/wireless technology

- **Step One:** we connected the jump wires from the breadboard to the Arduino Uno.
- **Step Two:** 2 IR Sensor for gate
 - ✚ we have 2 IR sensors for the gate it senses how many cars have entered the parking lot and if it full or not.
 - ✚ We connected the two IR sensors that has three jump wires connected to it to the Arduino Uno, two of them are in pot 4 and one in GND.
 - ✚ Since there is not a lot of pots in the Arduino we connected the jump wires to the Arduino pot 4 and GND pot and from there to the breadboard in order to connect more than one IR sensor to the same pot.
- **Step Three:** Servo Motor
 - ✚ One servo motor that have three wires two connected to GND and one to pot 3 in the Arduino Uno.
 - ✚ It senses the car movement from IR sensor for it to open the gate to allow the car to pass by and in the other side of the gate it has another IR sensor to sense the movement of the car for it close the gate. And decreases the available slots
- **Step Four:** 2 IR sensor for parking
 - ✚ Two more IR sensors are for the parking slots where it senses if there is a car parked in each parking or not.
 - ✚ The two IR sensor are connected from the Arduino Uno to the IR sensor.
 - ✚ The two sensors are connected by the jump wires to the Arduino. Two of the wires are connected form the IR sensor to the Arduino GND pot and one connected to pot 5.
- **Step Five:** RGB Light
 - ✚ for the RGB light we have inserted it in the breadboard with three jump wires surrounding it.
 - ✚ One wire connected the pot 7 and the other connected to the pot 8 and last connected to the GND.
 - ✚ As well two 220 R resister is inserted near the wires and the RGB light
 - ✚ It is programmed in order to turn red if the parking slot is not empty (car is parked), and turns green if the parking slot is empty.
- **Step Six:** LCD Display 16x2 with its serial interface adapter

- ✚ Has four wires connected to the serial interface adapter that is connected with the LCD display in order to connect it the Arduino. Four wires, two of the wires are connected to the GND pot and one to the A5 pot and the last wires is connected to the A4.
- ✚ It displays the output that is programmed. It displays first the welcome message and then “Network project” phrase. Then, it displays how many available slots and if the slots are empty or not. Also it displays for the cars if the parking lot is full.
- **Step Seven: Server**
 - ✚ We connected the Arduino Uno with all the sensors to connect to it to the laptop. Where it displays the code and the output. It updates continuously.

4- Codes and comments:

Code of server side:

```
lcd.init(); // Initializing the LCD Screen.
lcd.backlight(); // Printing a message to the LCD Screen.
lcd.setCursor(0, 1); // The LCD screen coordinates.
lcd.print(" Hi Welcome To "); // The text that will be displayed on the LCD screen.
lcd.setCursor(0, 2); // The LCD screen coordinates.
lcd.print(" -"); // the text that will display on LDC screen -
delay(5000); // Delay time by 5000.
lcd.clear(); // To clear the LCD screen.
lcd.setCursor(0, 0); // The screen coordinates.
lcd.print("NETWORK Project"); // the text that will display on the LCD screen.
lcd.setCursor(0, 1); // The LCD screen coordinates.
lcd.print(" Car Parking system"); // The text that will be displayed on the LCD screen.
lcd.setCursor(0, 2); // The LCD screen coordinates.
lcd.print("      System      "); // The text that will be displayed on the LCD screen.
delay(5000); // Delay time by 5000.
lcd.clear(); // To clear the LCD screen.
pinMode(redPin, OUTPUT); // Taking the output for the RGB light Red.
pinMode(greenPin, OUTPUT); // Taking the output for the RGB light Green.

Read_Sensor(); // Reading sensor info.

int total = S1 + S2; // To calculate the total parking spaces.
slot = slot - total; // To calculate a single parking space.
}

void loop() { // Loop for the devices

  Read_Sensor(); // Reading sensor info.

  lcd.setCursor(0, 0); // The LCD screen coordinates.
  lcd.print("  Avai Slot: "); // The text that will be displayed on the LCD screen.
  lcd.print(slot); // Printing number of avalibale slots.

  Serial.println("  Avai Slot: "); // The text that will be displayed on the LCD screen.
  Serial.println(slot); // Printing number of avalibale slots.
  lcd.print("      "); // The text that will be displayed on the LCD screen.
  Serial.println("      "); // The text that will be displayed on the LCD screen.

  lcd.setCursor(0, 1); // The LCD screen coordinates.
  if (S1 == 1) { // If statment if the slot 1 was not avalible.
    lcd.print("S1:Full "); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Full "); // The text that will be displayed on the LCD screen if the slot is full.
    digitalWrite(greenPin, LOW); // Not show the RGB Green light (set green light as LOW).
    digitalWrite(redPin, HIGH); // Show the RGB Red light (set red light as HIGH).
  }
  else { // Else for the if statement when slot 1 is avalibale.
    lcd.print("S1:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
    Serial.println("S1:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
    digitalWrite(redPin, LOW); // Not show the RGB Red light (set red light as LOW).
    digitalWrite(greenPin, HIGH); // Show the RGB Green light (set Green light as HIGH).
  }
}

lcd.setCursor(0, 1); // The LCD screen coordinates.
if (S2 == 1) { // If statment if the slot 2 was not avalible.
  lcd.print("S2:Full "); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Full "); // The text that will be displayed on the LCD screen if the slot is full.
  // there is suppose to be a code here for the RGB lights but the Light has burnedup and does not work.
}
else { // Else for the if statement when slot 2 is avalibale.
  lcd.print("S2:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
  // there is suppose to be a code here for the RGB lights but the Light has burnedup and does not work.
}

}

if (digitalRead(ir_enter) == 0 && flag1 == 0) { //If statment for the gate if the car parking is full.
  if (slot > 0) {
    flag1 = 1;
    if ((flag2 == 0) {
      myservo.write(180);
      slot = slot - 1;
    }
  }
  else {
    lcd.setCursor(0, 0); // The LCD screen coordinates.
    lcd.print("Parking Full"); // The text that will be displayed on the LCD screen.
    Serial.println("Parking Full"); // The text that will be displayed on the LCD screen.
    delay(1500); // Delay by 1500.
  }
}
```

Code for client side:

```
lcd.setCursor (8, 1); // The LCD screen coordinates.
if (S2 == 1) { // If statment if the slot 2 was not avalible.
  lcd.print("S2:Full "); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Full "); // The text that will be displayed on the LCD screen if the slot is full.
  // there is suppose to be a code here for the RGB lights but the Light has burnedup and does not work.
}
else { // Else for the if statement when slot 2 is avalibale.
  lcd.print("S2:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
  Serial.println("S2:Empty"); // The text that will be displayed on the LCD screen if the slot is full.
  // there is suppose to be a code here for the RGB lights but the Light has burnedup and does not work.
}

if (digitalRead (ir_enter) == 0 && flag1 == 0) { //If statment for the gate if the car parking is full.
  if (slot > 0) {
    flag1 = 1;
    if (flag2 == 0) {
      myservo.write(180);
      slot = slot - 1;
    }
  }
  else {
    lcd.setCursor (0, 0); // The LCD screen coordinates.
    lcd.print("Parking Full"); // The text that will be displayed on the LCD screen.
    Serial.println("Parking Full"); // The text that will be displayed on the LCD screen.
    delay(1500); // Delay by 1500.
  }
}

if (digitalRead (ir_back) == 0 && flag2 == 0) { //If statment for the gate if the car parking has avalibe slots.
  flag2 = 1;
  if (flag1 == 0) {
    myservo.write(180);
    slot = slot + 1;
  }
}

if (flag1 == 1 && flag2 == 1)
  delay (1000);
  myservo.write(90);
  flag1 = 0, flag2 = 0;
}

delay(1); //Delay by 1.
}

void Read_Sensor() {
  S1 = 0, S2 = 0; //Initiation for both sensors.

  if (digitalRead(ir_car1) == 0) { // if slot one = 1 it means its full
    S1 = 1;
  }
  if (digitalRead(ir_car2) == 0) { // if slot two = 1 it means its full
    S2 = 1;
  }
}
```

5- Snapshots of the application outputs.

Provide a screenshot of the output as it appears in the programming environment.
Make sure the output is readable.

NOTE:

Clients:

- ✚ Arduino
- ✚ IR Sensor
- ✚ servo motor

Server:

- ✚ computer connected with LCD screen to show the available and occupied parking spaces

Video:

<https://drive.google.com/drive/folders/14RS9etYydN2VQWFuWMclf0g39mRJshDy>

Screenshot:

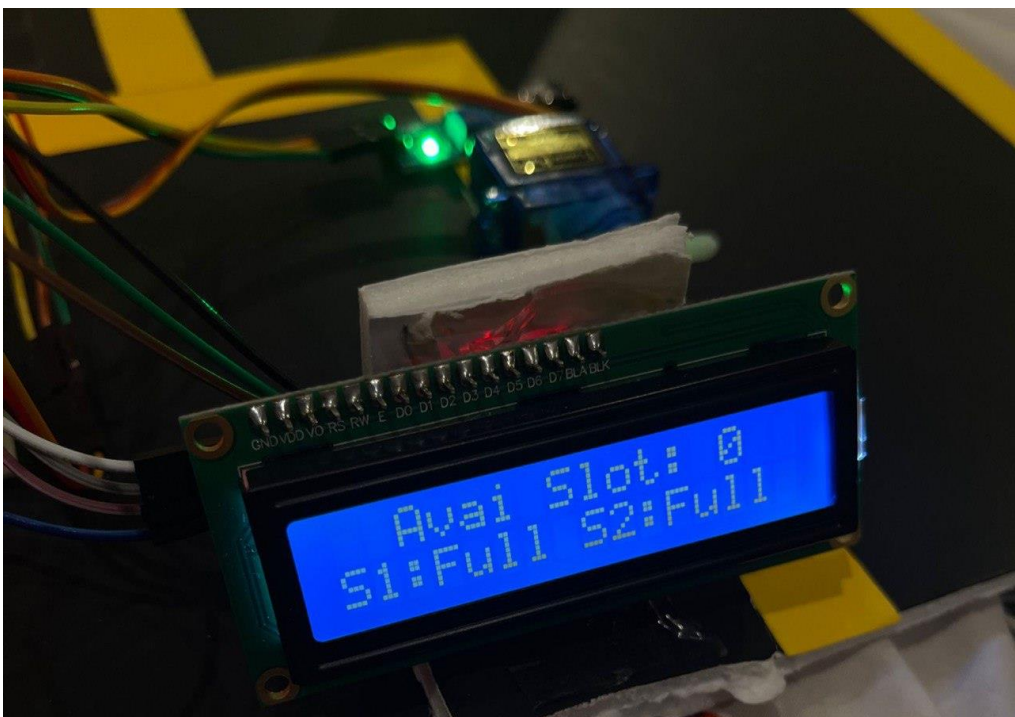
Before making any changes to the environment



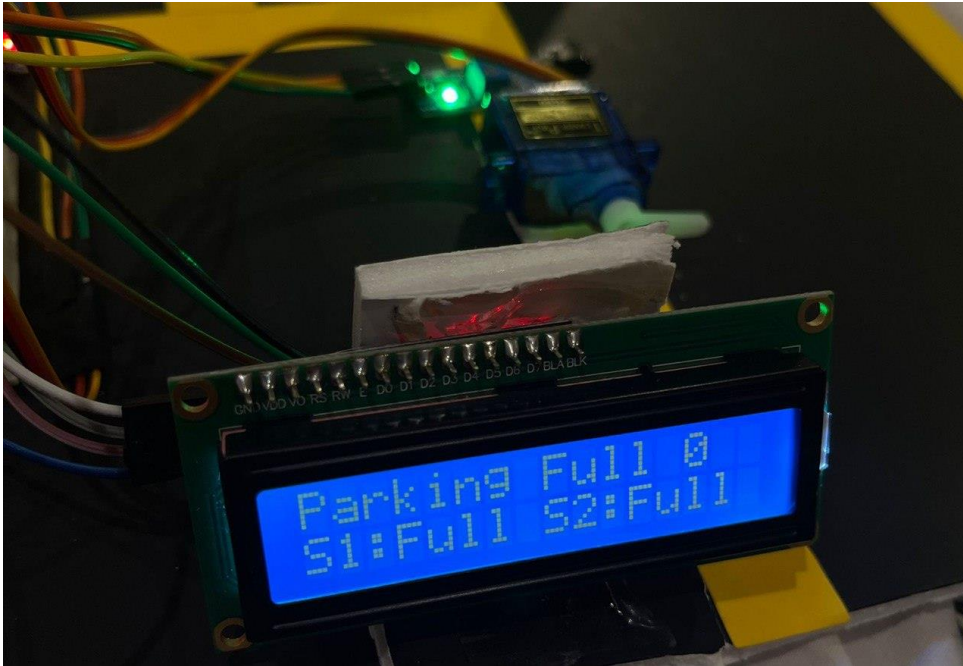
After passing the first car



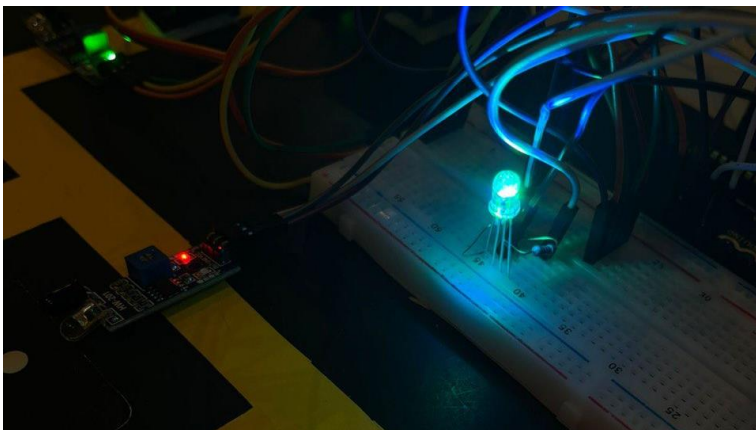
After the second car passed



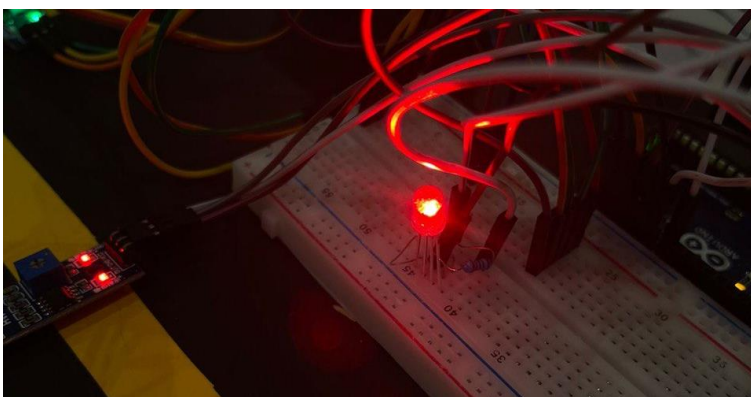
No more than two cars are allowed



The color of the lights is green when there is no car in the parking



The color of the lights is red when there is a car in the parking



6- Problems and solutions:

This is the most important section of the report. Describe all the problems and challenges you faced in completing the project and cite any resources you used to solve the problems.

If you leave this section empty then you are suggesting that you did not complete the project yourself. It is expected that any undergraduate student undertaking this project will face a number of technical and/or programming challenges.

Problem 1: The costs of building this project were very expensive where it cost up to **.700SAR** (even when trying to make it as affordable as possible)

Solution: -

.Problem 2: Understanding the project goal and what is needed to be achieved

.Solution: The team has asked their teacher questions regarding that topic

.Problem 3: Getting the right tools and hardware

Solution: The team has asked their teacher questions regarding the pieces needed for this project as well as researched some YouTube videos of similar projects

.Problem 4: Understanding how to build and plug the Arduino and how to code it

Solution: Searching codes and methods on YouTube, Arduino website and Github as well as understanding them

.Problem 5: The quality of the hardware

Solution: The team had to buy better quality hardware as the ones that were ordered where either damaged or broken. *one of the RGB lights have burned out due to poor quality

.Problem 6: Time

Solution: The team needed to abandon some pieces as there was no time to rebuild or order new pieces and had to work with what is available

.Problem 7: Time schedules

Solution: Team members have planned specific days and places to meet to be able to build the project together and be able to balance other projects as well

.Problem 8: The teams first time experience with the hardware

Solution: Team members have researched information about the hardware (what the hardware looks like and serves what purpose)

.Problem 9: Learning the Arduino coding language (Arduino C)

.Solution: Searching YouTube videos of explanations and visiting the Arduino website

.Problem 10: Finding the correct resources

Solution: Where the team have shared links or videos and websites that is closest for achieving this project.

NOTE:

Two RGB lights were purchased, but one of the two lights became damaged and one light was relied upon due to the inability to buy again.

References:

List any resources, you used to complete the project, in the order they appear in the report

- 1- <https://youtu.be/5Qi93MjlqzE>
- 2- https://youtu.be/6gccSyp_uJQ
- 3- <https://www.viralsciencecreativity.com/>
- 4- <https://www.arduino.cc>
- 5- <https://www.prateeks.in/2021/11/car-parking-system-using-arduino.html?m=1>