EC Project

Team Members:-

1. Abhishek Rajwanshi(200027)
2. Aman Singh(200104)
3. Ashish Yadav(200212)
4. Harsh Kumar Pandey(200414)
5. Ayush Garg(200241)

Section - C

Table No. - 5

Q1. What problem are you trying to solve, and why is it important/interesting?

Ans. We are making a finger-print based door lock system which solved the following problems:-

1. **Security:** Fingerprint-based door locks provide a higher level of security compared to traditional lock systems. Since fingerprints are unique to each individual, it is much harder for unauthorized individuals to gain access to a secured area.
2. **Convenience:** With fingerprint-based door locks, you no longer need to carry around a physical key or remember a combination code. This can be particularly convenient for individuals who frequently lose or misplace their keys.
3. **Key management:** Fingerprint-based door locks can eliminate the need for key duplication, reducing the risk of unauthorized key duplication and making key management easier.

Due to the above-mentioned problems, it will be quite interesting for us to see how the fingerprint is used to replace the old existent method for door unlocking. We are students of a technical institute who are interested in cutting-edge technology and innovation. Thus, making this project is of quite a relevance to us.

Q2. What are the existing solutions? Describe a few of them and list any shortcomings in them. Is your solution approach unique in some way?

Ans. There are existing solutions that solve this problem. They use different hardware for fingerprint sensors. Also, they make use of 3rd party apps, which can have a risk of losing personal data of high security. We are making our own app that doesn't store other data and, thus, is totally secure.

Q3. What resources do you require to complete the project? Give a breakup of tasks that you need to accomplish week by week to complete the project.

Ans. The components that we need are as follows:-

S.N	Components Name	Quantity
1	Arduino Nano	1
2	HC-05 Bluetooth Module	1
3	Solenoid Lock	1
4	Single Channel Relay Module	1
5	Buzzer	1
6	Breadboard	1
7	Jumper Wires	5

Week wise breakup of tasks -

Week 1:

- Research and acquire the necessary components for the project.
- Familiarize ourselves with the basic concepts of Arduino programming, Bluetooth communication, solenoid locks, and other components.
- Write and upload a simple Arduino program that blinks the LED light of Ardunio on and off.
- Measure and show the resistor value using a Arduino code

Week 2:

- Build the circuit on a breadboard and test the solenoid lock to make sure it works properly.
- Write and upload a simple Arduino program that turns on and off the solenoid lock using a pushbutton or switch.
- Write and upload an Arduino program that reads the fingerprint data from the sensor and sends it to a connected device over Bluetooth.
- Test the Bluetooth communication and make sure that the fingerprint data is transmitted correctly.

Week 3:

- Create an Android app that can connect to the Bluetooth module and receive fingerprint data.
- Write the code to verify the fingerprint data and control the solenoid lock accordingly.
- Test the complete system by scanning fingerprints and verifying that the solenoid lock opens or remains locked based on the authentication.

Project Description:-

We made a finger-print based door lock system. The system scans the person's fingerprint using the in-built fingerprint sensor of a mobile phone, compares it with the correct fingerprint already stored in the phone, and sends a signal to the Arduino board based on this result, using a Bluetooth module. Based on the signal received, the board, using the other components of the circuit, decides whether to open the gate or have it remain closed.
For scanning the fingerprint, we made an app using https://www.kodular.io/.

Components Used:-

1. **Arduino Nano-**

   It is the development board relying upon a microcontroller in it. It was used for changing the voltage levels depending upon the fingerprint scanned.

2. **HC05 Bluetooth Module -**

   It was the Bluetooth module used for sending serial data to the Arduino board based on the fingerprint scan.

3. **Dual-Channel Relay Module-**

   Earlier, a single-channel relay module was used, but it was found to be defective, so it was replaced with a dual-channel relay module. It was used as a relay switch between the digital pin of the Arduino and the 12 V solenoid lock, as 12 V can't be supplied directly to the Arduino.

4. **12 V Solenoid Lock-**

   It was the main door lock to be opened and is frequently used in modern-day doors.

5. **Buzzer-**

   It was used for producing a short beep sound or long duration sound, depending upon fingerprint recognition.

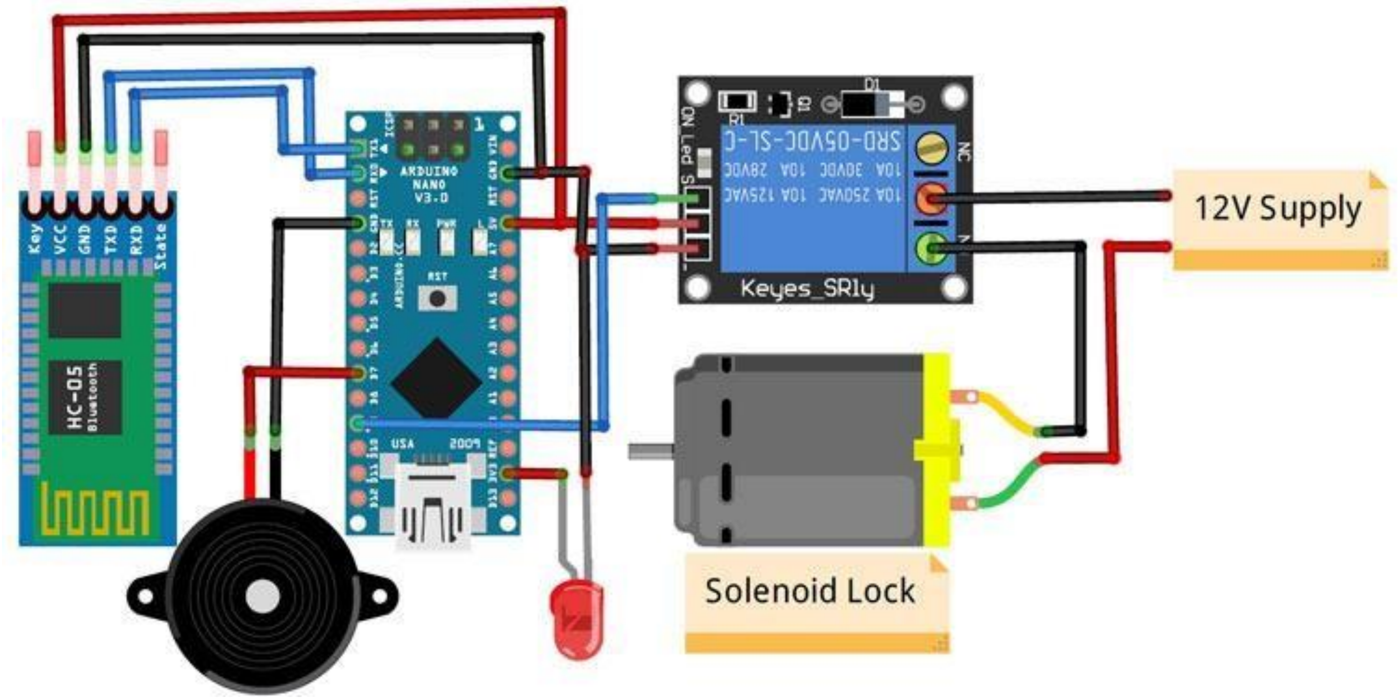
6. **Jumper Wires-**

   5 Jumper wires were used for connecting different modules.

7. **Breadboard-**

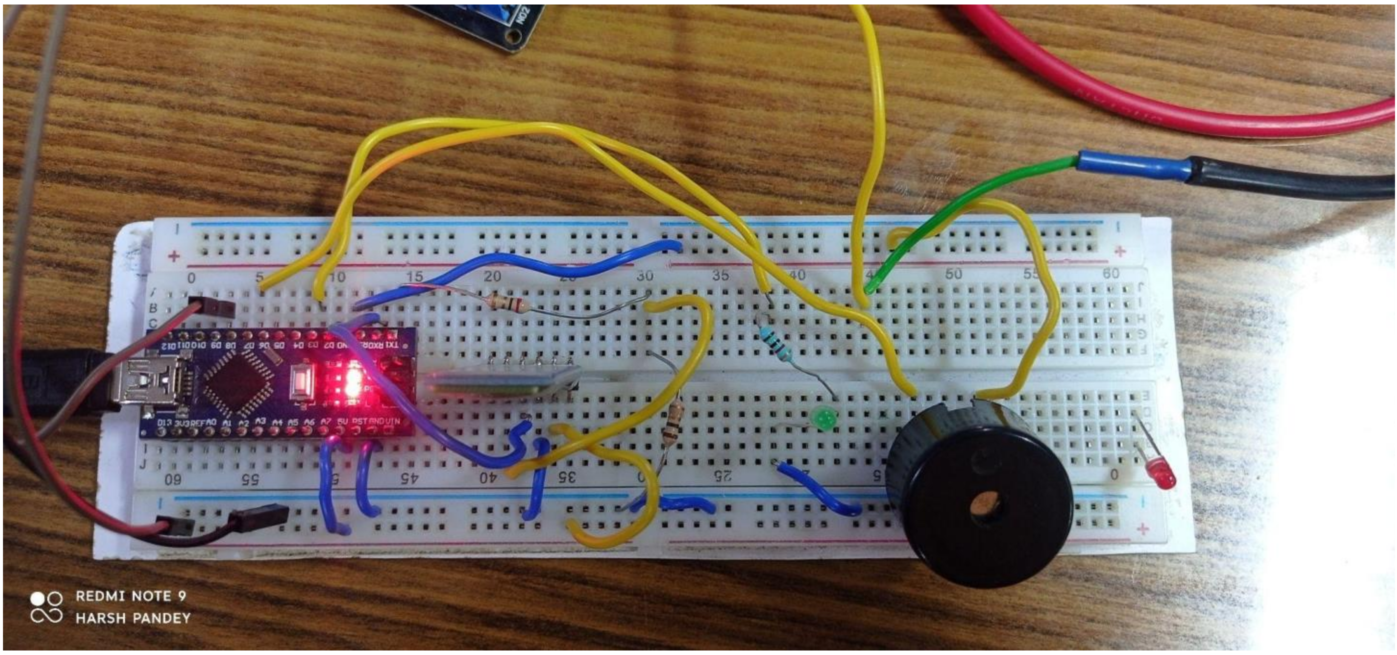
   It was used for making the whole circuit.

Circuit Diagram-



- The TX and RX pins of HC05 were connected to the RX and TX of the Arduino Nano. RX was directly connected to the TX of Arduino Nano, but the TX was connected to the RX of Arduino through a potential divider of 1 $k\Omega$ and 2 $k\Omega$.
- V_{CC} of the Bluetooth was connected to V_{CC}(5 V) of the Nano. The GND of the Bluetooth was connected to the GND of the Nano.
- The positive terminal of the buzzer was connected to digital pin-7(D7) of the Nano, while the negative terminal was connected to the GND of the Nano.

- Similarly, in the Dual Channel Relay module, the V_{CC} and GND were connected to V_{CC}(5 V) and GND of the Nano, respectively. The IN1 pin of the relay was connected to digital pin 9 (D9) of the nano.
- The Common terminal of the relay was a module connected to the ground of the 12 V supply. The NC1 terminal was connected to the negative terminal of the solenoid lock.
- The positive terminal of the lock was connected directly 12 V supply.

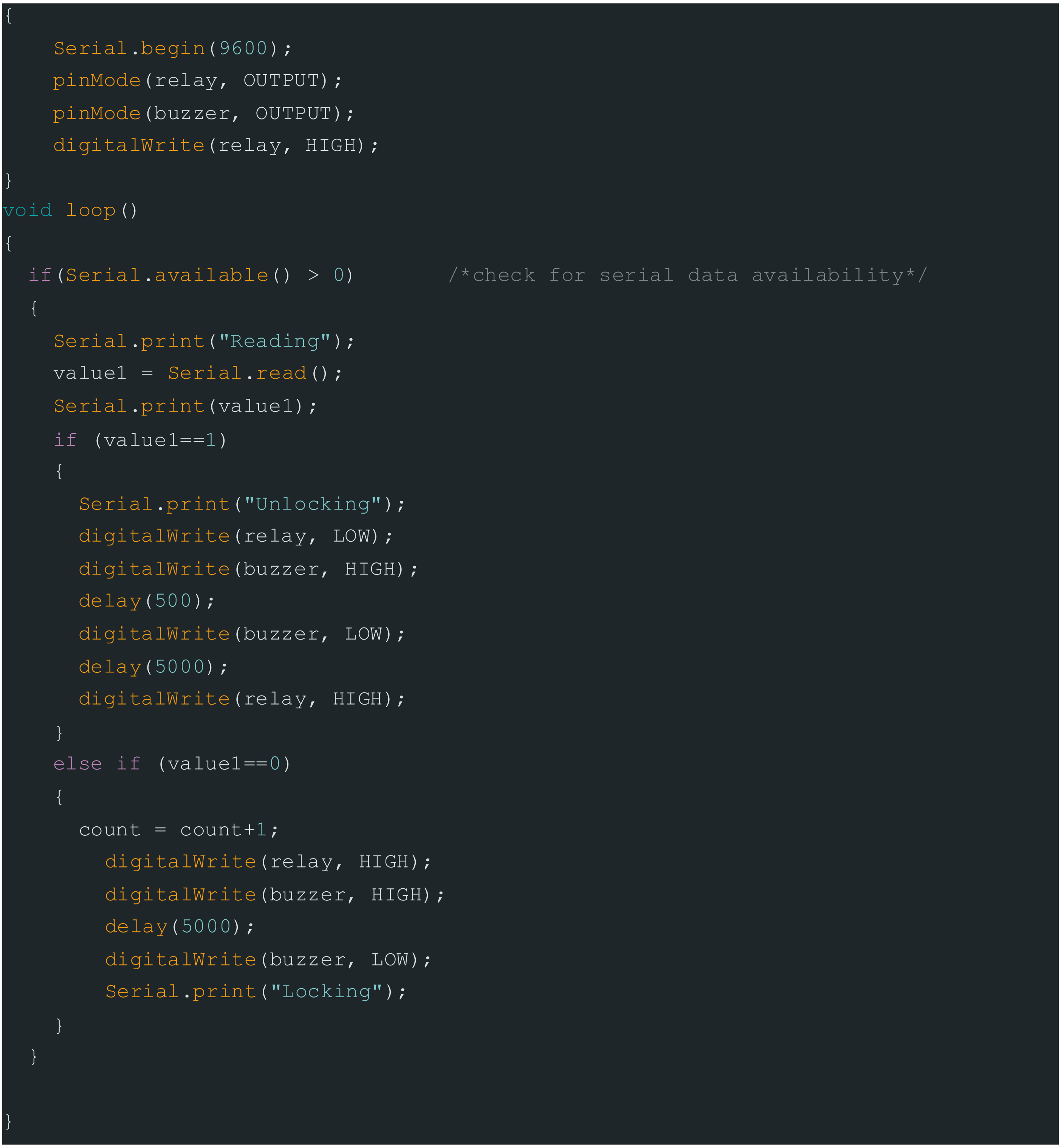


Working Explained-

- The TX and RX terminals were used by the HC05 to send binary data(1 or 0) to the Arduino nano depending on the fingerprint recognized.
- If the fingerprint recognized is correct, the Bluetooth module sends 1 to the Arduino Nano. Whenever 1 is received by the Nano, it sets pin 7(D7) and pin 9(D9) to HIGH and LOW, respectively. This plays the buzzer and opens the lock for a specified duration(which can be altered by changing the code).
- Similarly, if the fingerprint recognized is incorrect, the Bluetooth module sends 0 to the Arduino Nano. Whenever 0 is received by the Nano, it sets pin 7(D7) and pin 9(D9) to HIGH and HIGH, respectively. This plays the buzzer for a longer duration(which can be altered by changing the code) and has the lock in a closed state.

Code Used-

```
int value1;
int count = 0;
#define relay 9
const int buzzer = 7;
void setup()
```



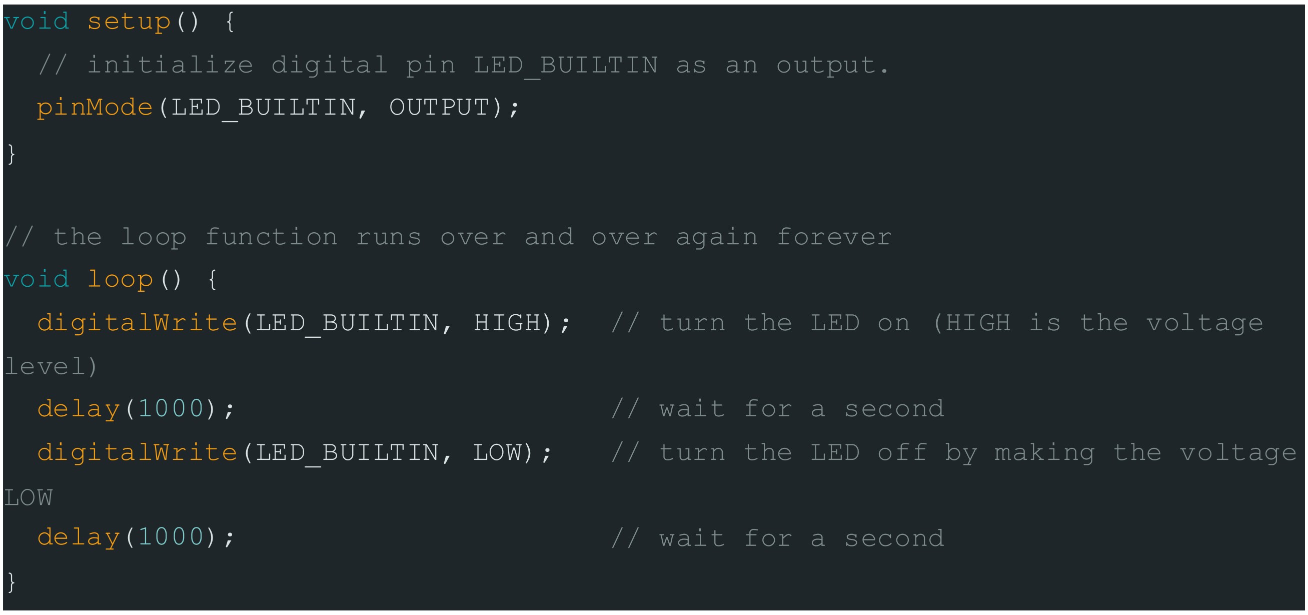
```
{
    Serial.begin(9600);
    pinMode(relay, OUTPUT);
    pinMode(buzzer, OUTPUT);
    digitalWrite(relay, HIGH);
}
void loop()
{
  if(Serial.available() > 0)       /*check for serial data availability*/
  {
    Serial.print("Reading");
    value1 = Serial.read();
    Serial.print(value1);
    if (value1==1)
    {
      Serial.print("Unlocking");
      digitalWrite(relay, LOW);
      digitalWrite(buzzer, HIGH);
      delay(500);
      digitalWrite(buzzer, LOW);
      delay(5000);
      digitalWrite(relay, HIGH);
    }
    else if (value1==0)
    {
      count = count+1;
        digitalWrite(relay, HIGH);
        digitalWrite(buzzer, HIGH);
        delay(5000);
        digitalWrite(buzzer, LOW);
        Serial.print("Locking");
    }
  }

}
```

About the App -

The app was created using Kodular's drag-and-drop blocks-based interface. The logic for sending data to the HC-05 Bluetooth module based on fingerprint detection was implemented using blocks in the 'Blocks' screen. The 'Fingerprint' component detected fingerprint input, and the 'Bluetooth Client' component was used to establish a connection and send data to the HC-05 Bluetooth module. When a correct fingerprint was detected, the app sent '1' to the Bluetooth module, and for incorrect fingerprint detection, it sent '0'. The 'List Picker' component allowed the user to select an option, and the selected option was displayed using the 'Text' block. Once the logic was implemented, the app was exported to a .apk file for installation on a device for testing or distribution.

Workflow-

- First of all, we tested the working of Arduino Nano by running a blink code given as an example in Arduino IDE.



```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage
level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage
LOW
  delay(1000);                      // wait for a second
}
```

- After that, we did connect the potentiometer to the Arduino and displayed the value of resistance.
- Next, we checked the Solenoid Lock by directly connecting it to the 12 V supply, and it got unlocked upon providing the supply.
- In the next step, we had to check the working of the Relay module, so we connected a LED along with a resistor and wrote a code that would ON and OFF the relay in a loop with a delay of 5s. On compiling and uploading, the LED should have been ON and OFF alternatively, but it didn't. We checked the circuit but found no mistake, so we changed the Relay module and used a Dual Channel Relay module, which worked thereafter.
- The next step was to check the HC05 Bluetooth module. We did the same thing above with the Bluetooth module. The LED would blink ON and OFF depending upon the data available from the Bluetooth. One thing that we had to take care of was while compiling and uploading, we had to remove the serial pin connections and connect them only after the code had compiled and uploaded correctly. This thing worked on sending 0 and 1 from the mobile app.
- The last thing was to check the working of the buzzer. Buzzer was connected to the D7 pin and was made ON and OFF alternatively with a delay of 5s. This also worked without any problem.
- Now, since all the modules were working correctly, we breadboarded the whole circuit, as shown above. 12 V voltage was supplied, and the code was fed into the nano. After that, the Bluetooth module was connected to the mobile, and the app was

used for scanning and recognizing the fingerprint. When the correct finger was used, the door opened for some duration, and the buzzer gave a beep sound. On using the incorrect finger, the door remained locked, and the buzzer gave a constant alarming sound for some duration. With this, our finger-print based door lock was made successfully.

Final Result -

The final project can be seen in the following video -
https://drive.google.com/file/d/1_GRQJTVPO2Lbnfmg-dmXJk98TI-mHPnk/view?usp=sharing