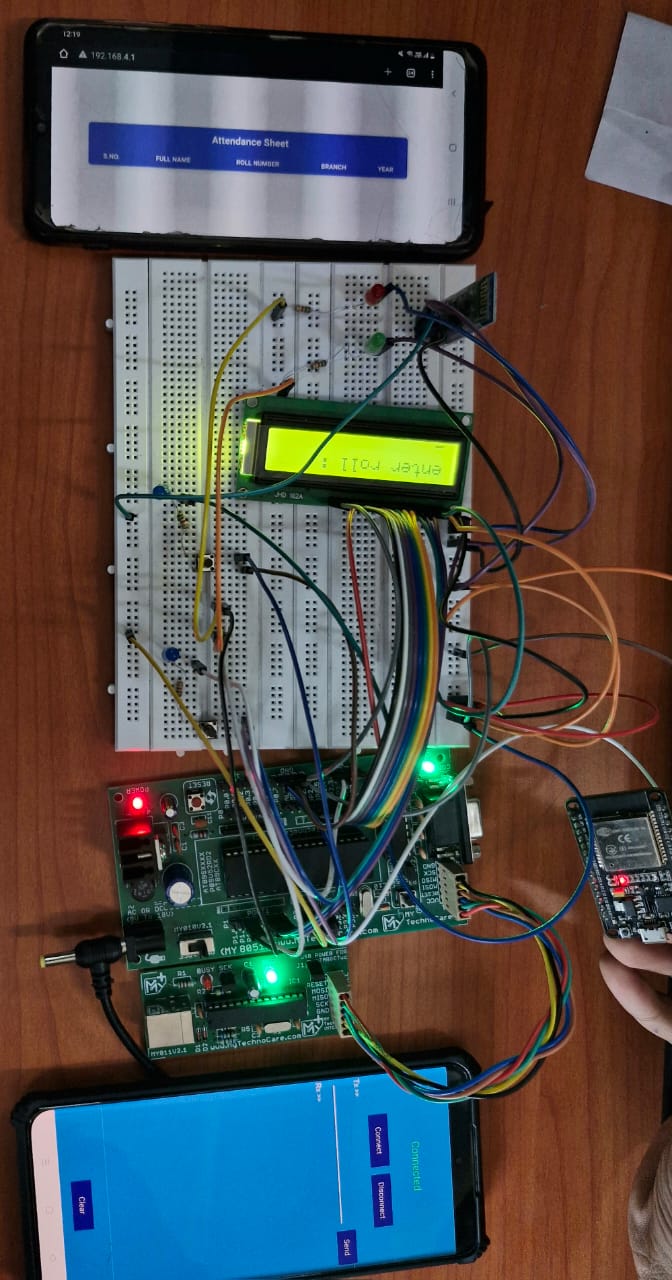


**EMBEDDED PROJECT  
EE3401  
PROJECT NUMBER-33  
Automated attendance system**

**Submission To :– Dr. Supratim Gupta Sir**

* **Sanjeeb Dash ( 122EE0515 )**
* **Anwesh Panda ( 122EE0833 )**
* **Subham Shibasis Ghose ( 122EE0598 )**

**INTRODUCTION**

* An attendance system is crucial for efficiently tracking student participation and ensuring accountability. It provides a reliable record of attendance, helping identify patterns of absenteeism and enabling timely intervention. By automating the process, it reduces administrative workload and promotes transparency. An effective attendance system encourages punctuality, enhances student engagement, and supports the overall management of educational institutions
* Attendance requirements are essential for fostering a disciplined and productive learning environment. Regular participation ensures students stay engaged, absorb course material effectively, and develop a sense of responsibility. Consistent attendance also promotes interaction with peers and instructors, contributing to a deeper understanding of the subject. Ultimately, it reinforces the value of commitment and helps achieve academic success.

**OBJECTIVE**

1. **Automate Attendance Tracking**

To automate the process of recording student attendance, eliminating manual entry errors and saving time

1. **Time Efficiency**

To streamline the attendance process, reducing class time spent on roll calls and increasing focus on learning.

1. **Centralized Record Management**

To store and manage attendance data in a centralized database for easy access and retrieval

1. **Enhanced Accuracy**

To minimize human error by using automated systems (like RFID or fingerprint recognition) for attendance marking

**COMPONENTS**

* **8051 Micro-controller**
* **LED Indicators**
* **LCD Display ( 2 X 16 )**
* **Jumper Wire ( M-M, M-F, F-F)**
* **Proteus Software**
* **8051 Trainer Kit**
* **HC-05 Bluetooth Module**
* **Registers ( 1.5 KΩ )**
* **Potentiometer**

**CIRCUIT DESCRIPTION**

**1.Microcontroller (8051) : The 8051** microcontroller is the core of the system, responsible for controlling the operation of all connected peripherals. It processes input from the push buttons and Bluetooth module, stores attendance data, and drives the LCD display to show attendance status.

**2. LCD Display :** A 16x2 LCD is used to show the status of student attendance. It displays whether a student is "Present" or "Absent" based on the data processed by the 8051 microcontroller. It also shows messages for enrollment or attendance checking operations.

**3. Push Buttons :** There are two push buttons in the system: enrolling students into the system. Enroll Button: This button is used for When pressed, the microcontroller enters an enrollment mode and receives data (like the student's roll number) from the Bluetooth module.

**4. Bluetooth Module (HC-05)** : The Bluetooth module (HC-05) is used for wireless communication. It allows the system to receive the student’s name and roll number from a Bluetooth-enabled device (like a smartphone or a computer) for enrollment or attendance checking. During enrollment, the roll number is sent from the device to the microcontroller via Bluetooth and stored in the system's memory. e During attendance checking, the Bluetooth module receives the roll number input from the user (via a smartphone or computer) and checks whether the student is present or absent based on prior enrollments

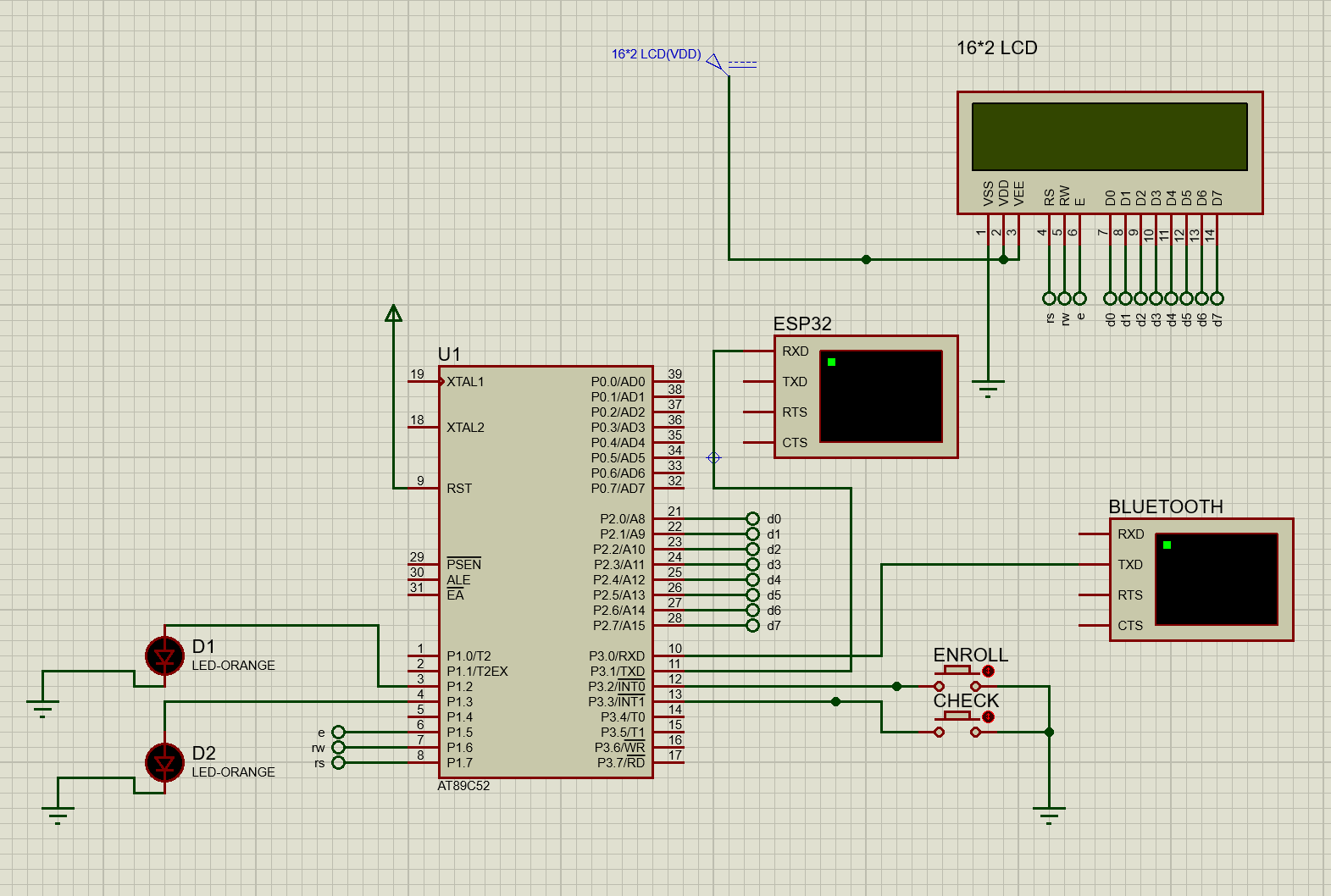
**5. Enrolling Students :** When the Enroll Button is pressed, the system enters e enrollment mode. The Bluetooth module waits for the roll number to be transmitted from the Bluetooth device. Once received, the roll number is stored in the microcontroller's memory or in external EEPROM if required for long-term storage. The LCD display shows a message confirming the successful enrollment of the student.

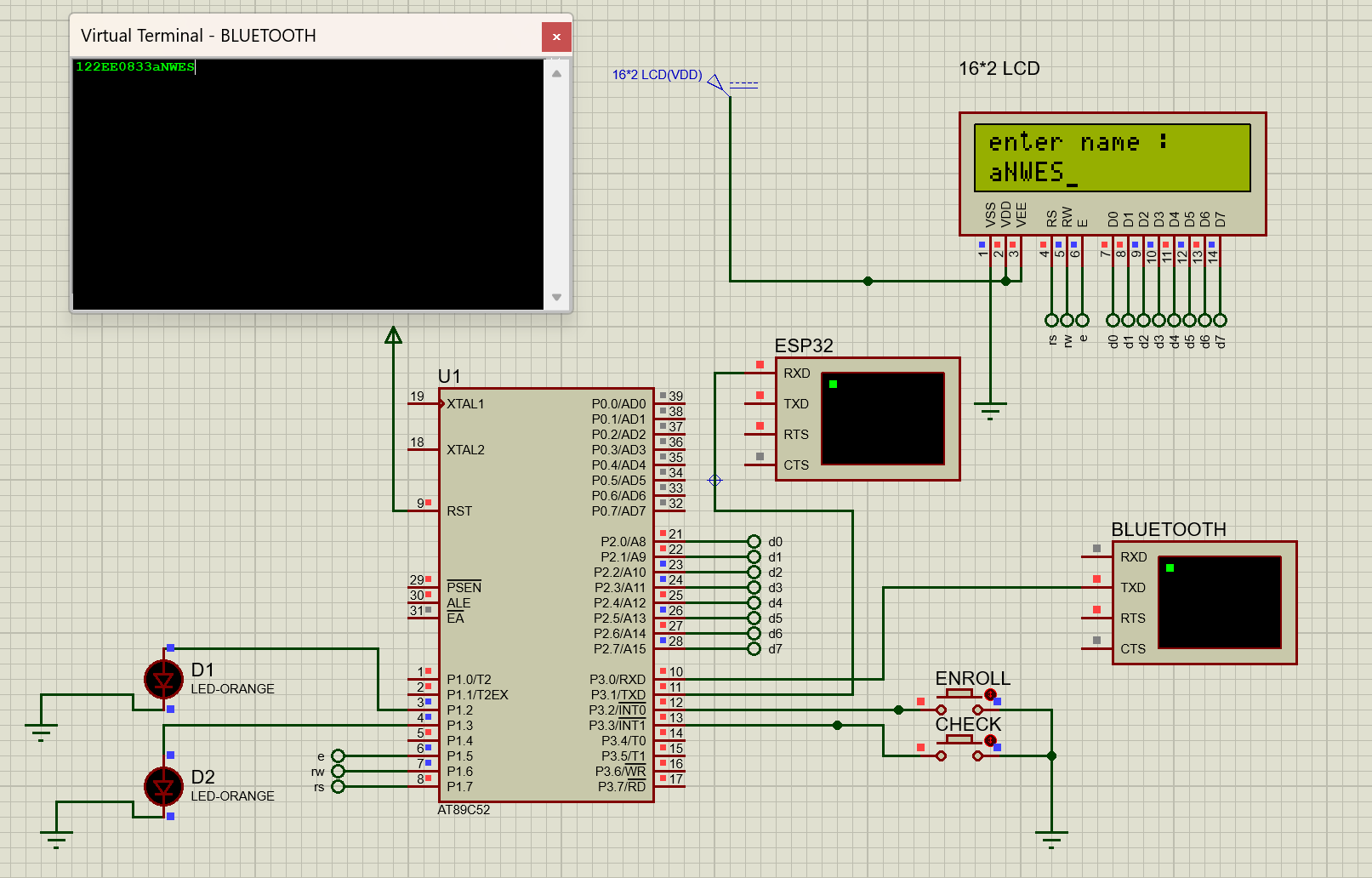
**6. Checking Attendance :** When the Check Attendance Button is pressed, the system prompts the user to input the student's roll number. The Bluetooth module receives the roll number from the device and sends it to the microcontroller The microcontroller compares the received roll number with the stored attendance records. If the student is found in the records, the LCD displays "Present"; otherwise, it shows "Absent”.

**7. Power Supply :** The entire circuit is powered using a regulated 5V DC supply. The 8051 microcontroller and other components like the LCD and Bluetooth module operate at 5V.

**8. Circuit Connections :** 8051 microcontroller is connected to the LCD display via data and control pins (RS, RW, EN, DO-D7). The push buttons are connected to the microcontroller's input pins (via pull-down resistors). The Bluetooth module is connected to the serial communication pins (TX, RX) of the 8051 microcontroller for data transmission. EEPROM (optional) can be connected to the 8051 if the attendance records are to be stored persistently

**HARDWARE DESCRIPTION**

****

****

**1.LED ( Light Emitting Diode)**

****

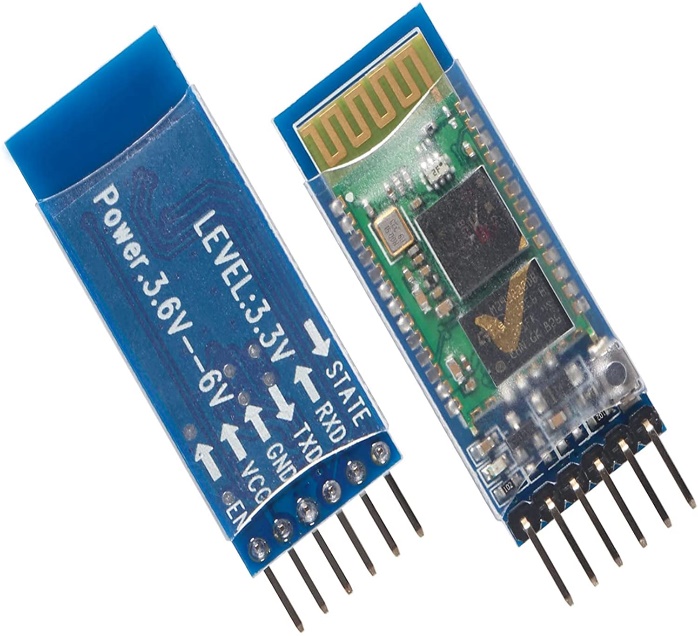
LEDs are semiconductor devices that emit light when an electric current flows through them. Known for their energy efficiency, durability, and versatility, LEDs are widely used in various applications, including as visual indicators in electronic circuits.

**2.Jumper wires**

****

Jumper wires are short, flexible wires equipped with connectors at both ends. They are essential for creating temporary connections between components on a breadboard or within a circuit. Jumper wires are available in different configurations, such as male-to-male, male-to-female, and female-to-female, to suit various prototyping and testing needs.

**3.HC-05 Bluetooth Module**

****

HC-05 is a class 2 Bluetooth module designed for transparent wireless serial communication. It is pre-configured as a slave Bluetooth device. Once it is paired to a master Bluetooth device such as PC, smart phones and tablet, its operation becomes transparent to the user.It operates at 9600 baud rate with in a power supply of 4-6 V

**4.ESP-32 WiFi Module**

****

ESP32 is a chip that provides Wi-Fi and (in some models) Bluetooth connectivity for embedded devices – in other words, for IoT devices. While ESP32 is technically just the chip, the modules and development boards that contain this chipare often also referred to as “ESP32” by the manufacturer.

**COMPONENT DESCRIPTION**

**Key Features of the Trainer Kit:**

**• Microcontroller :** AT89S52-based development board.

**• Integrated Components:**

o MAX232 for serial communication.

o On-board 16x2 alphanumeric LCD display with adjustable contrast.

o DS1307 Real-Time Clock (RTC) module.

**• Power Supply :** Includes an on-board 5V voltage regulator.

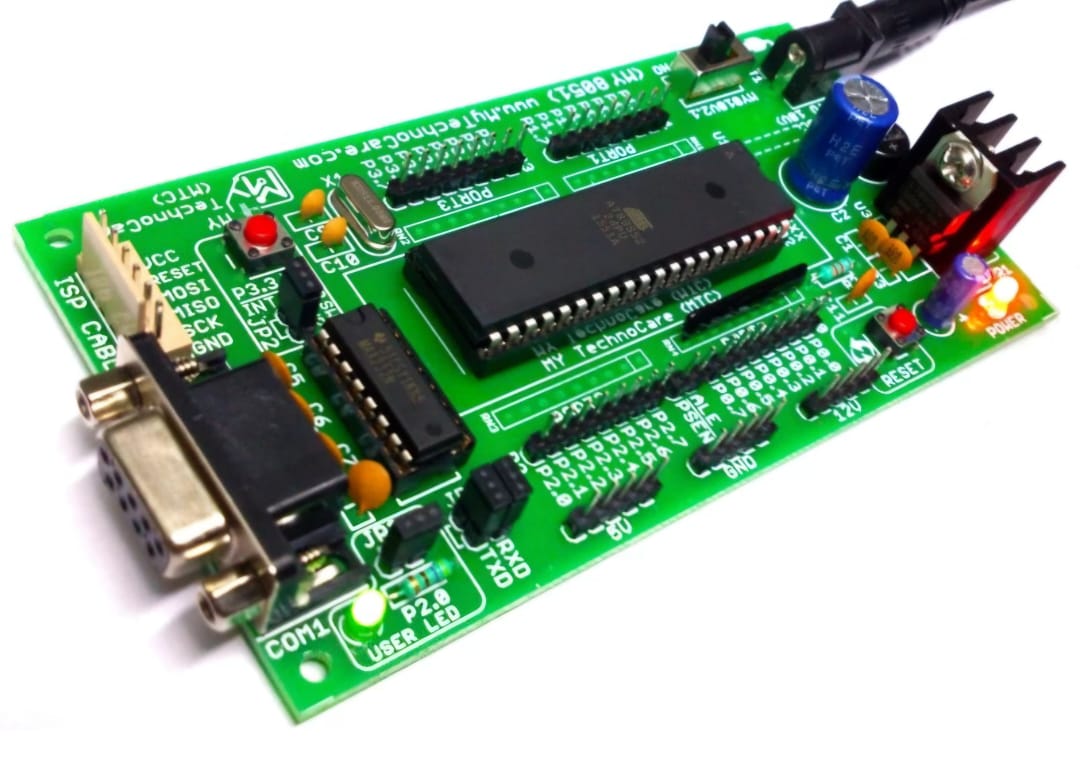
**• Additional Features:**

o Port expander for extended functionality.

o Reset button for system reinitialization.

o RS-232 communication connector for external interfacing.

**Key Features of the Trainer Kit:**



**• Microcontroller**: AT89S52-based development board.

**• Integrated Components**:

o MAX232 for serial communication.

o On-board 16x2 alphanumeric LCD display with adjustable contrast.

o DS1307 Real-Time Clock (RTC) module.

**• Power Supply**: Includes an on-board 5V voltage regulator.

**• Additional Features**:

o Port expander for extended functionality.

o Reset button for system reinitialization.

o RS-232 communication connector for external interfacing.

**CODE**

;====================================================================

; Main.asm file generated by New Project wizard

;

; Created: Tue Nov 26 2024

; Processor: AT89C52

; Compiler: ASEM-51 (Proteus)

;====================================================================

$NOMOD51

$INCLUDE (80C52.MCU)

;====================================================================

; DEFINITIONS

;====================================================================

e equ P1.5

rw equ P1.6

rs equ P1.7

dport equ P2

;====================================================================

; VARIABLES

;====================================================================

;====================================================================

; RESET and INTERRUPT VECTORS

;====================================================================

; Reset Vector

org 0000h

jmp Start

org 0003h

jmp check\_enroll

org 0013h

jmp check

;====================================================================

; CODE SEGMENT

;====================================================================

org 0100h

Start:

mov SP, #0f4h

lcall lcd\_initialisation

mov R0, #30h ;base address of database

mov IE, #95h ;serial, ext1, timer1, ext0

mov TH1, #0fdh ; for 9600 baud rate generation

mov TL1, #0fdh ; for subsequent timer generation

mov TMOD, #20h ; timer 1 in mode 2 (8-bit auto reload mode)

mov SCON, #50h ; 8-bit UART, recieve enable

setb TR1

lcall delay

Loop:

jmp Loop

;R0 - database address

;R1 - receive bytes

;R2, R3 - delay loop

;R4 - check register

;A stores no valuable information - it is variable

;cjne dest, src, addr == dest!=src go to addr, dest > src c=0, else c=1

check\_enroll:

cjne R0, #4eh, enroll

ljmp display\_storage

enroll:

jnc display\_storage

mov dptr, #lut\_enroll

lcall lcd\_display

mov A, #01h ;clear lcd screen

lcall lcd\_command

mov dptr, #lut\_enter\_roll

lcall lcd\_display

;roll no = 4 bytes store it in address database

mov R1, #04h ;number of bytes to be received

lcall receive\_bytes

mov A, #0C0h ;display to 2nd line

lcall lcd\_command

mov dptr, #lut\_enter\_name

lcall lcd\_display

;name = 2 bytes = first letter 2nd letter

mov R1, #02h

lcall receive\_bytes

;R1 will be pointing to next storage location

mov A, #01h ;clear screen

lcall lcd\_command

mov dptr, #lut\_enrolled\_success

lcall lcd\_display

mov A, #01h

lcall lcd\_command

clr RI

reti

display\_storage:

mov A, #01h

lcall lcd\_command

mov A, #"s"

lcall lcd\_data

mov A, #"t"

lcall lcd\_data

mov A, #"o"

lcall lcd\_data

mov A, #"r"

lcall lcd\_data

mov A, #"a"

lcall lcd\_data

mov A, #"g"

lcall lcd\_data

mov A, #"e"

lcall lcd\_data

mov A, #" "

lcall lcd\_data

mov A, #"f"

lcall lcd\_data

mov A, #"u"

lcall lcd\_data

mov A, #"l"

lcall lcd\_data

mov A, #"l"

lcall lcd\_data

mov A, #01h

lcall lcd\_command

reti

check:

mov A, #01h

lcall lcd\_command

mov dptr, #lut\_check

lcall lcd\_display

mov A, #01h

lcall lcd\_command

mov dptr, #lut\_enter\_roll

lcall lcd\_display

mov R1, #04h

lcall check\_bytes

clr RI

mov A, #0C0h

lcall lcd\_command

mov dptr, #lut\_checking

lcall lcd\_display

lcall attendance

reti

receive\_bytes:

clr RI

mov B, R1

receive\_loop:

wait\_rec:

jnb RI, wait\_rec

clr RI

mov A, SBUF

mov @R0, A

inc R0

lcall lcd\_data

djnz B, receive\_loop

clr RI

ret

check\_bytes:

clr RI

mov B, R1

mov 4Eh, R0

mov R0, #4fh

check\_loop:

wait\_check:

jnb RI, wait\_check

clr RI

mov A, SBUF

mov @R0, A

inc R0

lcall lcd\_data

djnz B, check\_loop

mov R0, 4Eh

mov 4Eh, #00h

clr RI

ret

attendance:

mov 4Eh, R0

mov R0, #4Fh ; Set R0 to start of buffer storage (4Fh)

mov R1, #30h ; Base address of database

mov R2, #05h ; Number of students in the database

mov A, #00h ; Clear accumulator (A) for comparison result

; Iterate through the 5 student records

loop\_students:

; Save starting address of the current student record

mov B, #04h ; Compare 4 bytes of roll number

mov A, #00h ; Reset A for XOR results

mov 55h, #00h

mov R0, #4Fh ; Set R0 to the start of the buffer

compare\_roll:

mov A, @R0 ; Get byte from buffer

xrl A, @R1 ; XOR with corresponding database byte

inc R0 ; Increment buffer pointer

inc R1 ; Increment database pointer

add A, 55h ; Accumulate XOR result

mov 55h, A ; Store intermediate result in 53h

djnz B, compare\_roll ; Repeat for all 4 bytes

;55h has sum of each student one student

; Check if the XOR result is zero

mov A, 55h ; Load the accumulated XOR result

cjne A, #00h, not\_found ; If not zero, no match, continue

lcall found\_logic

sjmp exit\_check ; Exit after finding the student

not\_found:

mov A, #00h

add A, R1

add A, #02h

mov R1, A

djnz R2, loop\_students ; Check next student if more remain

; If no match found after all iterations

lcall not\_found\_logic

exit\_check:

mov R0, 4Eh

clr RI

reti

found\_logic:

mov A, #01h

lcall lcd\_command

mov A, #"f"

lcall lcd\_data

mov A, #"o"

lcall lcd\_data

mov A, #"u"

lcall lcd\_data

mov A, #"n"

lcall lcd\_data

mov A, #"d"

lcall lcd\_data

mov A, #01h

lcall lcd\_command

ret

not\_found\_logic:

mov A, #01h

lcall lcd\_command

mov A, #"n"

lcall lcd\_data

mov A, #"o"

lcall lcd\_data

mov A, #"t"

lcall lcd\_data

mov A, #" "

lcall lcd\_data

mov A, #"f"

lcall lcd\_data

mov A, #"o"

lcall lcd\_data

mov A, #"u"

lcall lcd\_data

mov A, #"n"

lcall lcd\_data

mov A, #"d"

lcall lcd\_data

mov A, #01h

lcall lcd\_command

ret

lcd\_initialisation:

;DISPLAY IN 16\*2 LCD

MOV A, #38H ;8 BIT 2 LINE MODE

LCALL lcd\_command

MOV A, #0EH ;DISPLAY ON CURSOR ON

LCALL lcd\_command

MOV A, #01H ;CLEAR DISPLAY

LCALL lcd\_command

MOV A, #80H ;SET CURSOR AT BEGINNING OF FIRST LINE

LCALL lcd\_command

RET

lcd\_display:

mov A, #00h

movc A, @A+dptr

jz back

lcall lcd\_data

lcall delay

inc dptr

sjmp lcd\_display

ret

lut\_enroll:

db 'enroll', 00h

lut\_enter\_roll:

db 'enter roll :', 00h

lut\_enter\_name:

db 'enter name :', 00h

lut\_enrolled\_success:

db 'enrolled success', 00h

lut\_check:

db 'check', 00h

lut\_checking:

db 'checking...', 00h

lcd\_command:

CLR rs ;FOR COMMAND MODE

CLR rw ;FOR WRITE MODE

MOV dport, A

SETB e

LCALL delay

CLR e

RET

lcd\_data:

SETB rs ;FOR DATA MODE

CLR rw ;FOR WRITE MODE

MOV dport, A

SETB e

LCALL delay

CLR e

RET

;tunable delay

delay:

MOV R2, #255

MOV R3, #120

LONG\_DELAY:

DJNZ R2, LONG\_DELAY

DJNZ R3, LONG\_DELAY

RET

back:

ret

;====================================================================

END

**FLOW OF CODE**

1**. Initialization**: The program initializes the stack pointer, LCD display, and UART communication for serial data handling. Timer 1 is configured for a 9600 baud rate, and interrupt settings enable serial, external, and timer functionalities.

**2. Main Functionality:**

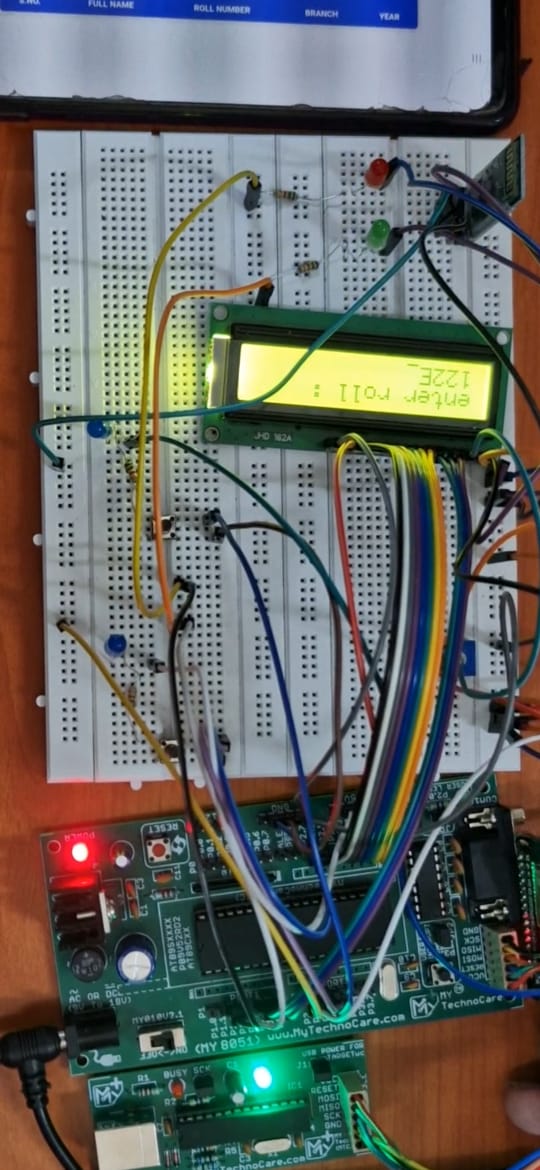
- **Enrollment:** Users can enroll by entering a 4-byte roll number and a 2-byte name. Data is stored in the database and displayed as "enrolled success".

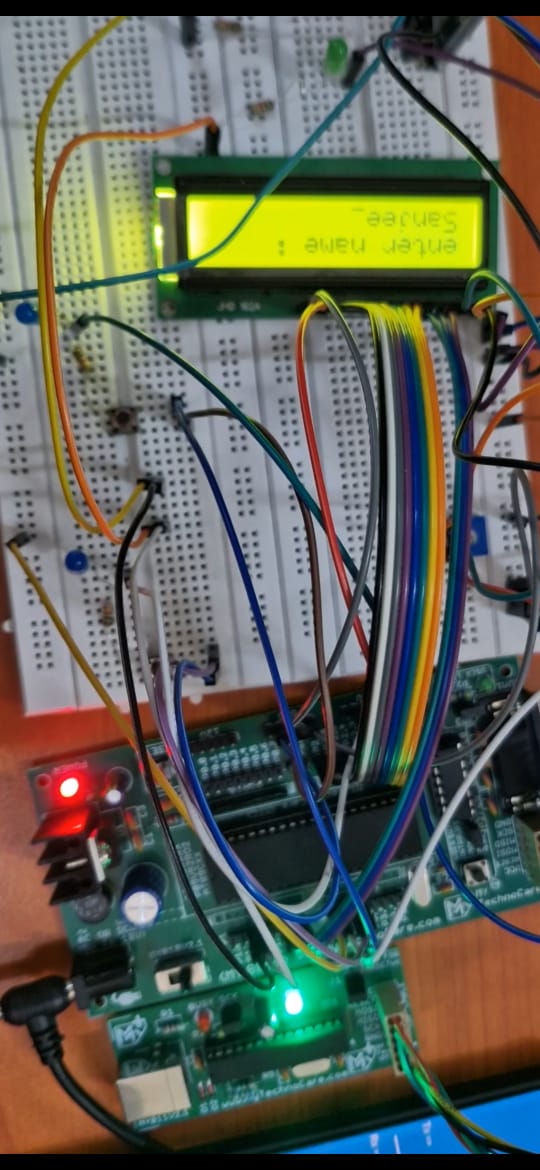
- **Attendance Check:** Input roll numbers are compared with stored database entries using XOR logic. Results ("found" or "not found") are displayed on the LCD.

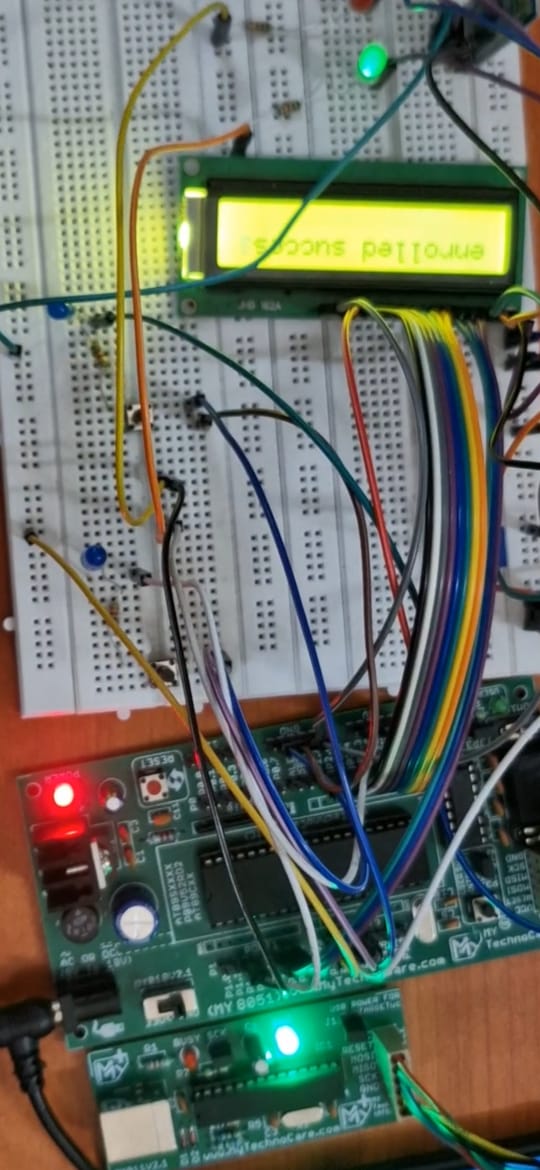
**3.LCD Handling:** Functions (lcd\_command, lcd\_data, lcd\_display) manage the LCD in 8-bit, 2-line mode to display prompts, results, and error messages using predefined strings in lookup tables.

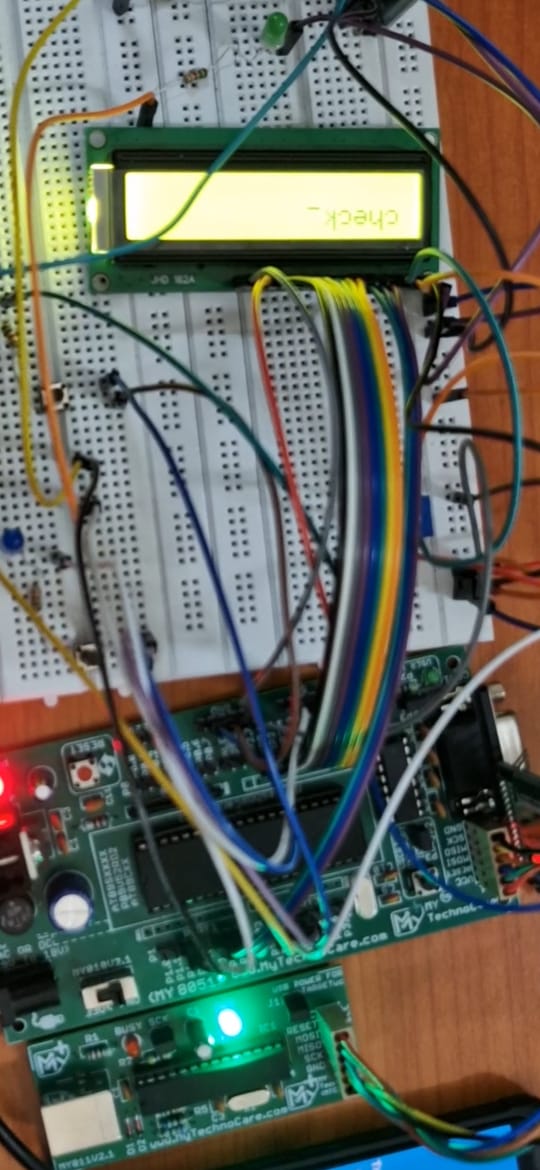
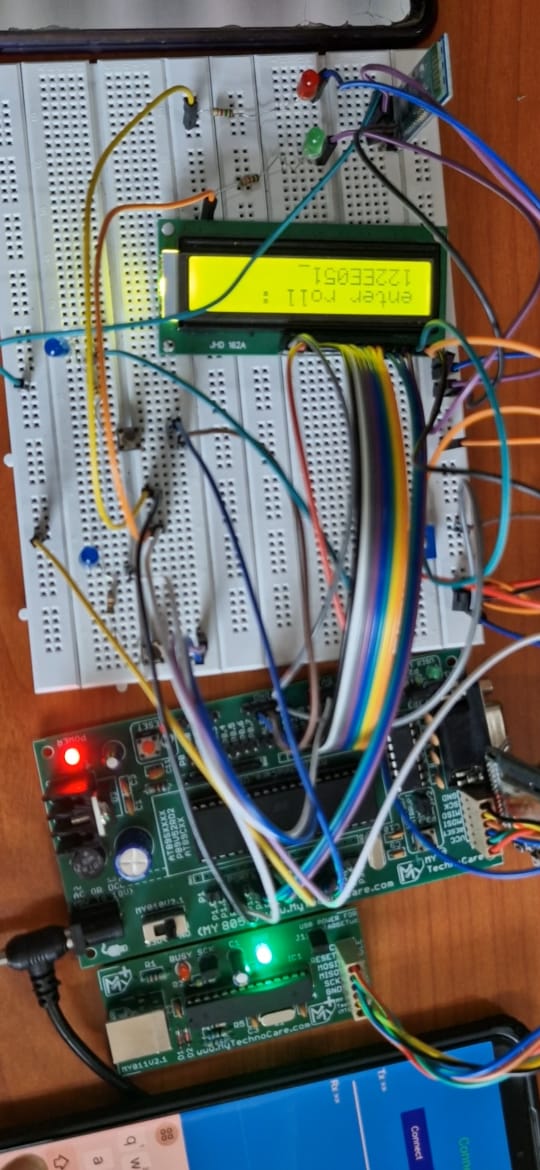
**4.UART Communication:** Data is received byte-by-byte from the serial buffer (SBUF) using receive\_bytes and verified against stored records in check\_bytes for attendance functionality.

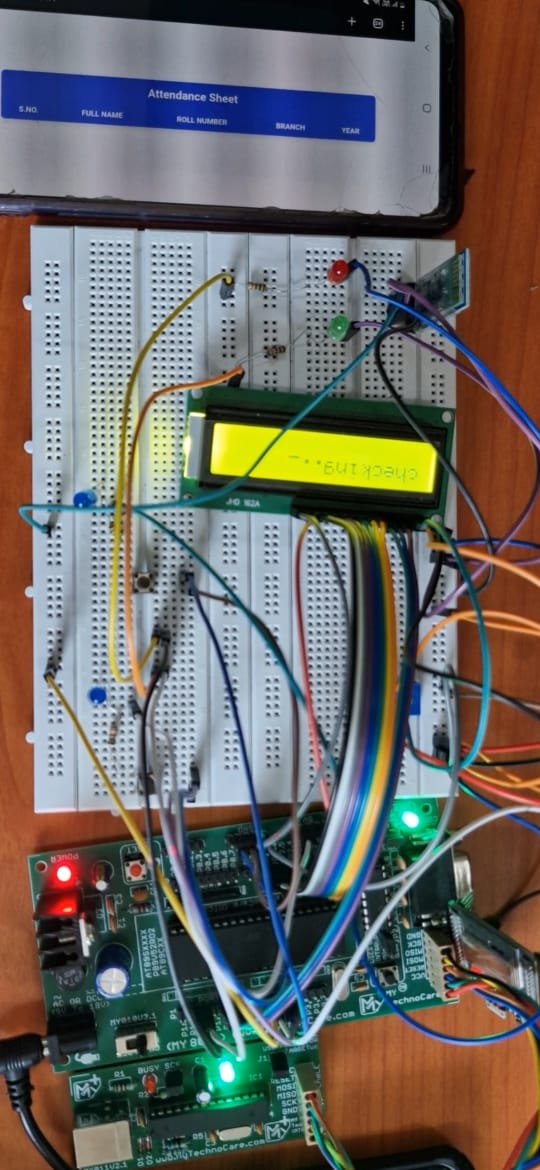
**5. Database and Logic:** The database starts at 30h, with roll numbers stored sequentially. XOR-based comparison ensures accuracy, while delay routines and loop structures manage timing and repeated checks effectively.

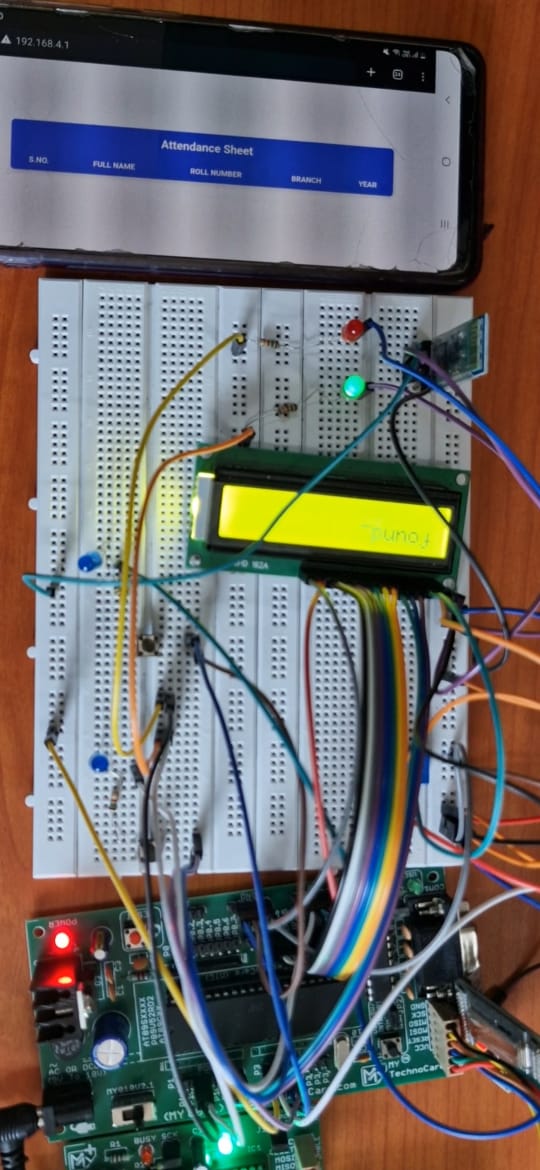
**ENROLL**



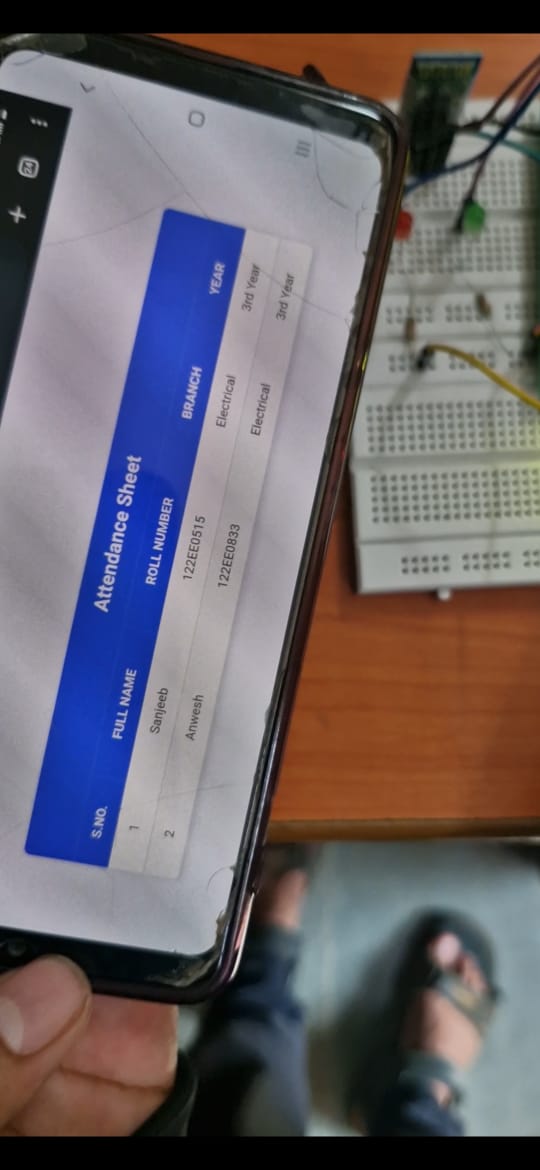


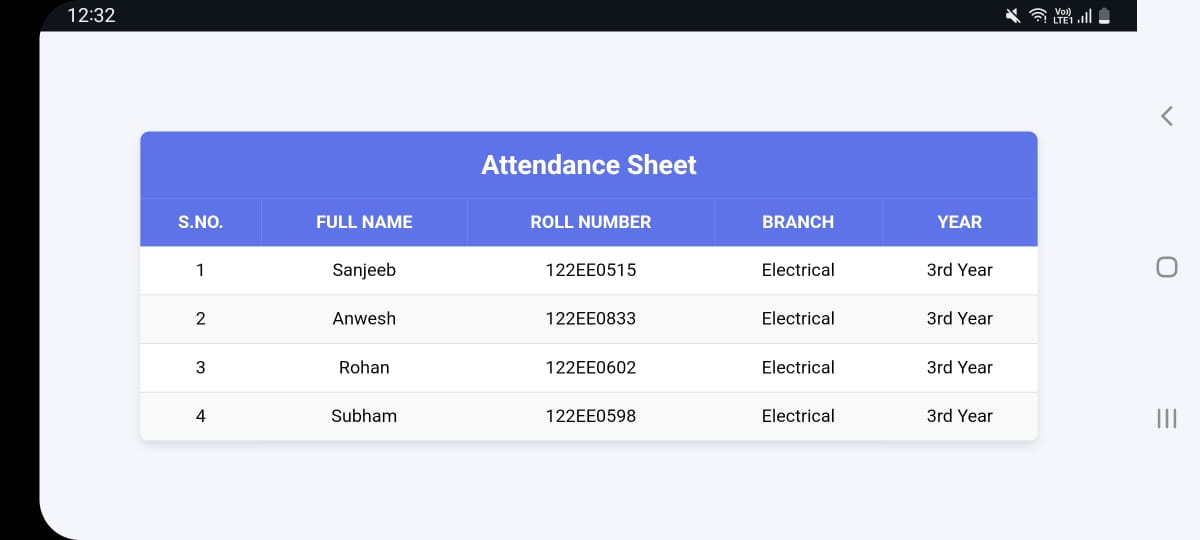
**CHECK**





**WEBPAGE**





**Issues Faced :-**

1. **LCD Interfacing Challenges:**

o Synchronizing the LCD display with LED indicators required precise code placement and proper timing delays to ensure smooth operation and no overriding.

1. **LED Voltage Compatibility:**

o The LED’s operating voltage was incompatible with the existing system, necessitating the use of resistors of 1.5k ohm to adapt for a 5V supply.

1. **Understanding fingerprint R-307 Specifications:**

o We mimicked the fingerprint operation by storing the data in 9 byte roll-number and 7 byte name format in the internal rom.

Where 9 byte roll number acts as an primary id for the database.

**Practical Applications:-**

**1. Educational Institutions**

-Classroom Attendance: Tracks student attendance automatically, saving time for teachers and ensuring accuracy.

-Remote Learning: Facial recognition systems verify students’ presence during online classes.

-Campus Security: Integrates with access control systems to monitor student movements on campus.

-Conclusion for 8051 Automated Attendance System

**2. Healthcare**

-Hospital Staff Monitoring: Ensures timely attendance of medical staff in hospitals.

- Patient Tracking: Monitors attendance of patients for scheduled check-ups or therapy sessions.

**3. Government and Public Services**

-Public Office Attendance: Monitors government employees to reduce absenteeism.

- Program Beneficiaries: Tracks attendance for public programs like skill development training.

**Benefits:**

- Saves time and reduces errors compared to manual systems.

- Improves accountability and transparency.

- Enhances security and ensures compliance with regulations.

- Provides analytics for better decision-making.

**CONCLUSION**

The **8051 Automated Attendance System** successfully streamlines the process of student attendance by combining efficient hardware components and a user-friendly interface. By utilizing an **8051 microcontroller**, **Bluetooth module**, and **LCD display**, the system automates attendance recording and retrieval, eliminating the need for manual roll calls. The use of push buttons for enrollment and attendance checking makes the system simple to operate, while Bluetooth connectivity allows for easy wireless interaction with the system, enhancing flexibility. The system's ability to store student data and provide real-time attendance updates contributes to increased accuracy and efficiency in attendance management. Overall, this project not only reduces administrative workload but also encourages a more organized and accessible approach to managing attendance in educational settings.

* **Sanjeeb Dash ( 122EE0515 )**
* **Anwesh Panda ( 122EE0833 )**
* **Subham Shibasis Ghose ( 122EE0598 )**