```
#include <SPI.h>
#include <SD.h>
#include <LiquidCrystal_I2C.h>
#include <avr/wdt.h>
#define BORROW_BTN_PIN 2
#define RETURN_BTN_PIN 3
#define CONTINUE_BTN_PIN 4
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Adjust the I2C address if needed
File file;
const char* filename1 = "books.csv";
const char *filename2 = "students.csv";

const int chipSelect = 9;  // Assuming your SD card is connected to pin 9
void setup() {
  Serial.begin(9600);
  pinMode(BORROW_BTN_PIN, INPUT);
  pinMode(RETURN_BTN_PIN, INPUT);
  pinMode(CONTINUE_BTN_PIN, INPUT);
  pinMode(2, INPUT); // Set pin 2 as input for EM-18 RFID reader module
  lcd.begin(16, 2);
  lcd.init();
  lcd.clear();
  lcd.backlight();      // Make sure backlight is on
  lcd.print("E&TC Library");
  lcd.setCursor(0, 1);
  lcd.print("Device 2024");
  if (!SD.begin(chipSelect)) {
    Serial.println("SD card initialization failed!");
    return;
  }
}
void loop() {
  if (digitalRead(BORROW_BTN_PIN) == HIGH) {
    Borrow();
  }

  if (digitalRead(RETURN_BTN_PIN) == HIGH) {
    Return();
  }

  delay(100);  // Adjust the delay as needed
}
void Borrow() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scanning QR Code");
  lcd.setCursor(0, 1);
  lcd.print("For Borrow Book");
```

```
  delay(5000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Press Continue button");
  while (digitalRead(CONTINUE_BTN_PIN) == LOW) {
    delay(50);
  }
  Serial.println("start the camera\n");
  // here write capture qr decoding coding part
  char camData[] = "1003"; // coming from the camera module using serial
communication
  String bookID = String(camData);
  String bookName = searchBook(bookID);
  if (bookName.length() > 0) {
    Serial.print("Book Name found: ");
    Serial.println(bookName);
  } else {
    Serial.println("Book ID not found.");
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(bookID);
  lcd.setCursor(0, 1);
  lcd.print(bookName);
  delay(5000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Press Continue button");
  while (digitalRead(CONTINUE_BTN_PIN) == LOW) {
    delay(50);
  }
  lcd.setCursor(0, 1);
  lcd.print("Tap RFID Card");
  delay(5000);
  String studentName = ""; // Declare studentName here
  String content = readRFID();
  if (content != "") {
    Serial.println("UID tag: " + content);
    studentName = searchStudent(content);
    if (studentName != "") {
      Serial.println("Student Name: " + studentName);
    } else {
      Serial.println("Student not found.");
    }
  }
  delay(1000); // Delay to avoid reading the same card multiple times
  lcd.clear();
  lcd.setCursor(0, 0);
```

```
  lcd.print(studentName);
  lcd.setCursor(0, 1);
  lcd.print(bookName);
  delay(5000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Press Continue button");
  while (digitalRead(CONTINUE_BTN_PIN) == LOW) {
    delay(50);
  }
  SendSMS();
  // reset the nano automatically
  wdt_enable(WDTO_15MS);
  while (1) {}
}
void Return() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Scanning QR Code");
  lcd.setCursor(0, 1);
  lcd.print("For Return Book");
  delay(5000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Press Continue button");
  while (digitalRead(CONTINUE_BTN_PIN) == LOW) {
    delay(50);
  }
  Serial.println("start the camera\n");
  // here write capture qr decoding coding part
  char camData[] = "1003"; // coming from the camera module using serial
communication
  String bookID = String(camData);
  String bookName = searchBook(bookID);
  if (bookName.length() > 0) {
    Serial.print("Book Name found: ");
    Serial.println(bookName);
  } else {
    Serial.println("Book ID not found.");
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(bookID);
  lcd.setCursor(0, 1);
  lcd.print(bookName);
  delay(5000);
  lcd.clear();
  lcd.setCursor(0, 0);
```

```
  lcd.print("Press Continue button");
  while (digitalRead(CONTINUE_BTN_PIN) == LOW) {
    delay(50);
  }
  lcd.setCursor(0, 1);
  lcd.print("Tap RFID Card");
  delay(5000);
  String studentName = ""; // Declare studentName here
  String content = readRFID();
  if (content != "") {
    Serial.println("UID tag: " + content);
    studentName = searchStudent(content);
    if (studentName != "") {
      Serial.println("Student Name: " + studentName);
    } else {
      Serial.println("Student not found.");
    }
  }
  delay(1000); // Delay to avoid reading the same card multiple times
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(studentName);
  lcd.setCursor(0, 1);
  lcd.print(bookName);
  delay(5000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Press Continue button");
  while (digitalRead(CONTINUE_BTN_PIN) == LOW) {
    delay(50);
  }
  SendSMS();
  // reset the nano automatically
  wdt_enable(WDTO_15MS);
  while (1) {}
}
String searchBook(const String& bookID) {
  file = SD.open(filename1);
  if (file) {
    while (file.available()) {
      String row = file.readStringUntil('\n');
      // Split the row into two parts using ","
      int splitIndex = row.indexOf(',');
      String currentBookID = row.substring(0, splitIndex);
      String bookName = row.substring(splitIndex + 1);

      // Remove leading and trailing whitespaces
      currentBookID.trim();
```

```
      bookName.trim();

      if (currentBookID.equals(bookID)) {
        file.close();
        return bookName;
      }
    }
    file.close();
  } else {
    Serial.println("Error opening file!");
  }

  return ""; // Return an empty string if book ID is not found
}
String readRFID() {
  String content = "";
  if (Serial.available()) {
    char receivedChar = Serial.read();
    content += receivedChar; // Append the received character to the string
    // Display the received data on the LCD
    lcd.print(receivedChar);
    // Check if the received character is a newline character
    // if (receivedChar == '\n') {
    //   // Print the complete received data to the Serial monitor
    //   Serial.print("Received data: ");
    //   Serial.println(receivedData);
    //   receivedData = ""; // Clear the received data for the next input
    //   lcd.clear(); // Clear the LCD for the next input
    //   lcd.print("Received:");
    //   lcd.setCursor(0, 1);
    // }
  }
  return content;
}
String searchStudent(String id) {
  file = SD.open(filename2);
  if (file) {
    while (file.available()) {
      String line = file.readStringUntil('\n');
      int commaIndex = line.indexOf(',');
      String studentID = line.substring(0, commaIndex);
      String studentName = line.substring(commaIndex + 1);
      if (studentID == id) {
        file.close();
        return studentName;
      }
    }
    file.close();
```

```
  } else {
    Serial.println("Error opening file!");
  }
  return ""; // Return empty string if student not found
}
void SendSMS(){
  while (digitalRead(CONTINUE_BTN_PIN) == LOW) {
    delay(50);
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Sending SMS");
  lcd.setCursor(0, 1);
  lcd.print("-");
  delay(1000);
  lcd.print("--");
  delay(1000);
  lcd.print("---");
  delay(1000);
  lcd.print("----");
  delay(1000);
  lcd.print("-----");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Thank You");
  lcd.setCursor(0, 1);
  lcd.print("For Using Device");
  Serial.println("Sending SMS");
  delay(1000);
}
```