Now let's test your circuit in **Wokwi**.

□Project 1:- How to Test Your LED Blink

□ Step 1: Check Your Code

Make sure your `sketch.ino` has this:

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

□ Step 2: Check Wiring

Make sure connections are:

- Pin 13 → Resistor
- Resistor → LED long leg (+)
- LED short leg (−) → GND

Resistor can be 220Ω or 330Ω.

□ Step 3: Start Simulation

Click the green ▶ **Start Simulation** button at the top.

□ What Should Happen?

- LED turns ON for 1 second
- LED turns OFF for 1 second

- Keeps repeating

If this happens your project works!

□ If It Does NOT Work

Check these common mistakes:

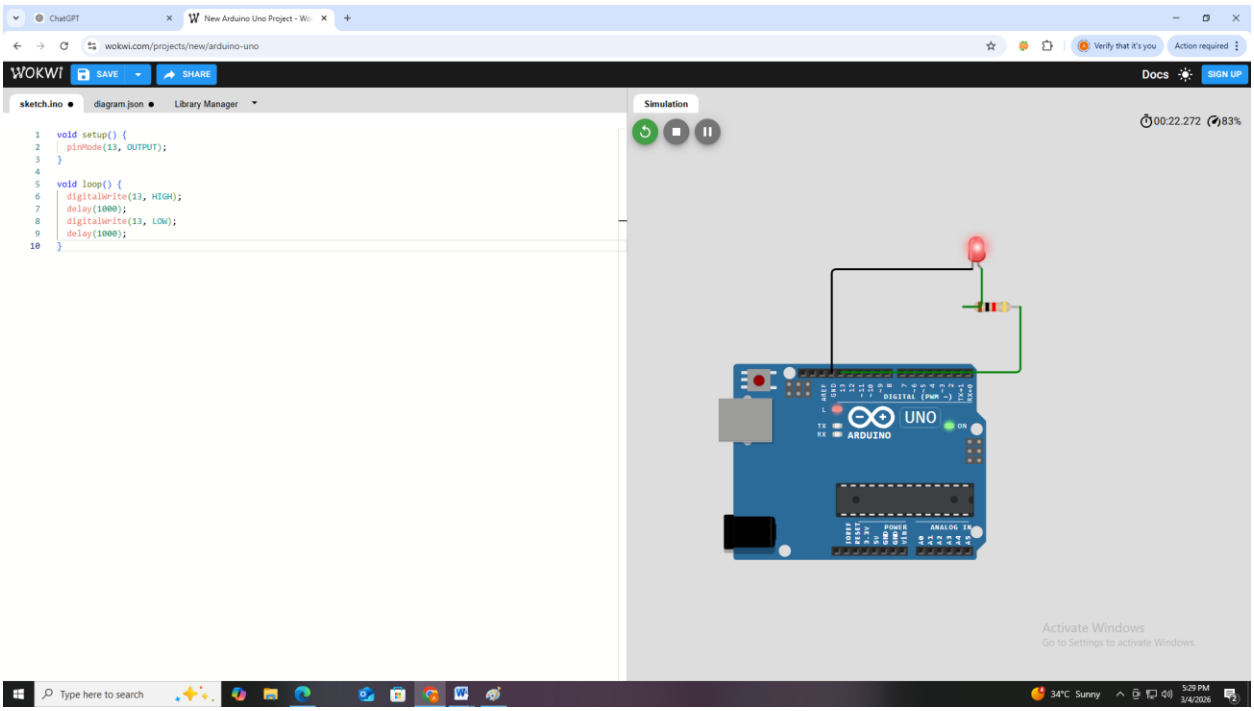
1. LED reversed (swap legs)
2. Not connected to GND
3. Wire not connected properly
4. Wrong pin number in code
5. Forgot to press Start Simulation

□ Quick Test Trick

Change delay to 200 ms to see faster blinking:

```
delay(200);
```

If LED blinks fast → circuit is correct.



Project 2:-

With lcd

Let’s make a **complete step-by-step guide** for your project: pressing a button (or keyboard key in Wokwi) displays a message on the LCD. I’ll cover **all the hardware and software setup in Wokwi**.

Step-by-Step Guide: Arduino Uno + LCD + Button Simulation in Wokwi

---

Step 1: Open Wokwi and Start a New Project

1. Go to https://wokwi.com/

2. Click **“Start a New Project”** → select **Arduino Uno**
3. The simulator workspace opens with a virtual Arduino Uno board.

Step 2: Add Components

In the Wokwi simulator:

1. Click the **“+” button** to add components.
2. Add the following:
    - **LCD 16x2 (LiquidCrystal)**
    - **Pushbutton** (this will simulate the `1` key)
    - **Potentiometer** (optional, used to adjust LCD contrast)

Step 3: Wire the LCD

Connect the LCD pins to the Arduino Uno like this:

LCD Pin	Arduino Pin / Notes
VSS	GND
VDD	5V
V0	Middle pin of potentiometer (for contrast)
RS	12
RW	GND
E	11
D4	5
D5	4
D6	3
D7	2
A (LED+)	5V
K (LED-)	GND

- Connect potentiometer sides to **5V and GND**.
- Middle pin → **V0** on the LCD.
- Turning the knob adjusts contrast.

 If you don’t add a potentiometer, you can connect **V0 directly to GND** for basic visibility.

Step 4: Wire the Pushbutton

1. Connect **one button pin → Arduino pin 7**
2. Connect **other pin → GND**
3. In code, enable **internal pull-up**, so the button reads `LOW` when pressed.

Step 5: Assign Keyboard Key in Wokwi (Optional)

- Click the pushbutton in Wokwi → in **Properties**, assign **Keyboard Key → 1**
- This lets you press **'1'** on your PC keyboard to simulate the button.

Step 6: Add Arduino Code

Copy this code into Wokwi:

```
#include <LiquidCrystal.h>

// Initialize the LCD: RS, E, D4, D5, D6, D7
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int buttonPin = 7; // Button for '1'
int buttonState = 0;

void setup() {
  lcd.begin(16, 2);           // LCD columns and rows
  lcd.print("Press 1 key");   // initial message
  pinMode(buttonPin, INPUT_PULLUP); // internal pull-up
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == LOW) {  // button pressed
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Button pressed");
    lcd.setCursor(0, 1);
    lcd.print("Value: 1");
    delay(200);               // debounce
  }
}
```

 This will display:

```
Button pressed
Value: 1
```

when the button is pressed.

Step 7: Run the Simulation

1. Click **“Start Simulation”**
2. Press the **‘1’ key** on your keyboard (or click the pushbutton in Wokwi)
3. The LCD will update and display your message.

Step 8: Optional Enhancements

- Add more buttons (for `2`, `3`, etc.) and map keyboard keys.
- Expand the code to detect which button is pressed and display it dynamically.
- Adjust the potentiometer in Wokwi to make the LCD text more readable.