Secure Distance Limitation Project’s Code:

```
// Pin Definitions
const int trigPin = 9;
const int echoPin = 10;
const int buzzerPin = 11;
const int ledPins[] = {2, 3, 4, 5, 6, 7, 8};

// Distance thresholds (in centimeters)
const int thresholds[] = {50, 40, 30, 20, 15, 10, 5}; // Thresholds for LEDs
const int buzzerThreshold = 3; // Buzzer activates at 3 cm

// Function to measure distance
long measureDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH);
  return duration * 0.034 / 2; // Convert to cm
}

void setup() {
  // Initialize pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);

  for (int i = 0; i < 7; i++) {
    pinMode(ledPins[i], OUTPUT);
  }

  Serial.begin(9600); // For debugging
}

void loop() {
  long distance = measureDistance();
  Serial.println(distance); // Print distance for monitoring

  // Control LEDs based on distance
  for (int i = 0; i < 7; i++) {
    if (distance <= thresholds[i]) {
      digitalWrite(ledPins[i], HIGH);
    } else {
      digitalWrite(ledPins[i], LOW);
    }
  }

  // Activate buzzer if distance is less than buzzerThreshold
  if (distance <= buzzerThreshold) {
    digitalWrite(buzzerPin, HIGH);
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(100); // Small delay for stability
}
```