

# National Textile University, Faisalabad



## Department of Computer Science

<b>Name</b>	Sarwat Fatima
<b>Class</b>	BSCS – 5 <sup>th</sup> – A
<b>Section</b>	5 <sup>th</sup> -A
<b>Registration No</b>	23-NTU-CS-1092
<b>Course Name</b>	Embedded IoT systems
<b>Submitted To</b>	Sir Nasir Mehmood
<b>Submission Date</b>	05/01/2025

## Smart Classroom Project Documentation

This project implements a **Smart Classroom** monitoring system using an **ESP32**, PIR and LDR sensors, Blynk IoT platform, and a custom web dashboard. It enables real-time tracking of student presence, presentation engagement, and performance feedback in a classroom environment, helping teachers to evaluate and record student activity efficiently.

## Overview

The code combines sensor data, Blynk virtual widgets, and a built-in web server to deliver a comprehensive view of classroom dynamics. It connects to WiFi, reads sensor inputs to gauge student activity and engagement, updates a mobile Blynk dashboard, and hosts a web page with analytics and feedback for each student.

## Key Features

- Student Activity Monitoring:** Detects presence and presentation activity using PIR and LDR sensors.
- Blynk Integration:** Visualizes and controls session data via a mobile app.
- Feedback & Analytics:** Generates feedback and engagement scores per student.
- Web Dashboard:** Custom web page showing real-time stats, feedback, and charts.

## Main Components & Architecture

---

- **ESP32**: Executes the main logic and hosts the web
- **PIR** : Detects motion (student presence)
- **LDR** : Detects light changes (projector state)
- **Blynk** : Provides virtual widgets and mobile
- **WebServer**: Hosts a dashboard with a table, analytics, and Chart.js

## Project Structure

Section	Purpose
WiFi Setup	Connects ESP32 to local WiFi
Sensor Reading	Reads PIR and LDR, calculates presence and presentation
Student Management	Selects and updates current student info
Feedback Generation	Provides automated feedback based on performance
Web Dashboard	Displays real-time data and analytics
Blynk Callbacks	Handles UI interactions and data saving

## Global Variables and Configuration

```
#define BLYNK_TEMPLATE_ID "TMPL6bnrG4pnK"
#define BLYNK_TEMPLATE_NAME "Smart ClassRoom"
#define BLYNK_AUTH_TOKEN "S6ratnohzoGe9T94WaLasaZ2I6pIge4s"

#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <WebServer.h>

char ssid[] = "Haru";
char pass[] = "Haruki12";

// ===== PINS =====
const int pirPin = 34;
const int ldrPin = 35;

// ===== VIRTUAL PINS =====
#define V_SELECT_STUD    V2
#define V_ACTIVE_STUD    V3
#define V_STUDENT_PIR    V5
#define V_PROJECTOR      V6
#define V_PRES_TIME      V10
#define V_ATTENDANCE     V12
#define V_SCORE           V15
#define V_REMARKS          V16
#define V_SAVE             V17
#define V_SAVED_SCORE     V18
#define V_SAVED_REMARKS   V19

WebServer server(80);
BlynkTimer timer;

// ===== STUDENT DATA =====
struct StudentData {
    String name;
    int presPercent;        // Presentation timing %
    int presencePercent;   // Attendance %
}
```

```

    int score;
    String remarks;
};

StudentData students[] = {
    {"Haruki", 0, 0, 0, ""},
    {"Mano", 0, 0, 0, ""},
    {"Ivana", 0, 0, 0, ""},
    {"Amna", 0, 0, 0, ""}
};

```

## Sensor Reading and Engagement Calculation

The system continuously reads inputs from the PIR and LDR sensors:

- **PIR Sensor:** Detects student motion, marking them present.
- **LDR Sensor:** Detects when the projector is ON (increased light level).
- **Presence Percent:** Calculated by time spent detected by PIR.
- Presentation Percent:** Calculated by time projector is ON.

### Engagement Calculation Algorithm:

```

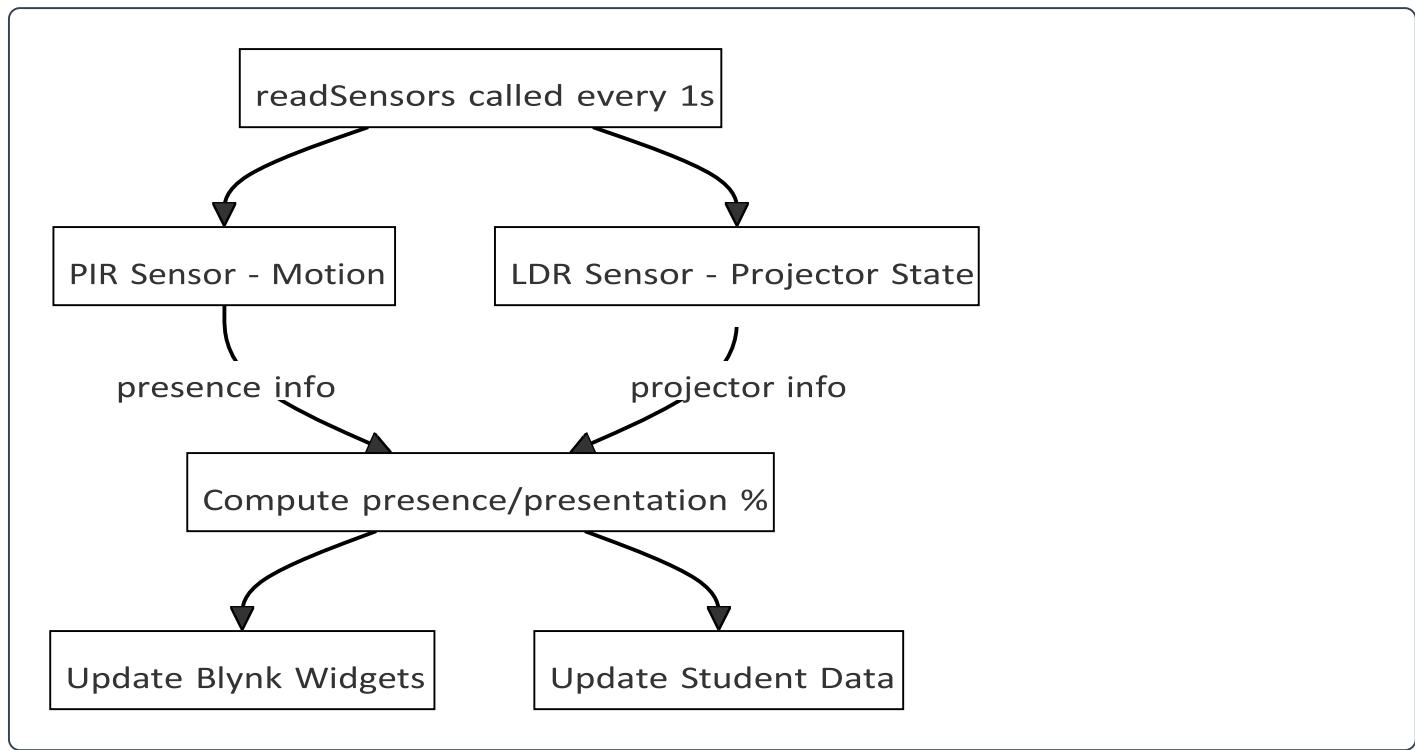
int calculateEngagement(StudentData s){
    int baseScore = (s.presencePercent * 0.4) + (s.presentPercent * 0.6);
    if(s.remarks.indexOf("participat") >= 0) baseScore += 10;
    if(s.presencePercent < 50) baseScore -= 15;
    if(baseScore < 0) baseScore = 0;
    if(baseScore > 100) baseScore = 100;
    return baseScore;
}

```

- **Weighting:** 40% attendance, 60% presentation.
- **Bonus:** +10 if remarks mention 'participation'.
- **Penalty:** -15 if presence < 50%.

**Clamped:** 0–100%.

## Sensor Data Update Flow



## Blynk Virtual Pins

Virtual Pin	Purpose
V2	Student selection dropdown
V3	Displays current student name
V5	PIR sensor status
V6	Projector state
V10	Presentation time %
V12	Attendance %
V15	Score input
V16	Remarks input
V17	Save button
V18	Saved score display
V19	Saved remarks display

## Student Selection, Scoring, and Saving

- **Score/Remarks:** Updates student struct fields.
- **Save:** Stores results and ends session.

## Feedback Generation

---

Feedback is dynamically generated for each student based on their attendance, presentation, and remarks:

```
String generateFeedback(StudentData s) {
    String feedback = "<ul class='feedback-list'>";

    // Attendance feedback
    if(s.presencePercent >= 15) {
        feedback += "<li> Excellent class participation!</li>";
    } else if(s.presencePercent >= 10) {
        feedback += "<li> Good engagement, try to participate more</li>";
    } else {
        feedback += "<li> Needs to increase participation</li>";
    }

    // Presentation feedback
    if(s.presPercent >= 15) {
        feedback += "<li> Outstanding presentation skills</li>";
    } else if(s.presPercent >= 10) {
        feedback += "<li> Solid presentation delivery</li>";
    } else if(s.presPercent > 0) {
        feedback += "<li> Practice presentation timing</li>";
    }

    // Custom remarks
    if(s.remarks.length() > 0) {
        feedback += "<li> Teacher note: " + s.remarks + "</li>";
    }

    feedback += "</ul>";
    return feedback;
}
```

## Web Dashboard

---

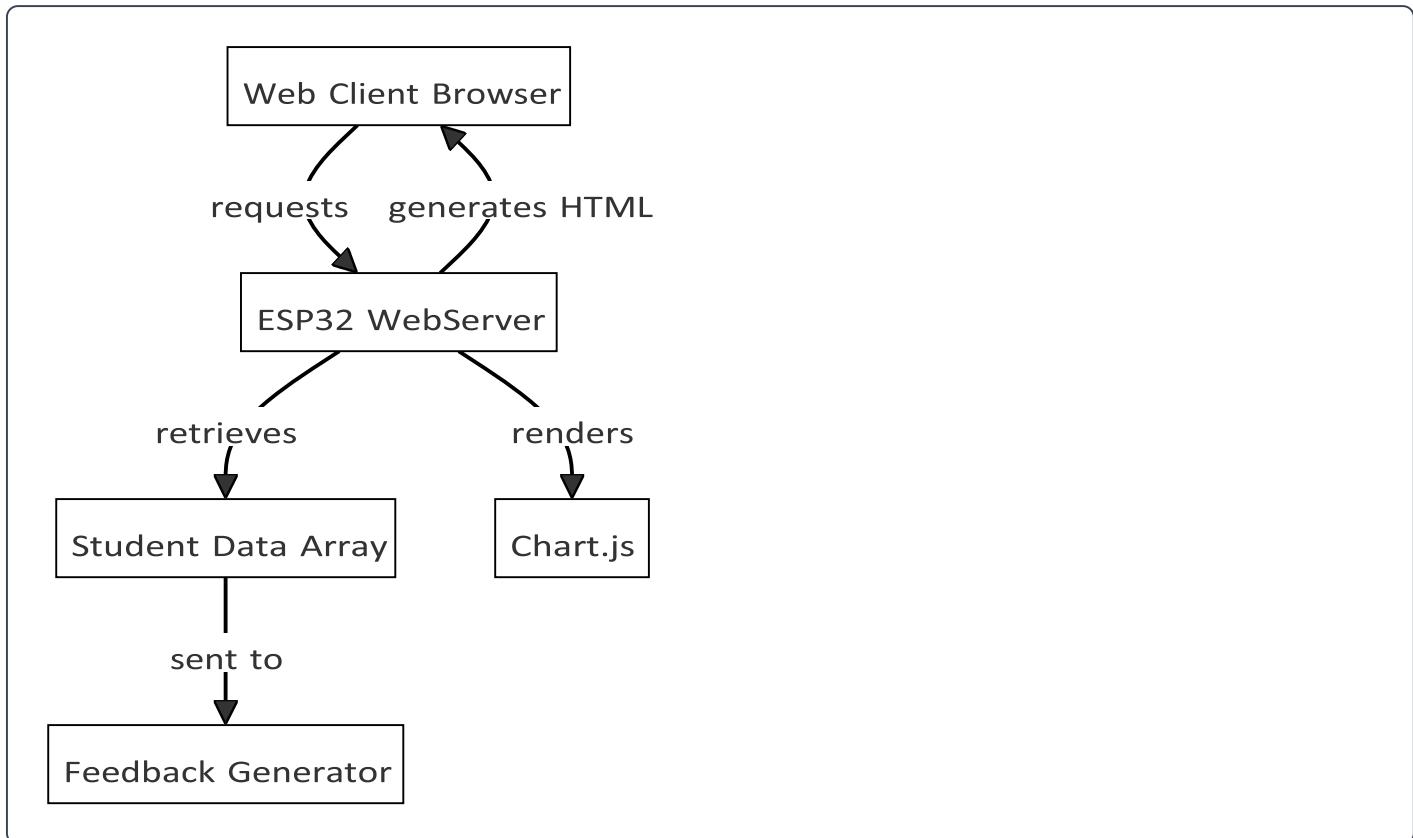
The ESP32 hosts a web server at its local IP, accessible over LAN.

### Web Page Features

- **Student Overview Table:** Displays all students with attendance, presentation, scores, remarks, and feedback.
- **Color Coding:** Attendance and presentation cells color-coded for quick status.
- **Teacher Notes:** Highlights the top and lowest engaged students.

**Chart.js Visualization:** Bar chart compares engagement scores.

## Web Dashboard Layout



## Code Flow Summary

- 1. Setup:** Initializes pins, Blynk, WiFi, timers, and web server.
- 2. Loop:** Handles Blynk UI, sensor timer, and web server requests.
- 3. Sensor Timer:** Reads sensors, updates student/session data, pushes to Blynk.
- 4. Blynk Handlers:** Receive UI changes (student, score, remarks, save).
- 5. WebServer Handler:** Generates styled HTML dashboard with live data and chart.

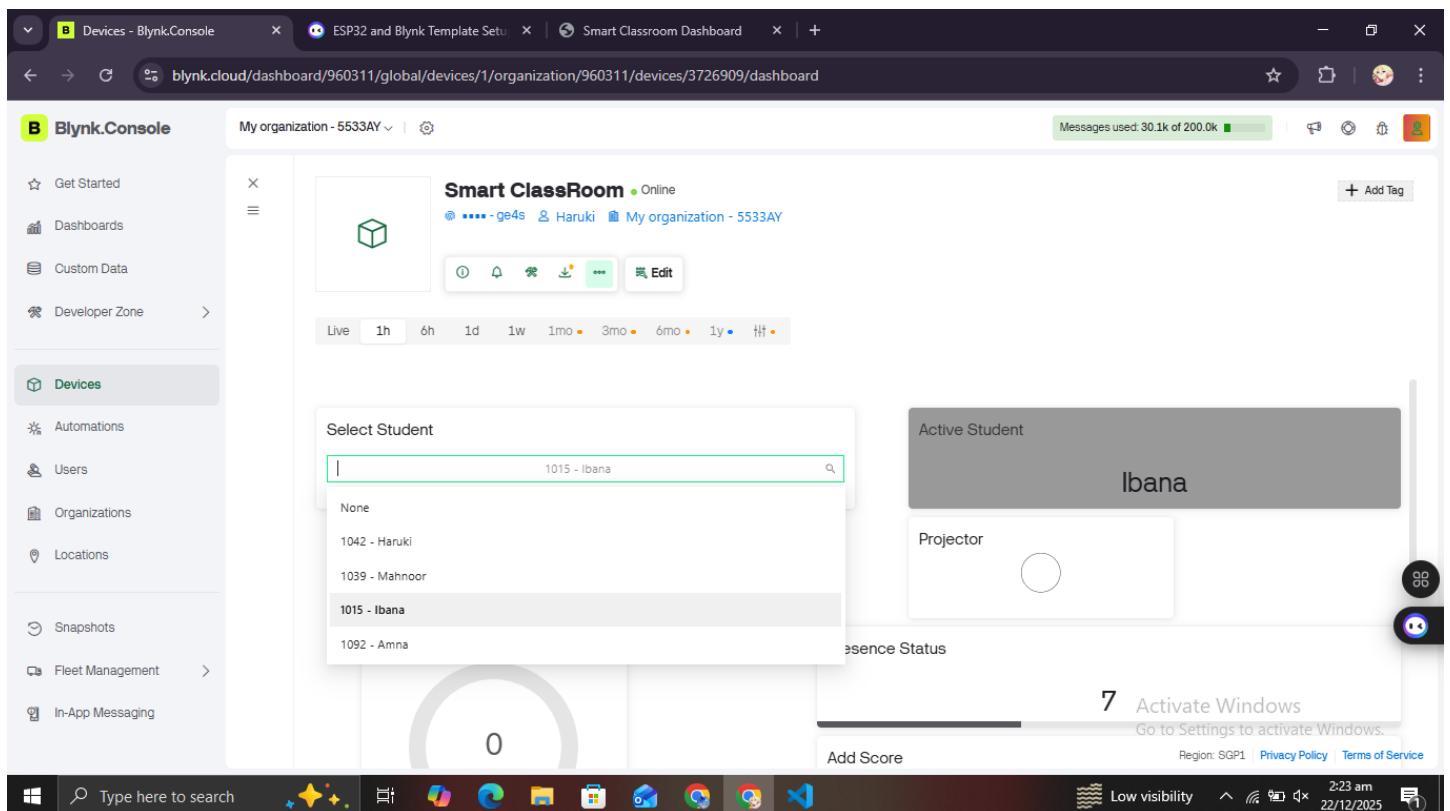
## Setup and Deployment

- 1. Connect Sensors:** Attach PIR to pin 34, LDR to pin 35.
- 2. Configure WiFi:** Update `ssid` and `pass` in the code.
- 3. PlatformIO:** Upload firmware to ESP32 via VSCode.
- 4. Blynk:** Use credentials and set up widgets as specified.
- 5. Access Web Dashboard:** Open `http://<ESP32-IP>` in browser on same network.

## Conclusion

This Smart Classroom project provides an interactive, sensor-driven evaluation system for teachers, combining mobile and web dashboards for optimal accessibility. The code can be customized for different environments by adjusting student data, sensors, and UI.

ScreenShots:



Screenshot of the Blynk Cloud Dashboard for a Smart Classroom application.

The dashboard title is "Smart ClassRoom" and it shows "Online".

Key components include:

- A "Select Student" section showing "1015 - Ibana".
- An "Active Student" section showing "Ibana".
- A "Student PIR" sensor icon.
- A "Presentation Time" gauge meter set at 0.
- A "Presence Status" progress bar set at 8.
- A "Projector" sensor icon.
- A "Presence Status" progress bar set at 0.

The Blynk sidebar on the left lists:

- Devices (selected)
- Automations
- Users
- Organizations
- Locations
- Snapshots
- Fleet Management
- In-App Messaging

The Windows taskbar at the bottom shows various pinned apps like File Explorer, Edge, and File History.

Screenshot of the Blynk Cloud Dashboard for a Smart Classroom application, showing updated data.

The dashboard title is "Smart ClassRoom" and it shows "Online".

Key components include:

- A "Presentation Time" gauge meter set at 0.
- A "Presence Status" progress bar set at 8.
- A "Add Score" input field with a value of 0.
- An "Activate Windows" message: "8 Activate Windows Go to Settings to activate Windows." (Region: SGP1 Privacy Policy Terms of Service)

The Blynk sidebar on the left lists:

- Devices (selected)
- Automations
- Users
- Organizations
- Locations
- Snapshots
- Fleet Management
- In-App Messaging

The Windows taskbar at the bottom shows various pinned apps like File Explorer, Edge, and File History.

Devices - Blynk.Console    ESP32 and Blynk Template Setup    Smart Classroom Dashboard

blynk.cloud/dashboard/960311/global/devices/1/organization/960311/devices/3726909/dashboard

**Blynk.Console** My organization - 5533AY

Get Started Dashboards Custom Data Developer Zone

Devices Automations Users Organizations Locations

Snapshots Fleet Management In-App Messaging

Smart ClassRoom Online

ge4s Haruki My organization - 5533AY

Live 1h 6h 1d 1w 1mo 3mo 6mo 1y

Type here

SAVE

Saved Score 89

Saved Remarks

Activate Windows Go to Settings to activate Windows.

Region: SGP1 Privacy Policy Terms of Service

Type here to search

Low visibility 2:23 am 22/12/2025

Devices - Blynk.Console    ESP32 and Blynk Template Setup    Smart Classroom Dashboard

Not secure 10.226.199.162

# Smart Classroom Dashboard

## Student Overview

Name	Attendance %	Presentation %	Score	Remarks	Feedback
Haruki	16%	15%	90	Excellent	<ul style="list-style-type: none"> <li>Excellent class participation!</li> <li>Outstanding presentation skills</li> <li>Teacher note: Excellent</li> </ul>
Mano	11%	11%	89	Good	<ul style="list-style-type: none"> <li>Good engagement, try to participate more</li> <li>Solid presentation delivery</li> <li>Teacher note: Good</li> </ul>
Ibana	23%	0%	45	Not bad	<ul style="list-style-type: none"> <li>Excellent class participation!</li> <li>Teacher note: Not bad</li> </ul>
Amna	6%	5%	23	Need Hard Work	<ul style="list-style-type: none"> <li>Needs to increase participation</li> <li>Practice presentation timing</li> </ul>

Activate Windows Go to Settings to activate Windows.

Low visibility 2:27 am 22/12/2025

Devices - Blynk.Console | ESP32 and Blynk Template Setup | Smart Classroom Dashboard | Not secure 10.226.199.162

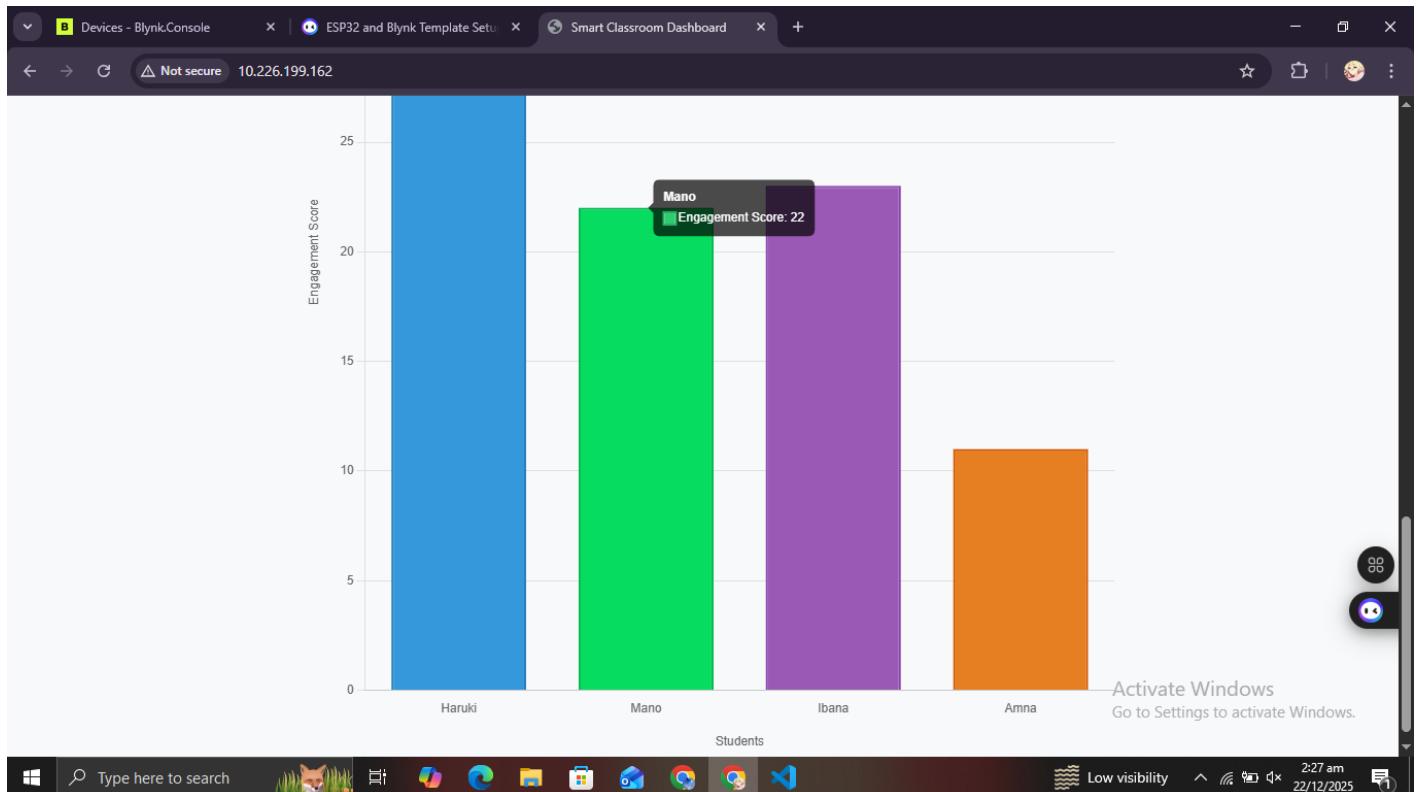
Ibana	23%	0%	45	Not bad	<ul style="list-style-type: none"><li>Excellent class participation!</li><li>Teacher note: Not bad</li></ul>
Amna	6%	5%	23	Need Hard Work	<ul style="list-style-type: none"><li>Needs to increase participation</li><li>Practice presentation timing</li><li>Teacher note: Need Hard Work</li></ul>

**Teacher Notes**

**Top Performer:** Haruki (Score: 90, Engagement: 31)

**Student to Focus:** Amna (Score: 23, Engagement: 11)

**Student Engagement Comparison**



2:58

5.81 KB/s WiFi 33%

## Smart ClassRoom



Select Student

V2: Choose... ▾

Active Student

Blynk

Student PIR



Projector



Presentation Time



Presence Status

V12: -

Add Score

- 0 +

Add Remarks

type here

V17: S

Saved Score

V18: -

Saved Remarks

Blynk



3:00

32.6 KB/s WiFi 33%



## Smart ClassRoom •

Select Student

1039 - Mahnoor



Active Student

Blynk

Student PIR



Projector



Presentation Time



Presence Status

9

Add Score

— 78 +

Add Remarks

Good

Save

Saved Score

78

Saved Remarks

Blynk

# **Node-RED Database Integration**

## **Purpose**

Node-RED is integrated as a database middleware to provide permanent storage of student data. It ensures attendance, presentation, scores, and remarks are saved persistently for historical analysis and reporting.

## **System Architecture with Node-RED**

ESP32 → MQTT (Port 1883) → Node-RED → Database (MongoDB / MySQL / InfluxDB)

## **MQTT Configuration**

Broker IP: 10.226.199.65

Port: 1883

Topic: smartclassroom/student

Protocol: MQTT v3.1.1

Client ID: ESP32\_SmartClass

## **Data Format Sent to Node-RED**

```
{ "student":"Student Name", "attendance":85, "presentation":92, "score":88, "remarks":"Teacher comments" }
```

## **Node-RED Flow Configuration**

Node-RED subscribes to MQTT topic smartclassroom/student, parses incoming JSON data, and stores it into a database. A dashboard is used for visualization and historical queries.

## **Advantages of Node-RED Integration**

- Permanent data storage
- Historical performance tracking
- Automated report generation
- Multi-user access
- Analytics and visualization