EMBEDDED SYSTEM AND IOT

Fitness Application Using Gyroscope

Presentation By
G Thirukumaran - 22AD116
P Thirumalaivasan - 22AD117

LIST OF CONTENTS

Problem Statement

Proposed Solution

About MPU6050

Application Details

Conclusion

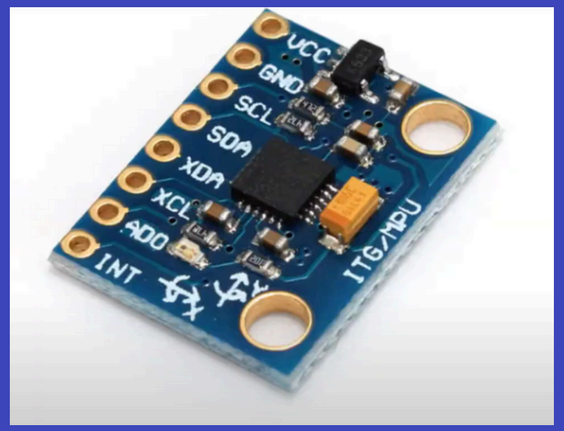
PROBLEM STATEMENT

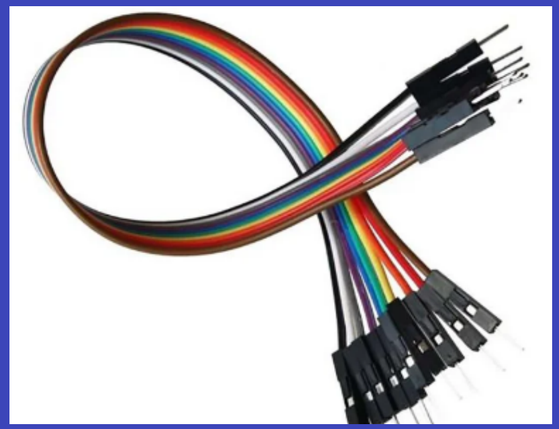
Design a real-time motion data from the gyro sensor to detect specific movement patterns or gestures in a fitness application

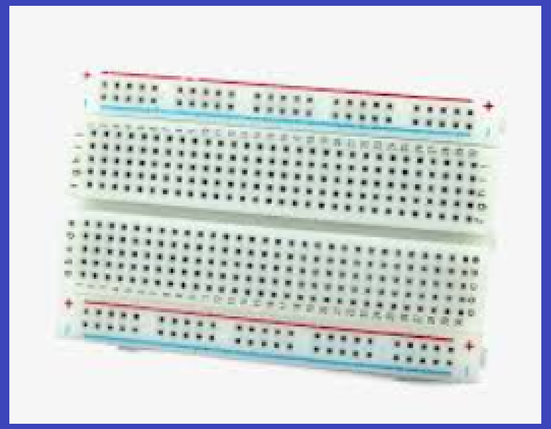
PROPOSED SOLUTION

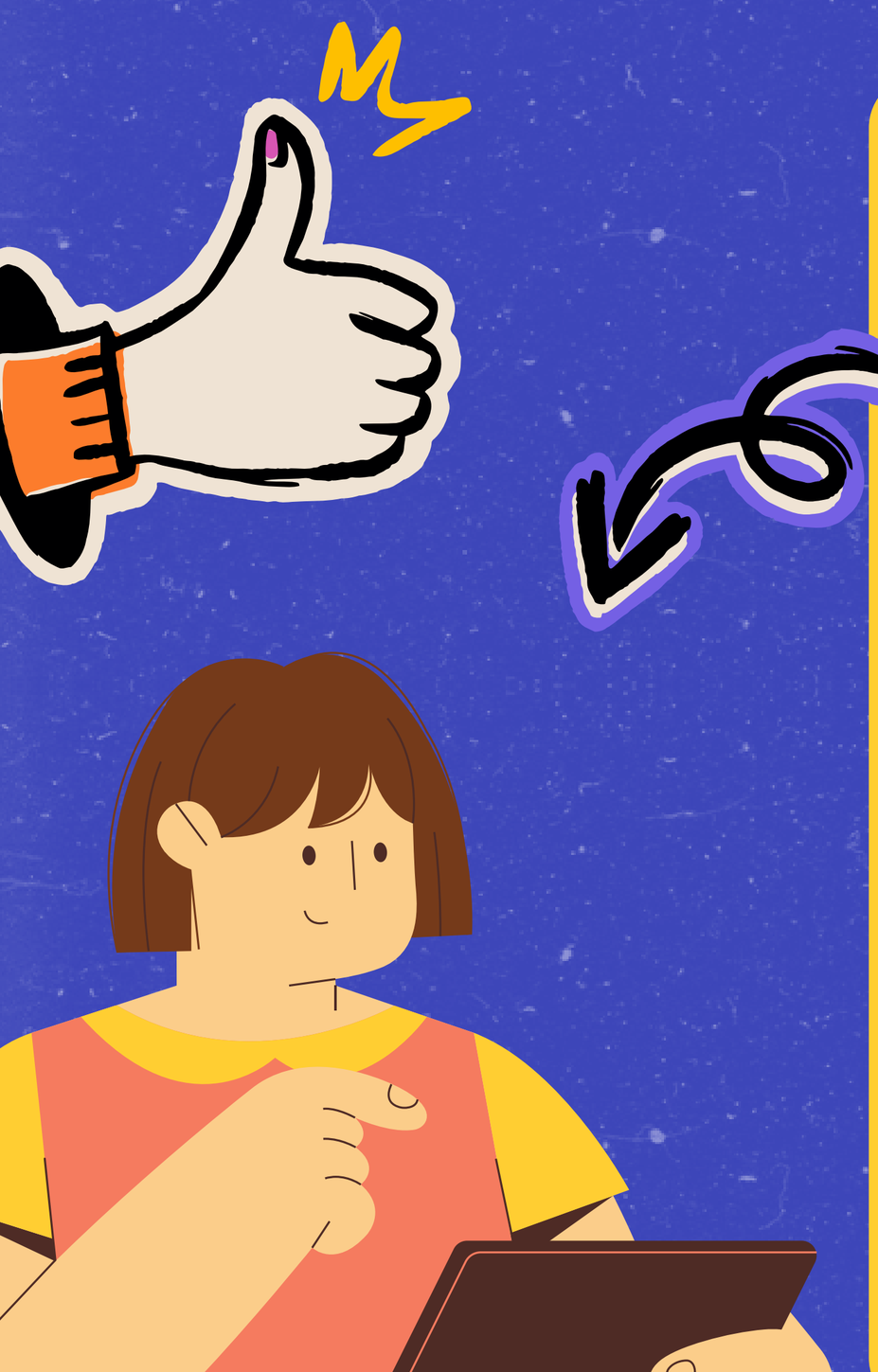
- Uses MPU6050 sensor with ESP32 devkitv1 microcontroller to capture and process accelerometer and gyroscope data for exercise detection.
- Detects specific exercises like steps, squats, jumping jacks, push-ups, and sit-ups by setting thresholds for movement patterns.
- Integrates with the Blynk app to display real-time updates and exercise counts directly on the user’s device.
- Provides visual feedback upon reaching daily fitness targets, keeping users informed of their progress.
- Enables users to set, monitor, and achieve daily exercise goals effectively with real-time tracking and feedback.

COMPONENTS REQUIRED





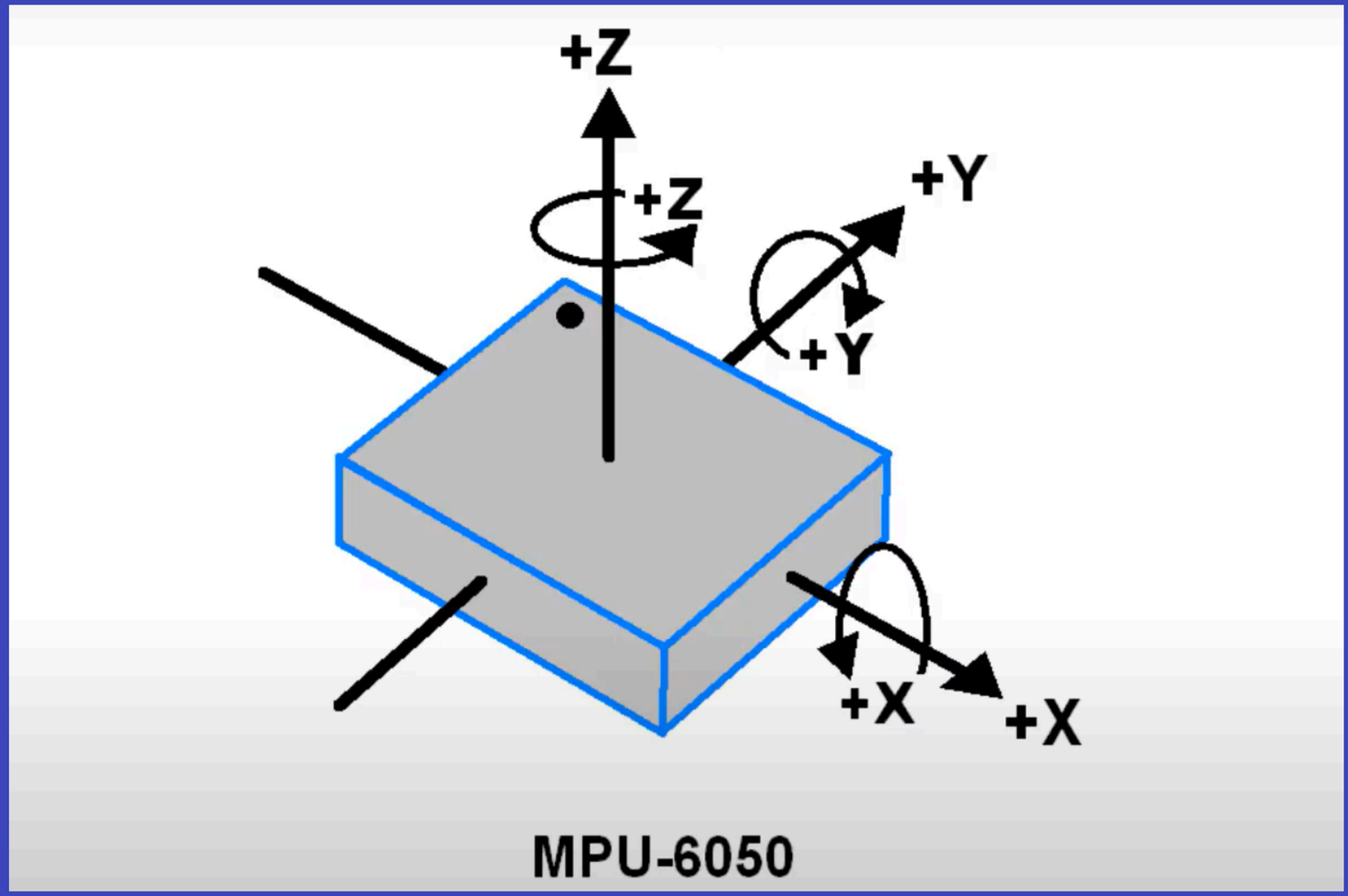




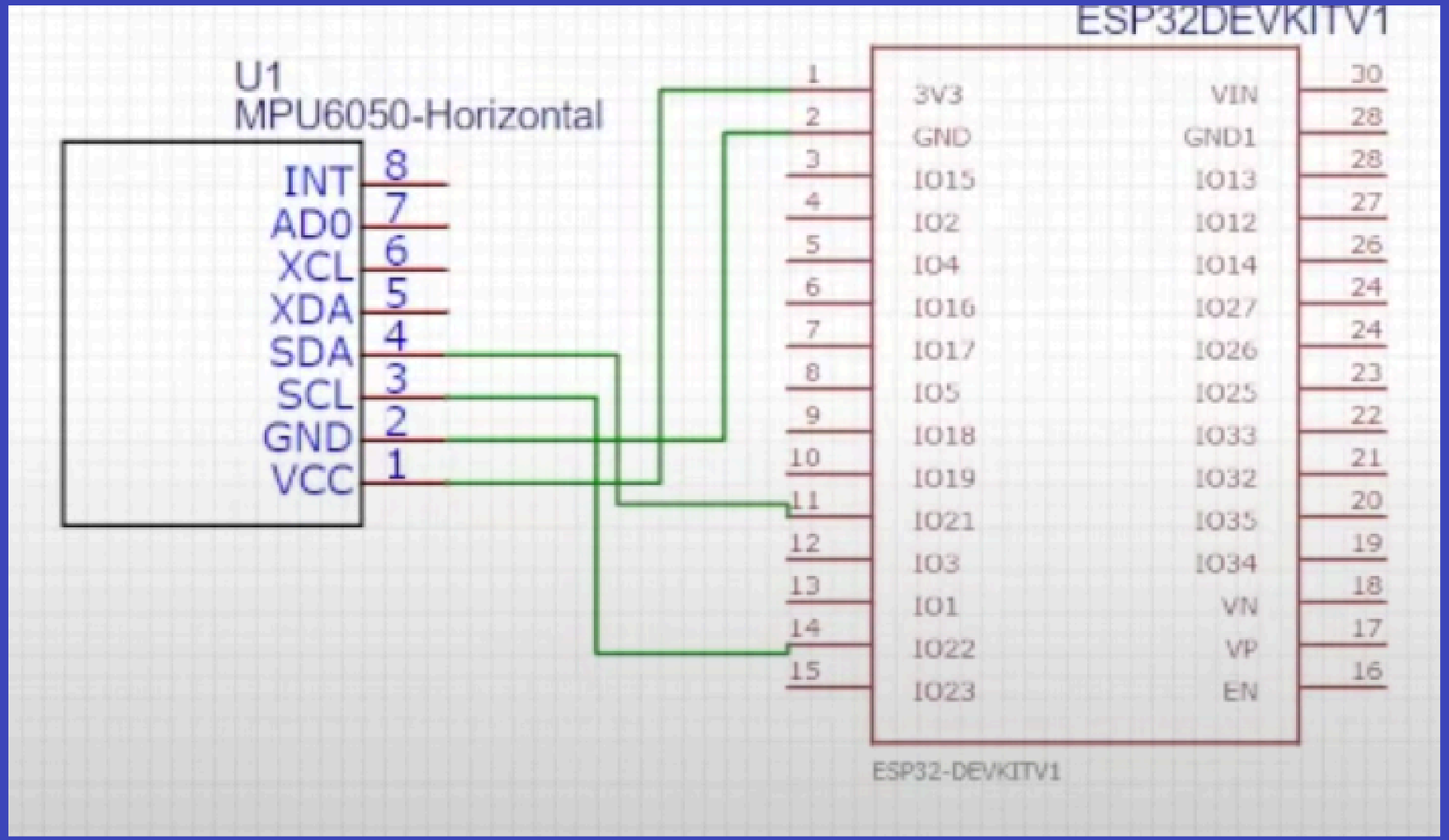
ABOUT MPU6050

- The MPU6050 is a 5-axis motion tracking sensor
- 3-axis accelerometer - Movement in X, Y, Z directions
- 2-axis gyroscope - rotational speed as roll, pitch
- The sensor is widely used in applications such as drones, fitness trackers, and gaming controllers.
- It communicates with ESP32 via I2C protocol

MPU6050 WORKING



HARDWARE CONNECTIONS



APPLICATION FEATURES

- STEPS COUNT
- JUMP COUNT
- SQUAT COUNTS
- SIT UPS COUNT
- PUSH UPS COUNT
- DETECT EXERCISE
- TARGET FIXING
- DISPLAY SENSOR DATA
- MOBILE APPLICATION

APPLICATION WORKING

- **Step Detection** - Detects based on vertical acceleration **(Z-axis) changes.**
- **Squat Detection** - Detects based on the dip in **Z-axis acceleratio**n as the user squats down and comes back up.
- **Jumping Jack Detection** - Uses both **X and Z acceleration to detect** the upward and lateral motion characteristic of jumping jacks.
- **Push-Up Detection** - Detects push-ups based on angular rotation data from the gyroscope **(Y-axis rotation).**
- **Sit-Up Detection** - Uses changes in **X-axis rotation and Z-axis acceleration** to identify the sit-up motion.

CODE USED

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#define BLYNK_TEMPLATE_ID "TMPL3FjCusafv"
#define BLYNK_TEMPLATE_NAME "Fitness Application"
#define BLYNK_AUTH_TOKEN "LKMjzJ1-qrr6XHWsmkz5LbdPlc9GY-33"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
BlynkTimer timer;
char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "GTK";
char pass[] = "gtk21032005";

Adafruit_MPU6050 mpu;

// Variables for step counting and exercise tracking
int stepCount = 0;
bool stepDetected = false;
float previousZ = 0;
float stepThreshold = 1.0; // Step detection threshold
int dailyStepTarget = 1000; // Daily step goal
```

CODE USED

```
int squatCount = 0;
bool squatDetected = false;
int dailySquatTarget = 50;

int jumpingJackCount = 0;
bool jumpingJackDetected = false;
int dailyJumpingJackTarget = 50;

int pushUpCount = 0;
bool pushUpDetected = false;
int dailyPushUpTarget = 50;

int sitUpCount = 0;
bool sitUpDetected = false;
int dailySitUpTarget = 50;

unsigned long lastMessageTime = 0;  // Last message display time
const unsigned long messageDisplayDelay = 1000;  // Delay of 1 second

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10);
```

CODE USED

```
void detectStep(float accelerationZ) {
  if (abs(accelerationZ - previousZ) > stepThreshold && !stepDetected) {
    stepCount++;
    stepDetected = true;
    Serial.print("Step detected! Total steps: ");
    Serial.println(stepCount);
    updateExerciseMessage("Step detected!");
  }
  if (abs(accelerationZ - previousZ) < (stepThreshold / 2)) {
    stepDetected = false;
  }
  previousZ = accelerationZ;

  if (stepCount >= dailyStepTarget) {
    Blynk.virtualWrite(V13, 255);  // Turn on LED
  }
}

void detectSquat(float accelerationZ) {
  if (accelerationZ < -8 && !squatDetected) {
    squatCount++;
    squatDetected = true;
    Serial.print("Squat detected! Total squats: ");
    Serial.println(squatCount);
```

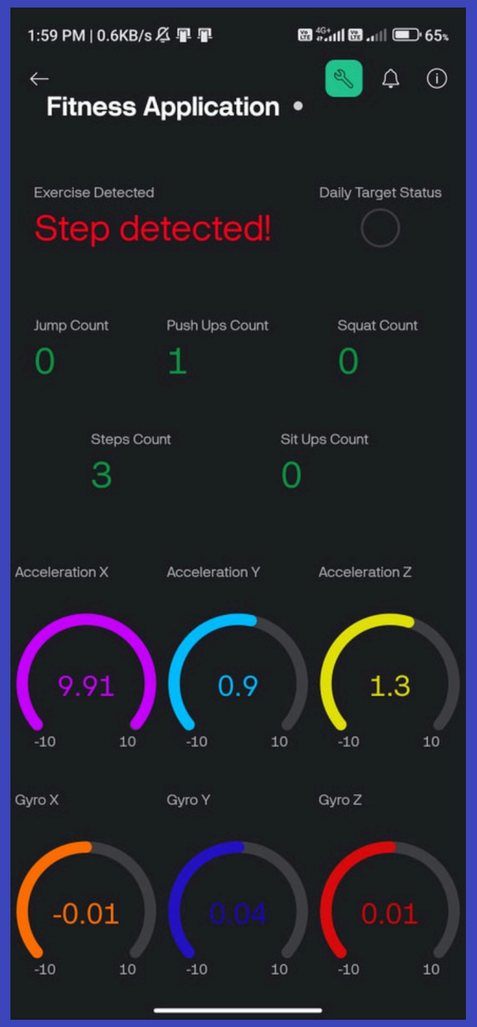
CODE USED

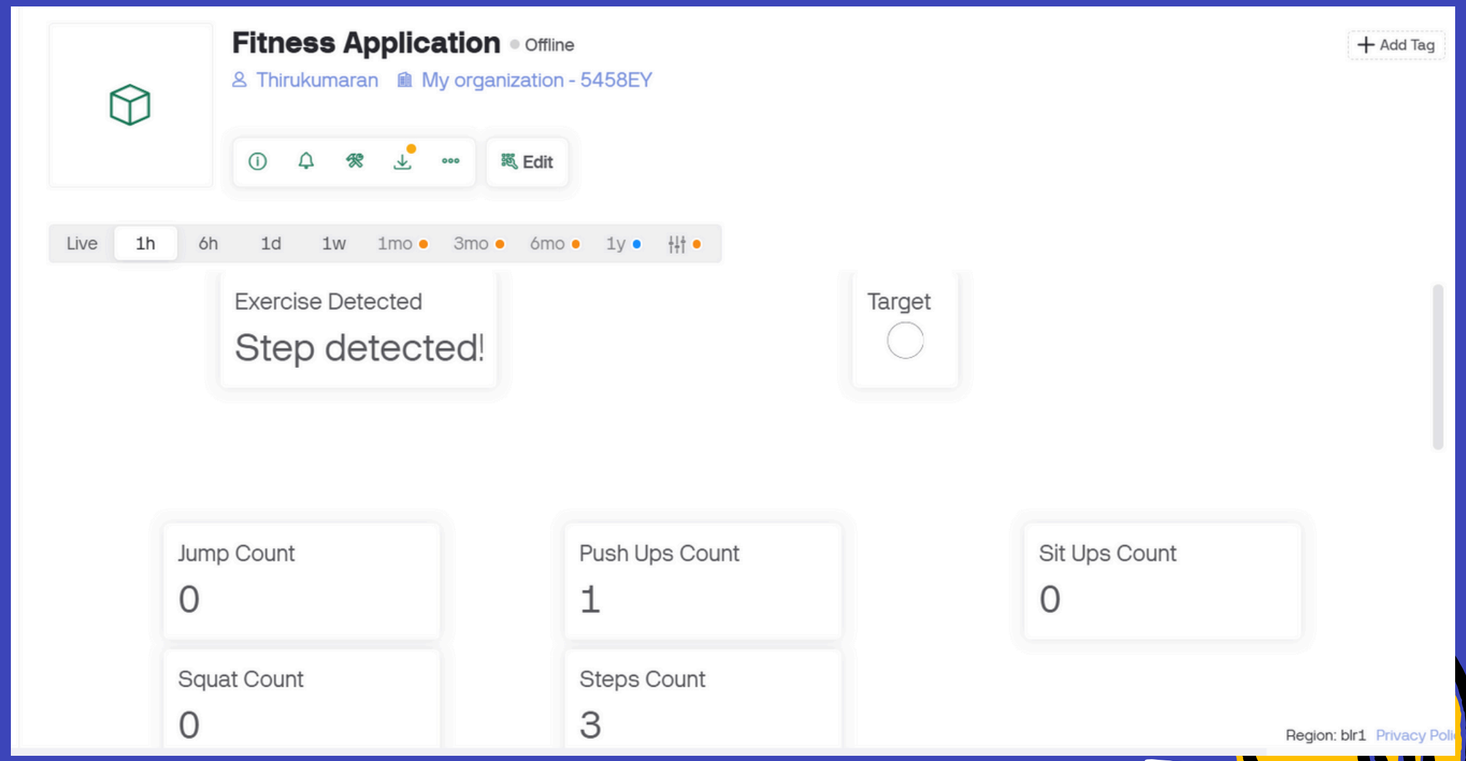
```
  Blynk.virtualWrite(V0, accelerationX);
  Blynk.virtualWrite(V1, gyroX);
  Blynk.virtualWrite(V2, temp.temperature);
  Blynk.virtualWrite(V3, accelerationZ);
  Blynk.virtualWrite(V4, accelerationY);
  Blynk.virtualWrite(V5, gyroY);
  Blynk.virtualWrite(V6, stepCount);
  Blynk.virtualWrite(V7, squatCount);
  Blynk.virtualWrite(V8, jumpingJackCount);
  Blynk.virtualWrite(V9, pushUpCount);
  Blynk.virtualWrite(V10, sitUpCount);
  Blynk.virtualWrite(V11, gyroZ);

  Serial.println("");
  delay(500);
}

void loop() {
  Blynk.run();
  timer.run();
}
```

APPLICATION INTERFACE





CONCLUSION

This fitness application leverages the MPU6050 sensor and ESP32 to detect exercises, count repetitions, and track daily goals in real-time. Integration with the Blynk app provides instant feedback and progress updates, enhancing user motivation and engagement.

THANK YOU