```
/*
  IoT Based Fall Detection System
  Board: Wemos D1 Mini (ESP8266)
  Sensor: MPU6050 (MPU6050_tockn library)
*/

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <MPU6050_tockn.h>
#include <Wire.h>

// ---------------- WIFI CONFIG ----------------
const char* ssid = "YOUR_WIFI_NAME";
const char* password = "YOUR_WIFI_PASSWORD";

// ---------------- SERVER CONFIG ----------------
String UID = "DS07";
String server = "http://iot.roboninja.in/index.php";

// ---------------- PIN DEFINITIONS ----------------
#define D7_PIN D7
#define BUZZER D5

// ---------------- MPU OBJECT ----------------
MPU6050 mpu6050(Wire);

// ---------------- VARIABLES ----------------
float accMagnitude;
float gyroMagnitude;
bool fallDetected = false;

unsigned long lastSend = 0;
unsigned long interval = 1000;  // Send data every 1 second

// ==================================================

void setup() {
  Serial.begin(115200);
  delay(2000);

  pinMode(D7_PIN, OUTPUT);
  pinMode(BUZZER, OUTPUT);

  digitalWrite(D7_PIN, LOW);
  digitalWrite(BUZZER, LOW);

  // Initialize I2C (Wemos: SDA=D2, SCL=D1)
  Wire.begin(D2, D1);

  // Initialize MPU6050
  mpu6050.begin();

  Serial.println("Calibrating Gyroscope... Keep Sensor Still");
  mpu6050.calcGyroOffsets(true);
  Serial.println("Calibration Complete");

  // Connect to WiFi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi Connected Successfully");
}

// ==================================================

void loop() {

  // Update MPU readings
  mpu6050.update();

  float ax = mpu6050.getAccX();
  float ay = mpu6050.getAccY();
  float az = mpu6050.getAccZ();

  float gx = mpu6050.getGyroX();
  float gy = mpu6050.getGyroY();
  float gz = mpu6050.getGyroZ();

  // Calculate magnitudes
  accMagnitude = sqrt(ax * ax + ay * ay + az * az);
  gyroMagnitude = sqrt(gx * gx + gy * gy + gz * gz);

  Serial.print("Acceleration Magnitude: ");
  Serial.print(accMagnitude);
  Serial.print(" | Gyro Magnitude: ");
  Serial.println(gyroMagnitude);

  // ---------------- FALL DETECTION LOGIC ----------------
  // Stage 1: Free fall detection
  if (accMagnitude < 0.5) {

    delay(200);  // Short confirmation delay

    // Stage 2: Impact or sudden rotation
    if (accMagnitude > 2.5 || gyroMagnitude > 250) {
      fallDetected = true;
      digitalWrite(BUZZER, HIGH);
      Serial.println("Fall Detected");
    }
  }
  else {
    fallDetected = false;
    digitalWrite(BUZZER, LOW);
  }

  // ---------------- IOT COMMUNICATION ----------------
  if (millis() - lastSend > interval) {
    sendAngularVelocity();
    readD7Control();
    lastSend = millis();
  }

  delay(100);
}

// ======================================================
// SEND ANGULAR VELOCITY TO SERVER
// ======================================================
void sendAngularVelocity() {

  if (WiFi.status() == WL_CONNECTED) {

    HTTPClient http;

    String url = server + "?action=write&UID=" + UID +
                 "&Angular_Velocity=" + String(gyroMagnitude);
    http.begin(url);
    http.GET();
    http.end();
  }
}

// =====================================================
// READ D7 STATE FROM SERVER
// =====================================================
void readD7Control() {

  if (WiFi.status() == WL_CONNECTED) {

    HTTPClient http;

    String url = server + "?action=read&UID=" + UID + "&D7";

    http.begin(url);
    int httpCode = http.GET();

    if (httpCode > 0) {
      String payload = http.getString();
      payload.trim();

      if (payload == "1") {
        digitalWrite(D7_PIN, HIGH);
      } else {
        digitalWrite(D7_PIN, LOW);
      }
    }

    http.end();
  }
}
```