Code for Arduino IDE:

```
#include <ESP32Servo.h>
#include <WiFi.h>
#include "esp_camera.h"
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "driver/rtc_io.h"
#include "esp_http_server.h"

// Pin Definitions
#define IR_SENSOR_LEFT 32
#define IR_SENSOR_RIGHT 33
#define SOUND_SENSOR 35
#define BUZZER_PIN 14
#define MOTOR_LEFT_FORWARD 26
#define MOTOR_LEFT_BACKWARD 25
#define MOTOR_RIGHT_FORWARD 17
#define MOTOR_RIGHT_BACKWARD 16

// Wi-Fi Credentials
const char* ssid = “SSID”;
const char* password = “Password”;

// Camera Server Function Prototypes
void startCameraServer();

// Camera configuration settings
camera_config_t config;

void setup() {
  // Disable brownout detector
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

  Serial.begin(115200);
  WiFi.begin(ssid, password);

  pinMode(IR_SENSOR_LEFT, INPUT);
  pinMode(IR_SENSOR_RIGHT, INPUT);
  pinMode(SOUND_SENSOR, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
```

```
pinMode(MOTOR_LEFT_FORWARD, OUTPUT);
pinMode(MOTOR_LEFT_BACKWARD, OUTPUT);
pinMode(MOTOR_RIGHT_FORWARD, OUTPUT);
pinMode(MOTOR_RIGHT_BACKWARD, OUTPUT);

// Start Camera
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = 5;
config.pin_d1 = 18;
config.pin_d2 = 19;
config.pin_d3 = 21;
config.pin_d4 = 36;
config.pin_d5 = 39;
config.pin_d6 = 34;
config.pin_d7 = 35;
config.pin_xclk = 0;
config.pin_pclk = 22;
config.pin_vsync = 25;
config.pin_href = 23;
config.pin_sscb_sda = 26;
config.pin_sscb_scl = 27;
config.pin_pwdn = 32;
config.pin_reset = -1;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_UXGA; // Change resolution here if needed
config.jpeg_quality = 10;
config.fb_count = 1;

// Init camera
if (esp_camera_init(&config) != ESP_OK) {
  Serial.println("Camera init failed");
  return;
}

// Start the camera server
startCameraServer();
Serial.println("Camera Ready! Use 'http://<ESP32-CAM IP>/' to connect");

// Wait for connection to Wi-Fi
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
```

```
    Serial.print(".");
  }
  Serial.println(WiFi.localIP());
}

void loop() {
  // Read sensors
  int irLeft = digitalRead(IR_SENSOR_LEFT);
  int irRight = digitalRead(IR_SENSOR_RIGHT);
  int soundDetected = digitalRead(SOUND_SENSOR);

  // Obstacle avoidance logic
  if (irLeft == LOW && irRight == LOW) {
    // No obstacle, move forward
    moveForward();
  } else if (irLeft == HIGH) {
    // Obstacle on left, turn right
    turnRight();
  } else if (irRight == HIGH) {
    // Obstacle on right, turn left
    turnLeft();
  }

  // Sound detection alert
  if (soundDetected == HIGH) {
    activateBuzzer();
  } else {
    deactivateBuzzer();
  }

  delay(100);  // Delay to avoid excessive processing
}

// Motor Control Functions
void moveForward() {
  digitalWrite(MOTOR_LEFT_FORWARD, HIGH);
  digitalWrite(MOTOR_LEFT_BACKWARD, LOW);
  digitalWrite(MOTOR_RIGHT_FORWARD, HIGH);
  digitalWrite(MOTOR_RIGHT_BACKWARD, LOW);
}

void turnRight() {
  digitalWrite(MOTOR_LEFT_FORWARD, HIGH);
```

```
  digitalWrite(MOTOR_LEFT_BACKWARD, LOW);
  digitalWrite(MOTOR_RIGHT_FORWARD, LOW);
  digitalWrite(MOTOR_RIGHT_BACKWARD, HIGH);
}

void turnLeft() {
  digitalWrite(MOTOR_LEFT_FORWARD, LOW);
  digitalWrite(MOTOR_LEFT_BACKWARD, HIGH);
  digitalWrite(MOTOR_RIGHT_FORWARD, HIGH);
  digitalWrite(MOTOR_RIGHT_BACKWARD, LOW);
}

void stopMoving() {
  digitalWrite(MOTOR_LEFT_FORWARD, LOW);
  digitalWrite(MOTOR_LEFT_BACKWARD, LOW);
  digitalWrite(MOTOR_RIGHT_FORWARD, LOW);
  digitalWrite(MOTOR_RIGHT_BACKWARD, LOW);
}

void activateBuzzer() {
  digitalWrite(BUZZER_PIN, HIGH);
}

void deactivateBuzzer() {
  digitalWrite(BUZZER_PIN, LOW);
}
```

Setting up the Camera:

The following function initializes the camera server for real-time video streaming:

```
void startCameraServer() {
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();
  httpd_start(&config, NULL);
}
```