NAME : MUHAWENIMANA CALORINE

EMAIL: 2023abit1921f@kab.ac.ug

GitHub Account: CALORINE1

**IOT training assignment two**

2) Design clear schematic diagrams for your projects identifying all necessary devices and modules required.

3) [MAIN POINT] identify all the required libraries and APIs for your modules to function according to the desired functionalit

**Solution**

**Devices and Modules required for automatic parking system**

Microcontroller: Central unit that controls the entire system (e.g., Arduino, Raspberry Pi).

Ultrasonic Sensors: Detect obstacles and measure the distance to guide parking.

Infrared (IR) Sensors: Detect the presence of vehicles and their position.

Motor Drivers: Control the motors for moving the vehicle (e.g., L298N motor driver).

Motors: Move the vehicle (e.g., DC motors or stepper motors).

Cameras: Provide visual feedback for advanced parking systems.

LCD Display: Display system status and instructions.

Wi-Fi Module: Enable remote control and monitoring (e.g., ESP8266).

Power Supply: Provide the necessary power to all components.

Buzzer: Provide audible alerts for obstacles and system status.

LED Indicators: Visual alerts for system status.

Servo Motors: Control steering mechanisms.

**Schematic Diagram (Description)**

The schematic diagram will include:

Connections from the microcontroller to the ultrasonic sensors, IR sensors, motor drivers, LCD display, Wi-Fi module, buzzer, and LED indicators.

Power connections to all components.

Signal pathways showing data flow between sensors, microcontroller, and actuators (motors and servos).

R**equired Libraries and APIs**

Libraries for Microcontroller (e.g., Arduino)

Ultrasonic Sensor Library :For handling distance measurements from ultrasonic sensors.

Example: `NewPing.h`

2. IR Sensor Library: For detecting the presence of vehicles.

Example: `IRremote.h`

3. Motor Driver Library: For controlling motor drivers.

Example: `AFMotor.h` (Adafruit Motor Shield library)

4. LCD Display Library: For displaying text and information on the LCD.

Example: `LiquidCrystal.h`

5. Wi-Fi Module Library : For enabling Wi-Fi communication.

Example: `ESP8266WiFi.h`

6. Buzzer Library : For controlling the buzzer for audible alerts.

Example: `Tone.h`

7. Servo Motor Library: For controlling servo motors.

Example: `Servo.h`

8. Camera Library: For handling camera input (if using Raspberry Pi).

Example: `picamera` (Python library for Raspberry Pi)

**APIs for Remote Control and Monitoring**

**1**.RESTful APIs . For sending and receiving data between the parking system and a remote server or application.

Example: Using `HTTPClient` library for making HTTP requests.

2. MQTT Protocol: For lightweight messaging between devices.

Example: `PubSubClient` library for MQTT communication.

3. WebSockets: For real-time communication between the system and a web application.

Example: `WebSockets` library for handling WebSocket communication.