Project Proposal

Course Code: **CSE413**
Course Title: **Big Data and IOT Lab**

Submitted To:

Md. Mahedi Hassan

Lecturer, Department of CSE

Daffodil International University

Submitted By:

Names:

Sumaya Akter (213-15-4450)

Oliza Umam Tinthy (213-15-4454)

Section: 60_D1

Department Of CSE

Daffodil International University

Date of Submission: **30/11/2024**

IoT-Based Car Parking System

Course and Program Outcome

Course Outcomes (COs):

- **CO1**: Understand and utilize IoT components like sensors, actuators, and Wi-Fi modules.
- **CO2**: Apply IoT concepts for real-world problem-solving, focusing on parking management systems.
- **CO3**: Design and implement efficient systems using embedded technology and cloud connectivity.
- **CO4**: Evaluate the effectiveness of IoT-based solutions in addressing parking challenges.

Table of Contents

Chapter 1: Introduction

1.1 Introduction

Efficient parking management is a pressing challenge in urban areas due to increased vehicle usage and limited parking spaces. Traditional systems often involve manual processes, leading to inefficiencies, longer wait times, and environmental impact due to vehicle idling.

This project introduces an IoT-based car parking system using modern technologies, including IR sensors for car detection, a servo motor for gate automation, and an ESP8266 microcontroller for cloud connectivity. The system integrates with the Blynk platform, enabling real-time slot monitoring and control via smartphones.

1.2 Motivation

The motivation behind this project stems from:

- The need to reduce manual intervention in parking systems.
- Enhancing convenience and time efficiency for users.
- Minimizing traffic congestion and environmental impact.
- Leveraging IoT to create scalable, cost-effective solutions for smart cities.

1.3 Objectives

The primary objectives of this project are:

1. Automate parking slot detection and gate operation using IoT components.
2. Provide real-time slot status updates to users via the Blynk app.
3. Optimize parking processes to reduce waiting times and enhance user convenience.
4. Develop a cost-effective solution that can be implemented in various environments.

1.4 Feasibility Study

Technical Feasibility:

The project uses readily available components, including IR sensors, servo motors, and the ESP8266 Wi-Fi module. These components are easy to integrate and program using platforms like Arduino IDE.

Economic Feasibility:

The system is cost-efficient compared to traditional systems, as it eliminates the need for extensive manual labor or expensive infrastructure.

Operational Feasibility:

The system is designed for ease of use, with a user-friendly Blynk interface for real-time updates and control.

1.5 Gap Analysis

Existing parking systems often lack automation and fail to provide real-time monitoring of parking slots. This project addresses these gaps by implementing IoT-driven automation and cloud-based monitoring.

1.6 Project Outcome

The project will produce:

1. An automated car parking prototype that detects vehicles and operates gates.
2. A functional interface on the Blynk app for real-time slot status updates.
3. A scalable framework suitable for multi-slot parking management.

Chapter 2: Proposed Methodology/Architecture

2.1 Requirement Analysis & Design Specification

Overview

The system comprises three primary components:

- Sensors: IR sensors for vehicle detection.
- Actuators: Servo motor for gate control.
- Controller: ESP8266 microcontroller for processing and communication.

Design Goals:

- Seamless detection of vehicles entering and exiting parking slots.
- Smooth gate operation with minimal delays.
- Accurate real-time status updates on the Blynk app.

2.2 Proposed System Design

- Vehicle Detection: IR sensor detects the presence of a car.
- Gate Control: Servo motor opens or closes the gate based on the sensor input.
- Cloud Connectivity: ESP8266 connects to the Blynk app for data updates.
- User Interface: The Blynk app displays slot status and controls gate operation.

Chapter 3: Implementation and Results

3.1 Implementation Details

Hardware Components:

- **IR Sensor**: Detects vehicle presence.
- **Servo Motor**: Operates the parking gate.
- **ESP8266 Wi-Fi Module**: Enables cloud connectivity for real-time updates.
- **Power Supply**: Ensures reliable operation of all components.

Software Components:

- **Arduino IDE**: Used for programming the ESP8266.
- **Blynk App**: Provides a graphical interface for monitoring and control.

Implementation Steps:

1. Assemble hardware components based on the circuit diagram.
2. Configure the ESP8266 with the Wi-Fi network and Blynk platform.
3. Develop code for vehicle detection, gate operation, and cloud connectivity.
4. Test the system for accuracy and efficiency.

3.2 Performance Analysis

The system’s performance is evaluated based on:

- Response Time: The gate operates within 3 seconds of vehicle detection.
- Accuracy: High detection accuracy under normal lighting conditions.
- Connectivity: Stable data updates on the Blynk app in real time.

3.3 Results and Discussion

The prototype successfully automates parking slot management, with real-time status updates visible on the Blynk app. The gate operates smoothly and reliably, ensuring a seamless user experience.

Chapter 4: Impact and Sustainability

4.1 Impact on Society and Environment

- **Social Impact:** Reduces the need for manual labor, streamlines parking processes, and enhances user convenience.
- **Environmental Impact**: Reduces vehicle idling, contributing to lower emissions.

4.2 Sustainability Plan

- **Energy Efficiency:** The system can be powered by solar energy.
- **Scalability**: The modular design allows for easy expansion to manage multiple parking slots.

Chapter 5: Conclusion

5.1 Summary

The IoT-based car parking system automates slot management, offering a smart and efficient solution for parking challenges. By integrating IoT technologies, the system ensures accuracy, real-time monitoring, and a user-friendly experience.

5.2 Limitations

- Requires a stable Wi-Fi connection for seamless operation.
- Sensor performance may be affected by extreme environmental conditions.

5.3 Future Work

- Incorporate a camera module for advanced vehicle recognition.
- Implement payment systems for automated fee collection.
- Enhance the system's scalability for large-scale parking facilities.
- Integrate a fire alarm sensor to detect fire incidents, trigger real-time alerts, and enhance safety in the parking area.

6.Code

```
#define BLYNK_TEMPLATE_ID "TMPL67K2WPNwS"
#define BLYNK_TEMPLATE_NAME "Car Parking"
#define BLYNK_AUTH_TOKEN "inwhRTDGGb_8wHbijgb_cToX5QD2XaWa"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>

// WiFi credentials
char ssid[] = "SII";
char pass[] = "password";
char auth[] = "inwhRTDGGb_8wHbijgb_cToX5QD2XaWa"; // Added semicolon
```

```
// Servo objects
Servo entryGate;
Servo exitGate;

// IR Sensor Pins
const int entrySensor = D1;
const int exitSensor = D2;
const int slot1 = D3;
const int slot2 = D4;
const int slot3 = D7;

// Blynk Virtual Pins
#define V_ENTRY_BUTTON V0
#define V_EXIT_BUTTON V1
#define V_SLOT1_LED V2
#define V_SLOT2_LED V3
#define V_SLOT3_LED V4

void setup() {
  // Serial for debugging
  Serial.begin(115200);

  // Blynk connection
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Servo initialization
  entryGate.attach(D5); // Entry gate servo
  exitGate.attach(D6);  // Exit gate servo
  entryGate.write(0);   // Closed position
  exitGate.write(0);    // Closed position

  // IR sensor setup
  pinMode(entrySensor, INPUT);
  pinMode(exitSensor, INPUT);
  pinMode(slot1, INPUT);
  pinMode(slot2, INPUT);
  pinMode(slot3, INPUT);
```

```
  Serial.println("System Initialized");
}

// Blynk button handlers
BLYNK_WRITE(V_ENTRY_BUTTON) {
  int buttonState = param.asInt();
  if (buttonState) {
    entryGate.write(90); // Open gate
    delay(3000);
    entryGate.write(0);  // Close gate
  }
}

BLYNK_WRITE(V_EXIT_BUTTON) {
  int buttonState = param.asInt();
  if (buttonState) {
    exitGate.write(90); // Open gate
    delay(3000);
    exitGate.write(0);  // Close gate
  }
}

void loop() {
  Blynk.run();

  // Slot Status Update
  Blynk.virtualWrite(V_SLOT1_LED, digitalRead(slot1) == LOW ? 255 : 0); // Occupied ->
LED ON
  Blynk.virtualWrite(V_SLOT2_LED, digitalRead(slot2) == LOW ? 255 : 0);
  Blynk.virtualWrite(V_SLOT3_LED, digitalRead(slot3) == LOW ? 255 : 0);

  // Automatic Entry Gate Control
  if (digitalRead(entrySensor) == LOW) {
    entryGate.write(90); // Open
    delay(3000);
    entryGate.write(0);  // Close
  }

  // Automatic Exit Gate Control
```

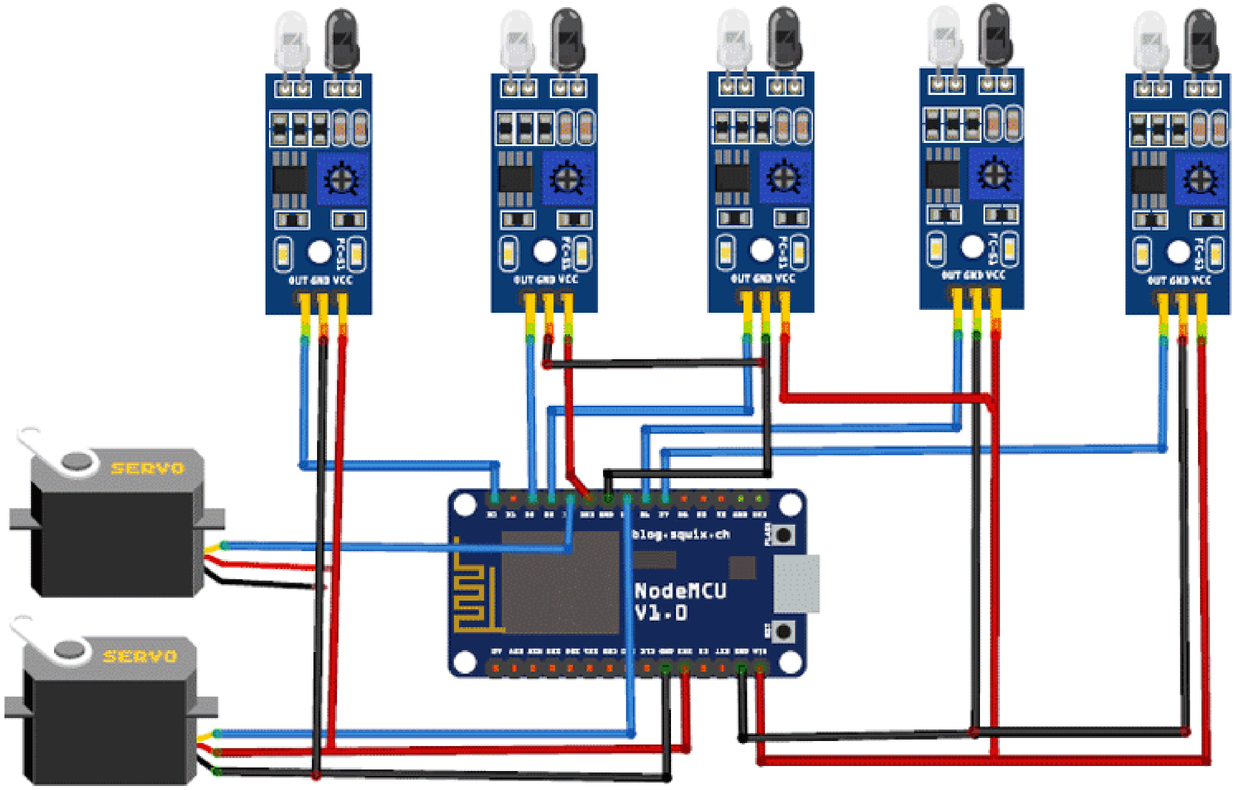
```
  if (digitalRead(exitSensor) == LOW) {
    exitGate.write(90); // Open
    delay(3000);
    exitGate.write(0);  // Close
  }
}
```

7.Circuit Diagram Overview

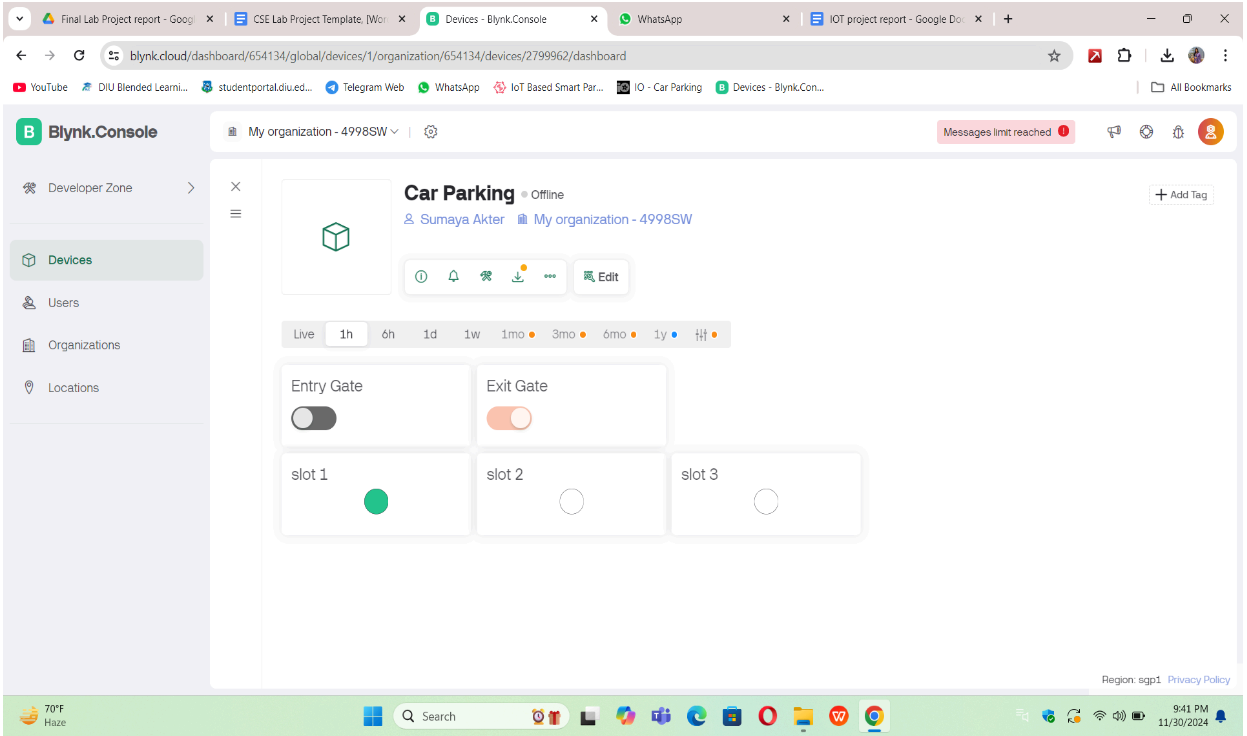
NodeMCU Pins:
D1: Entry IR Sensor
D2: Exit IR Sensor
D3: Slot 1 IR Sensor
D4: Slot 2 IR Sensor
D5: Entry Servo Signal
D6: Exit Servo Signal
D7: Slot 3 IR Sensor
Power Connections:
NodeMCU: 5V (Vin) and GND
Servos: Powered via 5V external supply (if needed) and share GND with NodeMCU.
Sensors:
All IR sensors connect to 3.3V and GND on NodeMCU.

8.Circuit Diagram

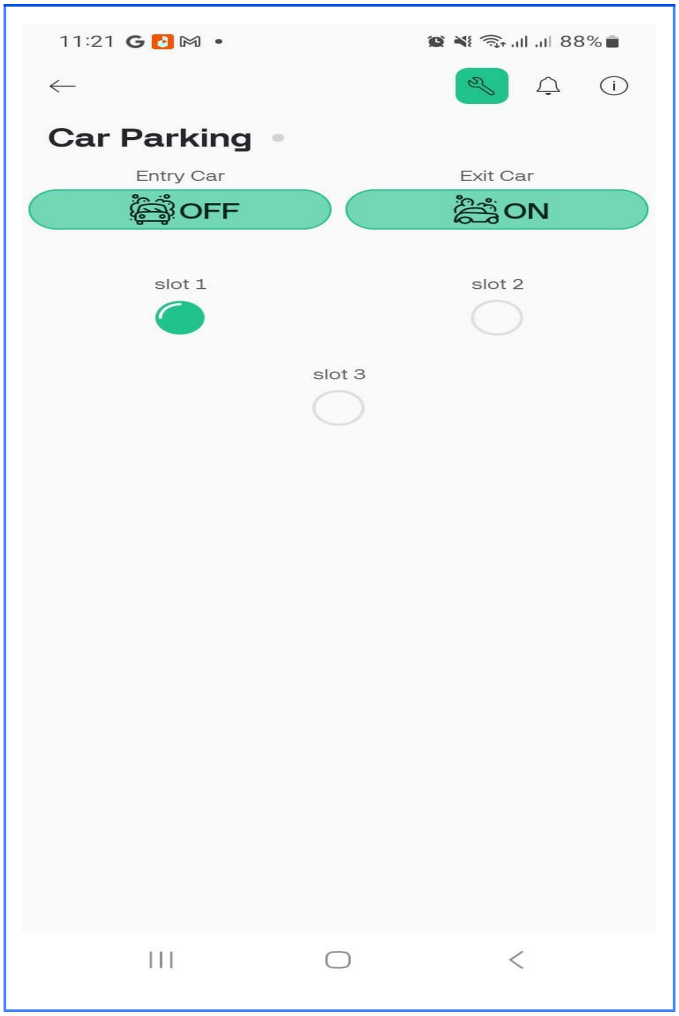
The circuit diagram for this **IoT based smart parking system project** is given below.



9.Blynk for web

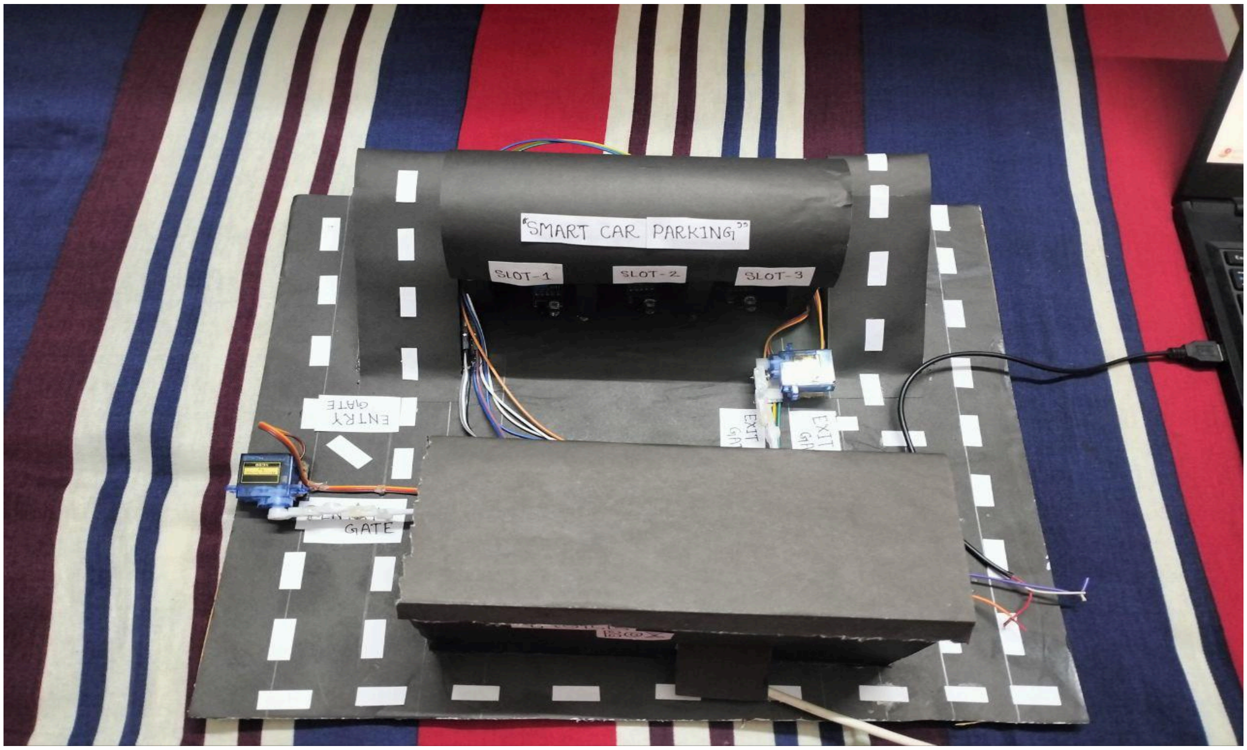


10.Blynk For Mobile App



11.Group photo

12.Project Photo



13.Project video

https://drive.google.com/file/d/1kjvTTMKA7RspSBY5-bgxysF_4EugdPdf/view?usp=drivesdk