Smart Parking & Traffic System

Team Members

Salma Abd-Elaziz
Mariam Badry
Fajr Abu Bakr
Arwa Salem

Understanding the problem

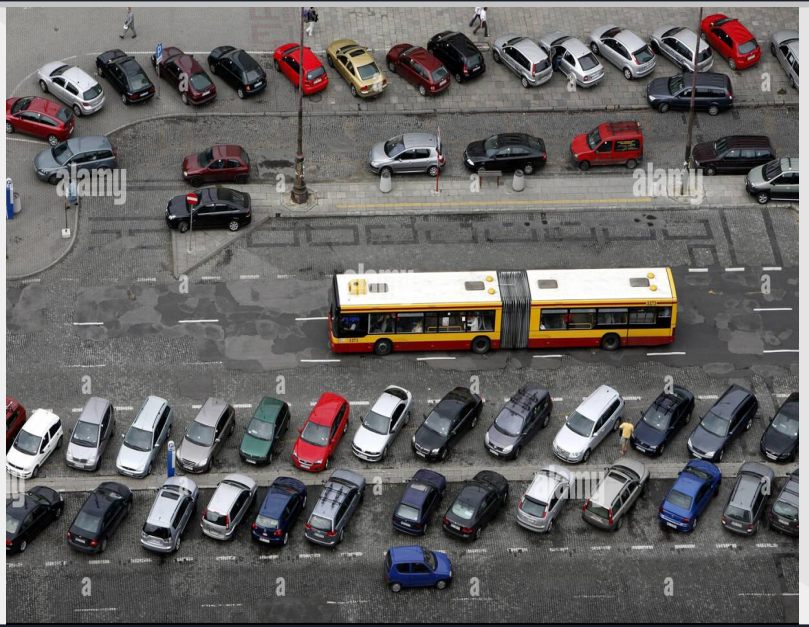
01 Drivers waste time searching for available parking spots.

02 Fixed-timer traffic lights cause unnecessary delays

03 Congestion leads to fuel waste, frustration, and pollution

04 Lack of real-time information for drivers

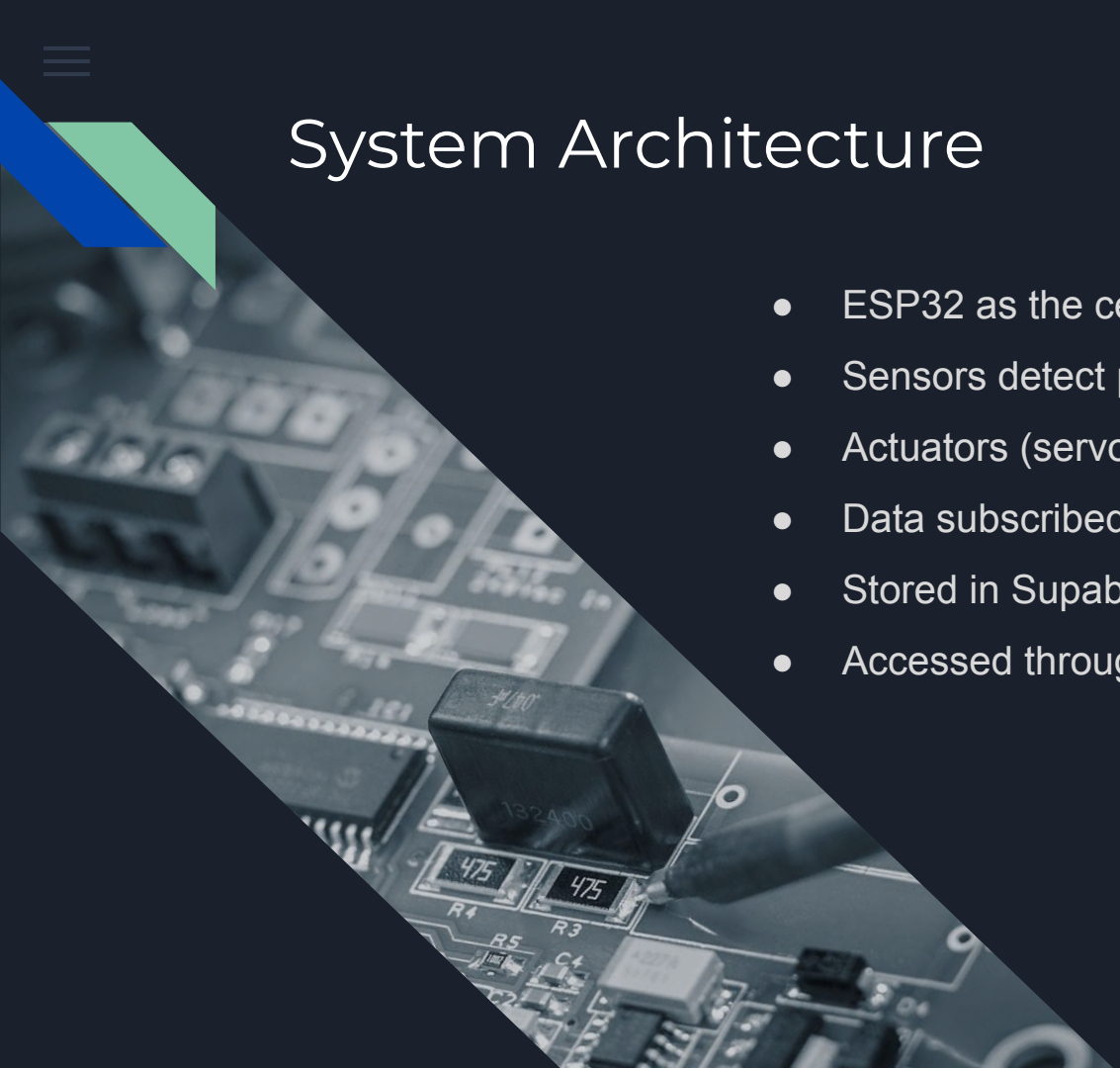
Understanding the problem



Project Objectives

Parking System	Detect available spots, display availability, and control gate access.
Traffic Management	Dynamically adjust green light duration based on congestion.
Integration	Use ESP32 with HiveMQ Cloud, Supabase, and Flutter app.
Goal	Enhance efficiency, reduce waiting times, and support smart city vision.

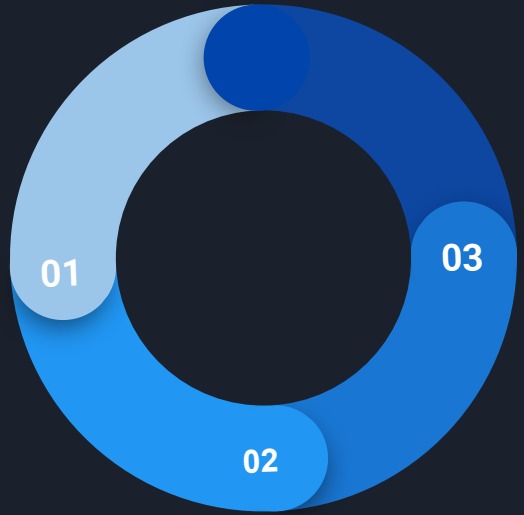
System Architecture



- ESP32 as the central IoT controller.
- Sensors detect parking availability & traffic conditions.
- Actuators (servo, LCD) respond to decisions.
- Data subscribed via **HiveMQ Cloud (MQTT)**.
- Stored in Supabase Database.
- Accessed through Flutter Mobile App.

Smart Parking System

Overview



Workflow

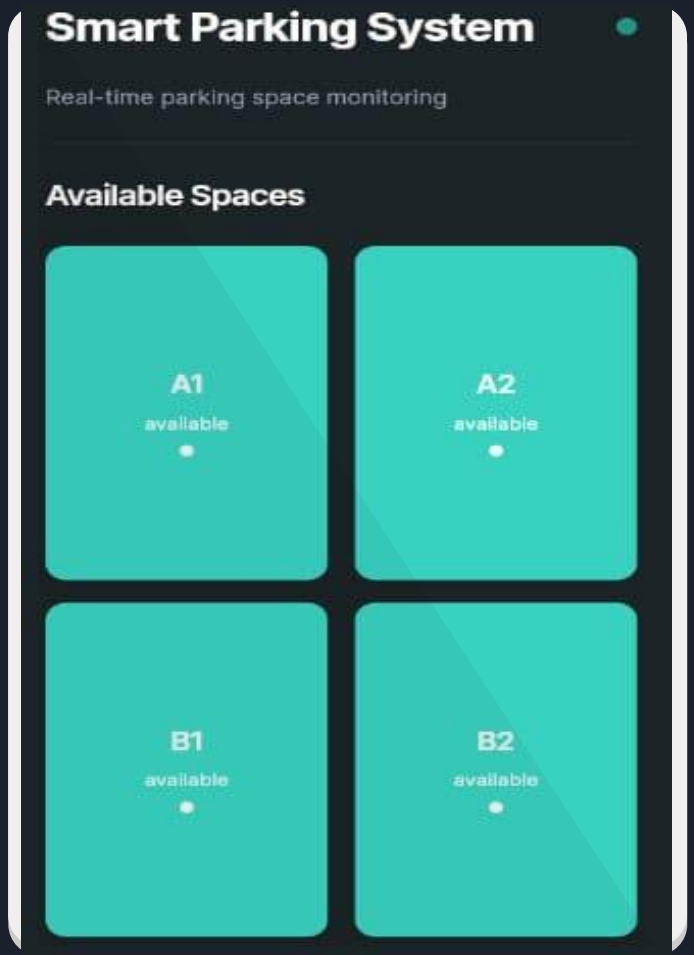
HardWare

Parking System Overview

Four parking slots monitored using IR break beam sensors.

Entrance gate controlled by IR sensor + Servo motor.

LCD display shows availability status (Available / Full).



Parking System Hardware

Inputs

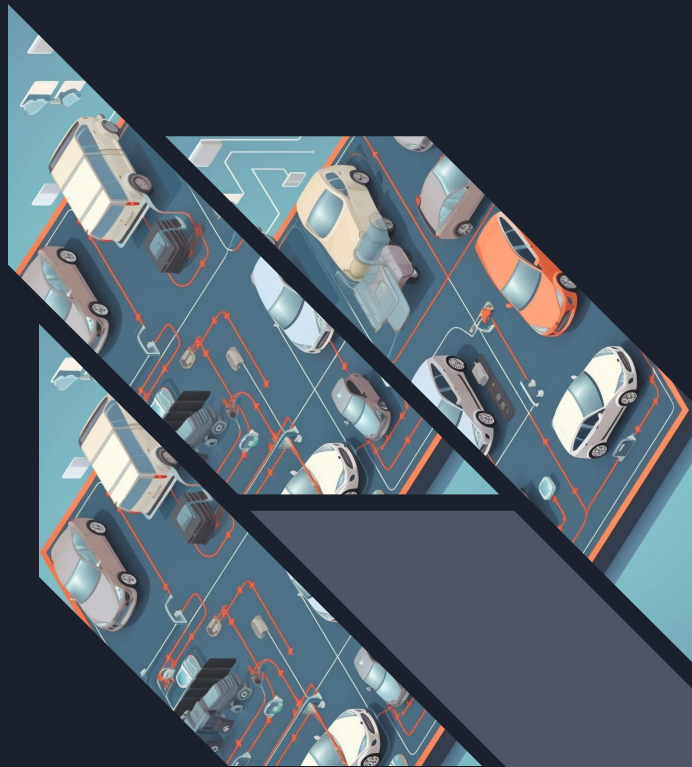
IR break beam sensors Digital Input (1 = free, 0 = occupied).

IR sensor at gate Digital Input (detects car presence).

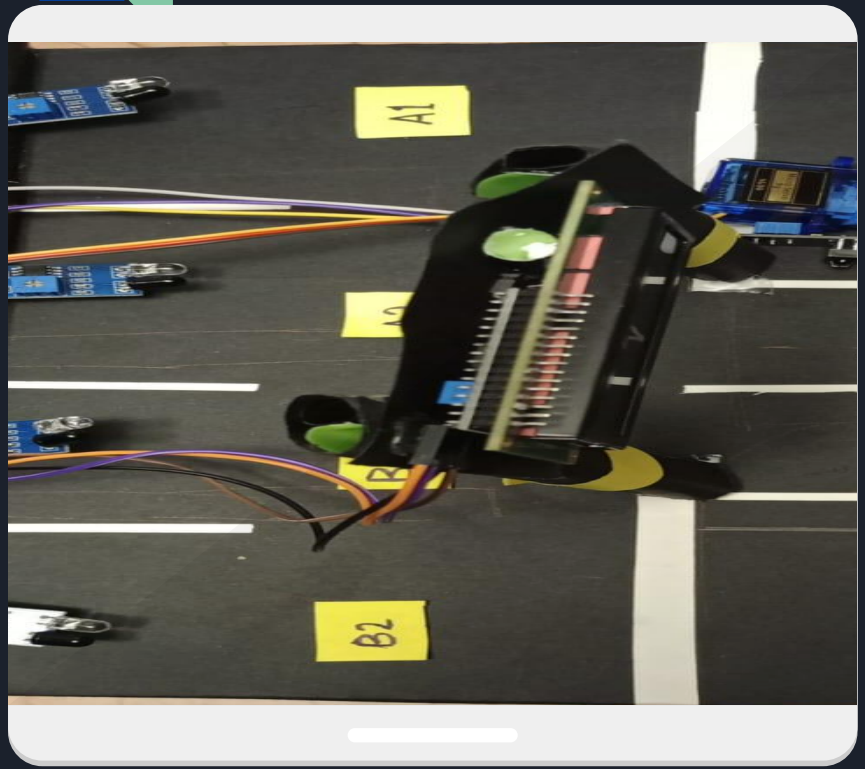
Outputs

Servo motor Digital PWM Output (rotates to open/close gate).

LCD Display (I2C) Digital Output (shows availability count).



ParkinG Workflow



01 Car arrives detected by IR sensor at gate.

02 System checks availability (IR sensors).

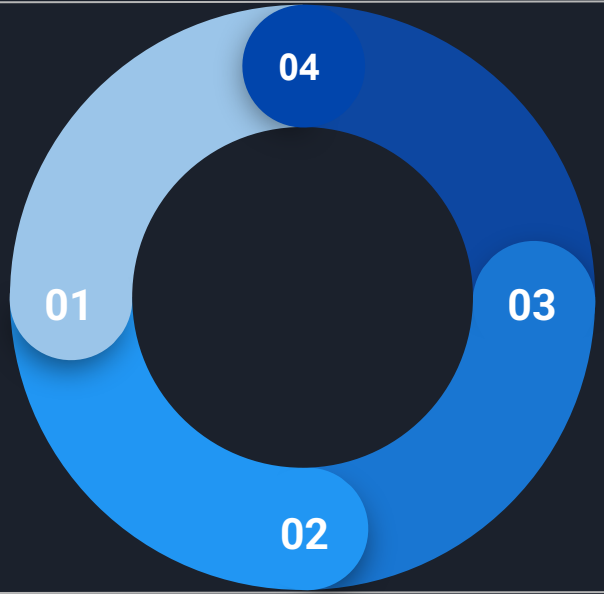
03 If space available Servo opens gate + LCD updates.

04 Else Gate stays closed + LCD shows “Full”.

Traffic Management System

Overview

Hardware



Smart Logic

Workflow

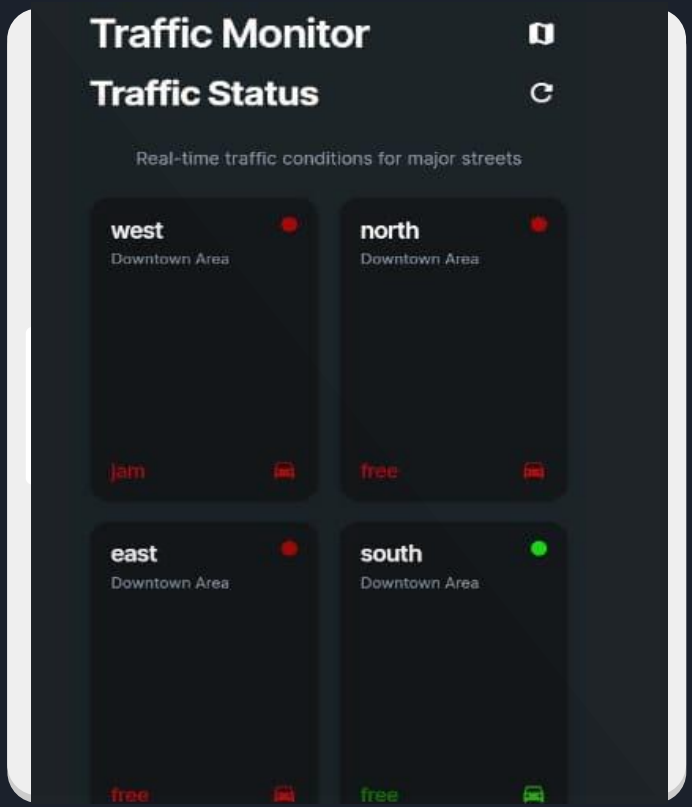
Traffic Management Overview

Four intersecting streets (2 pairs of signals).

Default cycle each pair gets 10s green.

Ultrasonic sensors detect congestion in 3 streets.

Green light extended to 15s if congestion detected.



Traffic System Hardware

Inputs

Ultrasonic sensors Digital pulse measurement converted to distance (cm).
Threshold < 7 cm = congestion.

Outputs

Traffic light modules Digital Outputs (HIGH = ON, LOW = OFF).

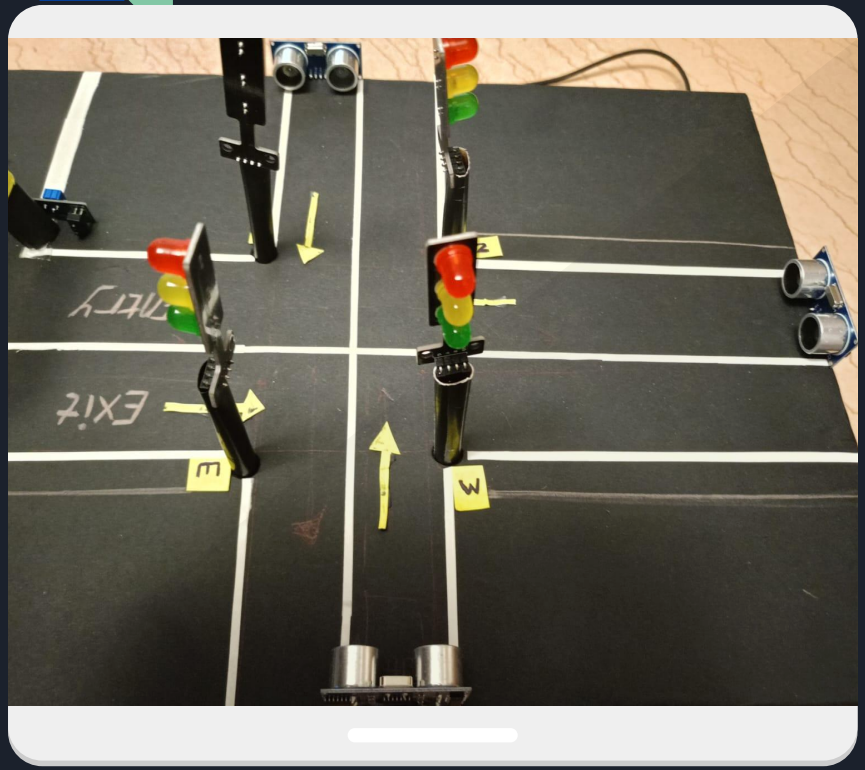
Controller

ESP32 reads sensor data, compares distances, and adjusts timing.

Traffic System Smart Logic

- Normal state 10s green light for each pair.
- If congestion detected Increase green light to 15s for its pair.
- If multiple congestions Compare distances, prioritize the street with closest distance (heavier traffic).

Traffic Workflow



01 Ultrasonic sensors measure distances.

02 ESP32 checks congestion against threshold.

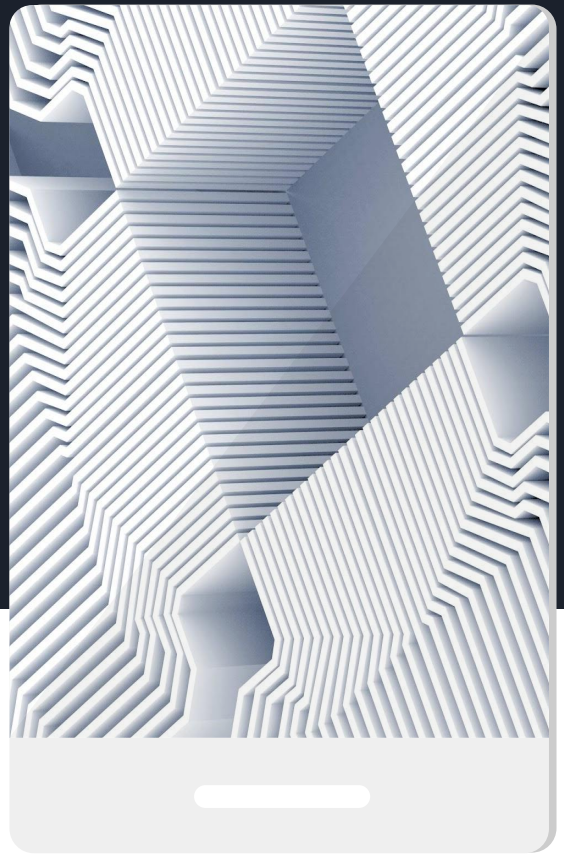
03 Selects which pair of lights to open.

04 Sets duration (10s or 15s).

05 Cycle repeats continuously.

IoT Integration

- ESP32 publishes real-time data via MQTT to HiveMQ Cloud.
- Data stored in Supabase Database for history and analytics.
- Flutter app retrieves data shows parking slots + traffic status.



Results & Benefits

01 Parking System

- Accurate detection of available spots.
- Automated gate access using IR + Servo.
- Real-time display of parking availability.

02 Traffic Management System

- Adaptive traffic light timing (10s/15s).
- Reduced waiting time at intersections.
- Prioritization based on real congestion levels.

03 Integration

- Cloud-based data flow with HiveMQ + Supabase.
- Mobile app access via Flutter.
- Smooth simulation & testing on Wokwi.

Conclusion & Future Work

Conclusion

Our IoT-based solution improves urban mobility by:

- Managing parking efficiently.
- Reducing congestion at intersections.
- Providing real-time data integration with cloud & mobile apps.

Future Work

- Deploy the system on real hardware in a smart city environment.
- Add AI/ML models for traffic prediction.
- Implement mobile notifications for drivers.
- Expand integration with navigation apps.

Thank you!