

f

**Faculty of Computers & Benha University**

**Informatics**

**“Smart Parking System Powered by AI”**

# IOT Project

**Computer Science Department**

## *Project Team*

1. **جمال عادل جمال خليفة سكشن 3**
2. **طه إبراهيم طه إبراهيم سكشن 4**
3. **احمد جمال عبدالله سليمان سكشن 2**
4. **ألاء أحمد محمد محمدعطية سكشن 2**
5. **عبدالحليم مصطفي محمد شبانة سكشن 9**

***Under Supervision of***

**Dr. Ahmed Shalaby**

**Eng. Ghada Refaay**

**Introduction**

Urban areas worldwide are grappling with the increasing challenge of managing parking spaces efficiently due to the rapid growth in vehicle numbers. Traditional parking systems are often plagued with inefficiencies, such as long wait times, manual entry processes, security vulnerabilities, and poor resource management. These issues not only inconvenience drivers but also contribute to traffic congestion and increased carbon emissions.

To address these challenges, our project proposes the development of a **Smart Parking System** that leverages artificial intelligence (AI) and Internet of Things (IoT) technologies. This system aims to automate and streamline the parking process, enhance security, and optimize resource usage within parking facilities. By integrating advanced sensors, AI badge detection, and automated control mechanisms, the smart parking system offers a modern solution to urban parking problems, significantly improving the user experience and operational efficiency. Additionally, a Flutter application will be developed to allow users to check parking slot availability, receive alerts, and get notifications for a more convenient parking experience.

The proposed system will include several components: an AI-powered badge detection model for secure entry, RFID for authentication, infrared (IR) sensors for vehicle detection, environmental monitoring using temperature and light sensors, and actuators for automated control of entry and exit doors. This combination of technologies will create a cohesive and intelligent parking management system, addressing both the needs of drivers and the operational requirements of parking facilities.

**Problem Statement**

The project addresses the challenge of inefficient and congested parking management in urban areas. Traditional parking systems lack automation, leading to wasted time, increased traffic congestion, and higher emissions. Additionally, these systems often have significant security vulnerabilities, making it easier for unauthorized individuals to access and potentially steal vehicles. The need for a more efficient, secure, and user-friendly parking solution is evident. Furthermore, there is a need for a user-friendly application to provide real-time parking slot availability and alert users in case of emergencies.

**Objectives**

 **Automate Parking Entry and Exit**: Use AI models and sensors to automate the process of entering and exiting the parking facility.

 **Enhance Security**: Implement secure badge detection model and RFID authentication to prevent unauthorized access and vehicle theft.

 **Monitor Environmental Conditions**: Use sensors to monitor and respond to environmental conditions within the parking area.

 **Improve User Experience**: Reduce the time and effort required to find and access parking spaces.

 **Optimize Resource Usage**: Use actuators and sensors to optimize the use of lights , Buzzers and other resources.

 **Develop User-Friendly Application**: Create a Flutter application to provide real-time parking slot availability, send alerts for high temperature, and assist users with notifications for a smoother parking experience.

**Scope**

The project will focus on developing a smart parking system that **includes**:

* Automated entry and exit using AI badge detection and IR sensors.
* Security measures using RFID for badge authentication and alarm systems for unauthorized access.
* Environmental monitoring with temperature sensors and photoresistors.
* User notifications through buzzers and LEDs.
* Integration of hardware components like cameras, sensors, and actuators.
* Development of a Flutter application for real-time parking slot monitoring, alerts, and notifications.

**Proposed Solution**

The proposed smart parking system will leverage a combination of AI, sensors, actuators, and a Flutter application to create a seamless, secure, and efficient parking experience:

1. **AI Badge Detection**:
   * **Camera and AI Model**: A camera captures images of badges, and an AI model verifies authorized badges.
   * **Servo Motor Operation**: Upon badge verification, servo motors open the entry door.
2. **RFID Authentication**:
   * **RFID Reader**: Reads RFID cards to authenticate users.
   * **Access Control**: Only authorized cards grant access. Unauthorized access triggers an alarm.
3. **IR Sensors**:
   * **Vehicle Detection**: IR sensors detect vehicles at entry and exit points.
   * **Exit Gate Operation**: The exit gate opens automatically when a vehicle is detected and authorized.
4. **Environmental Monitoring**:
   * **Temperature Sensor (DHT11)**: Monitors the temperature inside the parking area.
   * **High-Temperature Alert**: A buzzer sounds if the temperature exceeds a certain threshold.
   * **Flutter Application Alert**: The application sends notifications to users in case of high temperatures, indicating a potential fire hazard.
5. **Lighting Control**:
   * **Photoresistor**: Measures ambient light levels.
   * **LED Activation**: Turns on lights when ambient light is low.
6. **Security Integration**:
   * **Unauthorized Access Prevention**: The RFID system prevents unauthorized entry and exit. If an unauthorized RFID card is detected, a buzzer sounds to alert security personnel.
   * **Centralized Monitoring**: Security personnel can monitor real-time data from all sensors through a centralized dashboard.
   * **Incident Recording**: Sensor data are recorded for future review and investigation.
7. **Flutter Application**:
   * **Real-Time Parking Slot Availability**: Users can check if parking slots are available or full.
   * **Alerts and Notifications**: The app sends alerts for high temperature and unauthorized access attempts.
   * **User Assistance**: Notifications guide users through the parking process, making it more convenient and efficient.

This smart parking system aims to streamline the parking process, enhance security, and improve the overall user experience while efficiently managing resources. By addressing both efficiency and security, and incorporating a user-friendly application, the system provides a comprehensive solution to modern urban parking challenges.

**Inputs / Outputs**

|  |  |  |  |
| --- | --- | --- | --- |
| Components | Inputs | outputs | Description |
| |  | | --- | | **AI Badge Detection** |  |  | | --- | |  | | |  | | --- | | Camera image of the badge |  |  | | --- | |  | | |  | | --- | | Verification status |  |  | | --- | |  | | |  | | --- | | Captures images of badges and verifies if they are authorized. If verified, the system allows entry. |  |  | | --- | |  | |
| |  | | --- | | **RFID Reader** |  |  | | --- | |  | | |  | | --- | | RFID card |  |  | | --- | |  | | |  | | --- | | Access granted/denied status |  |  | | --- | |  | | |  | | --- | | Reads RFID cards presented by users to authenticate their identity. Determines if access is granted or denied. |  |  | | --- | |  | |
| |  | | --- | | **IR Sensors** |  |  | | --- | |  | | |  | | --- | | Presence of a vehicle |  |  | | --- | |  | | |  | | --- | | Detection signal |  |  | | --- | |  | | |  | | --- | | Detects the presence of vehicles at entry and exit points, triggering respective actions like opening doors. |  |  | | --- | |  | |
| |  | | --- | | **Temperature Sensor (DHT11)** |  |  | | --- | |  | | |  | | --- | | Temperature readings |  |  | | --- | |  | | |  | | --- | | Temperature data |  |  | | --- | |  | | |  | | --- | | Monitors the temperature inside the parking area to detect potential fire hazards. |  |  | | --- | |  | |
| **Photoresistor** | |  | | --- | | Ambient light level |  |  | | --- | |  | | |  | | --- | | Light level data |  |  | | --- | |  | | |  | | --- | | Measures the ambient light level to control the activation of LED lights. |  |  | | --- | |  | |
| |  | | --- | | **Servo Motors** |  |  | | --- | |  | | |  | | --- | | Control signals from AI badge detection and IR sensors |  |  | | --- | |  | | |  | | --- | | Door open/close action |  |  | | --- | |  | | |  | | --- | | Controls the opening and closing of entry and exit doors based on verification and vehicle detection. |  |  | | --- | |  | |
| **Buzzer** | |  | | --- | | Unauthorized access attempt, high temperature |  |  | | --- | |  | | |  | | --- | | Audible alert |  |  | | --- | |  | | |  | | --- | | Emits an audible alert in case of unauthorized access attempts or high temperature readings, indicating potential security or safety issues. |  |  | | --- | |  | |
| |  | | --- | | **LED Light** |  |  | | --- | |  | | |  | | --- | | Low ambient light detected by photoresistor |  |  | | --- | |  | | |  | | --- | | Light on/off |  |  | | --- | |  | | |  | | --- | | Activates to provide adequate lighting in low ambient light conditions, ensuring visibility and safety. |  |  | | --- | |  | |
| |  | | --- | | **Flutter Application** |  |  | | --- | |  | | |  | | --- | | Parking slot status, temperature data, security alerts |  |  | | --- | |  | | |  | | --- | | Real-time updates, notifications |  |  | | --- | |  | | Displays real-time parking slot availability, sends alerts for high temperature, and notifies users of unauthorized access attempts and guidance for parking. |

### Descriptions

* **Camera Image of the Badge**: Captured image used by the AI model for badge verification.
* **Verification Status**: Indicates whether the badge is authorized, allowing or denying entry.
* **RFID Card**: A card containing user identity information used for authentication.
* **Access Granted/Denied Status**: Result of the RFID authentication process.
* **Presence of a Vehicle**: Detected by IR sensors to initiate door operations.
* **Detection Signal**: Signal sent by IR sensors indicating vehicle presence.
* **Temperature Readings**: Data from the temperature sensor to monitor environmental conditions.
* **Temperature Data**: Information on the current temperature used for safety alerts.
* **Ambient Light Level**: Measured by the photoresistor to determine the need for lighting.
* **Light Level Data**: Data used to control LED light activation.
* **Image Capture**: Photos taken by the camera for AI processing and security purposes.
* **Image Data**: Data sent to the AI model or stored for security monitoring.
* **Control Signals**: Commands sent to servo motors to operate doors.
* **Door Open/Close Action**: Physical movement of entry and exit doors.
* **Unauthorized Access Attempt**: Detected by RFID or AI model indicating a security breach.
* **High Temperature**: Detected by the temperature sensor indicating potential fire hazard.
* **Audible Alert**: Sound produced by buzzers to alert users and security personnel.
* **Low Ambient Light**: Detected condition requiring additional lighting.
* **Light On/Off**: Activation or deactivation of LED lights.
* **Parking Slot Status**: Availability information sent to the Flutter application.
* **Real-Time Updates, Notifications**: Information provided to users through the Flutter app for a convenient and safe parking experience.

**Block Diagram**

|  |
| --- |
| Light on/off |

|  |
| --- |
|  |

|  |
| --- |
| Audible alert |

|  |
| --- |
|  |

Door open/close

Real-time updates, notifications

|  |
| --- |
| Temperature data |

|  |
| --- |
|  |

|  |
| --- |
| Light level data |

|  |
| --- |
|  |

|  |
| --- |
| Image data |

|  |
| --- |
|  |

|  |
| --- |
| Detection signal |

|  |
| --- |
|  |

|  |
| --- |
| Access granted/denied status |

|  |
| --- |
|  |

|  |
| --- |
| Verification status |

|  |
| --- |
|  |

|  |
| --- |
| Parking slot status, temperature data, security alerts |

|  |
| --- |
|  |

|  |
| --- |
| Presence of a vehicle |

|  |
| --- |
|  |

|  |
| --- |
| RFID card |

|  |
| --- |
|  |

|  |
| --- |
| Low ambient light detected by photoresistor |

|  |
| --- |
|  |

|  |
| --- |
| Unauthorized access attempt, high temperature |

|  |
| --- |
|  |

Control signals from AI badge detection and IR sensors

|  |
| --- |
| Image capture |

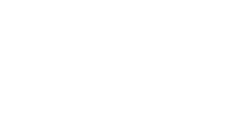
|  |
| --- |
|  |

|  |
| --- |
| Ambient light level |

|  |
| --- |
|  |

|  |
| --- |
| Temperature readings |

|  |
| --- |
|  |



Microcontroller



|  |
| --- |
| Camera image of the badge |

|  |
| --- |
|  |



**Sensors**

|  |  |
| --- | --- |
| **Sensors** | **Functionality** |
| **4 IR Sensor** | 1IR to activate the camera.  1IR to activate RFID sensor.  2IR close entry and leave gates |
| **4 IR Sensor** | It detects cars when they enter or leave the garage (count number of the cars in the garage) |
| **DHT11 Sensor** | It measures temperature and humidity |
| **Photoresistor sensor** | It measures the light intensity |
| **camera** | It detects the badge of the car to open the entry gate |
| **RFID sensor** | Sense RFID card to open the leave gate, also it detects robbery |

**Actuators**

|  |  |
| --- | --- |
| **Actuators** | **Functionality** |
| **Led** | **Turn on when photoresistor value is below 750** |
| **2 Buzzers** | **One buzzer for detecting fire.**  **One buzzer for detecting robbery.** |
| **2 servomotors** | **To open / close entry/leave gates** |

**System architecture**

**Mobile App Interaction**:

* Mobile app allows users to monitor real-time data and receive push notifications.
* Users can remotely control the parking through the app**.**

**Processes**:

1. **Data Flow :**

* Sensors collect data (DHT11 sensor , IR Sensor , RFID Sensor , photoresistor sensor , camera).
* A part of the data will be transferred from the Arduino to the NodeMCU
* And there is data specific to the NodeMCU itself.
* And after we collect the data, we use the broker in the NodeMCU to send the data from the sensor to the camera.
* After that, the camera starts to scan the badge and sends the result back to the NodeMCU.
* And we publish all this data from the NodeMCU to Firebase, which the application connects to, and we display the data on it.

1. **Alert Generation:**

* Arduino turn on the alarm of robbery(using the buzzer connected to the Arduino) and also send the robbery flag to the nodemcu via serial, then the node mcu update the value in the firebase so the application start sending notification and turn on the robbery alarm

**Wi-Fi:** Enables the NodeMCU to connect to the internet for communication.

**Internet:** Provides the network infrastructure for data transmission

**MQTT Broker**: Send data to the **AI object detection model** and receive the result to determine open the gate or not.

|  |  |  |
| --- | --- | --- |
| **Device**  **NodeMcu**  **Sensors:**  **Photoresistor sensor :** It measures the light intensity  **DHT11 sensor :** It measures temperature and humidity  **3IR :**1 IR for one car and 2 IR for open and close the gates  **Actuator:**  **1 Buzzer :** for detecting fire  **LED :** Turn on when photoresistor value is below 750 | **Device**  **Arduino**  **Sensors:**  **RFID sensor**: Sense RFID card to open the leave gate  **2 servo motors:** To open / close entry/leave gates  **Actuators:**  **1 Buzzer:** For robbery | **Device**  **Camera**  It detects the badge of the car to open the entry gate |

**Test Strategy / Test Cases**

we performed the **test strategy** based on **3 categories:**

1 - **Functional Testing** by Verify sensor accuracy, data transmission, and user app functionality to ensure the system detects and updates parking availability correctly.

2 - **Performance and Security Testing** by Assess system response under load, scalability, data encryption, and protection against unauthorized access.

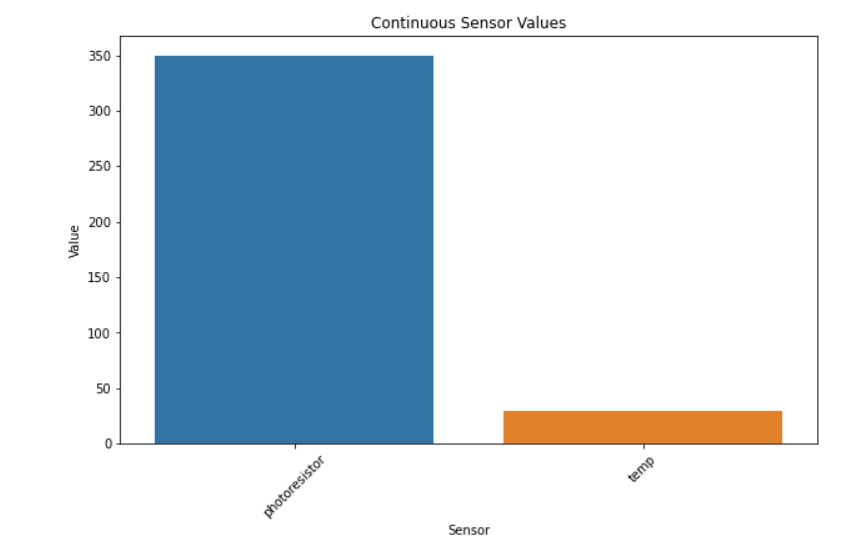
3 - **Integration and Usability Testing** by Ensure seamless communication between sensors, gateways, cloud services, and user interfaces, while maintaining an intuitive and user-friendly experience.

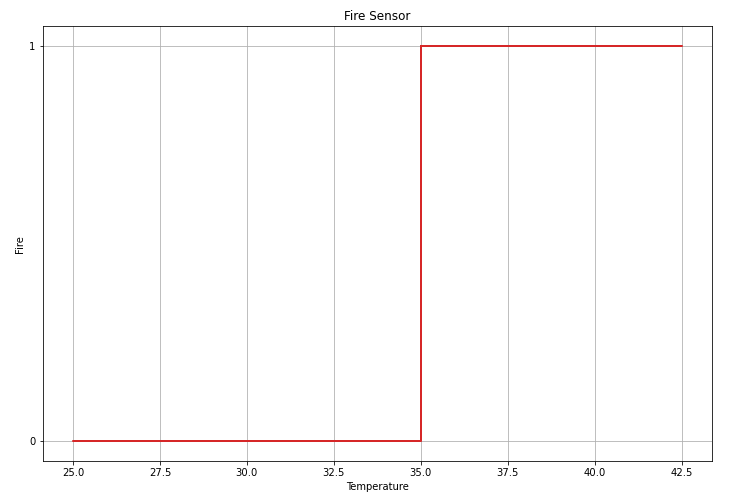
|  |  |  |
| --- | --- | --- |
| Number | Test case | Result |
| |  | | --- | | 1 | | **When a car enters the parking and don't have the badge** | **The entry gate won't open and produce a message contain "no badge detected"** |
| |  | | --- | | 2 | | **When a car enters the parking, has a badge and the parking is not full** | **The entry gate will open for the car successfully** |
| |  | | --- | | **3** | | **When a car enters the parking, has a badge but the parking is full** | **The entry gate won't open and produce a message contain "sorry parking is full"** |
| |  | | --- | | **4** | | **When the temp is >= 35 in the parking** | **The fire alarm will turn on** |
| **5** | **When the photoresistor value is below 750** | **The LEDs will turn on** |
| **6** | **When a car stays in its slot** | **The slot value will be updated in the firebase and the app as it reserved with a car** |
| **7** | **When a car leaves its slot** | **The slot will be updated in the firebase and the app as it become empty** |
| |  | | --- | | **8** | | **When a car tries to leave the parking without RFID card** | **Leave gate won't open** |
| |  | | --- | | **9** | | **When a car tries to leave the parking with wrong RFID card** | **The gate will give only 2 attempts to scan the wrong RFID card** |
| **10** | **When the user scans a wrong RFID card 3 times in a row** | **The leave gate will be locked, and the robbery alarm will turn on until the guards scan the RFID sensor with a special RFID tag used to reactivate the gate and stop the alarm** |
| **11** | **When 2 cars try to enter or leave the parking at the same time** | **It can't be done because the gates of the parking is allow only one car at a time. (The size of the gate fits only one car)** |
| **12** | **When the fire and robbery activate at the same time** | **The two alarms will be turned on at the same time too and the app will produce two alerts with two distinct notifications for the guards but only fire alarm for the normal users.** |

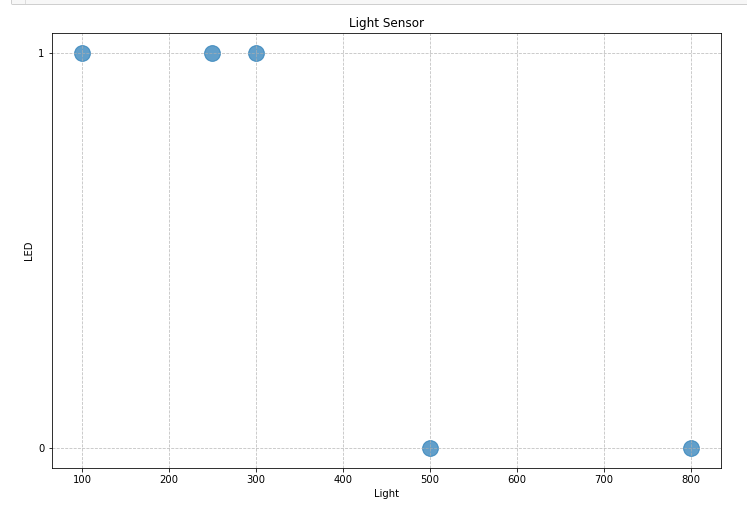
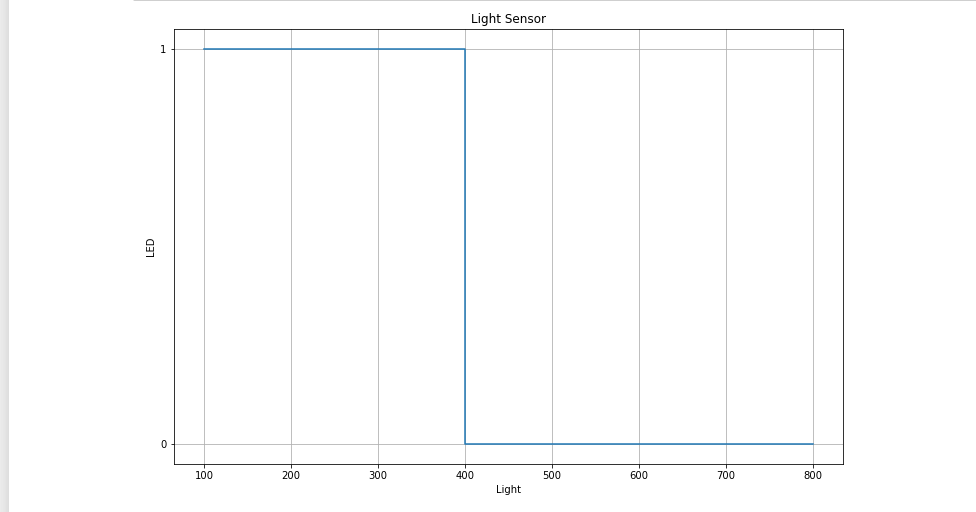
**Visualization of Data**

A graph of different colored rectangular bars

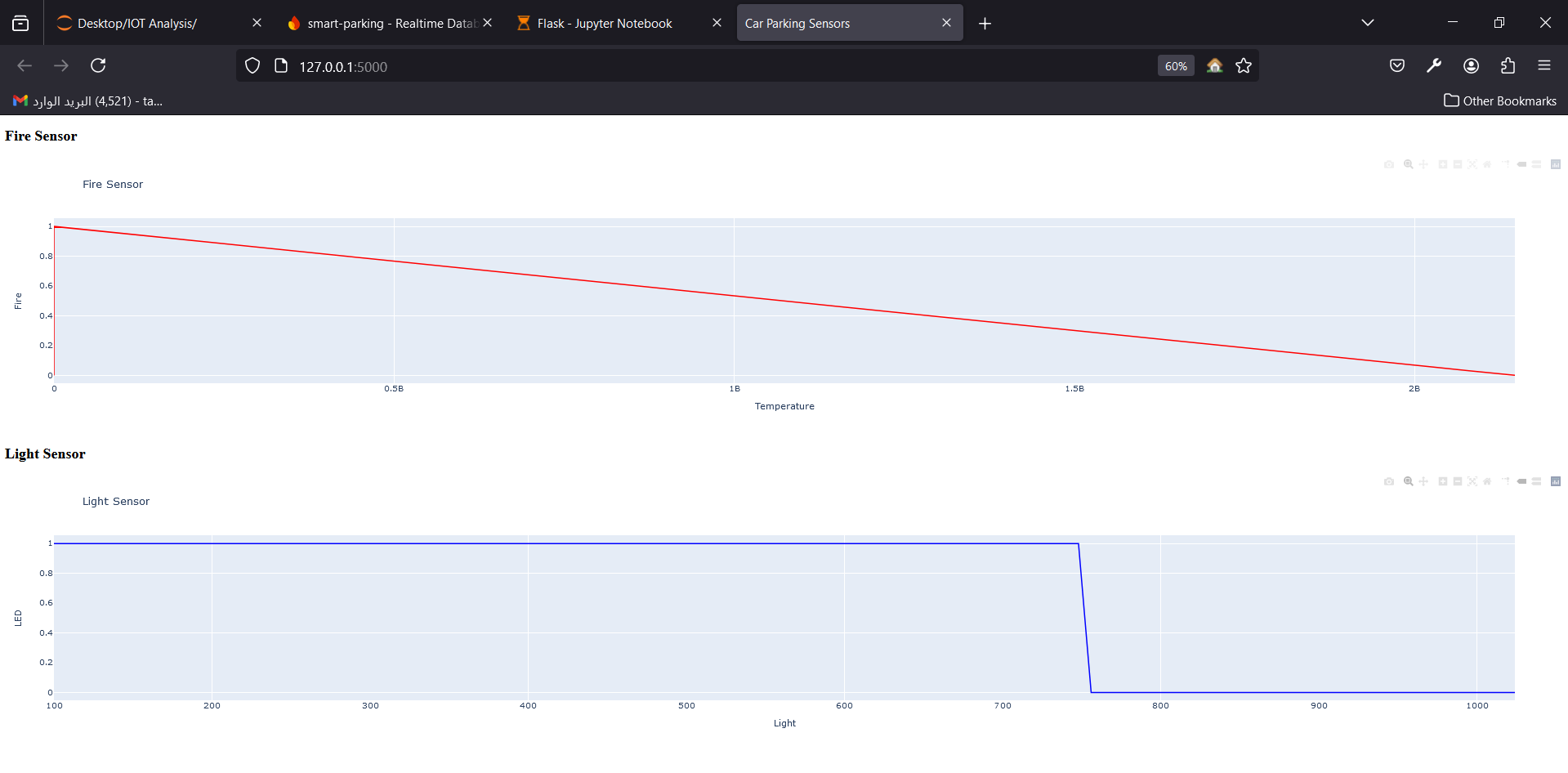
Description automatically generated with medium confidence

****

****

****

**Dashboard using Flask**



**Future Work**

– in the future we can put a camera inside the parking so the users can monitor their cars anytime anywhere using our app they only need an internet access to do that.

– we can put an abnormal motion detection ai model that run on a camera inside the parking to detect any strange or unusual movements occurring inside the parking.

– When implementing this idea on a larger scale or in the real world, the camera placed at the exit gate can recognize the car's license plate numbers. Therefore, in the event of an attempted car theft, we can identify the owner of the car and then automatically contact them through our application.

- **Enhanced Mobile application by** Add features like real-time navigation to the nearest available parking space, in-app payments, and reservation of parking spots.

- Implement machine learning models to predict parking space availability based on historical data, events, and time of day.

- Let the parking system can communicate with other smart city systems, such as traffic management and public transportation systems.

- Share real-time parking availability data with city-wide platforms to help drivers find parking more efficiently and reduce congestion.