Technology Rationale

Overview

For this project, I needed a way to take real-world data from a sensor, process it, and show it visually so people could know which parking spots were taken. I chose tools that were either familiar to me or easy to learn, worked well together, and didn’t cost anything since I’m using AWS Free Tier and low-cost hardware.

Hardware Layer

Raspberry Pi

I picked a Raspberry Pi because it’s something I’ve worked with before, and it’s great for running scripts and handling sensor data. It can connect to the internet and run Python easily, which is what I wrote the main script in. Other options like an ESP32 are cheaper, but they’re harder to work with and wouldn’t be as easy to connect to AWS.

Arduino Uno + Hall Effect Sensor

The Hall Effect sensor was used to detect if a vehicle is present in a spot. I connected it to an Arduino Uno, which sends the data over serial to the Pi. This setup worked well for testing, and it’s reliable. I thought about using an ultrasonic sensor, but it had more noise and wasn’t as accurate for what I wanted.

Cloud & Data Layer

AWS IoT Core

I used AWS IoT Core to send messages from my Pi to the cloud. It supports MQTT, which is fast and lightweight. Plus, AWS integrates with other services I needed, like DynamoDB. The setup was a bit tricky at first (especially the certificates), but once it was working, it was super reliable.

AWS Lambda

To display the parking data on a website, I used a Lambda function to fetch the current status from the database. This way, I don’t need to manage a full server, and it stays in AWS where everything else is.

Amazon DynamoDB

All the parking spot data gets stored in DynamoDB. I liked that it updates instantly and I can query it fast. Since I only needed to track things like spot\_id, status, and timestamp, a NoSQL setup was perfect. Using a regular database would’ve been overkill and slower.

Frontend Layer

HTML/CSS Web Page

For showing the parking lot, I made a simple web page using just HTML and CSS. It pulls data from the Lambda function and displays it in a way that looks like a map. I might upgrade this to use JavaScript in the future, but for now it works well for showing if a spot is taken or not.