

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Đề tài: Thiết kế hệ thống bãi đỗ xe thông minh

Học phần: IOT và ứng dụng
Tiến sĩ

Giảng viên hướng dẫn: **TS. Phạm Ngọc Hưng**

Nhóm sinh viên thực hiện: **Nhóm 9**

Nguyễn Bá Ngọc	20173287
Nguyễn Hải Dương	20173059
Lê Văn Cường	20172987
Nguyễn Đức Quyết	20177021
Đặng Đình Thái	20173357

Hà Nội, ngày 22 tháng 01 năm 2021

Mục lục

Phần I.	Tổng quan	6
1.	Đặt vấn đề	6
2.	Mục tiêu đặt ra	6
Phần II.	Thiết kế hệ thống	7
1.	Nguyên tắc hệ thống	7
2.	Biểu đồ tổng quan hệ thống	7
Phần III.	Thiết kế từng thành phần	10
1.	Sensor cảm biến vật cản (XKC-TD200-V12).....	10
1.1.	Giới thiệu	10
1.2.	Lắp đặt	11
2.	ESP8266 Node MCU.....	12
2.1.	Giới thiệu	12
2.2.	Lắp đặt	13
3.	Camera (DS-2CD4A26FWD-IZH của Hikvision)	13
3.1.	Giới thiệu	13
3.2.	Lắp đặt	16
4.	Raspberry Pi 4 Model B 2019 – 8GB RAM	17
4.1.	Giới thiệu	17
4.2.	Lắp đặt	18
5.	Màn hình	19
5.1.	Giới thiệu	19
5.2.	Lắp đặt	19
6.	Web server	21
Phần IV.	Công nghệ và thuật toán sử dụng	22
1.	MQTT Broker	22
2.	Module nhận diện biển số xe	22
2.1.	Giải quyết bài toán.....	23
2.2.	Các bước thực hiện	23
3.	Web Sever.....	24
3.1.	Ngôn ngữ lập trình python.....	24
3.2.	Django framework	24
4.	Database	26

4.1. MySQL	26
4.2. Elasticsearch	26
4.3. Redis	27
5. Triển khai hệ thống với Docker	27
Phần V. Hình ảnh demo.....	28
1. Module giám sát vị trí bãi đỗ xe	28
2. Module nhận diện biển số xe	29
3. Giao diện website.....	30
Phần VI. Kết luận và hướng phát triển	34
1. Các kết quả đạt được.....	34
2. Định hướng phát triển trong tương lai	34

Lời nói đầu

Cùng với tiến trình đô thị hóa là sự tăng lên mạnh mẽ của dân cư tại các thành phố lớn. Theo đó áp lực về giao thông đang đè nặng lên cơ sở hạ tầng hiện có. Tìm chỗ trống trong bãi đỗ xe ở các thành phố lớn ngày càng khó, nên rất cần các công nghệ tự động thông minh, để tiết kiệm thời gian cho các tài xế. Thêm vào đó, công việc ở những bãi đỗ xe tuy đơn giản nhưng lại yêu cầu duy trì trạng thái tập trung lâu dài, tốc độ xử lý nhanh và độ chính xác cao. Những công việc như vậy phù hợp cho máy móc đảm nhiệm, giải phóng sức lao động của con người. Để giải quyết vấn đề trên, nhóm em quyết định chọn đề tài “Thiết kế kiến trúc hệ thống Bãi đỗ xe thông minh” để phát triển trong học phần IoT và ứng dụng.

Trong quá trình thực hiện, nhóm em đã học hỏi được rất nhiều điều, phát triển được nhiều kỹ năng và thu được một số kết quả nhất định. Mỗi thành viên đã phát triển kỹ năng lập trình và debug, kiến thức về IoT nói chung cũng như kiến thức về MQTT Broker, AI và học máy, lập trình website, triển khai hệ thống qua docker nói riêng đều đã tăng thêm sau khi hoàn thành đề tài này.

Do thời gian thực hiện đề tài có hạn, cộng thêm những hạn chế về mặt kiến thức, kỹ năng và kinh nghiệm, vì vậy nên đề tài không thể tránh khỏi những thiếu sót. Vì vậy, nhóm em hy vọng sẽ nhận được ý kiến đóng góp từ thầy giáo để có thể hoàn thiện hơn nữa sản phẩm này.

Nhóm em xin gửi lời cảm ơn chân thành tới thầy giáo – TS. **Phạm Ngọc Hưng** vì đã giảng dạy, hướng dẫn, giúp đỡ và giải đáp những vấn đề phát sinh của nhóm trong quá trình thực hiện đề tài này.

Phần I. Tổng quan

1. Đặt vấn đề

Hiện nay, với các bãi đỗ xe thông thường dành cho ô tô thì mất khá nhiều thời gian ở khâu đi tìm vị trí đậu xe, hoặc xảy ra tình trạng xe vào bãi nhưng không còn vị trí trống. Tình trạng này gây mất thời gian và phiền toái cho phương tiện cũng như người quản lý bãi xe. Và khi phải tìm bãi đỗ khác, người dùng ô tô phải mất từ 300-1.000m chạy tìm bãi đỗ trên đường, vừa hao nhiên liệu - gây ô nhiễm môi trường vừa tốn thời gian, tăng phương tiện lưu thông trên đường - góp phần gây kẹt xe... Rất có hại. Thêm nữa, đối với một bãi đỗ mới, người cần gửi không biết giá trông xe trong khoảng bao nhiêu, thời gian bãi giữ xe hoạt động cũng như không chắc về số chỗ trong bãi có còn không.

Vấn đề nữa là người dùng sẽ cảm thấy khó chịu khi hằng ngày đi vào một bãi đỗ xe với hàng nghìn chỗ đỗ như nhau, rồi mù không có kiểm soát. Có những phiền toái như mỗi ngày gửi xe ở một nơi có quá nhiều xe đi lại, mang tới sự khó chịu khi phải chờ tới lượt lấy vé lúc vào hoặc thanh toán để ra khỏi bãi đỗ.

2. Mục tiêu đặt ra

Thiết kế hệ thống Bãi đỗ xe thông minh, cho phép người dùng tìm thấy chính xác vị trí các điểm đỗ xe của hệ thống xung quanh vị trí người dùng trong thời gian thực bằng ứng dụng website.

Kiểm soát số lượng chỗ trống trong bãi xe theo thời gian thực, giúp người dùng thay vì đi vòng quanh khu vực tìm kiếm một địa điểm mở, có thể lái xe trực tiếp đến các khu vực có nhiều chỗ đậu xe và tránh những khu vực không có bãi đậu xe theo hướng dẫn trên website.

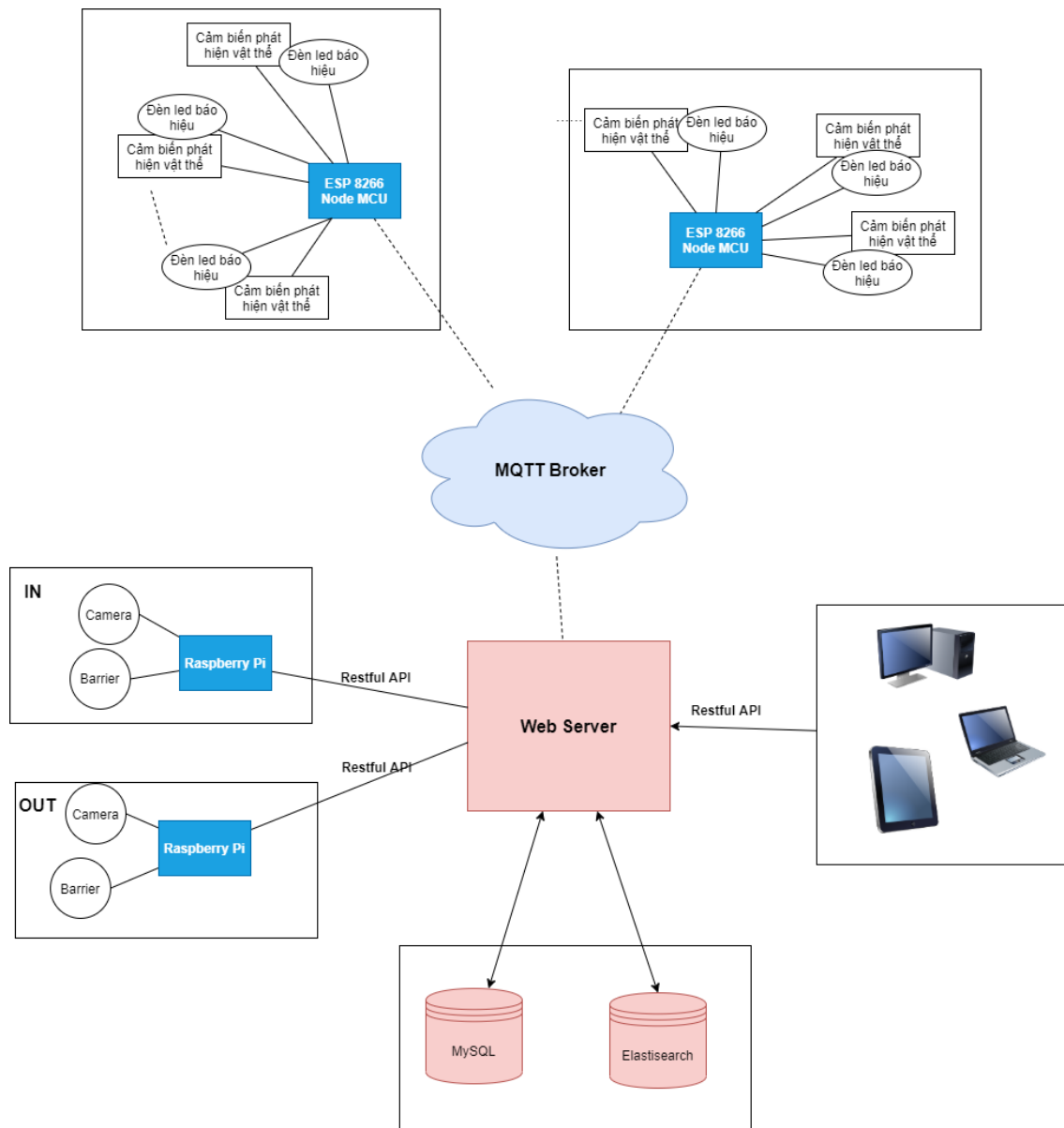
Xử lý việc ra vào bến xe một cách tự động bằng hệ thống camera kết hợp module AI nhận diện biển số và thẻ từ, tiết kiệm thời gian ra vào bãi xe. Nâng cao năng suất và tính chính xác của dịch vụ. Bên cạnh đó, giải phóng sức lao động của con người ở bãi đỗ xe.

Phần II. Thiết kế hệ thống

1. Nguyên tắc hệ thống

- Hệ thống giải quyết được các bài toán khó khăn đã đặt ra.
- Hệ thống có khả năng vận hành lâu dài và liên tục, phục hồi nhanh sau sự cố.
- Dễ dàng sử dụng, cho phép người dùng làm việc mà không bị phụ thuộc.
- Đảm bảo khả năng mở rộng.

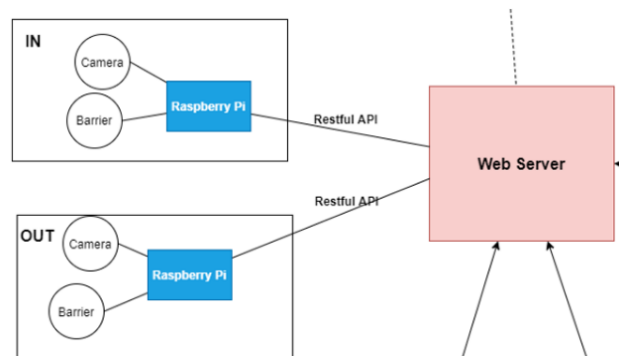
2. Biểu đồ tổng quan hệ thống



Sơ đồ tổng quan hệ thống

Hệ thống được chia thành 3 phần chính:

❖ Module giám sát vào ra bãi đỗ:



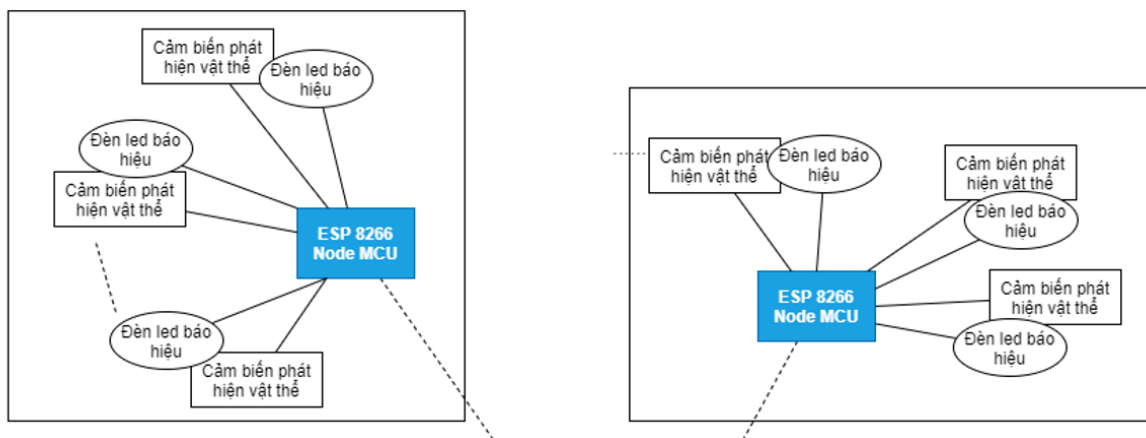
Sơ đồ module bãi đỗ

Chịu trách nhiệm kiểm tra xe vào ra bãi đỗ, lấy ảnh biển số thông qua camera, sau đó ảnh được đưa qua một modul AI xử lý ảnh để chuyển thành chuỗi biển số. Tiếp theo sẽ gọi API gửi chuỗi biển số này đến server để kiểm tra xe có thể ra/vào bãi hay không.

Bao gồm các thành phần:

- Máy tính nhúng Raspberry Pi được triển khai modul AI xử lý ảnh để điều khiển luồng hoạt động
- Camera để chụp ảnh biển số
- Barrier (thanh chắn) được điều khiển bởi Raspberry Pi

❖ Module giám sát trạng thái của bãi đỗ



Sơ đồ module giám sát trạng thái bãi đỗ

Là hệ thống các cảm biến và đèn báo cung cấp các thông tin về trạng thái của vị trí đỗ cũng như trạng thái của toàn bộ bãi đỗ. Các cảm biến này được điều khiển bởi các thiết bị nhúng là

ESP8266, các thông tin trạng thái sẽ được thiết bị điều khiển gửi lên một broker mqtt, khi đó server sẽ sử dụng các dữ liệu này để hiển thị trạng thái bản đồ bãi đỗ lên giao diện.

Ngoài việc gửi các dữ liệu trạng thái lên cho server, các thiết bị này cũng có thể nhận và thực hiện các yêu cầu điều khiển từ server thông qua broker MQTT.

❖ Module webserver cung cấp giao diện người dùng và quản lý

Với người dùng bình thường, webserver cung cấp các chức năng chính như đăng nhập, đăng kí, đăng kí xe, theo dõi trạng thái bãi đỗ.

Bên cạnh đó còn có giao diện cho người quản lý, một số chức năng chính như theo dõi bãi đỗ, theo dõi lịch sử vào ra, theo dõi điều khiển hoạt động của các thiết bị.

Phần III. Thiết kế từng thành phần

1. Sensor cảm biến vật cản (XKC-TD200-V12)

1.1. Giới thiệu

XKC-TD200-V12 là một công tắc cảm biến không tiếp xúc thông minh, sử dụng nguyên lý phản xạ khuếch tán ánh sáng hồng ngoại lên vật thể. Khi người hoặc vật thể đi vào phạm vi phát hiện ngay trước mặt cảm biến, cảm biến sẽ ngay lập tức tiếp nhận, từ đó điều khiển tín hiệu đầu ra.

Sau khi người hoặc vật thể rời khỏi phạm vi cảm biến, nó sẽ ngay lập tức tiếp nhận là không có vật thể và tín hiệu đầu ra được xuất để đạt được mục đích điều khiển, phục vụ nhiều mục đích khác nhau của người dùng.

❖ **Nhiệm vụ trong hệ thống:** Cảm biến sự xuất hiện của xe trong ô đồ.

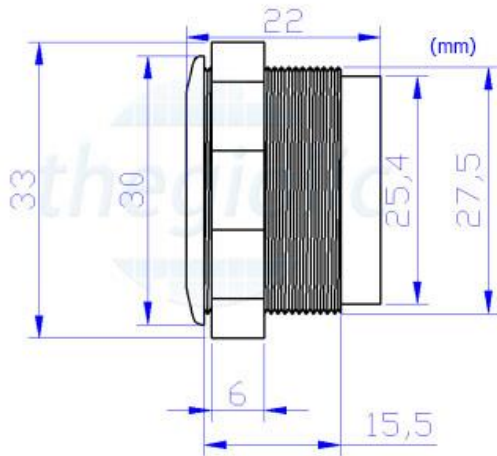


Hình ảnh minh họa sensor cảm biến vật cản (XKC-TD200-V12)

❖ **Thông số kỹ thuật:**

- Điện áp làm việc: 12VDC
- Khoảng cách cảm biến: 0-200cm
- Góc cảm biến: 30-40 °
- Phương pháp cảm ứng: hồng ngoại
- Dòng tĩnh: <3.5mA
- Dòng tải: <100mA
- Dòng tiếp điểm: 2A
- Nhiệt độ làm việc: -10 - 60 °C

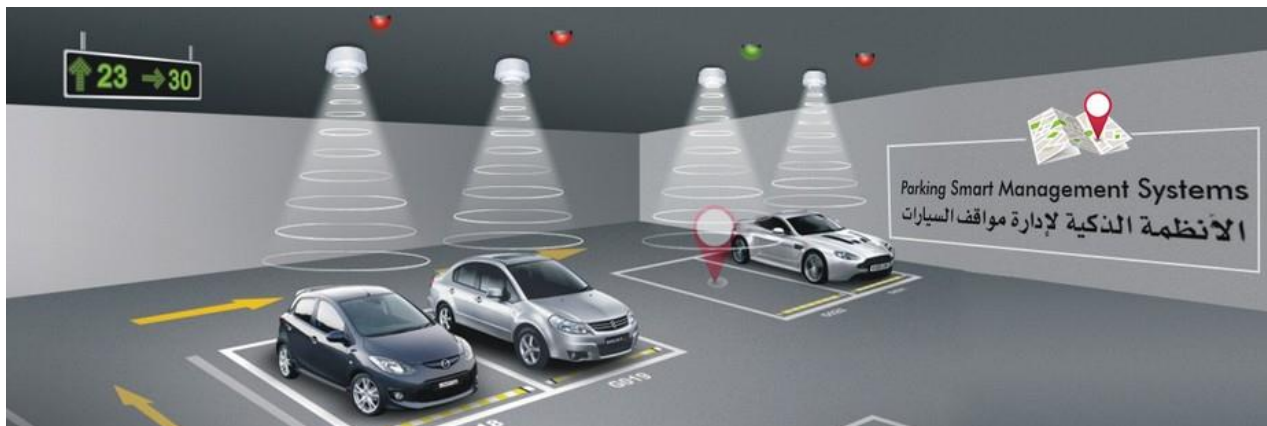
- Chiều dài cáp: 30cm
- Mức độ bảo vệ: IP67
- ❖ **Sơ đồ dây:**
 - Màu nâu: VCC
 - Màu xanh dương: GND
 - Màu đen: Tín hiệu



Ảnh minh họa thiết kế XKC-TD200-V12

1.2. Lắp đặt

- Lắp đặt tại vị trí phía trên đỉnh của từng ô đỗ (mỗi sensor sẽ theo dõi 1 ô đỗ).



Ảnh minh họa lắp đặt sensor cảm biến

2. ESP8266 Node MCU

2.1. Giới thiệu

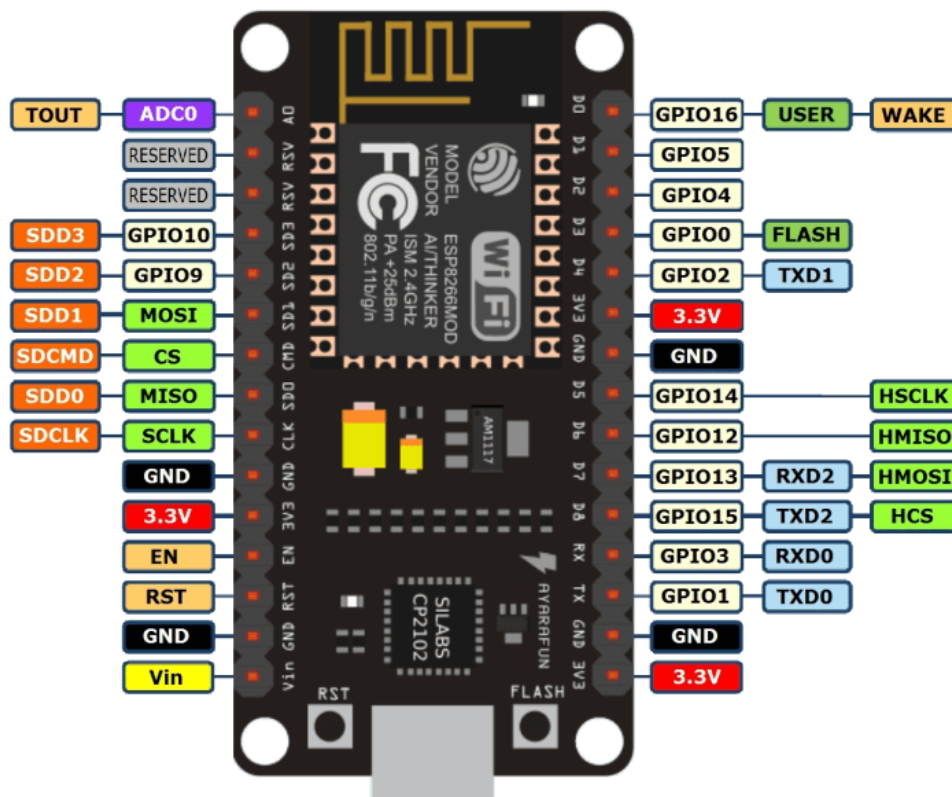
ESP8266 là dòng chip tích hợp Wi-Fi 2.4Ghz có thể lập trình được, rẽ tiền được sản xuất bởi một công ty bán dẫn Trung Quốc: Espressif Systems.

Được phát hành đầu tiên vào tháng 8 năm 2014, đóng gói đưa ra thị trường dạng Mô đun ESP-01, được sản xuất bởi bên thứ 3: AI-Thinker. Có khả năng kết nối Internet qua mạng Wi-Fi một cách nhanh chóng và sử dụng rất ít linh kiện đi kèm. Với giá cả có thể nói là rất rẻ so với tính năng và khả năng ESP8266 có thể làm được.

ESP8266 có một cộng đồng các nhà phát triển trên thế giới rất lớn, cung cấp nhiều Module lập trình mã mở giúp nhiều người có thể tiếp cận và xây dựng ứng dụng rất nhanh.

Hiện nay tất cả các dòng chip ESP8266 trên thị trường đều mang nhãn ESP8266EX, là phiên bản nâng cấp của ESP8266.

- ❖ **Nhiệm vụ trong hệ thống:** xử lý tín hiệu nhận được từ sensor và truyền dữ liệu qua giao thức MQTT tới server dựa trên kết nối wifi
- ❖ **Sơ đồ chân và sơ đồ khối:**



❖ **Thông số kỹ thuật:**

- 32-bit RISC CPU : Tensilica Xtensa LX106 running at 80 MHz
- Hỗ trợ Flash ngoài từ 512KiB đến 4MiB
- 64KBytes RAM thực thi lệnh
- 96KBytes RAM dữ liệu
- 64KBytes boot ROM
- Chuẩn wifi IEEE 802.11 b/g/n, Wi-Fi 2.4 GHz
- Tích hợp TR switch, balun, LNA, khuếch đại công suất và matching network
- Hỗ trợ WEP, WPA/WPA2, Open network Tích hợp giao thức TCP/IP
- Hỗ trợ nhiều loại anten 16 chân GPIO
- Hỗ trợ SDIO 2.0, UART, SPI, I²C, PWM, I²S với DMA
- 1 ADC 10-bit
- Dải nhiệt độ hoạt động rộng : -40C ~ 125C

2.2. Lắp đặt

- Cứ 6 chuồng đồ sẽ được lắp đặt 1 mạch ESP8266. Vị trí lắp tại trần để có thể tiết kiệm vị trí và tiết kiệm đường dây.

3. Camera (DS-2CD4A26FWD-IZH của Hikvision)

3.1. Giới thiệu

Dòng máy ảnh thông minh siêu nhạy sáng ANPR series có thể bắt xe và nhận biết biển số xe khác nhau các quốc gia và khu vực trong môi trường ánh sáng mờ

DS-2CD4A26FWD-IZH cung cấp độ phân giải cao, 120dB WDR, Auto-iris, PoE, Defog, EIS, 3D DNR, v.v., để đáp ứng nhiều ứng dụng

- ❖ **Nhiệm vụ trong hệ thống:** Chụp và gửi hình ảnh về Raspberry Pi để xử lý (1 ảnh/1s).



Ảnh minh họa camera

❖ **Mô tả tính năng:**

- Smart

- + Smart Recording : hỗ trợ ghi hình cạnh (edge recording) và dual-VCA
- + Mã hóa thông minh: hỗ trợ tốc độ bit thấp, độ trễ thấp, ROI tăng cường mã hóa

- Image

- + 1920×1080 @ 60 khung hình/giây
- + Hỗ trợ tự động iris, DC-drive
- + Hỗ trợ chế độ Rotate phù hợp với môi trường làm hành lang
- + Hỗ trợ cắt xén mục tiêu, chi tiết có thể được nhìn thấy với băng thông thấp
- + Cài đặt mượt mà cho các yêu cầu khác nhau về chất lượng hình ảnh và độ lưu loát
- + Hỗ trợ nén video H.264 + / H.264 / MPEG4 / MJPEG, cấu hình chất lượng video nhiều cấp + Cấu hình cơ bản / Cấu hình chính / Độ phân giải mã hóa H.264 của Cấu hình Cao.
- + Nhiều màu OSD: Đen & trắng tự điều chỉnh , tùy chỉnh
- + 120dB WDR có thể hỗ trợ cho môi trường đèn nền
- + Hỗ trợ defog, BLC, EIS, 3D DNR

- System

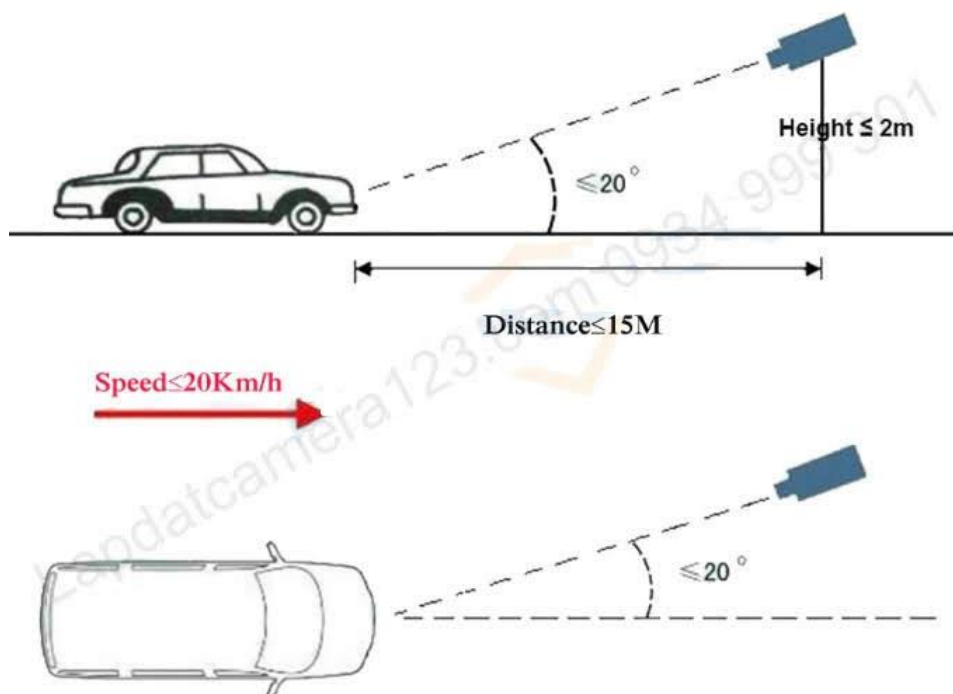
- + Hỗ trợ ONVIF (hồ sơ S / profile G), giao thức ISAPI
- + Hỗ trợ ba luồng và hỗ trợ 20 kênh xem trực tiếp cùng một lúc

- + Chống sét, bảo vệ tăng, bảo vệ tạm thời điện áp, chống tĩnh điện Interface
- + Hỗ trợ lưu trữ thẻ 128 G microSD / SDHC / SDXC chuẩn
- + Hỗ trợ cổng Ethernet 10M /100M
- + Hỗ trợ đầu ra analog CVBS (độ phân giải 4CIF)
 - Protection
- + Hỗ trợ quản lý xác thực người dùng ba cấp, ủy quyền người dùng và mật khẩu và lọc địa chỉ IP
- + Hỗ trợ chứng chỉ bảo mật dưới dạng HTTPS
- + Khóa IP của người dùng sau một số lần đăng nhập thất bại.

❖ **Thông số kỹ thuật:**

- Cảm biến sử dụng Cảm biến ánh sáng (CMOS):1/1.8" Progressive
- Độ phân giải 1920x1080@50fps/60fps
- Ống kính Zoom 2,8-12mm hoặc 8-32mm
- Zoom và tự động lấy nét
- Chuẩn nén sử dụng H.265+ / H.265 / H.264+ / H.264 / MJPEG
- Nâng cao hình ảnh 3D DNR , IRC , AGC , BLC , 120Db WDR , ROI (và dynamic tracking) , Defog , EIS , Rotate Mode
- Hồng ngoại tầm xa : 50m (2.8~12mm) / 100m(8~32mm)
- Chuẩn bảo vệ IP67
- Độ nhạy sáng 0.002 Lux @ (F1.2,AGC ON)
- Hỗ trợ PoE
- Tốc độ màn trập 1/3s đến 1/100000s
- Main Stream , Sub Stream , Third Stream
- Tính năng : Hỗ trợ 4 làn xe (Hoạt động tốt nhất ở 1-2 làn) và ở cả 2 chiều đi và đến , Danh sách trắng và đen (2048)
- Sử dụng tên miền Hik-Connect (miễn phí)
- Giao thức hỗ trợ Hikvision , ONVIF
- Lưu trữ Thẻ tối đa 128GB , NAS , ANR
- Có sườn cho camera
- Kích thước : 100 × 103.9 × 311.8 mm
- Cân nặng : 2075 g

Installation Instructions



Ảnh minh họa camera nhận diện biển số

3.2. Lắp đặt

- Camera được lắp đặt tại vị trí cổng ra và cổng vào của bãi đỗ xe



Ảnh minh họa lắp đặt camera

4. Raspberry Pi 4 Model B 2019 – 8GB RAM

4.1. Giới thiệu

Raspberry Pi 4 có hình dáng bên ngoài vẫn nhỏ bé như các phiên bản trước, tuy nhiên cấu trúc bo mạch thì đã thay đổi hoàn toàn. Vì vậy bạn không thể sử dụng vỏ case của các phiên bản trước đó cho phiên bản 4 này. Bên cạnh đó, hãng cũng có thông báo về việc phải sử dụng nguồn 5V – 3A trở lên cho bo mạch Raspberry Pi 4.

❖ Nhiệm vụ trong hệ thống:

- Xử lý hình ảnh nhận được từ camera và gửi dữ liệu lên sever.
- Nhận message từ sever để điều chỉnh barrier.

❖ Thông số kỹ thuật:

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM: 8GB LPDDR4-2400 SDRAM
- Wifi chuẩn 2.4 GHz và 5.0 GHz IEEE 802.11ac. Bluetooth 5.0, BLE
- Cổng mạng Gigabit Ethernet
- 2 cổng USB 3.0 và 2 cổng USB 2.0
- Chuẩn 40 chân GPIO, tương thích với các phiên bản trước
- Hỗ trợ 2 cổng ra màn hình chuẩn Micro HDMI với độ phân giải lên tới 4K
- Cổng MIPI DSI
- Cổng MIPI CSI
- Cổng AV 4 chân
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Khe cắm Micro-SD cho hệ điều hành và lưu trữ
- Nguồn điện DC 5V – 3A DC chuẩn USB-C 5V DC via GPIO header (minimum 3A*)
- Hỗ trợ Power over Ethernet (PoE) (yêu cầu có PoE HAT)



Ảnh minh họa Raspberry Pi 4

4.2. Lắp đặt

- Lắp đặt ở cổng ra và cổng vào của bãi đỗ. Mỗi cổng ứng với 1 máy.



Ảnh minh họa lắp đặt Raspberry Pi 4

5. Màn hình

5.1. Giới thiệu

- ❖ **Nhiệm vụ trong hệ thống:** Hiển thị danh sách các vị trí đỗ xe còn trống, thời gian đỗ, tiền phí.
- ❖ **Thông số kỹ thuật bảng led:**

Serial	Name	Specification
1	Pixel	7.62
2	Pixel of LED	16x96dot
3	LED	semi-outdoor LED module
4	Resolution	17222dots/m2
5	LED Brightness	>1000cd/m2
6	Driver for scanning	1/16
7	Bảng led hiển thị LED view angle	160
8	LED View Distance	about 100m
9	Voltage	AC110V-240V
10	Power	5V20A or 5V40A
11	Max Power	5V40A
12	Temperature	-35°C~ +85°C
13	Humidity	≤90~95%
14	Net Weight	9.5kgs
15	Bảng led hiển thị LED PCB Coating	waterproof

5.2. Lắp đặt

- Led vào: lắp đặt tại vị trí cổng vào hiển thị vị trí đỗ xe còn trống trong bãi



Ảnh minh họa led vào

- Led ra: Lắp đặt tại vị trí cổng ra hiển thị thời gian đỗ, tiền phí



Ảnh minh họa led ra

- Màn hình hiển thị sơ đồ, vị trí trống của bãi đỗ xe: Lắp đặt tại trên trần của từng khu vực tại vị trí người lái dễ theo dõi hoặc tại vị trí cổng vào.
- Led ngoài trời (Không bắt buộc): Lắp đặt ngoài bãi trước khi vào cổng đỗ, hiển thị số xe trống trong bãi



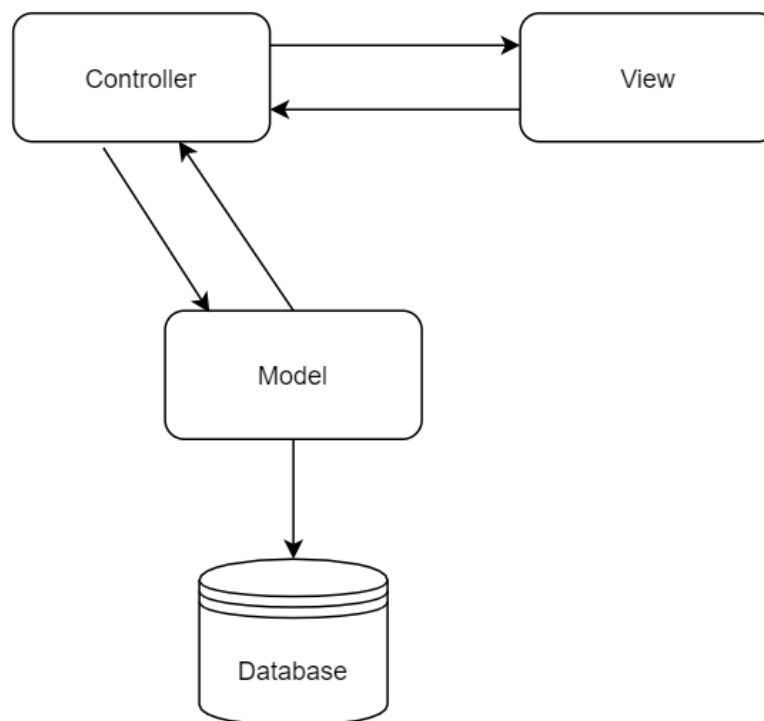
Ảnh minh họa led ngoài trời

6. Web server

Web server được xây dựng theo mô hình MVC.

MVC (Model – View – Controller) là một mẫu thiết kế được sử dụng phổ biến trong phát triển phần mềm mà ở đó chương trình được chia logic thành 3 thành phần có sự kết nối với nhau:

- ❖ **Model:** là thành phần trung tâm của mẫu thiết kế. Nó là các cấu trúc dữ liệu độc lập với giao diện người dùng, trực tiếp quản lý dữ liệu, các nghiệp vụ và quy tắc của ứng dụng
- ❖ **View:** là nơi chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh... (các file html, css, javascript) nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.
- ❖ **Controller :** là nơi tiếp nhận các yêu cầu xử lý được gửi từ người dùng và chuyển nó thành các lệnh hay thông điệp cho model hay view



Ảnh minh họa mô hình MVC

❖ Tương tác giữa các thành phần:

- Model có trách nhiệm quản lý dữ liệu của ứng dụng, tiếp nhận các dữ liệu đầu vào từ controller.

- View đảm nhiệm việc trình bày, biểu diễn dữ liệu lấy từ model theo một định dạng cụ thể.
- Controller phản hồi lại các yêu cầu từ người dùng và thực hiện tương tác với các đối tượng trong model. Controller tiếp nhận các dữ liệu đầu vào, xử lý, xác thực tính đúng đắn của dữ liệu và chuyển nó tới model.

Phần IV. Công nghệ và thuật toán sử dụng

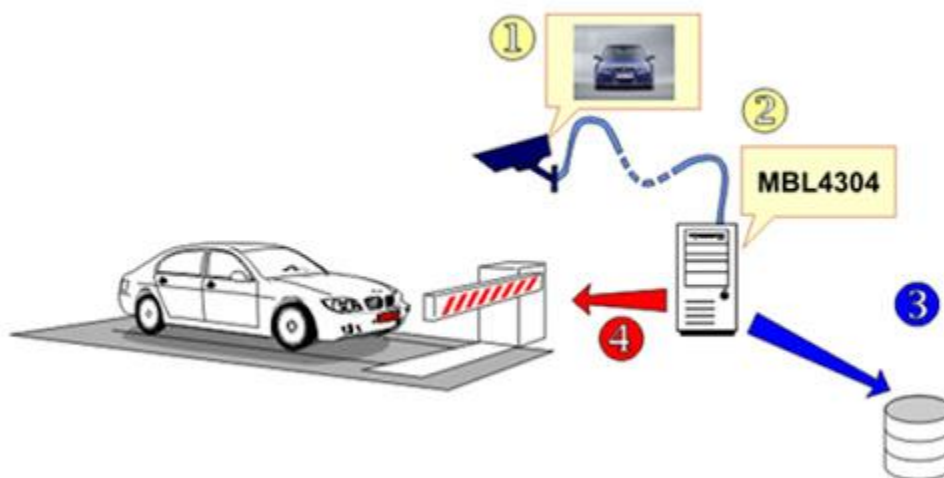
1. MQTT Broker

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình publish/subscribe (cung cấp / thuê bao), được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt.

MQTT là lựa chọn lý tưởng trong các môi trường như:

- Những nơi mà giá mạng viễn thông đắt đỏ hoặc băng thông thấp hay thiếu tin cậy.
- Khi chạy trên thiết bị nhúng bị giới hạn về tài nguyên tốc độ và bộ nhớ.
- Bởi vì giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng M2M (Machine to Machine).
- MQTT cũng là giao thức được sử dụng trong Facebook Messenger
- Nhóm chúng em dùng công nghệ này để bắn ảnh gửi từ camera lên server để detect

2. Module nhận diện biển số xe



Thị giác máy tính là một lĩnh vực đã và đang phát triển mạnh mẽ trên thế giới hiện nay. Bài toán nhận diện biển số xe là một bài toán không còn mới, ứng dụng của nó được triển khai rộng

rãi trong cuộc sống thường ngày như bãi đỗ xe thông minh. Dựa trên các lý thuyết về xử lý ảnh số, những kỹ thuật mới sử dụng deep learnings và sự ra đời của các mô hình thị giác máy tính tiên tiến trên thế giới.

2.1. Giải quyết bài toán

Thư viện sử dụng : keras, numpy, openCV3.x, scikit-learn

2.1.1. Model CNN

Convolutional neural network (mô hình mạng nơ-ron tích chập) là một mô hình deep learning tiên tiến. Giúp chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao. CNN được sử dụng rộng rãi trong các bài toán **nhận dạng đối tượng** (detection). Trong bài toán nhận diện biển số xe, sử dụng model CNN để nhận diện ký tự được tách ra từ ảnh đầu vào.

2.1.2. Mô hình YoLo Tiny V3

Là một mô hình CNN dùng cho các bài toán **phát hiện đối tượng**, ưu điểm nổi trội là tốc độ nhận diện nhanh hơn so với các mô hình cũ. Thậm chí có thể chạy tốt trên những **IoT device** như **raspberry pi**. Dùng để phát hiện, nhận diện vùng chứa biển số xe trong ảnh.

2.1.3. Thuật toán Segment

Sử dụng thư viện OpenCV, áp dụng **adaptive threshold** làm nổi bật lên vùng muốn lựa chọn và áp dụng **connected components analysis** để phân tách từng đối tượng trong vùng lựa chọn. Dùng cho nhận diện và phân tách ký tự từ biển số xe.

2.2. Các bước thực hiện

Bước 1: Xác định vùng chứa biển số xe sử dụng **YoLo Tiny V3**

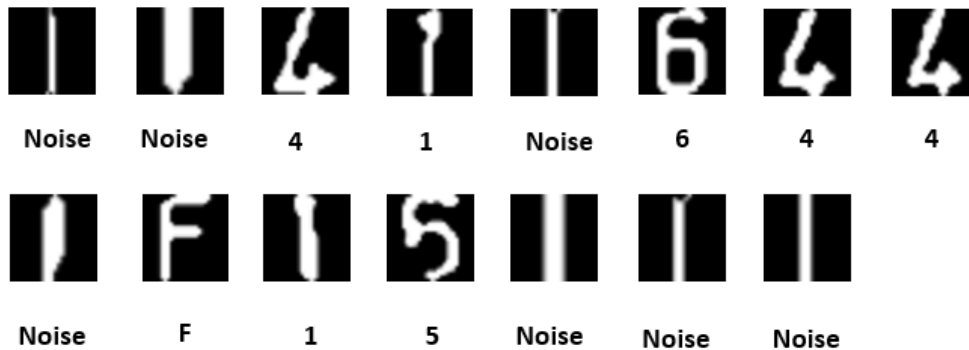


Bước 2: Sử dụng thuật toán Segment để tách từng ký tự trên biển số xe



Bước 3: Xây dựng model CNN để phân loại các ký tự (characters classification)

ALPHA_DICT = {0: 'A', 1: 'B', 2: 'C', 3: 'D', 4: 'E', 5: 'F', 6: 'G', 7: 'H', 8: 'K', 9: 'L', 10: 'M', 11: 'N', 12: 'P',
13: 'R', 14: 'S', 15: 'T', 16: 'U', 17: 'V', 18: 'X', 19: 'Y', 20: 'Z', 21: '0', 22: '1', 23: '2', 24: '3',
25: '4', 26: '5', 27: '6', 28: '7', 29: '8', 30: '9', 31: "Background"}



Bước 4: Định dạng lại biển số



3. Web Sever

3.1. Ngôn ngữ lập trình javascript trên nền tảng Nodejs

Đặc điểm nổi bật của Node.js là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho

mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của Javascript mà Node.js tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP

JSON APIs Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.

Ứng dụng trên 1 trang(Single page Application) Nếu bạn định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn. Shelling tools unix NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu suất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.

Streamming Data (Luồng dữ liệu) Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả sử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.

Ứng dụng Web thời gian thực Với sự ra đời của các ứng dụng di động & HTML 5 nên Node.js rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,...

3.2. Reactjs Framework

React (còn được gọi là Reactjs hay React.js) là một Thư viện javascript được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó cho phép những nhà phát triển web tạo ra giao diện người dùng nhanh chóng. Phần Views của Reactjs thường được hiển thị bằng việc chủ yếu dùng các component mà chứa các component cụ thể hoặc các thẻ HTML. Một trong những đặc trưng duy nhất của Reactjs là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client.

Nó cũng sử dụng khái niệm là Virtual DOM (DOM ảo). Virtual DOM tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp, nó liệt kê những thay đổi và sau đó là cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả. Điều này cho phép ta viết các đoạn code như thể toàn bộ trang được render lại dù thực tế là Reactjs chỉ render những component hay subcomponent nào thực sự thay đổi.

Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí. Chúng ta có thể viết một ví dụ đơn giản về ReactJS như sau.

4. Database

4.1. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng

So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row , MongoDB sẽ dùng các document thay cho row trong RDBMS.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

4.2. Socket.io

Là một module của NodeJs

Được xây dựng nhằm mục đích tạo ra real time NodeJS application. Socket.io cung cấp cho lập trình viên các đặc trưng như event, room và tự động phục hồi lại kết nối.

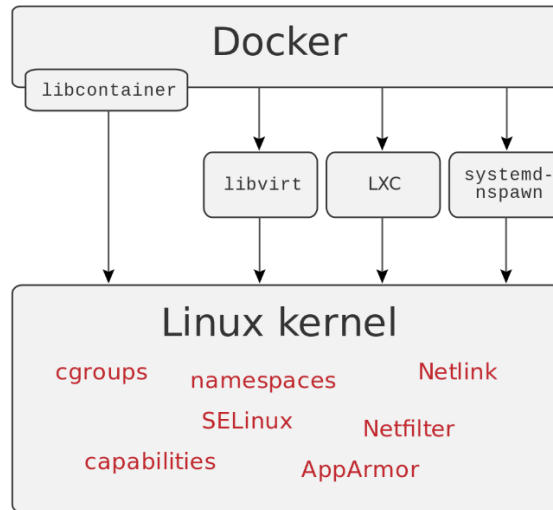
Khi chúng ta include Socket.io module vào trong ứng dụng của mình nó sẽ cung cấp cho chúng ta hai object đó là: socket server quản lý functionality phía server và socket client điều khiển functionality phía client.

Khi client muốn kết nối tới Socket.io server, nó sẽ gửi cho server một “handshake HTTP request”. Server sẽ phân tích request đó với những thông tin cần thiết trong suốt quá trình kết nối. Nó sẽ tìm cấu hình của middleware mà đã được đăng ký với server và thực thi chúng trước khi đưa ra sự kiện kết nối. Khi kết nối thành công thì connection event listener được thực thi, tạo ra một instance mới của socket có thể coi như định danh của client mà mỗi một client kết nối tới sẽ có 1 định danh.

Một module khác của Node.js là LightStreamer-adapter cũng có tạo các kết nối từ client tới server nhưng không trực tiếp mà thông qua LightStreamer Server, đó là các máy chủ theo thời gian thực và nằm ngoài tiến trình của Node.js Server

5. Triển khai hệ thống với Docker

Docker là nền tảng phần mềm cho phép ta dựng, kiểm thử và triển khai ứng dụng một cách nhanh chóng. Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là **container** có mọi thứ mà phần mềm cần để chạy, trong đó có thư viện, công cụ hệ thống, mã và thời gian chạy. Bằng cách sử dụng Docker, ta có thể nhanh chóng triển khai và thay đổi quy mô ứng dụng vào bất kỳ môi trường nào và biết chắc rằng mã của ta sẽ chạy được.



Ảnh minh họa Docker

Các container chạy trực tiếp trên nhân của máy chủ, ta có thể có nhiều container cùng chạy trên một tổ hợp phần cứng nhất định.

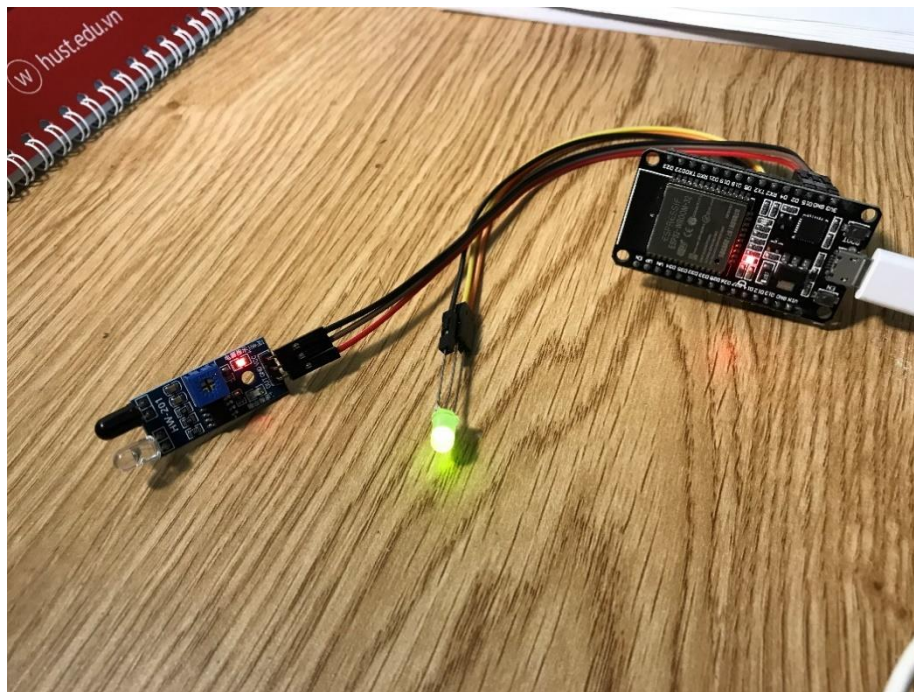
Nhóm sử dụng docker để triển khai đồng bộ các dịch vụ web, các server lưu trữ cũng như mqtt broker.

Phần V. Hình ảnh demo

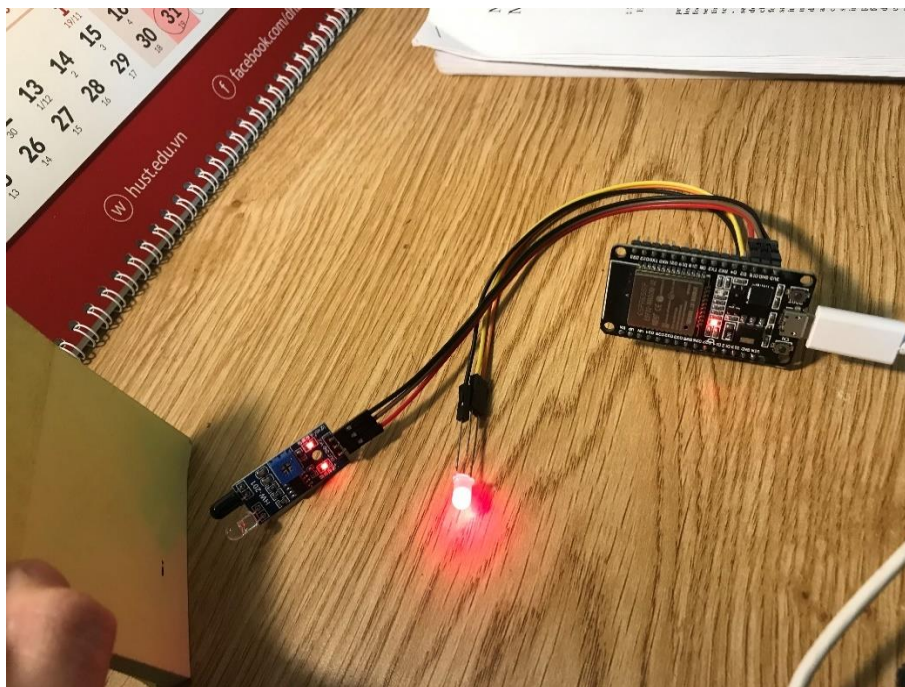
1. Module giám sát vị trí bãi đỗ xe

Hình dưới đây minh họa module giám sát trạng thái bãi đỗ xe. Khi bãi đỗ không có xe, đèn led xanh được bật để báo hiệu cho người dùng biết rằng vị trí còn trống. Ngược lại, khi bãi đỗ đã có xe đậu, đèn led màu đỏ được bật lên để người dùng có thể dễ dàng quan sát và chọn một vị trí trống khác.

Mỗi khi trạng thái của bãi đỗ thay đổi, cảm biến phát hiện vật cản sẽ gửi tín hiệu tới ESP8266. ESP8266 tiếp đó sẽ điều khiển đổi màu đèn led và gửi một message chứa thông tin về định danh của ESP8266, định danh của cảm biến (vị trí bãi xe) cùng với trạng thái của bãi đỗ lên MQTT Broker. Phía server thông qua MQTT Broker sẽ nhận được thông tin về bãi xe.



Mô phỏng bãi đỗ khi không có xe



Mô phỏng bãi đỗ khi có xe

2. Module nhận diện biển số xe

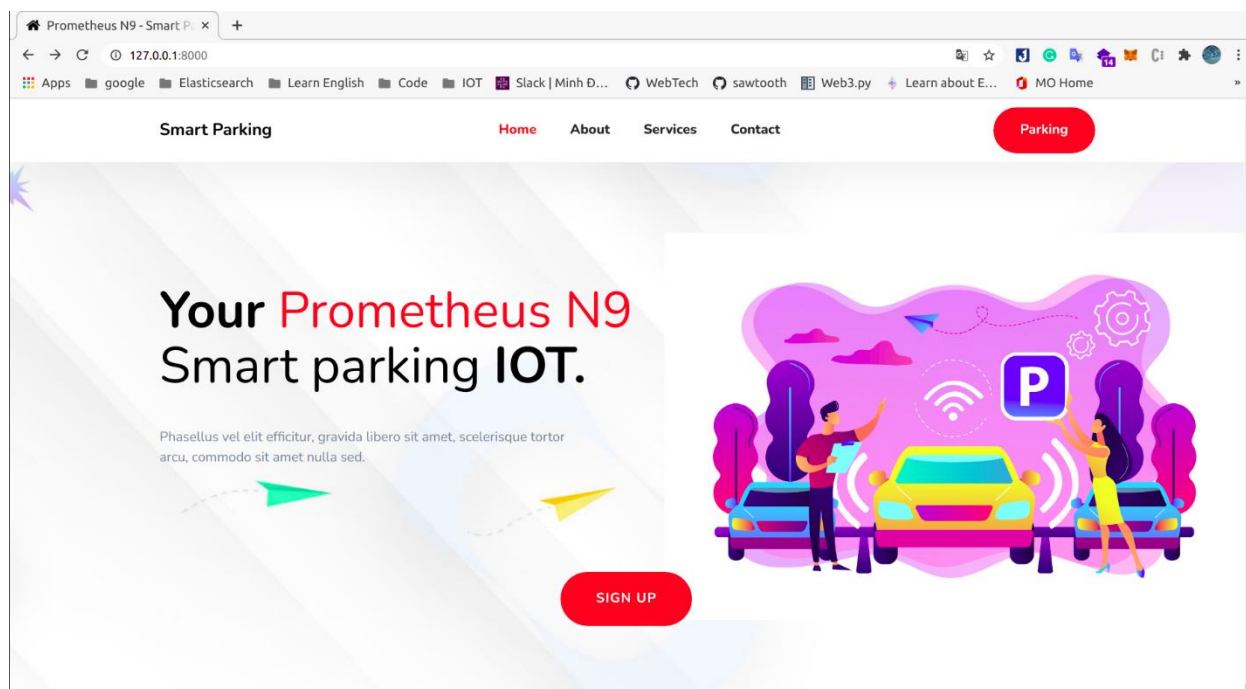


Ảnh minh họa module nhận diện biển số

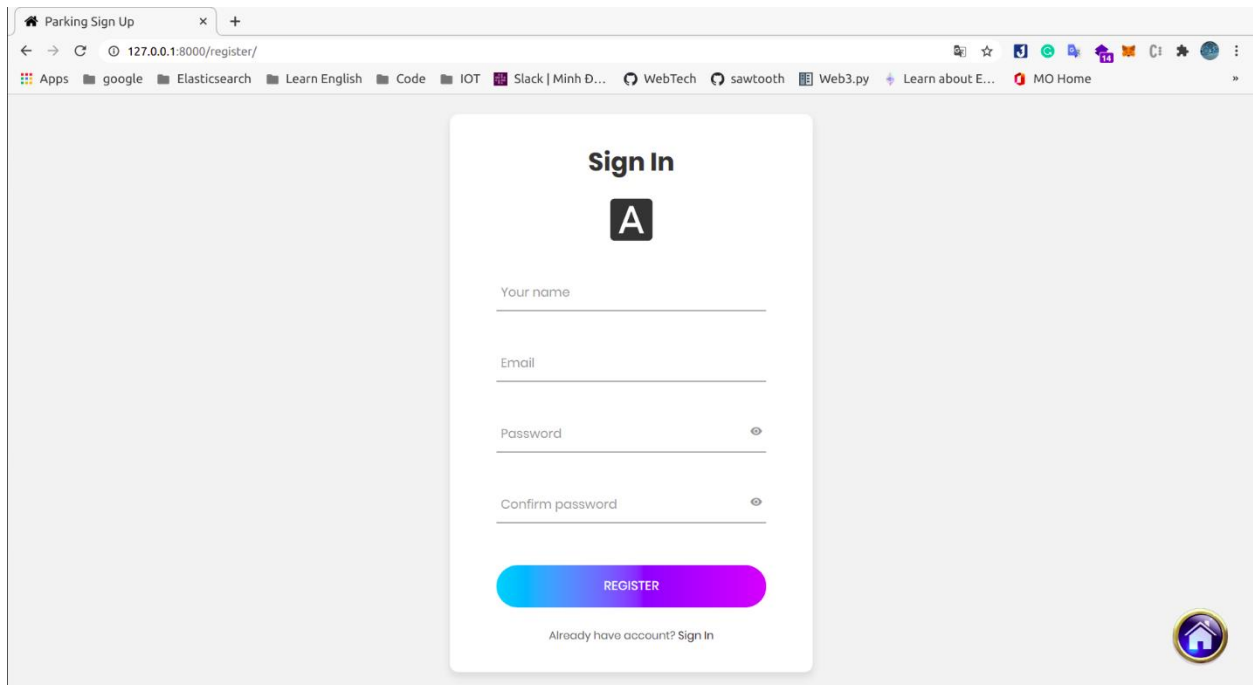


Ảnh minh họa định dạng lại biển số

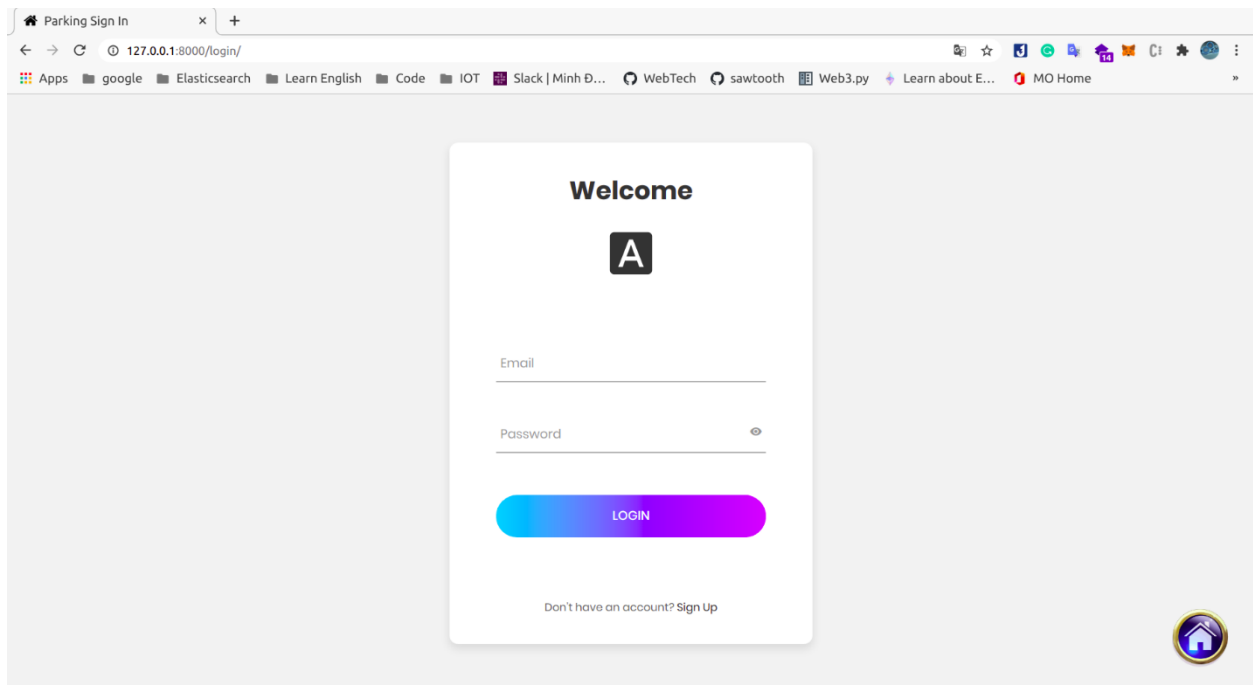
3. Giao diện website



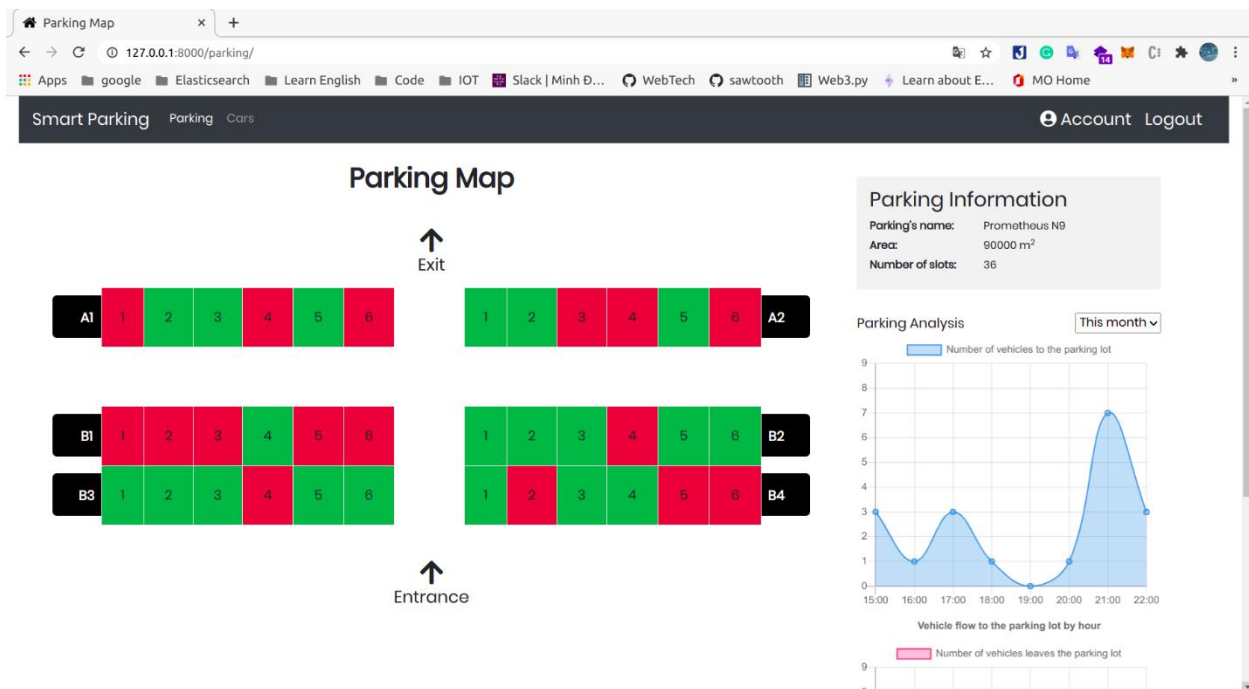
Ảnh giao diện trang chủ



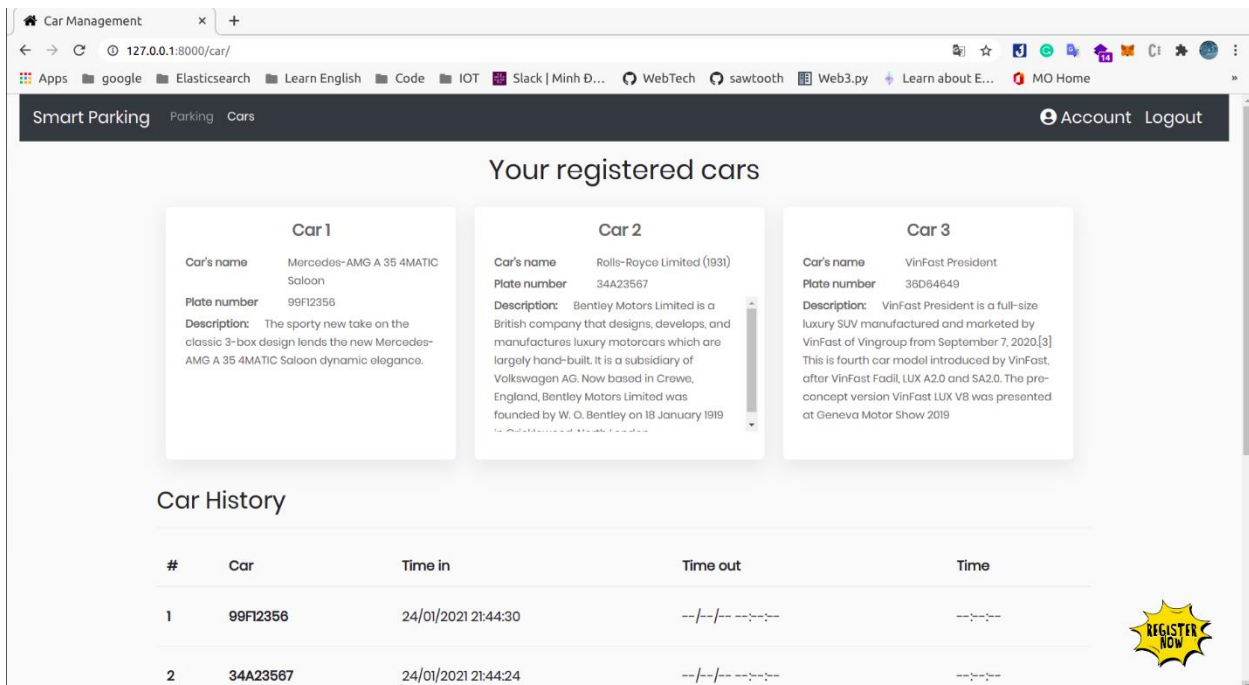
Ảnh giao diện trang đăng kí



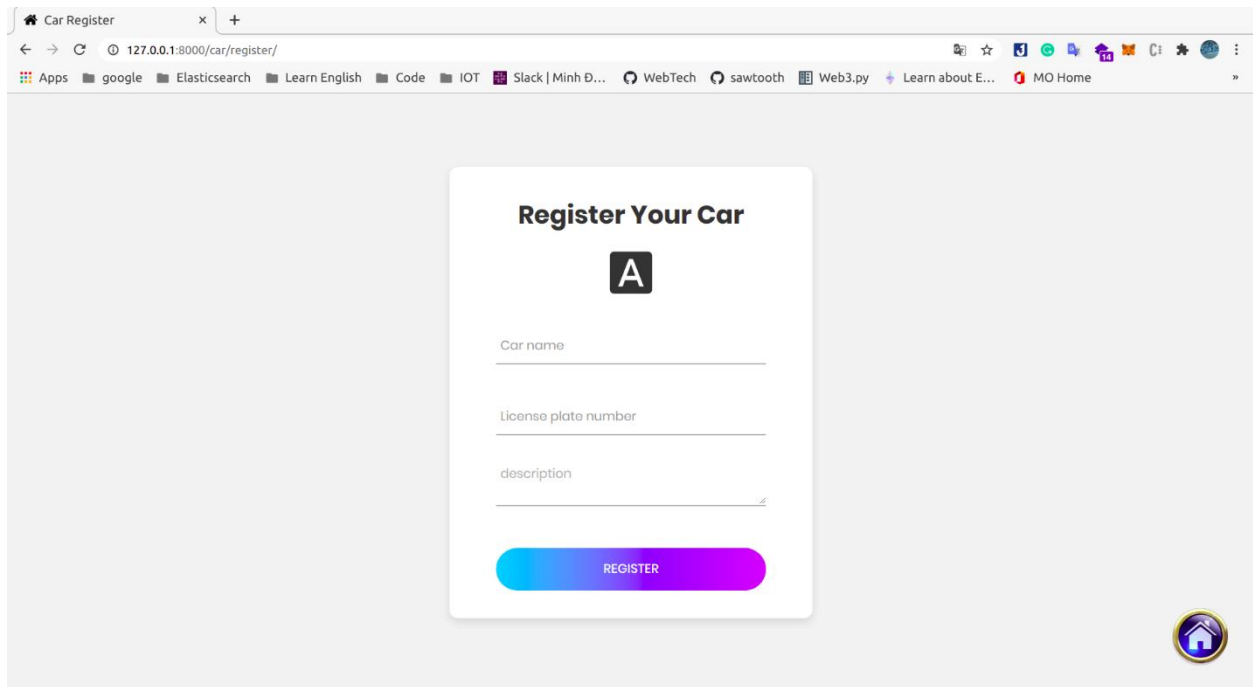
Ảnh giao diện trang đăng nhập



Ảnh giao diện sơ đồ bãi xe và thống kê



Ảnh giao diện trang quản lý xe



Ảnh giao diện trang đăng kí xe

Phần VI. Kết luận và hướng phát triển

1. Các kết quả đạt được

Về sản phẩm, nhóm đã cơ bản xây dựng và triển khai được một hệ thống điều khiển giám sát bãi đỗ xe thông minh, hệ thống có thể chạy tốt trên nhiều môi trường và thiết bị khác nhau, có giao diện thân thiện, dễ sử dụng, cung cấp những chức năng cơ bản nhất, cần thiết nhất để vận hành một bãi đỗ xe thông minh ứng dụng IOT.

Đối với các thành viên, đây là cơ hội tốt để mọi người cải thiện khả năng làm việc nhóm, tích lũy, học hỏi các kinh nghiệm, nắm bắt được quy trình xây dựng một hệ thống ứng dụng IOT trong thực tế, từ việc tìm hiểu, phân tích các yêu cầu, đưa ra giải pháp cho đến thiết kế, lựa chọn công nghệ, triển khai đồng bộ hệ thống.

Bên cạnh đó, hệ thống vẫn còn một số điểm chưa được hoàn thiện xuất phát từ những hạn chế về thời gian, kinh nghiệm, phần cứng và kinh phí. Hệ thống hiện tại chưa có chức năng trả phí gửi xe, chưa giải quyết được vấn đề an ninh khi xe ra khỏi bãi đỗ (chưa có xác thực từ chủ xe). Module nhận dạng xử lý biển số hoạt động cũng chưa thực sự tốt với các ảnh chụp biển số bị mờ, chỉ đạt độ chính xác khoảng 87%, đôi khi nhận dạng nhầm giữa “0” và “D”, “8” và “B”.

2. Định hướng phát triển trong tương lai

Trong tương lai, hệ thống cần phải giải được các vấn đề đã đặt ra ở trên, cụ thể một số hướng phát triển như sau:

- Xây dựng chức năng thanh toán, trả phí kết hợp với các tài khoản ngân hàng, ví điện tử, QR code.
- Cần có chức năng xác thực từ chủ xe khi ra khỏi bãi đỗ để tránh tình trạng xe bị trộm.
- Đánh giá và xây dựng lại module xử lý biển số xe, tăng độ chính xác và khả năng hoạt động trong những điều kiện tệ nhất.
- Xây dựng các chức năng hỗ trợ giám sát, điều khiển thiết bị, đưa ra cảnh báo, tình trạng hoạt động của các thiết bị.
- Sử dụng các thiết bị hiện đại hơn để tăng độ ổn định cũng như chính xác trong hoạt động của hệ thống.
- Lắp đặt thêm các camera theo dõi để tăng cường an ninh an toàn cho bãi xe, tránh các trường hợp xấu có thể xảy ra.

Phân công nhiệm vụ thành viên

Tên thành viên	Nhiệm vụ
Nguyễn Bá Ngọc	Webserver, triển khai hệ thống
Nguyễn Hải Dương	Module giám sát trạng thái của bãi đỗ
Lê Văn Cường	Module giám sát trạng thái của bãi đỗ
Nguyễn Đức Quyết	Module giám sát vào ra
Đặng Đình Thái	Giao diện web

Tài liệu tham khảo

- [1] Model-View-Controller, <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [2] Django documentation, <https://docs.djangoproject.com/en/3.1/>
- [3] Elasticsearch Reference, <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html>
- [4] Redis, <https://redis.io/documentation>
- [5] Docker documentation, <https://docs.docker.com/get-started/>
- [6] ESP8266 documentation, <https://arduino.esp8266.vn/>