0 -Lista de funcionários da portaria:

João Carlos Ferreira Nascimento

Matricula: 0001

-----------------------------------------------------------------------------------------------------

Laura Conceição Cardoso

Matricula: 0002

------------------------------------------------------------------------------------------------------

Anderson Pestana

Matricula: 0003

1. Cadastrar Veículo: Cadastrado

Placa : BEE4R22

CPF: 52998224725

Modelo : Palio

Nome: Falcão Cardoso Falópio

Bloco: 08 A

Apartamento: 805

-----------------------------------------------------------------------------------------------1- Cadastro de Veículo:

Placa: BCEA 4F33

CPF:098.754.855.52

Modelo: Honda Civic

Nome: Romano Pereira Carvalho

Bloco: 15 B

Apartamento: 501

-----------------------------------------------------------------------------------------------

1- Cadastro de Veículo:

Placa:

CPF:

Modelo:

Nome:

Bloco:

Apartamento:

1- Cadastro de Veículo:

Placa:

CPF:

Modelo:

Nome:

Bloco:

Apartamento:

1- Cadastro de Veículo:

Placa:

CPF:

Modelo:

Nome:

Bloco:

Apartamento:

Vamos fazer a perte visual com HTML,CSS, JAVASCRIPT

**✅ 1. Objetivo: melhorar a interface do usuário**

Atualmente, o sistema roda no terminal (modo texto). Você pode criar um **painel visual web** com:

* 📋 Formulários para cadastro de veículos e funcionários
* 🟢 Visualização das vagas ocupadas/livres
* 🕐 Relatórios de entrada/saída com datas
* 🚨 Alertas de tempo excedido
* 🔒 Login para supervisores

**🔁 2. Como conectar HTML/CSS/JS com seu código Python?**

**📌 Opção 1 – Flask (Recomendada e simples)**

Você transforma seu sistema em um mini servidor web:

* HTML e CSS criam a interface visual.
* JavaScript pode fazer requisições AJAX.
* Flask roda em Python e responde os dados (POST/GET).
* Tudo roda localmente (ou pode ser publicado em Render, PythonAnywhere, etc).

**Exemplo de fluxo:**

1. HTML exibe o botão “Estacionar veículo”
2. JavaScript envia os dados via fetch() para o Flask (POST)
3. Flask chama estacionar\_veiculo(...)
4. Retorna uma resposta (ex: “Veículo estacionado com sucesso”)
5. A tela atualiza com o status da vaga

**📌 Opção 2 – Interface Gráfica com Electron.js (avançado)**

Se quiser um aplicativo de desktop com interface web, pode usar:

* **Electron.js** com HTML/CSS/JS para a interface
* Python como backend usando sockets ou chamadas subprocess

**ETAPA 1 – Preparar o ambiente com Flask (backend em Python)**

**Objetivo:**

Criar um servidor web simples que possa receber requisições da interface (HTML/JS) e interagir com o seu código atual.

Contruido até agora

| **Item** | **Status** |
| --- | --- |
| Servidor Flask ativo (app.py) | ✅ OK |
| Rota /estacionar funcional | ✅ OK |
| HTML com formulário de cadastro | ✅ OK |
| Comunicação via fetch() (JavaScript) | ✅ OK |
| Estilo visual básico | ⚠️ Precisa melhorar |

Você está no caminho certo! Se quiser, posso gerar:

* ✅ A versão atualizada com <meta>, <label> e melhorias visuais
* ✅ Um style.css otimizado para esse HTML
* ✅ Componentização em páginas separadas

Atualização 22/06/2025.  
Corrigir o bug da hora no aquivo estacionamento.py