Certo! Com base nas suas informações e nos requisitos detalhados, vou criar a estrutura para o site da DSE Segurança Eletrônica e Serviços LTDA, focando em Python com Flask, SEO, responsividade e o tema escuro/dourado.

Aqui está um plano para o desenvolvimento do site, incluindo as descrições dos serviços, a distribuição das mídias e a estrutura técnica:

**Estrutura do Site DSE Segurança Eletrônica e Serviços LTDA**

**1. Visão Geral do Projeto**

O site da DSE será uma plataforma profissional e moderna, desenvolvida com Python (Flask) para apresentar os serviços da empresa, otimizada para SEO e com uma experiência de usuário fluida. O tema padrão será escuro com detalhes dourados, mas com a opção de alternar para um modo claro.

**2. Seções Principais do Site**

* **Página Inicial (Home):**
  + Banner de boas-vindas com o logo da DSE.
  + Breve introdução sobre a empresa e sua missão.
  + Destaque para os serviços mais procurados (com links para a seção de serviços).
  + Chamada para ação (CTA) para contato.
* **Sobre Nós:**
  + História da DSE.
  + Valores e compromissos da empresa.
  + Equipe (opcional).
* **Serviços:**
  + Lista detalhada de todos os serviços oferecidos com suas descrições.
  + Imagens e vídeos relevantes para cada serviço.
* **Portfólio/Projetos:**
  + Galeria de fotos e vídeos de projetos realizados (onde as imagens e vídeos fornecidos serão distribuídos).
* **Contato:**
  + Formulário de contato.
  + Informações de contato (telefone, e-mail, endereço).
  + Mapa de localização (se aplicável para a sede em Fortaleza).
* **Rodapé (Footer):**
  + Links para redes sociais (WhatsApp, Instagram).
  + Informações de copyright.
  + Links úteis (política de privacidade, termos de uso - se houver).

**3. Detalhes de Design e Funcionalidades**

* **Tema Padrão:** Escuro com detalhes dourados.
* **Alternância de Tema:** Botão visível para alternar entre tema escuro e claro. A preferência será salva via localStorage.
* **Botão Flutuante "Voltar ao Topo":** Um botão no canto inferior direito/esquerdo da tela que rola a página para o topo ao ser clicado.
* **Navegação:** Menu de navegação responsivo (hambúrguer em dispositivos móveis).
* **Links Sociais:** Botões para WhatsApp e Instagram flutuantes ou em destaque no cabeçalho/rodapé.
* **Otimização de Imagens e Vídeos:** Imagens comprimidas, lazy loading para carregamento sob demanda.

**4. Serviços com Descrição e Mídia Associada**

Aqui está a lista de serviços com suas descrições e a sugestão de onde as mídias fornecidas podem ser utilizadas:

**1. Eletricista Predial e Residencial**

* **Descrição:** Instalação, manutenção e reparo de sistemas elétricos em edifícios e residências, garantindo segurança e conformidade com as normas técnicas. Inclui desde a instalação de tomadas e interruptores até a montagem de quadros de distribuição e correção de falhas.
* **Mídia Sugerida:** image7.webp.jpeg (mostrando instalação elétrica).

**2. Eletricista Industrial**

* **Descrição:** Serviços especializados em instalações elétricas de grande porte para indústrias, incluindo montagem de painéis de comando, manutenção de máquinas e equipamentos, e implementação de sistemas de automação industrial, visando eficiência e segurança operacional.
* **Mídia Sugerida:** Não há uma imagem ou vídeo específico para isso nas mídias fornecidas, mas pode ser usada uma imagem genérica de ambiente industrial ou de painéis elétricos.

**3. Instalação de Antenas**

* **Descrição:** Instalação e direcionamento de antenas para TV digital, parabólicas e sistemas de comunicação, garantindo a melhor qualidade de sinal para residências e empresas.
* **Mídia Sugerida:** Não há imagem/vídeo específico.

**4. Instalação de Alarmes**

* **Descrição:** Implementação de sistemas de alarme robustos e personalizados para proteger propriedades contra invasões e acessos não autorizados, com sensores de movimento, abertura e pânico.
* **Mídia Sugerida:** image2.webp.jpeg (kit de cerca elétrica/alarme). Pode ser usada também na seção de "Segurança Eletrônica em Fortaleza".

**5. Instalação de Câmeras de Segurança (CFTV)**

* **Descrição:** Projeto e instalação de sistemas de Circuito Fechado de Televisão (CFTV), permitindo o monitoramento de ambientes internos e externos, com gravação e acesso remoto para maior segurança e controle.
* **Mídia Sugerida:** image5.webp.jpeg (câmeras de segurança). Pode ser usada também na seção de "Segurança Eletrônica em Fortaleza".

**6. Instalação de Alarmes Residenciais e Prediais**

* **Descrição:** Foco na segurança de residências e edifícios, com sistemas de alarme customizados que incluem sensores de portas/janelas, detectores de presença e integração com centrais de monitoramento. (Repetido com item 4, mas pode ter foco específico aqui).
* **Mídia Sugerida:** image2.webp.jpeg (kit de cerca elétrica/alarme).

**7. Instalação de Interfones**

* **Descrição:** Instalação e manutenção de sistemas de interfonia para condomínios, empresas e residências, facilitando a comunicação e o controle de acesso de visitantes.
* **Mídia Sugerida:** Não há imagem/vídeo específico.

**8. Automação de Portões**

* **Descrição:** Automação de portões residenciais, comerciais e industriais, oferecendo conveniência, segurança e durabilidade, com opções de motores e controles remotos.
* **Mídia Sugerida:** Não há imagem/vídeo específico.

**9. Projeto Elétrico**

* **Descrição:** Elaboração de projetos elétricos completos e detalhados para novas construções ou reformas, garantindo que a instalação seja segura, eficiente e em conformidade com as normas da ABNT.
* **Mídia Sugerida:** Não há imagem/vídeo específico.

**10. Instalação de Fio Terra**

* **Descrição:** Implementação de sistemas de aterramento elétrico (fio terra) para proteção de pessoas e equipamentos contra choques elétricos e surtos de energia, essencial para a segurança de qualquer instalação.
* **Mídia Sugerida:** image7.webp.jpeg (pode ser usada em conjunto com outros serviços elétricos).

**11. Alpinismo Industrial em Fortaleza**

* **Descrição:** Serviços especializados de acesso por corda para trabalhos em altura em ambientes industriais, como inspeções, manutenção e reparos em estruturas elevadas e de difícil acesso.
* **Mídia Sugerida:** altura.mp4 (vídeo do alpinista em silos), image4.webp.jpeg (alpinista com silos ao fundo).

**12. Pintura Predial em Fortaleza**

* **Descrição:** Pintura de fachadas e áreas externas de edifícios, utilizando técnicas de alpinismo industrial quando necessário, garantindo acabamento de alta qualidade e durabilidade.
* **Mídia Sugerida:** pintura.mp4 (vídeo dos pintores em altura), image1.webp.jpeg (pintor em prédio com vista para o mar).

**13. Manutenção Predial em Fortaleza**

* **Descrição:** Serviços de manutenção preventiva e corretiva para edifícios, incluindo reparos estruturais, hidráulicos, elétricos e de acabamento, visando a conservação e funcionalidade do imóvel.
* **Mídia Sugerida:** image1.webp.jpeg e image4.webp.jpeg (podem ilustrar trabalhos em altura para manutenção).

**14. Segurança Eletrônica em Fortaleza**

* **Descrição:** Soluções integradas de segurança eletrônica para residências, condomínios e empresas, englobando instalação e manutenção de sistemas de alarme, câmeras de CFTV, controle de acesso e cercas elétricas.
* **Mídia Sugerida:** image2.webp.jpeg (kit de cerca elétrica/alarme), image5.webp.jpeg (câmeras de segurança). image3.webp.jpeg (rack de servidor, pode ser usada para ilustrar a central de segurança ou infraestrutura de rede).

**5. Configuração Técnica (Python Flask)**

**Estrutura de Pastas:**

dse\_website/

├── app.py

├── static/

│ ├── css/

│ │ └── style.css

│ ├── js/

│ │ └── script.js

│ ├── img/

│ │ ├── logo\_dse.png

│ │ ├── image1.webp.jpeg

│ │ ├── image2.webp.jpeg

│ │ ├── ...

│ └── video/

│ ├── pintura.mp4

│ └── altura.mp4

├── templates/

│ ├── base.html

│ ├── index.html

│ ├── about.html

│ ├── services.html

│ ├── portfolio.html

│ ├── contact.html

│ └── search\_results.html (opcional, para busca)

└── venv/ (ambiente virtual)

├── requirements.txt

**requirements.txt:**

Flask

Jinja2

Flask-SEO (opcional, para URLs amigáveis e meta tags)

# Se for usar banco de dados:

# SQLAlchemy

# Flask-SQLAlchemy

**Exemplo de app.py (Flask Core):**

Python

from flask import Flask, render\_template, request, jsonify, url\_for

# from flask\_seo import FlaskSEO # Se for usar Flask-SEO

app = Flask(\_\_name\_\_)

# seo = FlaskSEO() # Se for usar Flask-SEO

# seo.init\_app(app) # Se for usar Flask-SEO

# Dados dos serviços para fácil acesso e renderização

SERVICES\_DATA = [

{

"id": "eletricista-predial-residencial",

"title": "Eletricista Predial e Residencial",

"description": "Instalação, manutenção e reparo de sistemas elétricos em edifícios e residências, garantindo segurança e conformidade com as normas técnicas. Inclui desde a instalação de tomadas e interruptores até a montagem de quadros de distribuição e correção de falhas.",

"image": "image7.webp.jpeg"

},

{

"id": "eletricista-industrial",

"title": "Eletricista Industrial",

"description": "Serviços especializados em instalações elétricas de grande porte para indústrias, incluindo montagem de painéis de comando, manutenção de máquinas e equipamentos, e implementação de sistemas de automação industrial, visando eficiência e segurança operacional.",

"image": None # Sem imagem específica fornecida

},

{

"id": "instalacao-de-alarmes",

"title": "Instalação de Alarmes",

"description": "Implementação de sistemas de alarme robustos e personalizados para proteger propriedades contra invasões e acessos não autorizados, com sensores de movimento, abertura e pânico.",

"image": "image2.webp.jpeg"

},

{

"id": "instalacao-cftv",

"title": "Instalação de Câmeras de Segurança (CFTV)",

"description": "Projeto e instalação de sistemas de Circuito Fechado de Televisão (CFTV), permitindo o monitoramento de ambientes internos e externos, com gravação e acesso remoto para maior segurança e controle.",

"image": "image5.webp.jpeg"

},

{

"id": "alpinismo-industrial",

"title": "Alpinismo Industrial em Fortaleza",

"description": "Serviços especializados de acesso por corda para trabalhos em altura em ambientes industriais, como inspeções, manutenção e reparos em estruturas elevadas e de difícil acesso.",

"video": "altura.mp4",

"image": "image4.webp.jpeg"

},

{

"id": "pintura-predial",

"title": "Pintura Predial em Fortaleza",

"description": "Pintura de fachadas e áreas externas de edifícios, utilizando técnicas de alpinismo industrial quando necessário, garantindo acabamento de alta qualidade e durabilidade.",

"video": "pintura.mp4",

"image": "image1.webp.jpeg"

},

{

"id": "seguranca-eletronica",

"title": "Segurança Eletrônica em Fortaleza",

"description": "Soluções integradas de segurança eletrônica para residências, condomínios e empresas, englobando instalação e manutenção de sistemas de alarme, câmeras de CFTV, controle de acesso e cercas elétricas.",

"images": ["image2.webp.jpeg", "image5.webp.jpeg", "image3.webp.jpeg"] # Múltiplas imagens

}

# Adicionar os outros serviços aqui

]

@app.route('/')

def index():

# SEO

# seo.add\_h1("DSE Segurança Eletrônica e Serviços LTDA - Fortaleza")

# seo.add\_meta\_keywords(["segurança eletrônica fortaleza", "eletricista fortaleza", "pintura predial fortaleza", "alpinismo industrial"])

# seo.add\_meta\_description("DSE: Soluções completas em segurança eletrônica, serviços elétricos e manutenção predial em Fortaleza. Qualidade e segurança garantidas.")

return render\_template('index.html', services=SERVICES\_DATA)

@app.route('/sobre')

def about():

# seo.add\_h1("Sobre a DSE - Segurança e Serviços em Fortaleza")

return render\_template('about.html')

@app.route('/servicos')

def services():

# seo.add\_h1("Nossos Serviços - DSE Segurança e Serviços")

return render\_template('services.html', services=SERVICES\_DATA)

@app.route('/servico/<service\_id>')

def service\_detail(service\_id):

service = next((s for s in SERVICES\_DATA if s["id"] == service\_id), None)

if service:

# seo.add\_h1(f"Serviço: {service['title']}")

# seo.add\_meta\_description(service['description'])

return render\_template('service\_detail.html', service=service)

return "Serviço não encontrado", 404

@app.route('/contato')

def contact():

# seo.add\_h1("Contato DSE - Segurança Eletrônica e Serviços")

return render\_template('contact.html')

@app.route('/search')

def search():

query = request.args.get('q', '').lower()

results = [s for s in SERVICES\_DATA if query in s['title'].lower() or query in s['description'].lower()]

return render\_template('search\_results.html', query=query, results=results)

# Rota para alternar tema (pode ser feita puramente com JS no frontend também)

@app.route('/toggle-theme', methods=['POST'])

def toggle\_theme():

# Este endpoint seria para persistência do tema no lado do servidor,

# mas para simplicidade, faremos com localStorage no JS.

return jsonify({"status": "ok"})

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**Exemplo de templates/base.html (Estrutura Base com Tema e Botão Voltar ao Topo):**

HTML

<!DOCTYPE **html**>

<html lang="pt-BR" data-theme="dark"> <head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>{% block title %}DSE Segurança Eletrônica e Serviços LTDA{% endblock %}</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

{% block head\_extra %}{% endblock %}

</head>

<body>

<header>

<nav class="navbar navbar-expand-lg {% if request.cookies.get('theme', 'dark') == 'dark' %}navbar-dark bg-dark{% else %}navbar-light bg-light{% endif %}">

<div class="container">

<a class="navbar-brand" href="{{ url\_for('index') }}">

<img src="{{ url\_for('static', filename='img/logo\_dse.png') }}" alt="DSE Logo" height="40">

</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav ms-auto">

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('index') }}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('about') }}">Sobre Nós</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('services') }}">Serviços</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('contact') }}">Contato</a>

</li>

<li class="nav-item">

<button id="theme-toggle" class="btn btn-outline-light ms-2">

<i class="fas fa-sun"></i> / <i class="fas fa-moon"></i>

</button>

</li>

<li class="nav-item d-flex align-items-center ms-3">

<form class="d-flex" action="{{ url\_for('search') }}" method="get">

<input class="form-control me-2" type="search" placeholder="Buscar serviços" aria-label="Search" name="q">

<button class="btn btn-outline-success" type="submit"><i class="fas fa-search"></i></button>

</form>

</li>

</ul>

</div>

</div>

</nav>

</header>

<main class="container my-4">

{% block content %}{% endblock %}

</main>

<footer class="footer mt-auto py-3 {% if request.cookies.get('theme', 'dark') == 'dark' %}bg-dark text-white{% else %}bg-light text-dark{% endif %}">

<div class="container text-center">

<p>&copy; {{ now.year }} DSE Segurança Eletrônica e Serviços LTDA. Todos os direitos reservados.</p>

<div class="social-links">

<a href="https://wa.me/SEUNUMERODOWHATSAPP" target="\_blank" class="text-decoration-none mx-2"><i class="fab fa-whatsapp fa-2x" style="color: #25D366;"></i></a>

<a href="https://www.instagram.com/SEUINSTAGRAM" target="\_blank" class="text-decoration-none mx-2"><i class="fab fa-instagram fa-2x" style="color: #E1306C;"></i></a>

</div>

</div>

</footer>

<button id="scrollToTopBtn" class="btn btn-warning rounded-circle shadow" title="Voltar ao Topo">

<i class="fas fa-arrow-up"></i>

</button>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

<script src="{{ url\_for('static', filename='js/script.js') }}"></script>

</body>

</html>

**Exemplo de static/css/style.css (CSS para Tema e Estilização):**

CSS

/\* Base styles \*/

body {

font-family: Arial, sans-serif;

transition: background-color 0.3s, color 0.3s;

}

/\* Dark Theme \*/

html[data-theme='dark'] body {

background-color: #1a1a1a; /\* Dark background \*/

color: #e0e0e0; /\* Light text \*/

}

html[data-theme='dark'] .navbar-dark {

background-color: #0d0d0d !important; /\* Even darker navbar \*/

}

html[data-theme='dark'] .navbar-nav .nav-link {

color: #e0e0e0 !important;

}

html[data-theme='dark'] .navbar-nav .nav-link:hover {

color: #ffd700 !important; /\* Golden hover \*/

}

html[data-theme='dark'] .footer.bg-dark {

background-color: #0d0d0d !important;

}

/\* Light Theme \*/

html[data-theme='light'] body {

background-color: #f8f9fa; /\* Light background \*/

color: #343a40; /\* Dark text \*/

}

html[data-theme='light'] .navbar-light {

background-color: #ffffff !important;

}

html[data-theme='light'] .navbar-nav .nav-link {

color: #343a40 !important;

}

html[data-theme='light'] .navbar-nav .nav-link:hover {

color: #0056b3 !important; /\* Blue hover \*/

}

html[data-theme='light'] .footer.bg-light {

background-color: #e9ecef !important;

}

/\* Golden Accent for elements that are always golden or specific to dark theme \*/

html[data-theme='dark'] h1, html[data-theme='dark'] h2, html[data-theme='dark'] h3 {

color: #ffd700; /\* Golden color for headings \*/

}

.btn-golden { /\* Custom button style \*/

background-color: #ffd700;

color: #1a1a1a;

border-color: #ffd700;

}

.btn-golden:hover {

background-color: #e6b800;

border-color: #e6b800;

color: #1a1a1a;

}

/\* Scroll to Top Button \*/

#scrollToTopBtn {

display: none; /\* Hidden by default \*/

position: fixed; /\* Fixed/sticky position \*/

bottom: 20px; /\* Place the button at the bottom of the page \*/

right: 30px; /\* Place the button 30px from the right \*/

z-index: 99; /\* Make sure it does not overlap \*/

border: none; /\* Remove borders \*/

outline: none; /\* Remove outline \*/

background-color: #ffd700; /\* Golden background \*/

color: #1a1a1a; /\* Dark text \*/

cursor: pointer; /\* Add a mouse pointer on hover \*/

padding: 15px; /\* Some padding \*/

border-radius: 50%; /\* Rounded corners \*/

font-size: 18px; /\* Increase font size \*/

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

transition: background-color 0.3s;

}

#scrollToTopBtn:hover {

background-color: #e6b800; /\* Darker golden on hover \*/

}

/\* Responsive adjustments \*/

@media (max-width: 768px) {

.navbar-collapse {

background-color: var(--bs-dark); /\* Ensure background for dropdown on mobile \*/

}

}

/\* Lazy loading for images \*/

img[data-src] {

opacity: 0;

transition: opacity 0.3s ease-in-out;

}

img.lazyloaded {

opacity: 1;

}

**Exemplo de static/js/script.js (JavaScript para Funcionalidades):**

JavaScript

document.addEventListener('DOMContentLoaded', function() {

const themeToggle = document.getElementById('theme-toggle');

const htmlElement = document.documentElement; // The <html> tag

const scrollToTopBtn = document.getElementById('scrollToTopBtn');

// Load saved theme from localStorage

const savedTheme = localStorage.getItem('theme') || 'dark'; // Default to dark

htmlElement.setAttribute('data-theme', savedTheme);

updateNavbarTheme(savedTheme); // Update navbar on load

// Function to update navbar theme

function updateNavbarTheme(theme) {

const navbar = document.querySelector('.navbar');

const footer = document.querySelector('footer');

if (navbar) {

if (theme === 'dark') {

navbar.classList.remove('navbar-light', 'bg-light');

navbar.classList.add('navbar-dark', 'bg-dark');

} else {

navbar.classList.remove('navbar-dark', 'bg-dark');

navbar.classList.add('navbar-light', 'bg-light');

}

}

if (footer) {

if (theme === 'dark') {

footer.classList.remove('bg-light', 'text-dark');

footer.classList.add('bg-dark', 'text-white');

} else {

footer.classList.remove('bg-dark', 'text-white');

footer.classList.add('bg-light', 'text-dark');

}

}

}

// Toggle theme on button click

if (themeToggle) {

themeToggle.addEventListener('click', function() {

let currentTheme = htmlElement.getAttribute('data-theme');

let newTheme = (currentTheme === 'dark') ? 'light' : 'dark';

htmlElement.setAttribute('data-theme', newTheme);

localStorage.setItem('theme', newTheme);

updateNavbarTheme(newTheme);

});

}

// Scroll to Top functionality

window.onscroll = function() { scrollFunction() };

function scrollFunction() {

if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {

scrollToTopBtn.style.display = "block";

} else {

scrollToTopBtn.style.display = "none";

}

}

if (scrollToTopBtn) {

scrollToTopBtn.addEventListener('click', function() {

document.body.scrollTop = 0; // For Safari

document.documentElement.scrollTop = 0; // For Chrome, Firefox, IE and Opera

});

}

// Lazy Loading for images and videos (basic example)

const lazyLoadElements = document.querySelectorAll('img[data-src], video[data-src]');

const lazyLoad = function(entries, observer) {

entries.forEach(entry => {

if (entry.isIntersecting) {

const element = entry.target;

if (element.tagName === 'IMG' && element.dataset.src) {

element.src = element.dataset.src;

element.classList.add('lazyloaded');

} else if (element.tagName === 'VIDEO' && element.dataset.src) {

element.src = element.dataset.src;

element.load(); // Load the video

element.classList.add('lazyloaded');

}

observer.unobserve(element);

}

});

};

const observer = new IntersectionObserver(lazyLoad, {

rootMargin: '0px 0px 50px 0px' // Load when 50px from viewport

});

lazyLoadElements.forEach(element => observer.observe(element));

});

**6. Considerações de SEO**

* **URLs Amigáveis:** As rotas Flask já foram definidas para serem amigáveis (ex: /servico/alpinismo-industrial).
* **Meta Tags:** Utilizar Jinja2 para injetar meta tags dinamicamente no <head> de cada página ({% block head\_extra %}). O Flask-SEO pode automatizar isso.
* **Tags Semânticas:** Usar <h1>, <h2>, <p>, <footer>, <nav>, <header>, <main>, <section> para estruturar o conteúdo.
* **Imagens:** Atributos alt descritivos para todas as imagens.
* **Conteúdo de Qualidade:** As descrições dos serviços são detalhadas, o que ajuda na relevância para buscas.
* **Palavras-chave:** Incluir palavras-chave relevantes naturalmente no texto (ex: "segurança eletrônica Fortaleza", "eletricista predial", "pintura em altura").

**7. Próximos Passos para Implementação**

1. **Configurar Ambiente:** Criar o ambiente virtual (python -m venv venv), ativar e instalar as dependências (pip install -r requirements.txt).
2. **Popular Conteúdo:** Preencher os templates HTML ( index.html, about.html, services.html, etc.) com o conteúdo real, usando os dados dos serviços definidos no app.py.
3. **Adicionar Logo e Imagens:** Salvar logo\_dse.png e todas as imagens/vídeos fornecidos na pasta static/img/ e static/video/ respectivamente. O image6.webp.jpeg (logo DSE) pode ser usado como logo\_dse.png.
4. **Ajustar CSS:** Refinar o static/css/style.css para um design mais polido e os detalhes dourados.
5. **Testar Responsividade:** Verificar o site em diferentes tamanhos de tela.
6. **Preencher Links:** Substituir SEUNUMERODOWHATSAPP e SEUINSTAGRAM nos links do rodapé.
7. **Testar Funcionalidades:** Verificar o toggle de tema, o botão "voltar ao topo" e a busca.

Este plano fornece uma base sólida para o desenvolvimento do site da DSE. Me diga se você gostaria de aprofundar em alguma parte específica ou se tem mais alguma funcionalidade em mente!

Ótimo! Vamos construir este site passo a passo, desde o ambiente de desenvolvimento até a publicação com um domínio gratuito.

**Guia Detalhado: Criando e Publicando o Site DSE com Flask**

**Parte 1: Configuração do Ambiente de Desenvolvimento**

Esta parte é crucial para organizar seu projeto e suas dependências.

**1. Verifique a Instalação do Python:**

* Abra o Terminal (Linux/macOS) ou Prompt de Comando/PowerShell (Windows).
* Digite: python --version ou python3 --version
* Você deve ver algo como Python 3.x.x. Se não tiver Python, baixe e instale-o do site oficial: [python.org](https://www.python.org/downloads/). Recomendo Python 3.9+.

**2. Crie a Pasta do Projeto:**

* Escolha um local no seu computador onde deseja guardar o projeto.
* Crie uma nova pasta para o projeto. Por exemplo:

Bash

mkdir dse\_website

cd dse\_website

**3. Crie e Ative um Ambiente Virtual (Venv):**

* Um ambiente virtual isola as bibliotecas do seu projeto do resto do seu sistema Python, evitando conflitos.
* Dentro da pasta dse\_website, execute:

Bash

python -m venv venv

(Isso cria uma pasta chamada venv dentro do seu projeto.)

* **Ative o ambiente virtual:**
  + **Windows:**

Bash

venv\Scripts\activate

* + **Linux/macOS:**

Bash

source venv/bin/activate

* + Você verá (venv) no início da linha de comando, indicando que o ambiente está ativado.

**4. Instale as Dependências do Projeto:**

* Com o ambiente virtual ativado, instale as bibliotecas necessárias.
* Crie um arquivo chamado requirements.txt na raiz da pasta dse\_website e cole o seguinte conteúdo:
* Flask==2.3.3
* # Para futuras funcionalidades (ex: Flask-Mail, Flask-SQLAlchemy)
* # Flask-SEO # Se quiser usar o Flask-SEO para SEO mais avançado

(Note: Fixei a versão do Flask para garantir compatibilidade. Você pode usar uma versão mais recente se quiser, apenas remova ==2.3.3)

* Agora, instale-as usando pip:

Bash

pip install -r requirements.txt

**Parte 2: Estrutura de Arquivos e Código Base**

Vamos criar os arquivos essenciais.

**1. Crie a Estrutura de Pastas:**

* Dentro da pasta dse\_website, crie as seguintes pastas:
* dse\_website/
* ├── app.py
* ├── static/
* │ ├── css/
* │ ├── js/
* │ ├── img/
* │ └── video/
* └── templates/

**2. Crie o app.py:**

* Na raiz da pasta dse\_website, crie um arquivo chamado app.py e cole o código Flask que forneci anteriormente.
* **Importante:** Atualize os dados de SERVICES\_DATA para incluir todos os serviços listados.

<!-- end list -->

Python

# dse\_website/app.py

from flask import Flask, render\_template, request, jsonify, url\_for

from datetime import datetime # Para o ano do copyright

app = Flask(\_\_name\_\_)

# Dados dos serviços (COMPLETO)

SERVICES\_DATA = [

{

"id": "eletricista-predial-residencial",

"title": "Eletricista Predial e Residencial",

"description": "Instalação, manutenção e reparo de sistemas elétricos em edifícios e residências, garantindo segurança e conformidade com as normas técnicas. Inclui desde a instalação de tomadas e interruptores até a montagem de quadros de distribuição e correção de falhas.",

"image": "image7.webp.jpeg"

},

{

"id": "eletricista-industrial",

"title": "Eletricista Industrial",

"description": "Serviços especializados em instalações elétricas de grande porte para indústrias, incluindo montagem de painéis de comando, manutenção de máquinas e equipamentos, e implementação de sistemas de automação industrial, visando eficiência e segurança operacional.",

"image": None

},

{

"id": "instalacao-de-antenas",

"title": "Instalação de Antenas",

"description": "Instalação e direcionamento de antenas para TV digital, parabólicas e sistemas de comunicação, garantindo a melhor qualidade de sinal para residências e empresas.",

"image": None

},

{

"id": "instalacao-de-alarmes",

"title": "Instalação de Alarmes",

"description": "Implementação de sistemas de alarme robustos e personalizados para proteger propriedades contra invasões e acessos não autorizados, com sensores de movimento, abertura e pânico.",

"image": "image2.webp.jpeg"

},

{

"id": "instalacao-cftv",

"title": "Instalação de Câmeras de Segurança (CFTV)",

"description": "Projeto e instalação de sistemas de Circuito Fechado de Televisão (CFTV), permitindo o monitoramento de ambientes internos e externos, com gravação e acesso remoto para maior segurança e controle.",

"image": "image5.webp.jpeg"

},

{

"id": "instalacao-de-alarmes-residenciais-prediais",

"title": "Instalação de Alarmes Residenciais e Prediais",

"description": "Foco na segurança de residências e edifícios, com sistemas de alarme customizados que incluem sensores de portas/janelas, detectores de presença e integração com centrais de monitoramento.",

"image": "image2.webp.jpeg" # Reutiliza imagem

},

{

"id": "instalacao-de-interfones",

"title": "Instalação de Interfones",

"description": "Instalação e manutenção de sistemas de interfonia para condomínios, empresas e residências, facilitando a comunicação e o controle de acesso de visitantes.",

"image": None

},

{

"id": "automacao-de-portoes",

"title": "Automação de Portões",

"description": "Automação de portões residenciais, comerciais e industriais, oferecendo conveniência, segurança e durabilidade, com opções de motores e controles remotos.",

"image": None

},

{

"id": "projeto-eletrico",

"title": "Projeto Elétrico",

"description": "Elaboração de projetos elétricos completos e detalhados para novas construções ou reformas, garantindo que a instalação seja segura, eficiente e em conformidade com as normas da ABNT.",

"image": None

},

{

"id": "instalacao-de-fio-terra",

"title": "Instalação de Fio Terra",

"description": "Implementação de sistemas de aterramento elétrico (fio terra) para proteção de pessoas e equipamentos contra choques elétricos e surtos de energia, essencial para a segurança de qualquer instalação.",

"image": "image7.webp.jpeg" # Reutiliza imagem

},

{

"id": "alpinismo-industrial-fortaleza",

"title": "Alpinismo Industrial em Fortaleza",

"description": "Serviços especializados de acesso por corda para trabalhos em altura em ambientes industriais, como inspeções, manutenção e reparos em estruturas elevadas e de difícil acesso.",

"video": "altura.mp4",

"image": "image4.webp.jpeg"

},

{

"id": "pintura-predial-fortaleza",

"title": "Pintura Predial em Fortaleza",

"description": "Pintura de fachadas e áreas externas de edifícios, utilizando técnicas de alpinismo industrial quando necessário, garantindo acabamento de alta qualidade e durabilidade.",

"video": "pintura.mp4",

"image": "image1.webp.jpeg"

},

{

"id": "manutencao-predial-fortaleza",

"title": "Manutenção Predial em Fortaleza",

"description": "Serviços de manutenção preventiva e corretiva para edifícios, incluindo reparos estruturais, hidráulicos, elétricos e de acabamento, visando a conservação e funcionalidade do imóvel.",

"image": "image1.webp.jpeg" # Pode reutilizar imagens de altura

},

{

"id": "seguranca-eletronica-fortaleza",

"title": "Segurança Eletrônica em Fortaleza",

"description": "Soluções integradas de segurança eletrônica para residências, condomínios e empresas, englobando instalação e manutenção de sistemas de alarme, câmeras de CFTV, controle de acesso e cercas elétricas.",

"images": ["image2.webp.jpeg", "image5.webp.jpeg", "image3.webp.jpeg"]

}

]

@app.route('/')

def index():

return render\_template('index.html', services=SERVICES\_DATA, now=datetime.now())

@app.route('/sobre')

def about():

return render\_template('about.html', now=datetime.now())

@app.route('/servicos')

def services():

return render\_template('services.html', services=SERVICES\_DATA, now=datetime.now())

@app.route('/servico/<service\_id>')

def service\_detail(service\_id):

service = next((s for s in SERVICES\_DATA if s["id"] == service\_id), None)

if service:

return render\_template('service\_detail.html', service=service, now=datetime.now())

return "Serviço não encontrado", 404

@app.route('/contato')

def contact():

return render\_template('contact.html', now=datetime.now())

@app.route('/search')

def search():

query = request.args.get('q', '').lower()

results = [s for s in SERVICES\_DATA if query in s['title'].lower() or (s['description'] and query in s['description'].lower())]

return render\_template('search\_results.html', query=query, results=results, now=datetime.now())

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**3. Crie os Arquivos HTML (Templates):**

* Dentro da pasta templates/, crie os seguintes arquivos e cole o código (adapte o conteúdo para cada página):
* **templates/base.html:** (O código que forneci anteriormente)
* **templates/index.html:**

HTML

{% extends 'base.html' %}

{% block title %}Home - DSE Segurança Eletrônica e Serviços LTDA{% endblock %}

{% block content %}

<section class="hero-section text-center py-5">

<h1 class="display-4 mb-4">DSE Segurança Eletrônica e Serviços LTDA</h1>

<p class="lead">Soluções completas e inovadoras em segurança eletrônica, serviços elétricos e manutenção predial em Fortaleza e região.</p>

<a href="{{ url\_for('contact') }}" class="btn btn-golden btn-lg mt-3">Fale Conosco</a>

</section>

<section class="services-overview py-5">

<h2 class="text-center mb-5">Nossos Principais Serviços</h2>

<div class="row row-cols-1 row-cols-md-3 g-4">

{% for service in services[:6] %} {# Mostra os 6 primeiros serviços na home #}

<div class="col">

<div class="card h-100 {% if request.cookies.get('theme', 'dark') == 'dark' %}bg-dark text-white border-golden{% else %}bg-light text-dark{% endif %}">

{% if service.image %}

<img data-src="{{ url\_for('static', filename='img/' + service.image) }}" class="card-img-top lazyload" alt="{{ service.title }}">

{% elif service.video %}

<video data-src="{{ url\_for('static', filename='video/' + service.video) }}" class="card-img-top lazyload" autoplay loop muted playsinline></video>

{% elif service.images %}

<img data-src="{{ url\_for('static', filename='img/' + service.images[0]) }}" class="card-img-top lazyload" alt="{{ service.title }}">

{% endif %}

<div class="card-body">

<h5 class="card-title">{{ service.title }}</h5>

<p class="card-text">{{ service.description[:100] }}...</p> {# Exemplo de descrição curta #}

<a href="{{ url\_for('service\_detail', service\_id=service.id) }}" class="btn btn-sm btn-outline-golden">Saiba Mais</a>

</div>

</div>

</div>

{% endfor %}

</div>

<div class="text-center mt-5">

<a href="{{ url\_for('services') }}" class="btn btn-golden btn-lg">Ver Todos os Serviços</a>

</div>

</section>

<section class="about-us-summary py-5">

<h2 class="text-center mb-4">Quem Somos</h2>

<p class="text-center w-75 mx-auto">A DSE Segurança Eletrônica e Serviços LTDA é uma empresa dedicada a oferecer soluções completas e de alta qualidade em segurança eletrônica, instalações elétricas e manutenção predial. Com anos de experiência no mercado de Fortaleza e região, nossa equipe de profissionais qualificados garante um serviço confiável e eficiente, focado na sua segurança e satisfação.</p>

<div class="text-center mt-4">

<a href="{{ url\_for('about') }}" class="btn btn-outline-golden">Conheça Nossa História</a>

</div>

</section>

{% endblock %}

* **templates/about.html:**

HTML

{% extends 'base.html' %}

{% block title %}Sobre Nós - DSE Segurança Eletrônica e Serviços LTDA{% endblock %}

{% block content %}

<h1 class="mb-4">Sobre a DSE</h1>

<p>A DSE Segurança Eletrônica e Serviços LTDA nasceu da paixão por inovação e do compromisso com a segurança e a excelência em serviços. Desde nossa fundação em Fortaleza, Ceará, temos nos dedicado a oferecer as melhores soluções em instalações elétricas, sistemas de segurança eletrônica e manutenção predial, atendendo às necessidades de residências, empresas e indústrias.</p>

<h2>Nossa Missão</h2>

<p>Ser a parceira de confiança de nossos clientes, proporcionando segurança, conforto e tranquilidade através de serviços de alta qualidade, tecnologia de ponta e um atendimento diferenciado.</p>

<h2>Nossos Valores</h2>

<ul>

<li><strong>Compromisso com a Qualidade:</strong> Buscamos a excelência em cada projeto, utilizando materiais de primeira linha e técnicas apuradas.</li>

<li><strong>Segurança:</strong> A segurança de nossos clientes e colaboradores é nossa prioridade máxima.</li>

<li><strong>Inovação:</strong> Estamos sempre atualizados com as últimas tendências e tecnologias do mercado.</li>

<li><strong>Transparência e Ética:</strong> Conduzimos nossos negócios com honestidade e clareza.</li>

<li><strong>Satisfação do Cliente:</strong> Nosso maior objetivo é superar as expectativas de quem confia em nossos serviços.</li>

</ul>

<p class="mt-4">Nossa equipe é composta por eletricistas, técnicos em segurança eletrônica e alpinistas industriais altamente qualificados e certificados, prontos para oferecer soluções personalizadas para cada desafio.</p>

<p>Confie na DSE para um futuro mais seguro e eficiente!</p>

{% endblock %}

* **templates/services.html:**

HTML

{% extends 'base.html' %}

{% block title %}Serviços - DSE Segurança Eletrônica e Serviços LTDA{% endblock %}

{% block content %}

<h1 class="mb-4">Nossos Serviços Completos</h1>

<p class="lead">A DSE oferece uma gama completa de serviços para garantir a segurança, funcionalidade e valorização do seu imóvel ou empresa. Explore nossas especialidades:</p>

<div class="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-4 mt-4">

{% for service in services %}

<div class="col">

<div class="card h-100 {% if request.cookies.get('theme', 'dark') == 'dark' %}bg-dark text-white border-golden{% else %}bg-light text-dark{% endif %}">

{% if service.image %}

<img data-src="{{ url\_for('static', filename='img/' + service.image) }}" class="card-img-top lazyload" alt="{{ service.title }}">

{% elif service.video %}

<video data-src="{{ url\_for('static', filename='video/' + service.video) }}" class="card-img-top lazyload" autoplay loop muted playsinline></video>

{% elif service.images %}

<img data-src="{{ url\_for('static', filename='img/' + service.images[0]) }}" class="card-img-top lazyload" alt="{{ service.title }}">

{% endif %}

<div class="card-body">

<h5 class="card-title">{{ service.title }}</h5>

<p class="card-text">{{ service.description[:150] }}...</p> {# Exemplo de descrição um pouco mais longa #}

<a href="{{ url\_for('service\_detail', service\_id=service.id) }}" class="btn btn-sm btn-outline-golden mt-2">Detalhes</a>

</div>

</div>

</div>

{% endfor %}

</div>

{% endblock %}

* **templates/service\_detail.html:**

HTML

{% extends 'base.html' %}

{% block title %}{{ service.title }} - DSE Serviços{% endblock %}

{% block content %}

<nav aria-label="breadcrumb">

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="{{ url\_for('index') }}">Home</a></li>

<li class="breadcrumb-item"><a href="{{ url\_for('services') }}">Serviços</a></li>

<li class="breadcrumb-item active" aria-current="page">{{ service.title }}</li>

</ol>

</nav>

<h1 class="mb-4">{{ service.title }}</h1>

<div class="row mb-4">

<div class="col-md-8">

{% if service.video %}

<div class="ratio ratio-16x9 mb-3">

<video data-src="{{ url\_for('static', filename='video/' + service.video)1 }}" controls autoplay loop2 muted playsinline class="lazyload"></video>

</div>

{% elif service.image %}

<img data-src="{{ url\_for('static', filename='img/' + service.image) }}" class="img-fluid rounded shadow-sm lazyload mb-3" alt="{{ service.title }}">

{% elif service.images %}

<div id="serviceCarousel" class="carousel slide" data-bs-ride="carousel">

<div class="carousel-inner">

{% for img in service.images %}

<div class="carousel-item {% if loop.first %}active{% endif %}">

<img data-src="{{ url\_for('static', filename='img/' + img) }}" class="d-block w-100 lazyload" alt="{{ service.title }} {{ loop.index }}">

</div>

{% endfor %}

</div>

{% if service.images|length > 1 %}

<button class="carousel-control-prev" type="button" data-bs-target="#serviceCarousel" data-bs-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="visually-hidden">Previous</span>

</button>

<button class="carousel-control-next" type="button" data-bs-target="#serviceCarousel" data-bs-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="visually-hidden">Next</span>

</button>

{% endif %}

</div>

{% endif %}

</div>

<div class="col-md-4">3

<div class="card {% if request.cookies.get('theme', 'dark') == 'dark' %}bg-dark text-white border-golden{% else %}bg-light text-dark{% endif %} p-3">

<h4 class="card-title">Detalhes do Serviço</h4>

<p class="card-text">{{ service.description }}</p>

<a href="{{ url\_for('contact') }}" class="btn btn-golden mt-3">Solicitar Orçamento</a>

</div>

</div>

</div>4

<p class="mt-4">Para mais informações sobre este serviço ou para agendar uma visita técnica, entre em contato conosco.</p>

{% endblock %}

```

* **templates/contact.html:**

HTML

{% extends 'base.html' %}

{% block title %}Contato - DSE Segurança Eletrônica e Serviços LTDA{% endblock %}

{% block content %}

<h1 class="mb-4">Fale Conosco</h1>

<p class="lead">Tem alguma dúvida, precisa de um orçamento ou deseja mais informações sobre nossos serviços? Entre em contato conosco! Nossa equipe está pronta para atendê-lo.</p>

<div class="row mt-5">

<div class="col-md-6">

<h2 class="mb-3">Informações de Contato</h2>

<p><i class="fas fa-phone-alt me-2"></i> \*\*Telefone:\*\* (XX) XXXX-XXXX</p>

<p><i class="fas fa-whatsapp me-2"></i> \*\*WhatsApp:\*\* <a href="https://wa.me/SEUNUMERODOWHATSAPP" target="\_blank">(XX) XXXXX-XXXX</a></p>

<p><i class="fas fa-envelope me-2"></i> \*\*E-mail:\*\* <a href="mailto:contato@dse.com.br">contato@dse.com.br</a></p>

<p><i class="fas fa-map-marker-alt me-2"></i> \*\*Endereço:\*\* Rua Exemplo, 123, Bairro Centro, Fortaleza - CE, CEP: XXXXX-XXX</p>

<div class="social-links mt-4">

<a href="https://wa.me/SEUNUMERODOWHATSAPP" target="\_blank" class="text-decoration-none me-3"><i class="fab fa-whatsapp fa-3x" style="color: #25D366;"></i></a>

<a href="https://www.instagram.com/SEUINSTAGRAM" target="\_blank" class="text-decoration-none"><i class="fab fa-instagram fa-3x" style="color: #E1306C;"></i></a>

</div>

</div>

<div class="col-md-6">

<h2 class="mb-3">Envie sua Mensagem</h2>

<form>

<div class="mb-3">

<label for="name" class="form-label">Nome Completo</label>

<input type="text" class="form-control" id="name" required>

</div>

<div class="mb-3">

<label for="email" class="form-label">E-mail</label>

<input type="email" class="form-control" id="email" required>

</div>

<div class="mb-3">

<label for="phone" class="form-label">Telefone (opcional)</label>

<input type="tel" class="form-control" id="phone">

</div>

<div class="mb-3">

<label for="message" class="form-label">Sua Mensagem</label>

<textarea class="form-control" id="message" rows="5" required></textarea>

</div>

<button type="submit" class="btn btn-golden">Enviar Mensagem</button>

</form>

</div>

</div>

<div class="mt-5 text-center">

<h2 class="mb-3">Nossa Localização</h2>

<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3981.3653198083893!2d-38.5280962!3d-3.7318721!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x7c749912345!2sFortaleza%2C%20CE!5e0!3m2!1spt-BR!2sbr!4v1718848900000!5m2!1spt-BR!2sbr" width="100%" height="450" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade"></iframe>

</div>

{% endblock %}

* + **Nota:** O formulário de contato é apenas HTML. Para que ele funcione e envie e-mails, você precisaria de uma lógica de backend (usando Flask-Mail, por exemplo) e um serviço de e-mail. Para este projeto inicial, ele será visualmente presente, mas não funcional sem essa integração.
* **templates/search\_results.html:**

HTML

{% extends 'base.html' %}

{% block title %}Resultados da Busca por "{{ query }}" - DSE{% endblock %}

{% block content %}

<h1 class="mb-4">Resultados da Busca por "{{ query }}"</h1>

{% if results %}

<div class="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-4">

{% for service in results %}

<div class="col">

<div class="card h-100 {% if request.cookies.get('theme', 'dark') == 'dark' %}bg-dark text-white border-golden{% else %}bg-light text-dark{% endif %}">

{% if service.image %}

<img data-src="{{ url\_for('static', filename='img/' + service.image) }}" class="card-img-top lazyload" alt="{{ service.title }}">

{% elif service.video %}

<video data-src="{{ url\_for('static', filename='video/' + service.video) }}" class="card-img-top lazyload" autoplay loop muted playsinline></video>

{% elif service.images %}

<img data-src="{{ url\_for('static', filename='img/' + service.images[0]) }}" class="card-img-top lazyload" alt="{{ service.title }}">

{% endif %}

<div class="card-body">

<h5 class="card-title">{{ service.title }}</h5>

<p class="card-text">{{ service.description[:150] }}...</p>

<a href="{{ url\_for('service\_detail', service\_id=service.id) }}" class="btn btn-sm btn-outline-golden mt-2">Detalhes</a>

</div>

</div>

</div>

{% endfor %}

</div>

{% else %}

<p>Nenhum serviço encontrado para sua busca.</p>

<p>Tente termos diferentes ou <a href="{{ url\_for('services') }}">veja todos os nossos serviços</a>.</p>

{% endif %}

{% endblock %}

**4. Crie os Arquivos CSS e JavaScript:**

* **static/css/style.css:** (O código que forneci, com a adição da classe border-golden)

CSS

/\* Base styles \*/

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif; /\* Fonte mais profissional \*/

transition: background-color 0.3s, color 0.3s;

min-height: 100vh; /\* Para o footer ir para o final da página \*/

display: flex;

flex-direction: column;

}

main {

flex-grow: 1; /\* Faz o conteúdo principal crescer e empurrar o footer \*/

}

/\* Dark Theme \*/

html[data-theme='dark'] body {

background-color: #121212; /\* Darker background \*/

color: #e0e0e0;

}

html[data-theme='dark'] .navbar-dark {

background-color: #0a0a0a !important; /\* Even darker navbar \*/

border-bottom: 1px solid #ffd700; /\* Linha dourada \*/

}

html[data-theme='dark'] .navbar-nav .nav-link {

color: #e0e0e0 !important;

}

html[data-theme='dark'] .navbar-nav .nav-link:hover {

color: #ffd700 !important;

}

html[data-theme='dark'] .navbar-nav .nav-link.active {

color: #ffd700 !important;

font-weight: bold;

}

html[data-theme='dark'] .footer {

background-color: #0a0a0a !important;

color: #e0e0e0;

border-top: 1px solid #ffd700; /\* Linha dourada \*/

}

html[data-theme='dark'] .card {

background-color: #1a1a1a;

border-color: #333;

}

html[data-theme='dark'] .card.border-golden {

border-color: #ffd700 !important;

}

html[data-theme='dark'] .form-control {

background-color: #333;

color: #e0e0e0;

border-color: #555;

}

html[data-theme='dark'] .form-control::placeholder {

color: #bbb;

}

html[data-theme='dark'] .breadcrumb-item a {

color: #ffd700;

}

html[data-theme='dark'] .breadcrumb-item.active {

color: #e0e0e0;

}

/\* Light Theme \*/

html[data-theme='light'] body {

background-color: #ffffff;

color: #343a40;

}

html[data-theme='light'] .navbar-light {

background-color: #f8f9fa !important;

border-bottom: 1px solid #dee2e6;

}

html[data-theme='light'] .navbar-nav .nav-link {

color: #343a40 !important;

}

html[data-theme='light'] .navbar-nav .nav-link:hover {

color: #0056b3 !important;

}

html[data-theme='light'] .navbar-nav .nav-link.active {

color: #0056b3 !important;

font-weight: bold;

}

html[data-theme='light'] .footer {

background-color: #e9ecef !important;

color: #343a40;

border-top: 1px solid #dee2e6;

}

html[data-theme='light'] .card {

background-color: #ffffff;

border-color: #dee2e6;

}

html[data-theme='light'] .form-control {

background-color: #ffffff;

color: #343a40;

border-color: #ced4da;

}

/\* Golden Accent \*/

h1, h2, h3, h4, h5, h6 {

color: inherit; /\* Inherit from body for light theme, overridden by dark theme specific rule \*/

}

html[data-theme='dark'] h1, html[data-theme='dark'] h2, html[data-theme='dark'] h3 {

color: #ffd700; /\* Golden color for headings in dark mode \*/

}

.btn-golden {

background-color: #ffd700;

color: #1a1a1a;

border-color: #ffd700;

font-weight: bold;

transition: background-color 0.3s, border-color 0.3s, color 0.3s;

}

.btn-golden:hover {

background-color: #e6b800;

border-color: #e6b800;

color: #1a1a1a;

}

.btn-outline-golden {

color: #ffd700;

border-color: #ffd700;

transition: all 0.3s;

}

.btn-outline-golden:hover {

background-color: #ffd700;

color: #1a1a1a;

border-color: #ffd700;

}

.border-golden {

border: 1px solid #ffd700 !important;

}

/\* Scroll to Top Button \*/

#scrollToTopBtn {

display: none;

position: fixed;

bottom: 20px;

right: 30px;

z-index: 99;

border: none;

outline: none;

background-color: #ffd700;

color: #1a1a1a;

cursor: pointer;

padding: 15px;

border-radius: 50%;

font-size: 18px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

transition: background-color 0.3s, transform 0.3s;

}

#scrollToTopBtn:hover {

background-color: #e6b800;

transform: translateY(-3px);

}

/\* Lazy loading for images and videos \*/

img[data-src], video[data-src] {

opacity: 0;

transition: opacity 0.5s ease-in-out;

min-height: 150px; /\* Previne layout shift \*/

background-color: #eee; /\* Placeholder background \*/

}

img.lazyloaded, video.lazyloaded {

opacity: 1;

}

.hero-section {

background: linear-gradient(rgba(0, 0, 0, 0.6), rgba(0, 0, 0, 0.6)), url('/static/img/hero\_bg.jpg') no-repeat center center; /\* Crie uma imagem hero\_bg.jpg \*/

background-size: cover;

color: white;

padding: 100px 0;

animation: fadeIn 1s ease-in-out;

}

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

/\* Social links styling \*/

.social-links a {

transition: transform 0.2s ease-in-out;

}

.social-links a:hover {

transform: scale(1.1);

}

* + **Nota:** Crie uma imagem hero\_bg.jpg para o background da seção hero da página inicial e coloque-a em static/img/. Pode ser uma imagem abstrata, de tecnologia ou de Fortaleza.
* **static/js/script.js:** (O código que forneci)

**5. Adicione Imagens e Vídeos:**

* Pegue todas as imagens e vídeos que você tem (image1.webp.jpeg a image7.webp.jpeg, pintura.mp4, altura.mp4, e o logo image6.webp.jpeg).
* **Renomeie image6.webp.jpeg para logo\_dse.png** (ou use outro formato de imagem) e coloque-o em static/img/.
* Coloque as imagens restantes na pasta static/img/.
* Coloque os vídeos na pasta static/video/.
* **Importante:** Mantenha os nomes dos arquivos exatamente como estão referenciados no app.py e nos templates HTML.

**Parte 3: Testando o Site Localmente**

Antes de colocar no ar, teste para ter certeza de que tudo funciona.

**1. Execute o Servidor Flask:**

* Certifique-se de que seu ambiente virtual ainda está ativado.
* Na raiz da pasta dse\_website (onde está app.py), execute:

Bash

flask run

* Você verá uma mensagem como: Running on http://127.0.0.1:5000
* Abra seu navegador e acesse esse endereço (http://127.0.0.1:5000).

**2. Verifique as Funcionalidades:**

* Navegue por todas as páginas (Home, Sobre Nós, Serviços, Contato).
* Clique nos links dos serviços para ver as páginas de detalhes.
* Teste o botão de alternância de tema (escuro/claro).
* Teste o botão "Voltar ao Topo".
* Teste o campo de busca.
* Verifique se as imagens e vídeos estão carregando corretamente (especialmente com lazy loading).
* **Ajuste:** Faça quaisquer ajustes necessários no CSS, HTML ou no app.py até que o site esteja como você deseja.

**Parte 4: Colocando o Site no Ar (Hospedagem Gratuita)**

Hospedar um site Flask com domínio gratuito é um pouco mais complexo do que um site estático simples, pois você precisa de um servidor que suporte Python/Flask.

**Opções de Hospedagem Gratuita para Flask:**

Domínios gratuitos geralmente vêm como subdomínios (ex: seusite.netlify.app, seusite.pythonanywhere.com).

**Opção Recomendada para Iniciantes (e Flask): Heroku (com algumas ressalvas) ou PythonAnywhere.**

**Método 1: Heroku (Pode exigir cartão de crédito para contas novas, mas o plano gratuito ainda existe para aplicações pequenas)**

O Heroku é uma plataforma de nuvem que suporta vários tipos de aplicativos, incluindo Python Flask.

**1. Prepare seu Projeto para o Heroku:**

* **Procfile:** Crie um arquivo chamado Procfile (sem extensão) na raiz do seu projeto dse\_website com o seguinte conteúdo:
* web: gunicorn app:app

(Instalaremos gunicorn em breve. Ele é um servidor WSGI necessário para o Heroku).

* **requirements.txt:** Certifique-se de que seu requirements.txt está atualizado.
* Flask==2.3.3
* gunicorn==20.1.0 # Adicione esta linha
* # ...outras dependências

(A versão do gunicorn pode ser mais recente, verifique a compatibilidade).

* **runtime.txt:** Crie um arquivo chamado runtime.txt na raiz do seu projeto para especificar a versão do Python.
* python-3.9.18 # Use a versão do Python que você está usando localmente. Verifique com `python --version`.

**2. Crie uma Conta no Heroku:**

* Vá para [heroku.com](https://www.heroku.com/) e crie uma conta.

**3. Instale o Heroku CLI:**

* Siga as instruções em [devcenter.heroku.com/articles/heroku-cli](https://devcenter.heroku.com/articles/heroku-cli) para instalar o Heroku CLI no seu computador.

**4. Faça Login no Heroku via CLI:**

* No seu Terminal/Prompt de Comando (com o ambiente virtual ATIVADO e dentro da pasta dse\_website):

Bash

heroku login

Isso abrirá uma página no seu navegador para você fazer login.

**5. Crie um Aplicativo Heroku:**

* No Terminal:

Bash

heroku create seu-nome-para-o-site-dse # Escolha um nome único. Heroku vai gerar um se você não especificar.

O Heroku lhe dará um URL como https://seu-nome-para-o-site-dse.herokuapp.com/. Guarde este URL.

**6. Configure o Git:**

* O Heroku usa Git para implantar. Se você não tem Git instalado, baixe-o em [git-scm.com](https://git-scm.com/downloads).
* Inicialize um repositório Git no seu projeto:

Bash

git init

* Adicione o Heroku como um "remote":

Bash

heroku git:remote -a seu-nome-para-o-site-dse

* Crie um arquivo .gitignore na raiz do seu projeto para não enviar arquivos desnecessários:
* # .gitignore
* venv/
* \_\_pycache\_\_/
* \*.pyc
* .env
* .DS\_Store

**7. Faça o Deploy para o Heroku:**

* Adicione seus arquivos ao Git:

Bash

git add .

* Faça um commit:

Bash

git commit -m "Primeiro deploy do site DSE"

* Envie para o Heroku:

Bash

git push heroku main # Ou git push heroku master, dependendo da sua branch padrão

* O Heroku detectará seu aplicativo Flask, instalará as dependências e o implantará. Isso pode levar alguns minutos.

**8. Abra seu Site:**

* Uma vez que o deploy termine, você pode abrir seu site no navegador:

Bash

heroku open

Ou vá diretamente para o URL que o Heroku lhe deu.

**Método 2: PythonAnywhere (Mais amigável para projetos Python pequenos)**

O PythonAnywhere é uma plataforma que hospeda aplicativos Python diretamente. Possui um plano gratuito generoso para pequenos projetos.

**1. Prepare seu Projeto:**

* Você não precisa do Procfile ou gunicorn especificamente para o PythonAnywhere, mas o requirements.txt é útil.
* Certifique-se de que seu app.py está pronto.

**2. Crie uma Conta no PythonAnywhere:**

* Vá para [pythonanywhere.com](https://www.pythonanywhere.com/) e registre-se para uma conta "Beginner" (gratuita).

**3. Crie um Novo Web App:**

* Após fazer login, vá para a aba **Web** no seu painel.
* Clique em **"Add a new web app"**.
* Escolha "Flask" como framework.
* Selecione a versão do Python (use a mesma que você está usando localmente, por exemplo, Python 3.9).
* O PythonAnywhere criará um arquivo flask\_app.py para você. Você pode usar ele como base ou apagar o conteúdo e colar o seu app.py nele.

**4. Faça Upload dos Seus Arquivos:**

* No PythonAnywhere, vá para a aba **Files**.
* Você pode criar a estrutura de pastas (static/css, static/js, etc.) e fazer o upload dos seus arquivos individualmente.
* **Alternativa mais fácil:** Se seu projeto for pequeno, você pode compactar (.zip) a pasta dse\_website (excluindo a pasta venv) no seu computador. No PythonAnywhere, vá para a aba **Files**, navegue até o diretório do seu web app (geralmente /home/seu-usuario/seu-dominio.pythonanywhere.com/) e faça o upload do .zip. Depois, descompacte-o lá (você pode precisar de comandos no console para descompactar se não houver opção de interface).

**5. Configure o Web App:**

* Na aba **Web** do PythonAnywhere, role para baixo até a seção "Code".
* **Source code:** Certifique-se de que este caminho aponta para a raiz do seu projeto Flask (ex: /home/seu-usuario/dse\_website).
* **WSGI configuration file:** Clique no link para editar este arquivo (geralmente /var/www/seu\_usuario\_pythonanywhere\_com\_wsgi.py).
  + Apague o conteúdo existente.
  + Cole o seguinte (ajuste SEU\_USUARIO e dse\_website):

Python

import sys

import os

# Adicione o caminho do seu projeto Flask ao PYTHONPATH

project\_folder = '/home/SEU\_USUARIO/dse\_website'

if project\_folder not in sys.path:

sys.path.insert(0, project\_folder)

# Importe o seu aplicativo Flask

from app import app as application # 'application' é o nome que o WSGI espera

* **Virtualenv:** Defina o caminho para o seu ambiente virtual criado no PythonAnywhere (ex: /home/SEU\_USUARIO/dse\_website/venv). Você precisará criá-lo e instalar as dependências lá.

**6. Instale as Dependências no PythonAnywhere:**

* Vá para a aba **Consoles** e abra um "Bash console".
* Navegue até a pasta do seu projeto: cd dse\_website
* Crie o ambiente virtual (se ainda não o fez): python3.9 -m venv venv (use a versão do Python que você escolheu para o web app)
* Ative o ambiente: source venv/bin/activate
* Instale as dependências: pip install -r requirements.txt (certifique-se de que requirements.txt foi carregado)

**7. Recarregue o Web App:**

* Após fazer todas as configurações e uploads, volte para a aba **Web** e clique no botão **"Reload your web app"**.

**8. Acesse seu Site:**

* Seu site estará online no subdomínio fornecido pelo PythonAnywhere (ex: seu-usuario.pythonanywhere.com).

**Domínio Gratuito (Subdomínios):**

* **Heroku:** O domínio será seu-nome-para-o-site-dse.herokuapp.com.
* **PythonAnywhere:** O domínio será seu-usuario.pythonanywhere.com.

Esses são domínios **subdomínios** gratuitos, o que significa que o nome da plataforma (.herokuapp.com ou .pythonanywhere.com) estará sempre presente. Para um domínio de nível superior (.com, .net, .br), você precisaria comprar um.

**Considerações Finais:**

* **SEO:** Após a publicação, submeta seu site ao Google Search Console e Bing Webmaster Tools para que os mecanismos de busca o indexem.
* **Manutenção:** Lembre-se que em planos gratuitos, o servidor pode "dormir" após um período de inatividade e levar alguns segundos para "acordar" quando acessado novamente. Para sites profissionais, uma hospedagem paga é sempre recomendada para maior performance, confiabilidade e controle sobre o domínio.
* **Segurança:** Mantenha suas dependências atualizadas e siga as boas práticas de segurança Flask.

Este guia é um passo a passo detalhado. Se você encontrar dificuldades em qualquer etapa, sinta-se à vontade para perguntar!