

DOC do CHECK LIST

Falta reformular da maneira que for solicitada pela área.

Parte 1: HTML (`index.html`)

Propósito:

O HTML define a estrutura e os elementos visíveis da aplicação. Ele cria um formulário onde os usuários podem inserir informações sobre o estoque.

Componentes Principais:

- Estrutura Inicial

```
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Checklist de Estoque</title>
  <link rel="stylesheet" href="style.css">
</head>
```

- **DOCTYPE e `html`**: Define o documento como HTML5 e especifica o idioma como português.
- **`<meta>` tags**: Define a codificação de caracteres como UTF-8 e ajusta a visualização para dispositivos móveis.
- **`<title>`**: Define o título da página exibido na aba do navegador.
- **`<link>`**: Importa o arquivo CSS para estilização.

- **Corpo da Página**

```
<body>
  <h1>Checklist de Estoque</h1>
  <div class="container">
    <form id="checklistForm">

      </form>
    </div>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.3.1/jspdf.umd.min.js"></script>
    <script src="script.js"></script>
  </body>
</html>
```

- **<body>**: Contém todo o conteúdo visível da página.
- **<h1>**: Cabeçalho principal que descreve o título da aplicação.
- **<div class="container">**: Um contêiner que centraliza e agrupa o formulário.
- **<form>**: O formulário onde os usuários inserem os dados do estoque. Ele é identificado por `id="checklistForm"`.
- **<script>:Scripts:** Inclui a biblioteca `jsPDF` e o script personalizado `script.js`.

- **Campos do Formulário**

```
<div class="form-group">
  <label for="storeCode">Código da Loja</label>
  <input type="text" id="storeCode" name="storeCode" required>
</div>
<button type="button" onclick="generatePDF()">Baixar PDF
```

```
</button>
```

- `<div class="form-group">`: Agrupa cada par rótulo/campo.
- `<label>`: Rótulo descritivo para cada campo do formulário.
- `<input>`: Campos onde os usuários inserem dados (texto, data, número).
- `<select>`: Caixa de seleção para escolher o status do item.
- `<button>`: Botão que, ao ser clicado, chama a função `generatePDF()` para gerar o PDF.

Parte 2: CSS (`style.css`)

Propósito:

O CSS define o estilo visual da aplicação, como fontes, margens, cores e layout. Ele melhora a aparência e a usabilidade do formulário.

Componentes Principais:

- **Estilo Geral do Corpo**

```
body {  
    font-family: Arial, sans-serif;  
    margin: 20px;  
}
```

- `body`: Aplica a fonte Arial e define uma margem ao redor da página para espaçamento.

- **Cabeçalho Principal**

```
h1 {  
    text-align: center;
```

```
}
```

- `h1`: Centraliza o texto do título principal.

- **Contêiner do Formulário**

```
.container {  
  max-width: 600px;  
  margin: 0 auto;  
  padding: 20px;  
  border: 1px solid #ccc;  
  border-radius: 10px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}
```

- `.container`: Define o layout e estilo do contêiner do formulário, incluindo largura máxima, centralização, bordas e sombras.

- **Estilo dos Campos do Formulário**

```
.form-group {  
  margin-bottom: 15px;  
}  
.form-group label {  
  display: block;  
  margin-bottom: 5px;  
}  
.form-group input, .form-group select {  
  width: 100%;  
  padding: 8px;  
  box-sizing: border-box;  
}
```

- `.form-group`: Adiciona espaçamento inferior entre os grupos de entrada.
- `.form-group label`: Exibe rótulos como blocos e adiciona espaçamento.
- `.form-group input, .form-group select`: Define largura total, padding e box-sizing para os campos.

• Estilo do Botão

```
button {
  display: block;
  width: 100%;
  padding: 10px;
  background-color: #4CAF50;
  color: white;
  border: none;
  cursor: pointer;
}
button:hover {
  background-color: #45a049;
}
```

- `button`: Define o estilo do botão, incluindo cor de fundo, largura total e efeitos ao passar o mouse.

• Responsividade

```
@media (max-width: 600px) {
  .container {
    padding: 10px;
  }
  button {
    padding: 8px;
  }
}
```

```
}
```

- `@media (max-width: 600px)`: Ajusta o layout e padding para telas menores, garantindo a adaptabilidade da interface.

Parte 3: JavaScript (`script.js`)

Propósito:

O JavaScript adiciona funcionalidade à aplicação, manipulando dados do formulário e gerando o PDF usando a biblioteca

`jsPDF`.

Componentes Principais:

- Função `generatePDF`

```
function generatePDF() {  
  const { jsPDF } = window.jspdf;  
  const doc = new jsPDF();
```

- `const { jsPDF } = window.jspdf`: Importa a biblioteca `jsPDF` do objeto global `window`.
- `const doc = new jsPDF()`: Cria uma nova instância do objeto `jsPDF` para gerar o PDF.

- Adição de Cabeçalho

```
  doc.setFontSize(18);  
  doc.text("Checklist de Estoque", 105, 20, null, null,  
  1, "center");  
  doc.setLineWidth(0.5);  
  doc.line(20, 25, 190, 25);
```

- `doc.setFontSize(18)`: Define o tamanho da fonte para 18.

- `doc.text("Checklist de Estoque", 105, 20, null, null, "center")` : Adiciona o título centralizado no PDF.
- `doc.setLineWidth(0.5)` : Define a largura da linha.
- `doc.line(20, 25, 190, 25)` : Desenha uma linha horizontal abaixo do título.

• Adição de Rótulos e Valores

```
doc.setFontSize(12);
doc.setFont("helvetica", "bold");
doc.text("Código da Loja:", 20, 40);
doc.text("Nome da Loja:", 20, 50);
doc.text("Data:", 20, 60);
doc.text("Nome do Item:", 20, 70);
doc.text("Tipo do Produto:", 20, 80);
doc.text("Quantidade:", 20, 90);
doc.text("Localização:", 20, 100);
doc.text("Status:", 20, 110);
```

- `doc.setFontSize(12)` : Define o tamanho da fonte para 12.
- `doc.setFont("helvetica", "bold")` : Define a fonte como Helvetica em negrito para os rótulos.
- `doc.text("...", 20, xx)` : Adiciona rótulos no PDF em posições específicas.

```
doc.setFont("helvetica", "normal");
doc.text(document.getElementById('storeCode').value,
60, 40);
doc.text(document.getElementById('storeName').value,
60, 50);
doc.text(document.getElementById('date').value, 60,
60);
doc.text(document.getElementById('itemName').value,
60, 70);
```

```

    doc.text(document.getElementById('productType').value, 60, 80);
    doc.text(document.getElementById('quantity').value, 60, 90);
    doc.text(document.getElementById('location').value, 60, 100);
    doc.text(document.getElementById('status').value, 60, 110);

```

- `doc.setFont("helvetica", "normal")`: Define a fonte como Helvetica normal para os valores.
- `doc.text(..., 60, xx)`: Adiciona os valores inseridos no formulário no PDF nas posições correspondentes.

• Adição de Rodapé

```

const pageCount = doc.internal.getNumberOfPages();
for (let i = 1; i <= pageCount; i++) {
    doc.setPage(i);
    doc.setFontSize(10);
    doc.text('Página ' + String(i) + ' de ' + String(
pageCount), 105, 290, null, null, 'center');
}

```

- Essa parte que estiver em verde, retirei do projeto.
- `const pageCount = doc.internal.getNumberOfPages()`: Obtém o número total de páginas.
- `for (let i = 1; i <= pageCount; i++)`: Itera sobre cada página.
- `doc.setPage(i)`: Define a página atual.
- `doc.setFontSize(10)`: Define o tamanho da fonte para o rodapé.
- `doc.text('Página ' + String(i) + ' de ' + String(pageCount), 105, 290, null, null, 'center')`: Adiciona numeração ao rodapé, centralizada.

- **Salvar o PDF**

```
doc.save('checklist-estoque.pdf');  
}
```

- `doc.save('checklist-estoque.pdf')`: Salva o PDF com o nome 'checklist-estoque.pdf'.