UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
PROGRAMAÇÃO VISUAL E AUTORIA WEB
PROFESSOR: TANIRO RODRIGUES

Trabalho 3: Front end para uma loja

O projeto consiste no desenvolvimento de um website dinâmico utilizando apenas JavaScript puro para manipulação do DOM através do objeto document. O conteúdo da página será gerado dinamicamente com base em dados obtidos de uma API local de cafés. A estrutura HTML base deverá conter apenas uma div com id="root", sendo todo o restante do conteúdo inserido via JavaScript.

1. Estrutura do Projeto

O website terá três páginas principais:

1. **Página Inicial (Index)**
    - Lista de cafés obtidos via uma requisição GET para a API local.
    - Cada item da lista deve exibir:
        - Título.
        - Descrição.
        - Preço.
        - Imagem.
        - Ingredientes.
        - Link ou botão para adicionar o café ao carrinho.
    - Um **ícone de carrinho de compras** no cabeçalho que:
        - Mostra o número atual de itens no carrinho.
        - É clicável, levando para a página de carrinho de compras.
        - Adiciona um item ao web storage.
2. **Página de Carrinho de Compras**
    - Lista todos os itens adicionados ao carrinho (web storage).
    - Cada item da lista deve exibir:
        - Nome do café.
        - Preço.
        - Quantidade.
    - Funcionalidades:
        - Botões para **adicionar** e **remover** unidades de cada item.
        - Exibição do total acumulado do carrinho.
        - Um botão **"Finalizar Compra"** que redireciona para a página de finalização.
3. **Página de Finalização de Compra**
    - Exibe os itens que estavam no carrinho no momento de sua finalização.
    - Inclui um formulário com os seguintes campos:
        - Endereço de entrega (campo de texto).
        - Seleção de método de pagamento (cartão de crédito, boleto, PIX, etc.).
    - Um botão **"Finalizar"** que:
        - Valida se os campos estão preenchidos de maneira adequada.

- Exibe uma mensagem de confirmação da compra.
- Limpa os dados do carrinho no **Web Storage**.

2. Funcionamento do Carrinho de Compras

O carrinho será implementado utilizando a **API Web Storage**:

- Os itens do carrinho devem ser armazenados no localStorage para persistência entre sessões.
- Operações de adicionar, remover e listar itens devem manipular diretamente os dados no localStorage.
- Ao finalizar a compra, os dados do carrinho devem ser removidos do localStorage.

3. API Local de Cafés

A API será criada utilizando a biblioteca json-server do NPM. Os dados serão armazenados em um arquivo JSON contendo os seguintes campos para cada café:

- id: Identificador único.
- name: Nome do café.
- description: Descrição do café.
- price: Preço.

Exemplo de arquivo JSON (db.json):


```
{
  "coffee": [
    {
      "title": "Black Coffee",
      "description": "Café preto puro, feito apenas com grãos moídos e água quente. É simples e
ideal para apreciar o sabor natural do café.",
      "ingredients": ["Café"],
      "price": 3.0,
      "image": "https://images.unsplash.com/photo-1494314671902-
399b18174975?auto=format&fit=crop&q=80&w=1887&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",
      "id": 1
    },
    {
      "title": "Latte",
      "description": "Uma mistura cremosa de espresso e leite vaporizado, com uma fina camada de
espuma por cima. Pode ser servido puro ou com sabores adicionais, como baunilha ou caramelo.",
      "ingredients": ["Espresso", "Leite vaporizado"],
      "price": 4.5,
      "image": "https://images.unsplash.com/photo-1561882468-
9110e03e0f78?auto=format&fit=crop&q=60&w=800&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8MTl8fGxhdHRlfGVufDB8fDB8fHww",
      "id": 2
    },
```


4. Tutorial de Configuração da API Local

1. Instale o json-server como dependência do projeto:

```
npm install  json-server
```

2. Crie o arquivo db.json na raiz do projeto com os dados do exemplo acima.
3. Execute o servidor JSON:

```
npx json-server db.json
```

4. A API estará disponível no endereço:
    - Obter todos os cafés: http://localhost:3000/cafes
    - Obter um café específico: http://localhost:3000/cafes/{id}

5. Requisitos Técnicos

- Apenas **JavaScript puro** será utilizado para toda manipulação do DOM.
- O website deve ser responsivo e funcional em dispositivos móveis.
- CSS através da biblioteca Boostrap.
- O código deve ser modularizado e organizado, separando as funções de manipulação do DOM, comunicação com a API e gerenciamento do carrinho.