1) pages/index.php

```
<?php include_once '../templates/header.php'; ?>

<h2>Produtos Disponíveis</h2>

<div class="container">
  <?php
  include_once '../includes/db_connection.php';

  try {
    // Seleciona todos os produtos da tabela 'produtos'
    $sql = "SELECT * FROM produtos";
    $stmt = $conn->prepare($sql);
    $stmt->execute();

    // Exibe os produtos em um loop
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
      echo "<div class='produto'>";
      echo "<img src='../uploads/{$row['imagem']}' alt='{$row['produto']}'>";
      echo "<h3>{$row['produto']}</h3>";
      echo "<a href='../pages/product_detail.php?id={$row['id']}'>Detalhes</a>";
      echo "</div>";
    }
  } catch(PDOException $e) {
    echo "Erro na conexão com o banco de dados: " . $e->getMessage();
  }
  ?>
</div>

<?php include_once '../templates/footer.php'; ?>
```

Vou explicar linha por linha o que o código PHP faz:

1. ```php
<?php include_once '../templates/header.php'; ?>
```

- Esta linha inclui o conteúdo do arquivo `'../templates/header.php'` no documento atual. Provavelmente, este arquivo contém o cabeçalho HTML comum a todas as páginas do site, como a tag `<head>`, barra de navegação, etc.

**2. ```html**
**<h2>Produtos Disponíveis</h2>**
```

- Esta linha exibe um título `<h2>` na página, indicando que é uma lista de produtos disponíveis.

3. ```html
** **
```

- Abre uma div com a classe CSS "container". Esta div provavelmente é usada para agrupar e estilizar visualmente os produtos exibidos na página.
```

4. ```php
<?php
include_once '../includes/db_connection.php';
```
- Esta linha inclui o arquivo `'../includes/db_connection.php'`, que provavelmente contém a lógica para estabelecer uma conexão com o banco de dados.

**5.** ```php
try {
```
- Início de um bloco de código `try`, que é usado para tentar executar um bloco de código que pode lançar exceções.

6. ```php
$sql = "SELECT * FROM produtos";
```
- Cria uma string SQL que seleciona todos os campos (`*`) da tabela `'produtos'`.

**7.** ```php
$stmt = $conn->prepare($sql);
```
- Prepara uma declaração SQL para execução.

8. ```php
$stmt->execute();
```
- Executa a declaração preparada. Isso vai buscar todos os produtos do banco de dados.

**9.** ```php
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
```
- Inicia um loop `while` que irá iterar sobre os resultados da consulta ao banco de dados, armazenando cada linha em `$row`.

10. ```php
echo " ";
```
- Inicia a div que representa um produto na página.

**11.** ```php
echo "<img src='../uploads/{$row['imagem']}' alt='{$row['produto']}'>";
```
- Exibe uma imagem do produto. O caminho da imagem é definido pela coluna `'imagem'` da tabela de produtos e o atributo `alt` é preenchido com o nome do produto.

12. ```php
echo "<h3>{$row['produto']}</h3>";
```
- Exibe o nome do produto.
```

13. ```php
    echo "<a href='../pages/product_detail.php?id={$row['id']}'>Detalhes</a>";
    ```
    - Cria um link para a página de detalhes do produto, passando o `id` do produto via parâmetro GET na URL.

14. ```php
    echo "</div>";
    ```
    - Fecha a div que representa um produto.

15. ```php
    } catch(PDOException $e) {
    ```
    - Captura exceções do tipo `PDOException`, caso ocorram durante a execução do bloco `try`.

16. ```php
    echo "Erro na conexão com o banco de dados: " . $e->getMessage();
    ```
    - Se uma exceção for capturada, exibe uma mensagem de erro indicando o problema.

17. ```php
    ?>
    ```
    - Finaliza o bloco PHP.

18. ```html
    </div>
    ```
    - Fecha a div com a classe "container", que agrupa os produtos.

19. ```php
    <?php include_once '../templates/footer.php'; ?>
    ```
    - Inclui o conteúdo do arquivo `'../templates/footer.php'`, que provavelmente contém o rodapé HTML comum a todas as páginas do site.

Resumindo, este código PHP está sendo usado para exibir uma lista de produtos disponíveis, recuperando esses produtos de um banco de dados e exibindo seus detalhes em uma página da web.

---

2) pages/product_detail.php

```
<?php include_once '../templates/header.php'; ?>

<div class="product-detail">
  <?php
  include_once '../includes/db_connection.php';

  // Verifica se o parâmetro de ID do produto foi fornecido na URL
```
```

```
    if (isset($_GET['id'])) {
        $product_id = $_GET['id'];

        try {
            // Consulta para selecionar o produto com base no ID fornecido
            $sql = "SELECT * FROM produtos WHERE id = :id";
            $stmt = $conn->prepare($sql);
            $stmt->bindParam(':id', $product_id);
            $stmt->execute();
            $product = $stmt->fetch(PDO::FETCH_ASSOC);

            // Verifica se o produto foi encontrado
            if ($product) {
                // Exibe as informações do produto
                echo "<h2>{$product['produto']}</h2>";
                echo "<img src='../uploads/{$product['imagem']}' alt='{$product['produto']}'>";
                echo "<p>R$ {$product['valor']}</p>";
                echo "<form action='../pages/cart.php' method='post'>";
                echo "<input type='hidden' name='product_id' value='{$product['id']}'>";
                echo "<input type='hidden' name='product_name' value='{$product['produto']}'>"; // Corrigindo este campo oculto
                echo "<input type='hidden' name='product_price' value='{$product['valor']}'>"; // Corrigindo este campo oculto
                echo "<label for='quantity'>Quantidade:</label>";
                echo "<input type='number' id='quantity' name='quantity' value='1' min='1'>";
                echo "<button type='submit'>Adicionar ao Carrinho</button>";
                echo "</form>";
            } else {
                echo "Produto não encontrado.";
            }
        } catch(PDOException $e) {
            echo "Erro na conexão com o banco de dados: " . $e->getMessage();
        }
    } else {
        echo "ID do produto não fornecido.";
    }
    ?>
</div>

<?php include_once '../templates/footer.php'; ?>
```

Vamos analisar linha por linha o que este código PHP faz:

1. ```php
<?php include_once '../templates/header.php'; ?>
```

- Inclui o conteúdo do arquivo `'../templates/header.php'` no documento atual. Isso geralmente contém o cabeçalho HTML padrão de todas as páginas do site.

2. ```html
** **

```
- Abre uma div com a classe CSS "product-detail". Isso provavelmente será usada para estilizar visualmente os detalhes do produto na página.

**3.** ```php
```
<?php
include_once '../includes/db_connection.php';
```
- Inclui o arquivo `'../includes/db_connection.php'`, que contém a lógica para estabelecer uma conexão com o banco de dados.

**4.** ```php
```
if (isset($_GET['id'])) {
```
- Verifica se o parâmetro de ID do produto foi fornecido na URL. Se sim, continua com a exibição dos detalhes do produto.

**5.** ```php
```
$product_id = $_GET['id'];
```
- Obtém o valor do parâmetro 'id' passado na URL e o armazena na variável `$product_id`.

**6.** ```php
```
try {
```
- Início de um bloco `try`, usado para tentar executar um bloco de código que pode lançar exceções.

**7.** ```php
```
$sql = "SELECT * FROM produtos WHERE id = :id";
```
- Cria uma string SQL para selecionar todos os campos (`*`) da tabela 'produtos' onde o ID é igual ao valor do parâmetro `:id`.

**8.** ```php
```
$stmt = $conn->prepare($sql);
$stmt->bindParam(':id', $product_id);
$stmt->execute();
$product = $stmt->fetch(PDO::FETCH_ASSOC);
```
- Prepara e executa a consulta SQL, passando o valor do ID como um parâmetro. Em seguida, obtém os detalhes do produto da consulta e os armazena em `$product`.

**9.** ```php
```
if ($product) {
```
- Verifica se o produto foi encontrado no banco de dados.

**10.** ```php
```
echo "<h2>{$product['produto']}</h2>";
echo "<img src='../uploads/{$product['imagem']}' alt='{$product['produto']}'>";
```
```

```
   echo "<p>R$ {$product['valor']}</p>";
   ```
   - Exibe o nome do produto, a imagem e o preço.

11. ```php
   echo "<form action='../pages/cart.php' method='post'>";
   ```
   - Inicia um formulário que enviará os dados do produto para a página do carrinho.

12. ```php
   echo "<input type='hidden' name='product_id' value='{$product['id']}'>";
   echo "<input type='hidden' name='product_name' value='{$product['produto']}'>"; // Corrigindo este campo oculto
   echo "<input type='hidden' name='product_price' value='{$product['valor']}'>"; // Corrigindo este campo oculto
   ```
   - Insere campos ocultos para enviar o ID, o nome e o preço do produto para a página do carrinho.

13. ```php
   echo "<label for='quantity'>Quantidade:</label>";
   echo "<input type='number' id='quantity' name='quantity' value='1' min='1'>";
   ```
   - Exibe um campo para o usuário inserir a quantidade desejada do produto.

14. ```php
   echo "<button type='submit'>Adicionar ao Carrinho</button>";
   echo "</form>";
   ```
   - Exibe um botão de envio para adicionar o produto ao carrinho.

15. ```php
   } else {
      echo "Produto não encontrado.";
   }
   ```
   - Se o produto não for encontrado no banco de dados, exibe uma mensagem indicando isso.

16. ```php
   } catch(PDOException $e) {
      echo "Erro na conexão com o banco de dados: " . $e->getMessage();
   }
   ```
   - Captura exceções do tipo `PDOException`, caso ocorram durante a execução do bloco `try`.

17. ```php
   } else {
      echo "ID do produto não fornecido.";
   }
   ```
   - Se o parâmetro de ID do produto não foi fornecido na URL, exibe uma mensagem indicando isso.
   ```

18. ```html
** **
```

- Fecha a div "product-detail".

19. ```php
<?php include_once '../templates/footer.php'; ?>
```

- Inclui o conteúdo do arquivo `'../templates/footer.php'`, que provavelmente contém o rodapé HTML comum a todas as páginas do site.

Em resumo, o código é parte de uma aplicação de comércio eletrônico que permite aos usuários visualizar e adicionar produtos ao carrinho de compras.

---

3) pages/cart.php

```php
<?php
include_once '../templates/header.php';
session_start();

// Verifica se existe um carrinho na sessão
if (!isset($_SESSION['cart'])) {
    $_SESSION['cart'] = [];
}

// Adiciona o produto ao carrinho quando o formulário é enviado
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['product_id'])) {
    $product_id = $_POST['product_id'];
    $quantity = $_POST['quantity'];
    $product_name = $_POST['product_name'];
    $product_price = $_POST['product_price'];

    // Verifica se o produto já está no carrinho
    if (isset($_SESSION['cart'][$product_id])) {
        // Se o produto já estiver no carrinho, apenas atualize a quantidade
        $_SESSION['cart'][$product_id]['quantity'] += $quantity;
    } else {
        // Se o produto não estiver no carrinho, adicione-o com a quantidade especificada
        $_SESSION['cart'][$product_id] = [
            'name' => $product_name,
            'price' => $product_price,
            'quantity' => $quantity
        ];
    }
}

echo "<h2>Carrinho de Compras</h2>";

// Botão para limpar o carrinho
echo "<form action='' method='post'>";
echo "<input type='hidden' name='clear_cart' value='true'>";
```

```
echo "<button type='submit'>Limpar Carrinho</button>";
echo "</form>";

// Verifica se o botão "Limpar Carrinho" foi acionado
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['clear_cart'])) {
    // Remove todos os itens do carrinho
    $_SESSION['cart'] = [];
    // Recarrega a página para refletir as mudanças
    header('Location: cart.php');
    exit();
}

// Verifica se o carrinho está vazio
if (empty($_SESSION['cart'])) {
    echo "<p>O carrinho está vazio.</p>";
} else {
    // Inicializa variáveis para o total de itens e o total geral da compra
    $total_items = 0;
    $total_price = 0;

    echo "<ul>";
    foreach ($_SESSION['cart'] as $product_id => $product) {
        // Certifica-se de que $product é um array antes de acessar seus elementos
        if (is_array($product) && array_key_exists('name', $product) && array_key_exists('price', $product) && array_key_exists('quantity', $product)) {
            // Calcula o subtotal do produto
            $subtotal = $product['price'] * $product['quantity'];

            // Exibe os detalhes do produto
            echo "<li>{$product['name']} - R$ {$product['price']} x {$product['quantity']} = R$ $subtotal</li>";

            // Adiciona a quantidade de itens e o subtotal ao total geral da compra
            $total_items += $product['quantity'];
            $total_price += $subtotal;
        }
    }
    echo "</ul>";

    // Exibe o total de itens e o total geral da compra
    echo "<p>Total de Itens: $total_items</p>";
    echo "<p>Total: R$ $total_price</p>";
}

include_once '../templates/footer.php';
?>
```

Vamos analisar cada linha do código PHP fornecido:

1. ```php
   <?php
```
```

```
- Início do bloco PHP.

2. ```php
   include_once '../templates/header.php';
   ```
   - Inclui o cabeçalho do site, que provavelmente contém a estrutura HTML comum a todas as páginas, como tags `<head>` e `<header>`.

3. ```php
   session_start();
   ```
   - Inicia ou retoma a sessão do PHP. Isso é necessário para usar variáveis de sessão para manter o estado do carrinho de compras.

4. ```php
   if (!isset($_SESSION['cart'])) {
       $_SESSION['cart'] = [];
   }
   ```
   - Verifica se existe uma variável de sessão chamada `'cart'`, que armazena os itens do carrinho de compras. Se não existir, inicializa-a como um array vazio.

5. ```php
   if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['product_id'])) {
   ```
   - Verifica se o método de requisição é POST e se foi enviado um campo `product_id` no formulário. Isso indica que um produto está sendo adicionado ao carrinho.

6. ```php
   $product_id = $_POST['product_id'];
   $quantity = $_POST['quantity'];
   $product_name = $_POST['product_name'];
   $product_price = $_POST['product_price'];
   ```
   - Obtém os valores dos campos do formulário POST que representam o ID, a quantidade, o nome e o preço do produto a ser adicionado ao carrinho.

7. ```php
   if (isset($_SESSION['cart'][$product_id])) {
   ```
   - Verifica se o produto já está presente no carrinho de compras.

8. ```php
   $_SESSION['cart'][$product_id]['quantity'] += $quantity;
   ```
   - Se o produto já estiver no carrinho, apenas atualiza a quantidade do produto.

9. ```php
   } else {
       $_SESSION['cart'][$product_id] = [
```
```

```
        'name' => $product_name,
        'price' => $product_price,
        'quantity' => $quantity
    ];
}
```

- Se o produto não estiver no carrinho, adiciona-o ao carrinho com a quantidade especificada.

10. ```php
    echo "<h2>Carrinho de Compras</h2>";
    ```
    - Exibe um título indicando que é a seção do carrinho de compras.

11. ```php
    echo "<form action='' method='post'>";
    ```
    - Inicia um formulário para permitir a limpeza do carrinho.

12. ```php
    echo "<input type='hidden' name='clear_cart' value='true'>";
    echo "<button type='submit'>Limpar Carrinho</button>";
    echo "</form>";
    ```
    - Cria um botão dentro do formulário para limpar o carrinho de compras.

13. ```php
    if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['clear_cart'])) {
    ```
    - Verifica se o formulário para limpar o carrinho foi enviado.

14. ```php
    $_SESSION['cart'] = [];
    ```
    - Remove todos os itens do carrinho de compras.

15. ```php
    header('Location: cart.php');
    exit();
    ```
    - Redireciona o usuário de volta para a página do carrinho após limpar o carrinho.

16. ```php
    if (empty($_SESSION['cart'])) {
        echo " O carrinho está vazio. ";
    } else {
    ```
    - Verifica se o carrinho está vazio ou não.

17. ```php
    foreach ($_SESSION['cart'] as $product_id => $product) {
    ```

- Itera sobre todos os itens presentes no carrinho de compras.

18. ```php
```php
if (is_array($product) && array_key_exists('name', $product) && array_key_exists('price', $product) && array_key_exists('quantity', $product)) {
```
- Verifica se o item do carrinho é um array e se ele contém chaves específicas (nome, preço e quantidade).

19. ```php
```php
$subtotal = $product['price'] * $product['quantity'];
```
- Calcula o subtotal do produto (preço multiplicado pela quantidade).

20. ```php
```php
echo "<li>{$product['name']} - R$ {$product['price']} x {$product['quantity']} = R$ $subtotal</li>";
```
- Exibe os detalhes do produto, incluindo nome, preço, quantidade e subtotal.

21. ```php
```php
$total_items += $product['quantity'];
$total_price += $subtotal;
```
- Atualiza o total de itens e o preço total da compra.

22. ```php
```php
echo "</ul>";
```
- Fecha a lista de itens do carrinho.

23. ```php
```php
echo "<p>Total de Itens: $total_items</p>";
echo "<p>Total: R$ $total_price</p>";
```
- Exibe o total de itens e o preço total da compra.

24. ```php
```php
include_once '../templates/footer.php';
```
- Inclui o rodapé do site, que provavelmente contém o fechamento das tags HTML, scripts e outros elementos comuns a todas as páginas.

25. ```php
```php
?>
```
- Fim do bloco PHP.

Em essência, o código permite aos usuários adicionar produtos ao carrinho, visualizar o conteúdo do carrinho, atualizar quantidades e remover produtos, proporcionando uma experiência de compra interativa em um site.