**Supprimer un produit**

**Mettre à jour le fichier index.ejs  
<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<title>Products</title>**

**<style>**

**table {**

**width: 100%;**

**border-collapse: collapse;**

**}**

**table, th, td {**

**border: 1px solid black;**

**}**

**th, td {**

**padding: 10px;**

**text-align: left;**

**}**

**button {**

**padding: 5px 10px;**

**cursor: pointer;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>Products</h1>**

**<table>**

**<thead>**

**<tr>**

**<th>Name</th>**

**<th>Price</th>**

**<th>Category</th>**

**<th>Actions</th>**

**</tr>**

**</thead>**

**<tbody>**

**<% products.forEach(product => { %>**

**<tr>**

**<td><%= product.name %></td>**

**<td>$<%= product.price %></td>**

**<td><%= product.category ? product.category.name : 'No category' %></td>**

**<td>**

**<!-- Action Modifier -->**

**<a href="/products/edit/<%= product.id %>">**

**<button>Edit</button>**

**</a>**

**<!-- Action Supprimer -->**

**<form action="/products/<%= product.id %>" method="POST" style="display:inline;">**

**<input type="hidden" name="\_method" value="DELETE">**

**<button type="submit" onclick="return confirm('Are you sure you want to delete this product?')">Delete</button>**

**</form>**

**</td>**

**</tr>**

**<% }) %>**

**</tbody>**

**</table>**

**<!-- Formulaire pour ajouter un produit -->**

**<h2>Add Product</h2>**

**<form action="/products" method="POST">**

**<input type="text" name="name" placeholder="Product Name" required>**

**<input type="number" step="0.01" name="price" placeholder="Product Price" required>**

**<select name="categoryId">**

**<% categories.forEach(category => { %>**

**<option value="<%= category.id %>"><%= category.name %></option>**

**<% }) %>**

**</select>**

**<button type="submit">Add Product</button>**

**</form>**

**</body>**

**</html>  
  
Ajouter la fonction delete dans le fichier productsRoutes**// Route DELETE pour supprimer un produit

router.delete('/:id', async (req, res) => {

const { id } = req.params; try {

await ProductService.deleteProduct(id);

res.redirect('/products'); // Rediriger vers la liste des produits après suppression } catch (error) { res.status(400).json({ error: error.message }); } });

**Method-override :**

Le middleware method-override permet de remplacer la méthode d'une requête HTTP lorsque vous recevez une requête POST contenant un champ caché \_method. Cela vous permet de contourner la limitation des formulaires HTML et d'envoyer des requêtes DELETE (et d'autres méthodes HTTP comme PUT) en utilisant un champ caché dans un formulaire.

**npm install method-override**

**Mettre à jour app.js**

const methodOverride = require('method-override');

// Middleware pour gérer la méthode \_method

app.use(methodOverride('\_method'));

**Mettre à jour repositories/ProductRepository.js**

**// Récupérer un produit par ID**

**async getProductById(id) { return await Product.findByPk(id); }**

**// Supprimer un produit par ID**

**async deleteProduct(id) {**

**const product = await this.getProductById(id);**

**if (!product) { throw new Error('Product not found');**

**} return await product.destroy(); }**

**Mettre à jour productService**// Méthode pour supprimer un produit

async deleteProduct(id) { const product = await ProductRepository.getProductById(id); if (!product) { throw new Error('Product not found'); } await ProductRepository.deleteProduct(id); }