Lista 7 de Exercício – Node – Express- Express-Handlenav

Desenvolvimento Web – Lab. Profª Angelina V S Melaré

Faça download do MongoCompass para visualizar o banco de maneira gráfica.

Siga os passos a seguir para desenvolver um projeto parecido com o da aula anterior.

1) Crie uma pasta para o projetoCadastro. Vá até a pasta e crie um projeto sem modelo, inserindo o comando:

   npm init –y

2) Insira os pacotes necessários para criar o projeto

   npm install mongodb nodemon mongoose express express-handlebars

3) Configure o arquivo de configuração do pacote (package.json), definindo que o nodemon iniciará executando o arquivo index.js na porta 3000 do servidor local:

   "start": "nodemon ./index.js localhost 3000"

4) Crie as pastas: db, controllers, models, views, routes, public. Dentro da pasta views crie as pastas layouts (para definir o layout geral) e a pasta produtos (para definir o layout de consulta e cadastro).

5) A pasta public terá apenas o arquivo style.css

6) Crie o **arquivo index.js na pasta raiz do projeto** e configure nele o express e o express-handlebars

```
const express = require('express')
const exphbs = require('express-handlebars')
const app = express()
const produtosRoutes = require('./routes/produtosRoutes')
app.engine('handlebars', exphbs.engine())
app.set('view engine', 'handlebars')
app.use(express.urlencoded({extended: true}),)
app.use(express.json())
app.use(express.static('public'))
app.use('/produtos', produtosRoutes)
app.listen(3000)
```

7) **Na pasta db crie o arquivo conn.js** que configuração da configuração com banco de dados MongoDB por meio do Mongoose:

```
const mongoose = require('mongoose')
async function main() {
  await mongoose.connect('mongodb://localhost:27017/DBCadastro')
  console.log('Conectando ao DB do Mongo!')
}
main().catch((err) => console.log(err))
module.exports = mongoose
```

8) **Dentro da pasta model crie o arquivo Produto.js** de configuração da coleção produto

```
const mongoose = require('../db/conn')
const { Schema } = mongoose
const Produto = mongoose.model(  'Produtos',  new Schema({
    nome:      { type: String, required: true,  },
    valor:     { type: Number,  required: true,   },
    descricao: { type: String, required: true,   },
    imagem: {type: String,  required: true ,},
  }),)

module.exports = Produto
```

9) **Crie na pasta controllers o arquivo ProdutoController.js** as operações de cadastro e consulta dos produtos

```
const Produto = require('../models/Produto')

module.exports = class ProdutoController {

 static async showProdutos(req, res) {
  const produtos = await Produto.find({}).lean()
  res.render('produtos/all', { produtos })
 }

 static async createProduto(req, res) {
  res.render('produtos/create')
 }

 static async createProdutoPost(req, res) {
  const nome = req.body.nome
  const valor = req.body.valor
  const descricao = req.body.descricao
  const imagem = req.body.imagem
  const produto = new Produto({ nome, valor, descricao, imagem })
  await produto.save()
  res.redirect('/produtos')
 }

  static async getProduto(req, res) {
  const id = req.params.id
  const produto = await Produto.findById(id).lean()
  res.render('produtos/produto', { produto })
 }

static async removeProduto(req, res) {
  const id = req.params.id
  await Produto.deleteOne({ _id: id })
  res.redirect('/produtos')
 }
```

```
static async editProduto(req, res) {
    const id = req.params.id
    const produto = await Produto.findById(id).lean()
    res.render('produtos/edit', { produto })
  }

  static async editProdutoPost(req, res) {
    const id = req.body.id
    const nome = req.body.nome
    const valor = req.body.valor
    const descricao = req.body.descricao
    const imagem = req.body.imagem
    const produto = { nome, valor, descricao, imagem }
    await Produto.updateOne({ _id: id }, produto)
    res.redirect('/produtos')
  }
}
```

10) **Na pasta routers crie o arquivo produtosRoutes.js** e configure as rotas de acordo com os controles criados.

```
const express = require('express')
const router = express.Router()
const ProdutoController = require('../controllers/ProdutoController')

router.post('/edit', ProdutoController.editProdutoPost)
router.get('/', ProdutoController.showProdutos)
router.get('/produtos', ProdutoController.showProdutos)
router.get('/create', ProdutoController.createProduto)
router.post('/create', ProdutoController.createProdutoPost)
router.get('/:id', ProdutoController.getProduto)
router.post('/remove/:id', ProdutoController.removeProduto)
router.get('/edit/:id', ProdutoController.editProduto)

module.exports = router
```

11) Agora vamos ao **frontend** criar as interfaces, começando pela principal que conterá as outras interfaces. **Crie dentro da pasta layout o arquivo main.handlebars** Veja que a extensão tem que ser handlebars. Este arquivo terá o padrão de um arquivo index.html, podendo estar linkada a um arquivo .css e buscando classes do Bootstrap. Veja que o diferencial é que todo o conteúdo a ser mostrado será renderizado dentro do **{{{body}}}**.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
```

```
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>CRUD Produtos Node Mongo</title>
  <link rel="stylesheet" href="/css/styles.css">
 </head>
 <body>
  <nav id="navbar">
   <ul>
    <li><a href="/produtos/">Visualizar Produtos</a></li>
    <li><a href="/produtos/create">Cadastrar Produtos</a></li>
   </ul>
  </nav>
  <div class="container">
   {{{ body }}}
  </div>
 </body>
</html>
```

12) Para visualizar configurar as interfaces do produto crie seus arquivos dentro da pasta produtos que está views. Teremos que criar os arquivos de mostrar todos os produtos, de edição e criação.

Crie o arquivo all.handlebars

```
<h1>Todos os produtos</h1>
<div class="produtos">
 {{#each produtos}}
  <div class="produto">
   <img src="{{this.imagem}}" alt="{{this.nome}}">
   <h3>{{this.nome}}</h3>
   <p>R${{this.valor}}</p>
   <div class="actions">
    <a href="/produtos/{{this._id}}" class="btnPrin">Detalhes</a>
    <a href="/produtos/edit/{{this._id}}" class="btnPrin">Editar</a>
   </div>
  </div>
 {{/each}}
</div>
```

Crie o arquivo produto.handlebars

```
<div class="produto-page">
 <h1>{{produto.nome}}</h1>
 <p>Preço: R${{produto.valor}}</p>
 <img src="{{produto.imagem}}" alt="{{produto.nome}}">
 <p>Descrição: {{produto.descricao}}</p>
 <form action="/produtos/remove/{{produto._id}}" method="POST">
  <input type="submit" class="btn" value="Excluir">
 </form>
</div>
```

Crie o arquivo edit.handlebars

```
<h1>Editar produto</h1>
<form action="/produtos/edit" method="POST">
  <input type="hidden" name="id" value={{produto._id}}>
  <div class="form-control">
    <label for="nome">Nome do produto:</label>
    <input type="text" name="nome" placeholder="Digite o nome do produto" value="{{produto.nome}}">
  </div>
  <div class="form-control">
    <label for="valor">Preço do produto:</label>
    <input type="text" name="valor" placeholder="Digite o preço" value="{{produto.valor}}">
  </div>
  <div class="form-control">
    <label for="descricao">Descrição do produto:</label>
    <textarea name="descricao" placeholder="Descrição do produto">{{produto.descricao}}</textarea>
  </div>
  <div class="form-control">
    <label for="image">Imagem:</label>
    <input type="text" name="image" placeholder="Insira a URL da imagem" value="{{produto.imagem}}" />
  </div>
  <input type="submit" class="btn" value="Salvar">
</form>
```

Crie o arquivo create.handlebars

```
<h1>Criar produto</h1>
<form action="/produtos/create" method="POST">
  <div class="form-control">
    <label for="nome">Nome do produto:</label>
    <input type="text" name="nome" placeholder="Digite o nome do produto">
  </div>
  <div class="form-control">
    <label for="valor">Preço do produto:</label>
    <input type="text" name="valor" placeholder="Digite o preço">
  </div>
  <div class="form-control">
    <label for="descricao">Descrição do produto:</label>
    <textarea name="descricao" placeholder="Descrição do produto"></textarea>
  </div>
  <div class="form-control">
    <label for="imagem">Imagem:</label>
    <input type="text" name="imagem" placeholder="Insira a URL da imagem" />
  </div>
  <input type="submit" class="btn" value="Cadastrar">
</form>
```

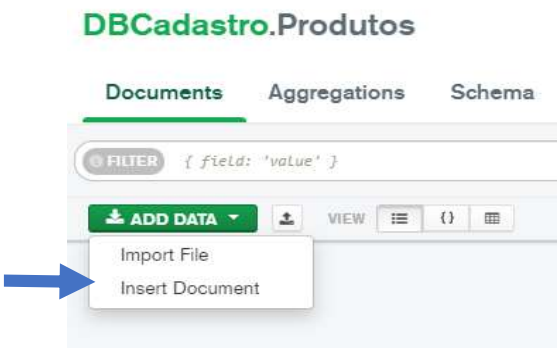
**** Não se esqueça que para ver o projeto em execução te que dar um start com o npm start**

** No browser coloque a rota para começar
http://localhost:3000/produtos

13) Não se esqueça de configurar o estilo. Crie o arquivo styles.css dentro da pasta public.

```
* {  font-family: Helvetica, sans-serif;  margin: 0;   padding: 0;  background-color: rgb(147, 182, 181);  color: #fff;}

#navbar {  padding: 1.2em 2.4em;  display: flex;  justify-content: space-between;  align-items: center;}

#navbar ul {  display: flex;    list-style: none;}

#navbar ul li {  margin-left: 3em;}

a {  text-decoration: none;}

.container {  padding: 2em;}

.container h1 {  margin-bottom: 2em;}

form {  max-width: 300px;}

.form-control {  display: flex;  flex-direction: column;}

.form-control label {  font-weight: bold;  margin-bottom: 0.4em;}

.form-control input,.form-control textarea {  border: 1px solid #fff;  padding: 5px;  margin-bottom: 1em;}

.btnPrin {
 background-color: rgb(90, 167, 177);  color: rgb(195, 187, 216);  padding: 2px 2px;  border: none;  margin: 10px;  cursor: pointer;}

.btn {
 background-color: rgb(118, 55, 55);
 color: rgb(195, 187, 216);  padding: 5px 10px;  border: none;  margin-top: 1em;  cursor: pointer;}

.produtos {  display: flex;  flex-wrap: wrap;}

.produto {  width: 23%;  margin: 1%;  text-align: center;  margin-bottom: 1em;}

.produto img {  max-width: 100%;  margin-bottom: 0.5em;}

.produto h3 {  margin-bottom: 1em;}

.produto-page img {  max-width: 250px;  margin-bottom: 1em;}

.produto-page p {  margin-bottom: 1em;}
```

14) Vá no Mongo Compass e insira um produto diretamente pelo ambiente



Vá na opção de inserir o document. Digite o json de inserção e depois pressione o botão Insert.

```
{
"nome":"Abobora",
"valor": 2,
"Descricao": "Moranga",
"imagem": "url de uma imagem da internet"
}
```

** Copie o endereço de uma imagem qualquer para o campo imagem