



UNIVERSIDADE DE SÃO CARLOS
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO
SCC0219 - Introduction to Web Development - 2ºSEM/2021

Grupo 4

Hugo de Azevedo Vitulli - 10295221

hugovitulli@usp.br

Matheus Vinicius - 10295217

mt_godoi@usp.br

Henrique Hiram Libutti Núñez - 11275300

henrique.nunez@usp.br

Prof. Dr. Dilvan de Abreu Moreira

1.Requirements

Além dos requerimentos previamente determinados pela proposta de trabalho, será necessário adicionar uma categorização dos calçados para que possamos implementar as funcionalidades relacionadas a filtragem por categoria dos produtos.

2. Project Description

Neste projeto implementamos as funcionalidades planejadas de cadastro de usuário, login e autenticação, verificação dos detalhes dos produtos, adição/remoção de produtos no carrinho, compra dos produtos, histórico de compra de cada cliente, para administradores ainda podemos editar informações dos usuários e deletar contas criadas no sistema pelo próprio site.

- **Mockup e Diagrama de Navegação:**

Não foi desenvolvida uma tela de conclusão de compra no mock up, após clicar em “checkout” do carrinho assumimos que a compra foi completa sem precisar de informações adicionais de cartão e compra.

[Figma project - Outsider Footwear](#)

[Figma presentation - Outsider Footwear](#)

- **Informação salva do Banco:**

Estamos salvando informações de cadastro dos usuários, sendo elas:

- email, senha, endereço, telefone, data de nascimento, informação se a conta é admin ou não;

Dos produtos estamos salvando:

- Título, preço, informação de estoque, imagem do produto, categoria do produto;

E por fim, também salvamos as transações do site:

- Cliente que comprou, itens comprados, data da compra, estado da compra;

3. Comments About the Code:

Na terceira milestone fizemos a implementação do backend, e todas funcionalidades previstas no site. Com isso, separamos o projeto em duas pastas:

backend:

- Possui todo código para utilizar nosso MongoDB, buscar e cadastrar dados
- Feito em JavaScript, por falta de tempo não conseguimos passar para TypeScript como o resto do frontend, porém fizemos tipagens globais e facilitou o uso dos nossos dados do backend
- Para autenticar o usuário como admin, criamos uma flag chamada “admin”, recebendo um booleano. Apenas os desenvolvedores possuem acesso a essa flag, garantindo que apenas eles conseguem criar um novo admin no sistema.
- Não realizamos autenticação para garantir o input de dados. (Por exemplo, a senha conter um número mínimo de dígitos; e-mail ser um e-mail válido; etc.)

frontend:

- O front de todo o sistema, com nossas páginas, rotas e funcionalidades
 - Utilizamos uma página para se passar por modal dos produtos, ao invés de criar um modal próprio
 - Também fizemos nossa própria autenticação, que pode não ser a mais segura
 - Geramos rotas espelhadas para cada tipo de cliente: Usuário comum e Administradores, com suas próprias funcionalidades.

4. Test Plan:

Utilizamos o Insomnia para testar as rotas do nosso sistema, e analisar as respostas providas pelo nosso MongoDB.

5. Test Results:

Presente no final do documento.

6. Build Procedures:

Para fazer o build do projeto e executar, temos que seguir alguns passos. Primeiro, temos que instalar os pacotes do projeto, separadamente na pasta do **backend** e no **frontend**.

Basta entrar na pasta correspondente e rodar os comandos:

- **yarn install**

ou

- **npm install global**

Dentro das duas pastas.

Após terminada a instalação, basta executar o backend e o frontend com o seguinte comando:

- **yarn start**

ou

- **npm start**

Novamente, precisamos rodar esse comando de start em cada uma das pastas, uma vez para iniciar o backend e outra para iniciar o frontend.

7. Problems:

Enfrentamos alguns problemas iniciais com a fonte de dados, tivemos uma certa dificuldade inicialmente para entender como extrair todos os dados que gostaríamos de utilizar do banco e também como armazenar os dados quando necessário.

Também enfrentamos dificuldade em realizar o “re-render” de várias páginas, quando ocorre uma alteração. Após estudar conseguimos solucionar esse problema com o uso do método “**useEffect**” e o local storage do site para conseguir atualizar as informações da página sem ser necessário redirecionar o usuário ou recarregar a página.

Também enfrentamos alguns problemas menores em usar React, já que não temos tanta experiência com o framework como um todo e com todas suas bibliotecas. Um exemplo disso foi o modal que aparece quando clicamos em um produto, não tínhamos conhecimento sobre a mesma anteriormente, então tivemos certos empecilhos para utilizar tal biblioteca.

Fora isso, enfrentamos alguns problemas que estamos cientes em algumas páginas, e anotamos na seguinte lista:

- **ManageAccount:** não estamos verificando os dados que são atualizados, então o usuário pode manter os campos vazios da sua conta, e enviar essas informações para o banco, que irá atualizar sua conta com esses dados.
- **ManageAccount:** não conseguimos carregar o valor do aniversário do usuário na página do perfil.
- **ManageAccount:** raramente as informações do usuário não são carregadas corretamente, e não sabemos exatamente o porquê.
- **SignIn/Profile:** devido a autenticação, mantivemos dois caminhos para todas as páginas, ou seja, temos uma página de perfil separada para o usuário e outra para o admin.
- **ManageProducts:** produtos cadastrados não possuem imagens registradas, apenas uma placeholder, por não conseguir utilizar os **blobs** do mongoDB para armazenar as imagens.
- **PurchaseHistory:** não conseguimos renderizar uma nested list das ordens de compra, então, o histórico de compra dos usuários contém apenas o ID da compra, sua data de criação e o estado da compra.

8. Comments:

Estamos cientes de algumas falhas de segurança por todo o sistema, além de alguns problemas de alinhamento em alguns itens no site, que poderiam ser refinados, porém, consideramos mais importante completar as funcionalidades do projeto do que utilizar o tempo para corrigir tais falhas.

Não implementamos o fluxo de compras (informações de cartão, geração de boleto, parcelamento, etc.), estamos apenas utilizando a autenticação do usuário e autenticação do produto para gerar uma compra. Estamos fazendo uma verificação de estoque para impedir a compra de mais unidades do que em estoque, e atualizações pelo admin serão refletidas em todo o site.

Com esse trabalho conseguimos desenvolver o aprendizado de CRUD, e desenvolvimento Front End, além, de conseguir relacionar isso com a visão do cliente, e

também utilização de ferramentas como o framework React, MongoDB, Node, Typescript, etc.

Como não temos experiência prévia com desenvolvimento web, sentimos um pouco de dificuldade na gestão do tempo do desenvolvimento, portanto não conseguimos implementar funções mais complexas e nos atentamos ao básico para desenvolver o aprendizado da disciplina.

Testes para Customers

Tipo do teste	Body	Esperado	Obtido
GET /customers	--	[{ "_id": "61aac983f5bf47ba3eb1d070", "name": "MAXIMUS ADMIN V2", "email": "admin@email.com", "password": "1", "address": "Rua do Admin", "phone": "9090 190", "admin": true, "status": true, "v": 0, "birthday": null, "gender": "male" }, { "_id": "61aac9b1f5bf47ba3eb1d072", "name": "Mega Brinde", "email": "brinde@email.com", "password": "1", "address": "Rua do Admin", "phone": "9090 190", "admin": false, "status": false, "v": 0, "birthday": null, "gender": "male" }]	[{ "_id": "61aac983f5bf47ba3eb1d070", "name": "MAXIMUS ADMIN V2", "email": "admin@email.com", "password": "1", "address": "Rua do Admin", "phone": "9090 190", "admin": true, "status": true, "v": 0, "birthday": null, "gender": "male" }, { "_id": "61aac9b1f5bf47ba3eb1d072", "name": "Mega Brinde", "email": "brinde@email.com", "password": "1", "address": "Rua do Admin", "phone": "9090 190", "admin": false, "status": false, "v": 0, "birthday": null, "gender": "male" }]

POST /customers	{ "name": "Shots", "email": "shots@email.com", "password": "1", "address": "Rua do Admin", "phone": "9090 190", "status": false }	{ "message": "Cliente cadastrado com sucesso" }	{ "message": "Cliente cadastrado com sucesso" }
PUT /customers/id	{ "name": "admin" }	{ "message": "Cliente cadastrado com sucesso" }	{ "message": "Cliente cadastrado com sucesso" }

Testes para Products

Tipo do teste	Body	Esperado	Obtido
GET /products ou /products/id	-/-	[{ "id": "61b10efa5fb01435d0112e3e", "title": "Tênis Caminhada Confortável Preto", "description": "lorem ipsum dolor", "slug": "tenis-caminhada-confortável-preto", "image": "https://raw.githubusercontent.com/MatheusGodoi/Grupo4-OutsiderFootwear/main/Milestone_2/outsider-footwear/src/assets/js/sondb/tenis_caminhada_2.jpg", "price": 139.9, "active": true, "amount": 5, "category_main": "casual", "category_secondary": "sports", "v": 0 }]	[{ "id": "61b10efa5fb01435d0112e3e", "title": "Tênis Caminhada Confortável Preto", "description": "lorem ipsum dolor", "slug": "tenis-caminhada-confortável-preto", "image": "https://raw.githubusercontent.com/MatheusGodoi/Grupo4-OutsiderFootwear/main/Milestone_2/outsider-footwear/src/assets/js/sondb/tenis_caminhada_2.jpg", "price": 139.9, "active": true, "amount": 5, "category_main": "casual", "category_secondary": "sports", "v": 0 }]

POST /products	{ "title": "Meia cano longo", "slug": "meia-cano-longo", "description": "lorem ipsum dolor", "price": 89.9, "image": "https://github.com/MatheusGodoi/Grupo4-OutsiderFootwear/blob/main/Milestone_2/outsider-footwear/src/assets/jsontdb/tenis_acessorio_2.jpg?raw=true", "amount": 14, "category_secondary": "Accessories" }	{ "message": "Produto cadastrado com sucesso" }	{ "message": "Produto cadastrado com sucesso" }
PUT /products/id	{ "amount": 10 }	{ "message": "Produto alterado com sucesso" }	{ "message": "Produto alterado com sucesso" }
DELETE /products/id	-/-	{ "message": "Produto removido com sucesso" }	{ "message": "Produto removido com sucesso" }

Teste para Orders

Tipo do teste	Body	Esperado	Obtido
---------------	------	----------	--------

GET /orders	-//-	<pre>[{ "_id": "61afd58d52beb0328329110a", "customer": { "_id": "61aac983f5bf47ba3eb1d070", "name": "MAXIMUS ADMIN" }, "status": "created", "items": [{}, { "product_id": { "_id": "61a7f2fc8579281d9d421a1a", "title": "Tênis Caminhada Confortável Rosa", "image": "https://raw.githubusercontent.com/MatheusGodoi/Grupo4-OutsiderFootwear/main/Milestone_2/outsider-footwear/src/assets/jsontdb/tenis_caminhada_3.jpg", "price": 219.9, "amount": 5 } }] }</pre>	<pre>[{ "_id": "61afd58d52beb0328329110a", "customer": { "_id": "61aac983f5bf47ba3eb1d070", "name": "MAXIMUS ADMIN" }, "status": "created", "items": [{}, { "product_id": { "_id": "61a7f2fc8579281d9d421a1a", "title": "Tênis Caminhada Confortável Rosa", "image": "https://raw.githubusercontent.com/MatheusGodoi/Grupo4-OutsiderFootwear/main/Milestone_2/outsider-footwear/src/assets/jsontdb/tenis_caminhada_3.jpg", "price": 219.9, "amount": 5 } }] }</pre>
POST /orders	<pre>{ "customer": "61aac983f5bf47ba3eb1d070", "items": [{ "quantity": 15, "product": "61b10efa5fb01435d0112e3e" }] }</pre>	<pre>{ "message": "Pedido cadastrado com sucesso" }</pre>	<pre>{ "message": "Pedido cadastrado com sucesso" }</pre>