ClosetConnect

Tecnologias e Programação Web – 2023/2024

Bernardo Pinto, 105926
José Mendes, 107188
Filipe Obrist, 107471

Índice

Introdução e Conceito 3
Estrutura 4
Server-Side: 4
Client-Side: 4
Funcionalidades 5
Utilizador sem conta iniciada 5
Utilizador normal com conta iniciada 5
Administrador 5
Funcionalidades em cada página: 6
Deploy 7
Utilizadores Disponíveis 7

Introdução e Conceito

O ClosetConnect surgiu da visão de combinar aspetos essenciais de uma rede social com as funcionalidades práticas de uma loja online. A proposta central da plataforma é permitir que os utilizadores não só se conectem e interajam, mas também publiquem produtos que já não utilizam para venda. Esta abordagem proporciona uma experiência única, unindo o poder das redes sociais com a praticidade de comprar e vender produtos usados.

Inicialmente, vale ressaltar que o desenvolvimento deste projeto é uma reimplementação do primeiro projeto da disciplina de Tecnologias de Programação Web. Para isso, foram desenvolvidas duas aplicações independentes: uma server-side, implementada com Django REST, e uma client-side, desenvolvida com as tecnologias do Angular Framework.

Esta abordagem dupla visa aprimorar a eficiência do projeto, isolando responsabilidades para alcançar características fundamentais de um sistema, como escalabilidade e facilidade de manutenção. A aplicação desenvolvida com Django REST estabelece uma base sólida para o gerenciamento de dados e a execução consistente da lógica de negócios. Além das marcantes características do framework, como a serialização eficiente e visão baseada em classes. Enquanto isso, a aplicação client-side oferece uma interface de usuário dinâmica e responsiva, aproveitando a lógica de componentes proporcionada pelo TypeScript e Angular. Essa divisão de responsabilidades não apenas otimiza o desenvolvimento, mas também permite uma evolução mais ágil e adaptável do sistema como um todo.

Estrutura

Como mencionado anteriormente, o projeto está separado em dois ambientes distintos que se comunicam através do protocolo HTTP.

Server-Side:

Na abordagem server-side, construímos uma API capaz de receber requisições HTTP e realizar as ações necessárias. Inicialmente, concentramo-nos no desenvolvimento dos Models do sistema, classes Python que representam as principais entidades e são mapeadas diretamente pelo ORM (Object-Relational Mapping) para a base de dados.

Em seguida, estabelecemos os pares url-view, especificando nas URLs as rotas que a aplicação Django deve mapear e qual view deve processá-la. Utilizando a notação *@api_view* na view, conseguimos, com base no método HTTP introduzido (GET, POST, PUT ou DELETE), executar a operação desejada. Isso engloba desde a criação de um novo produto até a adição de um item ao carrinho ou a ação de seguir outro utilizador. Essa estratégia proporciona uma organização clara e eficiente das funcionalidades implementadas pela aplicação.

De modo a permitir autenticação dos utilizadores, estamos a usar o módulo presente na REST Framework do Django, rest_framework.authtoken. Este permite gerar tokens de autenticação sempre que um user se regista ou dá login.

Por fim, realizamos todas as configurações necessárias para o correto funcionamento do Django REST, incluindo a configuração do arquivo "settings", a criação de "serializers", as configurações específicas do admin, os métodos de autenticação, entre outros. Essa abordagem abrangente assegura a integridade e a coesão do sistema, garantindo uma API robusta e funcional.

Client-Side:

Na abordagem client-side recorremos ao Angular. Graças aos princípios desta framework, foi nos possível criar componentes para cada página da aplicação, seguindo uma estrutura que simplifica a manutenção do código e a sua própria expansão à medida que fomos evoluindo a app. Foi também explorada a capacidade de reutilização dos componentes através da criação de componentes fáceis de incorporar em diferentes partes da aplicação (ex: product e profile-header).

Além dos componentes, introduzimos serviços para gerir a comunicação entre o front-end e o back-end. A nomenclatura de cada serviço retrata a interface relacionada com as operações que este fornece (ex: user.service encapsula operações relacionadas com a interface user).

Funcionalidades

Utilizador sem conta iniciada

- Registar
- Iniciar Sessão

Utilizador normal com conta iniciada

- Terminar Sessão
- Ver Produtos e os seus detalhes
- Adicionar/Remover produto ao carrinho
- Ver carrinho
- Comprar produtos no carrinho
- Adicionar/Remover/Ver produtos aos favoritos
- Ver/Atualizar/Apagar perfil
- Ver o perfil de utilizadores
- Seguir/Deixar de seguir utilizadores
- Adicionar/Remover comentários sobre um utilizador
- Vender um produto
- Editar/Eliminar um produto

Administrador

- Mesmas funcionalidades que um utilizador normal
- Remover produtos do website
- Remover comentários do website
- Banir utilizadores
- Pesquisar por utilizadores/produtos

Funcionalidades em cada página

- Em todas as páginas temos uma **Navbar** em que, caso não tenhamos conta iniciada, é possível efetuar log in/registo. Caso contrário, permite-nos navegar para a página do nosso utilizador, visualizar os produtos favoritos, editar o nosso perfil, adicionar um novo produto ou dar log out.
- A página **Login** permite ao utilizador realizar login no website.
- A página **Register** permite ao utilizador registar-se no website.
- A página **Cart** permite ao utilizador visualizar o seu carrinho com os produtos que lá colocou. Podemos remover produtos já não desejados e finalizar a compra.
- A página **Admin** apenas pode ser utilizada por administradores. Esta permite visualizar todos os utilizadores e produtos disponíveis no website, assim como, pesquisar por estes ou removê-los caso seja a sua intenção.
- A página **Account Settings** permite ao utilizador atualizar o seu perfil, foto e palavra-passe. Caso pretenda, pode também apagar a mesma.
- A página **Favorites** possui os produtos que o utilizador marcou como favoritos, permitindo visualiza-os ou adicioná-los ao carrinho.
- A página **Account Profile** permite ao utilizador visualizar o número de pessoas que segues e que o seguem, bem como, os seus próprios produtos, podendo apagá-los.
- A página **Account Product** permite ao visualizar algumas estatísticas de um produto do utilizador, permitindo também editá-lo.
- A página **Product** possui um produto de outro utilizador. Nesta podemos adicionar/remover o produto dos favoritos, adicionar este produto ao carrinho. Relativamente ao dono do produto, podemos visualizar outros produtos deste, visualizar o seu perfil e, por fim, segui-lo ou deixar de o seguir.
- A página **Cart ConfirmOrder** permite efetuar o pedido destes produtos.
- A página **Account Product Edit** permite editar um produto do próprio utilizador.
- A página **Profile** permite visualizar o perfil de outro utilizador, juntamente com os seus produtos. Aqui poderá adicionar/remover produtos aos favoritos, visualizá-los, seguir/deixar de seguir o utilizador e, por fim, adicionar comentários ou removê-los caso sejam dele.
- A página **Add Product** permite ao utilizador adicionar um novo produto a sua conta.

Deploy

Ambos os ambientes, isto é, o server e o client, encontram-se online. Para o server-side, utilizamos o PythonAnywhere, tal como foi aconselhado pelo professor e como foi utilizado no primeiro trabalho prático. Para o client-side, utilizamos o Netlify, uma vez possui várias vantagens e funcionalidades que o tornaram mais simples de utilizar em comparação com outros hosts.

O website encontra-se a funcionar corretamente, no entanto, como utilizamos várias imagens armazenadas na base de dados, estas têm de ser enviadas através de bytes o que pode ser um pouco demorado, daí o website ser um pouco mais lento. Localmente este problema não acontece, uma vez, que não há transferência de bytes pela rede.

Os links para aceder a estes ambientes são:

- Client-Side: https://closetconnect-tpw.netlify.app
- Server-Side: http://zemendes.pythonanywhere.com

Utilizadores Disponíveis

Utilizador normal:

- Username: joao; Password: user1234
- Username: jose; Password: user1234

Administrador:

- Username: manel123; Password: user1234