Grupo:

Luís Filipe Vasconcelos Peres	nUSP: 10310641
Marcelo Duchene	nUSP: 8596351
Michelle Wingter da Silva	nUSP: 10783243

PROJECT REPORT - PET SHOP FINAL VERSION

Índice:

1. **Requirements**:

Para este projeto, foi desenvolvido um website single-page application completo com as seguintes funcionalidades:

- Página inicial com uma descrição dos serviços / produtos oferecidos e área de login (para tanto cliente ou admin);
- Se o usuário for um cliente:
    - Tela com ações: agendar consulta, comprar um produto;
    - Tela Calendário da lista de mês, dia e hora para selecionar um serviço.
- Se o usuário for um admin:
    - Tela com ações: registrar administradores / clientes / produtos / serviços / etc) em um menu;
    - Tela para gerenciamento de serviços, onde você pode associar serviços a horários, liberar slots reservados, etc;
    - Tela Gerenciamento de estoque do produto (adicionar, atualizar, excluir, consultar)

2. **Project description**:

Para suprir todas as funcionalidades citadas, foram implementados um servidor com o Node.js e um banco de dados com CouchDB. Todas as páginas estáticas estão presentes no arquivo "index.html" (contido nesta pasta), distribuídas em tags section com a class "content". A mudança entre essas páginas é feita por meio da modificação da propriedade CSS "display" dessas sections.

Há também sections e outras tags que se encontram vazias no html. Nesse caso, são páginas carregadas dinamicamente a partir de solicitação do usuário por meio de métodos do protocolo HTTP enviados pelo cliente, como GET e POST, ou DELETE (somente pelo admin). Esses métodos são ativados por botões, forms ou mesmo pelo carregamento da página (no caso da renderização dos produtos mais vendidos e das listas de produtos, serviços, usuários e vendas da

página de administrador). No lado do cliente, foi utilizado jQuery em alguns métodos.

3. **Comments about the code**:

Todos comentários necessários já estão no código.

4. **Test plan**:

Para testar as funcionalidades do website e verificar se foram atingidos os requisitos exigidos na proposta inicial do projeto, alguns testes foram realizados. Para tanto, visou-se realizar toda e qualquer ação que um cliente leigo qualquer (com ou sem cadastro) realizaria ao navegar numa loja virtual, procurando por bugs, inconsistências e maneiras de melhorar a experiência de usuário. O análogo seria feito para o administrador. O objetivo é determinar se a maneira como cada funcionalidade foi implementada (e a proposta em si) faz sentido.

Tendo em mente que o projeto deve manter sua consistência em diferentes ambientes possíveis, a aplicação precisa ser testada em quantas configurações o grupo tiver acesso viável, sendo cabível a busca por discrepâncias. Ou seja, analisar portabilidade e responsividade. Assim, o site foi testado em vários browsers, como Google Chrome, Vivaldi, Microsoft Edge, Mozilla Firefox, todas no Windows 10. Posteriormente foram feitos testes no Safari para iOS e no Google Chrome e Vivaldi Beta para Android.

5. **Test results**:

As requisições feitas nos testes foram todas devidamente consolidadas, tanto com permissões de cliente quanto com permissões de admin. Dos testes realizados, qualquer usuário é capaz de consultar o catálogo de produtos e serviços (cliente ou admin, tendo feito ou não o login) contendo descrições, imagens e preços; qualquer

cliente (depois de cadastrado e ter feito login) é capaz de ver os produtos disponíveis ou reservar data e horário para serviço; qualquer admin é capaz de registrar ou apagar clientes, produtos e serviços, e gerenciar e visualizar os serviços e o estoque. No caso do cliente, o acesso aos produtos pode ser feito pelo menu horizontal ou então pela barra de busca no canto superior direito da página, ou mesmo na página principal, em Itens em Destaque. No entanto, também houveram falhas no teste, como adicionar produtos ao carrinho de compras, ou seja, apesar de o sistema de compras estar implementado, não está funcional pois não é possível adicionar produtos ao carrinho.

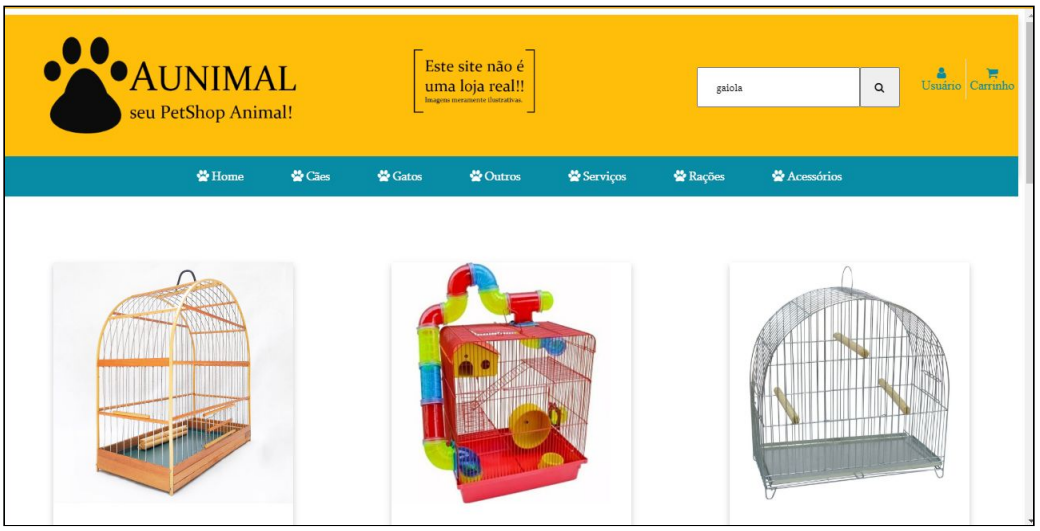

Imagem 1: Teste de busca no site

Imagem 2: Teste de agendamento de serviços

No desktop, a página funciona exatamente igual nos diferentes navegadores utilizados. A página é responsiva a mudanças na largura da janela, inclusive transformando menus horizontais em verticais, e “achatando” as imagens cuja largura excede a do corpo do documento, ou seja, sem overflow. A disposição horizontal dos divs também se torna vertical.

Para navegadores mobile, as funcionalidades se mantiveram e as imagens são levemente deformadas, além da localização de alguns elementos excederem o tamanho de seus containers.

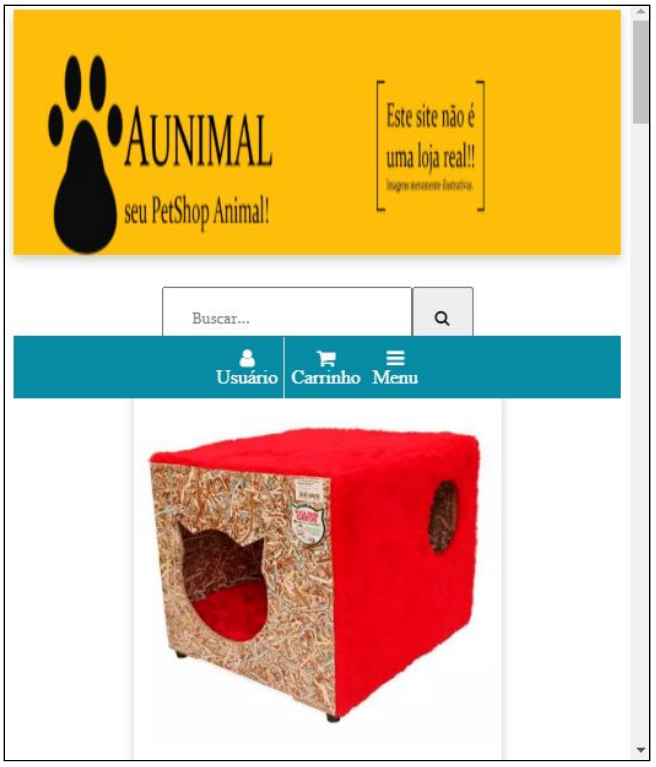

Imagem 3: Teste de responsividade para versão mobile

6. **Build Procedures**:

O site está hospedado em http://aunimal.pet/ , no qual já está devidamente configurado e pronto para o uso. Além disso, é possível também acessá-lo diretamente de sua máquina, seguindo as seguintes instruções:

- Configuração:

Antes de tudo, é necessário adequar sua máquina para rodar o servidor:

- ***Node***: Primeiro, instale o nodeJS e o npm na sua máquina conforme as instruções fornecidas pelo desenvolvedor. Feito isso, navegue até essa pasta por meio do seu terminal ou console e utilize o comando "npm install", que instalará todos os packages necessários.

- ***CouchDB:*** Instale o CouchDB na sua máquina, de acordo com a documentação provida pelos próprios desenvolvedores. A instalação inclui o Fauxton (GUI WEB para gerenciamento do Banco). Ao fim da instalação, cadastre o login que será utilizado. Para usar o Fauxton acesse o Banco de Dados localmente em http://127.0.0.1:5984/_utils/ e realize o login.

- Crie um Banco de Dados chamado petshop.

- O trabalho acompanhará um arquivo para popular e configurar o banco de dados, que poderá ser automaticamente populado usando o script couchbackup disponivel em https://github.com/cloudant/couchbackup (apenas linux).

- Execução:

Tendo a database criada e os pacotes do node instalados, basta navegar até o diretório do programa e utilizar o comando "node index.js" (ou "nodemon index.js", dependendo da sua preferência) para iniciar o servidor.

O site estará acessível em 127.0.0.1:3000 (por default, isso pode ser alterado).

7. **Problems**:

Os únicos problemas foram pela falta de tempo de resolver alguns bugs. Por exemplo, não é possível adicionar um produto ao carrinho de compras, e,

consequentemente, realizar a compra. Apesar disso, o sistema de compras está implementado, e pode ser testado se for feita a implementação do carrinho.

8. **Comments**:

No comments.