Tecnologias utilizadas:

• HTML: Usado como linguagem de marcação para a criação web;

• CSS: Foi utilizado para estilização e responsividade do website, com pacotes de fontes especiais do google e interface simples para melhor entendimento;

• JavaScript: Essa linguagem de script foi utilizada para processamento e armazenamento dos dados de forma dinâmica para salvar produtos favoritos nas pastas e criar o fluxo do site;

• Visual Studio Code: Usado como nosso principal editor de código (IDE) com diversas ferramentas que facilitaram o desenvolvimento, como Live Server e o Thunder Client;

• Git: Sistema criado para controlar as versões durante o desenvolvimento do software;

• GitHub: Utilizado para controle do repositório e de toda a documentação a respeito do projeto.

• Ferramenta do desenvolvedor do navegador: Essa ferramenta foi muito utilizada para realizar os testes do site durante o desenvolvimento, para checar o html, console e network.

• Miro: Plataforma usada para o processo de design thinking.

• Trello: Plataforma que foi usada para o Kanban.

• Canva: Plataforma usada para construção dos diagramas e de diversas imagens para o site.

• Powtoon: Plataforma para criação do vídeo.

• WireframeCC: Plataforma para desenvolvimento do Wireframe e do Userflow.

Projeto da Solução:

A solução do site Old Fashioned foi o desenvolvimento de uma plataforma que permite o usuário criar uma conta e encontrar produtos pré-selecionados que condizem com o estilo e o conforto para os idosos, de forma simples e intuitiva, por meio dos filtros de eventos, gênero e cor e pela pesquisa por palavras-chave. Cada produto possui uma descrição, que auxilia na escolha, e também apresenta guias de medidas específicas, já que um grande problema da moda para idosos é encontrar roupas com caimentos corretos de acordo com cada fisionomia mudada pela idade. A partir dessas peças, o usuário pode salvar suas favoritas na pasta geral ou criar novas pastas, em que, com elas, pode se organizar melhor por estilo, por eventos e o que deseja separar. O user também pode aprender sobre como as cores influenciam em sua aparência na seção de cores do site.

Plano de testes:

Os planos de testes foram realizados com uma listagem inicial dos possíveis erros e uma posterior testagem de todas as funcionalidades até a correção de todos os problemas encontrados. Foi também pedido que pessoas fora do grupo de desenvolvimentos testassem o programa, para que a atividade fosse feita sem possíveis parcialidades.

Registros de testes:

Alguns dos problemas encontrados foram o salvamento incorreto dos ids nos favoritos, em que nos cards eram salvos em inteiro e dentro do veja mais eram salvos como strings, mas isso foi consertado, o carrinho de produtos que foi retirado por não representar o rumo que queríamos tomar com o site e erros ao salvar o usuário no localStorage que também foi corrigido.

**Plano de Testes**

Os planos de testes foram detalhados no tópico acima. Eles foram feitos, inicialmente, através de um planejamento de possíveis erros, em seguida, uma pesquisa de testes necessários, logo após, a implementação, junto com testes paralelos, e, por fim, uma bateria de testes realizada por 5 pessoas diferentes afim de encontrar possíveis erros ou falhas junto com suas posteriores correções.

**Registros de Testes**

Na página de Login foram encontrados muitas formas diferentes de falha de entrada, em sua maioria foram feitos tratamentos que prevenissem a inserção de dados de diferentes tipos, assim também prevenindo diversas falhas de segurança. Alguns tipos de erros não são necessários fazer tratamento específico, devido às suas essências, como por exemplo, alguém tentando recuperar uma conta que nem sequer foi criada, sempre será apresentado um erro de conta não encontrada.

Dentro do site existem outros erros possíveis, como a inserção de dados inválidos na criação das notas no calendário. Estes erros são prevenidos com a opção do usuário entrar com o dado através de um pequeno calendário, selecionando manualmente o dia, ao invés de digitá-lo, o mesmo para a hora sendo selecionada em um relógio, lembra-se que cada uma dessas funcionalidades é herdada do browser que estiver sendo utilizado tornando impossível tratá-las caso o browser não possua esta opção. Para evitar a inserção do mês no lugar do dia e vice versa é feito uma sugestão de dados em forma de *placeholder* e controle de tipagem com *pattern*, mas ainda não é possivel identificar possíveis falhas caso a pessoa não se atente ou a conversão automática de dados do browser falhe.

**Referências**