

Projeto 14. Checklist

Projeto rejeitado sem revisão

- A marcação não foi portada para JSX.
- Parte da funcionalidade não foi implementada: o levantamento de contexto ou estado não foi implementado. Os componentes pop-up não foram criados.
- Há perguntas dirigidas aos revisores no projeto.

Requisitos do projeto

Geral

- Os arquivos de projeto de infraestrutura foram criados usando Vite.
- O projeto executa sem erros.
- Seu projeto contém os elementos a seguir:
 - Os arquivos `index.css` e `main.jsx` na pasta `src`
 - Um diretório `blocks`
 - Uma pasta `images`
 - O diretório `components`, que contém os seguintes:
 - `App.jsx`
 - Diretório `footer`
 - `Footer.jsx`
 - Diretório `header`
 - `Header.jsx`
 - Diretório `main`
 - Diretório `components`

- Diretório `card`
 - `Card.jsx`
- Diretório `popup`
 - Diretório `editAvatar`
 - `EditAvatar.jsx`
 - Diretório `editProfile`
 - `EditProfile.jsx`
 - Diretório `imagePopup`
 - `ImagePopup.jsx`
 - Diretório `newCard`
 - `NewCard.jsx`
 - Diretório `removeCard`
 - `RemoveCard.jsx`
 - `Popup.jsx`
- `Main.jsx`
- Diretório `card`
 - `Card.jsx`
- O diretório `contexts`, que contém `CurrentUserContext.js`
- O arquivo `README.md`
- O arquivo `.gitignore`, que fala para o git ignorar `node_modules` e `dist`
- Folhas de estilo estão conectadas.

O projeto está em conformidade com os seguintes requisitos de estilo de código:

- camelCase está sendo utilizado para os nomes de funções e variáveis.
- Somente substantivos foram usados como nomes de variáveis.

- Os nomes das variáveis descrevem claramente o que está armazenado nelas. Se o projeto tiver diversas variáveis com dados parecidos, então essas variáveis terão nomes exclusivos, mas descritivos.
- Os nomes descritivos foram usados para as funções, que refletem o que eles são.
Os nomes das funções começam com um verbo.
- Classes JS e componentes funcionais foram nomeados usando substantivos e começam com letra maiúscula.
- Os nomes não incluem abreviações impróprias ou pouco claras.

A funcionalidade foi implementada de acordo com o roteiro

- Cartões e informações sobre o usuário foram renderizados corretamente na página.
- Todos os pop-ups abrem corretamente clicando no respectivo elemento da interface.
- Os pop-ups podem ser fechados clicando no ícone fechar.
- As janelas pop-up foram configuradas: um avatar e perfil de usuário podem ser editados, um novo cartão pode ser adicionado.
- No formulário para editar um perfil, os dados foram substituídos pelos dados do usuário atual.
- O recurso para curtir e remover curtidas foi implementado.
- O recurso para remover o cartão de um usuário foi implementado corretamente.
- Todos os valores locais foram atualizados corretamente.

React

- O objeto de contexto `CurrentUserContext` foi exportado de um arquivo separado no diretório `contexts`.
- A marcação foi portada para JSX:
 - A marcação está incluída em `()`.

- A marcação foi movida para os componentes correspondentes.

Componentes:

- Hooks não foram utilizados dentro de instruções condicionais.
- Hooks foram chamados na função principal de um componente.
- Os componentes `Main` e `Card` foram inscritos no contexto `CurrentUserContext`.
- O estado foi levantado do componente `Main` para o componente `App`.
- O contexto foi incorporado no componente `App` via `CurrentUserContext.Provider`.
- A variável de estado `currentUser` foi criada no componente raiz do `App`. A variável foi usada como um valor do provedor de contexto.
- Os componentes pop-up contêm apenas os manipuladores de eventos de envio. Outros manipuladores, como `handleUpdateUser`, foram descritos dentro do componente `App`.
- Todas as solicitações da API foram descritas dentro do componente `App`.
- A solicitação da API para informações sobre o usuário e o vetor de cartões foi feita apenas uma vez — na montagem.
- Todas as variáveis de estado do roteiro foram criadas e definidas dentro dos componentes especificados no roteiro.
- O estado inicial das variáveis de estado contém o tipo de dados correto.