Чек-лист по самопроверке. Написать весь JSX и сверстать

В этом документе описаны критерии, которым должна соответствовать работа. Перед отправкой работы на ревью, убедитесь, что она соответствует всем критериям.

Работа отклоняется от проверки

Если не соблюсти хотя бы один из критериев этого блока, ревьюеры не станут проверять работу. Придётся её доделывать и отправлять на проверку снова.

- Пул-реквест не отправлен на проверку.
- Проект не собирается или не запускается.

Работа принимается

В этом блоке собраны требования к итоговой работе, по которым вы можете выявить и самостоятельно исправить частые ошибки. В следующих спринтах мы предложим вам новые критерии для самопроверки.

Общее

- Пул-реквест создан из указанной в задании ветки в `main`.
- Инфраструктурные файлы проекта созданы через CRA с флагом `--template` со значением `typescript`.
- Сборка и запуск проекта выполняются без ошибок.
- В зависимостях установлен пакет `@ya.praktikum/react-developer-burger-ui-components`.
- В проекте есть:
  - директория `components` с компонентами и стилями: `App`, `AppHeader`, `BurgerIngredients`, `BurgerConstructor`, `Modal`, `ModalOverlay`, `OrderDetails`, `IngredientDetails`;
  - если в проекте используются сторонние шрифты или изображения, то они хранятся в папках `fonts` и `images`;
  - файл `README.md`;
  - файл `.gitignore`.
- Стили портированы как модули. Если есть общие стили, они портированы в глобальную область видимости.
- Код оформлен без ошибок:
  - имена переменных и функций написаны в camelCase;
  - имена классов и функциональных компонент — существительные с прописной буквы;
  - имя функции отражает то, что она делает.

  Для именования запрещены:
  - транслит;
  - неуместные сокращения.
- Корректно выполняются запросы к API:
  - URL-адрес домена вынесен в отдельную константу;
  - есть проверка успешности выполнения запроса;
  - цепочка обработки промисов завершается блоком `catch`.

React

- Разметка портирована в JSX:
  - разметка заключена в `( )`;
  - разметка вынесена в соответствующие ей компоненты.

- Компоненты написаны корректно:
  - хуки не используются внутри условных блоков;
  - хуки вызываются в основной функции компонента;
  - при использовании классовых компонентов эффекты описаны внутри методов жизненного цикла компонента.
- Для компонентов с пропсами описан `propTypes`. Если в качестве пропсов передается объект или массив объектов, то в `propTypes` описана структура этого объекта.
- Функциональность из брифа реализована корректно:
  - компонент `AppHeader` использует UI-компоненты из библиотеки: логотип, иконки, типографику, систему отступов;
  - компонент `BurgerIngredients` отображает список полученных с сервера ингредиентов. В `BurgerIngredients` используются UI-компоненты из библиотеки: счётчики, иконки, переключатели, типографика, система отступов;
  - компонент `BurgerConstructor` отображает список ингредиентов, добавленных в бургер, полную стоимость и кнопку офоццрмления заказа. В `BurgerConstructor` используются UI-компоненты из библиотеки: `ConstructorElement`, иконки, кнопка, типографика, система отступов;
  - компонент `IngredientDetails` при открытии содержит данные, полученные от API с описанием конкретного ингредиента и использует UI-компоненты из библиотеки: иконки, типографику;
  - компонент `OrderDetails`содержит тестовые данные и использует UI-компоненты из библиотеки: иконки, типографику.
- Реализованы модальные окна:
  - есть компонент `Modal` — шапка с заголовком и иконкой закрытия;
  - содержимое модального окна передается в компонент `Modal` как `children`;
  - есть компонент `ModalOverlay` — фоновая подложка под модальным окном;
  - модальное окно с описанием ингредиента открывается при клике по ингредиенту;
  - модальное окно с описанием заказа открывается при клике по кнопке «Оформить заказ»;
  - модальные окна закрываются при клике на крестик, на `ModalOverlay` или нажатием на клавишу "Esc";
  - логика навешивания и удаления обработчиков события нажатия клавиши "Esc" описана в компоненте `Modal`;
  - в компоненте `Modal` используется портал;
  - использованы UI-компоненты из библиотеки: иконки, типографика.
- Запрос к API за информацией об ингредиентах работает корректно и выполняется единожды при монтировании компонента `App`.

Привычка проверять проектную работу по чек-листам — как привычка составлять список покупок перед походом в магазин — дисциплинирует, экономит время и помогает не пропустить самое важное. Возможно, навык работы с чек-листами пригодится вам и в будущем, когда потребуется сверять готовый проект со списком требований в техническом задании.