

Custom-components-lib

Экосистема:

Требуется инициализировать базовое приложение (без используя Boilerplate вроде CRA), основанное на React и реализовать функционал, описанный далее. В том числе, на личном GitHub-аккаунте создать публичный репозиторий с названием этого задания (указано в заголовке документа: Innowise Lab Internship: Custom-components-lib) и возможность создавать Pull request любыми лицами.

Суть приложения: создание собственной библиотеки компонентов и публикация ее, как NPM пакет. В качестве дизайна использовать компоненты <https://mui.com/>. После публикации NPM пакета установить в любой React проект и проверить работоспособность.

Необходимо реализовать следующие компоненты:

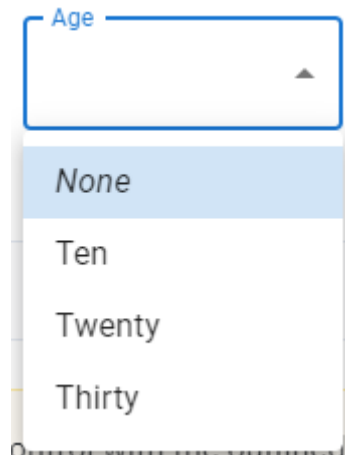
1. Button - в качестве примера использовать <https://mui.com/material-ui/react-button/> необходимо реализовать следующие пропсы:
 - variant: "text" | "contained" | "outlined" , contained as default
 - disabled
 - onClick
 - size: "small" | "medium" | "large"



2. TextField - в качестве примера использовать <https://mui.com/material-ui/react-text-field/> необходимо реализовать следующие пропсы:
 - disabled
 - error



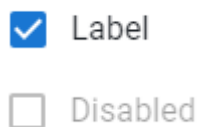
3. Select - в качестве примера использовать <https://mui.com/material-ui/react-select/>



4. Checkbox - <https://mui.com/material-ui/react-checkbox/>

Пропсы:

- checked
- onChange
- disabled
- label



5. Switch- <https://mui.com/material-ui/react-switch/>

Пропсы:

- checked
- onChange
- disabled



6. TextField - <https://mui.com/material-ui/react-text-field/>

Пропсы:

- variant: "outlined" | "filled" | "standart"
- error
- label
- disabled

7. Modal - <https://mui.com/material-ui/react-modal/>

Пропсы

- open
- onClose
- children

Все инпуты должны иметь hover и blur

Технические требования:

1. Перед разработкой надо настроить ESLint для приложения. В качестве дополнения можно настроить Prettier и связать его с ESLint. Будет дополнительным плюсом реализовать pre-commit hook, который не позволит сделать push в репозиторий, если в приложении присутствуют ошибки ESLint.
3. Настройка Webpack
2. Требуется использование TypeScript.
3. Допускается использование только CSS modules (SCSS)
4. Использование Storybook.
5. Написание unit tests используя Jest and React testing library для каждого из компонентов. Все пропсы должны быть покрыты тестами.
6. Написание документации к проекту; в README-файле проекта перед сдачей задания должна быть написана краткая документация на английском или русском, состоящая из следующих пунктов: "Task" (ссылка на этот документ), "How to run the app" (инструкции по установке модулей и запуску приложения)