Тестове завдання для Yalantis React.js School #5

Tech Stack: React + Axios/Fetch

Задача:

GET: `https://yalantis-react-school-api.yalantis.com/api/task0/users` повертає масив юзерів. У кожного юзера є наступні поля: id, firstName, lastName, dob (дата народження).

Реалізувати:

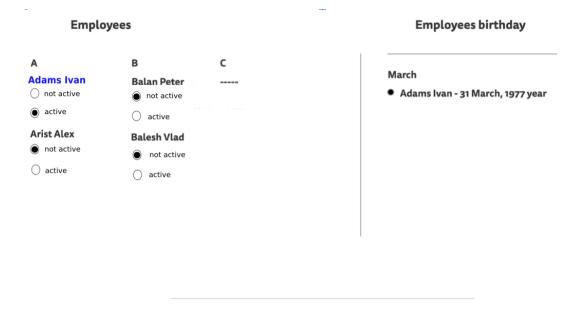
- 1. Завантажити список співробітників з сервера, при потраплянні на сторінку *employees* (сторінка /employees буде відкриватися відразу, навігації робити не потрібно).
- 2. Відобразити англійський алфавіт.
- 3. Співробітник = FirstName + LastName. Відобразити співробітників, за алфавітом, сортуємо за firstName. Якщо у літери немає співробітників, ставимо «No Employees». Біля кожного співробітника є по 2 RadioButton. По дефолту 'not active' (не активний), значення цього radio item буде = false
- 4. Якщо хочемо вибирати співробітника, натискаємо на 'active', зі значенням = true. Обраний співробітник повинен бути виділений синім кольором.
- 5. У правій частині сторінки є блок *Employees birthday*, в якому відображаються обрані співробітники, згруповані за назвою місяця їх дня народження та відсортовані по їх lastName (від A до Я).
- 6. Місяці повинні починатися з місяця виконання цього завдання і далі по черзі.
- 7. Якщо в місяці немає обраних співробітників/співробітника виводимо назву місяця з текстом «No Employees».
- 8. Якщо ви не обрали нікого зі співробітників, то в блоці *Employees birthday* пишемо *Employees List is empty*
- 9. Якщо сторінку перезавантажено обрані співробітники в правому блоку повинні бути збережені. А в лівому блоку *Employees*, у даних співробітників повинен бути активним radio item 'active' та виділено синім кольором їх Fisrst Name + Last Name.

Вимоги до коду:

- 1. Використання Prettier.
- 2. Використання функціональних компонентів та хуків, ES6 features.
- 3. Мінімальний розподіл проекту на логічні частини / папки.

Вимоги до проекту:

- 1. Розгорнути додаток, використовуючи create-react-app.
- 2. Створити публічний репозиторій на Github для проекту.
- 3. Використовувати UI ліби в проекті заборонено.
- 4. UI частина додатку залишається на ваш розсуд. Макет-шаблон, як повинно виглядати, ви знайдете нижче.



5. Задеплоїти виконане завдання за допомогою Github page

Буде плюсом:

- 1. Використання Redux (redux-toolkit, redux-thunk/saga) та/або React Context в проекті.
- 2. Використання TypeScript або опис типізація за допомогою PropTypes.
- 3. Мінімально покрити unit-тестами логічні частини в проекті.

ВАЖЛИВО! Якщо робота за функціоналом або візуально не відповідає ТЗ, описаному вище, вона не буде перевірятися.

Виконане завдання надіслати куратору школи з двома лінками:

- Link на GitHub репозиторій
- Link на GitHub Page.

Зверніть увагу! Фідбек по тестовому вам повідомлять до 02.12.21 включно.

Ми залишаємо за собою право відмовитись давати докладний фідбек по тестовому завданню.