**Wstęp**

Szkolenie to rozpoczyna cykl zadań podsumowujących z React.js. Przygotuj rozwiązania do każdego z zadań w 2 językach - JavaScript oraz TypeScript.

**Szkolenie 1 - Zadania**

Teoria do opanowania przed zadaniami:

* Tworzenie komponentów,
* Umiejscowienie komponentów w innych komponentach,
* Przekazywanie propsów do komponentów,

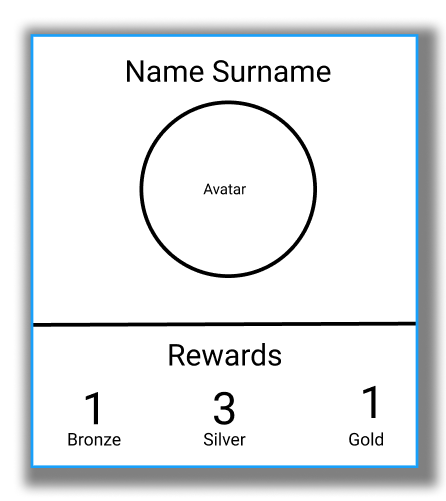
Pytania kontrolne:

* Czym jest komponent?
* Czym są „propsy”?

**Zad. 1**

**Komponent karty**

Dzięki szkoleniu wiesz już jak tworzyć komponenty. Czas podziałać z nimi w praktyce. Twoim zadaniem będzie stworzenie komponentu Card, który będzie wyświetlał imię i nazwisko użytkownika, jego awatar w górnej belce. Górna belka powinna mieć background image. W dolnej belce komponent Card będzie wyświetlał ilość zdobytych przez użytkownika medali brązowych, srebrnych i złotych. Komponent Card ma również box shadow, aby wyróżniał się na tle. Szkic poglądowy znajduje się poniżej. Nie chodzi tutaj o odwzorowanie „pixel perfect”, ale staraj się aby wygląd był zbliżony.



**Zad. 2**

**Użyj propsów**

Pamiętaj, że komponent może otrzymać swoje propsy, dzięki czemu wykorzystujemy komponenty jak klocki, które możemy modyfikować propsami. Wyobraź sobie, że współpracujesz z API wystawionym w firmie przez backendowców. Niestety nie zdążyli jeszcze zrobić API, dlatego wysłali Ci strukturę danych jaką otrzymasz w przyszłości (przy okazji ćwiczymy pracę na mockach danych) Dostosuj komponent do otrzymanych danych. Jeżeli w danych ilość medali to 0, wyświetl pozostałe medale (w przypadku gdy ilość medali brązowych to 0, srebrnych 1 i złotych 2, na karcie powinny pojawić się tylko ilość medali srebrnych i złotych).

Dane otrzymane od backendowców:

const data = [

    {

      name: "Alfred",

      surname: "Bogucki",

      avatar:

        "https://images.unsplash.com/flagged/photo-1570612861542-284f4c12e75f?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1770&q=80",

      background:

        "https://images.unsplash.com/photo-1544200175-ca6e80a7b323?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1281&q=80",

      bronze: 1,

      silver: 3,

      gold: 1,

    },

    {

      name: "Andrzej",

      surname: "Mikucki",

      avatar:

        "https://images.unsplash.com/flagged/photo-1570612861542-284f4c12e75f?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1770&q=80",

      background:

        "https://images.unsplash.com/photo-1544200175-ca6e80a7b323?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1281&q=80",

      bronze: 1,

      silver: 8,

      gold: 0,

    },

  ];

**Zad.3**

**Pozycjonowanie komponentów na stronie**

Stworzyć komponent grid, który będzie odpowiadał za rozmieszczenie elementów Card na stronie (pamiętaj o responsywności), a następnie rozszersz na podstawie mocka danych otrzymanych z backendu w poprzednim zadaniu tablicę obiektów. Przekaz dane do komponentu Card za pomocą funkcji wyższego rzędu map, a wszystko umieścić w komponencie grid.

**Podsumowanie do zadań 1, 2 i 3**

**Zadanie 1, 2 i 3 ćwiczy umiejętność tworzenia i dostosowywania komponentów do warunków zewnętrznych, renderowania warunkowego, stylowania i pozycjonowania elementów na stronie. Pamiętaj, że zgodnie z metodologią BEM komponent karty nie powinien mieć styli odpowiadających za pozycjonowanie, gdyż jest to aktualnie odpowiednik Bloku.**

**Wstęp do zadań 4 i 5**

Zadanie 4 oraz 5 ma ćwiczyć umiejętność budowania podkomponentów oraz budowanie responsywnych komponentów.

**Zad. 4**

**Responsywne menu z generycznymi ścieżkami.**

W aplikacji otrzymujesz z backendu ścieżki, do których użytkownik z następującymi uprawnieniami ma dostęp. Twoim zadaniem jest stworzenie responsywnego komponentu menu, który wygeneruje linki na podstawie otrzymanych ścieżek.

const menu = [

    {

      link: "/faktury",

      name: "faktury",

    },

    {

      link: "/dokumenty",

      name: "dokumenty",

    },

    {

      link: "/odczyty",

      name: "odczyty",

    },

    {

      link: "/kontakt",

      name: "kontakt",

    },

  ];

**Zad.5**

**Generic footer**

Wyobraź sobie, że pracujesz w korporacji mającej tysiące klientów, otrzymałeś zadnie stworzyć generyczny komponent footer. Twój footer będzie używany w każdej aplikacji tworzonej przez firmę dla klienta. Na podstawie szkiców oraz danych konfiguracyjnych aplikacji stwórz komponent Footer. Wymaganiem narzuconym przez menagerów jest to, że każdy podobiekt w obiekcie konfiguracyjnym footer może być opcjonalny (company, socials, links). Logo przekazywane w obiekcie company, może mieć maksymalne wymiary 50 x 200 px. Linki powinny być w kolumnach maksymalnie po 5 w jednej kolumnie. Ikony sociali powinny mieć wymiar 20x20px oraz być klikalnymi linkami. Linki powinny zmieniać kolor na zdarzeniu hover.

const footer = {

    company: {

      addressLine1: "Brzozowa 23",

      addressLine2: "Katowice 40-000",

      phone: "+48123456977",

      mail: "mail@example.com",

      name: "ABGF Sp. z o. o.",

      logo: "link-to-image"

    },

    socials: {

      fb: "facebook.com",

      instagram: "instagram.com",

      twitter: "twitter.com"

    },

    links: {

      help: "/help",

      pay: "/pay",

      agreement: '/agreement',

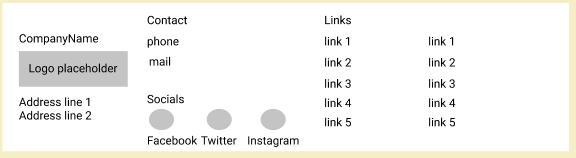
      dashboard: '/dashboard',

      reading: '/reading',

      reports: '/reports',

    }

  };



Hints: Object.entries