**Szkolenie 2 zadania**

Teoria do opanowania przed zadaniami:

* Obsługa zdarzeń,
* Hooki useState, useEffect, useRef, useMemo, useCallback, custom hook
* Context API,

Pytanie kontrolne:

* Opisz działanie hooków useState, useEffect, useRef, useMemo

**Zad 1**

Stwórz komponent wyświetlający następujące dane, które przekażesz do useState jako initialValue.

const initialData = {

    human: {

      sex: "man",

      surname: "Detic",d

    },

    accountBalance: 1000,

  };

Następnie przycisk, który będzie togglować płeć użytkownika (zdarzenie onClick handleSexChange) oraz przyciski do odjęcia 500 od stanu konta (zdarzenie onClick handleBuy) i 500 do dodania do stanu konta (zdarzenie onClick handleSell). Pamiętaj o destrukturyzacji i właściwości prevValue w setState, aby nie zgubić po drodze stanu.

**Zad 2**

Korzystając z fake rest api dostępnego pod adresem <https://jsonplaceholder.typicode.com> stwórz komponent post, który jako props przyjmie obiekt danych przyjęty z endpointa /posts wspomnianego api. Korzystając z hooka useEffect pobierz z api dane (tylko raz!) i załaduj je do hooka useState.

**Zad 3**

Stwórz komponent table, który jako props przyjmie obiekt data. W komponencie **table** stwórz stałą **columns**, która przechowa dane o nazwach kolumn. Do komponentu table przekaż data i columns.

Przykładowy obiekt zwrócony z API to:

const data = {

    "id": 1,

    "name": "Leanne Graham",

    "username": "Bret",

    "email": "Sincere@april.biz",

    "address": {

      "street": "Kulas Light",

      "suite": "Apt. 556",

      "city": "Gwenborough",

      "zipcode": "92998-3874",

      "geo": {

        "lat": "-37.3159",

        "lng": "81.1496"

      }

    },

    "phone": "1-770-736-8031 x56442",

    "website": "hildegard.org",

    "company": {

      "name": "Romaguera-Crona",

      "catchPhrase": "Multi-layered client-server neural-net",

      "bs": "harness real-time e-markets"

    }

  },

W tabeli jako kolumny wyświetl: name, surname, email, phone number oraz przycisk, który po kliknięciu powoduje rozwinięcie tabeli i pokaże dane adresowe, stronę internetową i dane firmy.

**Zad 4**

Stwórz custom hooka, który jako parametry przyjmie link do API oraz zwróci stan **isLoading** oraz otrzymane dane. Skorzystaj z fetcha.

**Zad 5**

Stwórz custom hooka **useClickOutside**, który jako parametr przyjmie wartość ref z hooka **useRef** oraz funkcję callback, który będzie służył do nasłuchiwania kliknięć na zewnętrzne elementy. Jako funkcję callback możesz przekazać funkcję zamykającą modal, do którego przypięty jest ref.

**Zad 6**

Wykorzystaj ContextAPI Reacta do przechowywania informacji o wybranym przez użytkownika THEME na stronie internetowej (darkMode, lightMode).

**Podpowiedź**: elementy na stronie będą wykorzystywać theme w następujący sposób:

const theme = useContext(ThemeContext); // konsumpcja contextu, np.

const mode=theme.state.mode // przykładowe wyciągnięcie zmiennej z contextu <div className={`${mode ? “dark”: “light”}`}>