II eliminacioni zadatak React

Za izradu zadatka koristiti ContextAPI ili Redux po izboru
Dummy JSON - https://dummyjson.com/docs/products

Get all products

```javascript
fetch('https://dummyjson.com/products')
.then(res => res.json())
.then(data => console.log(data));
```

Pri učitavanju aplikacije poslati poziv za dobijanje svih proizvoda. Dobijene podatke o proizvodima sačuvati u items varijabli/property-ju centralnog store-a (ContextAPI ili Redux).

U centralnom store-u kreirati funkcije/reducere za dodavanje, ažuriranje i brisanje proizvoda.
Funkcija za dodavanje proizvoda kao argument prima item (novi proizvod) i setuje items (sve proizvode) dodajući novi proizvod na kraju.
Funkcija za brisanje proizvoda kao argument prima id proizvoda i pomoću filter metoda uklanja proizvod sa tim id-jem iz items niza.
Funkcija za ažuriranje proizvoda kao argument prima product (ažurirani proizvod), pronalazi na osnovu id-ja taj proizvod u items nizu i mijenja ga sa ažuriranim proizvodom koji prima kao argument.

Iz centralnog store-a App komponenti proslijediti items. U slučaju da je items prazan niz prikazati Loading spinner, u suprotnom setovati brojač na 9 da prikaže prvih 9 elemenata iz items niza i dugme Loading more koji uvećava brojač za 9. U slučaju da je brojač veći od dužine items niza ne prikazivati dugme Loading more. Pomoću slice metode “izrezati” dio items niza koji treba da se prikazuje i map metodom proći kroz svaki element tako kreiranog niza. Za svaki element pozvati ProductItem komponenetu kojoj se prosleđuju dva property-ja: key (id proizovda) i product (sam proizvod).

U App komponenti koristiti rutiranje na način da se za putanju “/” izlistaju proizvodi, za putanju "/product/:id" prikažu detalji o proizvodu, za putanju "/product/edit/:id" prikaže forma za editovanje podataka o proizvodu i za putanju "/product/add” prikaže forma za dodavanje novog proizvoda.

ProductItem komponenta prikazuje detalje o proizvodu (naziv, cijenu, sliku) kao i dugmad ViewDetails, EditProduct i Delete. Klikom na dugme ViewDetails otvara se stranica pojedinačnog proizvoda sa detaljniim informacijama o samom proizvodu uz dugme Go back koji programski vraća korisnika na početnu stranu na kojoj se izlistavaju proizvodi (koristiti Link, template string, useParams, useNavigate).

Klikom na dugme obriši proizvod šalje se id proizvoda koji želite da obrišete funkciji deleteProduct. Ova funkcija zatim poziva pomoću fetch metode putanju za brisanje proizvoda koristeći template string

```javascript
fetch(“https://dummyjson.com/products/id”, {
   method: "DELETE",
 })
   .then((res) => res.json())
   .then((data) => console.log(data.id));
```

*koristiti backticks i id kao varijablu (template string)

Ova funkcija zatim poziva funkciju/reducer u centralnom store-u i prosleđuje mu id proizvoda koji treba da se obriše.

*brisanje nije trajno i na refresh će biti prikazan obrisani proizvod

Klikom na dugme EditProduct prikazuje se forma sa podacima o tom proizvodu. Korisnik ima opciju da izmijeni svojstva (cijenu, naziv, opis) i klikom na Edit product poziva fetch metodom putanju

JavaScript

```
fetch('https://dummyjson.com/products/id', {
  method: 'PUT',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    title: 'iPhone Galaxy +1'
  })
})
.then(res => res.json())
.then((data)  => console.log(data));
```

*koristiti backticks i id kao varijablu (template string)
Ova funkcija zatim poziva funkciju/reducer u centralnom store-u i prosleđuje mu objekat koji treba da se ažurira i pomoću navigate metode vraća korisnika na početnu stranicu.

Na početnoj stranici je potrebno prikazati i dugme Add Product koji bi učitavao komponentu za dodavanje proizvoda. Klikom na dugme pozvati funkciju koja šalje zahtjev putanji

JavaScript

```
   fetch("https://dummyjson.com/products/add", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({
        title: “naziv”,
              /* other product data */
      }),
    })
      .then((res) => res.json())
      .then((data) => console.log(data));
```

Proslijediti podatke iz state-a i zatim pozvati funckiju koja dodaje novi proizvod u items u centralnom store-u. Ne morate dodavati fotografiju, pa u slučaju da ProductItem ne prima product.images prop nemojte vratiti null kako se ne bi javljala greška