

Лабораторная добавление каталога

Упражнение 1: Добавление компонента `CatalogPage`

В этом упражнении нужно создать компонент `CatalogPage`, в котором будут выводиться товары каталога. Будет реализована возможность сортировки по *цене* и *рейтингу* товара, а также изменение отображения набора товаров - списком или табличным видом

1.0. Создайте файл `src\components\css\CatalogPage.css` и добавьте стили

```
.CatalogPage * {
  /* border: 1px solid #000; */
}

.CatalogPage {
  width: 80%;
  margin: auto;
}

.CatalogPageHeader {
  display: flex;
}

.CatalogPage_h2 {
  width: 90%;
  font-size: 1em;
  color: rgb(0, 87, 250);
}

.CatalogPage_sort_item {
  min-width: 120px;
  height: 20px;
  text-align: center;
  line-height: 1.8em;
  font-size: 0.7em;
}

.Catalog_items {
  display: flex;
  width: 100%;
  justify-content: space-between;
  flex-wrap: wrap;
}

.CatalogProduct {
  width: 30%;
  height: 200px;
  margin: 5px 0;
  border: 1px solid rgb(233, 233, 233);
  border-radius: 3px;
}
```

```
.CatalogProduct h3 {
  font-size: 0.8em;
}

.CatalogProduct_description {
  font-size: 0.8em;
}

.CatalogProduct_addtobasket {
  text-decoration: none;
  display: inline-block;
  font-size: 0.7em;
  margin-left: 10px;
}

.CatalogProduct_image {
  width: 100%;
  height: 100px;
}

.CatalogProduct h3,
.CatalogProduct_price,
.CatalogProduct_description {
  padding: 0 10px;
}

.CatalogPage_link {
  color: rgb(1, 105, 209);
  text-decoration: none;
}

.Catalog_items__list * {
  /* border: 1px solid #f00; */
}

.Catalog_items__list {
  flex-direction: column;
}

.Catalog_items__list .CatalogProduct {
  height: 100px;
  display: flex;
  width: 100%;
}

.Catalog_items__list .CatalogProduct_image {
  width: 200px;
}

.Catalog_items__list .CatalogProduct_addtobasket {
  height: 26px;
  width: 90px;
  font-size: 0.5em;
}

.CatalogProduct_star {
  color: rgb(253, 186, 2);
}
```

```
}  
.Catalog_items__list .CatalgProduct_star {  
  width: 100px;  
  font-size: 0.8em;  
}  
.Catalog_items__list .CatalogProduct h3 {  
  width: 200px;  
}  
.Catalog_items__list .CatalgProduct_price {  
  width: 50px;  
  font-size: 0.8em;  
}  
.Catalog_items__list .CatalgProduct_description {  
  width: 100px;  
}
```

1.1. Создайте новый файл **src\components\CatalogPage.js** 1.2. Импортируйте в файле **src\components\CatalogPage.js** **React** и стили

```
import React from "react";  
import "../css/CatalogPage.css";
```

1.3. Добавьте товары, которые будут отображаться в каталоге. Примечание: мы делаем это для краткости и удобства, в реальном решении товары запрашиваются с сервера

```
const items = [  
  {  
    uid: "86ed58db-082d-45ab-aa81-5218059349cb",  
    title: "Товар1",  
    description: "описание товара 1",  
    price: 1200,  
    qty: 1,  
    rating: 2,  
  },  
  {  
    uid: "05542e59-7a90-4e80-bf9d-78967f272049",  
    title: "Товар2",  
    description: "описание товара 2",  
    price: 800,  
    qty: 1,  
    rating: 1,  
  },  
  {  
    uid: "7793e4f0-fe86-47cc-98f6-e01b6beeb3af",  
    title: "Товар3",  
    description: "описание товара 3",  
    price: 250,  
    qty: 2,  
    rating: 2,  
  },  
]
```

```
  },  
  {  
    uid: "cdd8a823-a96e-4d91-90bf-fa4f96aa1bb1",  
    title: "Товар4",  
    description: "описание товара 4",  
    price: 3200,  
    qty: 1,  
    rating: 5,  
  },  
  {  
    uid: "26c4c669-7a50-4b8a-85dd-8d038f0ee30a",  
    title: "Товар5",  
    description: "описание товара 5",  
    price: 450,  
    qty: 1,  
    rating: 3,  
  },  
  {  
    uid: "ab8c9640-4033-4ac0-8d46-98af19b76245",  
    title: "Товар6",  
    description: "описание товара 6",  
    price: 5000,  
    qty: 2,  
    rating: 4,  
  },  
];
```

1.4. Опишите состояния для сортировки и отображения товаров

```
const [sort, setSort] = React.useState({ field: null });  
const [view, setView] = React.useState("table");
```

1.5. Опишите вспомогательные функции для работы с состоянием

```
const useViewTable = () => setView("table");  
const useViewList = () => setView("list");  
  
const usePriceDesc = () => setSort({ field: "price", direction: "DESC" });  
const usePriceAsc = () => setSort({ field: "price", direction: "ASC" });  
const useRatingDesc = () => setSort({ field: "rating", direction: "DESC" });  
const useRatingAsc = () => setSort({ field: "rating", direction: "ASC" });  
  
const changeSortPrice = () => {  
  const field = "price";  
  const direction = sort.direction === "ASC" ? "DESC" : "ASC";  
  setSort({ field, direction });  
};  
  
const changeSortRating = () => {
```

```
const field = "rating";
const direction = sort.direction == "ASC" ? "DESC" : "ASC";
setSort({ field, direction });
};
```

1.6. Опишите в `return` UI, который будет отображать товары в каталоге

```
return (
  <div className='CatalogPage'>

    <div className='CatalogPageHeader'>
      <h2 className='CatalogPage_h2'>Каталог</h2>
      <span className='CatalogPage_sort_item'>
        <a href='#' className='CatalogPage_link' onClick={changeSortPrice}>
          По цене
          {sort.field == 'price' && sort.direction == 'ASC' ? <span>↑</span> :
null}
          {sort.field == 'price' && sort.direction == 'DESC' ? <span>↓</span> :
null}
        </a>
      </span>
      <span className='CatalogPage_sort_item'>
        <a href='#' className='CatalogPage_link' onClick={changeSortRating}>
          По популярности
          {sort.field == 'rating' && sort.direction == 'ASC' ? <span>↑</span> :
null}
          {sort.field == 'rating' && sort.direction == 'DESC' ? <span>↓</span> :
null}
        </a>
      </span>
      <span className='CatalogPage_sort_item'>
        Вид: <a href='#' className='CatalogPage_link' onClick=
{useViewTable}>Таблица</a> | <a href='#' className='CatalogPage_link' onClick=
{useViewList}>Список</a>
      </span>
    </div>

    <div className={'Catalog_items Catalog_items__' + view}>
      {items.sort( (a,b) => {

        if(sort.direction == 'ASC') return a[sort.field] - b[sort.field]
        if(sort.direction == 'DESC') return b[sort.field] - a[sort.field]

      } ).map(item => (
        <div className='CatalogProduct'>
          <img src='/catalog_product.png' className='CatalogProduct_image' />

          <h3>{item.title.slice(0, 10)}</h3>

          <div className='CatalogProduct_price'>
            {item.price}₽
          </div>
```

```

        <div className='CatalgProduct_description'>
            {item.description.slice(0, 20)}
        </div>

        <div className='CatalgProduct_star' dangerouslySetInnerHTML={{__html:
'&starf;'.repeat(item.rating)}} />

        <a href='#' className='btn CatalgProduct_addtobasket' onClick={ev => {
ev.preventDefault(); alert('В корзину') }}>
            В корзину
        </a>
    </div>
    )}
</div>
</div>
);

```

1.7. Добавьте в `src\App.js` подключение компонента

```
import CatalogPage from "../components/CatalogPage.js";
```

1.8. Укажите его в `return`

```

return (
    <div className="App">
        <header className="App-header">
            <h1>Каталог</h1>

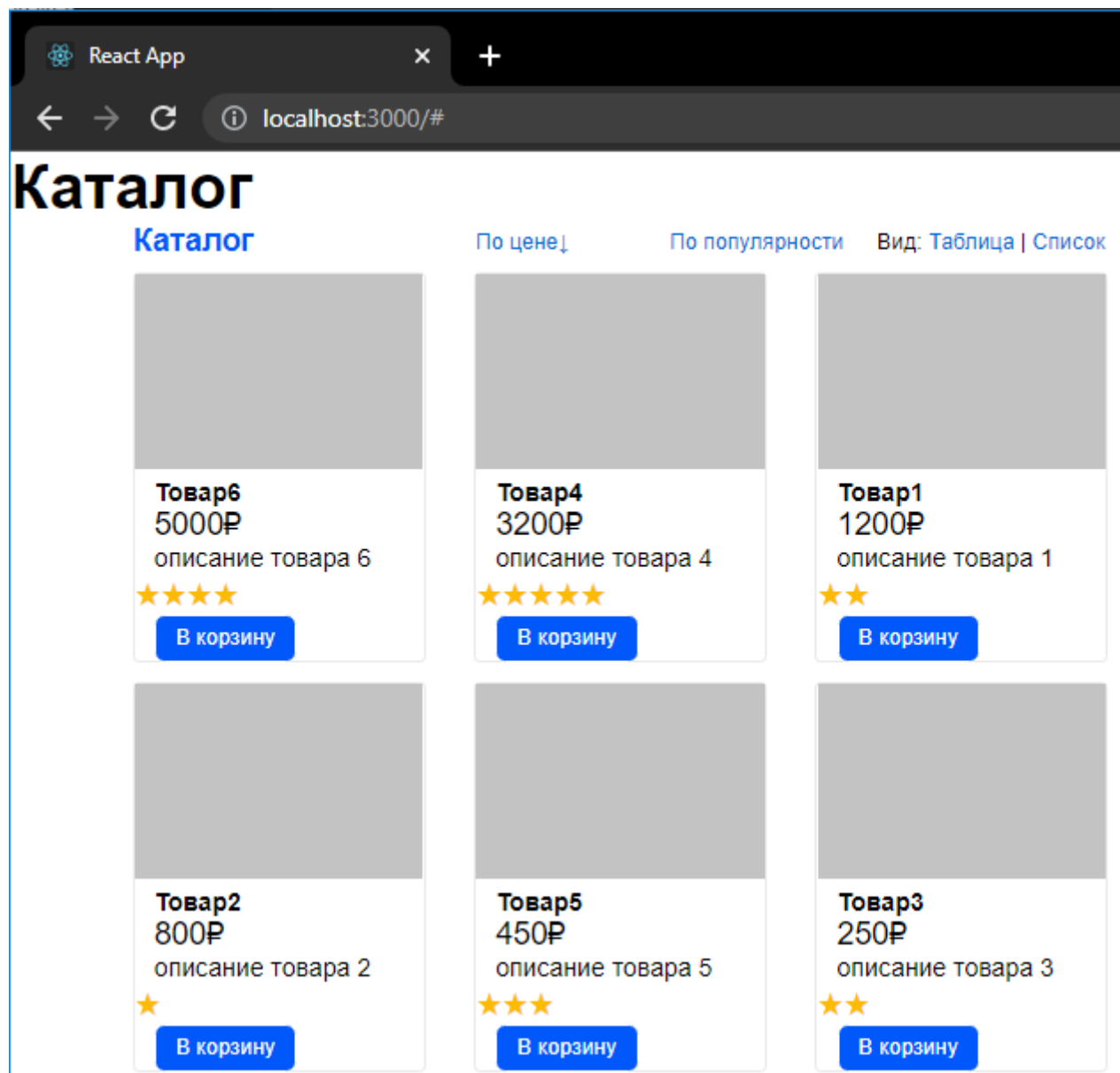
            <CatalogPage />

            <h1>Корзина</h1>

            {basketPlace}
        </header>
    </div>
);

```

1.9. Внешний вид должен быть похож на



1.10 Убедитесь, что компонент работает как запланировано. В случае возникновения ошибок - исправьте их