

Frontend Assignment Overview

Let's begin with the navigation bar:

Navigation:

Create a new component for the navigation bar. Let's call it `NavigationBar`.

Jsx

Import React from 'react';

Import { Link } from 'react-router-dom'; // Assuming you're using React Router

```
Const NavigationBar = ({ navbar }) => {  
  Return (  
    <nav>  
      <ul>  
        {navbar.map(item => (  
          <li key={item.id}>  
            <Link to={`/${item.id}`}>{item.name}</Link>  
            {item.child && (  
              <ul>  
                {item.child.map(childItem => (  
                  <li key={childItem.id}>  
                    <Link to={`/${childItem.id}`}>{childItem.name}</Link>  
                  </li>  
                )}  
              </ul>  
            )}  
          )}  
        </ul>  
      )}  
    )  
  )  
}
```

```
        </li>
      )}
    </ul>
  </nav>
);
};
```

Export default NavBar;

In your main App component, use the NavBar component and pass in the navbar array.

Jsx

Import React from 'react';

Import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';

Import NavBar from './NavBar';

```
Const App = () => {
```

```
  Const navbar = [
```

```
    { name: 'Home', id: 'home' },
```

```
    // ... other menu items
```

```
  ];
```

```
  Return (
```

```
    <Router>
```

```
      <NavBar navbar={navbar} />
```

```
      <Switch>
```

```
        { /* Define your routes here */ }
```

```
      </Switch>
```

```
    </Router>
```

```
  );
```

```
};
```

Export default App;

This example covers only a portion of your assignment.

API Integration – Product Listing:

Install Axios (if not already installed) using npm install axios.

Create a component to fetch and display product data. Let's call it ProductList.

Jsx

```
Import React, { useEffect, useState } from 'react';
```

```
Import axios from 'axios';
```

```
Const ProductList = () => {
```

```
  Const [products, setProducts] = useState([]);
```

```
  useEffect(() => {
```

```
    axios.get('https://fakestoreapi.com/products')
```

```
      .then(response => setProducts(response.data))
```

```
      .catch(error => console.error(error));
```

```
  }, []);
```

```
  Return (
```

```
    <div>
```

```
      <h2>Product List</h2>
```

```
      <ul>
```

```
    {products.map(product => (  
      <li key={product.id}>{product.title}</li>  
    ))}  
  </ul>  
</div>  
);  
};
```

Export default ProductList;

Banner:

Create a component for the banner. Let's call it Banner.

Jsx

Import React from 'react';

```
Const Banner = () => {  
  Return (  
    <div className="banner">  
      Fresh 2022 look  
    </div>  
  );  
};
```

Export default Banner;

Style the banner using CSS. You can include the CSS in your component or use a separate CSS file.

Css

```
.banner {  
  Background-color: #f0f0f0;
```

```
Text-align: center;
Padding: 20px;
Font-size: 24px;
}
```

Contact Form Validation:

Create a component for the contact form. Let's call it ContactForm.

Jsx

```
Import React, { useState } from 'react';
```

```
Const ContactForm = () => {
```

```
  Const [formData, setFormData] = useState({
```

```
    Name: "",
```

```
    Email: "",
```

```
    Message: ""
```

```
  });
```

```
  Const handleSubmit = event => {
```

```
    Event.preventDefault();
```

```
    // Perform validation and submission logic here
```

```
  };
```

```
  Return (
```

```
    <form onSubmit={handleSubmit}>
```

```
      <input
```

```
        Type="text"
```

```
        Placeholder="Name"
```

```
        Value={formData.name}
```

```
        onChange={e => setFormData({ ...formData, name: e.target.value })}
```

```
        required
```

```
    />
    <input
      Type="email"
      Placeholder="Email"
      Value={formData.email}
      onChange={e => setFormData({ ...formData, email: e.target.value })}
      required
    />
    <textarea
      Placeholder="Message"
      Value={formData.message}
      onChange={e => setFormData({ ...formData, message: e.target.value })}
      required
    />
    <button type="submit">Submit</button>
  </form>
);
};
```

Export default ContactForm;