Experiment: Connecting React Frontend to Express API Using Axios

Saksham Rathore(23BCS10814)

Objective:

Learn how to connect a React frontend application to a backend Express.js API using Axios to fetch data. This helps understand full stack integration and handling HTTP requests in React.

Task Description:

1. Create a simple Express.js API that returns a list of products (each with a name and price).
2. Build a React frontend application that uses Axios to fetch product data from the Express API when the component mounts.
3. Display the list of products in a clean layout showing their name and price.
4. Handle loading and error states.
5. Run both frontend and backend locally and verify correct integration.

Backend (Express.js):

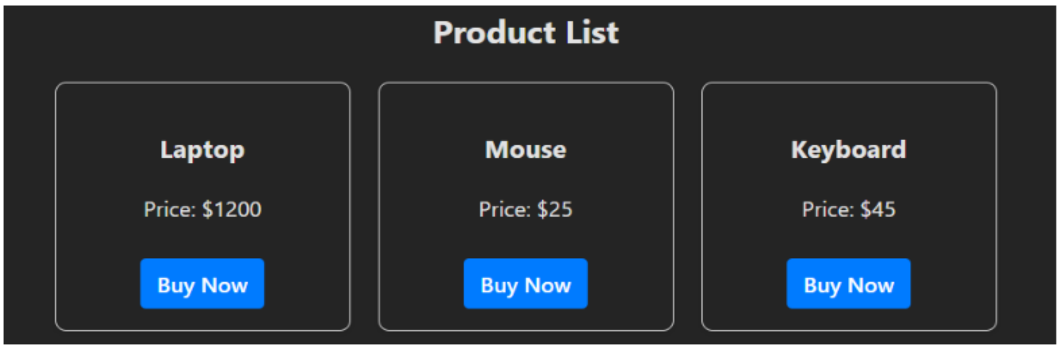
```
// server.js
const express = require('express');
const cors = require('cors');
const app = express();
app.use(cors());

const products = [
{ name: "Laptop", price: 1200 },
{ name: "Mouse", price: 25 },
{ name: "Keyboard", price: 45 }
];

app.get('/api/products', (req, res) => {
res.json(products);
});

app.listen(5000, () => console.log("Server running on port 5000"));
```

Frontend (React):

```
// ProductList.js
import React, { useEffect, useState } from "react";
import axios from "axios";

function ProductList() {
const [products, setProducts] = useState([]);
const [loading, setLoading] = useState(true);
const [error, setError] = useState(null);
useEffect(() => {
axios.get("http://localhost:5000/api/products")
.then(res => {
setProducts(res.data);
setLoading(false);
})
.catch(err => {
setError("Failed to fetch data");
setLoading(false);
});
}, []);

if (loading) return Loading...;
if (error) return {error};

return (

{products.map((product, index) => (

{product.name}
Price: ${product.price}
Buy Now

))}

);
}

export default ProductList;
```

Expected Output:

A user-friendly product list displaying product names and prices with a 'Buy Now' button, fetched dynamically from the Express.js backend.



Conclusion:

This experiment demonstrates the integration of a React frontend with an Express backend using Axios. It helps understand how to fetch and render data dynamically, a core concept in full stack web development.