**Assignment 1: Create React App Integration**

**Integrate your existing Express.js API with a Create React App.**

**Make API requests from React to your Express.js server**

let's integrate an existing Express.js API with a Create React App. I'll provide a basic example, assuming you have an Express.js API with some endpoints.

Step 1: Set Up Your Express.js API

Ensure that your Express.js API is running and has some endpoints. For example, you might have an endpoint for fetching data:

| // express-api/app.js  const express = require('express');  const app = express();  const port = 3001;  app.get('/api/data', (req, res) => {  // Replace this with your actual logic to fetch data  const data = [  { id: 1, name: 'Item 1' },  { id: 2, name: 'Item 2' },  { id: 3, name: 'Item 3' },  ];  res.json(data);  });  app.listen(port, () => {  console.log(`Express API is listening at http://localhost:${port}`);  }); |
| --- |

Step 2: Set Up Your React App

Create a new Create React App if you haven't already:

| npx create-react-app react-app-with-express-api  cd react-app-with-express-api |
| --- |

Step 3: Make API Requests from React

Install Axios (or another HTTP library) to make API requests:

npm install axios

Now, you can make API requests from your React components. For example, in src/App.js:

| // src/App.js  import React, { useState, useEffect } from 'react';  import axios from 'axios';  function App() {  const [data, setData] = useState([]);  useEffect(() => {  // Make an API request when the component mounts  axios.get('http://localhost:3001/api/data')  .then(response => {  setData(response.data);  })  .catch(error => {  console.error('Error fetching data:', error);  });  }, []); // Empty dependency array to run the effect only once  return (  <div>  <h1>Data from Express.js API</h1>  <ul>  {data.map(item => (  <li key={item.id}>{item.name}</li>  ))}  </ul>  </div>  );  }  export default App; |
| --- |

Step 4: Run Both Servers

Make sure your Express.js API is running (in one terminal):

node express-api/app.js

And start your React development server (in another terminal):

npm start

Visit http://localhost:3000 in your browser. You should see your React app fetching data from the Express.js API and displaying it.

This is a basic example to help you integrate an existing Express.js API with a Create React App. Adjust the code based on your specific API endpoints and data structures.