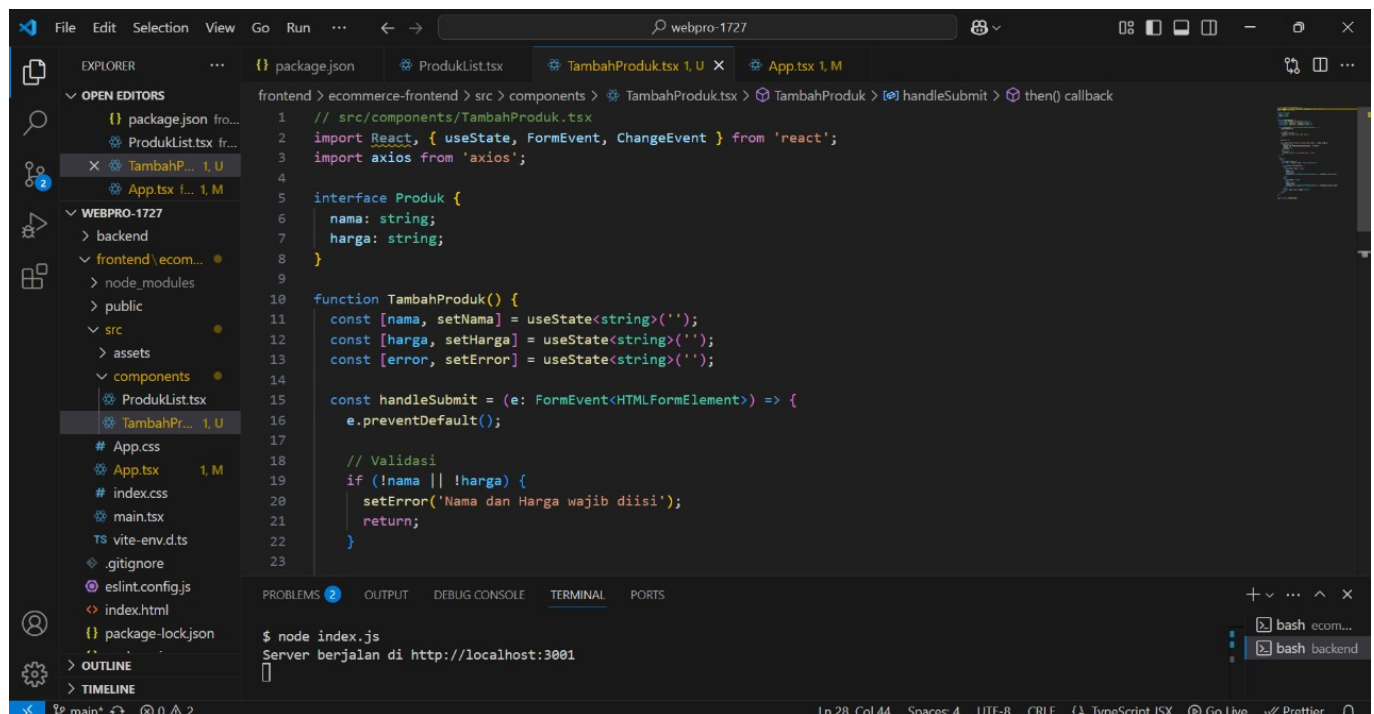


Assignment 04 - Web Programming

Ansellma Tita Pakartiwuri Putri (10231017 - BackEnd) Dahayu Azhka Daeshawnda (10231027 - FrontEnd)
<https://github.com/wounderfl/webpro-1727>

After following the module's instruction of today's class, we have successfully followed the instruction as below:

1. Component Implementation The backend developer successfully created and integrated the "TambahProduk.tsx" component into their React application. This component provides a user interface for adding new products to the system. The implementation follows React's component-based architecture, allowing for modular code organization and reusability while maintaining a clean separation of concerns within the application structure.



```
1 // src/components/TambahProduk.tsx
2 import React, { useState, FormEvent, ChangeEvent } from 'react';
3 import axios from 'axios';
4
5 interface Produk {
6   nama: string;
7   harga: string;
8 }
9
10 function TambahProduk() {
11   const [nama, setNama] = useState<string>('');
12   const [harga, setHarga] = useState<string>('');
13   const [error, setError] = useState<string>('');
14
15   const handleSubmit = (e: FormEvent<HTMLFormElement>) => {
16     e.preventDefault();
17
18     // Validasi
19     if (!nama || !harga) {
20       setError('Nama dan Harga wajib diisi');
21       return;
22     }
23
24     // ... (axios call and state updates)
25   };
26
27   return (
28     <div>
29       <input type="text" value={nama} onChange={setNama} />
30       <input type="text" value={harga} onChange={setHarga} />
31       <button onClick={handleSubmit}>Tambah Produk</button>
32       {error & <div>{error}</div>}
33     </div>
34   );
35 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
$ node index.js
Server berjalan di http://localhost:3001
```

Successfully adds new product

Aplikasi E-Commerce Sederhana

Tambah Produk

Nama Produk:

Harga:

Daftar Produk (From Database)

- Produk A - Rp10000
- Produk B - Rp20000
- Azhka - Rp0



Aplikasi E-Commerce Sederhana

Tambah Produk

Nama Produk:

Harga:

Daftar Produk (From Database)

- Produk A - Rp10000
- Produk B - Rp20000
- Azhka - Rp0
- Ansel - Rp0



2. Form Validation Implementation The developer implemented simple yet effective form validation within the product addition workflow. This validation ensures data integrity by preventing submissions with empty fields, which could otherwise lead to incomplete or corrupted database entries. The system properly displays error messages to guide users when input requirements are not met, enhancing the overall user experience and application reliability.

Error when adding product without name

Aplikasi E-Commerce Sederhana

Tambah Produk


Nama dan Harga wajib diisi

Nama Produk:

Harga:

Daftar Produk (From Database)

- Produk A - Rp10000
- Produk B - Rp20000
- Azhka - Rp0
- Ansel - Rp0



Error when adding product without price

Aplikasi E-Commerce Sederhana

Tambah Produk


Nama dan Harga wajib diisi

Nama Produk:

Harga:

Daftar Produk (From Database)

- Produk A - Rp10000
- Produk B - Rp20000
- Azhka - Rp0
- Ansel - Rp0



Error when adding product without name and price

Aplikasi E-Commerce Sederhana

Tambah Produk


Nama dan Harga wajib diisi

Nama Produk:


Harga:

Daftar Produk (From Database)

- Produk A - Rp10000
- Produk B - Rp20000
- Azhka - Rp0
- Ansel - Rp0



3. Database Integration The developer established successful database connectivity, enabling the application to persistently store newly added product data. The submitted form data was correctly processed, transmitted to the backend, and stored in the database. This confirms the end-to-end functionality of the application's data flow, from user input through the React frontend to storage in the backend database system.

Pretty-print 

```
[{"id":1,"nama":"Produk A","harga":10000}, {"id":2,"nama":"Produk B","harga":20000}, {"id":3,"nama":"Azhka","harga":0}, {"id":4,"nama":"Ansel","harga":0}]
```

