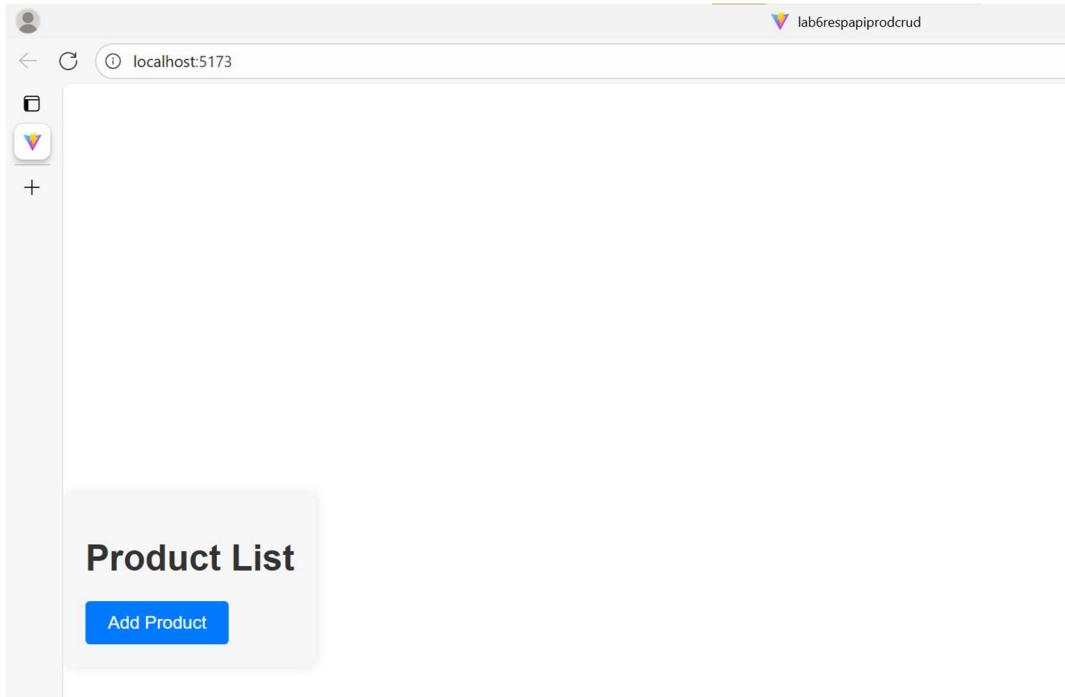


Lab6: React REST API output:

## Product CRUD RESULTS



## Add a new product:

The screenshot shows a browser window with developer tools open. The main content area displays a modal titled "Add New Product" with fields for Name, Description, Price, and Stock. The "Name" field contains "laptop for Deepika", "Description" contains "nvidia AI laptop recent versi", "Price" is 1200, and "Stock" is 100. Below the fields are "Add Product" and "Cancel" buttons. The developer tools sidebar shows the file structure under "Workspace" and the code for "ProductForm.jsx". The code includes a fetch call to "http://localhost:5000/products". The "Sources" tab is active, showing the code with line numbers. The "Console" tab on the right shows an error message: "Failed to load :5000/products:1 net::ERR\_CONNECTION\_REFUSED".

```
e.preventDefault();
fetch('http://localhost:5000/products',{
  method:'POST',
  headers:{'Content-Type':'application/json'},
  body:JSON.stringify(newProduct)
})
.then(response=>response.json())
.then(data =>{
  console.log('Product added:',data);
  onAddProduct(data);
})
.catch(error =>console.error('Error adding product:',error));
};

return(
```

From ProductForm.jsx Coverage: n/a

Console Issues Autofill +

Default levels 4 hidden

Failed to load :5000/products:1 net::ERR\_CONNECTION\_REFUSED

Error adding product: TypeError: Failed to fetch at handleSubmit (ProductForm.jsx:23:9)

Show ignore-listed frames

[NEW] Explain Console errors by using Copilot in Edge: click to explain an error. Learn more

Error since Db.json is not run.

## Starting DB.Json

The screenshot shows a code editor window with the title bar "lab6respapiprocrud". The main area displays the code for `Product.jsx`. The code uses the `useState` hook to manage product data. It defines two functions: `handleSave` and `handleDelete`. Both functions use the `fetch` API to interact with a local server at `http://localhost:5000`. The `handleSave` function performs a `PUT` request with the product data as JSON. The `handleDelete` function performs a `DELETE` request. Both functions log a message to the console and call an `onBack` callback. Error handling is included for both functions.

```
2
3     function Product({product,onBack}){
4     const [productData,setProductData] =useState(product);
5
6     const handleSave = ()=>{
7         fetch(`http://localhost:5000/products/${productData.id}`,
8             {method:'PUT',
9              headers: {'Content-Type':'application/json'},
10             body:JSON.stringify(productData)
11         })
12         .then(response=>response.json())
13         .then(data=>{
14             console.log('Product updated:',data);
15             onBack();
16         })
17         .catch(error=>console.error('Error updating product:',error));
18     };
19
20     const handleDelete = ()=> {
21         fetch(`http://localhost:5000/products/${productData.id}`,
22             {method:'DELETE'})
23         .then(()=>{
24             console.log('Product deleted');
25             onBack();
26         })
27         .catch(error=>console.error('Error deleting product:',error));
28     };
29 }
```

Below the code editor, there is a terminal window showing a command prompt in Windows. The command entered is `json-server --watch db.json --port 5000`. The terminal also shows some environment variables and the current working directory.

```
PS C:\Deepika\Azure Cloud\React\CHA_ReactTraining\ReactLabsPractice\ingit\lab6respapiprocrud> json-server --watch db.json --port 5000
```

The screenshot shows the VS Code interface with the following details:

- Editor:** The main editor area displays the file `Product.jsx`. The code implements a `Product` component that handles saving and deleting products via a JSON server at `http://localhost:5000`.
- Terminal:** The bottom terminal window shows the command `json-server --watch db.json --port 5000` being run in a PowerShell session. The output indicates the JSON server is started on port 5000.
- Output:** A separate output panel shows the JSON server configuration:
  - Index:** `http://localhost:5000/`
  - Static files:** Serving `./public` directory if it exists
  - Endpoints:** `http://localhost:5000/products`

Select a Product List from DB.Json file

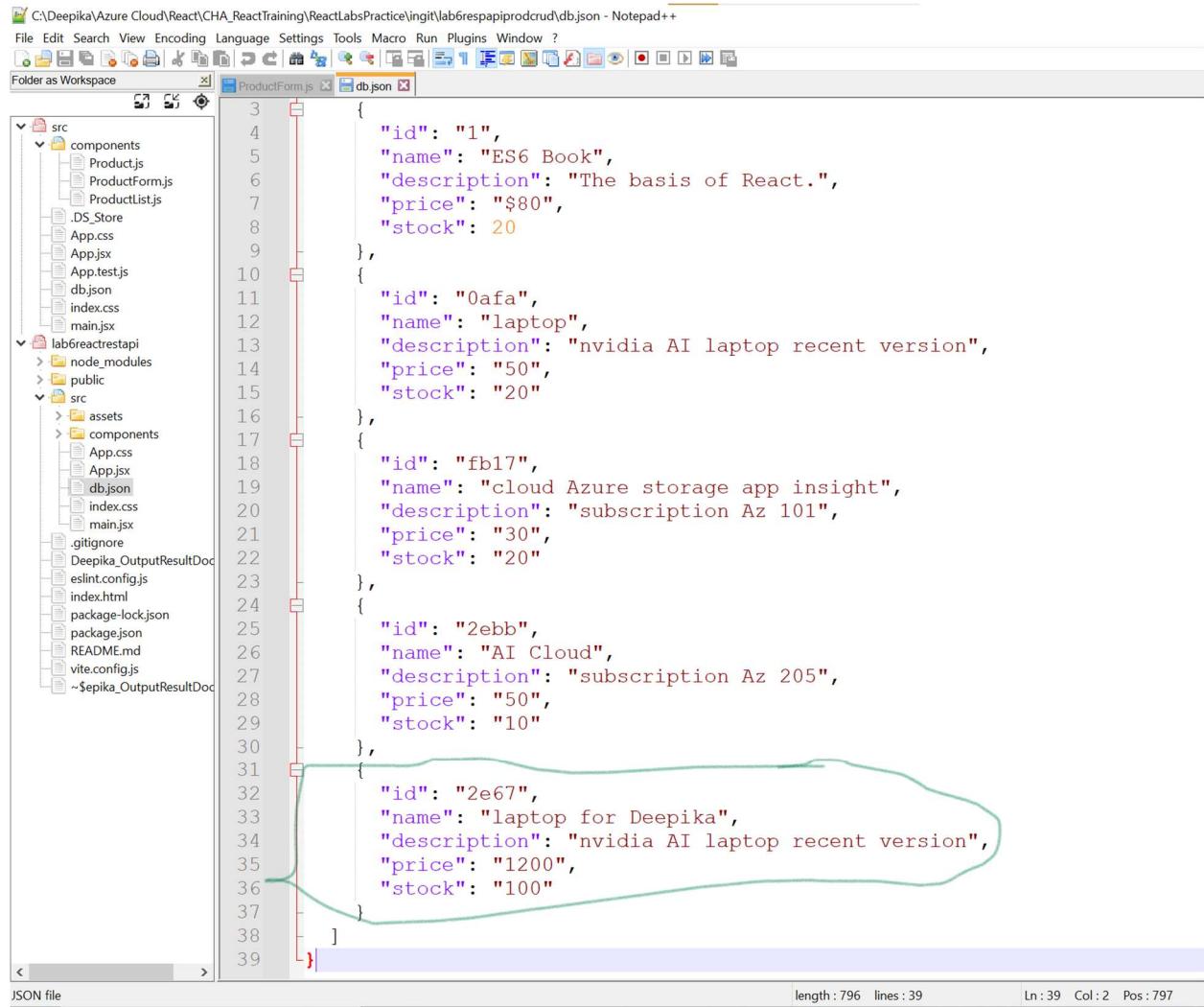
The screenshot shows a web application interface. At the top left is a user profile icon. Next to it are navigation icons: a back arrow, a refresh circle, and a search bar containing the text "localhost:5173". To the right of the search bar is a small circular icon with a question mark. On the far left of the main content area is a vertical sidebar with icons for a file folder, a plus sign, and a small logo.

**Product List**

- ES6 Book
- laptop
- cloud Azure storage app insight
- AI Cloud
- laptop for Deepika

**Add Product**

## New Product added to DB.Json file



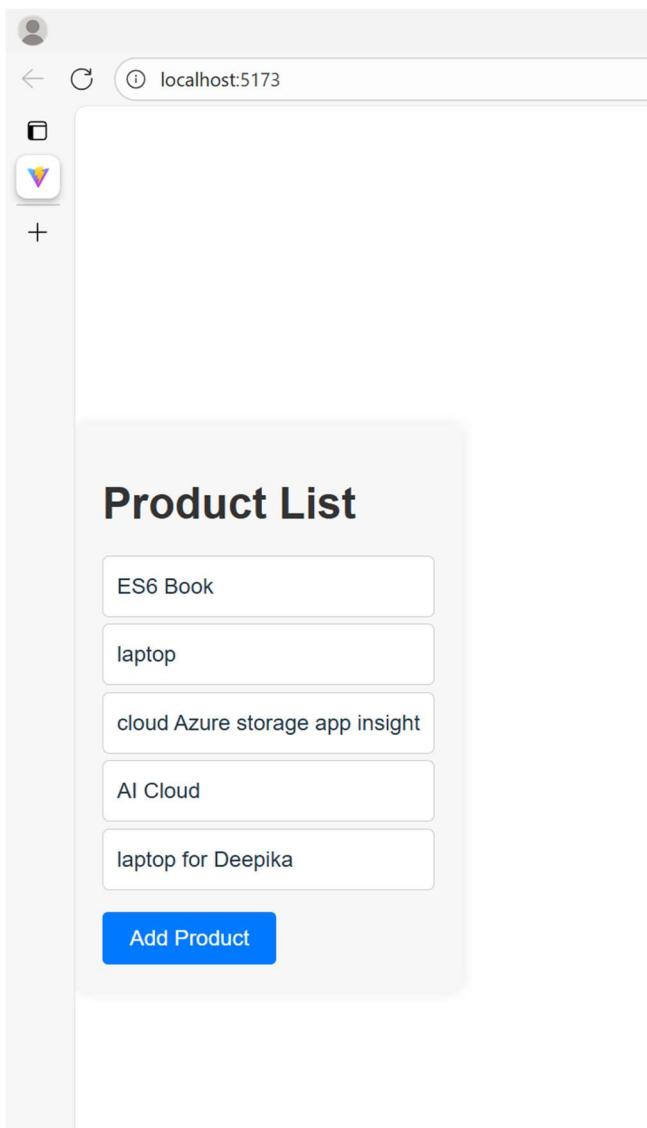
The screenshot shows the Notepad++ editor with the file "db.json" open. The file contains a JSON array of products. A new product entry has been added at the end of the array, highlighted with a green oval. The product details are as follows:

```
[{"id": "1", "name": "ES6 Book", "description": "The basis of React.", "price": "$80", "stock": "20"}, {"id": "0afa", "name": "laptop", "description": "nvidia AI laptop recent version", "price": "50", "stock": "20"}, {"id": "fb17", "name": "cloud Azure storage app insight", "description": "subscription Az 101", "price": "30", "stock": "20"}, {"id": "2ebb", "name": "AI Cloud", "description": "subscription Az 205", "price": "50", "stock": "10"}, {"id": "2e67", "name": "laptop for Deepika", "description": "nvidia AI laptop recent version", "price": "1200", "stock": "100"}]
```

Save

Delete a product:

Before delete



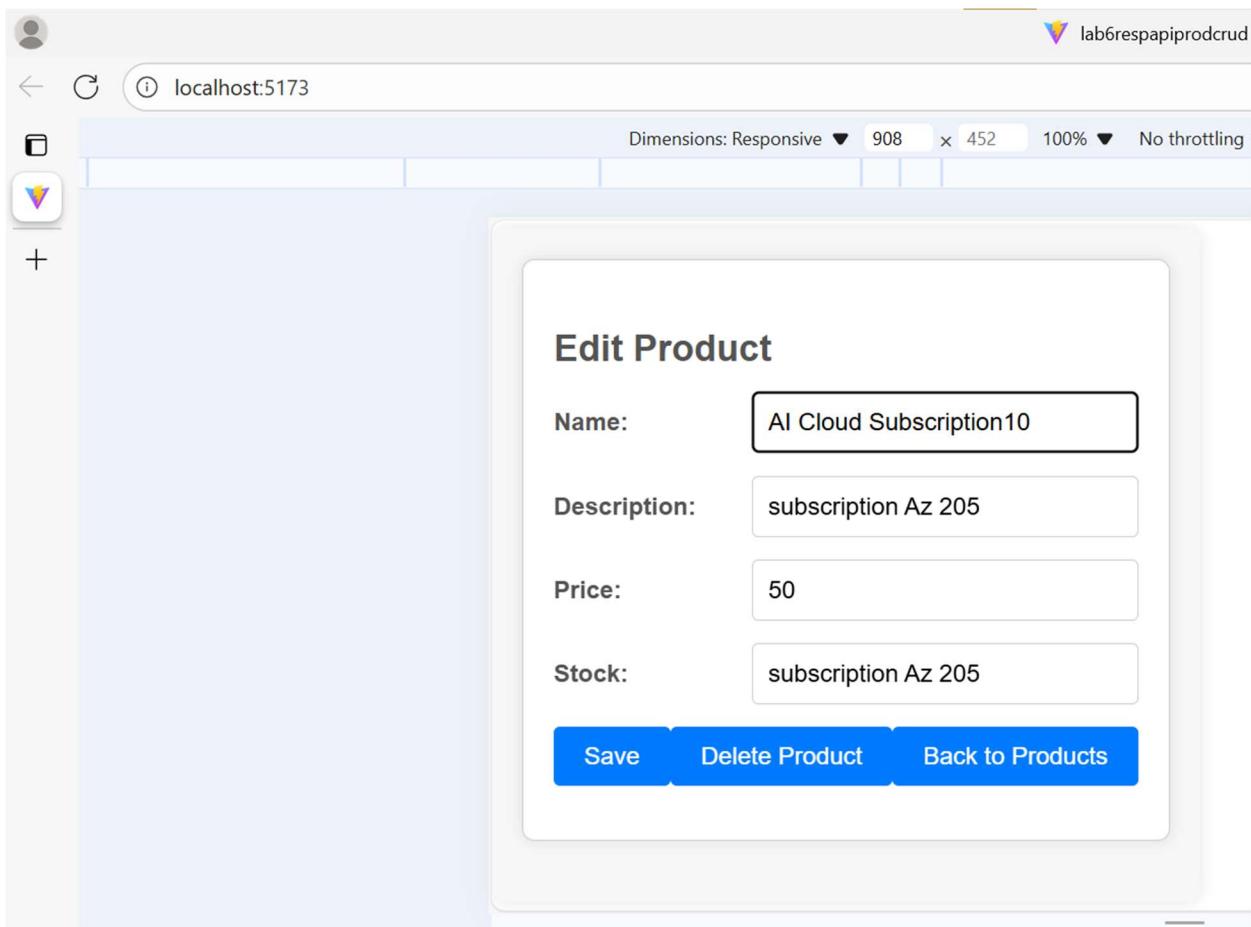
## Delete Operation

A screenshot of a web browser window titled "Edit Product". The URL is "localhost:5173". The page contains fields for Name, Description, Price, and Stock, all set to "cloud Azure storage app insight", "subscription Az 101", "30", and "subscription Az 101" respectively. At the bottom are three buttons: "Save", "Delete Product", and "Back to Products". The "Delete Product" button is highlighted with a yellow circle.

## After Delete

A screenshot of a web browser window titled "Product List". The URL is "localhost:5173". The page displays a list of products: "laptop", "AI Cloud", and "laptop for Deepika". A blue "Add Product" button is at the bottom. The product "cloud Azure storage app insight" has been removed from the list.

Saving (PUT) Operation



The screenshot shows a web application interface with the following elements:

- Header:** A top navigation bar with icons for user profile, back, forward, and address bar showing "localhost:5173".
- Toolbar:** On the left, there are icons for file operations (New, Open, Save, etc.) and a plus sign (+) for adding new items.
- Content Area:** A main area titled "Product List" containing three items in cards:
  - "laptop"
  - "AI Cloud Subscription10" (this item has a yellow swoosh graphic underneath it)
  - "laptop for Deepika"
- Buttons:** A blue "Add Product" button at the bottom of the card list.
- Header Options:** A "Dimensions: Responsive" dropdown menu at the top right.