Don Bosco Institute of Technology
Kurla (W), Mumbai 400 070

Internet Programming Lab

Name: Umme Atiya Quraishi
Class: TEIT
Roll no.: 47
Date : 30/08/2024
Experiment no. 8
Title: To Design a web page using React Props and State

Theory:

In React, **Props** (short for properties) and **State** are two fundamental concepts that enable dynamic and interactive user interfaces. They help manage data flow and component behavior within React applications.

Props

1. **Definition**:
    - Props are read-only attributes passed from a parent component to a child component. They allow components to receive data and configure themselves accordingly.
2. **Purpose**:
    - Props are primarily used to pass data down the component tree. This allows for reusable and composable components that can be customized based on the data provided.
3. **Characteristics**:
    - **Immutable**: Once a component receives props, it cannot modify them. This ensures a unidirectional data flow in React, making it easier to understand how data changes over time.
    - **Customizable**: Props allow components to be highly customizable. Different parent components can pass different props, enabling the same component to behave differently in various contexts.
4. **Usage**:

- Props can be of any data type, including strings, numbers, arrays, objects, or even functions. This flexibility allows components to be versatile and adaptable.
- Props can also be used to pass event handlers from parent to child components, enabling communication between components.

State

1. **Definition**:
    - State is a mutable data structure that represents the current condition or status of a component. It is local to the component and can be changed in response to user actions or other events.
2. **Purpose**:
    - State is used to manage dynamic data within a component. It allows components to respond to user input, manage form data, and trigger UI updates when the data changes.
3. **Characteristics**:
    - **Mutable**: Unlike props, state can be changed within the component using the `setState` method in class components or the `useState` hook in functional components.
    - **Local**: State is confined to the component it belongs to, making it suitable for managing component-specific data.
4. **Usage**:
    - State is commonly used to track user interactions, such as form inputs, toggle states, and loading statuses.
    - Changes in state trigger a re-render of the component, updating the UI to reflect the new state.

Conclusion

In summary, **Props** and **State** are essential concepts in React that work together to create interactive user interfaces. While props allow for data flow and customization of components, state enables components to manage their internal data and respond to user actions. Understanding how to effectively use props and state is crucial for building robust React applications.

Code:

ProductList.js:

```
// ProductList.js
import React, { useState } from 'react';
import Product from './Product';  // Child component to display individual product

function ProductList() {
  // State to store the list of products
  const [products, setProducts] = useState([
    { id: 1, name: 'Laptop', price: 999 },
    { id: 2, name: 'Phone', price: 499 },
  ]);

  // State to store the input values for new product
  const [newProduct, setNewProduct] = useState({ name: '', price: '' });

  // Handle input change for the new product
  const handleInputChange = (e) => {
    const { name, value } = e.target;
    setNewProduct({ ...newProduct, [name]: value });
  };

  // Function to add a new product to the list
  const addProduct = () => {
    if (newProduct.name && newProduct.price) {
      const newProductObj = {
        id: products.length + 1,
        name: newProduct.name,
        price: parseFloat(newProduct.price),
      };

      setProducts([...products, newProductObj]);  // Update the state with the new product
      setNewProduct({ name: '', price: '' });    // Clear input fields
    }
  };

  return (
    <div style={productListStyle}>
      <h1>Product List</h1>

      {/* Input form to add new product */}
      <div>
        <input
          type="text"
          name="name"
          placeholder="Product name"
          value={newProduct.name}
          onChange={handleInputChange}
          style={inputStyle}
        />
        <input
          type="number"
          name="price"
          placeholder="Product price"
          value={newProduct.price}
          onChange={handleInputChange}
          style={inputStyle}
        />
        <button onClick={addProduct} style={buttonStyle}>Add Product</button>
      </div>

      {/* List of products */}
      <div>
        {products.map(product => (
          <Product key={product.id} product={product} />
        ))}
      </div>
    </div>
  );
}

// Styles for the component
const productListStyle = {
  padding: '20px',
  textAlign: 'center',
  fontFamily: 'Arial, sans-serif',
};

const inputStyle = {
  margin: '5px',
  padding: '10px',
  borderRadius: '5px',
  border: '1px solid #ccc',
};

const buttonStyle = {
  padding: '10px 20px',
  margin: '10px',
  backgroundColor: '#007bff',
  color: 'white',
  border: 'none',
  borderRadius: '5px',
  cursor: 'pointer',
};

export default ProductList;
```

Product.js:

```
// Product.js
import React from 'react';

// Component to display individual product details (using props)
function Product({ product }) {
  return (
    <div style={productCardStyle}>
      <h3>{product.name}</h3>
      <p>Price: ${product.price}</p>
    </div>
  );
}

// Styles for product card
```

```
const productCardStyle = {
  border: '1px solid #ccc',
  padding: '15px',
  margin: '10px auto',
  width: '300px',
  borderRadius: '8px',
  textAlign: 'center',
};

export default Product;
```

App.js:

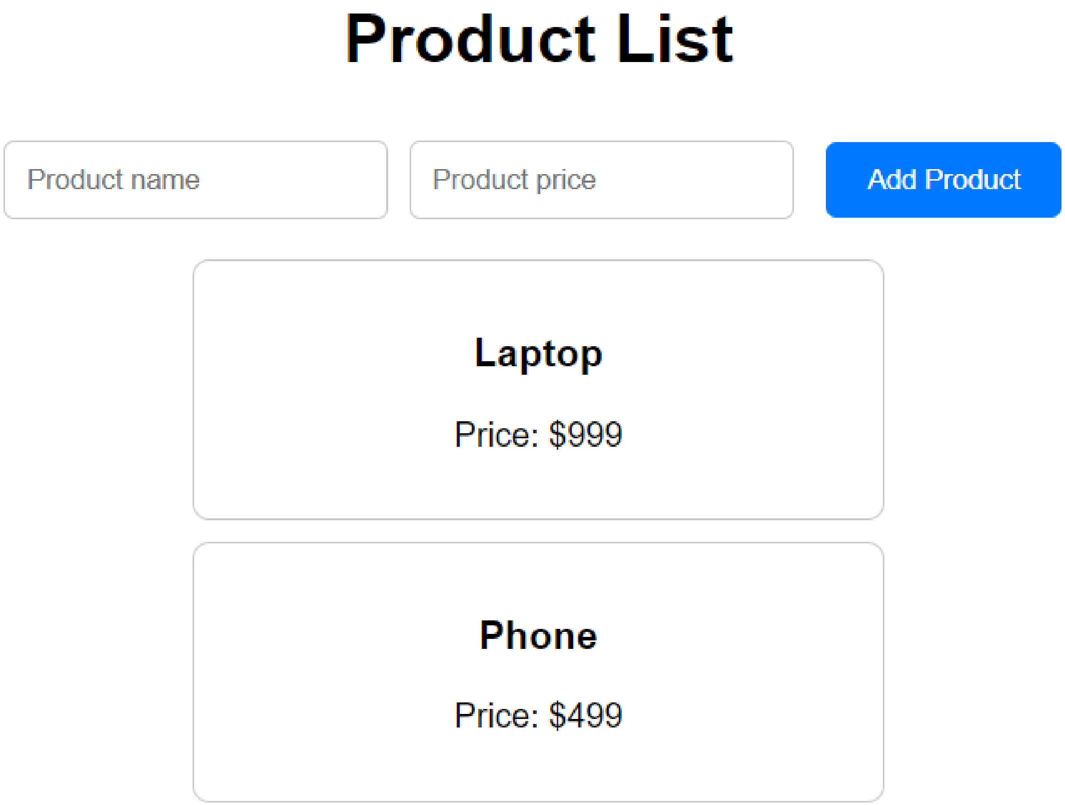
```
import React from 'react';
import './App.css';
import UserDashboard from './exp9';
import ProductList from './ProductList'; // Import ProductList component


function App() {
  return (
    <div className="App">
        {/* <UserDashboard />
        <Exp8 />
        */}
            <ProductList />

    </div>
  );
}

export default App;
```

Output:



Product List

Laptop

Price: $999

Phone

Price: $499