EXPERIMENT-08

- **AIM: -** To create a full-stack application where the frontend renders a ProductCard component using props received from a backend API.
- **THEORY:** - Concept of Props in React Props (short for properties) are a way of passing data from a parent component to a child component in React.
- They make components reusable, because the same component can display different data depending on the props passed.
- Props are read-only → child components cannot modify them, they can only use them.
- In this experiment, ProductCard is a child component, and App is the parent which passes name, price, description, and image as props.
- ProductCard Component A ProductCard is a reusable UI component that displays product details.
- Instead of hardcoding product info, we pass data as props.
- This makes the component dynamic (can render any product just by changing props).
- Example:jsx Copy Edit 3. Backend API (Node.js + Express)→ To simulate real-world e-commerce systems, product details are usually stored on a server.
- The backend provides product data in JSON format through an API endpoint (/products).
- This ensures separation of concerns:
  - Backend → Handles data storage & delivery.
  - Frontend → Handles UI display & user interaction.
  - Frontend-Backend Integration The React app uses fetch (or axios) to request product data from the backend.
- Once received, the data is stored in a state variable (useState).
- The map() function iterates over the product list and passes props to each ProductCard.

- This flow mimics real-world e-commerce applications like Flipkart, Amazon, or Myntra.

➢ Why Props are Important in Full-Stack Development?
Reusability: One ProductCard can render hundreds of products.

➢ Scalability: Changing product data on the server reflects instantly on the UI.

➢ Separation of Concerns: Backend manages data, frontend manages UI.

➢ Flexibility: Props can represent any data (text, images, numbers, functions).

- **CODE:-**

➢ BACKEND→

```
const express = require("express");
const cors = require("cors");

const app = express();
app.use(cors());

const products = [
  {
    id: 1,
    name: "Wireless Headphones",
    price: 2499,
    description: "High-quality sound with noise cancellation.",
    image: "https://via.placeholder.com/150"
  },
  {
    id: 2,
    name: "Smart Watch",
    price: 4999,
    description: "Track your fitness and stay connected.",
    image: "https://via.placeholder.com/150"
  }
];
```

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

app.listen(5000, () => {
  console.log("Server running on http://localhost:5000");
});
```

- FRONTEND→

```
export default function ProductCard({ name, price, description, image }) {
  return (
    <div className="max-w-xs bg-white rounded-2xl shadow-lg p-4 text-center">
      <img src={image} alt={name} className="w-full h-40 object-cover rounded-xl" />
      <h2 className="text-xl font-bold mt-3">{name}</h2>
      <p className="text-gray-600">{description}</p>
      <p className="text-lg font-semibold text-green-600 mt-2">₹{price}</p>
      <button className="mt-3 px-4 py-2 bg-blue-500 text-white rounded-xl hover:bg-blue-600">
        Buy Now
      </button>
    </div>
  );
}
```

-
```
import { useEffect, useState } from "react";
import ProductCard from "./ProductCard";

function App() {
  const [products, setProducts] = useState([]);
```

```
  useEffect(() => {
    fetch("http://localhost:5000/products")
      .then((res) => res.json())
      .then((data) => setProducts(data));
  }, []);

  return (
    <div className="min-h-screen bg-gray-100 flex flex-wrap gap-
6 p-6">
      {products.map((product) => (
        <ProductCard
          key={product.id}
          name={product.name}
          price={product.price}
          description={product.description}
          image={product.image}
        />
      ))}
    </div>
  );
}

export default App;
```

- **OUTPUT→**