**CODE :**

**Client**

**App.jsx**

import React from "react";

import ProductPage from "./ProductPage";

function App() {

return (

<div className="App">

<h1>Product Management</h1>

<ProductPage />

</div>

);

}

export default App;

**Product.jsx**

import React, { useEffect, useState } from "react";

import axios from "axios";

function ProductPage() {

const [products, setProducts] = useState([]);

const [form, setForm] = useState({ name: "", price: "", category: "" });

const fetchProducts = async () => {

const res = await axios.get("http://localhost:5000/api/products");

setProducts(res.data);

};

const handleSubmit = async (e) => {

e.preventDefault();

await axios.post("http://localhost:5000/api/products", form);

setForm({ name: "", price: "", category: "" });

fetchProducts();

};

const deleteProduct = async (id) => {

await axios.delete(`http://localhost:5000/api/products/${id}`);

fetchProducts();

};

useEffect(() => {

fetchProducts();

}, []);

return (

<div>

<form onSubmit={handleSubmit}>

<input

placeholder="Name"

value={form.name}

onChange={(e) => setForm({ ...form, name: e.target.value })}

/>

<input

placeholder="Price"

value={form.price}

onChange={(e) => setForm({ ...form, price: e.target.value })}

/>

<input

placeholder="Category"

value={form.category}

onChange={(e) => setForm({ ...form, category: e.target.value })}

/>

<button type="submit">Add Product</button>

</form>

<ul>

{products.map((p) => (

<li key={p.\_id}>

{p.name} - ₹{p.price} [{p.category}]

<button onClick={() => deleteProduct(p.\_id)}>❌</button>

</li>

))}

</ul>

</div>

);

}

export default ProductPage;

**SERVER :**

**ProductController.js**

import Product from "../models/Product.js";

export const getAllProducts = async (req, res) => {

const products = await Product.find();

res.json(products);

};

export const createProduct = async (req, res) => {

const newProduct = new Product(req.body);

await newProduct.save();

res.status(201).json(newProduct);

};

export const updateProduct = async (req, res) => {

const { id } = req.params;

const updated = await Product.findByIdAndUpdate(id, req.body, { new: true });

res.json(updated);

};

export const deleteProduct = async (req, res) => {

await Product.findByIdAndDelete(req.params.id);

res.json({ message: "Product deleted" });

};

**productModel.js**

import mongoose from "mongoose";

const productSchema = new mongoose.Schema({

name: { type: String, required: true },

price: Number,

description: String,

category: String,

});

export default mongoose.model("Product", productSchema);

**Productroutes.js**

import express from "express";

import {

getAllProducts,

createProduct,

updateProduct,

deleteProduct,

} from "../controllers/productController.js";

const router = express.Router();

router.get("/", getAllProducts);

router.post("/", createProduct);

router.put("/:id", updateProduct);

router.delete("/:id", deleteProduct);

export default router;

**.env**

MONGO\_URI=mongodb://127.0.0.1:27017/productsDB

**Index.js**

import express from "express";

import mongoose from "mongoose";

import dotenv from "dotenv";

import cors from "cors";

import productRoutes from "./routes/productRoutes.js";

dotenv.config();

const app = express();

app.use(cors());

app.use(express.json());

app.use("/api/products", productRoutes);

mongoose

.connect(process.env.MONGO\_URI)

.then(() => {

app.listen(5000, () => console.log("Server running on port 5000"));

})

.catch((err) => console.error(err));

**OUTPUT**



