





EXPERIMENT 4.1

AIM: CRUD Operations for Product Database Using Mongoose

Project Overview

We'll build a **Product Management System** where you can:

-  Add new products
-  View all products
-  Edit product details
-  Delete products

Project Structure

```
product-crud/
├── backend/
│   ├── models/
│   │   └── Product.js
│   ├── routes/
│   │   └── productRoutes.js
│   ├── server.js
│   └── .env
├── frontend/
│   ├── public/
│   └── src/
│       ├── components/
│       │   ├── ProductForm.js
│       │   ├── ProductList.js
│       │   └── EditProduct.js
│       ├── App.js
│       └── index.js
└── package.json
```

Backend (Node.js + Express + Mongoose)

1. Install Dependencies

```
mkdir backend && cd backend
npm init -y
npm install express mongoose cors dotenv
npm install nodemon --save-dev
```

Add "start": "nodemon server.js" to package.json.

2. server.js

```
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
require("dotenv").config();

const productRoutes = require("../routes/productRoutes");

const app = express();

// Middleware
app.use(express.json());
app.use(cors());

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

// MongoDB connection
mongoose.connect(process.env.MONGO_URI)
  .then(() => {
    console.log("MongoDB connected ✅");
    app.listen(5000, () => console.log("Server running on port 5000"));
  })
  .catch((err) => console.log(err));
```

3. models/Product.js

```
const mongoose = require("mongoose");

const productSchema = new mongoose.Schema({
  name: { type: String, required: true },
  price: { type: Number, required: true },
  description: String,
  category: String,
  stock: { type: Number, default: 0 }
}, { timestamps: true });

module.exports = mongoose.model("Product", productSchema);
```

4. routes/productRoutes.js

```
const express = require("express");
const Product = require("../models/Product");
const router = express.Router();

// CREATE product
router.post("/", async (req, res) => {
  try {
    const product = new Product(req.body);
    await product.save();
    res.json(product);
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});
```

```

    }
  });

  // READ all products
  router.get("/", async (req, res) => {
    try {
      const products = await Product.find();
      res.json(products);
    } catch (err) {
      res.status(500).json({ error: err.message });
    }
  });

  // READ single product
  router.get("/:id", async (req, res) => {
    try {
      const product = await Product.findById(req.params.id);
      if (!product) return res.status(404).json({ error: "Product not found" });
      res.json(product);
    } catch (err) {
      res.status(500).json({ error: err.message });
    }
  });

  // UPDATE product
  router.put("/:id", async (req, res) => {
    try {
      const updatedProduct = await Product.findByIdAndUpdate(
        req.params.id, req.body, { new: true }
      );
      res.json(updatedProduct);
    } catch (err) {
      res.status(500).json({ error: err.message });
    }
  });

  // DELETE product
  router.delete("/:id", async (req, res) => {
    try {
      await Product.findByIdAndDelete(req.params.id);
      res.json({ message: "Product deleted" });
    } catch (err) {
      res.status(500).json({ error: err.message });
    }
  });

  module.exports = router;

```

Frontend (React)

1. Setup React

```

npx create-react-app frontend
cd frontend
npm install axios react-router-dom

```

2. src/App.js

```
import React from "react";
import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
import ProductList from "../components/ProductList";
import ProductForm from "../components/ProductForm";
import EditProduct from "../components/EditProduct";

function App() {
  return (
    <Router>
      <div className="container">
        <h1>Product Management</h1>
        <Routes>
          <Route path="/" element={<ProductList />} />
          <Route path="/add" element={<ProductForm />} />
          <Route path="/edit/:id" element={<EditProduct />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

3. src/components/ProductList.js

```
import React, { useEffect, useState } from "react";
import axios from "axios";
import { Link } from "react-router-dom";

const ProductList = () => {
  const [products, setProducts] = useState([]);

  const fetchProducts = async () => {
    const res = await axios.get("http://localhost:5000/api/products");
    setProducts(res.data);
  };

  const deleteProduct = async (id) => {
    await axios.delete(`http://localhost:5000/api/products/${id}`);
    fetchProducts();
  };

  useEffect(() => {
    fetchProducts();
  }, []);

  return (
    <div>
      <Link to="/add">+ Add Product</Link>
      <ul>
        {products.map((p) => (
          <li key={p._id}>
            {p.name} - ${p.price}
            <Link to={`/edit/${p._id}`}> ✎ Edit</Link>
            <button onClick={() => deleteProduct(p._id)}> ✖
            Delete</button>
          </li>
        ))}
      </ul>
    </div>
  );
}
```

```

        </li>
      )})
    </ul>
  </div>
);
};

export default ProductList;

```

4. src/components/ProductForm.js

```

import React, { useState } from "react";
import axios from "axios";
import { useNavigate } from "react-router-dom";

const ProductForm = () => {
  const [form, setForm] = useState({ name: "", price: "", description: "",
category: "", stock: "" });
  const navigate = useNavigate();

  const handleChange = (e) => setForm({ ...form, [e.target.name]:
e.target.value });

  const handleSubmit = async (e) => {
    e.preventDefault();
    await axios.post("http://localhost:5000/api/products", form);
    navigate("/");
  };

  return (
    <form onSubmit={handleSubmit}>
      <input name="name" placeholder="Name" onChange={handleChange}
required />
      <input name="price" type="number" placeholder="Price"
onChange={handleChange} required />
      <input name="description" placeholder="Description"
onChange={handleChange} />
      <input name="category" placeholder="Category" onChange={handleChange}
/>
      <input name="stock" type="number" placeholder="Stock"
onChange={handleChange} />
      <button type="submit">Add Product</button>
    </form>
  );
};

export default ProductForm;

```

5. src/components/EditProduct.js

```

import React, { useEffect, useState } from "react";
import axios from "axios";
import { useParams, useNavigate } from "react-router-dom";

const EditProduct = () => {
  const { id } = useParams();
  const navigate = useNavigate();

```

```

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

    useEffect(() => {
        axios.get(`http://localhost:5000/api/products/${id}`)
            .then(res => setForm(res.data));
    }, [id]);

    const handleChange = (e) => setForm({ ...form, [e.target.name]:
e.target.value });

    const handleSubmit = async (e) => {
        e.preventDefault();
        await axios.put(`http://localhost:5000/api/products/${id}`, form);
        navigate("/");
    };

    return (
        <form onSubmit={handleSubmit}>
            <input name="name" value={form.name} onChange={handleChange} required
/>
            <input name="price" type="number" value={form.price}
onChange={handleChange} required />
            <input name="description" value={form.description}
onChange={handleChange} />
            <input name="category" value={form.category} onChange={handleChange}
/>
            <input name="stock" type="number" value={form.stock}
onChange={handleChange} />
            <button type="submit">Update Product</button>
        </form>
    );
};

export default EditProduct;

```



Product Management

+ Add Product

iPhone 14 - \$999



Edit



Delete

Nike Sneakers - \$120



Edit



Delete