**1. Set Up Your Development Environment**

Before you start, make sure you have Node.js, npm, MongoDB, and a code editor (like VSCode) installed.

**Dependencies**

* **MongoDB**: Database to store stock and transaction details.
* **Express.js**: Server-side framework for handling API requests.
* **Node.js**: JavaScript runtime to execute backend code.
* **React.js**: Frontend framework for building the user interface.

Install Node.js and npm if you haven’t already.

**2. Backend (Node.js + Express.js + MongoDB)**

**Step 1: Create a project directory**

bash

Copy code

mkdir stock-management

cd stock-management

npm init -y

**Step 2: Install backend dependencies**

bash

Copy code

npm install express mongoose cors body-parser dotenv

**Step 3: Create the server**

In the root directory, create a file named server.js:

javascript

Copy code

const express = require('express');

const mongoose = require('mongoose');

const cors = require('cors');

const bodyParser = require('body-parser');

require('dotenv').config();

const app = express();

// Middleware

app.use(cors());

app.use(bodyParser.json());

// MongoDB Connection

const mongoURI = process.env.MONGO\_URI;

mongoose.connect(mongoURI, { useNewUrlParser: true, useUnifiedTopology: true })

.then(() => console.log("MongoDB connected"))

.catch((err) => console.log("Error connecting to MongoDB:", err));

// Stock Schema

const stockSchema = new mongoose.Schema({

name: String,

quantity: Number,

price: Number,

});

const Stock = mongoose.model('Stock', stockSchema);

// API Endpoints

// Create new stock item

app.post('/api/stocks', async (req, res) => {

try {

const { name, quantity, price } = req.body;

const newStock = new Stock({ name, quantity, price });

await newStock.save();

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

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Get all stock items

app.get('/api/stocks', async (req, res) => {

try {

const stocks = await Stock.find();

res.status(200).json(stocks);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Update stock item

app.put('/api/stocks/:id', async (req, res) => {

try {

const stock = await Stock.findByIdAndUpdate(req.params.id, req.body, { new: true });

res.status(200).json(stock);

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Delete stock item

app.delete('/api/stocks/:id', async (req, res) => {

try {

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

res.status(204).send();

} catch (error) {

res.status(500).json({ error: error.message });

}

});

// Start Server

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

console.log(`Server running on port ${PORT}`);

});

**Step 4: Set up .env file**

Create a .env file in your root directory to store your MongoDB URI:

bash

Copy code

MONGO\_URI=mongodb://localhost:27017/stock\_management

**3. Frontend (React.js)**

**Step 1: Create a React app**

Navigate back to the root directory and create a new React app.

bash

Copy code

npx create-react-app client

cd client

**Step 2: Install frontend dependencies**

bash

Copy code

npm install axios react-router-dom

**Step 3: Create React components**

1. **App.js**

Replace the contents of src/App.js with the following:

javascript

Copy code

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

import axios from "axios";

import { BrowserRouter as Router, Route, Switch } from "react-router-dom";

import StockList from "./components/StockList";

import AddStock from "./components/AddStock";

import EditStock from "./components/EditStock";

const App = () => {

return (

<Router>

<div className="App">

<h1>Stock Management</h1>

<Switch>

<Route path="/" exact component={StockList} />

<Route path="/add" component={AddStock} />

<Route path="/edit/:id" component={EditStock} />

</Switch>

</div>

</Router>

);

};

export default App;

1. **StockList.js**

Create a src/components/StockList.js file for displaying all stock items.

javascript

Copy code

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

import axios from "axios";

import { Link } from "react-router-dom";

const StockList = () => {

const [stocks, setStocks] = useState([]);

useEffect(() => {

axios.get("http://localhost:5000/api/stocks")

.then((response) => {

setStocks(response.data);

})

.catch((error) => {

console.error("There was an error fetching the stock list!", error);

});

}, []);

const deleteStock = async (id) => {

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

setStocks(stocks.filter((stock) => stock.\_id !== id));

};

return (

<div>

<h2>Stock List</h2>

<Link to="/add">Add New Stock</Link>

<ul>

{stocks.map((stock) => (

<li key={stock.\_id}>

{stock.name} - {stock.quantity} - ${stock.price}

<Link to={`/edit/${stock.\_id}`}> Edit</Link>

<button onClick={() => deleteStock(stock.\_id)}>Delete</button>

</li>

))}

</ul>

</div>

);

};

export default StockList;

1. **AddStock.js**

Create a src/components/AddStock.js file for adding a new stock item.

javascript

Copy code

import React, { useState } from "react";

import axios from "axios";

import { useHistory } from "react-router-dom";

const AddStock = () => {

const [name, setName] = useState("");

const [quantity, setQuantity] = useState(0);

const [price, setPrice] = useState(0);

const history = useHistory();

const addStock = async () => {

await axios.post("http://localhost:5000/api/stocks", { name, quantity, price });

history.push("/");

};

return (

<div>

<h2>Add New Stock</h2>

<input

type="text"

placeholder="Stock Name"

value={name}

onChange={(e) => setName(e.target.value)}

/>

<input

type="number"

placeholder="Quantity"

value={quantity}

onChange={(e) => setQuantity(e.target.value)}

/>

<input

type="number"

placeholder="Price"

value={price}

onChange={(e) => setPrice(e.target.value)}

/>

<button onClick={addStock}>Add Stock</button>

</div>

);

};

export default AddStock;

1. **EditStock.js**

Create a src/components/EditStock.js file for editing a stock item.

javascript

Copy code

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

import axios from "axios";

import { useHistory, useParams } from "react-router-dom";

const EditStock = () => {

const { id } = useParams();

const history = useHistory();

const [stock, setStock] = useState({ name: "", quantity: 0, price: 0 });

useEffect(() => {

axios.get(`http://localhost:5000/api/stocks/${id}`)

.then((response) => {

setStock(response.data);

})

.catch((error) => {

console.error("Error fetching stock:", error);

});

}, [id]);

const updateStock = async () => {

await axios.put(`http://localhost:5000/api/stocks/${id}`, stock);

history.push("/");

};

return (

<div>

<h2>Edit Stock</h2>

<input

type="text"

value={stock.name}

onChange={(e) => setStock({ ...stock, name: e.target.value })}

/>

<input

type="number"

value={stock.quantity}

onChange={(e) => setStock({ ...stock, quantity: e.target.value })}

/>

<input

type="number"

value={stock.price}

onChange={(e) => setStock({ ...stock, price: e.target.value })}

/>

<button onClick={updateStock}>Update Stock</button>

</div>

);

};

export default EditStock;

**4. Final Steps**

1. **Run the Backend**: Run the Node.js/Express backend.

bash

Copy code

node server.js

1. **Run the Frontend**: In the client directory, run:

bash

Copy code

npm start