1. **Create the Project Directory**

mkdir ecommerce-platform

cd ecommerce-platform

**2. Set Up the Backend (Node.js + Express + PostgreSQL)**

**- Create the Backend Directory**

mkdir server

cd server

* **Initialize the Node.js Project**

npm init -y (This command creates a package.json file.)

* **To install packages**

npm install express pg bcrypt jsonwebtoken cors dotenv

ecommerce-platform/

├── server/

│ ├── db.js <-- Database connection file

│ ├── index.js <-- Main server file

│ ├── controllers/

│ │ └── authController.js

│ ├── routes/

│ │ └── auth.js

└── client/

src/

├── components/

│ ├── Header.js

│ ├── Footer.js

│ ├── ProductList.js

│ ├── ProductItem.js

│ ├── Cart.js

│ └── Login.js

├── App.js

├── App.css

└── index.js

**--- // server/db.js ---**

const { Pool } = require('pg');

const pool = new Pool({

user: 'your\_db\_user', // replace with your PostgreSQL username

host: 'localhost',

database: 'your\_database', // replace with your PostgreSQL database name

password: 'your\_password', // replace with your PostgreSQL password

port: 5432,

});

module.exports = pool;

**--- // server/controllers/authController.js ---**

const db = require('../db');

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

// Register a new user

const register = async (req, res) => {

const { username, email, password } = req.body;

const hashedPassword = await bcrypt.hash(password, 10);

try {

const result = await db.query('INSERT INTO users (username, email, password) VALUES ($1, $2, $3) RETURNING \*', [username, email, hashedPassword]);

res.status(201).json(result.rows[0]);

} catch (error) {

res.status(500).json({ error: 'User registration failed' });

}

};

// Login

const login = async (req, res) => {

const { email, password } = req.body;

try {

const result = await db.query('SELECT \* FROM users WHERE email = $1', [email]);

const user = result.rows[0];

if (user && await bcrypt.compare(password, user.password)) {

const token = jwt.sign({ id: user.id }, 'your\_jwt\_secret', { expiresIn: '1h' });

res.json({ token });

} else {

res.status(401).json({ error: 'Invalid credentials' });

}

} catch (error) {

res.status(500).json({ error: 'Login failed' });

}

};

module.exports = { register, login };

**--- // server/routes/auth.js ---**

const express = require('express');

const { register, login } = require('../controllers/authController');

const router = express.Router();

router.post('/register', register);

router.post('/login', login);

module.exports = router;

**--- // server/index.js ---**

const express = require('express');

const cors = require('cors');

const authRoutes = require('./routes/auth');

const app = express();

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

app.use(cors());

app.use(express.json());

app.use('/api/auth', authRoutes);

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

res.send('E-commerce API is running');

});

app.listen(PORT, () => {

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

});

**POSTGRES**

**CREATE TABLE users (**

**id SERIAL PRIMARY KEY,**

**username VARCHAR(50) NOT NULL,**

**email VARCHAR(50) NOT NULL UNIQUE,**

**password VARCHAR(255) NOT NULL**

**);**

**Step 3: Setting Up the Frontend (React)**

**1. Create the React Application**

1. **Open a new terminal window.**
2. **Navigate back to the main project folder**

cd ecommerce-platform

npx create-react-app client

cd client

npm install axios react-router-dom

**--- App.js ---**

import React from 'react';

import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';

import Header from './components/Header';

import Footer from './components/Footer';

import ProductList from './components/ProductList';

import Cart from './components/Cart';

import Login from './components/Login';

import './App.css';

function App() {

return (

<Router>

<div className="App">

<Header />

<Routes>

<Route path="/" element={<ProductList />} />

<Route path="/cart" element={<Cart />} />

<Route path="/login" element={<Login />} />

</Routes>

<Footer />

</div>

</Router>

);

}

export default App;

**--- Header.js ---**

import React from 'react';

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

import './Header.css'; // You can create this CSS file for styling

function Header() {

return (

<header className="Header">

<h1>My E-Commerce Store</h1>

<nav>

<Link to="/">Home</Link>

<Link to="/cart">Cart</Link>

<Link to="/login">Login</Link>

</nav>

</header>

);

}

export default Header;

**--- ProductList.js ---**

import React from 'react';

import ProductItem from './ProductItem';

const products = [

{ id: 1, name: 'Product 1', price: 29.99, image: 'link\_to\_image\_1' },

{ id: 2, name: 'Product 2', price: 39.99, image: 'link\_to\_image\_2' },

// Add more products as needed

];

function ProductList() {

return (

<div className="ProductList">

{products.map((product) => (

<ProductItem key={product.id} product={product} />

))}

</div>

);

}

export default ProductList;

**--- ProductItem.js ---**

import React from 'react';

function ProductItem({ product }) {

return (

<div className="ProductItem">

<img src={product.image} alt={product.name} />

<h2>{product.name}</h2>

<p>${product.price}</p>

<button>Add to Cart</button>

</div>

);

}

export default ProductItem;

**--- Cart.js ---**

import React from 'react';

function Cart() {

// Ideally, you would manage cart state with a context or state management solution

return (

<div className="Cart">

<h2>Your Shopping Cart</h2>

{/\* Render cart items here \*/}

</div>

);

}

export default Cart;

**--- Login.js ---**

import React from 'react';

function Login() {

return (

<div className="Login">

<h2>Login</h2>

<form>

<input type="email" placeholder="Email" required />

<input type="password" placeholder="Password" required />

<button type="submit">Login</button>

</form>

</div>

);

}

export default Login;

**Style Your Components**

**Create src/components/Header.css, src/components/ProductItem.css, and so on to style your components. You can make it look more like Amazon by using a grid layout for products and styling the header.**

**/\* src/components/Header.css \*/**

.Header {

background-color: #232f3e; /\* Amazon's dark blue color \*/

color: white;

padding: 10px;

display: flex;

justify-content: space-between;

align-items: center;

}

.Header h1 {

font-size: 24px;

margin: 0;

}

.Header nav {

display: flex;

gap: 20px; /\* Space between links \*/

}

.Header a {

color: white;

text-decoration: none;

font-weight: bold;

transition: color 0.3s;

}

.Header a:hover {

color: #ff9900; /\* Amazon's orange hover color \*/

}

**/\* src/components/ProductItem.css \*/**

.ProductItem {

border: 1px solid #ddd; /\* Light gray border \*/

border-radius: 5px; /\* Rounded corners \*/

padding: 10px;

text-align: center;

margin: 15px; /\* Space around items \*/

transition: box-shadow 0.3s;

background-color: #fff; /\* White background \*/

}

.ProductItem:hover {

box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1); /\* Shadow effect on hover \*/

}

.ProductItem img {

max-width: 100%; /\* Responsive image \*/

height: auto;

margin-bottom: 10px; /\* Space below image \*/

}

.ProductItem h2 {

font-size: 18px;

margin: 10px 0; /\* Space above and below product name \*/

}

.ProductItem p {

font-size: 16px;

color: #b12704; /\* Amazon's red price color \*/

margin: 5px 0; /\* Space above and below price \*/

}

.ProductItem button {

background-color: #ff9900; /\* Amazon's orange button color \*/

color: white;

border: none;

padding: 10px 15px;

border-radius: 5px; /\* Rounded corners \*/

cursor: pointer;

transition: background-color 0.3s;

}

.ProductItem button:hover {

background-color: #e68a00; /\* Darker orange on hover \*/

}

**/\* src/components/ProductList.css \*/**

.ProductList {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(200px, 1fr)); /\* Responsive grid \*/

gap: 15px; /\* Space between grid items \*/

padding: 20px; /\* Padding around the grid \*/

}

**/\* src/components/Cart.css \*/**

.Cart {

padding: 20px;

background-color: #f2f2f2; /\* Light gray background \*/

}

.Cart h2 {

font-size: 24px;

}

.Cart ul {

list-style-type: none; /\* Remove bullet points \*/

padding: 0;

}

.Cart li {

border: 1px solid #ddd; /\* Light gray border \*/

border-radius: 5px; /\* Rounded corners \*/

padding: 10px;

margin: 10px 0; /\* Space between items \*/

background-color: #fff; /\* White background \*/

}

**/\* src/components/Login.css \*/**

.Login {

padding: 20px;

max-width: 400px; /\* Limit width \*/

margin: auto; /\* Center the login form \*/

background-color: #fff; /\* White background \*/

border: 1px solid #ddd; /\* Light gray border \*/

border-radius: 5px; /\* Rounded corners \*/

}

.Login h2 {

font-size: 24px;

text-align: center; /\* Center the heading \*/

}

.Login form {

display: flex;

flex-direction: column; /\* Stack inputs vertically \*/

}

.Login input {

padding: 10px;

margin: 10px 0; /\* Space between inputs \*/

border: 1px solid #ddd; /\* Light gray border \*/

border-radius: 5px; /\* Rounded corners \*/

}

.Login button {

background-color: #ff9900; /\* Amazon's orange color \*/

color: white;

border: none;

padding: 10px;

border-radius: 5px; /\* Rounded corners \*/

cursor: pointer;

}

.Login button:hover {

background-color: #e68a00; /\* Darker orange on hover \*/

}

**6. Import CSS Files**

**Don't forget to import the CSS files into your respective components. For example, at the top of Header.js, ProductItem.js, ProductList.js, Cart.js, and Login.js, add:**

**javascript**

**Copy code**

**import './Header.css'; // For Header component**

**import './ProductItem.css'; // For ProductItem component**

**import './ProductList.css'; // For ProductList component**

**import './Cart.css'; // For Cart component**

**import './Login.css'; // For Login component**