# React E-Commerce Cart with Express Backend

## Overview

This is a simple e-commerce cart application built with React for the frontend and Express (Node.js) for the backend. It allows users to:

- Browse products

- Add, update, and remove items from the cart

- Checkout and view order history

The backend uses dummy in-memory data, but can easily be upgraded to use a database like MongoDB or MySQL.

## Project Structure

ecommerce-project/

├── frontend/ # React app (Vite)

│ ├── src/

│ │ └── App.jsx

│ ├── package.json

│ └── vite.config.js

└── backend/ # Express backend

├── server.js

└── package.json

## Features

### Frontend (React)

- Display products dynamically

- Add to cart, remove from cart, update quantity

- View total price

- Checkout functionality

- Display past orders

### Backend (Express)

- REST API endpoints for products, cart, and orders

- CORS enabled for frontend communication

- Body-parser for handling POST request data

- In-memory storage of cart and orders

## Installation

### 1. Clone the repository

git clone https://github.com/yourusername/ecommerce-project.git

cd ecommerce-project

### 2. Install backend dependencies

cd backend

npm install

### 3. Install frontend dependencies

cd ../frontend

npm install

## Running the Project

### Start the backend server

cd backend

node server.js

Runs on http://localhost:5000

### Start the frontend

cd frontend

npm run dev

Runs on http://localhost:5173

## API Endpoints

| Endpoint | Method | Description |

|----------|--------|-------------|

| /api/products | GET | Get all products |

| /api/cart/add | POST | Add product to cart { "id": 1 } |

| /api/cart/update | POST | Update cart item { "id": 1, "qty": 2 } |

| /api/cart/remove | POST | Remove cart item { "id": 1 } |

| /api/cart/checkout | POST | Checkout and create order |

| /api/orders | GET | Get all orders |

## Packages Used

### Frontend

- react

- react-dom

- vite

- @vitejs/plugin-react

- axios (optional)

- react-router-dom (optional)

- react-toastify (optional)

- bootstrap / tailwindcss (optional)

### Backend

- express

- cors

- body-parser

- nodemon (dev, optional)

- dotenv (optional)

- mongoose / mysql2 (optional)

- bcryptjs / jsonwebtoken (optional)

## Future Improvements

- Persist data in MongoDB or MySQL instead of in-memory arrays

- Add user authentication (signup/login)

- Implement proper error handling & validations

- Add search, filters, and product categories

- Add UI enhancements (Bootstrap/Tailwind, modals, notifications)

## Author

Your Name

GitHub Profile: https://github.com/yourusername