

HARI ARUNACHALAM  
SEKAR -30445

# E-Commerce

## Description

- eShop: a simple, modern e-commerce web app
- Users can browse products, add to cart, checkout, and track orders
- Admins can manage the catalog and monitor orders
- Built with Angular (frontend) and Python Flask microservices (backend)

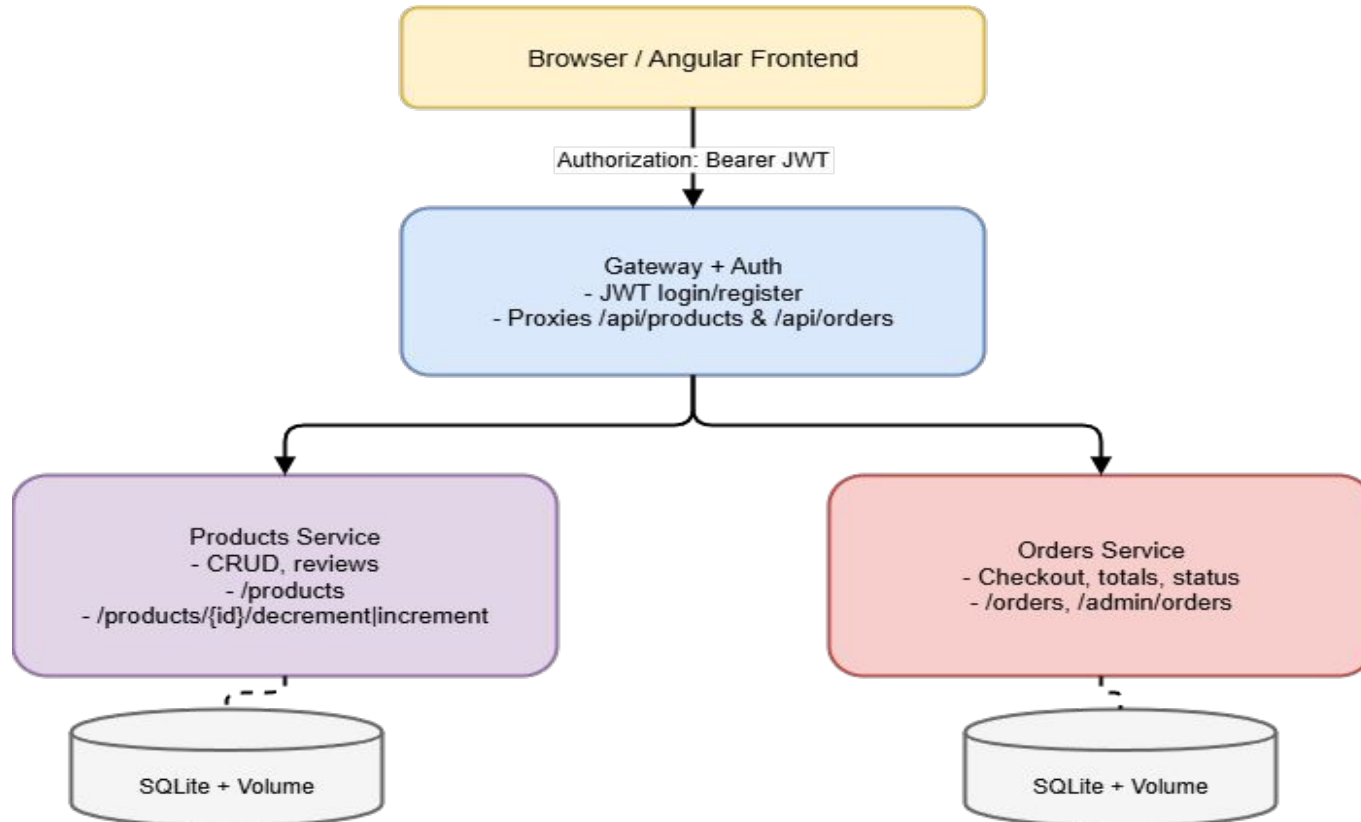
# Purpose of the project

- Deliver a clean shopping flow end-to-end
- Show microservices best practices (separate services, clear APIs)
- Add auth with JWT for user/admin actions
- Keep data persistent with volumes (SQLite)

# Personas

- Guest: can view the login/register pages
- Shopper (User): browses, reviews, checks out, views own orders
- Admin: manages and monitors orders

# Architecture diagram



All external calls hit /api at Gateway → forwarded to internal services over HTTP/JSON

# Microservices & APIs (Overview)

## **Products Service – product catalog, reviews, stock**

- Lists products & details
- Admin can add/update/delete
- Stock adjust: decrement/increment during checkout

## **Orders Service – checkout & order lifecycle**

- Create order (calculates totals, reserves stock)
- View my orders
- Admin list & status update/cancel

## **Gateway + Auth—entry point & security**

- Register / Login / Me (JWT)
- Proxies /api/products\* and /api/orders\* to internal services

# Frontend (Angular) Components

- Auth pages: Login & Register with validation; Guards (Auth, Admin, Guest)
- Catalog: product list, filters, detail + reviews
- Shop: cart & checkout; orders history for the user
- Admin: orders dashboard; shared header/notifications/interceptors

# Deployment (Docker Compose)

- 4 containers: frontend, gateway, products, orders
- Service-name DNS for internal calls (gateway → products/orders)
- Named volumes keep DBs & uploads safe across restarts
- Restart policy for stability; environment variables for config