Microservices Architecture Projects

Generated on 2025-07-28

Project 1: Inventory Service

Data Model - Product Table

Field	Type	Constraints
product_id	UUID / INT	Primary Key
name	STRING	Not Null, Unique
description	STRING	Optional
price	DECIMAL	Not Null
quantity	INTEGER	Not Null
created_at	TIMESTAMP	Default: now()
updated_at	TIMESTAMP	Auto-updated

API Endpoints (CRUD)

Method	Endpoint	Description
GET	/products	Get all products
GET	/products/{id}	Get product by ID
POST	/products	Create new product
PUT	/products/{id}	Update existing product
DELETE	/products/{id}	Delete a product

Project 2: Cart Service

Data Model - Cart Table

Field	Type	Constraints
cart_id	UUID	Primary Key
customer_id	UUID	Not Null, FK
total_price	DECIMAL	Computed field

created_at	TIMESTAMP	Default: now()
updated_at	TIMESTAMP	Auto-updated

CartItem Table

Field	Type	Constraints
item_id	UUID	Primary Key
cart_id	UUID	Foreign Key Cart(cart_id)
product_id	UUID	Foreign Key Product(product_id)
quantity	INTEGER	Not Null
price	DECIMAL	Snapshot of product price

total_price in Cart should be recalculated on insert/update/delete of CartItem.

API Endpoints (CRUD)

Method	Endpoint	Description
GET	/carts	Get all carts (optional)
GET	/carts/{cart_id}	Get specific cart
POST	/carts	Create new cart
PUT	/carts/{cart_id}	Update cart items
DELETE	/carts/{cart_id}	Delete a cart

Cart Items Management (Nested Resource)

Method	Endpoint	Description
POST	/carts/{cart_id}/items	Add item to cart
PUT	/carts/{cart_id}/items/{item_id}	Update item quantity
DELETE	/carts/{cart_id}/items/{item_id}	Remove item from cart

Project 3: User Authentication Service

Data Model - User Table

Field	Type	Constraints
user_id	UUID	Primary Key
email	STRING	Not Null, Unique
username	STRING	Not Null, Unique
password	STRING	Hashed (bcrypt, argon2, etc.)
role	STRING	e.g. 'customer', 'admin'
is_active	BOOLEAN	Default: true
created_at	TIMESTAMP	Default: now()
updated_at	TIMESTAMP	Auto-updated

Authentication Flow (JWT-based)

- Login -> Validate credentials and issue JWT.

- JWT Access Token -> Sent in Authorization: Bearer <token> header.

- Token Verification -> Middleware verifies JWT on protected routes.

- Token Expiry -> Set a reasonable TTL (e.g., 15m access, 7d refresh).

API Endpoints

Method	Endpoint	Description
POST	/auth/register	Register a new user
POST	/auth/login	Login and receive JWT token
GET	/auth/me	Get current logged-in user info
POST	/auth/logout	(Optional) Invalidate token/client
POST	/auth/refresh	(Optional) Refresh access token