BACKEND ARCHITECTURE

QuickBite

Scalable REST API for a modern food delivery platform.

Prashant

B.Tech CSE

Project Overview

Bridging Hunger & Logistics

QuickBite is a high-performance backend system designed to manage the complex lifecycle of food ordering, from menu browsing to real-time order tracking.

- **Clean Layered Architecture**
- **DTO Pattern & Security**
- **H2 & PostgreSQL Support**

Core Features

User Mgmt

Secure registration, profile management, and role-based access control.

Restaurants

Vendor profiles, dynamic menu creation, and category organization.

Smart Cart

Persistent shopping cart sessions with real-time total calculation.

Orders

Full lifecycle management: Placed, Confirmed, Prepared, Delivered.

Coupons

Flexible discount engine supporting percentage and flat-rate codes.

Error Handling

Global exception handling with semantic JSON error responses.

Technology Stack

Java 17 & Spring Boot 3.5.4

Core Framework

PostgreSQL

Production DB

H2 Database

Testing DB

Maven

Build Tool

Spring Web

REST API

JUnit 5 & Mockito

Robust Testing Suite

System Architecture

```
📦 com.quickbite.backend
   ├── config
   ├── controller // Endpoints
   ├── service // Logic
   ├── repository // Data
   ├── entity
   └── dto
```

Domain Entities

User
Credentials, Address, Orders (1:N)

Restaurant
Profile, Products (1:N), Orders

Product
Name, Price, Category, Restaurant Ref

Cart
Session Cart, CartItems (1:N)

Order
Status, Total, Payment Ref, User Ref

Coupon
Code, Discount Value, Expiry

DTO & Mapper Pattern

Why separate DTOs?

Security
Hides sensitive fields like password hashes and internal database IDs.

Serialization Safety
Prevents infinite recursion loops in JSON (Bi-directional relationships).

Bandwidth Optimization
Sends only the data the frontend needs.

```
// Mapper Usage

UserDTO dto = mapper.toDto(user);

// Resulting JSON
{
  "id": 1,
  "name": "Prashant",
  "role": "ADMIN"
}
```

Data Access Layer

Spring Data JPA Magic

We extend `JpaRepository` to get standard CRUD operations without writing a single line of SQL.

```
public interface OrderRepository extends JpaRepository {

  // Custom Derived Query
  List findByUserIdAndStatus(Long uid, String status);
}
```

Dynamic Queries

Pagination

Sorting

Business Logic (Service)

1. Validation

Check stock availability, user funds, and restaurant hours.

2. Calculation

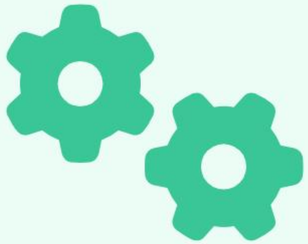
Compute cart subtotal, tax, and apply coupon discounts.

3. Execution

Save Order entity, reduce inventory, and clear User's cart.

4. Notification

Trigger email confirmation (async) and update status.



REST Controller API

METHOD	ENDPOINT	FUNCTIONALITY
POST	/api/users/register	Register a new customer account
GET	/api/restaurants/{id}	Retrieve restaurant details & menu
PUT	/api/cart/items	Add or update items in the cart
POST	/api/orders/checkout	Finalize cart and place order
GET	/api/orders/track/{id}	Get real-time order status

Exception Handling

Global Controller Advice

We use `@ControllerAdvice` to catch exceptions globally and return a standardized JSON error response.

- **ResourceNotFound**
- **InvalidCoupon**
- **OutOfStock**

```
HTTP 404 Not Found

{
  "timestamp": "2023-10-27T10:00:00",
  "status": 404,
  "error": "Not Found",
  "message": "Product with ID 55 not found"
}
```

Dual Database Strategy

Dev: H2

In-memory, fast, zero-configuration. Ideal for rapid prototyping and unit testing. Resets on restart.

Prod: PostgreSQL

Enterprise-grade object-relational database. Handles complex queries, concurrency, and persistence.

Data Bootstrapping

CommandLineRunner

We implement a bootstrapper that runs on startup to populate the database if it's empty.

- Creates Admin & User Accounts
- Onboards 5 Dummy Restaurants
- Adds 20+ Menu Items

Future Roadmap

Q1

JWT Auth

Stateless Security

Q2

Payments

Stripe Integration

Q3

Analytics

Vendor Dashboard

Q4

Microservices

Docker & K8s

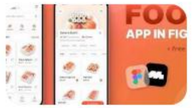
Thank You!

Ready for questions?

/prashant-quickbite

prashant@example.com

Image Sources

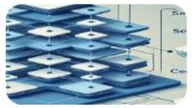


https://i.ytimg.com/vi/nh4E48zsfE8/hq720.jpg?sqp=-oaymwEhCK4FEIIDSFryq4qpAxMIARUAAAAAGAEIAADIQj0AgKJD&rs=AOn4CLB6uq87PTBHI2f1D6Wgz2tkt62eIw

Source: www.youtube.com

https://i.redd.it/programming-setup-hfw-personal-v0-oty760gty4o81.png?width=4032&format=png&auto=webp&s=c1b840828198aab54fe05662231482e7a96b1f1f

Source: www.reddit.com



https://media2.dev.to/dynamic/image/width=1000,height=420,fit=cover,gravity=auto,format=auto/https%3A%2F%2Fdev-to-uploads.s3.amazonaws.com%2Fuploads%2Farticles%2Fi3ru1ekr0gowr8gjud8a.png

Source: dev.to