

Project Title

Smart E-Commerce Price, Review & Quality Comparison Platform

Problem Statement

Online shoppers often struggle to choose the best product due to price differences, fake reviews, and lack of quality comparison across platforms like Amazon and Flipkart. Users waste time switching between multiple websites and still remain uncertain about the best deal.

This project provides a unified platform to compare **price, reviews, ratings, and quality indicators** of products across multiple e-commerce websites, helping users make informed purchasing decisions.

Module Breakdown (8 Modules)

Frontend Modules (3)

F1. Authentication & User Interface Module

- User registration and login
- Clean product search UI
- Secure session handling

F2. Product Comparison Module

- Search products by name/category
- Side-by-side comparison (Amazon vs Flipkart)
- Price, rating, review count comparison

F3. User Dashboard & Review Module

- View search history
 - Save favorite products
 - Submit platform feedback
-

Backend Modules (5)

B1. Authentication & Authorization Service

- JWT-based authentication
- Role-based access (User/Admin)

B2. Product Aggregation Service

- Fetch product data from multiple platforms (API / scraping)
- Normalize data into common format

B3. Price & Review Analysis Service

- Compare prices across platforms
- Calculate average ratings and sentiment score

B4. Recommendation & Ranking Service

- Rank products based on price + rating + review quality
- Suggest best deal

B5. Database & Persistence Layer

- Store users, products, price history, reviews
 - Maintain analytics data
-

Requirements

Functional Requirements

- User login and authentication
- Search products across multiple e-commerce sites
- Compare prices, ratings, and reviews
- Identify best value product
- View historical price trends

Non-Functional Requirements

- Low latency search response
 - Scalable for high traffic
 - Secure handling of user data
 - High availability
 - Fault tolerance
-

High-Level Design (HLD)

The system follows a **microservices-based cloud architecture**.

Users access the web application built with React. All requests are routed through an **API Gateway** which handles authentication, rate limiting, and routing.

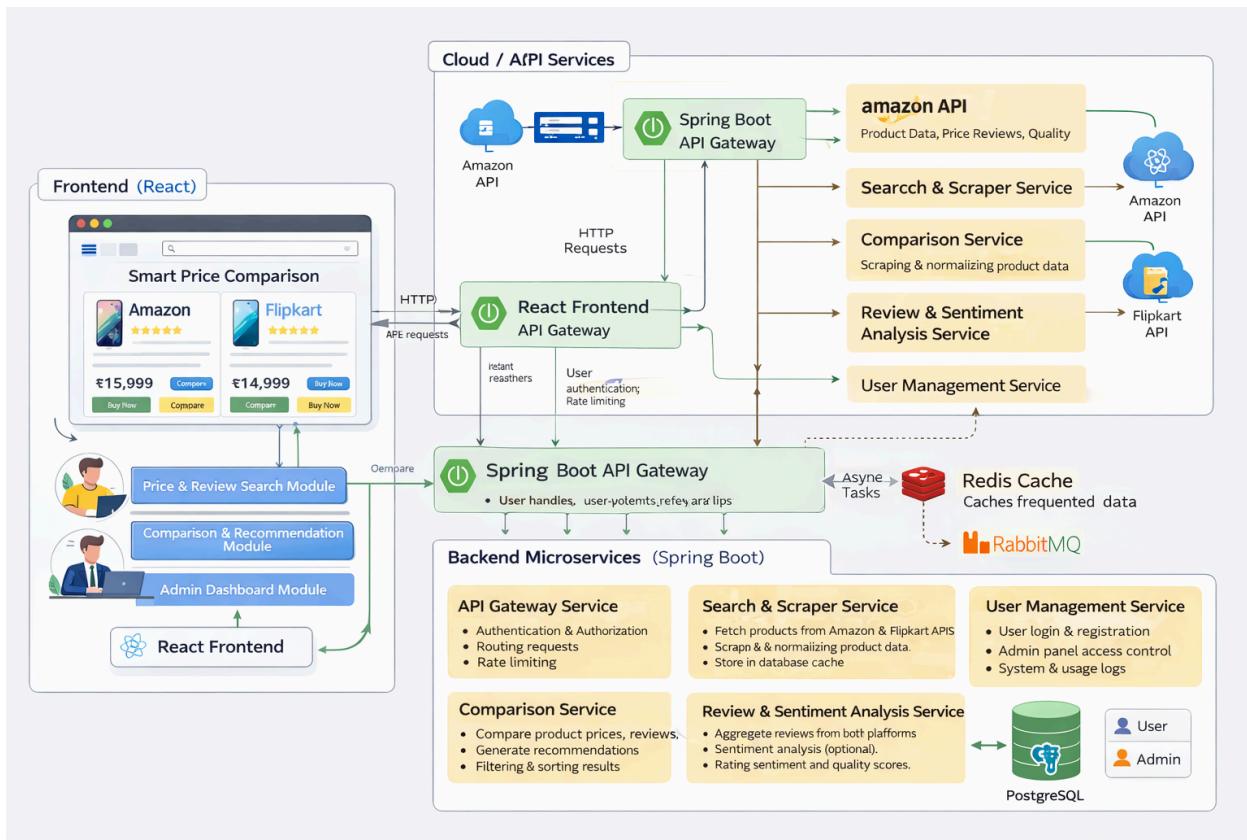
Each backend function is implemented as an independent **Spring Boot microservice**. Product data is fetched from external e-commerce platforms, processed, cached using Redis, and stored in a relational database.

A recommendation engine ranks products based on price, reviews, and quality metrics. Analytics dashboards help visualize trends.

System Architecture Flow



System Design



Low-Level Design (LLD)

API Gateway

- Request routing
- JWT validation
- Rate limiting

Authentication Service

- User login & registration
- Token generation
- Role validation

Product Aggregation Service

- Fetch data from Amazon / Flipkart
- Data normalization
- Error handling

Analysis Service

- Price comparison logic
- Review sentiment scoring
- Product ranking algorithm

Database Design

- Users Table
 - Products Table
 - Price History Table
 - Reviews Table
-

Technology Stack

Frontend: React.js, Tailwind / Material UI

Backend: Spring Boot, Spring Security, JWT

Database: MySQL / MongoDB

Caching: Redis

DevOps: GitHub, Render.

My Role

Full Stack Developer & Data Analyst

- Designed end-to-end system architecture
- Developed React frontend and Spring Boot APIs
- Implemented product comparison logic

- Analyzed review and pricing data
 - Optimized performance using caching
-

Conclusion

This project solves a real-world e-commerce problem by providing a single platform for transparent price and quality comparison. The system is scalable, secure, and extensible for future platforms.