



01

Cloud-Based Infrastructure

- I'd opt for a serverless architecture like AWS Lambda or Google Cloud Functions
- NoSQL database like DynamoDB for user-specific cart data
- Kubernetes allows the application to automatically scale



02

Minimal Frontend

- Instead of overloading the cart with heavy JavaScript frameworks
- mobile-friendly using CSS frameworks like Bootstrap or Tailwind
- I would implement lazy loading for images and cache cart data locally



03

Personalization

- I would implement basic personalization by analyzing user shopping behaviors and simple machine learning algorithms
- Integrating a big data platform (e.g., Apache Spark) could help process data for large-scale recommendation systems.



04

Basic but Robust Features

- Add/Remove items
- Quantity modification
- Real-time price updates
- Simple checkout process
- Save items for later

05

Microservices Architecture:

Break down cart functionalities (e.g., user management, product listing, checkout) into separate microservices so that each service can scale independently based on demand.

Secure Payments with Third-Party Providers:

I would integrate third-party payment gateways like Google pay and UPI payments.

