**DANISH RANJAN**,   
Bankend Developer @ Infosys   
2.5 years of relevant Experience  
  
Sets of answer

**Project: Equinox eCommerce Web Application (Infosys owned)**

**a. What did the system do?**

The Equinox eCommerce Web Application is designed to enhance the online shopping experience by integrating social features and providing personalized recommendations. It includes:

* **Social Experience Integration**: Allowing users to share their purchases, reviews, and wishlists on social media platforms.
* **Recommendation Engine**: Offering product suggestions based on user behavior and preferences.
* **Admin Orchestration Service**: Managing backend operations efficiently to ensure smooth performance and scalability.

**b. What other systems have you seen in the wild like that?**

Similar systems include:

* **Amazon**: Known for its robust recommendation engine and seamless social sharing features.
* **eBay**: Offers personalized recommendations and integrates social features to enhance user engagement.
* **Shopify**: Provides eCommerce solutions with social media integration and personalized product suggestions.

**c. How do you approach the development problem?**

1. **Requirement Gathering**: Collaborate with cross-functional teams to understand user needs and business requirements.
2. **Design**: Create technical specifications and design efficient REST API architectures.
3. **Implementation**: Develop and integrate features using Java, Spring Boot, and microservices.
4. **Optimization**: Focus on performance tuning, caching strategies, and server-side optimizations for low latency and high throughput.
5. **Testing**: Use tools like Postman for API testing and ensure the system meets quality standards.
6. **Deployment**: Deploy the application using CI/CD pipelines and monitor its performance.

**d. What were interesting aspects where you copied code from Stack Overflow?**

* **Caching Implementation**: Found a solution for implementing Redis caching to improve the scalability of the application.
* **Error Handling**: Used a code snippet for global exception handling in Spring Boot to standardize error responses across the application.
* **Pagination**: Implemented pagination for REST APIs using a Stack Overflow example to handle large datasets efficiently.

**e. What did you learn from some very specific copy-paste? Mention explicitly some of them.**

* **Redis Caching**: Learned how to configure Redis with Spring Boot and use it to cache frequently accessed data, significantly reducing database load.