CODE REVIEW

TRAINEE: IBRAHIM SAFIANU

What Was Done Right

1. Your use of DTO validation demonstrate solid understanding of input sanitization.
2. The centralized shared module is a smart approach of preventing code duplication and follows the DRY principle.
3. You did a great job documenting of your code, however it should cut across your entire code base.
4. You did a good job writing robust and maintainable tests.
5. Good project structure and segregation of concerns was on point.

Areas for Improvement

1. Consider separation of concerns when using User model and UserDetails, this approach works however it does not follow best practices.
2. Using Java Records for your request and response DTOs can improve data immutability.
3. Commented blocks of code in codebase impacts readability.
4. Use a simple and consistent DTO for carrying all responses to the consumer, this ensures consistency across endpoints.
5. Use LoggerFactory rather than the generic (System.out.println) for logging.
6. Avoid using @Transactional on GET methods unless lazy loading is required.
7. Use proper API versioning in your application.
8. End to end tests should cover edge cases and other flows.

Overall Assessment

This was a well-structured micro-services application following best practices, your codebase reflects the following strengths:

i. Strong Foundation in building robust applications.
ii. Good Practices: You provided a clean code structure with well-considered security practices.
iii. Scalable Design.

Solid enterprise grade application, however applying the above stated improvements could potentially move this code to the next level.

In all, this was a good work done, keep it up and do not forget to consider suggested improvements.