Next Steps for Branko

Topics for Continued Development as an Engineer

Spring Boot Related:

- **Testing**
  - AAA in Testing (LinkedIn post, Medium article)
  - Mockito (link)
  - JUnit (link)
  - Unit Tests (what are unit tests and why are they important)
  - Integration Tests (what are integration tests and why are they important)
- **Functional Programming** (we discussed this during the internship)
  - Map, Filter, Reduce
  - Functional Interfaces in Java (advanced) - link
- **Optional**
- Specifications for Search and Filter
- Pagination
- Spring Security (covered in internship)

General Principles:

- **Logging** (Investigate what logging is and its importance)
  - Useful link for logging in Spring Boot - link
- **Clean Code Principles** (we implemented some of these, but here is a reminder):
  - Clean Code (a very famous book) - link

- Clean Code in Java - link
- SOLID - link
- DRY - link
- If there is a logical whole, separate it into a method
- Pay attention to naming; it should be clear what a method/class does
- A class/method should do only one thing; if it does multiple things, it could probably be separated into another class/method
- Refactor your code multiple times. Clean code requires time, and there is a rule of three in refactoring, which states that your code will probably be clean after the third refactoring iteration - link. We should revisit and think about the naming and structure of our code.
- **Methods**
  - Make methods small and clear.
  - Make them do only one thing.
  - Don’t pass too many parameters to the method (if possible, no more than three). If there are more, store those parameters in some other utility class.

Database:

- **Optimistic/Pessimistic Locking**
- **Transactions in Databases**
- **Types of Databases:**
  - Relational (SQL databases)
  - NoSQL Databases (they don’t have relations) - link
- **Relational Databases**
  - Primary Key
  - Foreign Key
- **Database Indexes and Their Types**

- **Database Triggers**

After Learning Spring Boot:

When you finish with Spring Boot and general principles, you can focus on other technologies. I recommend these:

- **Docker** (very important)
- **CI/CD** (important) - what it is, and investigate some tools like GitHub Actions
- **Kafka** (important)
- **Redis**

Advanced:

- **Kubernetes** (advanced and very important)
- **Cloud Technologies** - AWS, Azure (nice to know; will be useful for conventional projects)
- **ELK Stack**