**Assignment 1 SpringSocial - Report: Architecture, Challenges, and Lessons Learned**

**Team Size:** 2 **Team Members:** Casey Hsu, Matthew Price  
**Project Name:** GBC\_SpringSocial-68

**Objective**

The objective of our first assignment was to build the foundational structure of the microservices-based social media platform named "SpringSocial." This involved creating a social media platform with key features such as user management, creating posts and commenting.

**Environment Setup and Basic RESTful Service**

**Environment Setup**

We began by setting up the necessary development environment and tools, which included:

* **Java**: Installed and configured Java to work with Spring Boot.
* **Gradle**: Used Gradle as our build tool for managing project dependencies.
* **Docker / Docker Desktop**: Utilized Docker for containerization, enabling us to run our services consistently across different environments.
* **MongoDB (docker container)**: Deployed MongoDB in a Docker container, allowing flexibility in data schema for the Post entity.
* **Postgres (docker contianer)**: Deployed Postgres in a Docker container, allowing for a relational database to be implemented for the User and Comment entities
* **Spring Boot**: Leveraged Spring Boot as the primary framework for building microservices.

**Challenges Faced**

1. **Database Integration**: Integrating MongoDB for Post entities and JPA for User and Comment entities required an understanding of data modeling and persistence.
2. **Coordinating services:** Managing the interaction and coordination between microservices.
3. **Docker Data:** Bootstraping the data into the database was more difficult than expected.

**Lessons Learned**

1. **Modular Microservices**: We gained valuable experience in building modular microservices that can be scaled independently, facilitating future platform expansion.
2. **Data Modeling**: Understanding different database systems, including relational and NoSQL databases, is essential for designing a flexible and robust data layer.
3. **Containerization Proficiency**: Mastering Docker and Docker Compose is crucial for maintaining a consistent and portable application environment.
4. **Testing as a Priority**: Implementing thorough testing, including integration and real-time testing with containers, is integral to ensuring the reliability and correctness of microservices.
5. **Docker init Data**: how to properly add data to a database on initialization. Init scripts, sequencing of events, and how to connect to the proper Databases

In conclusion, our work on the "SpringSocial" project has been a valuable learning experience. We've successfully laid the foundation for a microservices-based social media platform, overcoming various challenges and acquiring essential skills along the way. This report provides a high-level overview of our progress, with more detailed documentation available for reference.