Annotated Bibliography

Jungho Park

October 13, 2025

References

[1] Kamran Ahmed. Backend developer roadmap, Sep 2025.

This website provides a recommended order for learning backend web development concepts. The road map gives a well structured guideline of which areas within the backend development are essential and need to be focused on. Furthermore, the roadmap includes a summary of each concept to give an overview of the topics, and also links to resources that would help the learning process. The roadmap will be used to explore the area of interest and identify the skills that are needed to complete the project, and also a great way to find other resources for the project.

[2] Tom Barker. *Intelligent caching.* O'Reilly Media, Inc, 2017.

This book will be utilized to have an high level concept around how to utilize caching to protect the infrastructure from an enormous scale. While this book does not introduce all caching strategies, it gives an in-depth review of content delivery network(CDN) and frontend scaling strategies. As caching has significant effects on the performance of the application, this book will be used to understand and leverage the CDN (chapter 2), learn internal caching rules (chapter3), and how to avoid common problems when designing caching systems (chapter 4).

[3] Domenico Bianculli, Sartaj Hassan, Vasilios Andrikopoulos, Cesare Pautasso, Tommi Mikkonen, Jennifer Perez, Tomas Bures, Martina De Sanctis, Henry Muccini, Elena Navarro, Mohamed Soliman, and Uwe Zdun. Assessing the impact of asynchronous communication on resilience and robustness: A comparative study of microservice and monolithic architectures. In *Software Architecture. ECSA 2025 Tracks and Workshops*, pages 171–186, Cham, 2026. Springer Nature Switzerland.

This research article explains the differences between a distributed architecture(using asynchronous communication) and a monolithic architecture. The research question in the article are beneficial to learn an indepth understanding of how system architecture has an effect in case of failure, system robustness,

such as error rates and response times during fault conditions, and tradeoffs including complexity and resource usage. This article will be used to further investigate and aid the understanding gained from the Head First Software Architecture.

[4] Raju Gandhi, Mark Richards, and Neal Ford. *Head first software architecture*. O'Reilly, 2024.

This is a textbook that introduces different types of architectures with great examples and explanations. Chapter 5, Architectural Styles will be helpful to identify the two main types of software architecture monolithic and distributed. The brief introduction and explanation of both of these types will be beneficial. Chapters 6 through 8 focuses deeply on monolithic applications, and chapters 10 through 11 focuses on the distributed architecture. These chapters will be beneficial when learning deeply about specific architectural patterns applied in backend web development.

[5] Jan L. Harrington. *Relational database design and implementation*. Morgan Kaufmann/Elsevier, 2016.

This is a textbook that dives deeper into relational databases. The chapters that would be helpful to my project will be the ones related to normalization forms and increasing performance of the database. Chapter 7 introduces what normalization is in database systems and normalization forms. The concepts are well organized and provides examples that are easy to understand, which will help me when deciding the normalization form for the project. Moreover, chapter 8 contains how to design and tune database to increase performance, which will be a critical aspect when designing the database for the project. This textbook will be mostly used for researching normalization forms and applying those theories.

[6] Martin Kleppmann. *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. O'Reilly, 2017.

This textbook introduces how to design data-intensive applications. The textbook emphasizes the three aspects, reliability, scalability, and maintainability, that is essential when designing a database system. The author also explains strategies to increase the performance of the database when scaling databases, such as replication and partitioning, introduced at chapters 5 and 6, respectively. These chapters will provide an understanding of the concepts that are essential when creating a database that has the potential to scale in the future.

[7] Amazon Web Services. Relational vs nonrelational databases - difference between types of databases - aws, n.d. What's the Difference Between Relational and Non-relational Databases?

This document provided by AWS explains what relational and non-relational databases are. It is a good starting point for researching the two main types of databases. The article clearly explains the concepts and provides comparisons between the two types. The included tables help concisely understand the key differences. It also provides external links to other AWS documents for deeper learning.

[8] Alexi Turcotte, Mark W. Aldrich, and Frank Tip. reformulator: Automated refactoring of the n+1 problem in database-backed applications. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ASE '22, New York, NY, USA, 2023. Association for Computing Machinery.

Although this article is mostly focused on the code refactoring to solve the N+1 Problem, it is a great explanation of the N+1 problem and why it occurs. This is specifically contained in the abstract and the introduction section. This article will be helpful to understand the N+1 problem that is common in database queries which has high relation to the performance of the application. Furthermore, the solution that the article focuses on could be a potential research topic after understanding the common solutions for the N+1 problem.