Database & Caching Interview Questions

1. Explain the concept of normalization in the context of frontend development and its importance in data management.
2. How does data normalization contribute to a more efficient and maintainable frontend application?
3. Describe the purpose of HTTP caching in a web application and its impact on performance.
4. What are the common HTTP headers related to caching, and how are they used to control caching behaviour?
5. What is a service worker, and how can it be utilized for caching in a frontend application?
6. Discuss the advantages and challenges of using service workers for caching compared to traditional browser caching.
7. How can you implement caching strategies for API calls in a frontend application?
8. Explain the role of cache invalidation and cache expiration in API caching.
9. Compare and contrast local component state with global state management in a frontend application.
10. What are the benefits and drawbacks of using a state management library/framework (e.g., Redux, Vuex) in a frontend project?
11. What is LocalStorage, and how does it differ from other client-side storage options?
12. Discuss scenarios where LocalStorage is suitable for storing data in a frontend application.
13. Explain the purpose of Session Storage and how it differs from LocalStorage.
14. In what situations would you choose Session Storage over other storage options?
15. Describe how cookies are used for storage in a web application.
16. What are the security considerations when working with cookies, and how can you enhance their security?
17. What is IndexedDB, and how does it enable client-side storage in a web application?

18. Discuss scenarios where IndexedDB is preferable over other client-side storage options.
19. Compare and contrast the data structures available in Local Storage and IndexedDB.
20. What are the size limits of Local Storage and IndexedDB?
21. In a real-world scenario, how would you approach integrating normalization, HTTP caching, service worker caching, API caching, state management, LocalStorage, Session Storage, Cookie Storage, and IndexedDB to create a cohesive and efficient frontend architecture?