Here is the high-level overview of the storage and caching landscape from a frontend perspective.

---

1. Client-Side Storage (The "Database")

In the browser, we choose our storage mechanism based on the data's complexity and the required persistence.

- **IndexedDB:** The "heavy lifter." A non-relational, key-value database that allows for large amounts of structured data. It's asynchronous and supports indexes, making it ideal for offline-first apps or caching large API responses.
- **LocalStorage / SessionStorage:** Simple, synchronous key-value pairs (strings only). Limited to ~5MB. Best for simple preferences (e.g., "Dark Mode") or auth tokens.
- **Cookies:** Small 4KB, sent with every HTTP request. Used primarily for session management and security (via HttpOnly and SameSite flags).

---

2. The Browser Caching Layers

Caching on the frontend happens at multiple levels of the network stack to prevent unnecessary round-trips to the server.

- **Memory Cache (L1):** Short-lived, volatile storage (JS variables, Redux/Zustand state). Extremely fast but cleared on page refresh.
- **Service Worker Cache (Cache API):** Programmatic control over network requests. This allows us to intercept requests and serve assets or data even when the user is offline.
- **HTTP Cache (Browser Cache):** Controlled via headers (Cache-Control, ETag). This determines how the browser reuses files like JS, CSS, and images.

---

3. Data Management & State Sync

This is where the "logic" of your frontend database lives.

- **Normalized State:** Treating your frontend store like a database (e.g., using byId and allIds patterns) to avoid data duplication.
- **Stale-While-Revalidate (SWR):** A caching strategy where you show cached data immediately, then fetch fresh data in the background to update the UI.
- **Optimistic UI:** Updating the "client-side database" before the server confirms the change to make the app feel instantaneous.

Summary Comparison

Storage Type	Capacity	Lifetime	Use Case
IndexedDB	Large (GBs)	Persistent	Offline data, complex objects
Cache API	Large (GBs)	Persistent	PWA assets, Offline assets
LocalStorage	~5MB	Persistent	User settings, Auth state
In-Memory	RAM	Session	Current UI state, Form data