**Firebase research answers – Generated by ChatGPT 4o**

**1. Firebase Authentication Methods (Short Summary)**

* **Email/Password**: Familiar and secure, good for apps with account-based access. Pros: recoverable, built-in support. Cons: requires signup flows and management.
* **Anonymous**: Great for guest access or trials. Pros: fast, simple. Cons: not persistent, can't identify user unless upgraded.
* **OAuth Providers (Google, Facebook, Apple, etc.)**: Best for social or public-facing apps. Pros: fast and verified login. Cons: setup complexity with third-party APIs.
* **Phone Number**: Useful for mobile-first apps. Pros: easy for users without email. Cons: rate limits, requires SMS handling.
* **Custom Tokens**: For enterprise or external systems. Pros: full control. Cons: complex setup, backend needed.

**2. Firebase Realtime Database vs Firestore vs MongoDB**

* **Realtime Database**: Ideal for small, real-time apps like chats or quick prototypes. Pros: simple, fast syncing. Cons: flat structure, limited querying.
* **Firestore**: Best for scalable apps needing structured data and strong querying. Pros: documents/collections, better filters. Cons: higher complexity, pricing.
* **MongoDB**: Great for full-control backends and custom APIs. Pros: powerful queries, flexible schema. Cons: no built-in real-time sync, needs more setup.

**When to Use:**

* Use Realtime DB for fast, low-complexity sync apps.
* Use Firestore for structured cloud apps with moderate scale.
* Use MongoDB for full backend control or large-scale systems.

**3. Firebase Realtime Database: REST API vs SDK**

* **REST API**: Best for backend services or environments without Firebase SDK support (e.g., Python). Pros: simple HTTP requests, works in any language, respects security rules. Cons: no real-time sync, no offline support, more manual setup.
* **SDK**: Best for mobile/web apps built with supported platforms (e.g., JavaScript, Flutter). Pros: built-in real-time listeners, offline persistence, easier data binding. Cons: limited to supported languages, requires client-side integration.

**When to Use:**

* Use REST API for server-side scripts or Python apps.
* Use SDK for client apps needing live updates and native support.