ADR

**Title:**

Decision of Backend

**Context + Problem Statement:**

We need a reliable and secure backend for the application, and to make sure it is compatible with client requirements such as a large storage.

**Decision Driver:**

* The organization estimates a 10% growth yearly, which means the backend needs to be able to handle an increased amount of users and media data.
* Ensure that features the client wants can be implemented using the chosen backend and are compatible with the rest of the stack and architecture.

**Option:**

Firebase, SQLite, MongoDB, Oracle, MySQL, Cloud, Express, Node.js

**Considered Options:**

* Firebase: Firebase is a cloud-based database, and it helps to build web-based applications and mobile applications such as Flutter, C++, Unity, the Web, etc. (Configr Technologies, 2024).
* MySQL: is a relational database, inside it data is structured and efficiently managed, it is a widely used and many industries use it as its an open source (Domantas, 2018).

**Decision Outcomes:**

Chosen option: Firebase, because it is most compatible with our design and stack decisions. It uses services that would be beneficial, such as authentication, hosting, and real-time database.

**Consequences:**

Good, because it provides tools needed for features we would implement, such as authentication and real-time.

Good, because it facilitates javascript which is widely used and could enable more manageable maintainability.

Good, because it can maintain the application performance even when it needs to handle a heavy load of data. (Configr Technologies, 2024)

Bad, because it may raise restrictions in the real-time or firestore database in the future as it may not be able to hold as much data as it would for an SQL-based database.

Bad, because it has limited data migration capabilities, making it difficult to switch to another database and move it over if needed.

Bad, because it's non-relational and may make it more difficult to use than if it were a relational database.

**Confirmation:**

Before starting the architecture design.

**Pros and Cons:**

Pros: Can handle real-time updates for the inventory database.

Cons: Scalability is limited? Can’t handle the complexity as a traditional data management provider?

Pros:

* Web developers can focus on the frontend because the Firebase database platform takes care of backend development, which is convenient for the web developer. (Clark, 2022)
* Firebase database has the cloud-base database and real-time database, which are popular and provide robust capabilities. (Sureka, 2022)
* Firebase supports the authentication service, and it accepts email, phone number, and password verification. (Sureka, 2022)

Cons:

* Pricing is usage-based, which can be unpredictable without proper monitoring.
* No built-in tools for moving data between databases
* The maximum number of index queries per document is 40,000 (Firebase, 2025). This means the scalability is limited.

# Resources

Domantas, G. (2018, December 14). What is MySQL: MySQL Explained for Beginners. Hostinger Tutorials. <https://www.hostinger.co.uk/tutorials/what-is-mysql>

Configr Technologies. (2024, February 23). *Is Firebase a Viable Solution in 2024? A*

*Comprehensive Assessment for Developers*. Medium; Medium.

[https://configr.medium.com/is-firebase-a-viable-solution-in-2024-a-comprehensive-assessment-f](https://configr.medium.com/is-firebase-a-viable-solution-in-2024-a-comprehensive-assessment-for-developers-23ef4889e8d9)

[or-developers-23ef4889e8d9](https://configr.medium.com/is-firebase-a-viable-solution-in-2024-a-comprehensive-assessment-for-developers-23ef4889e8d9)

Clark, J. (2022, September 8). *Firebase Advantages and Disadvantages*. Back4app.

<https://blog.back4app.com/firebase-advantages-and-disadvantages/>

Sureka, S. (2022). *Should Professionals Use Firebase? Pros and Cons | Pangea.ai*. Pangea.ai.

<https://pangea.ai/resources/should-professionals-use-firebase-pros-and-cons>

Firebase. (2025). *Index types in Cloud Firestore | Firebase*. Firebase.

<https://firebase.google.com/docs/firestore/query-data/index-overview#index_entries>