PERFORMANCE and FUNCTIONALITY REPORT

Team Name: API MAFIAS

INTRODUCTION

This report presents the performance and functionality evaluation of VAULTER, a secure API key management application. Vaulter enables developers to safely store, encrypt, and manage API keys using AES-256 encryption and Clerk-based authentication. The purpose of this report is to analyze the system's responsiveness, reliability, and usability under real-world scenarios to ensure optimal user experience and security compliance.

SYSTEM OVERVIEW

Vaulter is a full-stack web application built using:

- **Frontend**: React, Vite, and TailwindCSS
- **Backend**: Next.js API Routes
- **Database**: Supabase
- **Authentication**: Clerk (JWT-based)
- **Encryption**: AES-256 for secure API key storage
- **Deployment**: Vercel

The application ensures that no API key is ever stored in plaintext. All keys are encrypted before storing at the backend and can only be decrypted by authorized users.

FUNCTIONALITY REPORT

Functionality	Description	Test Performed	Result	Status
User Authentication	Login via Email, Google, GitHub using Clerk	Tested all methods	All successful	✅
Add New API Key	Adds encrypted key to DB	Added 5 keys	Keys stored securely	✅
View Dashboard	Displays stats: total keys, recent keys, tags	Loaded instantly	Data accurate	✅
Copy Key	Copies masked key to clipboard	Tested with 3 keys	Works instantly	✅

Delete Key	Removes key from database	Deleted 2 keys	Deleted successfully	✅
Search/Filter	Filters by tag	Tested multiple tags	Fast results	✅
Masked Display	API keys hidden by default	Verified	Secure	✅

PERFORMANCE EVALUATION

Performance Metric	**Description**	**Tool Used**	**Result**	**Observation**
Page Load Time	Dashboard load after login	Lighthouse	1.3 sec	Fast
API Response Time	/api/keys (GET request)	Postman	180 ms	Excellent
Add Key Request	/api/keys (POST)	Postman	200 ms	Acceptable
Database Write	Encrypted key insertion	Supabase logs	<100 ms	Efficient
Memory Usage	During CRUD operations	DevTools	210 MB	Normal

SECURITY EVALUATION

Vaulter implements multi-layered security through:

- AES-256 encryption for all stored API keys
- JWT-based authentication via Clerk
- Environment-protected encryption keys
- Supabase role-based access control (RLS)
- Masked display of sensitive data on frontend

These measures ensure that even if the database is compromised, API keys remain encrypted and inaccessible.

CONCLUSION

The testing and evaluation results confirm that Vaulter is a strong, secure, and high-performing tool for managing API keys. It meets all the functional requirements, offering a quick, user-friendly interface along with solid data encryption. Future updates can focus on enhancing analytics features and introducing team-based functionality.