

Final Project Report (Living Document)

Project Deliverable

Course B1A1

Class:

Author: François LANGE

Date: 2026/06/19

Table of Contents

- What Has Been Done Page 2
- What Needs To Be Done (Day 2 Operations) Page 2
- What Is The Next Step Page 2

The current version is #ident
 "@(#)\$Format:LocalFoodAI_lanfr144:generate_docs.py:Francois
 Lange:lanfr144@school.lu:2026/06/16 21:48:22:Francois
 Lange:lanfr144@school.lu:2026/06/16
 21:48:22:2a8ed056889f3b796f9266feda591b12b72f3b96:HEAD -> main, origin/main:\$"

Final Project Report (Living Document)

What Has Been Done

- **Core Architecture:** Deployed a resilient 8-container local fallback Docker Compose stack (MySQL, Streamlit UI, local Ollama LLM, anonymous SearXNG search, secure Nginx proxy, and local Zabbix Server/Web/Agent observability suite).
- **Database Optimization:** Successfully loaded OpenFoodFacts records and utilized advanced vertical partitioning and FULLTEXT indices.
- **Clinical Subquery Strategy:** Refactored the core Pandas/SQL query pipeline to use subquery limiting, resolving Cartesian join explosions and reducing query latency to ~0.04s.
- **Monitoring & Security:** Nginx securely proxies traffic on Port 80. Zabbix actively monitors proxy and server health, dynamically handling SNMP/alert loops in local/offline fallback mode.
- **Git Versioning:** Implemented Git `.gitattributes` to push #ident

```
"@(#)$Format:LocalFoodAI_lanfr144:generate_docs.py:Francois
Lange:lanfr144@school.lu:2026/06/16 21:48:22:Francois
Lange:lanfr144@school.lu:2026/06/16
21:48:22:2a8ed056889f3b796f9266feda591b12b72f3b96:HEAD -> main, origin/main:$"
```

 tracking directly into the Python Application UI.

What Needs To Be Done (Day 2 Operations)

- **SSL/TLS Certificates:** The Nginx proxy is functional on HTTP port 80. Port 443 (HTTPS) must be configured with a Let's Encrypt certificate for true production encryption.
- **User Acceptance Testing (UAT):** Clinical dietitians should rigorously test the AI Chat constraints and Plate Builder to ensure edge cases are handled safely.
- **Advanced Rate Limiting:** Limit the number of AI requests per user using a sliding window algorithm in `app.py`.

What Is The Next Step

- Execute the `data_sync.sh` cron job monthly.
- Maintain the automated `backup_db.sh` 7-day retention cycle.
- Begin the hand-off to the operational team for Phase 2 feature requests.

References

- **OpenFoodFacts Dataset & API Catalog:** Detailed food ingredients database. (<https://>

world.openfoodfacts.org/)

- **Ollama Local LLM Inference Engine:** Lightweight instruction-following llama3.2 runtimes. (<https://ollama.com/>)
- **Zabbix Enterprise Telemetry and Monitoring:** System health and performance logging. (<https://www.zabbix.com/>)

Index

- **AI:** Page 2, Page 4
- **MySQL:** Page 2, Page 4
- **Zabbix:** Page 2, Page 3, Page 4
- **Docker:** Page 2, Page 4
- **Streamlit:** Page 2, Page 4
- **Nginx:** Page 2, Page 4
- **RAG:** Page 4
- **Allergens:** Page 4
- **Vitamins:** Page 4
- **Minerals:** Page 4
- **Clinical:** Page 2, Page 4
- **WSL:** Page 4
- **Ollama:** Page 2, Page 3, Page 4
- **LLM:** Page 2, Page 3, Page 4
- **Database:** Page 2, Page 4
- **Security:** Page 2, Page 4
- **Telemetry:** Page 3, Page 4
- **Backup:** Page 2, Page 4
- **Firewall:** Page 4
- **SMTP:** Page 4