



Multilingual Customer Feedback Analyzer v1.0

⚠️ **Time Expectation:** This project is designed to be completed within approximately 1 week.

Aim for a working, demonstrable solution rather than perfection. You may prioritize core features and leave some optional parts if time is tight.



Functional Requirements

Build a tool for collecting and analyzing customer feedback across languages. The app must:

- Accept text reviews in multiple languages
 - Use Gemini Studio to detect language, translate, and classify sentiment
 - Show dashboard of feedback trends (e.g., % positive)
 - Allow admins to search or filter by product or language
-



Stack Boundaries

- **Backend:** FastAPI (REST + OpenAPI)
 - **Frontend:** React SPA
 - **Database:** PostgreSQL or MongoDB
 - **AI Integration:** Gemini Studio
 - **Containerization:** Docker Compose setup required
-



Expected Endpoints (Example)

- POST /api/feedback – Submit feedback
 - GET /api/feedback – View entries
 - GET /api/stats – Show sentiment overview
 - POST /api/translate – Translate input (Gemini)
-



Data Examples

```
// User input
{ "text": "Ce produit est excellent!" }

// Gemini AI response
{
  "translated_text": "This product is excellent!",
  "sentiment": "positive"
}
```



Docker Compose Structure

- Frontend (React)
- Backend (FastAPI)
- Database (PostgreSQL or MongoDB)



Data Persistence Expectations

- Saved itineraries persist via volumes.

Performance Considerations (Optional)

- Async loading indicators for LLM calls
- Paginated API responses if large datasets returned

Bonus (Optional)

- Unit tests
- Responsive design
- CI/CD pipeline

Deliverable Expectations

Submit:

1. A fully working Docker Compose project
2. Your own README.md with:
 - Short project summary
 - Getting started steps
 - How to run it
 - API routes and usage
 - Frontend + backend overview
 - Data schema
 - Gemini Studio integration
 - Any limitations or known issues