**TDA-16 Startprompt**

**Context**  
Ik werk aan de Turkish Diaspora App, een AI-gedreven locatieplatform dat automatisch Turkse ondernemingen en gemeenschapslocaties in Nederland ontdekt, classificeert en weergeeft.

Het project bestaat uit:  
– **Backend:** FastAPI (Python) met Supabase (PostgreSQL)  
– **Frontend:** React + Vite + Leaflet  
– **AI-laag:** OpenAI GPT-4.1-mini voor classificatie & verificatie  
– **Hosting:** Render (backend + cronbots) en Vercel (frontend)

Ik voeg de volgende bestanden toe als context, zodat je uitsluitend deze bronnen gebruikt voor informatie:

1. 📘 **The New Testament.docx** – het Master Plan met architectuur, data-model, AI-pipeline en epics.
2. 📙 **Backlog.docx** – alle epics, user stories en acceptance criteria t/m Epic 8.
3. 📗 **TDA13.docx** – Core Map View met Leaflet (frontend)
4. 📗 **TDA14.docx** – Alleen Turkse zaken tonen (AI-gebaseerde filtering, frontend + backend integratie)  
   (optioneel kun je ook TDA11.docx bijvoegen als referentie voor de AI-classificatieservice, mocht TDA-16 daarop voortbouwen).

TDA-16: MonitorBot & Freshness Policy Engine  
Implement the monitor\_bot.py worker. It should periodically select records where next\_check\_at is in the past, based on defined Freshness Policy. It then re-triggers verification process for these records.

Acceptance Criteria

Bovenkant formulier

* Bot correctly calculates next\_check\_at for new records based on confidence and status
* Bot processes records in batches (e.g., MONITOR\_MAX\_PER\_RUN=200)
* After run, no record in database has next\_check\_at older than 90 days
* Changes to location's business\_status detected and updated

Onderkant formulier

Definition of Done

Bovenkant formulier

* MonitorBot operational
* Freshness policy implemented and working
* Automatic re-verification functioning
* Data quality maintained over time

Onderkant formulier

Technical Notes

Bovenkant formulier

This is the "heartbeat" of the application, ensuring long-term data qualityOnderkant formulier

**Instructies voor jou (ChatGPT):**

* Gebruik uitsluitend informatie uit de bijgevoegde bestanden.
* Als iets niet in deze context staat, vraag om verduidelijking — **geen aannames of hallucinaties.**
* Lever alle code als **volledige bestanden**, klaar om te plaatsen in /Backend/app/... of /Frontend/src/....
* Houd dezelfde schrijfstijl, structuur en toon aan als in de bestaande TDA-documenten.
* Gebruik Pydantic v2-stijl, async SQLAlchemy, en React + TypeScript component-stijl.
* Sluit af met een korte sectie “Next Steps” die suggereert welke user story logisch volgt op TDA-16.

**Outputformat:**  
1 **Analyse & Doelstelling** – wat deze story doet, waarom het nodig is, en welke onderdelen geraakt worden  
2️ **Technische Deliverables** – bestandsstructuur, code, configuratie  
3️ **Test & Validatie** – hoe ik het lokaal run en verwachte output  
4️ **Jira Comment (Final)** – in dezelfde format als TDA-14 met DoD + Acceptance Criteria

**Extra context:**  
Het project bevindt zich momenteel aan het einde van Epic 4 (Frontend Map & List Application).  
De volgende epics op de roadmap zijn:  
– **Epic 5:** Continuous Monitoring & Data Freshness  
– **Epic 6:** Admin CMS & Human-in-the-Loop  
– **Epic 7:** Observability & Security Hardening  
– **Epic 8:** Pilot Deployment & Roll-out

Houd daar rekening mee bij het bepalen van de scope van TDA-16: het moet logisch aansluiten op deze fasering.

**Te uploaden bijlagen:**

The New Testament.docx

Backlog.docx

TDA13.docx

TDA14.docx

(opt: TDA11.docx)