Volledig uitgewerkt plan voor Optie B (RAG: Retrieval-Augmented Generation)

1) Doelen & scope

Doelen:

- Vragen stellen en relevante verzen + antwoord krijgen.
- Citaties naar bronverzen tonen.
- Ondersteunen van meerdere vertalingen.
- Privacy & kosten laag houden.

Niet-doelen:

- Geen volledige theologische discussie-agent.

2) Licentie & vertalingen

Publiek domein:

- Statenvertaling
- King James Version

Niet publiek domein:

- HSV, NBV, etc.

3) Architectuur

Front-end: Chat UI, dropdown vertaling, resultatenpaneel.

Back-end: Node.js service (/embed, /ask).

Database: PostgreSQL + pgvector (aanbevolen).

LLM: GPT-4.1 / 4o / Groq.

4) Datamodel & chunking

1 vers = 1 document.

Document bevat id, translation, book, chapter, verse, text, embedding.

5) Prompting

System prompt: Assistent beantwoordt alleen obv context.

User prompt: vraag + context.

6) Implementatiestappen

1. Data normaliseren.

2. Database & pgvector inrichten.

3. Embedding pipeline maken.

4. Retrieval endpoint bouwen.

5. WordPress plugin integreren.

6. Styling & caching.

7) Beveiliging

- API keys server-side

- Rate limiting

- CORS restrictions

8) Kosteninschatting

- Embeddings: €10–€25 eenmalig

- Runtime per vraag: €0.002–€0.02

9) Performance & caching

- Query cache

- Streaming

- Warm start

10) Testplan

Functionele tests, accuracy checks, regressietests.

11) Uitrol

- Dev, staging, prod

- .env configuratie

- Docker compose optioneel

12) Roadmap

Dag 1: Data, pgvector, embeddings, endpoint.

Dag 2: WP plugin, styling, caching.

Later: Meertaligheid, extra vertalingen.