LLM-based smart recommendation engine for Vedaz that analyzes user chat history and profile to recommend astrologers.

1.) LLM Stack:

For a pure embedding-based recommendation engine, I'd go with an open-source SentenceTransformer model (e.g. all-MiniLM-L6-v2 from Hugging Face) rather than a full chat-style LLM.

- Cost-effective: No per-token API fees—once you host the model, embedding every new query is free aside from your infra cost.
- Low latency: Embedding on a modest CPU server typically takes < 5 ms per request, compared to 50–100 ms for managed APIs.
- Lightweight footprint: The model is under 100 MB on disk and runs easily on a 2 vCPU machine, avoiding the need for expensive GPUs.
- High semantic quality: Despite its size, MiniLM-L6 delivers embeddings that rival larger models for short-text similarity, making it ideal for matching chat transcripts to astrologer profiles.

2.) Hosting and Scaling

- Package the SentenceTransformer model behind a lightweight FastAPI or Flask endpoint in Docker.
- Using vector database, If you outgrow memory, migrate to a managed vector DB (Pinecone, Weaviate) with sharding
- Front with an Application Load Balancer (ALB) or API Gateway.

3.) Cost for hosting for 50k users:

Assuming each user triggers 3 recommendation calls per month (150 000 embeddings total):

- Compute: t3.medium @ $0.0416/hr → $30 /mo
- Storage & network egress: negligible for small payloads (< $5/mo)
- Ops overhead: zero additional license or API fees

Total cost approximately will be $35 per month for 50k monthly active users

4.) Safety and safety

- Role-based IAM for admin/developer versus inference endpoints.
- Embeddings can memorize sensitive phrases so to avoid that I would use hash identifiers before embeddings.
- Rate-limit per-user embedding calls.