Requirements for astrological predictions:

- \*\*Date of Birth\*\*

- \*\*Time of Birth\*\*

- \*\*Place of Birth\*\*

- \*\*Location and timezone\*\*

- \*\*Today's date (for creating transit chart)\*\*

- \*\*Lagna Chart\*\*

- \*\*Rasi Chart (D-1 Chart)\*\*

- \*\*Navamsa Chart (D-9 Chart)\*\*

- \*\*Dasamsa Chart (D-10 Chart)\*\*

- \*\*Transit Chart\*\*

- \*\*Partner's Birth Data (for relationship scenarios)\*\*

- \*\*Child's Birth Data (for parenting scenarios)\*\*

- \*\*Transit Calculations (to predict change is dasha)\*\*

- \*\*Panchang data - Tithi, Nakṣatra, Yoga, Karaṇa(for muhurta, compatibility, and event timing)\*\*

- \*\*Vimśottarī Daśā table (and optionally Yogini, Chara, Kalachakra, etc.) (for sequence of planetary periods and sub-periods for the native)\*\*

- \*\*Saptāṁśa (D7)(children & creativity)\*\*

- \*\*Shadbala, Ashtakavarga(to explain why a planet acts in a certain way during dashā/transit)\*\*

**Recommended low-cost architecture for your astrology agent**

* **User profile & PII, birth details, preferences, feature flags** → **Firestore** (documents per user).
* **Computed charts (D1, D9, D10, transit snapshots, panchāṅga, dashā tables)** → store as **JSON docs in Firestore** (denormalized so you can fetch what you need in 1–2 reads).
* **Large artifacts** (rendered PNG/PDF reports, exported data) → **Cloud Storage** (store URL in Firestore).
* **Embeddings for RAG** → start with **Cloud SQL Postgres (small instance) + pgvector** *or* a single **Qdrant** container on a small VM. Move to managed vector service only when usage grows.
* **Caching**: cache “today’s transit” and frequently requested interpretations in memory (Cloud Run/VM) or in **Redis (MemoryStore)** if needed to cut Firestore reads.