PROJOMANIA · DEVELOPMENT

Tech Stack Decision Matrix

How we pick stacks for custom builds. It is deliberately opinionated — the goal is a defensible decision, not a vendor bakeoff.

Dimension A — Project fit

- Does the stack match the workload? (Real-time !’ event-driven; content-heavy !’ SSG; data-heavy !’ strong ORM ecosystem.)
- Are the performance SLOs achievable without heroic engineering?
- Does the stack offer primitives for the tricky parts, or will you rebuild them?

Dimension B — Team capability

- Does the team have production experience with the stack?
- If not, is training cost included in the schedule?
- Who owns the system in year 2 — us, you, or a handover?

Dimension C — Ecosystem & longevity

- Is the core project healthy (releases, contributors, security responsiveness)?
- Are first-party libraries available for the hard problems (auth, payments, search)?
- Is there commercial support available if needed?

Dimension D — Total cost of ownership

- Hosting cost at 1×, 10×, and 100× current traffic.
- License cost (open-source is not always free — dual-licensed projects, gated features).
- Talent market: can you hire for it in your region at your budget?

Dimension E — Risk & lock-in

- Single-vendor lock-in risk (proprietary DBs, proprietary runtimes).
- Migration cost away from the stack in 3 years.
- Data portability: can you export everything in standard formats?

Default recommendations we tend to reach for

- Content-heavy marketing: Astro + React islands + headless CMS.
- Transactional web app: Next.js + PostgreSQL + Redis; Node.js or Python API.
- Mobile: React Native if teams are web-native; Flutter if not. Native only when the app demands native APIs.
- Data platform: Python + FastAPI + PostgreSQL; ClickHouse for analytics, Airflow for orchestration.

- Enterprise line-of-business: Odoo if the requirements map to ERP/CRM modules; custom otherwise.

This matrix is our starting point for every custom-development consultation. Bring yours to the call and we'll stress-test it together.