**Maano AI — Full Project Documentation**

**Tagline:** *Where Wisdom Meets Technology*

**1) Executive Summary**

**Maano AI** is a learning platform that brings multiple AI models—(e.g., ChatGPT, Claude, Gemini, Qwen etc)—into a single, safe, and student‑friendly interface. It helps learners pick the right AI for each task, compare answers side‑by‑side, and build AI literacy while supporting teachers with classroom tools, analytics, and content safeguards.

**Initial focus:** Secondary school and university students.

**Core value:** Access + AI literacy + safety + local language support.

**2) Goals & Non‑Goals**

**Goals**

* Provide one hub for multiple AI models with quick switching and side‑by‑side comparison.
* Teach students *when* to use *which* AI via a **Know Your AI** guide and smart recommendations.
* Support English and Zambian languages for inclusivity.
* Ensure school‑ready safety: content filtering, auditing, role‑based controls.
* Work in low‑connectivity contexts with offline/lite caching.

**Non‑Goals (v1)**

* Building our own foundation model.
* Replacing teachers; instead, we augment and support them.
* Full LMS replacement (basic classroom features only at first).

**3) Target Users & Personas**

* **Student (Secondary/University)**: Needs explanations, practice, references, and project help.
* **Teacher/Lecturer**: Curates prompts, monitors usage, assigns tasks, reviews analytics.
* **School Admin**: Manages org, billing, policies, integrations (SSO), compliance.
* **Parent/Guardian (optional add‑on)**: Receives progress summaries and usage alerts.

**4) Feature Set (v1 → v2)**

**v1 Must‑Haves**

1. **Multi‑AI Access** (Chat interface per model + switcher)
2. **AI Explorer** (ask multiple AIs at once; side‑by‑side answers)
3. **Know Your AI** (cards: strengths, weaknesses, best tasks, examples)
4. **Smart Recommender** (suggest best AI based on intent)
5. **Subject Tutors** (Math, Science, Coding, Languages)
6. **History & Bookmarks** (per user)
7. **Safety & Moderation** (filters, rate limits, role policies)
8. **Accounts & RBAC** (Student/Teacher/Admin)

**v2 Nice‑to‑Haves**

* **Classrooms/Groups** with assignments and shared prompts
* **Offline Lite** (cached lessons, prompt packs, glossary)
* **Pronunciation & TTS** for languages
* **Plagiarism checker and citation builder**
* **Mobile app (React Native/Flutter)**

**5) System Architecture**

**5.1 High‑Level Diagram**

flowchart TD

subgraph Client[Clients]

W[Web App (React/Tailwind)]

M[Mobile App (React Native/Flutter) - v2]

end

subgraph Edge[Edge & CDN]

CDN[CDN/Asset Cache]

end

subgraph API[Backend API]

G[API Gateway / BFF]

A[Auth Service (JWT/OAuth, SSO)]

P[Policy & Moderation]

R[Routing Engine (Model Selector)]

C[Conversation Service]

U[User & Org Service]

L[Logging & Analytics]

end

subgraph Ext[External AI Providers]

OAI[OpenAI]

ANT[Anthropic]

GOO[Google Gemini]

QWN[Qwen]

end

subgraph Data[Data Stores]

PG[(PostgreSQL)]

MG[(MongoDB - transcripts)]

RS[(Redis - cache/queues)]

OB[(Object Storage - uploads/exports)]

end

W -->|HTTPS| CDN --> G

M -->|HTTPS| CDN

G --> A

G --> P

G --> C

G --> U

G --> R

C --> MG

U --> PG

R -->|provider API calls| OAI

R --> ANT

R --> GOO

R --> QWN

P --> RS

L --> OB

L --> PG

**5.2 Sequence — Multi‑AI Comparison**

sequenceDiagram

participant User

participant Web as Web App

participant API as API Gateway

participant Route as Routing Engine

participant OAI as OpenAI

participant ANT as Anthropic

User->>Web: Enter question + choose Explorer

Web->>API: POST /compare {question, models:[gpt, claude,...]}

API->>Route: fan‑out requests per model

Route->>OAI: call model A

Route->>ANT: call model B

OAI-->>Route: response A

ANT-->>Route: response B

Route-->>API: aggregate & normalize

API-->>Web: JSON array of answers + metadata

Web-->>User: side‑by‑side UI with tags & citations

**6) Technology Choices**

* **Frontend:** React + Vite, TailwindCSS, Zustand/Redux Toolkit, shadcn/ui.
* **Mobile (v2):** React Native or Flutter.
* **Backend:** Node.js + Express/Fastify **or** Python + FastAPI. (Pick one stack for v1; Node pairs nicely with React.)
* **Database:** PostgreSQL (users/orgs/policies), MongoDB (conversations), Redis (cache/queues), S3‑compatible storage.
* **Auth:** JWT (access/refresh), optional SSO (Google, Microsoft 365 for schools).
* **CI/CD:** GitHub Actions → Docker → Render/Railway/AWS.
* **Observability:** OpenTelemetry, Loki/ELK, Grafana.

**7) Data Model (v1)**

classDiagram

class User {

uuid id

string email

string name

enum role (student|teacher|admin)

uuid orgId

jsonb settings

timestamptz createdAt

}

class Organization {

uuid id

string name

jsonb policy

timestamptz createdAt

}

class Conversation {

uuid id

uuid userId

string model

json transcriptJson

timestamptz createdAt

}

class Message {

uuid id

uuid convId

enum sender(user|assistant|system)

text content

json meta

timestamptz createdAt

}

class ModelProfile {

string key

string displayName

string provider

string strengths[]

string weaknesses[]

string tags[]

json config

}

User --> Organization

Conversation --> User

Message --> Conversation

**8) API Design (sample)**

**Auth**

* POST /auth/register — create account
* POST /auth/login — get tokens
* POST /auth/refresh — rotate tokens

**Conversations**

* POST /chat — send message {model, prompt, contextId}
* POST /compare — send once → fan‐out to models ["gpt-4o", "claude-3.5", "gemini-1.5", "qwen-max"]
* GET /history — list user conversations

**Catalog**

* GET /models — list enabled models with tags/strengths
* GET /subjects — list tutor modes (Math, Science, etc.)

**Admin**

* GET /org/policy | PUT /org/policy — set filters, hours, limits
* GET /analytics/usage — org usage, cost, safety flags

**Errors** use RFC 7807 style with type, title, status, detail.

**9) Model Routing & Recommendations**

* **Intent Detection:** lightweight classifier over prompt (regex + small model) → intents: coding, math, creative, summary, research, translation.
* **Rules Engine:** map intents to preferred models (configurable per org). Example:
  + creative → GPT default; long\_summary → Claude; web\_research → Gemini; multilingual → Qwen.
* **Fallbacks:** timeouts, error budgets; retry different provider.
* **Side‑by‑Side Normalization:** unify JSON output (text, safety flags, tokens, duration, cost estimate).

**10) Safety, Privacy & Compliance**

* **Moderation pipeline:**
  1. Pre‑screen prompt → blocklists, PII patterns, age‑sensitive filters.
  2. Provider safety settings (e.g., harmful content switches).
  3. Post‑screen responses → redact flagged items.
  4. Rate limits & quotas per role.
* **Data handling:**
  1. Store transcripts in MongoDB with per‑org encryption at rest.
  2. Do not use student data for model training; opt‑out flags for providers.
  3. Tokenized logs; access via RBAC only.
* **Compliance targets:**
  1. COPPA/child safety principles (for under‑13 if supported later),
  2. GDPR‑like rights (export/delete),
  3. Local school data policies.

**11) Internationalization (i18n) & Local Languages**

* Content packs for Bemba, Nyanja, Tonga, Lozi (progressively add).
* Prompt templates with language toggle.
* TTS/ASR integration (v2) for pronunciation practice.

**12) Accessibility (a11y)**

* WCAG 2.2 AA: keyboard navigation, ARIA labels, color contrast, captions for audio.
* Dyslexia‑friendly mode (OpenDyslexic/spacing), adjustable reading level.

**13) Analytics & Metrics**

* **Learning metrics:** time on task, question diversity, model variety.
* **Outcome metrics:** quiz improvement, assignment quality (teacher rubric input).
* **System metrics:** latency per provider, cost per session, safety incidents.

**14) Deployment & Environments**

* **Envs:** Dev → Staging → Production.
* **Secrets:** .env managed by Vault/Parameter Store.
* **CI/CD:** Lint → Test → Build → Docker → Deploy; blue/green or canary.
* **Infra:** Container platform (Render/Railway/AWS ECS), managed Postgres, MongoDB Atlas, Redis.

**15) Cost Planning (rough, adjustable)**

* Provider API costs (per 1K tokens) × projected usage.
* Hosting (DB + compute + storage) per month.
* Observability and CDN.
* School pricing model: *per seat* or *per org flat rate*; free tier for pilots.

**16) Roadmap & Milestones**

**Month 0–1: Foundation**

* UX flows, design system, data model, auth, basic chat with single model.

**Month 2–3: Multi‑Model + Safety**

* Integrate 3–4 providers, moderation, history, Know Your AI, Explorer.

**Month 4: Pilot**

* 2–3 schools/universities, analytics, policy controls, refine recommender.

**Month 5+:**

* Mobile app, offline packs, TTS/ASR, classroom assignments, plagiarism/citation tools.

**17) Test Plan (high level)**

* **Unit:** routing, parsers, safety filters, RBAC.
* **Integration:** API↔provider calls (mocks for rate/cost), DB migrations.
* **E2E:** login→ask→compare→save→export; teacher policies.
* **Security:** authZ bypass attempts, rate‑limit, SSRF on provider calls.
* **Load:** 95p latency < 2.5s on compare with 2 models.

**18) UX Notes & Screens (spec)**

* **Home/Dashboard:** Quick actions: Ask, Compare, Subject Tutors.
* **Chat Screen:** Model switcher, safety status, export (PDF/Docx), reading level toggle.
* **Explorer (Compare):** Two to four columns, per‑model tags, vote which answer helped.
* **Know Your AI:** Cards with strengths/limits, examples, badges (#CreativeWriting #Coding #MathHelp #Multilingual).
* **Teacher Console:** Policies (filters/time windows), prompt packs, analytics overview.

**19) Branding Quick Spec**

* Colors: Deep Blue #1B365D, Gold #FFD54F, Soft Gray #F4F4F4, Navy #0E2747.
* Typography: Headings — Poppins; Body — Inter/Nunito.
* Logo concept delivered separately (Wisdom Brain).

**20) Sample Prompts & Templates**

* **Explain:** "Explain photosynthesis to a Grade 11 student. Use simple steps and a real‑life analogy."
* **Compare:** "Give three different thesis statements about climate change for a university essay."
* **Code Tutor:** "Explain this Python error and fix it: TypeError: unsupported operand type(s)".
* **Bilingual:** "Translate this paragraph to Bemba and highlight key vocabulary."

**21) Risk Register (selected)**

* Provider policy changes → **Mitigation:** pluggable providers + fallbacks.
* Cost overruns → **Mitigation:** caps, caching, educator hours.
* Safety incidents → **Mitigation:** layered moderation + human review workflow.
* Low connectivity → **Mitigation:** offline packs + SMS/USSD micro‑features (v2 idea).

**22) Success Criteria (KPIs)**

* 80% of pilot students report increased understanding/confidence.
* 20% improvement on formative quizzes after 4 weeks.
* <2% safety incident rate per 1,000 interactions.
* Org renewal rate > 70% after pilot.

**23) Implementation Checklist (v1)**

* Design system & components (shadcn/ui + Tailwind)
* Auth (JWT) + RBAC
* Providers: OpenAI, Anthropic, Gemini, Qwen
* Chat + History
* Explorer (side‑by‑side)
* Know Your AI (catalog)
* Moderation pipeline
* Org policies & analytics (MVP)
* Docs & teacher guides

**24) Appendix A — Example .env (backend)**

PORT=8080

DATABASE\_URL=postgres://user:pass@host:5432/maano

MONGODB\_URI=mongodb+srv://...

REDIS\_URL=redis://...

JWT\_SECRET=change\_me

OPENAI\_API\_KEY=...

ANTHROPIC\_API\_KEY=...

GEMINI\_API\_KEY=...

QWEN\_API\_KEY=...

ALLOWED\_ORIGINS=https://app.maano.ai,https://staging.maano.ai

PROVIDER\_TIME\_LIMIT\_MS=25000

**25) Appendix B — Sample Endpoint (Node/Express)**

// POST /compare

app.post('/compare', authMiddleware, async (req, res) => {

const { prompt, models } = req.body;

// validate

if (!prompt || !Array.isArray(models) || models.length === 0) {

return res.status(400).json({ error: 'prompt and models[] required' });

}

// fan‑out with Promise.allSettled

const results = await Promise.allSettled(models.map(m => callProvider(m, prompt)));

const normalized = results.map((r, i) => ({

model: models[i],

ok: r.status === 'fulfilled',

output: r.status === 'fulfilled' ? r.value.text : null,

meta: r.status === 'fulfilled' ? r.value.meta : { error: String(r.reason) }

}));

res.json({ prompt, answers: normalized });

});

*End of document.*