rozmoWA (Duolingo-like App)

Background

Goal. Teach **Polish kids and adults** English in a fun, game-like app with daily bite-sized lessons, pronunciation practice, and motivating streaks—optimized for PL→EN learners.

Why now. Poland has strong demand for English proficiency for work, travel, and education. Modern ASR/TTS and on-device personalization enable affordable, high-quality speaking and listening practice tuned for Polish accents.

Initial scope (confirmed/updated).

- **Audience:** Two tracks — **Kids (7–15)** with parental controls & simplified UI; **Adults (16+)** with full features.
- **Content domain: English for Polish speakers**, CEFR **A0–B1** to start, with a placement test and adaptive path.
- **Platforms at launch: iOS, Android, and Web** (responsive PWA) with shared backend and content.
- **Pedagogy:** Spaced Repetition (SM-2 variant), mastery thresholds (≥90% per skill), micro-lessons (3–5 min), deliberate practice for speaking/listening; interleaved grammar + vocabulary.
- **Key features:** Polish UI & hints; bilingual explanations; speech recognition for English with Polish-accent tolerance; TTS for models/dialogues; skill tree (unit→lesson→exercise); daily streaks, XP, leagues, hearts; offline lesson packs; **parent/teacher dashboards**; AI tutor chat (safe-mode) for guided practice.
- **Business model:** Freemium in PL; subscription removes ads, unlocks unlimited review + AI tutor; **B2B(2C) school licenses** with classroom management.
- **Regions & compliance: Poland-first** launch; GDPR with **parental consent for users under 16** (Poland); data minimization; WCAG 2.1 AA accessibility; kid-safe defaults for the junior track.
- **Success metrics:** D1/7/30 retention, weekly sessions/user, lesson completion rate, **A0→A1 time-to-mastery**, pronunciation score uplift, conversion to paid, school renewal rate.

(If this Background looks good, I'll draft the Requirements next.)

Requirements

MoSCoW Prioritization

Must Have (MVP) - **Polish-localized UX**: full PL UI, hints/explanations in PL; English content with PL scaffolding. - **Accounts & parental controls**: email/Apple/Google sign-in; **child accounts** with verified parent; age-gating (≤16 requires consent in PL); anonymous play → soft wall. - **Placement test (A0–A2)**: adaptive entry using quick vocab, phonics, and listening probes; CEFR tagging. - **Skill tree A0–A2**: Units→Lessons→Exercises; per-skill mastery ≥90% to advance; review queue (SRS, SM-2 variant). - **Exercise types**: multiple-choice, tap-to-order, type-the-answer, listen-and-select, **listen-and-type**, **speak (ASR)**, image-prompt vocab, drag-match, fill-in-the-blank. - **Audio**: high-quality TTS for prompts/dialogues; recorded native clips for minimal pairs; **accent-tolerant ASR** with profanity filtering. - **Gamification**: daily

streaks, XP, hearts/lives, streak freeze; push notifications and email reminders (opt-in). - **Offline support (mobile)**: downloadable lesson packs with background sync; conflict resolution on re-connect. - **Responsive Web (PWA)**: same content & progress; limited offline (Cache API) for last-visited lessons. - **Content tooling**: internal CMS for units/lessons/exercises with CEFR and phoneme tags; preview & A/B flags. - **Analytics & A/B**: event schema for funnel (placement→first lesson→return), cohort retention, difficulty heatmaps. - **Payments**: subscriptions via **Apple IAP**, **Google Play Billing**, **Stripe** for Web; entitlements sync across platforms. - **Privacy & compliance (GDPR)**: parental consent records, data minimization, export/delete (DSAR), telemetry opt-in; COPPA-like kid-safe defaults for junior track. - **Security**: oauth flows, rate limiting, WAF/CDN, at-rest (AES-256) and in-transit (TLS 1.2+) encryption; least-privilege roles for CMS. - **Accessibility**: WCAG 2.1 AA; captions on audio, adjustable font sizes, color-contrast, **dyslexia-friendly font toggle**. - **Basic educator/parent dashboards (pilot)**: view progress by unit, time-on-task, pronunciation attempts; roster upload via CSV.

Should Have (v1.x) - **AI tutor chat (safe mode)**: constrained topics, guardrails; role-play scenarios; explain-my-mistake in Polish. - **Adaptive hints**: error-pattern-based tips; targeted review injection. - **Pronunciation scoring**: phoneme-level feedback and practice loops (e.g., minimal pairs /ɪ/ vs /iː/). - **Social**: friend leaderboard, weekly quests, leagues; classroom “assignments” for pilot schools. - **Teacher tools (pilot+)**: simple assignment builder; printable homework; parent invite codes. - **Content localization**: Polish cultural references and names; holiday themes.

Could Have (backlog) - Home screen widgets; lock-screen streak; seasonal events; AR mini-games for kids; browser extension for quick review; referral program.

Won’t Have (now) - UGC content marketplace; advanced SIS/LMS integrations beyond CSV; C1+ proficiency; multi-base-language support; live 1:1 tutoring marketplace.

Non-Functional Requirements

- **Performance**: p95 API < **300 ms** (EU region); client input-to-feedback < **120 ms** for local checks, < **600 ms** for ASR.
- **Availability**: **99.9%** monthly; graceful degradation when ASR/TTS is unavailable.
- **Scalability**: design for **50k MAU** MVP, path to **500k MAU**; multi-tenant for school pilots.
- **Platforms**: iOS **15+**, Android **8+**, evergreen browsers (last 2 versions of Chrome/Edge/Safari/Firefox).
- **Data retention**: raw audio $\leq$ **30 days** (training disabled by default), analytics events 18 months; opt-in for model improvement.
- **Internationalization**: full i18n framework; right-to-left readiness for future.

(If these Requirements look good, I’ll proceed to the Method next.)

Method

System Architecture (High Level)

- **Clients**:
- **iOS (Swift/SwiftUI)** using `SFSpeechRecognizer` for on-device ASR; `AVSpeechSynthesizer` for TTS; background downloads for lesson packs.

- **Android (Kotlin/Jetpack Compose)** using `SpeechRecognizer` for on-device ASR; `TextToSpeech` for TTS.
- **Web (Next.js/React PWA)**: Web Speech **Synthesis** API for TTS; **cloud ASR only** (limited browser support for recognition); mic capture → secure websocket to backend ASR proxy.
- **Backend**:
- **API**: NestJS (Node 20+), GraphQL (queries) + REST (uploads), WebSockets for live ASR; hosted in EU (e.g., Azure West Europe/Poland Central).
- **Services**:
    - **Content & Progress** (PostgreSQL 15) via Prisma ORM.
    - **Cache/Queues**: Redis (rate limits, SRS due queues, websocket presence).
    - **Search**: Postgres full-text for now; optional Meilisearch later.
    - **ASR/TTS Proxy**: routes to device/on-prem/cloud; integrates with **Azure Speech** for fallback + pronunciation assessment; caches TTS mp3/ogg.
    - **Analytics & A/B**: segment-like event collector → BigQuery (EU) or Postgres warehouse; feature flags.
- **CMS (internal)**: Next.js Admin + NestJS admin APIs; role-based auth; versioned content; preview.
- **CDN/Storage**: Cloud storage (Azure Blob S3-compatible) for audio/images; CDN for media and lesson packs.
- **Auth**: Cognito/Auth0-like or custom OAuth2/OIDC with Apple/Google/Email; parental-consent flow.
- **Payments**: Apple IAP, Google Play Billing, Stripe (PL methods incl. cards, BLIK) — entitlements service reconciles cross-platform.

Component Diagram (PlantUML)

```
@startuml
skinparam componentStyle rectangle
actor User
node "iOS App" as iOS
node "Android App" as Android
node "Web PWA" as Web
node "CDN" as CDN
node "API Gateway (NestJS)" as API
node "ASR/TTS Proxy" as Speech
database "PostgreSQL" as PG
queue "Redis" as Redis
node "Analytics Ingest" as Analytics
node "CMS Admin" as CMS

User --> iOS
User --> Android
User --> Web

iOS --> API : GraphQL/REST
Android --> API : GraphQL/REST
Web --> API : GraphQL/REST

Web --> Speech : WS (audio)
```

```
iOS --> Speech : WS (fallback)
Android --> Speech : WS (fallback)

API --> PG
API --> Redis
API --> CDN
Speech --> CDN : cache TTS
Speech --> external "Azure Speech" : ASR/TTS fallback
API --> Analytics : events
CMS --> API : admin APIs
CMS --> CDN : upload media
@enduml
```

Data Model (Core Tables)

```
-- users & auth
users(id uuid pk, email text unique, locale text default 'pl-PL', dob date
null, child boolean default false,
      created_at timestamptz, updated_at timestamptz)
parents(id uuid pk, user_id uuid fk users, verified boolean, consent_record
jsonb)
identities(id uuid pk, user_id uuid fk users, provider text, subject text
unique)

-- content
units(id uuid pk, cefr text check (cefr in ('A0','A1','A2','B1')), order int,
title_pl text, title_en text)
lessons(id uuid pk, unit_id uuid fk units, order int, title_pl text, title_en
text, est_minutes int)
exercises(id uuid pk, lesson_id uuid fk lessons, type text, prompt jsonb,
answer jsonb,
          skills text[], phonemes text[], difficulty int, tags text[])
assets(id uuid pk, url text, kind text, md5 text, locale text)

-- progress & SRS
attempts(id uuid pk, user_id uuid fk users, exercise_id uuid fk exercises, ts
timestamptz, correct boolean,
         latency_ms int, details jsonb)
mastery(user_id uuid fk users, skill text, rating float, last_seen timestamptz)
srs_queue(user_id uuid, exercise_id uuid, due_at timestamptz, interval_days
int, ease float,
          primary key(user_id, exercise_id))

-- gamification
streaks(user_id uuid, current_days int, best_days int, last_active date)
xp_ledger(id uuid pk, user_id uuid, source text, delta int, ts timestamptz)
```

```
-- schools (pilot)
classes(id uuid pk, owner_user_id uuid fk users, name text)
class_enrollments(class_id uuid fk classes, student_user_id uuid fk users,
primary key(class_id, student_user_id))
assignments(id uuid pk, class_id uuid fk classes, lesson_id uuid fk lessons,
due_date date)

-- commerce
entitlements(user_id uuid, product text, status text, expires_at timestamptz)
payment_events(id uuid pk, platform text, payload jsonb, ts timestamptz)
```

Content Schema (Exercise JSON)

```
{
  "type": "listen_and_repeat",
  "prompt": { "en": "She sells seashells.", "slow": true, "audio_asset_id":
"..." },
  "scoring": { "target_phonemes": ["ʃ","iː","s","ɛ","l","z"], "min_score":
0.75 },
  "hints_pl": ["Zwróć uwagę na 'sh' (ʃ)."],
  "tags": ["minimal_pair","/ɪ/ vs /iː/"],
  "cefr": "A1"
}
```

ASR/TTS Decision Flow

1) **Mobile (default)**: try on-device ASR for ≤10s utterances; if confidence < threshold or unsupported, stream to cloud. 2) **Web**: capture mic → stream to **ASR Proxy → Azure Speech**; return interim results for live feedback. 3) **Pronunciation scoring**: for speaking drills, always run cloud **pronunciation assessment** on final recording to get phoneme-level feedback; store scores but not raw audio beyond retention window. 4) **TTS**: prefer on-device; pre-generate cloud neural TTS for long dialogs; cache in CDN.

SRS (SM-2 variant) for Exercise Scheduling

```
for each exercise_attempt:
  q = quality_from_attempt(correct, latency_ms)  # 0..5
  if first_time: interval=1, ease=2.5
  else if q < 3: interval=1
  else:
    interval = previous_interval * ease
    ease = ease + (0.1 - (5 - q) * (0.08 + (5 - q) * 0.02))
    ease = max(1.3, ease)
```

```
due_at = now + interval_days
upsert into srs_queue(exercise_id, user_id, due_at, interval_days, ease)
```

Lesson Progress & Mastery

- Each **skill** aggregates multiple exercises. Mastery increases with correct attempts weighted by difficulty and recency.
- Gate progression when mastery < 0.9; inject targeted review from SRS queue.

Offline & Sync

- **Mobile**: lesson pack ZIP (JSON + audio) per unit; versioned with hash; client keeps LRU of 3 packs; sync via ETag.
- Conflict policy: client sends attempt UUIDs; server de-dupes by primary key; idempotent writes.

Security & Privacy

- All media signed URLs (short TTL). Row-level security for user-scoped reads. Kid accounts: telemetry minimal; **audio retained ≤30 days**; DSAR export.
- Rate limits per IP/token; WAF at edge; JWT with rotating keys; CMS roles (admin, editor, reviewer).

Analytics (key events)

- `placement.start|finish`, `lesson.start|complete`, `exercise.attempt`, `srs.review`, `streak.day`, `purchase.success`, `asr.fallback`.

Competitive Notes (for validation)

- Duolingo-style **skill tree & hearts**; Babbel focuses on dialogues; ELSA Speak excels at **pronunciation scoring**—we combine tree + **phoneme-level feedback** tuned for Polish learners.

```
@startuml
skinparam monochrome true
actor Learner
participant Client
participant "API" as API
participant "ASR Proxy" as ASR
participant "Azure Speech" as Cloud
Learner -> Client : speak utterance
Client -> Client : on-device ASR
Client --> API : POST attempt (if confident)
Client -> ASR : WS stream (fallback)
ASR -> Cloud : stream audio
Cloud --> ASR : partial + final text + scores
ASR --> Client : transcript + score
Client --> API : POST attempt + assessment
API -> API : update mastery, SRS
```

```
API --> Client : next exercise
@enduml
```

(If this Method works for you, I'll move on to a step-by-step Implementation plan.)

Implementation

0) Team & Ways of Working

- **Team**: PM, UX/UI (1), Content Lead (1) + Authors (2), iOS (1), Android (1), Web (1), Backend (2), QA (1), DevOps/Sec (0.5), Data (0.5).
- **Cadence**: 2-week sprints; trunk-based dev; feature flags for all user-visible changes.
- **Environments**: `dev` → `staging` → `prod` (EU regions only). Data masking in non-prod.

1) Cloud & DevOps (Azure-first)

- **Regions**: Primary **Poland Central**, failover **West Europe**.
- **Compute**: Azure **Container Apps** for API, Speech Proxy, and CMS; autoscale by RPS and CPU.
- **DB**: Azure Database for **PostgreSQL Flexible Server** (v15), gp_Standard, HA enabled.
- **Cache/Queues**: **Azure Cache for Redis** (Standard) for SRS queue, sessions, rate limits.
- **Storage/CDN**: Azure **Blob Storage** for media + lesson packs, **Azure CDN** (Front Door) for edge.
- **Identity/Secrets**: **Azure AD B2C** (or Auth0) for OAuth2/OIDC; **Key Vault** for secrets & signing keys.
- **Observability**: **Application Insights** + Log Analytics; SLOs: API p95 <300 ms, Speech Proxy p95 <600 ms.
- **Push**: **Azure Notification Hubs** (brokers APNs & FCM) with locale-aware templates.
- **CI/CD**: GitHub Actions — build/test → containerize → deploy to Container Apps; Fastlane for iOS/TestFlight; Play Console CI for Android; Vercel-style preview builds for Web (or static export via Azure Static Web Apps for the PWA shell).
- **Security**: Front Door WAF, OWASP rules; rate-limit per IP/JWT; signed blob URLs; DB RLS for user-scoped rows; weekly dependency scans (Dependabot) and SAST.

2) Backend Services (NestJS)

- **APIs**
- **GraphQL** (Apollo): `viewer`, `units/lessons(expose metadata)`, `lesson(exercises paged)`, `progress`, `streak`, `srs.next`, `classroom` (pilot), `entitlements`.
- **REST**: `POST /attempts` (idempotent by UUID), `POST /asr/recordings` (for scoring), `GET /packs/:unitId` (signed), `POST /consent`, `POST /iap/validate`.
- **WS**: `/asr` duplex stream for fallback/live captions.
- **Services**: AuthN/Z (AAD B2C tokens), Payments (Stripe + IAP/Play), Content, Progress, SRS, Gamification, Classroom, Analytics Ingest.
- **Jobs**: SRS re-queue, streak rollovers (UTC midnight + user TZ), pack pre-generation, TTS pre-warm.

3) Speech/TTS Path

- **On-device first** on iOS/Android; capture confidence and timing locally.

- **Fallback** via Speech Proxy → **Azure Cognitive Services Speech** (EU endpoint) with **Pronunciation Assessment** for speaking drills.
- **Web**: mic → WS to Speech Proxy (no browser ASR). Return interim partials for live feedback.
- **Audio storage**: short-lived blob with SAS URL; auto-delete after 30 days via Lifecycle Management.

4) Content & CMS

- **Admin app** (Next.js + RBAC): unit/lesson/exercise CRUD, CEFR tagging, phoneme targets, hint authoring (PL), preview-as-learner.
- **Validation**: JSON schema linting; required TTS/recordings; profanity checks; min/max length per type.
- **Seeding A0–A2** (8–10 units): Polish-specific trouble spots: **th** /θ/ /ð/, **w/v**, **ship/sheep**, **ed** endings, **articles**, **prepositions**, basic **tenses**; common Polish-to-English error patterns drive hints.
- **Localization**: PL UI strings in i18n catalog; content versioning with dark launches behind flags.

5) Clients

- **iOS (SwiftUI)**: offline packs (BackgroundTasks), on-device ASR/TTS, lesson player with exercise plugins, purchase flow (StoreKit 2), consent/child flows, Notification permission warmup.
- **Android (Compose)**: offline packs (WorkManager), on-device ASR/TTS, BillingClient v7, parental flows.
- **Web PWA (Next.js/React)**: installable; Cache API for last pack; WS to Speech Proxy; Stripe checkout; same lesson player.

6) Data & Analytics

- **Event spec**: `placement.start/finish`, `exercise.attempt{type,latency,origin:device|cloud}`, `lesson.complete`, `srs.review`, `streak.tick`, `iap.success`, `stripe.success`.
- **Pipelines**: ingest → Postgres warehouse initially; dashboards in Metabase; cohort retention (D1/D7/D30), funnel, difficulty heatmaps, pronunciation score uplift.

7) Payments & Entitlements

- **Products**: `sub_monthly`, `sub_yearly` (intro price), `family_addon` (later); server validates Apple/Play receipts; Stripe for Web (PL cards + BLIK). Entitlements table is the only source of truth across platforms.

8) Privacy, Consent, and Safety

- **Age gate** at sign-up; ≤16 triggers parent email verification + consent capture (stored JSON with timestamp/IP).
- **Kid mode**: no social; restricted notifications; curated avatars; safe-mode AI tutor (when enabled) with blocked topics.
- **DSAR**: self-service export/delete; purge audio after 30 days; analytics opt-in toggle at first run.

9) Testing

- **Device matrix**: iOS 15–18 (A-series mix), Android 8–15 (mid/low-end), Chrome/Safari/Edge.

- **Speech**: latency budget tests (local <120 ms feedback; cloud <600 ms); accent corpus with Polish speakers; noisy environments.
- **Offline**: airplane-mode runs; sync conflict tests; power-loss during pack writes.
- **Compliance**: accessibility checks (WCAG), GDPR consent flows, purchase edge cases (refunds, grace periods).

10) Rollout Plan (Sprints ~2 weeks)

1. **S1**: Repo + CI/CD; DB schema; auth; skeleton apps; Hello World CMS.
2. **S2**: Content CRUD; lesson player v0 (MCQ, tap-order); streak/XP; event logging.
3. **S3**: On-device ASR/TTS; pronunciation drills (client-side); Azure Proxy MVP; packs download.
4. **S4**: Placement test; SRS queue + review mode; payments (Stripe + IAP/Play validation).
5. **S5**: A0 core units seeded; offline polish; dashboards (Metabase); Notification Hubs integration.
6. **S6**: Web PWA parity; consent/child flows; classroom pilot dashboards; beta with 200 users.
7. **S7 (Launch)**: Perf hardening; CDN warm; bug bash; public release in Poland.

11) Definition of Done (MVP)

- Users complete placement → first 3 lessons in <15 min on any platform.
- p95 Speech fallback <600 ms; crash-free sessions > 99.5%.
- A0 unit completion rate $\geq$ 55% in beta; D7 retention $\geq$ 20%.
- Payments working across iOS/Android/Web; entitlements sync within 5 min.

(Ready to move on to Milestones. I'll tailor dates and success gates next.)

Milestones

Timeline assumes 2-week sprints starting **Oct 20, 2025**. Adjust as needed.

M0 — Project Kickoff (Oct 20–Oct 31, 2025)

- Staff hired; repos + CI/CD live; Azure subscriptions, Front Door, Container Apps, Postgres, Redis provisioned.
- Definition of Done, coding standards, event taxonomy signed off.

M1 — Core Loop Alpha (Nov 3–Nov 28, 2025)

- iOS/Android/Web lesson player (MCQ, tap-order, fill-blank), streak/XP, placement v0.
- CMS CRUD + preview; A0 Unit 1–2 authored and playable.
- **Exit criteria:** 20 internal testers complete 3 lessons; p95 API < 300 ms.

M2 — Speech & Offline Beta (Dec 1–Dec 26, 2025)

- On-device ASR/TTS on mobile; Azure Speech fallback + pronunciation assessment; offline packs; Notification Hubs.
- **Exit criteria:** median ASR fallback latency < 600 ms; crash-free > 99%; 50 testers finish A0 Unit 1.

M3 — Payments & SRS (Jan 5–Jan 30, 2026)

- Stripe (PLN, incl. **BLIK**), Apple IAP, Google Play Billing; entitlements sync; SRS review mode; A0 Units 1–4.
- **Pricing SKUs:** PLN **29.99/mo**, **199.99/yr**; intro month **9.99**.
- **Exit criteria:** test purchases across platforms; entitlement reconciliation < 5 min; cohort D1 ≥ 35% (closed beta).

M4 — Content & Consent (Feb 2–Feb 27, 2026)

- A0 complete; A1 Units 1–2 drafted; child accounts + parental consent; accessibility pass (WCAG AA); dashboards v0.
- **Exit criteria:** 200 external beta users; A0 completion ≥ 55%; DSAR export works; parental consent flow verified.

M5 — Public Launch PL (Mar 2–Mar 27, 2026)

- Store listings (PL), marketing site, CDN warm; production analytics dashboards; customer support runbook.
- **Exit criteria:** Production launch in Poland; p95 API < 300 ms; Speech < 600 ms; crash-free > 99.5%.

M6 — School Pilot (Apr 6–May 1, 2026)

- Classroom: roster CSV, assignments, teacher dashboard; simple reporting exports.
- Onboard **2–3 partner schools**; collect teacher feedback.
- **Exit criteria:** 80% weekly active in pilot classes; teacher NPS ≥ 30.

M7 — Phase 2 Prep (May 4–May 29, 2026)

- A2 complete; begin **B1** authoring (license decision); AI Tutor (safe mode) technical spike; pricing review post-beta.
- **Exit criteria:** roadmap updated; paywall A/B results; retention improvements documented.

Risk Gates & Mitigations - ASR latency spikes → pre-buffer, shorter utterances, CDN edge for proxy, degrade to non-spoken variants. - Content velocity → daily content standups, template libraries, authoring QA checklists. - Payments rejects → sandbox parity tests, receipt edge-case harness, grace periods & retry logic.

(If these Milestones look right, I'll add the Gathering Results section next.)

Gathering Results

Beta Plan & Cohorts

- **Size:** 200 external users; **50% adults / 50% kids**; stratify by region (top PL cities) and device mix.
- **Cohorts:** placement level (A0 vs A1/A2), device (iOS/Android/Web), speech path (device vs cloud fallback).

Success Metrics (Launch KPIs)

- **Engagement/Retention:** D1 ≥ 35%, **D7 ≥ 20%**, D30 ≥ 10%; weekly sessions/user ≥ 3.
- **Learning Progress: A0 completion ≥ 55%** within 21 days; avg mastery per skill ≥ 0.9; pronunciation score uplift ≥ 10% from week 1→3.
- **Quality/Perf:** p95 API < 300 ms; speech fallback < 600 ms; crash-free > 99.5%; content issue rate < 1 per 100 lessons.
- **Monetization:** paywall CTR, trial start rate, conversion to paid (target 2–4% in first 60 days).

Instrumentation & Dashboards

- **Dashboards:** Cohort retention (D1/D7/D30), funnel (placement → first lesson → return), heatmaps by exercise type/difficulty, speech latency & fallback rate, A/B results (paywall, hinting).
- **Quality loops:** automated item analysis (discrimination index), flagged exercises for review, author QA queue.

Experiments (first 60 days)

- **Paywall timing**: after placement vs after first unit completion.
- **Hints**: static vs adaptive error-pattern hints.
- **Speech policy**: always-cloud scoring vs confidence-gated fallback threshold tuning.

Feedback & Support

- In-app quick surveys (Polish), NPS after 7 days, teacher feedback for pilots, support triage playbook (FAQs + crash/receipt triage).

Governance

- Weekly metrics review; post-launch “Red/Yellow/Green” status; rollback protocol for content and app releases.

Need Professional Help in Developing Your Architecture?

Please contact me at sammuti.com :)