## 1. System Overview & High-Level Architecture

A cloud-native, AI-driven language-learning platform with a modular frontend, RESTful backend, ML microservice, and managed database. All layers deploy on free-tier hosting.

- Client Layer: Web SPA (React), Future Mobile App (React Native/Expo)
- API Layer: Node.js/Express or Laravel 10+, Auth & billing services, Rate-limiting & logging
- ML Microservice: Python Flask/FastAPI, Hosts translation, pronunciation feedback, spaced-repetition
- Data Layer: PostgreSQL, Redis for cache & spaced-repetition scheduling
- DevOps & CDN: Vercel/Netlify for frontend, Heroku/Railway/Render for backend & ML, GitHub Actions for CI/CD, Docker Compose for local dev

## 2. Detailed Tech Stack

Frontend: React, Tailwind CSS, React Query
Backend: Node.js + Express or Laravel PHP
Database: PostgreSQL
Cache/Queue: Redis
ML/AI: Python, Hugging Face Transformers, FastAPI, PyTorch/TensorFlow
Auth & Payments: Auth0 or Firebase Auth, Stripe
Analytics: Google Analytics, Plausible
CI/CD: GitHub Actions
Containerization: Docker, Docker Compose

## 3. Data Model / Schema

Tables:

- users (id, email, password_hash, name, locale, plan, created_at, updated_at)
- lessons (id, language, level, title, content, created_at)
- quizzes (id, lesson_id, questions, created_at)
- progress (id, user_id, lesson_id, status, score, spaced_rep_queue, updated_at)
- billing (id, user_id, stripe_subscription_id, plan, status, amount, started_at, expires_at)

## 4. API Design

Endpoints:

- Auth: POST /api/register, POST /api/login, GET /api/me
- Lessons: GET /api/lessons, GET /api/lessons/:id, POST /api/progress, GET /api/progress
- Quizzes: GET /api/quizzes/:lesson_id, POST /api/quizzes/:id/submit
- Billing: POST /api/subscribe, POST /api/webhook/stripe

## 5. ML Pipeline

- Data Collection: Bilingual corpora, user-generated text, pronunciation recordings
- Preprocessing: Text tokenization, audio noise reduction & feature extraction
- Model Training: Hugging Face (T5-small, wav2vec2), fine-tuning on user corrections
- Evaluation: BLEU score, WER (Word Error Rate)
- Serving: FastAPI + Uvicorn, endpoints (/translate, /analyze_pronunciation, /recommend_review)
- Iteration: User feedback loop, nightly re-training

## 6. CI/CD & Infrastructure Setup

- GitHub Actions: Linting, testing, Docker builds
- Docker Compose: Local services (app, Postgres, Redis, ML service)
- Frontend Deployment: Vercel/Netlify
- Backend Deployment: Heroku/Railway/Render
- Database/Redis: Heroku or Railway add-ons
- Secrets: GitHub Secrets for keys & DB URLs

## 7. Deployment Roadmap & Sprint Plan (1 Week)

Day 1: Project scaffolding
Day 2: Auth module & APIs
Day 3: Lessons schema & endpoints
Day 4: Quiz engine
Day 5: Progress tracking, spaced-rep stub
Day 6: Billing integration, plan pages
Day 7: ML service stub, deploy & test

## 8. Monetization & Pricing Tier Breakdown

Free: Basic lessons, limited quizzes
Monthly: $9.99/mo, all lessons & quizzes, no ads
Quarterly: $24.99/quarter (17% discount)
Yearly: $79.99/year (33% discount, test-series vouchers)
Test Packs: $14.99 one-time, 4 mock tests

## 9. UI/UX Key User Flows & Wireframe Suggestions

- Onboarding: Collect goals, native & target language
- Dashboard: Progress bar, next lesson, weekly score
- Lesson Player: Text/audio toggle, pronunciation feedback
- Quiz Interface: MCQs, fill-in, audio recording
- Subscription Page: Plan comparison, payment form

## 10. Recommended Open-Source Tools & Free Hosting

- UI: React, Tailwind, React Query
- Backend: Laravel Breeze, Node.js
- ML: Hugging Face Transformers, SpeechBrain
- Hosting: Vercel, Netlify, Heroku, Railway, Render
- Analytics: Google Analytics, Plausible, Sentry
- Utilities: Redis, PostgreSQL