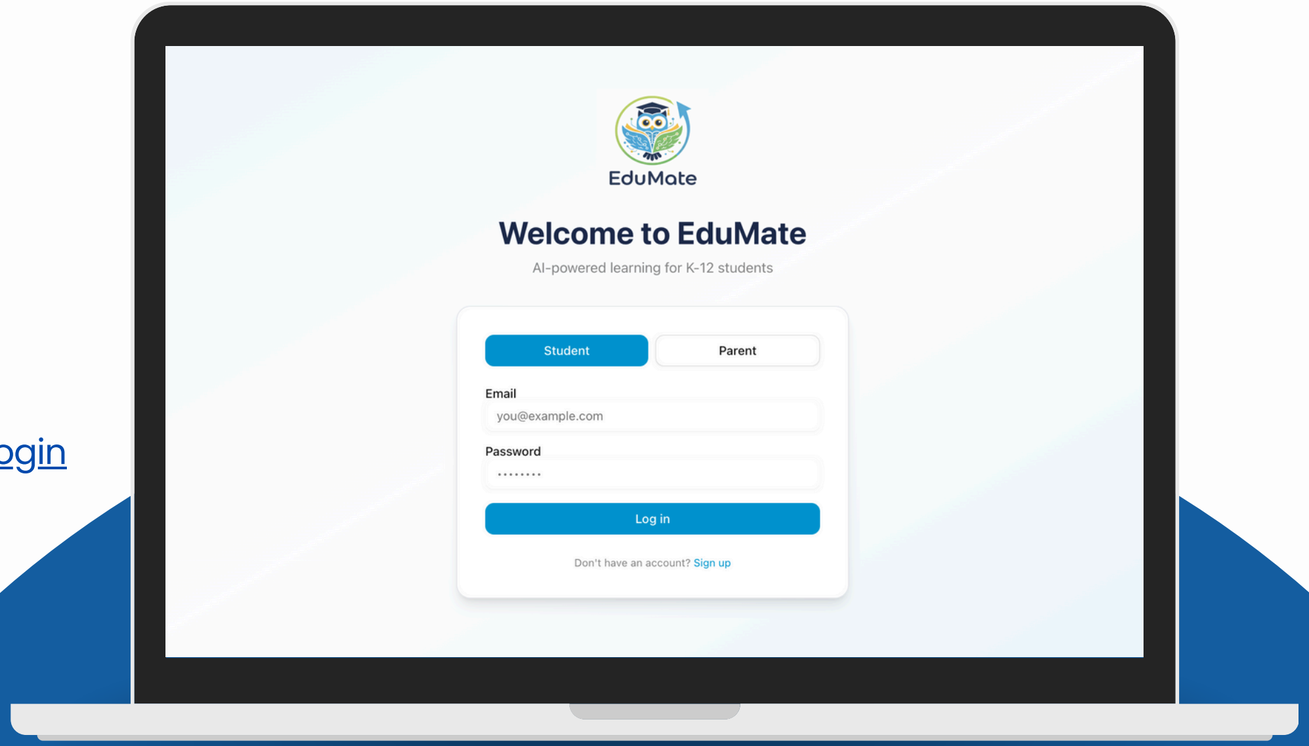
EduMate

EduMate

By: Aditya Kansara

Developed URL: https://project-check-point-1-edumate.vercel.app/login

Github URL: https://github.com/SALT-Lab-Human-AI/project-check-point-1-edumate.git



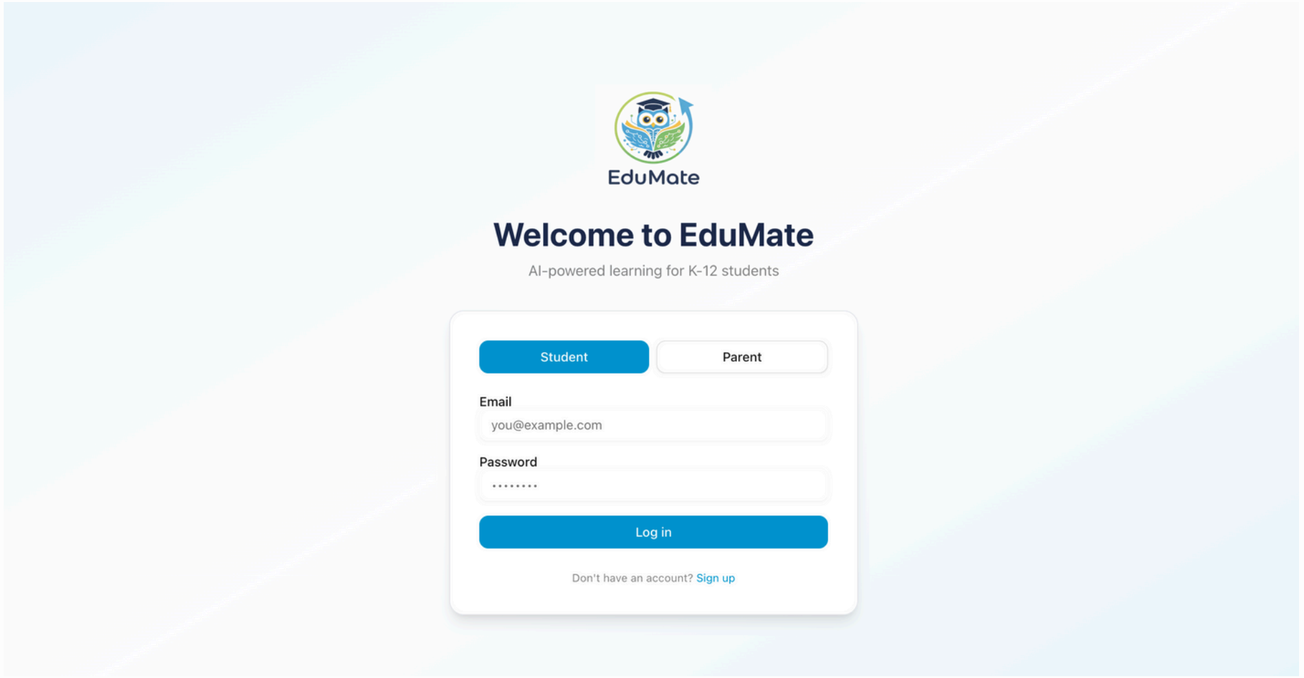
The Problem

- Students need instant, grade-level help outside class
- Generic AI tutors hallucinate and skip steps
- Parents lack visibility into learning progress
- Tutoring is expensive and hard to scale

Current AI tools provide answers, not understanding.

EduMate - K-12 Tutor

- RAG-grounded answers from curriculum-aligned knowledge base
- Step-by-step guidance (not just answers)
- Auto-graded quizzes with misconception-based distractors
- Parent dashboard with real-time progress tracking



Success Metrics

Time Saved:

- 24/7 instant access vs. waiting for tutors
- Auto-grading reduces manual work

Skills Enhanced:

- Structured problem-solving (5-phase approach)
- Error detection and corrective feedback
- Adaptive practice based on weaknesses

Hard-to-Get Information:

- Real-time parent insights (topic heatmaps, time-on-task)
- Skill-level breakdowns by concept

Measurable outcomes + parent trust = success.

APP DEMO

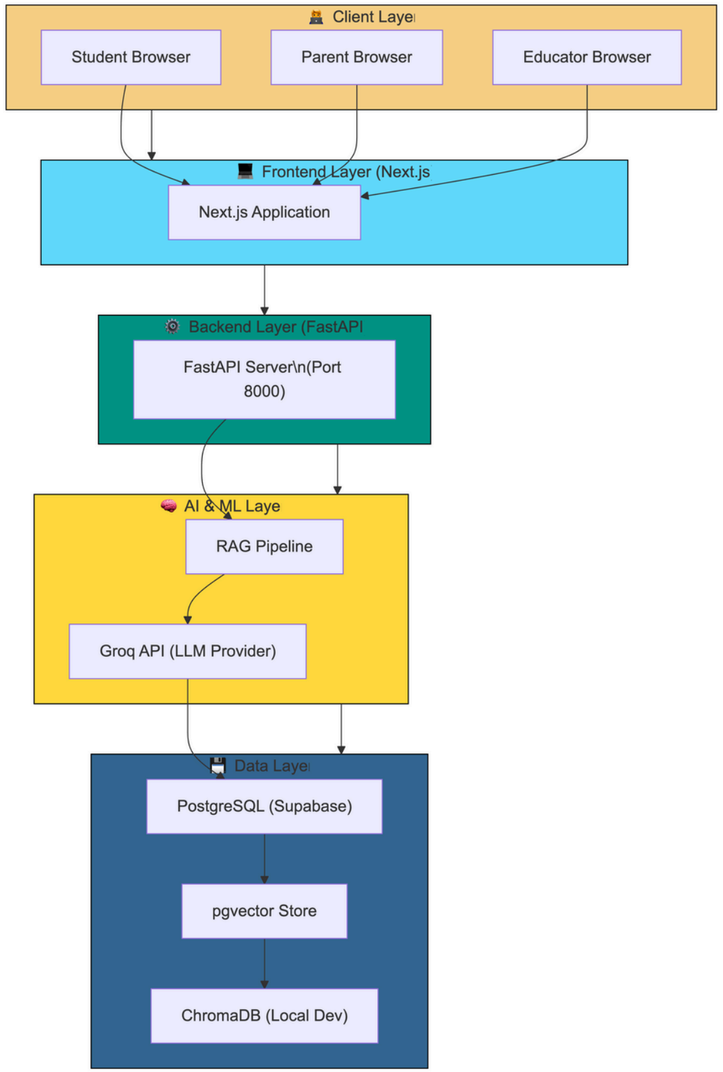
AI Stack & Technical Foundation

AI Models:

- openAI (Gpt 4) — fast inference for tutoring & quiz generation
- SentenceTransformer (all-MiniLM-L6-v2) — 384-dim embeddings for RAG retrieval
- Why: Groq for speed/cost; SentenceTransformer for efficient semantic search

Architecture:

- RAG Pipeline: Embed → Retrieve (Supabase Vector) → Generate with context
- Database: Supabase PostgreSQL with pgvector
- Connection Pooling: 15-connection pool for performance



User Data & Analytics

Data Collected:

- Quiz Attempts: topic, grade, difficulty, score, answers (JSON)
- Practice Sessions: S1/S2 questions, solutions, feedback mode
- Time Tracking: per-module time, daily totals, session timestamps
- User Profiles: grade, parent-student relationships

Analysis Capabilities:

- Parent Dashboard: accuracy trends, topic heatmaps, time-on-task
- Progress Tracking: skill-level breakdowns, strength/weakness identification
- Privacy: Role-based access; parents see only their children's data

Technical Challenges Overcome

Challenge 1: LaTeX Formatting Issues

- Problem: LLM generated malformed LaTeX ($\begin${aligned}$ instead of \begin{aligned})
- Solution: 20+ regex patterns in _normalize_latex() to fix common errors
- Result: 95%+ KaTeX render success on frontend

Challenge 2: Database Migration

- Problem: Migrated from SQLite + ChromaDB to Supabase PostgreSQL
- Solution: Created VectorCollection wrapper for compatibility; connection pooling
- Result: Unified database, better scalability, vector search in PostgreSQL

Challenge 3: Memory Optimization

- Problem: Embedding model loaded on every request (slow startup)
- Solution: Lazy loading — model loads only when first needed
- Result: 3x faster cold starts, reduced memory footprint

User Data & Analytics

Data Collected:

- Quiz Attempts: topic, grade, difficulty, score, answers (JSON)
- Practice Sessions: S1/S2 questions, solutions, feedback mode
- Time Tracking: per-module time, daily totals, session timestamps
- User Profiles: grade, parent-student relationships

Analysis Capabilities:

- Parent Dashboard: accuracy trends, topic heatmaps, time-on-task
- Progress Tracking: student’s activities, consistency and progress
- Export: CSV/PDF reports for educators
- Privacy: Role-based access; parents see only their children's data

PII Protection & Access Control

Privacy & Data Security:

- Password Protection: SHA256 hashing (never stored/returned in plaintext)
- Secure Storage: Encrypted Supabase PostgreSQL database
- Access Control: Role-based (students vs. parents), JWT authentication
- Data Isolation: Parents only see linked students; no cross-user leakage

AI Safety:

- Content Filtering: System prompts refuse cheating requests (validated 5/5 safety score)
- Grade-Appropriate: Content filtered by student grade level
- Parent Controls: Toggle direct answers, lock difficulty, set daily goals

System Reliability:

- Error Handling: Try-catch on all endpoints with graceful fallbacks
- Database: Connection pooling (10 connections) for performance
- Validation: Input validation + database constraints
- Infrastructure: Managed Supabase (99.9% uptime SLA)

Next Step

Features to update:

Optimizing Prompts further to fix the Latex rendering issue.

Data Collection:

Quantitative Metrics:

- Time Saved: Track session duration (S1/S2/S3) vs. traditional methods
- Learning Outcomes: Pre/post quiz scores, accuracy trends over time
- Engagement: Daily active users, time-on-task per module, completion rates
- Skill Improvement: Topic-wise accuracy heatmaps, strength/weakness shifts

Qualitative Metrics:

- Parent Surveys: "Did you see improved understanding?" (Likert scale 1-5)
- Student Interviews: "Did step-by-step guidance help?" (open-ended)
- Teacher Feedback: Compare homework quality before/after

THANK YOU!