

StudyGapAI Complete Technical Project Brief

_For Team Alignment & AI Tool Context

Project Overview

Project Name: StudyGapAI

Tagline: AI-powered diagnostic tool that identifies hidden knowledge gaps holding Nigerian JAMB students back

Timeline: 10 days (Nov 1-10, 2025)

Team Size: 4 (Backend Dev, AI/SE, Frontend Dev, Product Manager)

Submission Deadline: Monday, November 10, 2025

Problem Statement

The Core Problem:

Every year, 1.8 million Nigerian students take the JAMB exam, but only ~50% score above 200 (the threshold for most universities). The traditional explanation is "students don't study hard enough," but data suggests otherwise: (NOTE: THIS SECTION NEEDS REAL RESEARCH TO BACK IT UP)

Key Insights:

- Students study inefficiently, not insufficiently** - They spend 15-25 hours/week studying but focus on the wrong topics
- Hidden knowledge gaps** - A student weak in algebra might actually have a foundational gap in basic arithmetic that's never diagnosed
- Expensive alternatives** - Private tutors cost ₦10,000-30,000/month, inaccessible to most students
- Generic study resources** - YouTube has thousands of JAMB videos, but students don't know which ones to watch

The Market Gap: Existing solutions (Gap Analyzer, IXL, Mathos AI) are:

- Generic (global math, not JAMB-specific)

- Expensive (₦10,000+/month)
 - Require high bandwidth (video-heavy)
 - Only identify weak topics (don't diagnose root causes)
-

Our Solution

StudyGapAI is a two-step diagnostic platform that:

1. **Diagnoses** - 30-question multiple-choice quiz + text explanations for wrong answers
2. **Analyzes** - AI identifies not just weak topics, but the foundational gaps causing those weaknesses
3. **Prescribes** - Generates a personalized 6-week study plan with curated, free resources

Key Differentiators:

Feature	StudyGapAI	Competitors
Curriculum	JAMB-specific (Nigerian syllabus)	Generic global math
Diagnostic Method	2-step (MC + reasoning)	MC only or essay-based
Analysis Depth	Root cause (finds foundational gaps)	Surface-level (weak topics)
Pricing	₦2,500-3,000/month	₦10,000-30,000/month
Bandwidth	Text-based (works on 2G/3G)	Video-heavy
Resources	Curated free content	Generic or paywalled

User Flow

Step 1: Diagnostic Quiz (15-20 mins)

- Student takes 30-question multiple-choice quiz (Mathematics only for MVP)
- Questions cover 15 key JAMB Math topics (Algebra, Geometry, Calculus, etc.)
- For each wrong answer, student explains their reasoning in text

Step 2: AI Analysis (2-3 mins)

- AI (Gemini 2.0 Flash) analyzes:
 - Which answers were wrong

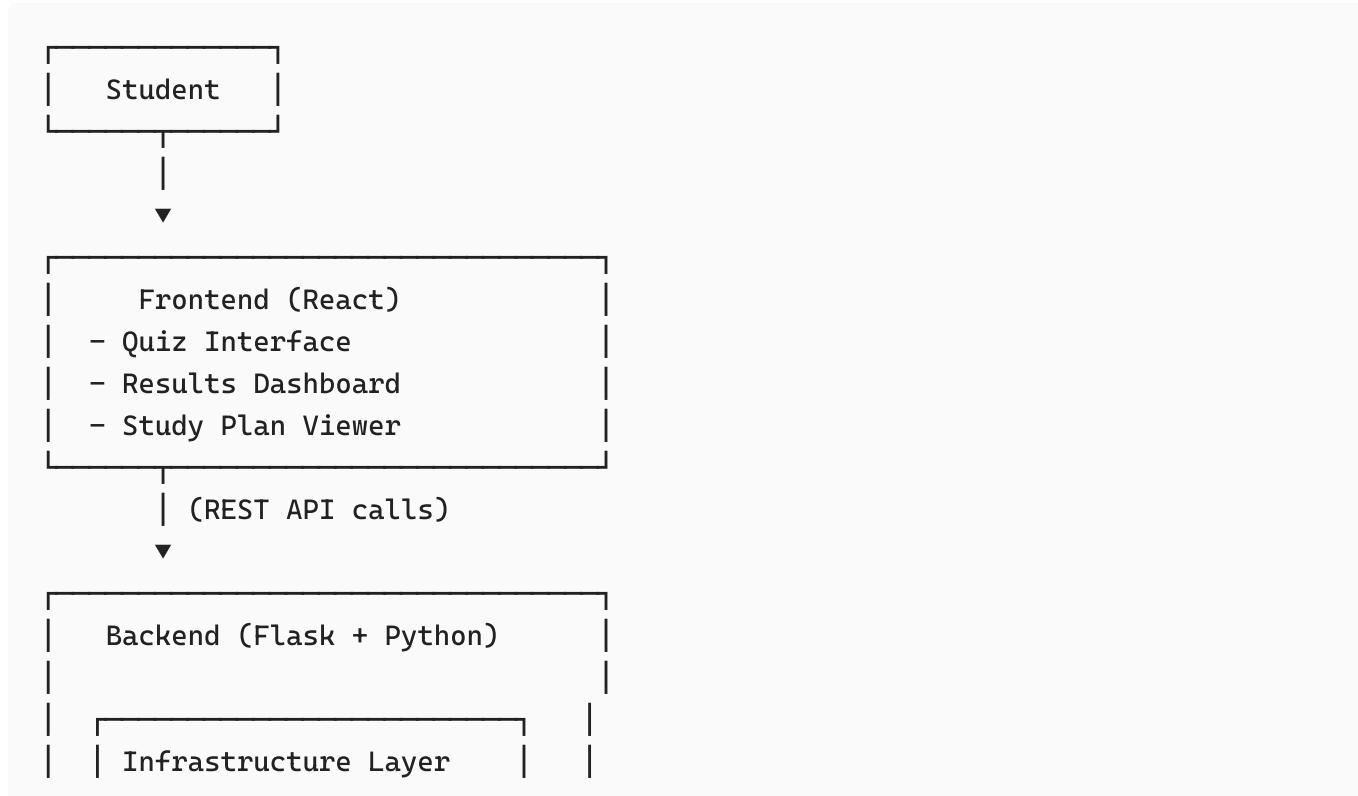
- Why the student got them wrong (from their explanations)
- Underlying foundational gaps (e.g., "weak in quadratic equations because doesn't understand FOIL method")
- Generates diagnostic report:
 - **Weak topics** with severity scores
 - **Strong topics** to build confidence
 - **Root causes** (the foundational gaps)
 - **Projected JAMB score** if student takes exam today

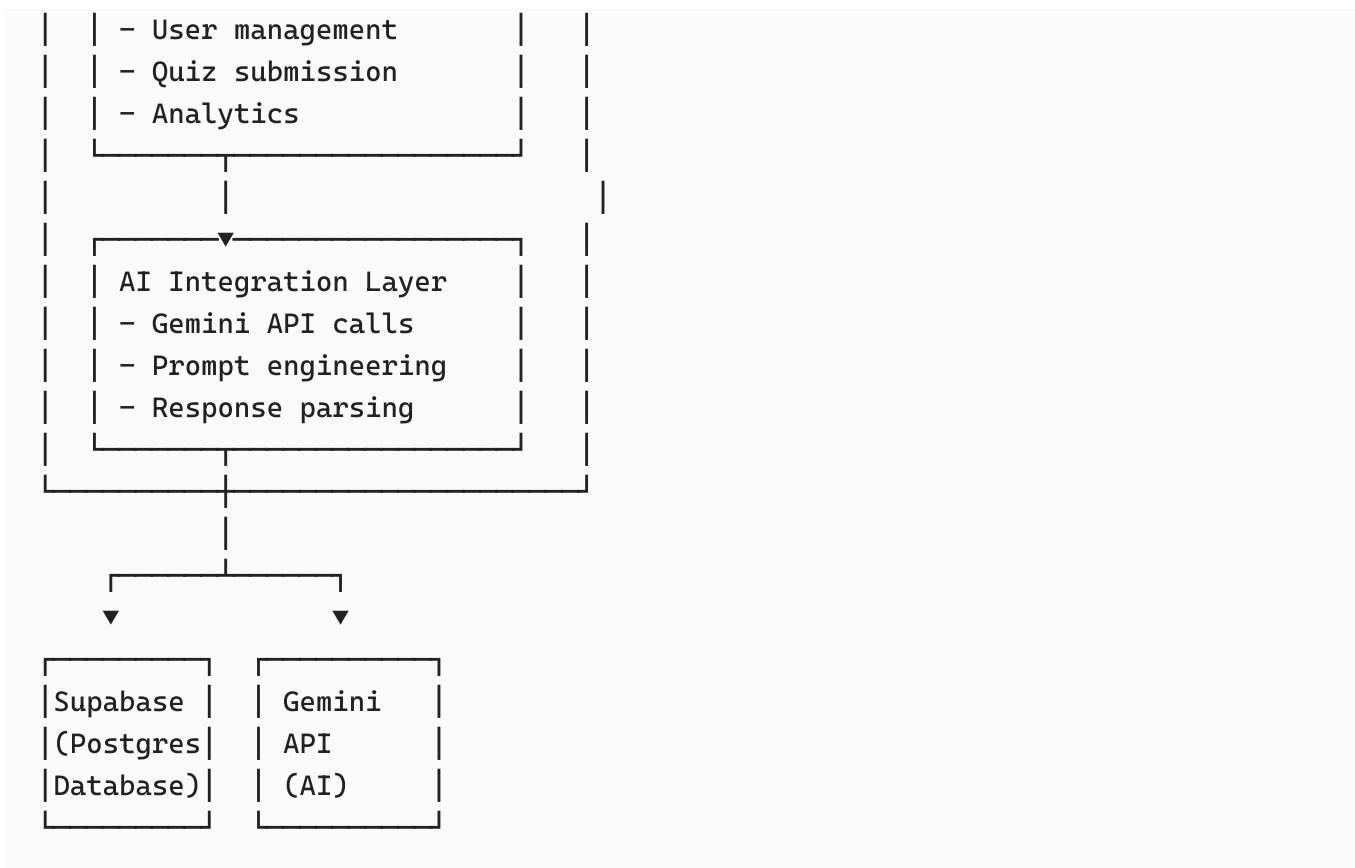
Step 3: Personalized Study Plan (instant)

- AI generates 6-week study plan:
 - Week-by-week topic breakdown
 - Daily goals (30-45 mins/day)
 - Curated resources for each topic:
 - 2 YouTube tutorial videos
 - 2 practice problem sets (JAMB past papers)
-

Technical Architecture

System Design:





🗄️ Database Schema (Supabase/PostgreSQL)

Tables:

1. users

```

id: UUID (primary key)
email: VARCHAR(255)
name: VARCHAR(255)
phone: VARCHAR(20)
target_score: INT (e.g., 250)
created_at: TIMESTAMP
last_active: TIMESTAMP

```

2. diagnostic_quizzes

```

id: UUID (primary key)
user_id: UUID (foreign key → users.id)
started_at: TIMESTAMP
completed_at: TIMESTAMP
total_questions: INT (30)

```

```
correct_answers: INT  
score_percentage: FLOAT
```

3. quiz_responses

```
id: UUID (primary key)  
quiz_id: UUID (foreign key → diagnostic_quizzes.id)  
question_id: UUID (foreign key → questions.id)  
student_answer: VARCHAR(1) (A/B/C/D)  
correct_answer: VARCHAR(1)  
is_correct: BOOLEAN  
explanation_text: TEXT (student's reasoning)  
time_spent_seconds: INT
```

4. questions

```
id: UUID (primary key)  
topic_id: UUID (foreign key → topics.id)  
question_text: TEXT  
option_a: TEXT  
option_b: TEXT  
option_c: TEXT  
option_d: TEXT  
correct_answer: VARCHAR(1)  
difficulty: VARCHAR(20) (easy/medium/hard)  
subtopic: VARCHAR(100)
```

5. topics

```
id: UUID (primary key)  
name: VARCHAR(100) (e.g., "Algebra", "Geometry")  
description: TEXT  
prerequisite_topic_ids: UUID[] (array of topic IDs)  
jamb_weight: FLOAT (% of JAMB questions from this topic)
```

6. ai_diagnoses

```
id: UUID (primary key)  
quiz_id: UUID (foreign key → diagnostic_quizzes.id)  
weak_topics: JSONB ([{topic_id, severity, root_cause}])  
strong_topics: JSONB ([{topic_id, score}])  
analysis_summary: TEXT
```

```
projected_score: INT  
foundational_gaps: JSONB ([{gap_description, affected_topics}])  
generated_at: TIMESTAMP
```

7. study_plans

```
id: UUID (primary key)  
user_id: UUID (foreign key → users.id)  
diagnostic_id: UUID (foreign key → ai_diagnostics.id)  
plan_data: JSONB (full 6-week plan structure)  
created_at: TIMESTAMP  
updated_at: TIMESTAMP
```

8. resources

```
id: UUID (primary key)  
topic_id: UUID (foreign key → topics.id)  
type: VARCHAR(20) (video/practice_set)  
title: VARCHAR(255)  
url: TEXT  
source: VARCHAR(100) (YouTube/JAMB Past Papers)  
duration_minutes: INT (for videos)  
difficulty: VARCHAR(20)  
upvotes: INT (for future rating feature)
```

🔌 API Endpoints

Backend Infrastructure (Built by Backend Dev):

User Management:

```
POST  /api/users/register  
POST  /api/users/login (optional for MVP)  
GET   /api/users/:userId  
PUT   /api/users/:userId/target-score
```

Quiz Management:

```
POST  /api/quiz/start          # Creates new quiz session  
POST  /api/quiz/:quizId/submit # Saves all responses
```

```
GET /api/quiz/:quizId/results # Gets quiz results  
GET /api/questions # Gets 30 diagnostic questions
```

AI Integration Layer (Built by AI/SE):

Diagnostic Analysis:

```
POST /api/ai/analyze-diagnostic  
Request: {  
    userId: UUID,  
    quizId: UUID,  
    responses: [{  
        questionId: UUID,  
        studentAnswer: "B",  
        correctAnswer: "C",  
        isCorrect: false,  
        explanationText: "I thought..."  
    }]  
}  
  
Response: {  
    weakTopics: [{  
        topicId: UUID,  
        topicName: "Algebra",  
        severity: "high",  
        rootCause: "Doesn't understand FOIL method"  
    }],  
    strongTopics: [{topicId, topicName, score}],  
    analysisSummary: "You struggle with...",  
    projectedScore: 165,  
    foundationalGaps: [{  
        gapDescription: "Weak in basic fraction operations",  
        affectedTopics: ["Algebra", "Calculus"]  
    }]  
}
```

Study Plan Generation:

```
POST /api/ai/generate-study-plan  
Request: {  
    userId: UUID,  
    diagnosticId: UUID,  
    weakTopics: [...],  
    targetScore: 250,
```

```
weeksAvailable: 6
}

Response: {
  weeks: [
    {
      weekNumber: 1,
      focus: "Foundational Arithmetic",
      topics: [
        {
          topicId: UUID,
          topicName: "Fractions",
          dailyGoals: "Practice 10 problems",
          estimatedTime: "30 mins/day",
          resources: [
            {
              type: "video",
              title: "Fractions Made Easy",
              url: "youtube.com/...",
              duration: 15
            }
          ],
          milestones: "Complete 50 practice problems"
        }
      ]
    }
}
```

Answer Explanation:

```
POST /api/ai/explain-answer
Request: {
  questionId: UUID,
  studentAnswer: "B",
  correctAnswer: "C",
  studentReasoning: "I thought..."
}

Response: {
  explanation: "You chose B because...",
  correctReasoning: "The right approach is...",
  commonMistake: "Many students confuse...",
  relatedTopics: ["Algebra", "Functions"]
}
```

Plan Adjustment:

```
POST /api/ai/adjust-plan
Request: {
```

```
userId: UUID,  
studyPlanId: UUID,  
completedTopics: [UUID],  
newWeakTopics: [UUID] (from mini-quizzes)  
}  
  
Response: {  
    updatedPlan: {...}  
}
```

AI Implementation Details

Model: Google Gemini 2.0 Flash (Experimental)

Why Gemini?

- **Free tier:** 1,500 requests/day (enough for 50-75 students/day)
- **Fast:** ~2-3 second response time
- **JSON mode:** Returns structured data (easier to parse)
- **Multimodal ready:** Can add image analysis later (for showing work)

Prompt Engineering Strategy:

Diagnostic Analysis Prompt:

```
You are an expert JAMB Mathematics tutor in Nigeria.
```

A student just completed a 30-question diagnostic quiz.

Analyze their performance and identify:

1. Weak topics (topics where they scored <60%)
2. Strong topics (topics where they scored >80%)
3. ROOT CAUSES of weaknesses (e.g., "struggles with quadratics because doesn't understand factoring")
4. Foundational gaps (basic concepts they're missing)

Student data:

```
{quiz_responses}
```

Return a JSON object with this structure:

```
{  
    "weakTopics": [...],  
    "strongTopics": [...],
```

```
"analysisSummary": "...",
"projectedScore": 165,
"foundationalGaps": [...]
}
```

Be specific. Nigerian students need actionable feedback.

Study Plan Prompt:

You are a JAMB prep expert. Create a 6-week study plan for a student with these weak topics: {weak_topics}

Target score: {target_score}

Current projected score: {current_score}

Rules:

- Start with foundational gaps FIRST
- Build progressively (don't jump to advanced topics)
- Each week should have 3-4 topics max
- Include daily time estimates (30-45 mins/day)
- Prioritize topics with highest JAMB weight

Return JSON structured as 6 weeks of daily study goals.

Content Strategy:

Question Bank (30 Questions):

- 15 topics × 2 questions each
- Difficulty distribution: 40% easy, 40% medium, 20% hard
- Sourced from: JAMB past papers (2018-2024)

15 Topics:

1. Numbers & Numeration
2. Fractions, Decimals, Percentages
3. Algebra (Linear Equations)
4. Algebra (Quadratic Equations)
5. Simultaneous Equations
6. Indices & Logarithms
7. Functions & Graphs
8. Geometry (Angles, Triangles)

9. Mensuration (Area, Volume)
10. Trigonometry (Basic Ratios)
11. Trigonometry (Identities)
12. Calculus (Differentiation)
13. Calculus (Integration)
14. Statistics (Mean, Median, Mode)
15. Probability

Resource Curation (45-60 Resources):

For each topic:

- **2 YouTube videos:**
 - 1 tutorial (10-20 mins)
 - 1 worked examples (5-10 mins)
 - Criteria: Nigerian accent preferred, JAMB-specific, <5M views (quality over popularity)
- **2 Practice sets:**
 - 1 JAMB past paper questions (10-15 questions)
 - 1 WAEC/NECO crossover questions
 - PDFs hosted on Google Drive (free access)



Success Metrics

Demo Day Metrics (What Judges Will See):

1. **Diagnostic accuracy:** Can the AI correctly identify weak topics?
2. **Study plan quality:** Are recommendations logical and actionable?
3. **User experience:** Can a student complete the flow in <25 mins?
4. **Mobile responsiveness:** Does it work on 2G/3G networks?

Post-Hackathon Metrics (If We Build This for Real):

1. **User engagement:** % of users who complete study plan Week 1
2. **Score improvement:** Average JAMB score increase for active users
3. **Retention:** % of users who return after 1 week
4. **Conversion:** % of free users who upgrade to premium

Business Model

Freemium Pricing:

Tier	Price	Features
Free	₦0	- 1 diagnostic test/month - Basic study plan - Limited resource access
Premium	₦2,500/month	- Unlimited diagnostics - Detailed AI feedback - Adaptive study plans - Progress tracking - All resources unlocked
School Licensing	₦150,000/year	- 100 student accounts - Teacher dashboard - Class analytics - Priority support

Revenue Projections:

Year 1 (Conservative):

- Target: 1% of 1.8M JAMB candidates = 18,000 users
- Conversion rate: 10% (1,800 paid users)
- MRR: $1,800 \times ₦2,500 = ₦4.5M/\text{month}$
- ARR: ₦54M (~\$70k USD)

Why Parents Will Pay:

- Nigerian parents already spend ₦10k-30k/month on tutors
- StudyGapAI is 5-10x cheaper
- Education is non-negotiable spending in Nigerian culture
- Proven results = word-of-mouth growth

Tech Stack Summary

Layer	Technology	Why?
Frontend	React + Vite	Fast, modern, team knows it

Layer	Technology	Why?
Styling	Tailwind CSS	Rapid UI development, mobile-first
Backend	Python + Flask	Backend dev's strength, simple
Database	Supabase (PostgreSQL)	Free, relational, auto-generated APIs
AI	Google Gemini 2.0 Flash	Free, fast, JSON mode
Hosting (Backend)	Railway	GitHub auto-deploy, free tier
Hosting (Frontend)	Vercel	Instant deploys, CDN, free tier
Version Control	GitHub	Industry standard

🎯 Objectives

Primary Objective:

Build a working MVP that demonstrates:

1. AI can accurately diagnose knowledge gaps from quiz + explanations
2. Personalized study plans are actionable and JAMB-specific
3. The product is usable on low-end mobile devices

Secondary Objectives:

1. Validate market demand (judges' reactions, Q&A)
2. Identify technical challenges for post-hackathon development
3. Build team experience in AI + EdTech

Win Conditions (Demo Day):

- Live demo completes without crashes
- Judges understand the problem/solution fit
- At least 2 judges ask "When can I try this?"
- Team is confident answering technical questions

🚧 Risks & Mitigation

Risk	Likelihood	Impact	Mitigation
Gemini API quota exceeded	Medium	High	Implement caching, mock responses for demo
Database schema changes mid-sprint	High	Medium	Lock schema on Day 2, document changes
Frontend-backend integration bugs	High	High	Daily integration tests, clear API contracts
Live demo fails	Medium	Critical	Record backup video, prepare screenshots
Team member drops out	Low	Critical	Cross-train on Day 5, document everything



For AI Tools (Context Summary)

When prompting AI assistants (Cursor, Copilot, Claude), include:

"I'm building StudyGapAI, a JAMB diagnostic platform in Nigeria.

- **Stack:** Python Flask backend, React frontend, Supabase database, Gemini AI
- **Key features:** 30-question quiz, AI diagnostic analysis, personalized study plans
- **Database:** PostgreSQL with 9 tables (users, quizzes, questions, etc.)
- **Timeline:** 10-day hackathon sprint
- **My role:** [Backend Dev / AI Engineer / Frontend Dev / PM]

[Then ask your specific question]"

This gives AI tools full context for better code generation.

✓ Success Checklist (Demo Day)

Technical:

- Backend deployed and accessible
- Frontend deployed and mobile-responsive
- Database seeded with questions and resources
- All API endpoints working

- Gemini API calls succeed
- No console errors

Demo:

- Demo script practiced 5+ times
- Backup video recorded
- Screenshots of perfect states ready
- Internet connection tested
- Devices charged

Pitch:

- Deck finalized (15-20 slides)
- JAMB statistics verified
- Q&A answers prepared
- Team roles clear (who speaks when)