

 **TUTORUG SECONDARY - COMPLETE DESIGN DOCUMENT**

 **EXECUTIVE SUMMARY**

Project Name: TutorUG Secondary

Tagline: "Your AI-Powered Path to UNEB Excellence"

Target Market: Ugandan O-Level students (S1-S4), ages 13-17

Pricing: 25,000 UGX/month (Monthly) | 250,000 UGX/year (Annual - save 2 months + 50k)

Timeline: 10 weeks to MVP launch

Team: 3 co-founders (1 Backend Dev, 1 Frontend Dev, 1 Full-stack/Marketing)

 **PRODUCT VISION****Problem Statement**

- Private tuition costs 50,000-200,000 UGX/month per subject
- Coaching centers charge 100,000-300,000 UGX/month
- Many students in rural areas lack access to quality teachers
- Students need 24/7 help with homework and exam prep
- Parents want affordable, quality education for their children
- Lack of peer learning opportunities for students in remote areas
- Traditional education doesn't adapt to individual learning pace

Solution

An AI-powered mobile and web app that provides:

- Personalized tutoring for all O-Level subjects
- Unlimited access to curriculum-aligned content
- Interactive quizzes and practice tests
- Progress tracking for students and parents
- Community learning features
- All for just 25,000 UGX/month (less than 1,000 UGX per day)

COMPREHENSIVE SYSTEM DESCRIPTION

Overview

TutorUG Secondary is an intelligent, adaptive learning platform designed specifically for the Ugandan education system. The platform combines artificial intelligence, peer learning, and gamification to create an engaging educational experience that's accessible to every Ugandan student, regardless of their location or economic background.

Core Philosophy

The system is built on three fundamental principles:

1. **Adaptive Learning:** The AI tutor adapts to each student's pace, learning style, and comprehension level
 2. **Community-Driven:** Students learn better together - peer explanations and collaborative study create deeper understanding
 3. **Ugandan Context First:** Every example, explanation, and interaction reflects Ugandan culture, locations, and real-world applications
-

System Components

1. INTELLIGENT AI TUTORING ENGINE

Conversational Learning Interface

- Natural language processing understands student questions in English and Luganda
- Context-aware responses that reference previous topics and identify knowledge gaps
- Multi-modal explanations: text, voice, visual diagrams, and step-by-step breakdowns
- Ugandan-contextualized examples using local scenarios (boda bodas, markets, schools)

Adaptive Assessment System After each subtopic, the AI implements a sophisticated check-for-understanding protocol:

LEARNING FLOW:

1. AI teaches concept with Ugandan examples
2. Interactive checkpoint: "Before we move on, let me verify your understanding..."
3. AI asks 2-3 targeted questions (varying difficulty)
4. Response Analysis:

SCENARIO A - All Correct (100%)

- "Ayee! You're getting it! 🎉"
- Award points/badges
- Proceed to next topic
- Mark topic as "Mastered"

SCENARIO B - Partial Understanding (50-80%)

- "Good effort! Let me clarify [specific concept]..."
- Provide alternative explanation (different approach)
- Re-ask similar questions with hints
- Only proceed when 80%+ mastery achieved

SCENARIO C - Struggling (<50%)

- "No stress, let's go back to basics..."
- Break down concept into smaller pieces
- Use more visual/concrete examples
- Provide step-by-step guided practice
- Suggest related foundational topics to review

5. Continuous adaptation based on student's response patterns

Natural Text-to-Speech Integration

- Every lesson can be read aloud in natural Ugandan-accented English
- Students can learn while commuting or doing chores
- Reduces reading fatigue and accommodates different learning styles
- Powered by OpenAI TTS with optional ElevenLabs upgrade for premium voice quality

Smart Content Delivery

- Curriculum mapped directly to UNEB syllabus (by term and class)
- Pre-requisite checking: "Before learning quadratics, you should understand linear equations"
- Personalized learning paths based on student's level and progress
- Automatic difficulty adjustment based on performance

2. COMPREHENSIVE CURRICULUM COVERAGE

Multi-Level Support The system covers the entire Ugandan education spectrum:

- **Primary Level (P1-P7):** Foundation subjects with age-appropriate engagement
- **O-Level (S1-S4):** Complete UNEB curriculum for all subjects
- **A-Level (S5-S6):** Advanced topics by combination (PCM, BCM, HEG, etc.)
- **University:** Specialized subjects (Engineering Math, Statistics, etc.)

Subject Categories

Core Sciences & Mathematics:

- Mathematics (Arithmetic, Algebra, Geometry, Calculus, Statistics)
- Physics (Mechanics, Electricity, Waves, Modern Physics)
- Chemistry (Organic, Inorganic, Physical Chemistry)
- Biology (Cell Biology, Genetics, Ecology, Human Anatomy)

Languages & Literature:

- English Language (Grammar, Composition, Comprehension)
- Luganda (Reading, Writing, Grammar)
- Literature in English (Poetry, Prose, Drama, African Writers)

Humanities & Social Sciences:

- History (Ugandan, East African, World History)
- Geography (Physical, Human, Economic Geography)
- Christian/Islamic Religious Education (CRE/IRE)
- Entrepreneurship Education

Applied Subjects:

- ICT/Computer Science (Programming, Digital Literacy)
- Commerce (Accounting, Business Studies)
- Agriculture (Crop Production, Animal Husbandry)

Content Structure per Subject:

SUBJECT (e.g., Mathematics O-Level)

└─ TOPICS (e.g., "Algebra", "Geometry")

 └─ SUBTOPICS (e.g., "Quadratic Equations")

 └─ Introduction & Theory

 └─ Ugandan Contextualized Examples

 └─ Step-by-Step Worked Solutions

 └─ Common Mistakes & How to Avoid Them

 └─ Practice Problems (Easy → Hard)

 └─ Real-World Applications

 └─ Assessment Quiz

3. INTERACTIVE COMMUNITY LEARNING HUB

Subject-Based Discussion Channels

MATHEMATICS CHANNEL (2,341 active students)

- Students post questions: "How do you solve simultaneous equations?"
- Peer responses: Other students explain in their own words
- AI Tutor participation: Jumps in when needed or validates answers
- Upvoting system: Best explanations rise to top
- "Solved" marking: Question asker marks when satisfied

Real-Time Collaboration Features:

- Live chat with typing indicators
- Markdown support for mathematical equations
- Image sharing for diagrams/handwritten work
- Voice messages for verbal explanations
- Thread-based conversations to keep discussions organized

Study Groups (Micro-Communities) Students can create or join focused study groups:

EXAMPLE STUDY GROUP:

Name: "S4 UNEB 2026 Mathematics Warriors"

Members: 8 students (from different schools)

Subjects: Math, Physics, Chemistry

Features:

- Private group chat
- Shared study schedules
- Group quiz competitions
- Collaborative notes
- Accountability check-ins
- Video study sessions (Phase 2)

Safety & Moderation System:

- **AI-Powered Content Filtering:** OpenAI Moderation API scans every message
- **Automated Blocks:** Profanity, personal information, bullying keywords filtered instantly
- **Human Moderation:** Trained moderators review flagged content within 2 hours
- **User Reporting:** Easy one-click report button on every message
- **Age Verification:** Required during signup with parental consent for under-16
- **Anonymous Usernames:** Students identified as "John_S3_Kampala" (no personal details visible)
- **Clear Community Guidelines:** Displayed prominently with consequences for violations

4. GAMIFICATION & MOTIVATION ENGINE

Reputation & Ranking System Students earn points through positive actions:

POINT SYSTEM:

- +2 pts - Daily login (maintain streak)
- +5 pts - Complete a subtopic
- +10 pts - Pass a quiz with 70%+
- +15 pts - Score 90%+ on practice test
- +5 pts - Answer peer's question in community
- +10 pts - Your answer marked "helpful"
- +20 pts - Create study guide that helps 10+ students
- +25 pts - Maintain 7-day study streak

STUDENT RANKS:

-  Learner (0-100 pts)
-  Helper (101-300 pts)
-  Tutor (301-700 pts)
-  Expert (701-1,500 pts)
-  Master (1,501-3,000 pts)
-  Legend (3,000+ pts)

Achievement Badges

SUBJECT MASTERY:

-  "Math Wizard" - Complete 20 Math topics
-  "Chemistry King/Queen" - Score 85%+ average in Chemistry
-  "Physics Pro" - Solve 50 Physics problems

COMMUNITY CONTRIBUTION:

-  "Team Player" - Help 25 fellow students
-  "Problem Solver" - Provide 10 "helpful" answers
-  "Study Buddy" - Active in community 30 days straight

ACHIEVEMENT MILESTONES:

-  "On Fire!" - 7-day study streak
-  "Unstoppable" - 30-day study streak
-  "Perfectionist" - 5 perfect scores (100%) on quizzes
-  "Comeback Kid" - Improve subject score by 30%+

School Leaderboards Friendly competition without tribal tension:

TOP SCHOOLS THIS MONTH (Based on collective student performance)

1.  St. Mary's College Kisubi - 45,234 pts
2.  King's College Budo - 43,876 pts
3.  Gayaza High School - 42,198 pts

...

Your School: Ntinda View College (#18)

Gap to #15: Only 800 points! 

Weekly School Challenge:

"Physics Week" - School with most Physics quiz completions wins!

Prize: Featured on homepage + all students get bonus badge

Daily Challenges & Streaks

TODAY'S CHALLENGE:

- Login (2 pts) - DONE
- Complete 1 topic (5 pts) - In Progress
- Answer 2 community questions (10 pts)
- Take a quiz (10 pts)

Complete all = 27 pts + BONUS 10 pts! 

STUDY STREAK:  7 DAYS

Your longest: 12 days

Keep going to beat your record!

5. COMPREHENSIVE ASSESSMENT SYSTEM

Practice Quiz Library

- 1,000+ UNEB-style questions across all subjects
- Categorized by: Subject → Topic → Difficulty
- Timed mode (simulates exam conditions)
- Untimed mode (learning-focused)
- Instant feedback with detailed explanations

Past Papers Integration (Phase 2)

- Complete UNEB past papers (2010-2025)

- Marking schemes with explanation of marking criteria
- Time management tips for each paper
- Question pattern analysis

Intelligent Progress Tracking

STUDENT DASHBOARD SHOWS:

Overall Progress

- Topics Completed: 45/150 (30%)
- Average Quiz Score: 78%
- Total Study Time: 28.5 hours
- Study Streak: 12 days
- Current Rank: Tutor (487 pts)

Subject Breakdown

- Mathematics: 65% complete (Avg: 82%)
- Physics: 40% complete (Avg: 75%)
- Chemistry: 35% complete (Avg: 68%)
- Biology: 45% complete (Avg: 80%)

Weak Areas Identified (AI Analysis)

- Mathematics: Word problems, Trigonometry
- Physics: Electricity calculations
- Chemistry: Organic chemistry nomenclature

Recommended Next Steps

- Review: "Linear Equations" (prerequisite for current topic)
- Practice: Take Physics electricity quiz again
- Explore: Start "Trigonometry Basics"

Goals This Week

- Complete 5 topics
- Maintain study streak
- Score 80%+ on all quizzes

Parent Dashboard (Phase 2) Parents can monitor progress through secure access:

- Weekly email reports: topics covered, time spent, quiz scores
- Access to child's dashboard (read-only)
- Alerts for prolonged inactivity

- Celebration notifications: "Your child earned Math Wizard badge!"

6. MOBILE MONEY PAYMENT INTEGRATION

Seamless Ugandan Payment Experience

- **MTN Mobile Money:** Direct USSD integration
- **Airtel Money:** One-tap payment
- **Flutterwave Gateway:** Handles all payment processing securely
- **Automatic Subscription Management:** Auto-renewal with SMS reminders

Flexible Subscription Options

FREE TRIAL (7 Days):

- ✓ 10 AI tutor questions/day
- ✓ Access to 2 subjects
- ✓ Read-only community access
- ✓ Basic progress tracking
- ✗ No study groups
- ✗ No text-to-speech
- ✗ Limited practice quizzes

PREMIUM MONTHLY (25,000 UGX):

- ✓ Unlimited AI tutor questions
- ✓ ALL subjects (Primary to University)
- ✓ Full community participation (post, reply, react)
- ✓ Create/join 5 study groups
- ✓ Text-to-speech on all content
- ✓ Unlimited practice quizzes
- ✓ Priority support
- ✓ Detailed progress analytics
- ✓ Downloadable notes (offline access)

PREMIUM ANNUAL (250,000 UGX):

- ✓ Everything in Monthly
- ✓ SAVE 50,000 UGX (2 months free!)
- ✓ Exclusive "Annual Member" badge
- ✓ Early access to new features
- ✓ Priority in community rankings

Referral Program

REFER A FRIEND, BOTH WIN:

When your friend subscribes:

- You get: 1 month free extension
- Friend gets: 50% off first month
- Share your code: TUG-JOHN123

Leaderboard:

Top Referrer this month: Sarah_S4 (23 referrals) 

Reward: 6 months free + Featured on homepage

User Experience Flow

New Student Onboarding

STEP 1: Registration

- Enter phone number
- Receive OTP via SMS
- Verify OTP
- Create profile:
 - First Name, Last Name
 - Select Level (Primary/O-Level/A-Level/University)
 - Select Class (e.g., S3)
 - School Name (optional)
 - Region (e.g., Kampala)
- Start 7-day FREE trial

STEP 2: Interest Selection

- Choose your struggling subjects (AI prioritizes these)
- Set learning goals (e.g., "Pass UNEB with Div 1")
- Choose study schedule preference

STEP 3: Quick Tutorial

- "How to ask the AI tutor questions"
- "How to navigate subjects and topics"
- "How to join the community"
- "How to track your progress"

STEP 4: First Learning Session

- AI suggests starting point based on level
- Complete one topic with guidance
- Take first mini-quiz
- Earn first badge: "Welcome to TutorUG!" 

Daily Learning Session

TYPICAL USER JOURNEY:

1. Login → Dashboard loads
 - See study streak (motivates continuation)
 - Today's challenge displayed
 - "Continue where you left off" button

2. Choose Learning Path:

OPTION A: Structured Learning

- Browse Subjects → Select Math
- Browse Topics → Select "Quadratic Equations"
- Read/Listen to subtopic content
- AI asks comprehension questions
- Pass checkpoint → Next subtopic
- Complete topic → Take quiz

OPTION B: Question-Based Learning

- Click "Ask AI Tutor"
- Type/speak: "I don't understand how friction works"
- AI explains with Ugandan examples
- Student: "Can you give me a problem to solve?"
- AI provides practice problem
- Repeat until satisfied

OPTION C: Community Learning

- Open Community tab
- Browse #physics-o-level channel
- See question: "How do you calculate momentum?"
- Read peer explanations
- AI Tutor's official answer also shown
- Post own question or help others
- Earn points for contributions

3. Take Breaks with Gamification

- Check leaderboard position
- View newly earned badges
- See school ranking
- Get motivated to continue

4. End Session

- Progress automatically saved
- Streak updated

— Summary shown: "Today you completed 2 topics, scored 85% on quiz!"

— Tomorrow's goal suggested

Taking a Quiz

QUIZ EXPERIENCE:

1. Select Quiz

- Topic-specific (10-15 questions)
- Mixed practice (20-30 questions)
- Mock exam (50+ questions, timed)

2. During Quiz

- Timer (if timed mode)
- Question counter: "5/10"
- Skip and return later
- Flag questions for review
- Auto-save progress

3. Submit Quiz

- Confirmation: "Are you sure? You have 2 unanswered questions"
- Submit

4. Instant Results

- Score: 85% (8.5/10)
- Pass/Fail indicator
- Time taken: 12 minutes
- Percentile: "Better than 68% of students"
- Points earned: +15 pts

5. Detailed Breakdown

FOR EACH QUESTION:

- Your answer (marked correct/wrong)
- Correct answer
- Explanation (AI-generated)
- Concept being tested
- Link to review relevant subtopic

6. Actionable Insights

- "You struggle with word problems - review this topic"
- "Great job on factorization!"

Key Differentiators

What Makes TutorUG Secondary Unique:

1. **Ugandan Context First**: Every explanation uses local examples - boda bodas, Owino Market, Kampala streets, not generic "Train A and Train B" problems
2. **Adaptive Check-for-Understanding**: Unlike passive video platforms, our AI actively verifies comprehension before proceeding, ensuring no student gets left behind
3. **Peer Learning Community**: Students don't just learn from AI - they teach and learn from each other, creating a collaborative ecosystem
4. **Natural Voice Integration**: Content can be listened to in natural Ugandan-accented English, making it accessible for auditory learners and students with limited time
5. **Comprehensive Coverage**: One subscription gives access to ALL subjects from Primary to University - competitors charge per subject
6. **Gamified Motivation**: School leaderboards, badges, and study streaks transform learning from a chore into an engaging competition
7. **Affordable at Scale**: At 25,000 UGX/month (833 UGX per subject), it's 75-90% cheaper than traditional tuition while providing superior 24/7 access

Success Metrics

Month 3:

- 100 paying students
- 2,500,000 UGX revenue
- 50+ active community discussions/day
- Average 4.3/5 satisfaction rating

Month 6:

- 500 paying students
- 12,500,000 UGX revenue
- 200+ daily active users in community
- 15,000+ questions answered (AI + peers)
- 4.5/5 average rating

Month 12:

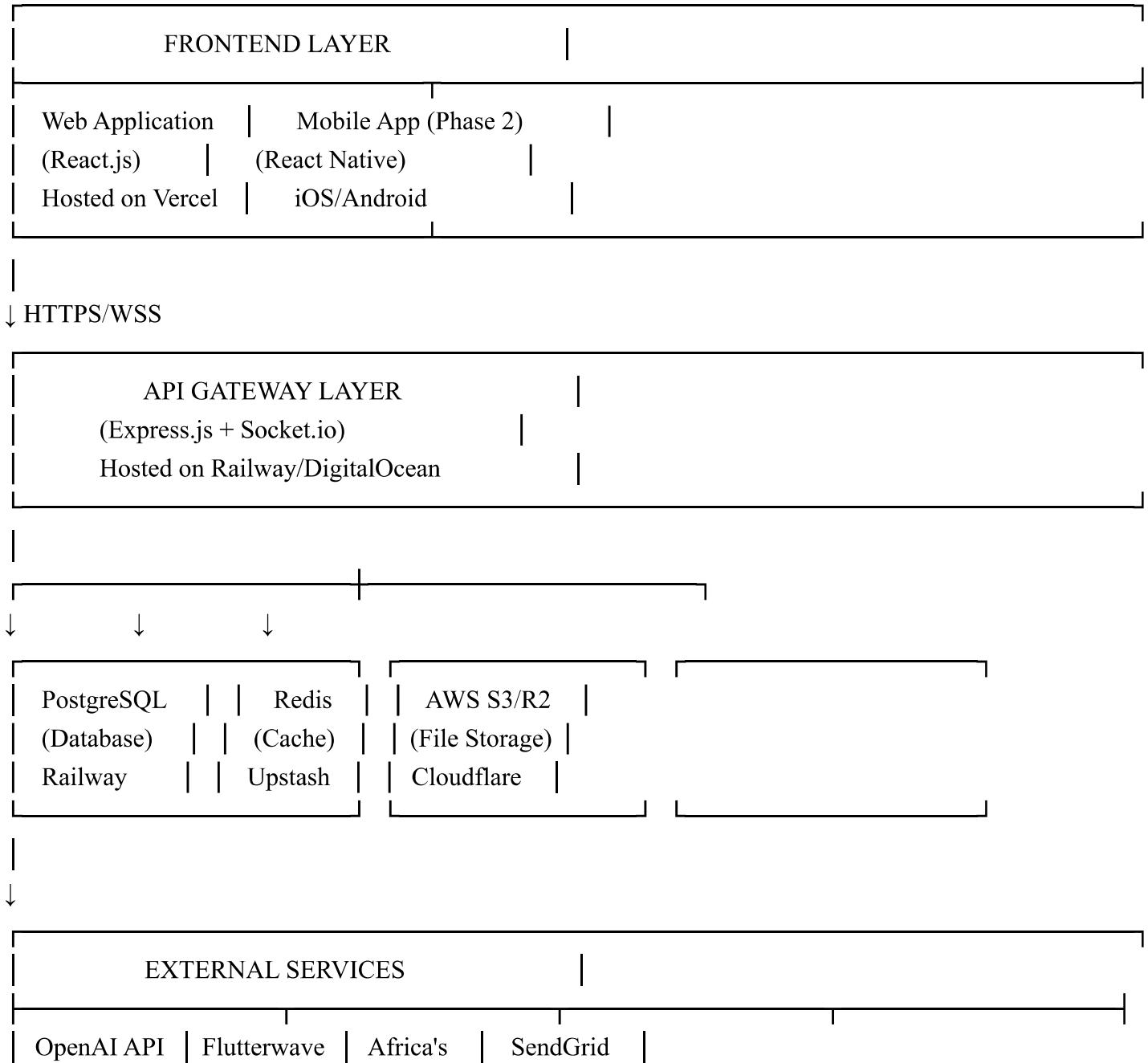
- 2,000 paying students
- 50,000,000 UGX revenue
- 1,000+ daily active users
- 50+ schools represented
- <8% monthly churn
- 4.7/5 average rating
- Featured in Ugandan education news

Engagement Targets:

- 60%+ daily active users
- Average 45 minutes per session
- 85%+ trial-to-paid conversion
- 40%+ students use community weekly
- 70%+ students maintain 7+ day streak

SYSTEM ARCHITECTURE

High-Level Architecture



(AI Tutor)	(Payments)	Talking	(Email)	
	(SMS)			

Technology Stack

Frontend:

- Framework: React.js 18+
- Routing: React Router v6
- State: Redux Toolkit / Zustand
- Styling: Tailwind CSS
- UI: Headless UI + Custom
- HTTP: Axios
- Real-time: Socket.io-client
- Forms: React Hook Form + Zod
- Charts: Recharts
- Animation: Framer Motion
- PWA: Workbox

Backend:

- Runtime: Node.js 18+
- Framework: Express.js
- Database: PostgreSQL 14+
- Cache: Redis
- ORM: Sequelize / Prisma
- Auth: JWT + bcrypt
- Validation: Joi / Zod
- Real-time: Socket.io
- Upload: Multer + AWS SDK
- Email: SendGrid
- SMS: Africa's Talking
- Payments: Flutterwave SDK
- AI: OpenAI SDK
- Cron: node-cron
- Testing: Jest + Supertest

DATABASE DESIGN

Complete Database Schema (PostgreSQL)

```
```sql
```

```
-- =====
-- USERS & AUTHENTICATION
-- =====
```

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

```
CREATE TABLE users (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 phone_number VARCHAR(15) UNIQUE NOT NULL,
 email VARCHAR(255),
 password_hash VARCHAR(255),
```

```
-- Profile Info
first_name VARCHAR(100),
last_name VARCHAR(100),
current_class VARCHAR(10) NOT NULL, -- S1, S2, S3, S4
school_name VARCHAR(255),
region VARCHAR(100),
```

```
-- Account Status
account_status VARCHAR(20) DEFAULT 'active',
email_verified BOOLEAN DEFAULT FALSE,
phone_verified BOOLEAN DEFAULT TRUE,
```

```
-- Subscription
subscription_status VARCHAR(20) DEFAULT 'trial',
subscription_plan VARCHAR(20),
subscription_start_date TIMESTAMP,
subscription_end_date TIMESTAMP,
trial_end_date TIMESTAMP,
```

```
-- Preferences
preferred_language VARCHAR(10) DEFAULT 'en',
notification_enabled BOOLEAN DEFAULT TRUE,
email_notifications BOOLEAN DEFAULT TRUE,
sms_notifications BOOLEAN DEFAULT FALSE,
```

```
-- Metadata
created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW(),
last_login TIMESTAMP,
login_count INTEGER DEFAULT 0,
```

```
-- Referral System
referral_code VARCHAR(20) UNIQUE,
referred_by UUID REFERENCES users(id),
```

```
-- Parent Access
```

```
parent_phone VARCHAR(15),
parent_email VARCHAR(255),
parent_access_enabled BOOLEAN DEFAULT FALSE,
parent_access_code VARCHAR(6)
);
```

```
CREATE INDEX idx_users_phone ON users(phone_number);
CREATE INDEX idx_users_subscription ON users(subscription_status);
CREATE INDEX idx_users_class ON users(current_class);
```

```
-- =====
-- OTP VERIFICATION
-- =====
```

```
CREATE TABLE otp_verifications (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 phone_number VARCHAR(15) NOT NULL,
 otp_code VARCHAR(6) NOT NULL,
 purpose VARCHAR(20) NOT NULL,
 verified BOOLEAN DEFAULT FALSE,
 expires_at TIMESTAMP NOT NULL,
 created_at TIMESTAMP DEFAULT NOW(),
 attempts INTEGER DEFAULT 0
);
```

```
CREATE INDEX idx_otp_phone ON otp_verifications(phone_number);
CREATE INDEX idx_otp_expires ON otp_verifications(expires_at);
```

```
-- =====
-- SUBJECTS & CURRICULUM
-- =====
```

```
CREATE TABLE subjects (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 name VARCHAR(100) NOT NULL,
 code VARCHAR(10) UNIQUE NOT NULL,
 level VARCHAR(20) NOT NULL,
 category VARCHAR(20),
 description TEXT,
 icon_url VARCHAR(255),
 color_hex VARCHAR(7),
 uneb_code VARCHAR(20),
 display_order INTEGER,
 is_active BOOLEAN DEFAULT TRUE,
```

```
created_at TIMESTAMP DEFAULT NOW()
);
```

```
INSERT INTO subjects (name, code, level, category, description, color_hex, display_order) VALUES
('Mathematics', 'MATH', 'O-Level', 'core', 'Numbers, Algebra, Geometry, Calculus', '#3B82F6', 1),
('Physics', 'PHYS', 'O-Level', 'core', 'Mechanics, Waves, Electricity, Magnetism', '#8B5CF6', 2),
('Chemistry', 'CHEM', 'O-Level', 'core', 'Matter, Atoms, Reactions, Organic Chemistry', '#10B981', 3),
('Biology', 'BIO', 'O-Level', 'core', 'Cells, Nutrition, Reproduction, Ecology', '#F59E0B', 4),
('English Language', 'ENG', 'O-Level', 'core', 'Grammar, Writing, Reading, Literature', '#EF4444', 5);
```

```
-- =====
-- TOPICS (Main curriculum sections)
-- =====
```

```
CREATE TABLE topics (
```

```
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 subject_id UUID NOT NULL REFERENCES subjects(id) ON DELETE CASCADE,
 title VARCHAR(255) NOT NULL,
 slug VARCHAR(255) NOT NULL,
 description TEXT,
```

```
 -- Curriculum Info
 uneb_reference VARCHAR(50),
 class_level VARCHAR(10),
 term INTEGER,
```

```
 -- Learning Metadata
 difficulty VARCHAR(20),
 estimated_duration_minutes INTEGER,
 learning_objectives TEXT[],
```

```
 -- Ordering
 display_order INTEGER,
 parent_topic_id UUID REFERENCES topics(id),
```

```
 -- Prerequisites
 prerequisites UUID[],
```

```
 -- Content
 introduction TEXT,
 key_concepts JSONB,
```

```
 -- Status
 is_published BOOLEAN DEFAULT FALSE,
```

```
created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW(),

UNIQUE(subject_id, slug)
);

CREATE INDEX idx_topics_subject ON topics(subject_id);
CREATE INDEX idx_topics_class ON topics(class_level);
CREATE INDEX idx_topics_order ON topics(display_order);

-- =====
-- SUBTOPICS (Detailed lesson content)
-- =====

CREATE TABLE subtopics (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 topic_id UUID NOT NULL REFERENCES topics(id) ON DELETE CASCADE,
 title VARCHAR(255) NOT NULL,
 slug VARCHAR(255) NOT NULL,

 -- Content
 content TEXT NOT NULL,
 summary TEXT,

 -- Examples
 ugandan_examples JSONB,
 practice_problems JSONB,

 -- Teaching Points
 key_points TEXT[],
 common_mistakes JSONB,
 tips_and_tricks TEXT[],

 -- Media
 images JSONB,
 videos JSONB,
 diagrams JSONB,

 -- Metadata
 estimated_read_time_minutes INTEGER,
 difficulty VARCHAR(20),
 display_order INTEGER,

 -- Status
);
```

```
is_published BOOLEAN DEFAULT FALSE,
created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW(),

UNIQUE(topic_id, slug)
);

CREATE INDEX idx_subtopics_topic ON subtopics(topic_id);
CREATE INDEX idx_subtopics_order ON subtopics(display_order);
```

```
-- =====
```

```
-- USER PROGRESS TRACKING
```

```
-- =====
```

```
CREATE TABLE user_progress (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
 subject_id UUID NOT NULL REFERENCES subjects(id) ON DELETE CASCADE,
 topic_id UUID REFERENCES topics(id) ON DELETE CASCADE,
 subtopic_id UUID REFERENCES subtopics(id) ON DELETE CASCADE,

 -- Progress Status
 status VARCHAR(20) DEFAULT 'not_started',
 completion_percentage INTEGER DEFAULT 0,

 -- Performance
 quiz_attempts INTEGER DEFAULT 0,
 best_quiz_score DECIMAL(5,2),
 latest_quiz_score DECIMAL(5,2),
 average_quiz_score DECIMAL(5,2),

 -- Time Tracking
 time_spent_seconds INTEGER DEFAULT 0,
 last_accessed TIMESTAMP,
 started_at TIMESTAMP,
 completed_at TIMESTAMP,

 -- Learning Analytics
 review_count INTEGER DEFAULT 0,
 struggle_indicators JSONB,

 created_at TIMESTAMP DEFAULT NOW(),
 updated_at TIMESTAMP DEFAULT NOW(),
```

```
 UNIQUE(user_id, subtopic_id)
);
```

```
CREATE INDEX idx_progress_user ON user_progress(user_id);
CREATE INDEX idx_progress_subject ON user_progress(subject_id);
CREATE INDEX idx_progress_status ON user_progress(status);
```

```
-- =====
-- AI CHAT SESSIONS
-- =====
```

```
CREATE TABLE chat_sessions (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
```

```
 -- Context
 subject_id UUID REFERENCES subjects(id),
 topic_id UUID REFERENCES topics(id),
 session_type VARCHAR(20) DEFAULT 'learning',
```

```
 -- Session Info
 title VARCHAR(255),
 started_at TIMESTAMP DEFAULT NOW(),
 ended_at TIMESTAMP,
 total_messages INTEGER DEFAULT 0,
```

```
 -- Analytics
 total_tokens_used INTEGER DEFAULT 0,
 cost_usd DECIMAL(10,4) DEFAULT 0,
 satisfaction_rating INTEGER,
```

```
 -- Status
 is_active BOOLEAN DEFAULT TRUE
);
```

```
CREATE INDEX idx_chat_user ON chat_sessions(user_id);
CREATE INDEX idx_chat_subject ON chat_sessions(subject_id);
```

```
-- =====
-- CHAT MESSAGES
-- =====
```

```
CREATE TABLE chat_messages (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
```

```
session_id UUID NOT NULL REFERENCES chat_sessions(id) ON DELETE CASCADE,
user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
```

```
-- Message Content
```

```
role VARCHAR(20) NOT NULL,
content TEXT NOT NULL,
content_type VARCHAR(20) DEFAULT 'text',
```

```
-- AI Metadata
```

```
model_used VARCHAR(50),
tokens_used INTEGER,
response_time_ms INTEGER,
```

```
-- Features
```

```
is_helpful BOOLEAN,
is_flagged BOOLEAN DEFAULT FALSE,
flag_reason TEXT,
```

```
created_at TIMESTAMP DEFAULT NOW()
```

```
);
```

```
CREATE INDEX idx_messages_session ON chat_messages(session_id);
CREATE INDEX idx_messages_created ON chat_messages(created_at);
```

```
-- ======
-- QUIZZES
-- ======
```

```
CREATE TABLE quizzes (
```

```
id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
topic_id UUID NOT NULL REFERENCES topics(id) ON DELETE CASCADE,
```

```
-- Quiz Info
```

```
title VARCHAR(255) NOT NULL,
description TEXT,
quiz_type VARCHAR(20) DEFAULT 'topic',
```

```
-- Configuration
```

```
total_questions INTEGER NOT NULL,
passing_score INTEGER DEFAULT 70,
time_limit_minutes INTEGER,
difficulty VARCHAR(20),
```

```
-- Questions (JSONB format)
```

```
questions JSONB NOT NULL,

-- Metadata
created_by UUID REFERENCES users(id),
is_published BOOLEAN DEFAULT TRUE,
created_at TIMESTAMP DEFAULT NOW(),
updated_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_quizzes_topic ON quizzes(topic_id);
CREATE INDEX idx_quizzes_type ON quizzes(quiz_type);

-- ======
-- QUIZ ATTEMPTS
-- ======

CREATE TABLE quiz_attempts (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
 quiz_id UUID NOT NULL REFERENCES quizzes(id) ON DELETE CASCADE,

 -- Attempt Info
 attempt_number INTEGER NOT NULL,
 started_at TIMESTAMP NOT NULL,
 completed_at TIMESTAMP,

 -- Results
 score DECIMAL(5,2),
 marks_obtained INTEGER,
 total_marks INTEGER,
 time_taken_seconds INTEGER,

 -- Answers (JSONB format)
 answers JSONB NOT NULL,

 -- Performance Analysis
 questions_correct INTEGER,
 questions_wrong INTEGER,
 questions_skipped INTEGER,
 weak_areas TEXT[],

 -- Status
 status VARCHAR(20) DEFAULT 'in_progress',
```

```
 created_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_attempts_user ON quiz_attempts(user_id);
CREATE INDEX idx_attempts_quiz ON quiz_attempts(quiz_id);
CREATE INDEX idx_attempts_completed ON quiz_attempts(completed_at);

-- -----
-- PAYMENTS & SUBSCRIPTIONS
-- -----

CREATE TABLE payments (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
 -- Payment Info
 amount DECIMAL(10,2) NOT NULL,
 currency VARCHAR(3) DEFAULT 'UGX',
 payment_method VARCHAR(50),
 -- Flutterwave Transaction
 transaction_id VARCHAR(255) UNIQUE,
 flutterwave_ref VARCHAR(255),
 -- Status
 status VARCHAR(20) DEFAULT 'pending',
 failure_reason TEXT,
 -- Subscription Details
 subscription_type VARCHAR(20),
 subscription_months INTEGER,
 -- Metadata
 ip_address VARCHAR(45),
 user_agent TEXT,
 -- Timestamps
 created_at TIMESTAMP DEFAULT NOW(),
 completed_at TIMESTAMP,
 refunded_at TIMESTAMP
);

CREATE INDEX idx_payments_user ON payments(user_id);
CREATE INDEX idx_payments_status ON payments(status);
```

```
CREATE INDEX idx_payments_transaction ON payments(transaction_id);

-- =====
-- NOTIFICATIONS
-- =====

CREATE TABLE notifications (
 id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
 user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
 -- Notification Content
 type VARCHAR(50) NOT NULL,
 title VARCHAR(255) NOT NULL,
 message TEXT NOT NULL,
 action_url VARCHAR(255),
 -- Status
 is_read BOOLEAN DEFAULT FALSE,
 read_at TIMESTAMP,
 -- Delivery
 delivery_method VARCHAR(20)[],
 email_sent BOOLEAN DEFAULT FALSE,
 sms_sent BOOLEAN DEFAULT FALSE,
 created_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_notifications_user ON notifications(user_id);
CREATE INDEX idx_notifications_read ON notifications(is_read);

-- =====
-- GAMIFICATION - USER REPUTATION
-- =====

CREATE TABLE user_reputation (
 user_id UUID PRIMARY KEY REFERENCES users(id) ON DELETE CASCADE,
 -- Points & Ranking
 total_points INTEGER DEFAULT 0,
 current_rank VARCHAR(50) DEFAULT 'Learner',
 rank_progress INTEGER DEFAULT 0,
 -- Badges (JSONB format)
);
```

```
badges JSONB DEFAULT '[]'::JSONB,

-- Streaks
current_study_streak INTEGER DEFAULT 0,
longest_study_streak INTEGER DEFAULT 0,
last_activity_date DATE,

-- Activity Stats
topics_completed INTEGER DEFAULT 0,
quizzes_taken INTEGER DEFAULT 0,
perfect_scores INTEGER DEFAULT 0,
help_given INTEGER DEFAULT 0,

updated_at TIMESTAMP DEFAULT NOW()
);
```

```
-- =====
-- REPUTATION HISTORY
-- =====
```

```
CREATE TABLE reputation_history (
id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

-- Transaction
action VARCHAR(50) NOT NULL,
points_change INTEGER NOT NULL,
points_after INTEGER NOT NULL,

-- Context
related_entity_type VARCHAR(50),
related_entity_id UUID,
description TEXT,

created_at TIMESTAMP DEFAULT NOW()
);
```

```
CREATE INDEX idx_rep_history_user ON reputation_history(user_id);
CREATE INDEX idx_rep_history_date ON reputation_history(created_at);
```

```
-- =====
-- SYSTEM CONFIGURATION
-- =====
```

```

CREATE TABLE system_config (
 key VARCHAR(100) PRIMARY KEY,
 value JSONB NOT NULL,
 description TEXT,
 updated_at TIMESTAMP DEFAULT NOW()
);

INSERT INTO system_config (key, value, description) VALUES
('pricing', '{"monthly": 25000, "annual": 250000}::JSONB, 'Subscription pricing in UGX'),
('trial_duration_days', '7'::JSONB, 'Free trial duration'),
('openai_model', '"gpt-4o-mini"'::JSONB, 'Default OpenAI model'),
('max_daily_questions_free', '10'::JSONB, 'Max AI questions for trial users'),
('max_daily_questions_paid', '999'::JSONB, 'Max AI questions for paid users'),
('referral_reward_days', '30'::JSONB, 'Days of free subscription for referrals');

-- =====
-- FUNCTIONS & TRIGGERS
-- =====

-- Auto-update updated_at timestamp
CREATE OR REPLACE FUNCTION update_updated_at_column()
RETURNS TRIGGER AS $$$
BEGIN
 NEW.updated_at = NOW();
 RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER update_users_updated_at BEFORE UPDATE ON users
 FOR EACH ROW EXECUTE FUNCTION update_updated_at_column();

CREATE TRIGGER update_topics_updated_at BEFORE UPDATE ON topics
 FOR EACH ROW EXECUTE FUNCTION update_updated_at_column();

-- Auto-generate referral code
CREATE OR REPLACE FUNCTION generate_referral_code()
RETURNS TRIGGER AS $$$
BEGIN
 IF NEW.referral_code IS NULL THEN
 NEW.referral_code := 'TUG' || UPPER(SUBSTRING(MD5(RANDOM()::TEXT) FROM 1 FOR 6));
 END IF;
 RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```
CREATE TRIGGER generate_user_referral_code BEFORE INSERT ON users
FOR EACH ROW EXECUTE FUNCTION generate_referral_code();
```

```
-- Initialize reputation for new users
```

```
CREATE OR REPLACE FUNCTION initialize_user_reputation()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
 INSERT INTO user_reputation (user_id) VALUES (NEW.id);
```

```
 RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER create_user_reputation AFTER INSERT ON users
```

```
FOR EACH ROW EXECUTE FUNCTION initialize_user_reputation();
```

```
-- =====
```

```
-- VIEWS FOR COMMON QUERIES
```

```
-- =====
```

```
-- Student Progress Overview
```

```
CREATE VIEW student_progress_overview AS
```

```
SELECT
```

```
 u.id as user_id,
```

```
 u.first_name,
```

```
 u.last_name,
```

```
 u.current_class,
```

```
 s.id as subject_id,
```

```
 s.name as subject_name,
```

```
 COUNT(DISTINCT up.topic_id) as topics_attempted,
```

```
 COUNT(DISTINCT CASE WHEN up.status = 'completed' THEN up.topic_id END) as topics_completed,
```

```
 AVG(up.best_quiz_score) as avg_quiz_score,
```

```
 SUM(up.time_spent_seconds) / 3600.0 as hours_spent
```

```
FROM users u
```

```
CROSS JOIN subjects s
```

```
LEFT JOIN user_progress up ON u.id = up.user_id AND s.id = up.subject_id
```

```
GROUP BY u.id, u.first_name, u.last_name, u.current_class, s.id, s.name;
```

```
-- Leaderboard
```

```
CREATE VIEW leaderboard AS
```

```
SELECT
```

```
 u.id,
```

```
 u.first_name || ' ' || SUBSTRING(u.last_name FROM 1 FOR 1) || '.' as display_name,
```

```
 ur.total_points,
```

```
ur.current_rank,
ur.current_study_streak,
ur.topics_completed,
ur.quizzes_taken,
ROW_NUMBER() OVER (ORDER BY ur.total_points DESC) as rank_position
FROM users u
JOIN user_reputation ur ON u.id = ur.user_id
WHERE u.subscription_status = 'active'
ORDER BY ur.total_points DESC;
...
...

```

## ## 🛡 API ENDPOINTS SPECIFICATION

### ### Base URLs

- \*\*Production:\*\* `https://api.tutoruganda.com`
- \*\*Development:\*\* `http://localhost:3000`

### ### Authentication

All protected endpoints require JWT token:

Authorization: Bearer <token>

---

### ### 1. AUTHENTICATION ENDPOINTS

#### #### POST /api/auth/register

Register new user

\*\*Request:\*\*

```
```json
{
  "phone_number": "0772123456",
  "first_name": "John",
  "last_name": "Mugisha",
  "current_class": "S3",
  "school_name": "King's College Budo",
  "region": "Kampala"
}
```

```

\*\*Response (200):\*\*

```
```json
{
  "success": true,
  "data": {
    "payment_id": "uuid",
    "transaction_id": "TUG-1702551234-uuid",
    "amount": 250000,
    "status": "pending",
    "payment_url": "https://checkout.flutterwave.com/..."
  }
}
```

```

### ### 6. PROGRESS & ANALYTICS ENDPOINTS

#### #### GET /api/users/me/progress

Get user's overall progress

\*\*Response (200):\*\*

```
```json
```

```
{  
  "success": true,  
  "data": {  
    "overall": {  
      "topics_completed": 45,  
      "total_topics": 150,  
      "completion_percentage": 30,  
      "total_study_time_hours": 28.5,  
      "study_streak_days": 12  
    },  
    "subjects": [  
      {  
        "subject_name": "Mathematics",  
        "topics_completed": 18,  
        "average_quiz_score": 82  
      }  
    ]  
  }  
}  
...  
}
```

GET /api/users/me/dashboard
Get personalized dashboard data

```
**Response (200):**  
```json  
{
 "success": true,
 "data": {
 "welcome_message": "Welcome back, John! 🔥 ",
 "study_streak": {
 "current": 12,
 "longest": 18
 },
 "quick_actions": [
 {
 "type": "continue_learning",
 "title": "Continue Quadratic Equations",
 "url": "/learn/mathematics/quadratic-equations"
 }
]
 }
}
...
```
```

🎨 FRONTEND STRUCTURE

Folder Structure

```
frontend/
  └── public/
    ├── index.html
    ├── manifest.json
    └── icons/

  └── src/
    ├── assets/
      ├── images/
      ├── icons/
      └── fonts/

    ├── components/
      ├── common/
        ├── Button.jsx
        ├── Input.jsx
        ├── Modal.jsx
        └── Loading.jsx

      ├── layout/
        ├── Header.jsx
        ├── Sidebar.jsx
        └── Footer.jsx

      ├── auth/
        ├── LoginForm.jsx
        ├── RegisterForm.jsx
        └── OTPInput.jsx

      ├── dashboard/
        ├── WelcomeCard.jsx
        ├── ProgressOverview.jsx
        └── QuickActions.jsx

      └── learning/
```

```
|- |   |- SubjectCard.jsx  
|- |   |- TopicList.jsx  
|- |   |- SubtopicContent.jsx  
  
|- chat/  
|   |- ChatInterface.jsx  
|   |- MessageBubble.jsx  
|   |- ChatInput.jsx  
  
|- quiz/  
|   |- QuizCard.jsx  
|   |- QuizInterface.jsx  
|   |- QuizResults.jsx  
  
|- pages/  
|   |- LandingPage.jsx  
|   |- LoginPage.jsx  
|   |- DashboardPage.jsx  
|   |- SubjectsPage.jsx  
|   |- ChatPage.jsx  
|   |- ProfilePage.jsx  
  
|- hooks/  
|   |- useAuth.js  
|   |- useChat.js  
|   |- useProgress.js  
  
|- store/  
|   |- slices/  
|   |   |- authSlice.js  
|   |   |- userSlice.js  
|   |   |- progressSlice.js  
|   |- store.js  
  
|- services/  
|   |- api.js  
|   |- authService.js  
|   |- chatService.js  
|   |- quizService.js  
  
|- utils/
```

```
|
|   └── constants.js
|   └── validators.js
|   └── formatters.js
|
└── App.jsx
    └── index.jsx
|
└── package.json
```

🌐 BACKEND STRUCTURE

Folder Structure

```
backend/
    └── src/
        ├── config/
        |   ├── database.js
        |   ├── redis.js
        |   └── env.js
        |
        ├── models/
        |   ├── User.js
        |   ├── Subject.js
        |   ├── Topic.js
        |   ├── ChatSession.js
        |   ├── Quiz.js
        |   └── Payment.js
        |
        ├── controllers/
        |   ├── authController.js
        |   ├── subjectController.js
        |   ├── chatController.js
        |   ├── quizController.js
        |   └── paymentController.js
        |
        └── routes/
            ├── auth.routes.js
            └── subject.routes.js
```

```
|- chat.routes.js  
|- quiz.routes.js  
|- payment.routes.js  
  
|- middleware/  
|   |- auth.js  
|   |- validation.js  
|   |- errorHandler.js  
|   |- rateLimiter.js  
|   └── subscription.js  
  
|- services/  
|   |- aiService.js  
|   |- smsService.js  
|   |- emailService.js  
|   |- paymentService.js  
|   └── cacheService.js  
  
|- utils/  
|   |- jwt.js  
|   |- otp.js  
|   └── validators.js  
  
|- jobs/  
|   |- subscriptionReminder.js  
|   └── cleanupExpiredOTP.js  
  
|- app.js  
|- server.js  
  
└── package.json
```

🔒 SECURITY CONSIDERATIONS

Authentication & Authorization

JWT Token Structure:

```
```json
{
 "user_id": "uuid",
 "phone_number": "0772123456",
 "subscription_status": "active",
 "subscription_end": "2026-12-14T00:00:00Z",
 "iat": 1702551234,
 "exp": 1703156034
}
````
```

Token Strategy:

- Access Token: 7 days
- Refresh Token: 30 days
- Auto-refresh 1 day before expiry

Data Protection

- **Password Hashing:** bcrypt (12 rounds)
- **OTP Codes:** 6 digits, expires in 5 minutes, max 3 attempts
- **Phone Verification:** Required for all users
- **SQL Injection:** Parameterized queries (ORM)
- **XSS Protection:** Input sanitization
- **HTTPS Only:** Force SSL in production
- **CORS:** Whitelist specific origins

Rate Limiting

- Auth endpoints: 5 requests/minute/IP
- AI Chat: 20 requests/minute/user
- Payment: 3 requests/minute/user

PII (Personal Info)

ENCRYPTED:

- Phone numbers (AES-256)

- Parent contact info

****HASHED:****

- Passwords (bcrypt)

****ANONYMIZED IN LOGS:****

- Phone: 0772***456

- Email: j***n@example.com

****NOT STORED:****

- OTP codes (deleted after verification)

- Payment card details

📱 MOBILE RESPONSIVENESS

Breakpoints (Tailwind)

```css

sm: '640px' // Mobile landscape

md: '768px' // Tablets

lg: '1024px' // Small laptops

xl: '1280px' // Desktops

2xl: '1536px' // Large screens

```

Key Mobile Features

- Touch-optimized UI (min 44x44px tap targets)

- Swipe gestures (navigate between topics)

- Offline mode (cache key content)

- Pull-to-refresh

- Bottom navigation

- Collapsible sections

- Mobile keyboard optimization

- Data saver mode

🚀 DEPLOYMENT GUIDE

Environment Setup

```
**Development:**
```

```
```bash
Frontend
cd frontend
npm install
npm run dev # http://localhost:5173
```

```
Backend
cd backend
npm install
npm run dev # http://localhost:3000
```
```

Railway Deployment (Backend + Database)

```
```bash
1. Install Railway CLI
npm install -g @railway/cli
```

```
2. Login
railway login
```

```
3. Initialize project
railway init
```

```
4. Add PostgreSQL
railway add postgresql
```

```
5. Add Redis
railway add redis
```

#### # 6. Set environment variables

```
railway variables set OPENAI_API_KEY=sk-xxx
railway variables set FLUTTERWAVE_SECRET_KEY=xxx
railway variables set JWT_SECRET=xxx
railway variables set NODE_ENV=production
```

```
7. Deploy
railway up
```

```
8. Run migrations
railway run npm run migrate
```

```
9. Seed database
railway run npm run seed
```

```
```
### Vercel Deployment (Frontend)
```bash
1. Install Vercel CLI
npm install -g vercel

2. Login
vercel login

3. Deploy
cd frontend
vercel --prod

4. Set environment variables
VITE_API_URL=https://api.tutoruganda.com
VITE_WS_URL=wss://api.tutoruganda.com
```

### Environment Variables

**Backend (.env):**
```bash
Server
NODE_ENV=production
PORT=3000
API_URL=https://api.tutoruganda.com
FRONTEND_URL=https://tutoruganda.com

Database
DATABASE_URL=postgresql://user:pass@host:5432/tutorUG
REDIS_URL=redis://host:6379

JWT
JWT_SECRET=your-super-secret-jwt-key
JWT_EXPIRES_IN=7d

OpenAI
OPENAI_API_KEY=sk-proj-xxxxxx
OPENAI_MODEL=gpt-4o-mini

Flutterwave
FLUTTERWAVE_PUBLIC_KEY=FLWPUBK-xxxxxx
FLUTTERWAVE_SECRET_KEY=FLWSECK-xxxxxx
```

```
Africa's Talking
AT_API_KEY=xxxxx
AT_USERNAME=tutorUG

SendGrid
SENDGRID_API_KEY=SG.xxxxxx
```

**Frontend (.env):**
```bash
VITE_API_URL=https://api.tutoruganda.com
VITE_WS_URL=wss://api.tutoruganda.com
VITE_FLUTTERWAVE_PUBLIC_KEY=FLWPUBK-xxxxx
```
```

```

## ## 📈 MONITORING & ANALYTICS

```
Application Monitoring
```javascript
// Sentry for error tracking
import * as Sentry from "@sentry/node";

Sentry.init({
  dsn: process.env.SENTRY_DSN,
  environment: process.env.NODE_ENV,
  tracesSampleRate: 1.0,
});
```
```

```

Business Metrics Dashboard

```
**DAILY:**
- New signups
- Trial to paid conversion
- Daily active users (DAU)
- AI questions asked
- Quizzes taken
```

```

```
WEEKLY:
- Weekly active users (WAU)
- Churn rate
```

```

- Revenue (MRR)
- Customer acquisition cost

MONTHLY:

- Monthly recurring revenue (MRR)
- User retention cohorts
- Feature usage
- Subject popularity

🚀 10-WEEK DEVELOPMENT TIMELINE

WEEK 1: Foundation & Setup

Backend:

- [] Setup Node.js + Express
- [] Configure PostgreSQL (Railway)
- [] Setup Redis
- [] Create database schema
- [] Setup JWT authentication
- [] Implement OTP verification

Frontend:

- [] Setup React + Vite
- [] Configure Tailwind CSS
- [] Setup routing
- [] Create basic layout
- [] Setup auth context

Deliverables:

- User registration works
- OTP verification functional
- Basic responsive layout

WEEK 2: Core Learning Features

Backend:

- [] Create curriculum models
- [] Build CRUD endpoints
- [] Seed Math curriculum (5 topics)
- [] Implement progress tracking

****Frontend:****

- [] Build Dashboard
- [] Create Subjects page
- [] Build Topic/Subtopic interface
- [] Implement progress bars

****Deliverables:****

- Browse subjects/topics
- Content displays correctly
- Progress tracking works

WEEK 3: AI Tutor Integration

****Backend:****

- [] Setup OpenAI SDK
- [] Create AI service
- [] Build chat endpoints
- [] Implement rate limiting

****Frontend:****

- [] Build Chat interface
- [] Create message bubbles
- [] Add typing indicator
- [] Implement real-time messaging

****Deliverables:****

- AI tutor responds < 3s
- Ugandan context works
- Chat history saved

WEEK 4: Quiz System

****Backend:****

- [] Create Quiz models
- [] Build quiz endpoints
- [] Implement auto-grading
- [] Seed 3 quizzes per topic

****Frontend:****

- [] Build Quiz interface
- [] Add timer
- [] Create results page
- [] Add score visualization

****Deliverables:****

- Quizzes work end-to-end
- Auto-grading accurate
- Results detailed

WEEK 5: Payment Integration

****Backend:****

- [] Setup Flutterwave
- [] Create Payment model
- [] Build payment endpoints
- [] Implement webhooks

****Frontend:****

- [] Create subscription page
- [] Add payment selection
- [] Build status page
- [] Add trial countdown

****Deliverables:****

- MTN/Airtel payments work
- Subscription activates
- Trial enforced

WEEK 6: Gamification & Progress

****Backend:****

- [] Create reputation model
- [] Build badge system
- [] Create leaderboard
- [] Track study streaks

****Frontend:****

- [] Create Profile page
- [] Build reputation display

- [] Add leaderboard
- [] Show achievements

****Deliverables:****

- Reputation system works
- Badges earned automatically
- Leaderboard updates

WEEK 7: Polish & Optimization

****Backend:****

- [] Implement caching
- [] Add compression
- [] Setup error logging
- [] Optimize queries

****Frontend:****

- [] Optimize images
- [] Add loading states
- [] Implement error boundaries
- [] Add PWA features

****Deliverables:****

- API < 500ms response
- Frontend < 3s load
- Smooth mobile UX

WEEK 8: Content Creation

****Team Task:****

- [] Complete Math curriculum (40 topics)
- [] Write 100+ Ugandan examples
- [] Create 50+ quizzes
- [] Review all content
- [] Write legal docs (Privacy, Terms)

****Deliverables:****

- Full Math curriculum live
- Content reviewed
- Legal docs ready

WEEK 9: Testing & Bug Fixes

Testing:

- [] Unit tests
- [] Integration tests
- [] E2E tests
- [] Load testing
- [] Security testing
- [] Beta testing (20-30 users)

Deliverables:

- All critical bugs fixed
- Tested on 5+ devices
- Security patched

WEEK 10: Launch Preparation

Pre-Launch:

- [] Deploy to production
- [] Setup custom domain
- [] Configure SSL
- [] Setup monitoring
- [] Test payment flow

Marketing:

- [] Create landing page
- [] Design social graphics
- [] Write announcement
- [] Prepare email campaign
- [] Record demo video

Launch Day:

- [] Final checks
- [] Send announcements
- [] Post on social media
- [] Monitor for issues
- [] Track metrics

Deliverables:

- App live in production
- First 50 users onboarded
- Support active

📈 POST-LAUNCH ROADMAP

Month 2-3: Growth

- Add Physics curriculum
- Add Chemistry curriculum
- Community chat features
- School partnerships
- Referral rewards

Month 4-6: Scale

- Add English & Biology
- Past papers integration
- Voice input/output
- Parent dashboard
- Mobile app (React Native)

Month 7-12: Expansion

- Add A-Level content
- Video tutorials
- Live tutoring sessions
- School bulk subscriptions
- Regional expansion

🎨 BRAND GUIDELINES

Colors

```
```css
/* Primary Colors */
--primary-blue: #3B82F6; /* Mathematics */
--primary-purple: #8B5CF6; /* Physics */
--primary-green: #10B981; /* Chemistry */
--primary-orange: #F59E0B; /* Biology */
--primary-red: #EF4444; /* English */
```

### /\* Neutral \*/

```
--gray-50: #F9FAFB;
```

--gray-900: #111827;

/\* Semantic \*/

--success: #10B981;

--error: #EF4444;

---

### ### Typography

```css

font-family: 'Inter', sans-serif;

--text-xs: 0.75rem;

--text-sm: 0.875rem;

--text-base: 1rem;

--text-xl: 1.25rem;

--text-4xl: 2.25rem;

Logo

- **Primary:** TutorUG (graduation cap icon)

- **Tagline:** "Your AI-Powered Path to UNEB Excellence"

- **Icon:** Graduation cap + Uganda flag colors

SUPPORT & MAINTENANCE

Customer Support

- **WhatsApp:** +256 XXX XXX XXX

- **Email:** support@tutoruganda.com

- **In-app:** Chat support

- **FAQ:** Help center

Support Hours

- Mon-Fri: 8 AM - 8 PM EAT

- Saturday: 9 AM - 6 PM EAT

- Sunday: 10 AM - 4 PM EAT

SLA

- Critical (payment, login): < 2 hours

- High (content errors): < 4 hours
- Medium (features): < 24 hours
- Low (general): < 48 hours

LAUNCH CHECKLIST

Technical

- [] All API endpoints tested
- [] Database migrations run
- [] SSL certificates installed
- [] Environment variables set
- [] Monitoring active
- [] Backups configured

Content

- [] Math curriculum complete
- [] 100+ examples written
- [] 50+ quizzes created
- [] Content proofread

Legal

- [] Terms of Service
- [] Privacy Policy
- [] Data protection measures
- [] Refund policy

Payment

- [] Flutterwave production mode
- [] Webhooks tested
- [] Receipts working
- [] Subscription logic verified

Marketing

- [] Landing page live
- [] Social accounts created
- [] Announcement ready
- [] Demo video recorded

Team

- [] Support trained
- [] Response templates ready
- [] On-call schedule set

- [] Celebration planned! 🎉

🎉 FINAL WORDS

You now have everything you need to build TutorUG Secondary!

This comprehensive document covers:

- Complete database schema
- Full API specification
- Frontend/Backend architecture
- 10-week timeline
- Deployment guide
- Testing strategy
- Business metrics

Next Steps:

1. Review with your team
2. Setup development environment
3. Start Week 1 tasks
4. Ship weekly, iterate fast
5. Launch in 10 weeks!

Remember:

- Start small (Math only)
- Ship fast, iterate faster
- Listen to users
- Focus on quality
- Celebrate small wins

Let's make education accessible to every Ugandan student! UG🚀

💼 PRICING SUMMARY

| Plan | Price | Duration | Savings |
|-------------|-------------|-----------|----------------------------|
| **Monthly** | 25,000 UGX | 1 month | - |
| **Annual** | 250,000 UGX | 12 months | 50,000 UGX (2 months free) |

Trial Period: 7 days free

Cost Per Day: Less than 1,000 UGX

****Target:**** Less than 1/5 the cost of private tuition

Document Version: 1.0

Last Updated: December 14, 2025

Prepared by: TutorUG Team200):******

```json

```
{  
  "success": true,  
  "message": "OTP sent to your phone",  
  "data": {  
    "verification_id": "uuid",  
    "expires_in": 300  
  }  
}
```

```

POST /api/auth/verify-otp

Verify OTP and complete registration

****Request:****

```
```json  
{
 "phone_number": "0772123456",
 "otp_code": "123456"
}
```
```

****Response (200):****

```
```json  
{
 "success": true,
 "message": "Account created successfully",
 "data": {
 "user": {
 "id": "uuid",
 "phone_number": "0772123456",
 "first_name": "John",
 "subscription_status": "trial",
 "trial_end_date": "2025-12-21T00:00:00Z"
 },
 "token": "jwt_token_here",
 "expires_in": 604800
 }
```
```

```
    }
}
```
POST /api/auth/login
Login existing user

Request:

```json
{
  "phone_number": "0772123456"
}
```

```

```
Response (200):

```json
{
  "success": true,
  "message": "OTP sent to your phone",
  "data": {
    "verification_id": "uuid",
    "expires_in": 300
  }
}
```

```

## ## 2. SUBJECTS & CURRICULUM ENDPOINTS

```
GET /api/subjects
Get all available subjects

Response (200):

```json
{
  "success": true,
  "data": [
    {
      "id": "uuid",
      "name": "Mathematics",
      "code": "MATH",
      "description": "Numbers, Algebra, Geometry",
      "icon_url": "https://cdn.tutoruganda.com/icons/math.svg",
    }
  ]
}
```

```

```
"color_hex": "#3B82F6",
"topics_count": 40,
"user_progress": {
 "completed_topics": 15,
 "completion_percentage": 37.5
}
}
]
}
}
...
}
```

#### GET /api/subjects/:subjectId/topics

Get all topics for a subject

\*\*Query Params:\*\*

- `class` (optional): S1, S2, S3, S4
- `term` (optional): 1, 2, 3

\*\*Response (200):\*\*

```
```json
{
  "success": true,
  "data": {
    "subject": {
      "id": "uuid",
      "name": "Mathematics"
    },
    "topics": [
      {
        "id": "uuid",
        "title": "Algebra",
        "slug": "algebra",
        "difficulty": "intermediate",
        "estimated_duration_minutes": 180,
        "user_progress": {
          "status": "in_progress",
          "completion_percentage": 62
        }
      }
    ]
  }
}
...
}
```

3. AI TUTOR ENDPOINTS

POST /api/chat/sessions

Start new chat session

Request:

```
'''json
{
  "subject_id": "uuid",
  "topic_id": "uuid",
  "session_type": "learning"
}
'''
```

Response (200):

```
'''json
{
  "success": true,
  "data": {
    "session_id": "uuid",
    "welcome_message": "Hi John! 🤖 I'm here to help you with Mathematics."
  }
}
'''
```

POST /api/chat/sessions/:sessionId/messages

Send message to AI tutor

Request:

```
'''json
{
  "content": "How do I solve quadratic equations?",
  "context": {
    "current_subtopic_id": "uuid"
  }
}
'''
```

Response (200):

```
'''json
{
  "success": true,
```

```
"data": {
  "message_id": "uuid",
  "role": "assistant",
  "content": "Let me explain quadratic equations...",
  "suggestions": [
    "Show me an example",
    "Give me practice questions"
  ]
}
}
}
...
---
```

4. QUIZ ENDPOINTS

GET /api/quizzes/topic/:topicId

Get quizzes for a topic

****Response (200):****

```
```json
{
 "success": true,
 "data": [
 {
 "id": "uuid",
 "title": "Quadratic Equations - Practice Quiz",
 "total_questions": 10,
 "passing_score": 70,
 "time_limit_minutes": 15,
 "attempts_count": 2,
 "best_score": 85
 }
]
}
...
```
```

POST /api/quizzes/attempts/:attemptId/submit

Submit quiz answers

****Request:****

```
```json
{
 "answers": {
 ...
 }
}
...
```
```

```
"q1": { "selected": 0, "time_spent": 25 },
"q2": { "selected": 2, "time_spent": 45 }
}
}
```
Response (200):
```json
{
  "success": true,
  "data": {
    "score": 85,
    "marks_obtained": 8.5,
    "passed": true,
    "detailed_results": [
      {
        "question_id": "q1",
        "is_correct": true,
        "explanation": "Correct! Factor as (x-2)(x-3) = 0..."
      }
    ],
    "reputation_earned": 15
  }
}
```
```
---
```

5. PAYMENT & SUBSCRIPTION ENDPOINTS

```
#### GET /api/subscriptions/plans
Get available subscription plans
```

Response (200):

```
```json
{
 "success": true,
 "data": {
 "plans": [
 {
 "id": "monthly",
 "name": "Monthly Plan",
 "price": 25000,
 "currency": "UGX",
 }
]
 }
}
```
---
```

```
"duration_months": 1
},
{
  "id": "annual",
  "name": "Annual Plan",
  "price": 250000,
  "currency": "UGX",
  "duration_months": 12,
  "savings": 50000,
  "badge": "BEST VALUE"
}
]
}
}
```

```

#### POST /api/payments/initiate

Initiate payment

\*\*Request:\*\*

```
```json
{
  "plan_id": "annual",
  "payment_method": "mtn_mobile_money",
  "phone_number": "0772123456"
}
```

```

\*\*Response (