# Children's Learning Platform

*AI-Based Personalized Learning for Bangladeshi Children*

**Week 9 Presentation**

Requirement Analysis & System Architecture

Group 11 — IND-1 (Optimizely) — BUET CSE

# Solving the Complex Engineering Problem (CEP)

## The Challenge: Static & Unmonitored Learning

- **Content Gap:** Standard AI lacks NCTB curriculum context and ”hallucinates” facts.
  - **Parental Concern:** Risk of unmonitored AI interaction and lack of progress tracking.
  - **Inactivity:** Static video lessons fail to verify if a child actually understood the topic.

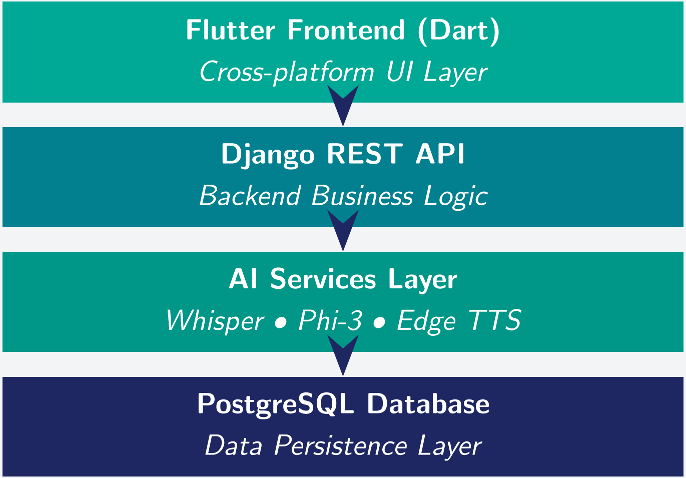
## Our Solution: Adaptive AI Voice Tutor

- **RAG Pipeline:** Grounding **Phi-3** with NCTB books for 100% academic accuracy.
  - **Active Assessment:** AI initiates 2-way Bangla dialogue to verify lesson mastery.
  - **Personalization:** Adaptive knowledge-tracking based on student’s current level.
  - **Parental Gate:** Active monitoring dashboard for progress and safety control.

# Presentation Agenda

1. System Architecture Overview
2. Data Pipeline & Framework
3. UML Diagrams
4. Database Schema
5. API Endpoints
6. Technology Stack Justification
7. Mock UI Demonstration

# System Architecture Overview



**Flutter Frontend (Dart)**

*Cross-platform UI Layer*

**Django REST API**

*Backend Business Logic*

**AI Services Layer**

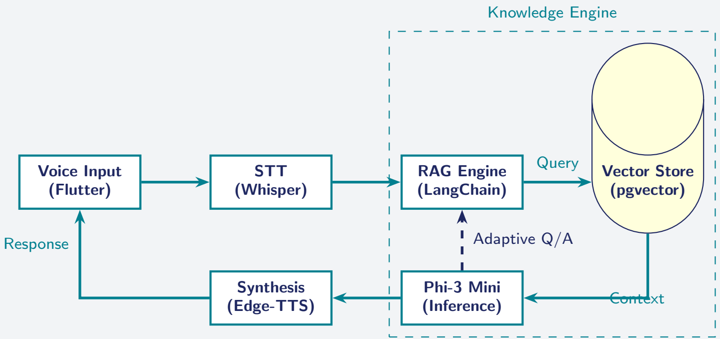
*Whisper • Phi-3 • Edge TTS*

**PostgreSQL Database**

*Data Persistence Layer*

# Data Pipeline: Knowledge Retrieval and Assessment

## End-to-End Semantic Flow



- **Vector Database:** NCTB content stored as embeddings in **pgvector** for semantic search.
- **Adaptive Assessment:** AI evaluates student's answer against the retrieved book context.
- **Active Learning:** Model initiates questions based on lesson completion to verify mastery.

# Integrated Framework Architecture

## Backend: Django REST

- **Core Logic:** Managing API endpoints and User Authentication.
- **AI Orchestration:** Using LangChain to bridge PostgreSQL and Ollama.
- **Media Management:** Handling static assets via Cloudinary.

## Frontend: Flutter

- **State Management:** Provider/Bloc for reactive UI updates.
- **Hardware Access:** Native plugins for low-latency audio recording.
- **UX Design:** Child-friendly, high-contrast visual elements.

**Database Strategy: PostgreSQL for Structured Data + pgvector for Unstructured Text Embeddings**

# Justification: Frontend and Backend

## Strategic Choice: Mobile and Logic Layer

- **Flutter (Mobile-First Approach)**
  - Children primarily interact via **Tablets and Smartphones**.
  - Smooth animations and interactive UI to maintain engagement.
  - Single codebase for both Android and iOS platforms.
- **Django (AI-Ready Backend)**
  - Native support for Python-based **AI/ML libraries**.
  - Built-in User Authentication and Content Management.
  - Fast development cycle for complex REST APIs.

# Justification: AI Model and Database

## Strategic Choice: Intelligence and Data Layer

- **Phi-3 Mini (Small Language Model)**
  - Optimized for high reasoning in a **local environment**.
  - Runs on standard CPU/Hardware; no expensive GPUs needed.
  - Ensures consistent performance without external API lag.
- **PostgreSQL + pgvector (Unified DB)**
  - Stores both **Structured Data** and **Vector Embeddings**.
  - Reduces architectural complexity by avoiding multiple DBs.
  - Reliable, industry-standard persistence layer.

# Deployment Stack

## Our Production Infrastructure

### Server Infrastructure

- **Cloud Server:** Linux VPS for 24/7 global availability.
- **Web Server: Nginx** for security and request handling.
- **WSGI: Gunicorn** as the Python application server.

### Operations AI

- **Docker:** Containerizing all services for consistency.
- **Ollama:** Local serving engine for Phi-3 SLM.
- **Media:** Cloudinary for efficient asset hosting.

# Deployment Strategy  Benefits

## Why This Deployment Approach?

- **Data Privacy  Security**
  - All AI processing happens locally; no data leaves the server.
- **Cost Optimization**
  - Eliminates recurring token-costs of proprietary models.
- **Reliability  Portability**
  - Docker-based setup allows deployment on any VPS or Local Intranet.
- **Edge Deployment Ready**
  - Can be hosted on-premise in schools with limited internet.

# UML Diagrams

**Class Diagram**

*17 classes with attributes, methods, relationships*

**ER Diagram**

*Database entities and relationships*

**Sequence Diagrams**

*Authentication, Enrollment, Quiz, AI Tutor*

**Use Case Diagram**

*Parent, Student, AI System interactions*

✓ *All diagrams available in documentation*

# Database Schema – PostgreSQL

17 Tables with Relationships

### Core Users

- Parent
- Student

### Course Management

- Course
- Content
- Enrollment
- Progress

### Assessment

- Quiz
- QuizAttempt

### AI & Analytics

- AITutorSession
- LearningAnalytics
- ContentEngagement

### Resources

- Resource
- ResourceInteraction

### Gamification

- StudentRating
- DailyGoals
- LearningPath

### Communication

- Notification

# API Endpoints – Auth & Parent Dashboard

*6 Categories • 20 Total Endpoints*

| Endpoint | Method | Purpose | Request / Response |
|---|---|---|---|
| **1. Authentication & User Management** | | | |
| /api/auth/register/parent | POST | Register new parent account | Req: {name, email, phone, password} |
| /api/auth/login/parent | POST | Authenticate parent login | Req: {email,password} Res: {token,user_role} |
| /api/auth/login/student | POST | Authenticate student login | Req: {username,password} Res: {token,child_id} |
| /api/auth/logout | POST | End the user session | Res: {success: true} |
| **2. Parent Dashboard** | | | |
| /api/parent/children | GET | Get list of linked children | Res: {id, name, grade, xp} |
| /api/parent/children | POST | Add a new child profile | Req: {name, age, grade, username} |
| /api/parent/notifications | GET | Fetch achievement alerts | Res: {id, type, message, date} |
| /api/parent/settings | PUT | Update parental controls | Req: {screen_time, sound_enabled} |

# API Endpoints – Student & Courses

| Endpoint | Method | Purpose | Request / Response |
|---|---|---|---|
| **3. Student Dashboard & Gamification** | | | |
| /api/student/profile | GET | Get student stats (XP, Streak) | Res: {name, level, xp, streak_days} |
| /api/student/xp | POST | Update XP after activity | Req: {activity_type, xp_gained} |
| /api/student/streak | POST | Update daily streak login | Res: {current_streak: 8} |
| **4. Courses & Learning Content** | | | |
| /api/courses | GET | Get enrolled courses | Res: {id, title, progress, icon} |
| /api/library | GET | All available courses (Parent) | Res: {id, title, grade_level} |
| /api/enroll | POST | Enroll student in a course | Req: {child_id, course_id} |
| /api/courses/{id}/lessons | GET | Get lesson list for course | Res: {lesson_id, title, type} |
| /api/courses/{id}/resources | GET | Get Textbook/Audiobook links | Res: {textbook_url, audiobook_url} |

# API Endpoints – Games, Quizzes & AI Tutor

| Endpoint | Method | Purpose | Request / Response |
|---|---|---|---|
| **5. Interactive Games & Quizzes** | | | |
| `/api/quiz/{subject}` | GET | Fetch quiz questions for subject | Res: {question, options:[], answer} |
| `/api/quiz/submit` | POST | Submit quiz, calculate score | Req: {quiz_id, answers:[]}<br>Res: {score, xp_awarded} |
| `/api/games/{game_id}/data` | GET | Get game config data | Res: {config_data, difficulty_level} |
| **6. AI Tutor (Chat)** | | | |
| `/api/ai/chat` | POST | Send message to AI, get reply | Req: {message, context:’math’}<br>Res: {reply_text, suggested_activity} |

**Auth: 4 • Parent: 4 • Student: 3 • Courses: 5 • Games/Quiz: 3 • AI Chat: 1 = 20 Total Endpoints**

# Mock UI Demonstration

**Interactive Math Puzzles**

Fill-in-the-blank multiplication grids

**Pronoun Detective Quest**

Tap words in story
to identify pronouns

**Grammar Drag-and-Drop**

Drag auxiliary verbs
into sentences

**Math Tree Adventure**

Visual multiplication learning

**AI Voice Tutor**

Bilingual voice
conversation

**Progress Dashboard**

Parent & student analytics

[WEB] *Full interactive demo available in HTML file*

# Summary

✓ Comprehensive system architecture with 4-layer design

✓ Real-time AI pipelines for voice, quiz, and analytics

✓ 17 database tables with full relationships

✓ 29 REST API endpoints across 9 categories

✓ Justified tech stack: Django, Flutter, PostgreSQL, AI services

✓ Cost-effective deployment: $0.60 per student annually

✓ Interactive UI with gamification and hands-on learning

**Ready for Implementation Phase**

# Thank You

*Questions & Discussion*

Group 11 — IND-1 (Optimizely)

Children's Learning Platform

Bangladesh University of Engineering and Technology