PES UNIVERSITY

UE22CS320A- Project Phase - 2

Capstone Project –Phase 2 –review 1

Project Title : Intelligent Concept Acquisition System - An AI-Driven Platform for Concept Mastery and Personalized Knowledge Enhancement

Project ID : 170

Project Guide : Prof. Sheela Devi

Project Team : Naveen Radhakrishnan - PES2UG22CS344
Nitish Kumar - PES2UG22CS374
Rohit Yakkundi - PES2UG23CS819
Sharan Surpur - PES2UG23CS821

Agenda

- Problem Statement
- Abstract and Scope
- Suggestions from Phase-1
- Requirements Specification
- Design Approach
- Design Constraints, Assumptions & Dependencies
- Proposed System / Approach
- Architecture
- Design Description
- Project Progress
- References

Problem Statement

Students struggle with personalized learning during self-study, especially when preparing for exams, due to the lack of adaptive assessment tools and intelligent feedback mechanisms. Traditional educational platforms provide static content that does not analyze individual learning patterns, making it difficult for students to identify weak concepts, track progress, and receive tailored guidance.

Without an AI-driven system that can scan study materials, generate targeted questions, and adapt teaching strategies based on student performance, learners face inefficient study sessions, slower progress, and reduced academic confidence. To address these challenges, an intelligent learning platform is required to provide real-time feedback, dynamic assessments, and personalized study plans to ensure efficient concept mastery and continuous improvement.

Abstract and Scope

The growing demand for personalized education has highlighted the shortcomings of traditional learning platforms, which often lack adaptive assessment tools and interactive feedback systems. This project introduces an AI-powered learning platform that revolutionizes self-study by enabling students to upload study materials, which the AI analyzes, extracts key concepts, and generates targeted assessment questions.

The system intelligently adapts based on student performance, adjusting question difficulty, scheduling study sessions, and providing real-time feedback to enhance concept retention. By bridging knowledge gaps through customized learning paths, this solution empowers students to track progress, refine their understanding, and achieve academic success efficiently.

Abstract and Scope

Scope:

- Custom Question Papers:
  Generates personalized assessments based on user-uploaded study materials, focusing on individual learning needs to strengthen key areas of knowledge.

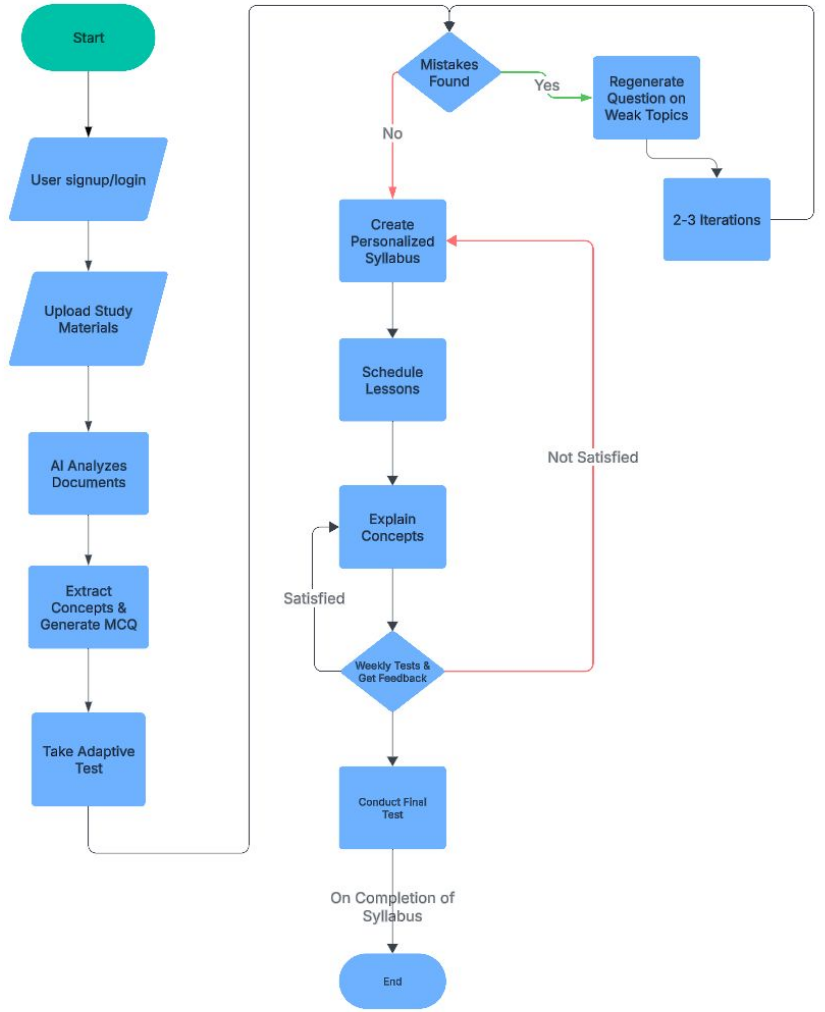
- Targeted Concept Reinforcement:
  Identifies specific areas requiring improvement and offers tailored exercises and practice questions, ensuring that students focus on concepts needing the most attention.

- Interactive Teaching:
  Employs AI-driven virtual instructors to conduct dynamic learning sessions, delivering real-time, customized feedback to keep students engaged and support different learning styles.

- Progressive Learning Pathways:
  Adjusts content and pace based on assessment results, ensuring mastery of fundamental concepts before progressing to more advanced material, thereby reducing knowledge gaps.

Architecture

Flow Chart



Suggestions from Phase-1

Panel Concern:
The panel highlighted the potential overlap between the proposed system and existing large language models (LLMs) such as Chat GPT,Gemini and Deepseek, questioning what sets the project apart.

Response:
While LLMs like ChatGPT and Gemini provide generalized responses, our system is designed specifically for educational purposes. Key differentiators include:

- **Domain-Specific Focus:** Tailored algorithms analyze and generate content based on user-uploaded study materials, ensuring relevance to specific subjects and exam needs.
- **Targeted Feedback:** The platform identifies individual learning gaps through assessments and provides focused, concept-specific feedback rather than generic responses.
- **Progressive Learning Pathways:** Unlike general-purpose LLMs, our system tracks long-term progress and adapts the learning trajectory to ensure mastery before advancing to complex topics.

Suggestions from Phase-1

Panel Concern:
The panel questioned whether the proposed adaptive learning mechanisms, personalized feedback, and dynamic content generation could be realistically implemented.

Response:
Several measures are being taken to ensure feasibility:

- **Modular Implementation:** The system is being developed in phases, starting with core features like personalized question generation and targeted reinforcement. This approach allows us to validate each component before integrating more advanced features.
- **AI and Machine Learning Techniques:** Existing AI models, such as recommendation algorithms and NLP for content generation, are being adapted to suit the educational context. Prototyping and pilot testing will help refine these methods.
- **Scalability Testing:** Initial development focuses on a controlled dataset and user group, allowing us to test system performance and scalability in a manageable environment before broader deployment.

Requirements Specification

- **Hardware:**
  - Local server for backend AI processing and database storage.
  - User access through a public-facing UI hosted via a web server.
  - Devices: Desktop, laptop, or mobile with an internet connection.

- **Software & Technologies:**
  - **Programming Languages:** Python (AI/ML, backend), JavaScript (frontend).
  - **AI/ML Libraries:** TensorFlow, PyTorch for adaptive learning models.
  - **NLP Tools:** SpaCy, Hugging Face transformers for text analysis.
  - **Database:** PostgreSQL/MySQL (structured data), MongoDB (unstructured data).
  - **Frameworks:** Django/Flask (backend), React/Next.js (frontend).
  - **Hosting:**
    - Backend services and database hosted on a local server.
    - Frontend UI exposed via an Nginx/Apache reverse proxy for public access.
  - **Dataset:** ALIN2022 for training the AI models.

Requirements Specification

Why These Technologies?

- **Python:** Strong ML support and flexibility.
- **TensorFlow/PyTorch:** Robust AI model training and adaptability.
- **PostgreSQL/MySQL:** Reliable, scalable relational database.
- **React:** Interactive and dynamic frontend for users.
- **Nginx/Apache:** Secure and efficient UI exposure while keeping backend local.

Design Approach

Adopted Approach:

- Modular architecture with independent components for:
  - Adaptive learning engine.
  - Automated assessment generation.
  - Real-time feedback processing.
  - Secure UI-to-backend communication.

Design Approach

Infrastructure Design:

- **Local Backend for Security & Control:** AI processing and database remain **on-premises** to ensure data privacy.
- **Public UI Exposure:** A web server (Nginx/Apache) allows controlled user access to the platform.
- **Scalability Consideration:** The system is designed with the ability to scale up infrastructure if needed:
  - **Horizontal Scaling:** Additional local servers can be added to distribute computational workload.
  - **Hybrid Cloud Option:** If demand grows, selected components (like UI hosting or AI inference) can be migrated to a cloud-based infrastructure while keeping sensitive data local.
  - **Load Balancing:** Future scalability plans include a load balancer to efficiently handle multiple user requests.

Design Approach

Benefits:

- **Personalization:** AI dynamically tailors learning content.
- **Security:** Local backend ensures data privacy and control.
- **Scalability-Ready:** Infrastructure can expand based on demand without major redesign.

Drawbacks:

- **Resource Management:** Local AI computations require high-performance hardware.
- **Initial Hardware Costs:** Future scalability may require additional investment in local server capacity.

Alternative Approaches Considered:

- **Fully Cloud-Based Infrastructure** (rejected due to privacy concerns).
- **Standalone Local App** (not ideal for public accessibility).

Design Constraints, Assumptions & Dependencies

Design Constraints:

- Local Backend:
  - Requires dedicated server maintenance and monitoring.
  - Computational load must be optimized for AI inference.
- Public UI Exposure:
  - Needs a secure API gateway to prevent unauthorized access.
  - Load balancing mechanisms may be required for high traffic.
- Limited Dataset Availability:
  - AI personalization relies on quality training data (ALIN2022).

Design Constraints, Assumptions & Dependencies

Assumptions:

- Users will interact with the system through the public UI.
- AI-generated assessments will effectively adapt to student needs.
- The local server will handle concurrent processing efficiently.

Dependencies:

- Reverse Proxy (Nginx/Apache) for routing UI requests securely to the backend.
- Authentication Mechanism (OAuth, JWT) to manage user access.
- Data Processing Pipeline to handle AI-generated feedback and question papers.

Design Details

1. Platforms & Systems

- Backend: Django/Flask (Python) running on a local server for AI processing and database management.
- Frontend: React/Next.js for a responsive and dynamic user interface, accessible via a web server.
- Database: PostgreSQL/MySQL (structured data) and MongoDB (unstructured data).
- AI/ML Models: TensorFlow/PyTorch for adaptive learning and NLP-based question generation.
- Hosting: Backend remains local, while the UI is exposed via an Nginx/Apache reverse proxy.

Design Details

2. Key Design Considerations

Novelty & Innovativeness

- AI-driven **adaptive learning paths** for personalized content recommendations.
- Dynamic **assessment generation** using NLP-based techniques.
- **Real-time performance tracking** for students through an interactive dashboard.

Interoperability

- REST API-based backend allows integration with other educational tools and platforms.
- Supports **multiple devices (desktop, tablet, mobile)** with a responsive UI.

Performance & Scalability

- **Efficient AI inferencing** using model quantization and caching.
- **Load balancing** mechanisms (Nginx) can be introduced for handling higher traffic.
- **Horizontal scalability**: More local servers can be added if computational demand increases.

Design Details

Security & Reliability

- **Authentication:** OAuth/JWT ensures secure user logins.
- **Data Encryption:** SSL/TLS for secure communication.
- **API Gateway:** Controls backend access, preventing unauthorized requests.

Maintainability & Portability

- **Modular architecture** allows independent updates to UI, backend, and AI models.
- **Portable Backend:** Can be deployed on local infrastructure or cloud if needed.
- **Minimal Downtime:** System updates can be applied with minimal service interruptions.

Legacy to Modernization & Reusability

- Designed to be **future-proof**, allowing integration with new AI models.
- Reusable **learning modules and assessment components** can be adapted for different subjects.
- Supports **continuous learning improvements** based on user performance analytics.

Application Compatibility

- Compatible with **modern web browsers** for seamless user access.
- AI model training can leverage **existing educational datasets** to expand system capabilities.

Design Details

Vital Changes (If Needed)

- **Database Sharding:** If user data grows, implement **partitioning strategies** for better performance.
- **Hybrid Cloud Option:** If local servers cannot handle increasing demand, migrate some components to the cloud.
- **Enhanced AI Models:** Regular updates to the adaptive learning algorithms based on real-world student interactions.

Proposed System / Approach

1. System Overview

- AI-driven adaptive learning platform that personalizes content based on student performance.
- Uses NLP and machine learning models to generate custom assessments and feedback dynamically.
- Ensures students follow a progressive learning pathway, mastering foundational concepts before advancing.

Proposed System / Approach

2. Core Functionalities

1. **Content Processing:**
   - AI analyzes **user-uploaded study materials** to extract key concepts.
   - Generates **contextual questions** based on different difficulty levels.
2. **Dynamic Assessments:**
   - Automatically creates **custom question papers** tailored to the student's knowledge level.
   - Tracks student responses and identifies conceptual gaps.
3. **Personalized Learning Paths:**
   - Adaptive system **adjusts content recommendations** based on test performance.
   - Provides **interactive feedback** to reinforce weak areas.
4. **Real-Time Progress Tracking:**
   - Dashboard visualizes **learning progress** and concept mastery.
   - Helps students understand areas that need improvement.

Proposed System / Approach

1. Technical Approach

- **Natural Language Processing (NLP):**
  - Extracts key topics from uploaded study material.
  - Generates concept-based questions automatically.
- **Machine Learning Models:**
  - Analyzes student responses and identifies learning gaps.
  - Adjusts question difficulty based on student performance.
- **User Interaction & Feedback Mechanism:**
  - Provides instant explanations for answers.
  - Suggests additional study material based on weak areas.

Proposed System / Approach

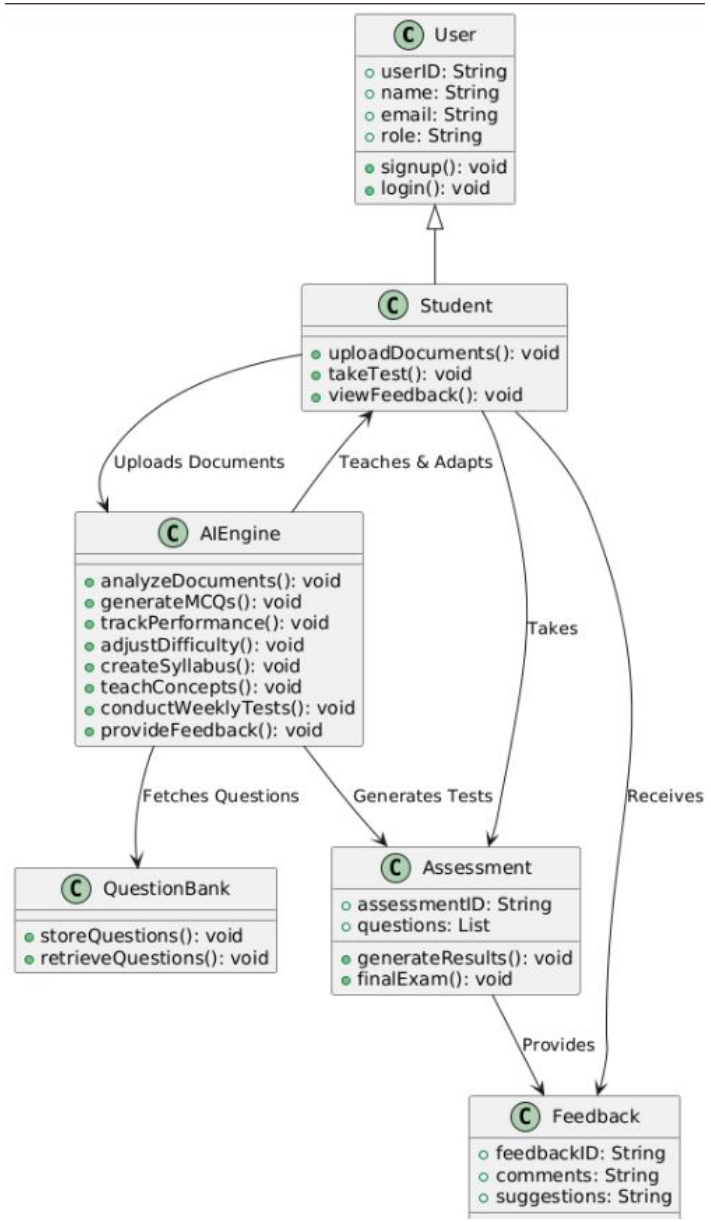
2. Infrastructure Plan

- **Local Backend Deployment:**
  - AI models and database run on a local server for security.
  - Ensures **data privacy and full control over processing**.
- **Public UI Exposure:**
  - Users access the system via a web-based interface.
  - **Nginx/Apache reverse proxy** handles secure routing.

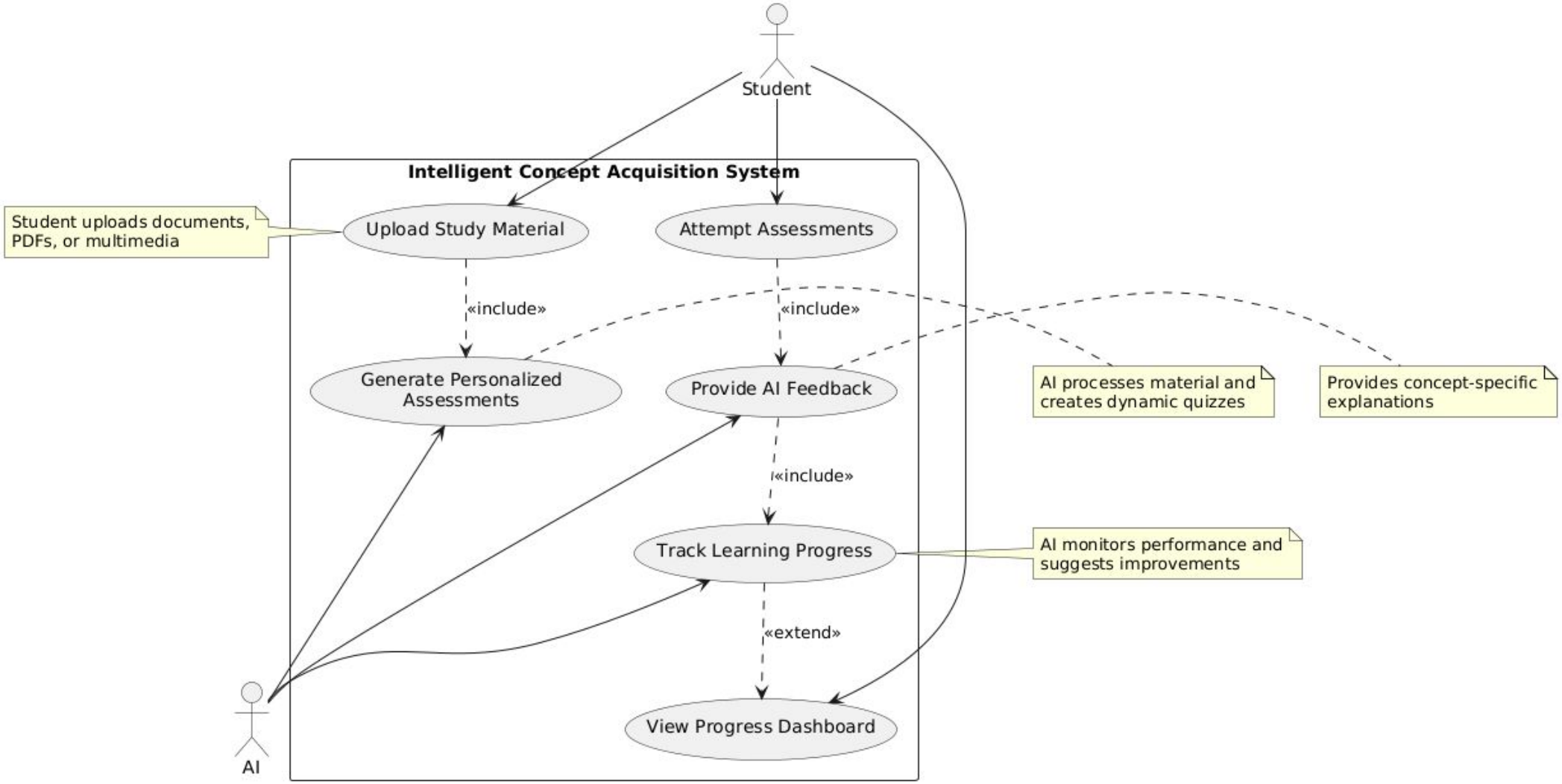
3. Expected Benefits

1. **Personalized Learning Experience** – Custom assessments help students focus on their weak areas.
2. **Efficient Self-Study** – Adaptive pathways ensure systematic learning.
3. **Scalability & Security** – Localized backend with a public UI ensures **controlled** growth.

Master class diagram



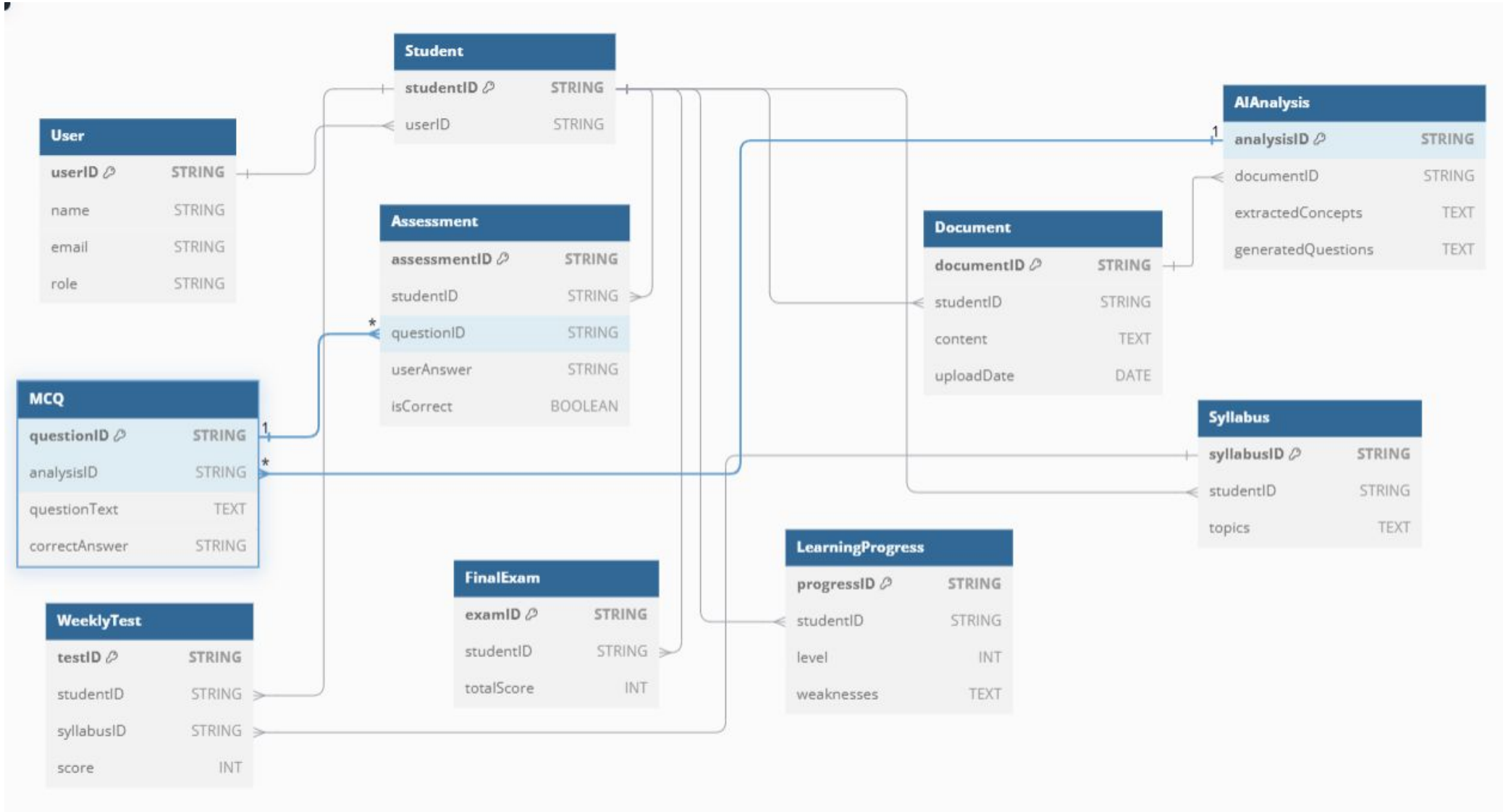
User Interface Diagrams/ Use Case Diagrams



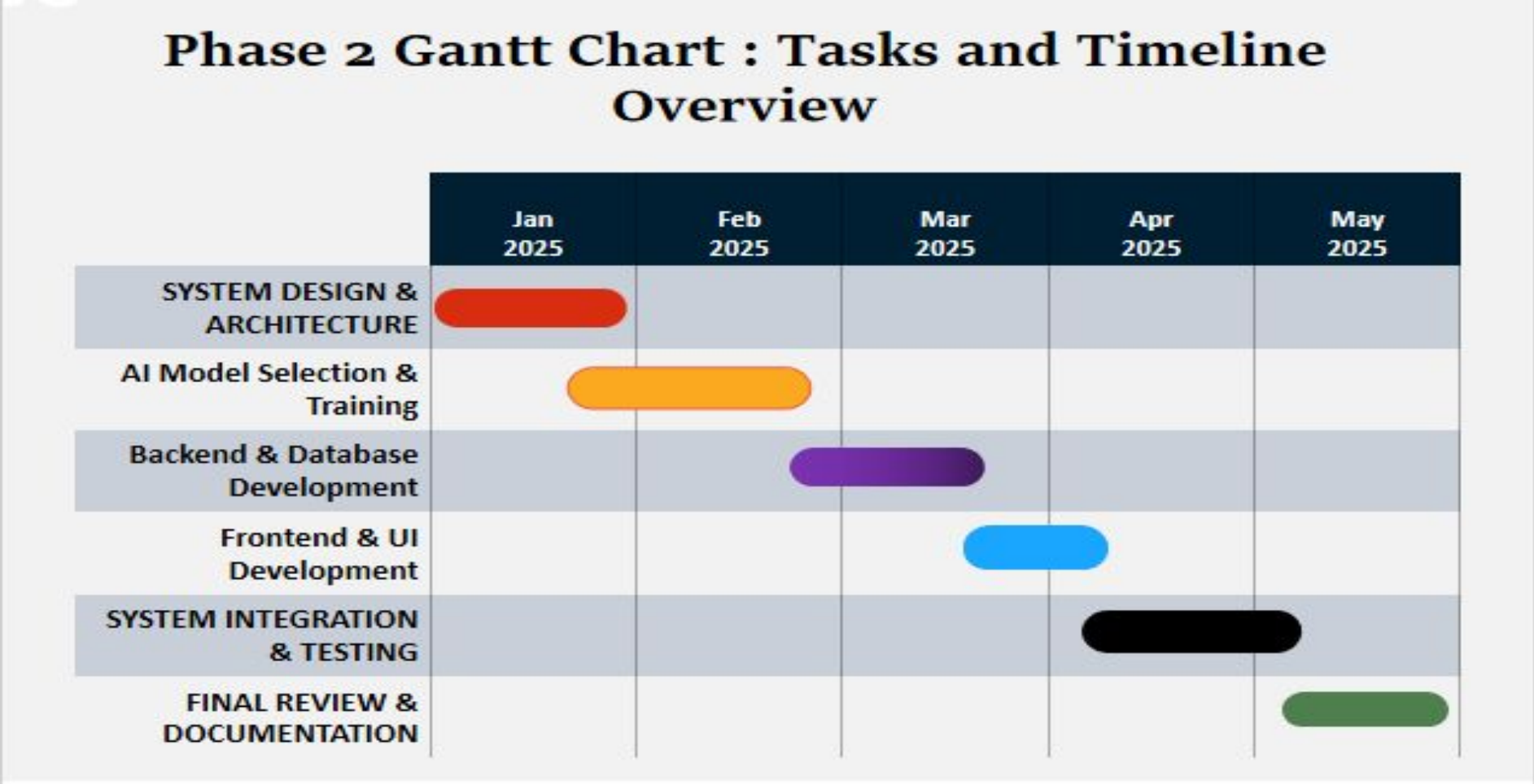
External Interfaces

- **User Interface (React.js / Next.js)** – Handles student interactions and study material uploads.
- **API Services (FastAPI / Flask)** – Facilitates AI processing and communication between components.
- **Database (PostgreSQL / Firebase)** – Stores assessments, user progress, and feedback.
- **AI Models (GPT-4, BERT, XGBoost)** – Powers question generation, adaptive learning, and feedback processing.
- **Cloud Services (AWS / Google Cloud)** – Provides scalability and smooth AI model execution.

ER Diagram


Student
studentID STRING
userID STRING
User
userID STRING
name STRING
email STRING
role STRING
Assessment
assessmentID STRING
studentID STRING
questionID STRING
userAnswer STRING
isCorrect BOOLEAN
MCQ
questionID STRING
analysisID STRING
questionText TEXT
correctAnswer STRING
WeeklyTest
testID STRING
studentID STRING
syllabusID STRING
score INT
FinalExam
examID STRING
studentID STRING
totalScore INT
LearningProgress
progressID STRING
studentID STRING
level INT
weaknesses TEXT
Document
documentID STRING
studentID STRING
content TEXT
uploadDate DATE
AIAnalysis
analysisID STRING
documentID STRING
extractedConcepts TEXT
generatedQuestions TEXT
Syllabus
syllabusID STRING
studentID STRING
topics TEXT
1
*
1
*
1

Capstone Phase-II Project Timeline



Conclusion

This research project developed an AI-driven adaptive learning platform to enhance personalized education by generating questions and feedback from user-provided study materials. By analyzing documents content, the system tailors learning experiences to address individual needs, bridging conceptual gaps and providing customized feedback. This approach aims to improve self-study effectiveness and student engagement.

Despite limitations such as data variability and a small pilot user base, the platform demonstrates the potential of AI in transforming education. It lays a strong foundation for future research, suggesting that AI-driven tools can create more personalized and effective learning environments, ultimately enhancing educational outcomes.

References

[1]S. -C. Kong and Y. Yang, "A Human-Centered Learning and Teaching Framework Using Generative Artificial Intelligence for Self-Regulated Learning Development Through Domain Knowledge Learning in K–12 Settings," in *IEEE Transactions on Learning Technologies*, vol. 17, pp. 1588-1599, 2024, doi: 10.1109/TLT.2024.3392830.

[2]M. Zafari, J. S. Bazargani, A. Sadeghi-Niaraki and S. -M. Choi, "Artificial Intelligence Applications in K-12 Education: A Systematic Literature Review," in *IEEE Access*, vol. 10, pp. 61905-61921, 2022, doi: 10.1109/ACCESS.2022.3179356.

[3]I. Gligorea, M. Cioca, R. Oancea, A.-T. Gorski, H. Gorski, and P. Tudorache, “Adaptive Learning Using Artificial Intelligence in e-Learning: A Literature Review,” *Education Sciences*, vol. 13, no. 12, pp. 1216–1216, Dec. 2023, doi: https://doi.org/10.3390/educsci13121216.

References

[4]S. Chen, Z. Wang, and M. Zhang, "Supporting Teachers' Professional Development WithGenerative AI: The Effects on Higher Order Thinkin*earning Technologies*, vol. 17, no. 1, pp. 1-12, Feb. 2024. doi: 10.1109/TLT.2024.3369690

[5]A. M. Turing, "Computing Machinery and Intelligence," *Mind*, vol. 59, no. 236, pp. 433-460, 1950. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X22000650. Accessed: Feb. 12, 2025.

[6]Effects of AI-Based Personalized Adaptive Learning System in Higher Education

Y. Cho, "Effects of AI-Based Personalized Adaptive Learning System in Higher Education," *International Journal of Artificial Intelligence in Education*, vol. 31, no. 2, pp. 1-12, 2021.

References

[7]AI-assisted Knowledge Assessment Techniques for Adaptive Learning Environments

S. Minn, "AI-assisted Knowledge Assessment Techniques for Adaptive Learning Environments," *Computers & Education: Artificial Intelligence*, vol. 2, p. 100017, 2022. doi: 10.1016/j.caeai.2022.100050.

[8]F. Miao and W. Holmes, *Guidance for Generative AI in Education and Research*, UNESCO Report, 2023. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000386693. Accessed: May 16, 2023.

[9]L. Huang, J. Li, and Y. Wei, "Exploring Generative Artificial Intelligence Preparedness Among University Language Instructors: A Case Study," *IEEE Transactions on Learning Technologies*, vol. 17, no. 2, pp. 34-45, Apr. 2024. doi: 10.1109/TLT.2024.3392830

Thank
You