

RETAINIQ — AI-POWERED REVISION QUIZ ENGINE

Round-2 Prototype Documentation

Team: CognitiveX

Project Type: AI-Based Personalized Revision & Quiz Platform

Team Members

1. Sahaja Sandasani
2. Kusumitha Vemali
3. Tharalika Siriki
4. Sanjay Ravuri

Abstract:

RetainIQ is an AI-driven intelligent revision and quiz platform designed to enhance long-term learning and conceptual retention. Traditional quiz systems measure only scores, offering no personalized revision, weak-concept detection, or memory reinforcement mechanisms. Students forget most concepts without structured revision, and educators lack tools to track conceptual gaps over time.

RetainIQ solves these limitations using **AI-powered question generation**, **automatic weak concept identification**, and **spaced-repetition-based revision scheduling**. The platform dynamically adapts to each learner's performance and provides personalized revision quizzes, visual analytics, and mastery tracking. RetainIQ transforms assessments into a **continuous, retention-oriented learning system** benefiting students, educators, and corporate learners alike.

Problem Statement:

Educational institutions rely on quizzes for evaluation, engagement, and feedback, but current solutions fall short in supporting long-term learning. Major drawbacks include the absence of weak-concept detection, lack of adaptive revision, and no scientific strategy to reinforce memory retention. **Challenges Identified:**

1. **Manual and non-personalized revision:**
Students revise randomly without guidance, leading to rapid forgetting and poor long-term retention.

2. **Lack of AI-based quiz generation:**

Existing platforms cannot automatically generate MCQs from topics, textbooks, or PDF content.

3. **No weak concept analysis:**

Platforms do not analyze patterns like difficulty, response time, or repeated mistakes to identify conceptual gaps.

4. **No spaced-repetition scheduling:**

Current systems fail to scientifically plan revision (Day 1, Day 3, Day 7, etc.).

5. **No performance-driven adaptation:**

All learners receive the same revision regardless of their mastery level.

Core Problem

The absence of a unified platform that can **generate quizzes, detect weak concepts, schedule scientific revision, and personalize learning paths** forms the central issue RetainIQ solves.

Objectives:

Functional Objectives:

1. Develop an AI engine capable of generating structured MCQs from text, topics, or PDFs.
2. Provide personalized revision quizzes based on weak concepts identified from user performance.
3. Automate revision planning using spaced repetition to ensure long-term retention.
4. Deliver user dashboards to visualize progress, mastery scores, and upcoming revision cycles.

Technical Objectives:

1. Build a modular architecture enabling independent scaling of quiz, AI, and scheduling services.
2. Implement backend engines for weak concept detection and automated revision scheduling.
3. Integrate LLM-based pipelines for concept extraction, difficulty estimation, and validated MCQ generation.

4. Create secure APIs for quiz evaluation, revision management, and history tracking.

User Experience Objectives:

- Provide a clean and intuitive quiz interface with real-time feedback.
- Deliver a personalized revision planner showing upcoming tasks.
- Enable smooth cross-device interaction and visually rich dashboards.
- Allow learners to track mastery using charts and memory retention graphs.

Literature Review Summary:

1. AI-Based Question Generator:

Large Language Models (LLMs) generate high-quality MCQs through structured prompting and validation. Key processes include:

- Concept extraction
- Distractor generation
- Difficulty assessment
- Answer verification

2. NLP for Document Understanding:

Advanced NLP tasks help extract meaning from PDFs/text:

- Keyphrase extraction
- Summarization
- Named Entity Recognition (NER)

These ensure AI-generated quizzes remain contextually accurate and relevant.

3. Spaced Repetition Learning Frameworks:

Widely validated cognitive techniques show that structured revision intervals significantly improve long-term memory. RetainIQ uses intervals like Day 1, Day 3, Day 7, and Day 14.

4. Adaptive Learning Systems:

Performance-based adaptation enhances engagement and mastery by tailoring question difficulty to learners.

System Architecture and Overview:

RetainIQ adopts a modular, scalable architecture designed for maintainability and personalization at scale.

Core Architectural Components:

1. Frontend — React

- Interactive quiz interface
- Personalized dashboard with analytics and retention graphs
- Smart revision planner with upcoming schedules
- REST API integration

2. Backend (Node.js / FastAPI):

- Quiz logic and evaluation services
- Weak concept detection engine
- Automated spaced-repetition scheduler
- User profile, history, and performance management

3. AI Layer — LLM Powered:

- Concept extraction from text/PDFs
- MCQ generation with structured templates
- Adaptive revision quiz generator
- Difficulty scaling based on learner performance

4. Databases (MongoDB / Firebase):

- Quiz attempts and logs
- Weak concepts and revision schedules
- Mastery scores
- User progress tracking

Your backend Python script further manages database schema upgrades by adding timestamps and improving data consistency.

5. Deployment Infrastructure:

- Cloud deployment (AWS/Render/Netlify)
- Scalable backend services
- Cached LLM calls for optimized performance
- CI/CD integration

Detailed Module Design:

1. AI Quiz Generator:

- Cleans and segments input text/PDFs
- Extracts key topics
- Generates MCQs using LLM prompts
- Validates answers and distractors
- Assigns question difficulty

2. Weak Concept Detection Module:

Analyzes:

- Incorrect answers
- Time taken
- Difficulty level
- Repeated mistakes
- Topic mastery score

3. Revision Scheduling Module:

- Implements spaced repetition (Day 1 → Day 3 → Day 7 → Day 14)
- Adjusts schedule dynamically based on improvement or decline

4. Analytics Module:

- Accuracy tracking per topic
- Mastery-level visualization
- Retention curve modeling
- Weekly progress summaries

5. Frontend Module:

- Quiz interface with smooth animations
- Dashboard for analytics and revision planning
- Interactive charts for retention and mastery
- Fully responsive design

Implementation Details (Frontend, Backend, AI):

1. Frontend Implementation:

- React + Tailwind CSS for clean UI
- Recharts/Chart.js used for analytics
- State management for quiz flow and revision tasks
- Responsive and optimized layouts

2. Backend Implementation:

- REST APIs for quiz creation, evaluation, scheduling
- SQLite/MongoDB integration for data storage
- Python script (uploaded) manages DB schema updates
- Modular architecture for easy maintainability

3. AI Module Implementation:

- LLM integration for quiz and revision creation
- Prompt pipelines: extract → generate → validate
- Ability to process text/PDF inputs

API Specifications:

1. POST /api/v1/ai/generate

Generates MCQs from topic/text/PDF.

Response: Questions, options, correct answer, difficulty.

2. POST /api/v1/quiz/submit

Submits quiz responses and triggers weak concept detection.

3. GET /api/v1/revision/schedule

Returns upcoming spaced-repetition tasks.

4. GET /api/v1/analytics/progress

Displays mastery scores and retention curves.

Deployment and Scalability:

1. Deployment Process:

- Frontend deployed on Netlify
- Backend deployed on Render/AWS

- Database hosted on Firebase/MongoDB
- LLM calls optimized through caching

2. Scalability Strategies:

- Modular microservices
- AI caching for repeated topics
- Cloud auto-scaling
- Efficient data handling for thousands of learners

Testing, Evaluation and Metrics:

Testing Procedures:

1. **Unit Testing**
 - Validation of AI outputs and revision logic
2. **Integration Testing**
 - AI quiz generation → weak concept detection → revision scheduling
3. **Load Testing**
 - Stress testing dashboard and quiz API performance
4. **Usability Testing**
 - Interface refinement based on user feedback

Evaluation Metrics:

1. Feature completeness: 70–85%
2. Precision of weak-concept detection
3. Engagement improvement through revision scheduling
4. Usability score based on UI responsiveness
5. AI correctness and quality of MCQs

Conclusion:

RetainIQ presents a powerful solution for modern learning challenges by integrating AI-based quiz generation, weak concept detection, and spaced-repetition-driven revision planning into a single intelligent platform. Its adaptive approach ensures every learner receives personalized guidance, leading to improved memory retention and deeper conceptual understanding.

RetainIQ is scalable, practical, and transformative for students, educators, coaching centers, and corporate learning environments. It redefines digital learning by turning every quiz into a step toward **long-term mastery**.

