

Section 1: Problem Definition

Problem Statement

In higher education, academic dishonesty and cheating undermine learning outcomes and devalue qualifications. A major challenge is that standardized assignments and exam formats fail to account for diverse student learning abilities and preferences. This uniformity allows students to share answers, plagiarize, or misuse AI tools. Teachers also struggle to design varied exercises that reflect both their teaching expectations and individual student progress.

- **Problem chosen:** Lack of personalized assessments enables cheating and reduces fairness in higher education evaluation.
- **Potential consequences:** Unreliable assessment of knowledge, unfair grading, and reduced trust in higher education systems.
- **Target audience:**
 - *Students* → directly affected by unfair practices, peer pressure, and ineffective evaluation.
 - *Teachers* → face workload pressure in creating fair, adaptive assessments.

Problem Significance

Standardized tests cannot address personalized learning journeys. A system that adapts to student ability while aligning with teacher expectations will ensure fairer, cheating-resistant assessments.

Target Questions

- How can cheating be minimized by delivering personalized assignments to each student?
 - How can professors' expectations and intended learning outcomes be embedded into adaptive assessment design?
-

Section 2: The Solution

Proposed Solution

We propose a prototype application that generates personalized, adaptive assignments to address the growing challenge of academic dishonesty in higher education. By combining a question generator engine, student profiling module, and professor input interface within a user-friendly web platform, the system tailors tasks to individual learning abilities while aligning with course objectives.

Goals and Outcomes

- Develop a prototype application that generates personalized assignments for students.
- Ensure the assignment generation process aligns with both student progress and teachers' course objectives.
- Reduce cheating opportunities by designing engaging, individualized, and ability-matched tasks.

Expected Impact

- **Impact:** Better academic integrity, fairer student evaluations which further helps in reducing plagiarism and answer-sharing.
- **Checking success:**
 - Pilot with small student groups: fewer identical answers across submissions.
 - Professor feedback: reduced effort in assignment design.
 - Student feedback: perception of fairness and relevance.
- **Evaluation strategy:** Compare outcomes (cheating detection rate, professor workload, student feedback) before and after prototype use.

Requirements

1. Assignment Generator Engine – generates varied and adaptive exercises (*High priority; Tech Developer*).
2. Student Profiling Module – captures learning ability and preferences (*High priority; Data Analyst*).
3. Professor Input Interface – integrates learning objectives and question banks (*High priority; Education Lead*).
4. Usability Design – user-friendly web interface for seamless use (*Medium priority; UX Designer*).

Validation: Initial testing with mock student data, verified by professors for pedagogical alignment.

Section 3: The Methodology

Assigned Methods

- Requirement 1 (Assignment generator) → *LLM + template-based Assignments variation*.
- Requirement 2 (Student profiling) → *Questionnaire*
- Requirement 3 (Professor interface) → *Questionnaire + structured input forms /interface to add files*

Data Collection Methods

- Questionnaire: student learning abilities, teacher expectations
- Interviews: professor pain points (difficulties, challenges, or frustrations) in designing assignments).

Data Analysis Techniques

- Descriptive statistics (student engagement).
- Thematic analysis (feedback from professors/students).

Method Evaluation

- Success measured by: diversity of generated assignments, reduced identical answers, improved student engagement.

- Effectiveness checked through mock test runs and feedback cycles.
-

Section 4: Timeline

Plan

1. Day 1: Define requirements and collect professor/student inputs.
2. Day 2: Develop question generator prototype and prepare presentation.

Priorities

- First: Core functionality (Assignment generator + professor interface).
- Second: Student profiling & adaptive logic.
- Third: Usability design.

Task Responsibility

- Assignment Generator Engine: Qasim Sefaldeen
- Student Profiling Module: Shanti Pandey
- Professor Input Interface: Shanti Pandey, Yann Bernard, Qasim Sefaldeen
- Usability Design (Web Interface): Yann Bernard

Key Milestones

- Day 1: Problem definition and requirements finalized.
 - Day 2: Functional prototype of assignment generator completed.
-

Section 5: Evaluation Plan

Evaluation Criteria

- Reduction in similarity rates across student submissions.
- Positive feedback from professors (ease of use, fairness, reduced workload).
- Student satisfaction (perceived fairness, alignment with learning ability).
- Technical quality (error-free generation of unique assignments).

Risk Analysis

- Risk: Generated assignments may become repetitive.
 - *Mitigation:* Expand randomization templates and integrate dynamic content variation.
- Risk: Professors may be reluctant to adopt new tools.
 - *Mitigation:* Provide intuitive input interface, training sessions, and onboarding support.

Next Steps

- Enhance algorithms for greater assignment variation.
- Expand professor input options across disciplines and assignment formats.
- Integrate teacher evaluation checkpoints during/after assignment completion.
- Refine project goals based on pilot outcomes.