QuizVault: An Intelligent, Secure, and Rapid Quiz Platform for Post-Lecture Assessments

Mohammed Maaz Khan
Dept. of Computer Engineering
FCRIT, Vashi
Navi Mumbai, India
1024203@comps.fcrit.ac.in

Shadab Nasim Khan
Dept. of Computer Engineering
FCRIT, Vashi
Navi Mumbai, India
1024204@comps.fcrit.ac.in

Clarissa Noronha
Dept. of Computer Engineering
FCRIT, Vashi
Navi Mumbai, India
102424@comps.fcrit.ac.in

Mohammed Yahya Shaikh
Dept. of Computer Engineering
FCRIT, Vashi
Navi Mumbai, India
1024246@comps.fcrit.ac.in

Dr. Chhaya Pawar
Dept. of Computer Engineering
FCRIT, Vashi
Navi Mumbai, India
chhaya.pawar@fcrit.ac.in

***Abstract*—In todays world teachers need things that work well and are also safe and can be changed. QuizVault is a tool that helps teachers do this by letting them give tests like Continuous Internal Assessments right after they teach. This helps students remember what they learned because it is still fresh in their minds. It also helps students pay attention and do better in school. QuizVault lets teachers make tests in ways like using a computer to make them scanning papers or typing them in themselves. This means teachers can make tests that fit what they are teaching and how they teach. QuizVault is a tool for many different teaching needs. To make sure everything is fair QuizVault checks who the students are when they log in and watches what they do on their computers during tests. It also writes down when students start and finish their tests so everything is open and honest. QuizVault is easy to use and works with the systems that schools already have so it is a good way to do online learning whether it is a quick test or a very important one.**

***Index Terms*—Quiz Generation, Educational Technology AI, in Education Document Parsing, Tab Monitoring, Student Assessment**

I. Introduction

Teachers need to know what students have learned and where they need help. Giving students quizzes right after a class is really helpful because the material is still fresh in the students minds. These short quizzes can show the students and the teachers what they need to work on some more.

The thing is, a lot of quiz tools are not very good. Some are too easy to use. They do not have basic things to stop cheating. Others are too hard to set up. You need to get extra software.

QuizVault is trying to be easy to use and quick to set up. It is also trying to be fair. QuizVault focuses on quizzes that are about one topic and teachers can make these quizzes quickly or the system can generate them. QuizVault also has some checks to make sure students are not cheating like making sure they sign in and it keeps track of what they are doing on their computer. This way the results are more accurate. The students do not feel like they are being watched all the time. QuizVault and its quizzes are made to help students and teachers and QuizVault is a tool, for this.

A. Motivation

Teachers need to get feedback on time so they can make their lessons better. If they have to wait for days to get the results of a quiz it interrupts the process of learning. Reduces the chance to correct mistakes. Short quizzes, also known as classroom assessments or formative checks help students find and fix their mistakes while the material is still fresh in their minds. This helps them remember things better and learn effectively. For teachers getting results from these quizzes gives them useful information about what the whole class and individual students are struggling with. This allows them to adjust the pace of their lessons teach concepts again or give targeted practice in the next lesson. A tool that makes the process of creating and giving these quizzes easier. From making question banks to scoring and showing results in a simple way. Reduces the workload for teachers and makes the cycle of assessing and teaching faster. When this tool also lets teachers quickly change quiz questions see how hard each question is and easily export results to grade books it helps make assessing a part of teaching and reflecting on their work. Research shows that giving feedback and assessing students often leads to better results and more engagement especially when the feedback is specific and students have the chance to practice and correct their mistakes. Therefore making it fast and easy to create give and review quizzes can significantly improve how well teachers teach and how much students learn. However the design of such a tool must also consider fairness, accessibility. Not taking up too much instructional time.

B. Aim and Objectives

Aim: Build a web-based quiz platform that supports immediate quiz creation after lectures, verified student access, simple integrity checks, and clear reports for teachers.

Objectives:

- Use institute login for student verification.
- Monitor tab switching and log timestamps.
- Offer quick quiz creation via manual input, document upload, and AI-assisted suggestions.
- Provide basic analytics and feedback for teachers and students.

II. Study of the System

QuizVault is split into three parts: a frontend that students and teachers use, a backend that runs logic and stores data, and helper tools for AI and document parsing. The frontend uses common web technologies (HTML, CSS, JavaScript) to create a responsive experience. The backend uses Spring Boot to serve APIs and handle quizzes, while PostgreSQL stores data. Small helper tools (document scanner, tab monitor) automate useful tasks and keep the system light.

The document scanner helps teachers by extracting text from uploaded lecture notes and suggesting candidate questions. The AI module proposes question wording and options; teachers keep final control and edit suggestions before publishing. This balance speeds up creation but avoids fully trusting automated output.

A. Related +

Many existing platforms support quiz creation, but they differ markedly in security, scalability, and pedagogical focus. Simple, user-friendly tools such as Google Forms[https://docs.google.com]; however, they generally lack robust identity verification, proctoring, and fine-grained behavioral monitoring features, making them vulnerable for high-stakes testing. Open-source learning management systems like Moodle[https://docs.moodle.org/test/Quiz_module.] offer extensive assessment functionality and can integrate with institutional authentication, gradebook, and rostering systems, which supports secure workflows at scale, but their flexibility and feature richness often come with significant configuration, maintenance, and training overhead for administrators and instructors.Research on online assessment and learning technologies highlights several factors that shape both student performance and academic integrity. Attali and van der Kleij (2017) show that the order and randomization of test questions can significantly influence how students behave during exams. Randomizing items reduces opportunities for answer sharing and encourages independent thinking, though care must be taken to ensure fairness in difficulty distribution. Complementing this, Noorbehbahani et al. (2022) provide a broad review of cheating in online exams, emphasizing that while digital platforms increase accessibility, they also introduce new avenues for academic dishonesty. Their work underscores the importance of combining technical safeguards—such as proctoring tools, browser restrictions, and behavioral monitoring—with well-designed assessment strategies. In parallel, Diedenhofen and Musch (2017) introduce tools like PageFocus, which enable tracking of user attention during online activities. Such tools can help detect disengagement or suspicious behaviors, offering additional insight into test-taking patterns beyond final scores. Meanwhile, Khaldi et al. (2023) examine the role of gamification in higher education, finding that elements like rewards, progress tracking, and interactive design can boost motivation and engagement, particularly in low-stakes or formative contexts. Study-oriented services such as Quizlet[https://quizlet.com/] and ProProfs[https://www.proprofs.com/quiz-school/] concentrate on active recall, flashcards, and practice quizzes; these platforms are excellent for formative learning and low-stakes practice but are not designed to enforce timed conditions, prevent collusion, or capture evidence of candidate identity and behavior during an exam. Empirical research indicates that assessment design elements — notably question randomization (item banks and shuffling) and real-time or recorded monitoring (proctoring, browser lockdowns, video/audio capture, and behavioral analytics) — substantially reduce simple forms of cheating (e.g., answer sharing, copy-paste, and collusion) and increase perceived fairness among examinees. Combining pedagogically sound item banks with technical controls and transparent policies (e.g., clear honor codes and privacy safeguards) produces the best balance between academic integrity and user experience, though trade-offs remain between cost, privacy, accessibility, and administrative complexity. Existing systems can often be circumvented through one or more techniques, as they typically address only specific aspects of assessment security or engagement. In contrast, Quiz Vault adopts a more comprehensive approach by integrating multiple safeguards and features into a single platform, providing an all-in-one solution that strengthens both integrity and user experience.

III. Proposed System

This section describes QuizVault from the perspectives of teachers, students, and system administrators.

A. Teacher View

Teachers log in to the system using the schools authentication. They get to see a dashboard that lists all the Rooms, which are like class groups. For each Room teachers have a lot of options. They can:

- Make quizzes from scratch using an editor that lets them add multiple-choice questions and short answers.
- Upload notes from lectures, like PDF or DOCX files. Use the Document Scanner to find sentences that could be used as questions.
- Ask the AI helper to come up with questions and answers based on a topic or the notes they uploaded.
- Choose how the quiz works, like how time students have, if the questions are, in a random order if students can check their work and what happens if they switch tabs.
- Try out a quiz before giving it to the students to make sure it works okay.

Teachers can also view analytics after the quiz: per-question correctness, average time per question, distribution of scores, and a simple list of flagged attempts if tab-switch rules were violated.

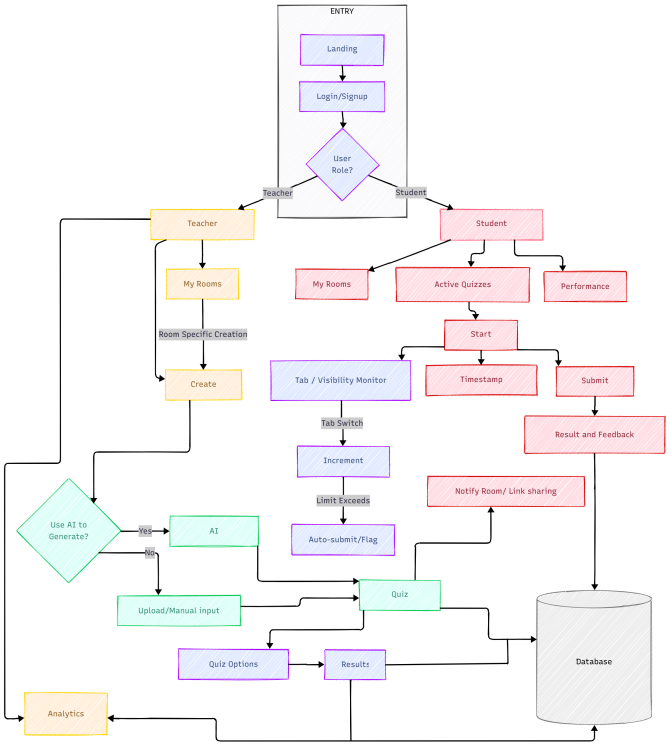

Fig. 1: System Architecture

B. Student View

Students access their assigned Room and open an active quiz. On starting, QuizVault logs the start time and begins a visibility logger. The student interface is kept uncluttered: a single column view with the question, options, navigation buttons, and remaining time. The tab monitor runs silently; when the student leaves the page, an event is recorded. If configured, students receive a gentle warning on first violations. Teachers choose whether to show answers immediately or after grading.

After submission, the system may show a score summary and brief feedback. If teacher review is required, students see a confirmation and later receive results.

C. Administrator View

Administrators manage Rooms, user roles, and institute-level settings such as SSO/OAuth configuration, retention policy for logs, and allowed external AI services. Admins can review system-wide flagged activity and audit logs for compliance.

IV. Design and Architecture

QuizVault uses a layered architecture for clarity and easier maintenance.

A. Frontend Layer

Built with React, the frontend has components for:

- Dashboard and Room lists.
- Quiz editor (question creation, import tools).
- Quiz runner (student interface with timer).
- Reports and analytics pages.

Small client-side scripts handle tab visibility using the Page Visibility API and send minimal events to the server.

B. Backend Layer

Spring Boot exposes REST endpoints for:

- User and room management.
- Quiz lifecycle management (create, publish, assign, start, submit).
- AI orchestration and document parsing jobs.
- Visibility event logging and scoring.

Security uses institute SSO with JWT tokens. Services are split into controllers, services, and repositories for testability.

C. Data Layer

PostgreSQL stores the main data. A simple schema keeps things readable and indexed for common queries. Supabase or Redis can be added for realtime notifications and caching.

D. AI and Document Tools

The Document Scanner uses OCR for images and simple parsing for PDFs and DOCX files. It applies heuristics (identify bolded headings, bullet lists, sentences with definitions) to find candidate question stems.

The AI Suggestion module calls a transformer-based API (teacher-provided keys) with prompt templates that request short multiple-choice questions and distractors. Returned candidates are sanitized and presented for teacher editing.

E. Simple Sequence Flow

- Teacher uploads notes and requests suggestions (backend triggers OCR and AI jobs).
- Teacher selects or edits suggestions and publishes the quiz to a Room.
- Students open the quiz; start event is logged and visibility monitor begins.
- Student submits (manual or auto on time out); submit time logged and grader runs (auto for MCQ).
- Results and reports are generated; teachers review flagged attempts.

V. Implementation Details and Examples

This section uses plain examples to show how pieces work together.

A. Creating a Question from Notes

A teacher uploads a short PDF lecture on “Sorting Algorithms.” The Document Scanner detects the sentence: “Bubble sort repeatedly compares adjacent elements and swaps them.” The AI-assisted authoring module generates a candidate multiple-choice question (MCQ) with suggested distractors and metadata (estimated difficulty, topic tags). The suggested item is presented in the instructor review queue; the teacher may edit, accept, or reject it. Example generated item:

1) Which sorting algorithm repeatedly compares adjacent elements and swaps them?

- A: Quick sort
- B: Bubble sort (correct)
- C: Merge sort
- D: Heap sort

The teacher accepts the question and moves on. This hybrid workflow saves typing while preserving human oversight for pedagogical correctness and alignment with learning objectives.

B. Tab-Switch Policy

A teacher configures a quiz with the following parameters:

- **Allowed tab switches:** 2
- **On 3rd switch:** auto-submit

During delivery the client-side proctoring component uses the Page Visibility API and focus/blur events to log each focus-loss with a timestamp and increment a tab-switch counter. If a student switches tabs three times, QuizVault:

1) auto-submits the quiz,
2) records the focus-change events (timestamps, event type, session id) and the reason for submission,
3) flags the attempt in the instructor dashboard for review.

These logs allow instructors to reconstruct sessions for adjudication and to tune conservative or permissive behaviors (e.g., soft warning, score penalty, or auto-submit).

C. Scoring and Simple Analytics

After automatic grading of objective items (MCQs), the analytics pipeline computes standard aggregate metrics:

- Mean and median score per quiz,
- Question-wise percent-correct (item difficulty),
- Average time taken per question,
- Tab-switch frequency distributions and correlation with scores (for integrity analysis).

Instructors may use these metrics to identify weak topics (low percent-correct) or poorly performing distractors, then revise content or devote class time to remediation. Aggregated, anonymized performance signals can also be fed back to the AI question-generation component to improve future item quality.

D. Authoring and Review UI Examples

The authoring interface presents questions as editable cards with inline controls (edit, delete, reorder) followed by an "Add New Question" form for manual entry. Generated items appear in the same card layout so teachers have a consistent review workflow. Each card displays metadata (marks, estimated difficulty, topic tags) to help rapid triage. Teachers can also perform roster actions from the authoring view, such as removing students from the eligible list to conduct re-assessments or to revoke access for specific sessions.

E. Student Experience

The Student Experience is very important. Students have a dashboard that shows them what is coming up. This includes:

- quizzes with the title start time how long it will take and the room or subject.
- Personal statistics like the score recent attempts and how they did on each question
- The history of their attempts. Any attempts that were flagged.

This helps students plan and get ready for what's coming up. It also helps them see what they need to work on. The Classroom and Room Model is set up so that each subject in a class has its room. This means that

F. Classroom and Room Model

Quizzes are for a specific class and subject:

- quizzes are scoped to a specific class and subject,
- instructors manage quizzes and rosters per room,
- analytics and logs are isolated by room for easier reporting and access control.

G. Event Logging and Audit

All assessment-related events are persisted to the backend, including:

- Student responses and submission timestamps,
- Focus/visibility change events with timestamps,
- Instructor actions (generate, edit, publish, delete, roster changes) with actor id and timestamps,
- Auto-submit reasons and flagged-attempt markers.

These logs support (1) per-student session reconstruction for auditing suspected misconduct, (2) item-level psychometric analysis, and (3) instructor-facing reports for grading and remediation.

H. Security and Privacy Considerations

Server endpoints require authenticated requests and enforce role-based access control. Event data is transmitted over TLS and client-side storage is minimized; sensitive fields are excluded from persistent client caches. Access to raw logs and roster management is limited to authorized instructor and administrative roles in accordance with institutional policies.

VI. Proposed Evaluation Plan

To evaluate effectiveness, conduct a controlled pilot in a single course over one academic module.

A. Pilot Design

Run QuizVault for four weeks of class assessments. Keep content and contact hours the same, as before. Use a study design to compare results before and after using QuizVault.

- Treatment group: current term using QuizVault for weekly CIAs (4 weeks).
- Comparison group: previous term (or parallel section) that used delayed/quasi-summative quizzes.
- Optional: randomize at section level if feasible to reduce confounds.

B. Data Collection

Collect the following data streams during the pilot:

- Assessment results: per-student quiz scores, timestamps, item-level responses.
- Integrity logs: focus/visibility events, flagged attempts, auto-submit incidents.
- Authoring effort: teacher time spent creating/editing quizzes (self-reported or instrumented).
- Retention measure: short follow-up quiz administered 1–2 weeks after topics are taught (optional).
- Surveys: brief teacher and student questionnaires on usability, perceived fairness, and satisfaction.
- System telemetry: AI-generation acceptance rate, edit frequency, and time-to-publish for generated quizzes.

C. Outcome Metrics

Primary and secondary metrics:

- **Primary:** Change in class mean score between treatment and comparison; short-term retention measured by follow-up quiz.
- **Secondary:** Median score, score distribution (variance), percent-correct per item, and time-on-question.
- **Integrity metrics:** Number and rate of flagged attempts per student, proportion of auto-submits, and correlation between tab-switch frequency and scores.
- **Authoring burden:** Average teacher time per quiz (manual vs. AI-assisted), proportion of AI-generated items accepted without edits.
- **User satisfaction:** Likert-scale scores from teacher and student surveys and qualitative open comments.

D. Analysis Plan

Perform the following analyses:

- Compare means using t-tests or nonparametric alternatives (Mann–Whitney) depending on distributional assumptions.
- Analyze retention via paired comparisons for students who take both immediate and follow-up quizzes.
- Model integrity signals (tab-switch count) against scores using regression to estimate association and potential thresholds for action.
- Report effect sizes (Cohen's d) and confidence intervals for primary outcomes.
- Summarize qualitative feedback using thematic coding and report representative quotes.

E. Ethics and Practical Considerations

Obtain institutional approval if student data or experiments affect grading. Ensure informed consent for participation in surveys and optional retention quizzes. Anonymize analysis datasets and limit access to raw logs. Provide instructors with guidelines for using flagged-attempt data fairly (review before penalizing).

F. Success Criteria

Define pilot success with measurable targets, for example:

- Reduced average teacher authoring time per quiz by 30% (or other baseline-relative target).
- No statistically significant drop in mean student performance compared to comparison group, and improved short-term retention.
- Acceptable false-positive rate for flagged attempts (instructor-reviewed) and positive teacher feedback on usefulness of logs.

G. Next Steps

If pilot results are positive, scale to additional courses, iterate on AI-generation quality using aggregated anonymized performance signals, and refine proctoring thresholds based on observed precision/recall trade-offs.

VII. Limitations and Ethical Considerations

QuizVault targets light-weight integrity checks and streamlined workflow. It is not a replacement for high-stakes proctoring systems. The tab-switch monitor is easy to implement but can be imperfect: students may lose focus for reasons other than cheating. Any stronger monitoring (screen recording, camera monitoring, facial recognition) raises privacy and consent issues; such features are left out or optional only when policies and consent are in place.

Data retention must follow institutional rules. Minimal personal data should be stored and logs should be retained only as long as needed for academic or legal reasons.

VIII. Future Work

Planned improvements include:

- Adaptive quizzes that change difficulty based on responses.
- Better AI filtering to avoid poor-quality auto-generated questions.
- Mobile app with offline-first support so students can take short quizzes without stable internet, syncing later.
- Deeper LMS integration for grade syncing and class roster updates.
- Visual dashboards for instructors with cohort trends over time.

IX. Conclusion

QuizVault aims to make short, post-lecture quizzes easy and useful. By combining quick question creation, light integrity checks, and instant feedback, the platform supports a learning cycle where teachers can adjust teaching quickly and students get timely guidance. The system is intentionally simple to adopt, balancing usefulness with respect for student privacy and teacher control.

ACKNOWLEDGEMENT

We thank our mentor Dr. Chhaya Pawar, project coordinator Mrs. Chetana Badgujar, Dr. M. Kiruthika (HOD), Principal Dr. S. M. Khot, and the faculty of Fr. C. Rodrigues Institute of Technology, Vashi for guidance and support. We also thank teammates for their contribution.

REFERENCES

[1] Y. Attali and F. van der Kleij, "Effects of item order and item randomization on test-taker behavior and performance," *Journal of Educational Measurement*, vol. 54, no. 2, pp. 123–140, 2017.

[2] M. Khaldi, A. Benabdelouahed, and S. El Khatib, "Gamification in higher education: A systematic review," *International Journal of Educational Technology*, vol. 10, no. 1, pp. 45–60, 2023.

[3] Google, "Google Forms Documentation," https://docs.google.com/forms.

[4] B. Diedenhofen and J. Musch, "PageFocus: A tool for measuring attention and engagement in online surveys," *Behavior Research Methods*, vol. 49, no. 3, pp. 1–10, 2017.

[5] F. Noorbehbahani *et al.*, "Cheating in online exams: A state-of-the-art review," *Computers in Applied Engineering Education*, vol. 30, no. 1, pp. 1–12, 2022.

[6] Google Gemini, "AI Studio," https://www.aistudio.google.com/.

[7] Moodle, "Quiz module," https://docs.moodle.org/test/Quiz_module.

[8] Quizlet, "Study Tools and Learning Resources," https://quizlet.com/.

[9] ProProfs, "Quiz Maker," https://www.proprofs.com/quiz-school/.

[10] Stack Overflow, "JavaScript Page Visibility API examples," https://stackoverflow.com/questions/1060008.