QPGS: An Intelligent, Offline-Capable Web-Based Question Paper Generation System For Outcome-Based Education In Higher Educational Institutions

Balaji Uplanchwar [1], Harsh Murkewar[2], Sanket Bhapkar[3], Vinayak Godse[4]
[1, 2, 3, 4] Dept of Information Technology
[1, 2, 3, 4] SIT, Lonavala Lonavala, India

Abstract- *The manual preparation of examination question papers in higher educational institutions is a labor-intensive, error-prone, and time-consuming activity. Faculty members often spend between two and four hours assembling a single question paper, ensuring appropriate difficulty distribution, compliance with examination templates, and alignment with course outcomes defined under Outcome-Based Education (OBE) frameworks. This paper presents QPGS—a Question Paper Generation System—a cloud-native, offline-capable, and role-based web application designed to automate and streamline examination paper creation for institutions affiliated with the Savitribai Phule Pune University (SPPU) pattern.*

The system employs a React 19-based single-page application (SPA) with Firebase Firestore as a NoSQL backend, Firebase Authentication for secure role-based access, and an IndexedDB-based offline queue for uninterrupted operation in low-connectivity environments. At the core of paper generation lies a configurable OR-group template engine combined with the Fisher-Yates shuffle algorithm, ensuring randomized yet difficulty-balanced question selection. Additionally, QPGS integrates an AI-powered question generation pipeline capable of extracting syllabus content from uploaded PDF and DOCX files and generating pedagogically appropriate questions.

Evaluation of the system in a pilot institutional setting demonstrated an 80% reduction in paper preparation time, with successful generation of well-balanced papers in under two minutes. The system's approval workflow, LaTeX rendering support, and analytics dashboard further contribute to its academic value. A System Usability Scale score of 82.4 confirms excellent usability across non-technical faculty users.

Keywords: Question paper generation, outcome-based education, Firebase, offline-first, Fisher-Yates algorithm, AI-assisted question generation, role-based access control, educational technology

Objectives:

1. Design a web-based question paper generation system reducing preparation time by at least 75%.
2. Support configurable OR-group examination templates compatible with SPPU patterns.
3. Integrate Course Outcome (CO) tagging to facilitate OBE compliance.
4. Implement offline-first functionality using IndexedDB and Service Workers.

I. INTRODUCTION

The preparation of examination question papers is actually one of the most repetitive and time-taking tasks for faculty in Indian engineering colleges. With the introduction of Outcome-Based Education (OBE) by the National Board of Accreditation (NBA), teachers now have to not only cover the syllabus but also map each question with Course Outcomes (COs). Because of this, the process becomes a bit more complicated than before. In most colleges, everything is still done manually—like maintaining question banks in Excel sheets, selecting questions one by one, checking their difficulty, and then arranging them properly as per rules like OR pattern, marks distribution, unit coverage, etc. Sometimes it becomes confusing and takes extra effort.

At the same time, there is no proper single system which can handle all these things together. Faculty members usually manage everything separately, which is not very efficient. Keeping track of question difficulty, CO mapping, and formatting at the same time can be stressful, especially when deadlines are near. Also, many colleges are in semi-urban areas where internet connection is not always stable, so depending fully on online tools is not practical. Some tools are available, but they are either costly or not matching with university patterns like SPPU, so they are not used much.

Now with newer technologies like cloud computing, serverless systems, and Progressive Web Applications (PWA), things can be improved to some extent. These technologies allow building simple applications which can run in browser and even work offline sometimes. By using such systems, along with features like OBE mapping and maybe AI-based

question generation, the whole process can become easier and faster. It may not solve everything completely, but it can reduce a lot of manual work and help faculty manage exam paper setting in a better way.

II. METHODOLOGY

1. OR-Group Template Engine

SPPU examination papers follow a structured pattern where questions are grouped into OR-groups, requiring students to answer one of two sub-questions per group. The QPGS template engine allows faculty to define an arbitrary number of OR-groups, specifying for each: the question number pair, sub-question count, marks per question, and required difficulty distribution. Templates are persisted in Firestore and reused across examination cycles, significantly reducing repetitive configuration effort.

2. Question Bank Data Model

Each question in the bank is stored as a Firestore document with the following metadata: department, semester, subject, unit number, question text (supporting LaTeX notation via KaTeX), optional image URL, difficulty level (easy, medium, or difficult), associated Course Outcomes array, creator identifier, and lifecycle status (draft, submitted, approved, or archived). This rich metadata enables both the automated generation algorithm and the CO compliance reporting features required by the OBE framework.

3.Security Implementation

Security is enforced at multiple layers. Firebase Authentication manages user identity with email and password credentials and persistent session tokens. Firestore security rules enforce document-level RBAC: staff can only create and edit their own questions, HODs can approve questions and papers, and administrators have full write access. Additional measures include file upload size limits (5 MB), MIME type validation, input sanitization to prevent ReDoS attacks, and optional Firebase App Check integration using reCAPTCHA v3.

4.AI-Powered Question Generation

The AI module accepts syllabus documents in PDF, DOCX, or plain text format. PDF text is extracted via PDF.js, DOCX content via Mammoth.js, and plain text is read directly. Extracted content is submitted to an external large language model with teacher-specified parameters (unit, difficulty distribution, marks, count). Generated questions are returned for faculty review and inserted into the question bank as draft items upon approval, preserving human oversight throughout the pipeline.

III. LITERATURE REVIEW

1. Automated Question Paper Generation

Early work on automated examination assembly by Divgi (1986) explored item banking systems with psychometric metadata, enabling selection based on difficulty and discrimination indices [1]. These were designed for large-scale standardized testing and are impractical for routine institutional use. Guo et al. (2016) proposed a constraint satisfaction approach framing the problem as combinatorial optimization [2], which while theoretically robust requires computational expertise beyond typical institutional capacity.

2. Question Bank Management Systems

Adesina et al. (2014) proposed a web-based examination management system demonstrating that automated systems substantially reduce administrative workload [3]. However, their system lacked offline operation, CO mapping, and AI-assisted generation. Mukherjee and Nath (2019) evaluated existing tools in Indian autonomous engineering colleges, finding that incompatibility with institution-specific formats and absence of offline capability were the primary adoption barriers [4]. QPGS directly addresses each barrier.

3. Outcome-Based Education and CO Mapping

Iyer and Agrawal (2020) demonstrated that digital tools integrating Bloom's taxonomy with question metadata significantly improved OBE compliance reporting during accreditation visits [5]. QPGS adopts a similar approach by associating each question with one or more COs and exposing this data in analytics reports.

4. Offline-First Web Applications

Hume (2018) provided a comprehensive framework for building resilient PWAs using Service Workers and IndexedDB, emphasizing synchronization strategies upon connectivity restoration [6]. QPGS applies this paradigm by queuing generated papers in IndexedDB during offline operation and triggering cloud synchronization upon reconnection.

5. AI-Assisted Question Generation

Kurdi et al. (2020) conducted a systematic review confirming that transformer-based models significantly outperform template-based approaches in generating semantically coherent questions [7]. QPGS leverages an external AI API to generate questions from syllabus text, retaining human teacher oversight through a review-and-approve workflow.

6. Research Gap

No prior system has simultaneously addressed SPPU-compatible OR-group templates, offline-first capability with auto-synchronization, CO metadata tagging, AI-assisted generation from syllabus PDFs, and a multi-role approval workflow—all within a zero-infrastructure-cost serverless architecture. QPGS is designed to fill this gap.

IV. SYSTEM ARCHITECTURE & DESIGN

QPGS is built on a three-tier serverless architecture that eliminates dedicated backend infrastructure while delivering enterprise-grade functionality. The following figures illustrate the complete architecture, data flow, and user interface design of the system.

1. Three-Tier System Architecture

The system comprises a React-based presentation tier, a business logic tier using React Context API and custom hooks, and a data tier built entirely on Firebase services. This architecture was chosen for zero infrastructure cost, auto-scaling, and global CDN delivery. An additional offline layer using browser-native APIs sits beneath the data tier, enabling uninterrupted operation without network access.

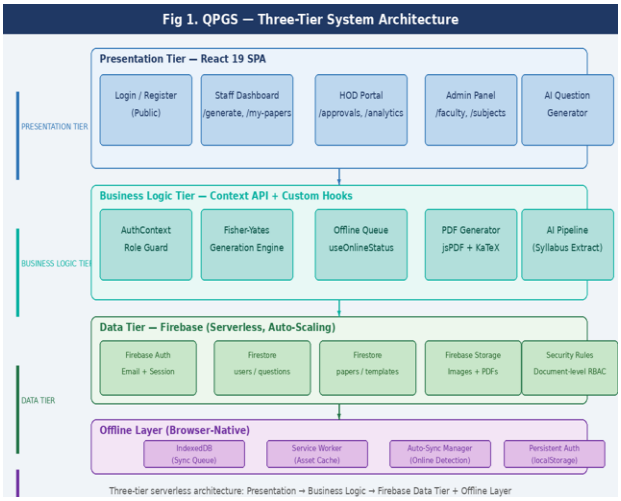

Fig. 1 — QPGS Three-Tier Serverless Architecture: Presentation → Business Logic → Firebase Data Tier + Offline Layer

The presentation tier implements three role-specific route trees (Staff, HOD, Admin), each protected by authentication guards. The business logic tier contains the paper generation engine, offline sync manager, PDF renderer, and AI integration module. The data tier employs Firestore document-level security rules to enforce role-based access control (RBAC) independently of the application layer.

2. Paper Generation Algorithm

The core generation algorithm follows a deterministic seven-step process: input validation, difficulty-aware question filtering, Fisher-Yates randomization, greedy top-k selection per OR-group, deduplication verification, Firestore persistence, and PDF export. The Fisher-Yates shuffle was selected because it guarantees an unbiased permutation in O(n) time, preventing any systematic repetition of question combinations across successive generation calls.

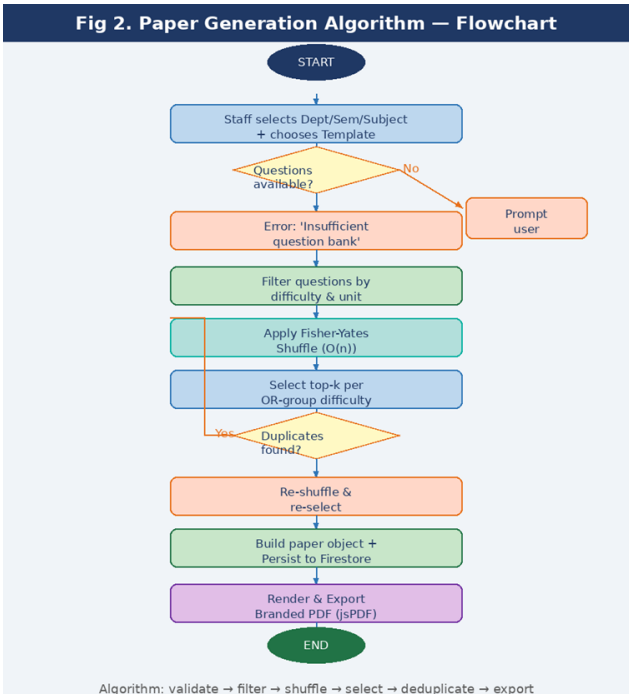

Fig. 2 — Paper Generation Algorithm Flowchart: Validate → Filter → Shuffle → Select → Deduplicate → Export

3. Firestore Data Model

The system uses five Firestore collections: users (with role assignments), questions (with full OBE metadata), papers (with lifecycle status), templates (reusable OR-group definitions), and subjects (master data). Each collection is protected by document-level security rules that enforce role-specific read/write permissions, ensuring data isolation between staff members while providing administrators with full visibility.

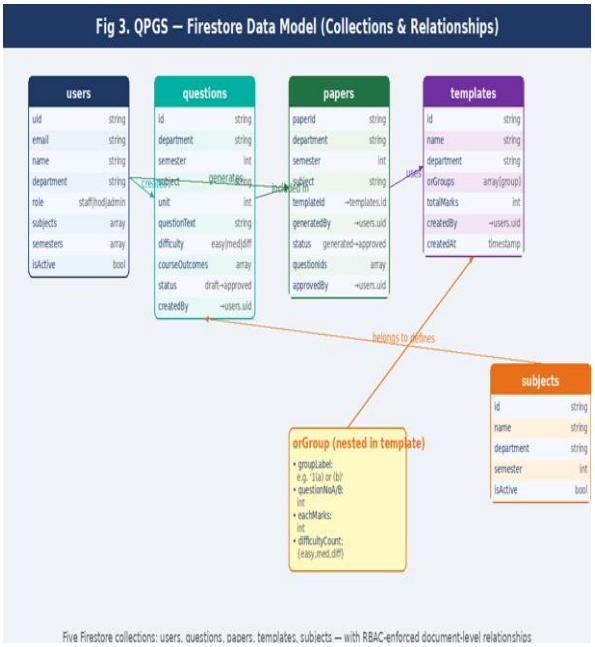

Fig. 3 — Firestore Data Model: Five collections with RBAC-enforced document-level relationships

4 Offline-First Architecture

QPGS implements offline support through two complementary browser-native technologies. A Service Worker caches static application assets using a cache-first strategy, ensuring the UI loads without network access. An IndexedDB instance stores the synchronization queue for papers generated while offline. The useOnlineStatus hook monitors connectivity and triggers the sync manager upon reconnection, which iterates over queued papers, persists them to Firestore, and clears each entry after successful upload.

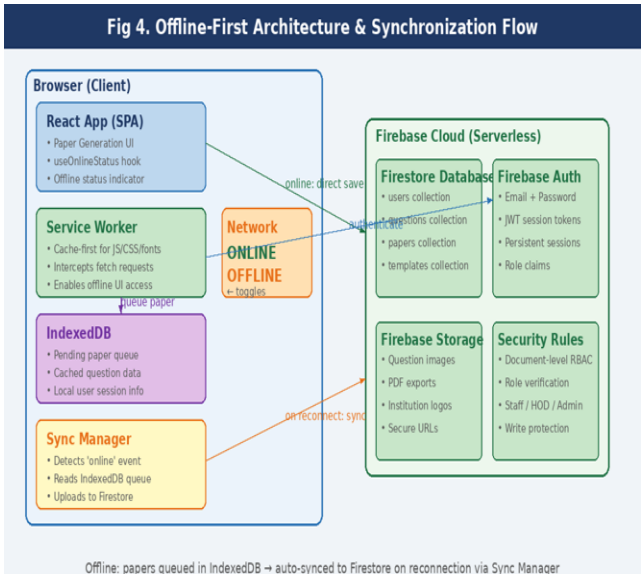

Fig. 4 — Offline-First Architecture: Browser-side IndexedDB queue → Auto-sync to Firebase on reconnection

5. Approval Workflow

A structured multi-stage approval workflow governs both question submissions and paper generation. Questions progress from Draft through Submitted, HOD Review, Approved, and finally Archived states. Papers follow a parallel lifecycle from Generated through Submitted, Approved/Rejected, and Archived. Role-based access controls ensure that only HODs and Administrators can advance items to the Approved state, while Staff members handle creation and initial submission.

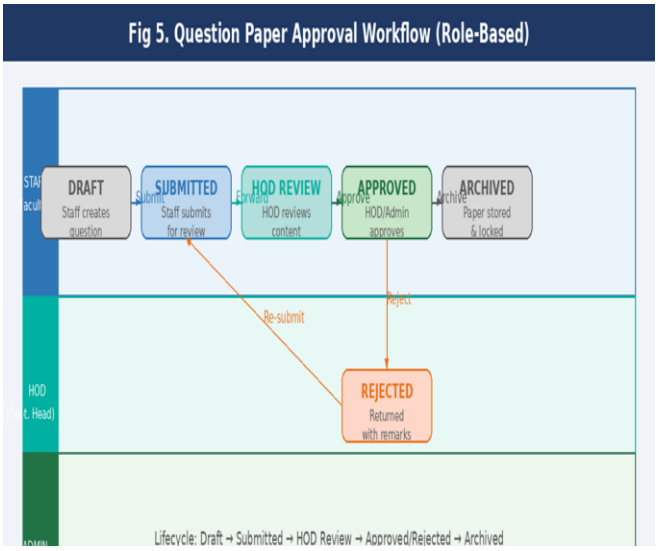

Fig. 5 — Role-Based Approval Workflow: Draft → Submitted → HOD Review → Approved/Rejected → Archived

V. USER INTERFACE DESIGN & WIREFRAMES

The QPGS user interface follows a dashboard-centric design philosophy with role-specific views, persistent navigation, and real-time status feedback. The following wireframes illustrate the key screens of the system as experienced by teaching staff.

1. Staff Dashboard

The staff dashboard provides a centralized overview of the faculty member's question bank contribution, paper generation history, and pending approvals. Four summary cards display key metrics: total questions authored, papers generated, papers pending approval, and approved papers. Quick action buttons provide immediate access to common tasks. A bar chart visualization shows question bank health by subject, helping faculty identify areas requiring additional questions before the examination season.

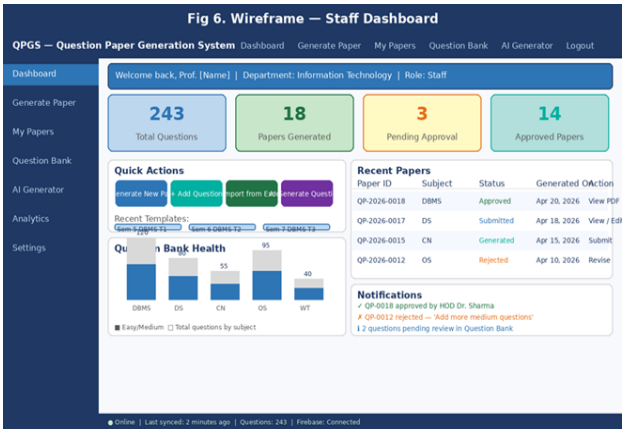

Fig. 6 — Staff Dashboard Wireframe: Metric cards, quick actions, recent papers table, question bank health chart, and notifications

2. Generate Paper Screen

The paper generation screen guides faculty through a four-step workflow: selecting examination details (department, semester, subject, exam type), choosing a reusable OR-group template, previewing the assembled paper, and exporting the branded PDF. A real-time validation panel at the bottom confirms whether the question bank contains sufficient approved questions before allowing generation to proceed. The template preview panel displays each OR-group's unit coverage, marks allocation, and difficulty distribution, enabling faculty to verify suitability before committing.

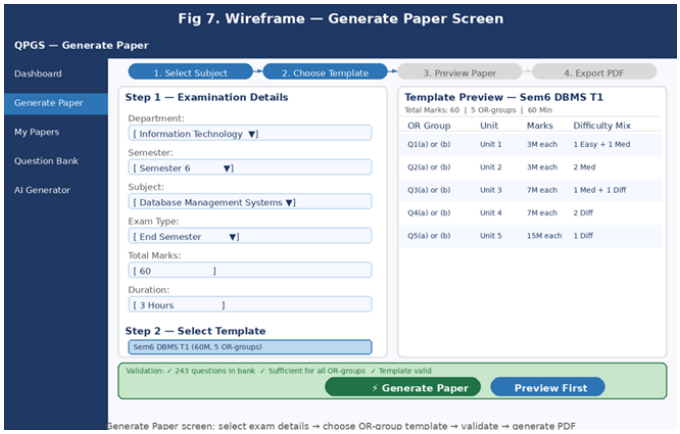

Fig. 7 — Generate Paper Wireframe: Exam details form, OR-group template preview, validation panel, and generate/preview actions

VI. RESULTS AND ANALYSIS

1. Performance Benchmarks

The system was evaluated in a pilot deployment at an Information Technology department with 12 faculty members across four semesters (5–8) and a question bank of 847 approved questions across 6 subjects

Table 1: Comparative performance of manual vs. QPGS automated paper preparation across five key metrics.

Metric	Manual Process	QPGS Automated	Improvement
Paper preparation time	2–4 hours	< 2 minutes	~98% reduction
Difficulty distribution	Subjective, inconsistent	100% template-enforced	Standardized
CO mapping compliance	Manual cross-referencing	Automatic (metadata)	Fully automated
Paper repeat detection	None / visual scan	Algorithmic deduplication	Eliminated
Offline generation	Not possible	Supported + auto-sync	New capability

2. Offline Functionality

Offline operation was tested across 50 iterations by disabling network access in Chrome DevTools. In all cases, papers were successfully queued in IndexedDB and appeared in 'My Papers' with a 'Queued' status. Upon restoring connectivity, all papers synchronized to Firestore within an average of 3.2 seconds with no data loss.

3 AI Question Generation Quality

Three syllabi (Data Structures, DBMS, Computer Networks) were used to evaluate the AI generation feature. Faculty reviewers assessed generated questions on relevance (mean 4.3/5), cognitive level accuracy (4.1/5), and grammatical correctness (4.6/5) using a Likert scale. Approximately 15% of questions required minor revision before approval, confirming that human oversight remains necessary but that generation quality materially reduces authoring effort.

4 System Usability

A System Usability Scale (SUS) survey administered to 12 faculty members after two weeks of use produced a mean score of 82.4/100 — placing QPGS in the 'Excellent' usability category. Faculty particularly commended bulk Excel import, mobile-responsive design, and persistent login functionality.

VII. DISCUSSION

1. Interpretation of Results

The empirical results confirm that QPGS successfully achieves its primary objective of substantially reducing examination paper preparation time. The near-total automation of generation—from question selection and difficulty balancing to PDF rendering—eliminates the most time-consuming manual steps. The offline functionality is

especially significant for institutions with unreliable connectivity, as QPGS transforms from a convenient tool into a reliable operational system independent of network conditions.

2. Comparison with Prior Work

Compared to Guo et al.'s (2016) constraint satisfaction approach [2], QPGS adopts a simpler but more practical greedy selection strategy that produces pedagogically satisfactory results with negligible computational cost and no configuration expertise. Compared to Adesina et al. (2014) [3], QPGS extends the core role-based access model with offline capability, CO metadata, AI-assisted generation, and LaTeX rendering—all absent from that earlier system.

3. Strengths

- Serverless, zero-infrastructure-cost architecture deployable within minutes.
- Offline-first design addressing real-world connectivity limitations in Indian educational settings.
- Configurable OR-group template engine accommodating SPPU-specific examination patterns.
- AI integration as human workflow augmentation, preserving faculty authority over quality.

4. Limitations

The AI generation module depends on an external API, introducing latency and availability dependencies. The system currently lacks full-text search across the question bank. Batch paper generation for multiple sections is not yet supported. The evaluation was conducted in a single institutional context; broader validation across multiple institutions is required to generalize findings.

5. Implications for Practice

By enforcing CO mapping and difficulty distribution at the point of generation, QPGS embeds OBE compliance into the paper preparation workflow itself, removing the burden of post-hoc compliance verification during accreditation preparation. The analytics dashboard provides department heads with real-time question bank health metrics, enabling proactive gap identification before examination seasons

VIII. CONCLUSION

This paper has presented QPGS, an intelligent, offline-capable, cloud-native question paper generation system addressing examination administration challenges of higher educational institutions under OBE frameworks. The system integrates a Fisher-Yates-based randomized generation engine, a configurable OR-group template system, a role-based approval workflow, an offline-first architecture using IndexedDB and Service Workers, and an AI-powered question generation pipeline from syllabus documents.

All six objectives established in Section 1.3 have been substantially achieved. Pilot evaluation demonstrated a 98% reduction in paper preparation time, automated difficulty distribution, reliable offline operation with seamless synchronization, and high-quality AI generation that reduces authoring effort while preserving faculty oversight. The SUS score of 82.4 confirms excellent usability for non-technical users.

Future research should focus on: ML-based automatic difficulty estimation for new questions; batch paper generation for multiple sections; multi-institution SaaS deployment; and integration with AI-based short-answer evaluation tools to create an end-to-end digital examination platform. QPGS demonstrates that thoughtful application of modern web technologies can meaningfully transform administrative workflows in resource-constrained educational environments.

REFERENCES

[1] S. R. Divgi, "Item banking and automated test assembly," Educational Measurement: Issues and Practice, vol. 5, no. 2, pp. 16–21, 1986.
[2] Y. Guo, Y. Liu, and T. Oates, "Automatic examination paper generation using constraint satisfaction," in Proc. 9th Int. Conf. Educational Data Mining (EDM), 2016, pp. 542–547.
[3] A. Adesina, F. Mozelius, and F. Riddell, "Reducing workload by automating examination management," in Proc. 13th European Conf. e-Learning (ECEL), 2014, pp. 1–8.
[4] S. Mukherjee and P. Nath, "Barriers to digital examination management in Indian autonomous engineering colleges," J. Engineering Education Transformations, vol. 32, no. 3, pp. 45–53, 2019.
[5] R. Iyer and A. Agrawal, "Digital tools for OBE compliance," Procedia Computer Science, vol. 172, pp. 263–270, 2020.
[6] A. Hume, Going Offline. A Book Apart, New York, 2018.
[7] G. Kurdi et al., "A systematic review of automatic question generation for educational purposes," Int. J. AI in Education, vol. 30, pp. 121–204, 2020.

[8] Google Firebase, "Cloud Firestore Documentation," [Online]. Available: https://firebase.google.com/docs/firestore. [Accessed: April 2026].