DS Journal of Language, Linguistics and Literature
Volume 2 Issue 4, 1-8, Oct - Dec 2024
ISSN: 2583-8490/ https://doi.org/10.59232/LLL-V2I4P101

Review Article

EduRAG: Transforming Education with AI-Powered Personalized Study Assistance

R.U. Rathish[1], G. Venkatraman[2], T. Dhanajeyam[3]

[1,2,3]*Artificial Intelligence and Data Science, Sri Shakthi Institute of Engineering and Technology, Tamilnadu, India.*

[3]dhanajeyam56@gmail.com

Received: 07 October 2024; Revised: 12 November 2024; Accepted: 30 November 2024; Published: 21 December 2024

Abstract - Personalized learning has emerged as a crucial paradigm in contemporary education, yet current platforms frequently fail to provide suggestions for information relevant to each student's requirements. In order to overcome this constraint, this work presents EduRAG, a domain-specific, AI-driven learning platform based on Retrieval-Augmented Generation (RAG). Students may submit their own study materials, such as lecture notes and textbooks, to EduRAG, which uses the provided materials alone to provide contextually appropriate answers and insights. The platform combines scalable serverless architecture with a sophisticated NLP algorithm to provide correct and timely replies. Three essential elements form the foundation of EduRAG's system architecture: (1) processing documents utilizing Optical Character Recognition (OCR) and FAISS for embedding-based indexing; (2) a RAG pipeline that combines optimized language generation models with a high-performance retriever; and (3) Flexible study plans that let users customize their learning and give priority to particular subjects. With a concentration on STEM fields (mathematics, physics, and chemistry), the system's initial implementation provides domain-specific accuracy and insights. An inventive blueprint-based query ranking system, thorough assessments of answer quality, and performance benchmarking against cutting-edge learning platforms are some of this work's main achievements. The findings show that while managing several user requests simultaneously, EduRAG delivers improved engagement metrics, lower latency, and greater scalability. By lowering AI bias and implementing strong data protection safeguards, the platform also prioritizes ethical issues. This article highlights EduRAG's potential as a game-changing tool in customized education by discussing its roadmap, which includes future integrations of multilingual support, gamification, and increased subject coverage.

Keywords - Natural Language Processing, Retrieval-Augmented Generation, Personalized education, Artificial Intelligence-driven learning, Study blueprints, Scalable architecture, Domain-specific platforms, Data privacy, STEM education, Ethical Artificial Intelligence.

1. Introduction

Students' access to and use of educational resources has changed dramatically in recent years thanks to individualized learning platforms. These platforms meet various academic demands by offering resources for interactive learning and the arrangement of study materials. Nevertheless, a lot of current solutions fall short of addressing important user issues.

First, most platforms provide general resources not tailored to specific study schedules, which lessens their usefulness and relevance. Second, their query features are frequently restricted, making it challenging for students to find certain explanations or answers in their course materials. Lastly, concerns about data privacy and transparency weaken consumer confidence in recommendation algorithms.

In order to get over these restrictions, this article presents EduRAG, an AI-powered, domain-specific learning platform that uses Retrieval-Augmented Generation (RAG) technology. Students may submit customized study resources, including lecture notes and textbooks, to EduRAG, providing contextually relevant answers based on the provided material. The platform uses a serverless architecture based on Google Cloud Run to provide scalability, combines adaptive study plans to prioritize important subjects and uses powerful Natural Language Processing (NLP) to analyze user queries.

Because they include interactive courses and study aids, a number of well-known e-learning platforms, like Quizlet, Coursera, and Khan Academy, have established standards for online education. While Coursera offers a variety of academic topics and credentials, Khan Academy is best at offering free, organized courses. Quizlet uses practice exams and flashcards to promote gamified learning.

2. Materials and Methods

2.1. Security Implications and Best Practices for JSON Web Tokens (JWT)

2.1.1. Understanding JWT in Authentication

JSON Web Tokens (JWT) are a compact, self-contained, and cryptographically signed standard for securely transmitting information between a client and a server. They are extensively used for authentication and session management in modern web applications, including platforms utilizing Generative AI like EduRAG. In EduRAG, JWTs play a critical role in authenticating users and safeguarding sensitive operations such as uploading study materials, generating personalized responses, and managing user-specific configurations like study blueprints.

A JWT consists of three parts:

- Header: Specifies the token type (JWT) and the signing algorithm (HS256 or RS256).
- Payload: Contains claims such as user ID, roles, and expiration time. For EduRAG, this includes metadata about access permissions, user activity limits, and session scopes.
- Signature: A cryptographic hash created using the header, payload, and a secret key. This ensures the integrity and authenticity of the token

2.1.2. Security Implications of JWT in Generative AI Applications

While JWTs offer robust authentication, platforms integrating Generative AI have unique security considerations.

Key risks include:

- Token Tampering: Attackers may try to modify the payload to elevate permissions (e.g., accessing restricted GenAI models or bypassing usage limits).
- Token Theft: A stolen token can give an attacker unauthorized access to AI-generated resources or user-uploaded documents.
- Replay Attacks: A valid JWT stolen from a session can be reused by an attacker, exposing sensitive study blueprints or user-specific GenAI results.
- Extended Expiration: Tokens with long lifetimes increase the risk of misuse if compromised.
- Model Access Exploitation: A compromised token could grant illegitimate access to expensive computational resources, such as fine-tuned Generative AI models2.1.3. Best Practices for Securing JWTs

2.1.3. Best Practices for Securing JWTs

To mitigate these risks, EduRAG implements the following practices:

- Secure Transaction: All JWTs are transmitted over HTTPS to prevent interception. AI model interactions, often involving sensitive data or high-value computations, are secured using encrypted WebSocket connections when needed.
- Token Expiration and Refresh Tokens: JWTs are configured with short expiration times (exp claim) to limit the vulnerability window. Refresh tokens are issued for long-running sessions to re-authenticate users without requiring frequent logins.
- Secure Storage: Tokens are stored in HTTP-only cookies with the Secure and SameSite attributes enabled. This prevents Cross-Site Scripting (XSS) attacks from accessing the token. Unlike local storage, cookies minimize exposure to client-side vulnerabilities.
- Token Validation: The server validates tokens using RS256 (asymmetric encryption), where the private key signs the token, and the public key verifies its integrity. Critical claims like exp (expiration), iat (issued at), and aud (audience) are checked against predefined values to ensure validity.
- Revoke Compromised Tokens: EduRAG maintains a token blacklist to revoke access immediately upon logout, session expiration, or suspected compromise. This blacklist is implemented using Redis for real-time lookups, ensuring revoked tokens are invalidated promptly.
- Mitigating XSS Risks: The application incorporates strict Content Security Policies (CSPs) and sanitizes user inputs to prevent malicious scripts from accessing JWTs.
- Fine-Grained Permissions: Tokens include granular access levels (e.g., read-only, admin) to limit operations such as GenAI model invocation or high-cost computation requests.

2.1.4. JWT in the Context of EduRAG

In EduRAG, JWTs are central to ensuring secure and efficient user authentication. They enable the following key features:

- Encrypting Tokens with RS256: Asymmetric encryption ensures secure token generation and validation, with the private key protected on the server and the public key distributed for verification.
- Session Management with Refresh Tokens: Short-lived access tokens reduce exposure risk, while refresh tokens maintain session continuity for operations like document uploads and AI-generated query processing.
- Secure Cookie Storage: Tokens are stored in HTTP-only cookies to minimize exposure to XSS attacks, ensuring a secure client-server communication pipeline.
- Token-Based Authorization:
  - ➢ EduRAG's APIs validate user roles via claims embedded in the JWT. For example:
  - ➢ A student role may query generative models within usage limits.
  - ➢ An admin role can access system-level features like monitoring GenAI usage or managing fine-tuned models.
- Real-Time Token Revocation: A server-side database (e.g., Redis) tracks active tokens, enabling instant revocation in cases of suspicious activity or policy violations.
- Preventing Replay Attacks: Each JWT session includes unique identifiers (JTI claim) tracked on the server. Duplicate or reused tokens are flagged and rejected.

2.2. Database Design and Relationships

Relationships and Database Design a strong and organised database is necessary for AI-enabled learning platforms like EduRAG to function effectively. Dynamic relationships between users, documents, queries, and AI-generated answers must all be supported by the database. In order to manage structured data as well as high-performance search and retrieval, a mix of NoSQL databases (like MongoDB) and Vector databases (like Pinecone FAISS) is used.

The application relies heavily on the following schemas:

- MongoDB Schemas: MongoDB stores metadata, uploaded documents, and user data.
- User Schema:
    - Purpose: Oversees user data, roles, and preferences.
    - Fields:
        - Email: Unique identifier for each user.
        - Password: Securely hashed password for authentication (e.g., bcrypt).
        - Role: Defines whether the user is a student, educator, or admin.
        - Study_blueprints: Array storing study priorities and preferences.
        - Usage_limits: Tracks the number of AI queries made (e.g., to enforce free-tier limits).
    - Relationships:
        - Each user can upload multiple documents.
        - Users can define study blueprints linked to their uploaded resources.
- Document Schema
    - Purpose: Stores metadata about user-uploaded documents (e.g., lecture notes, textbooks)
    - Fields:
        - Title: Name of the document.
        - Type: Document type (e.g., PDF, DOCX).
        - Upload_date: Timestamp of when the document was uploaded.
        - Owner: Reference to the user who uploaded the document.
        - Metadata: Extracted details (e.g., keywords, topics).
        - Vector_id: Links to embeddings stored in the vector database for retrieval.
    - Relationships:
        - Each document is owned by one user.
        - Documents are linked to their vector representations in the vector database.
- Query Log Schema
    - Purpose: Tracks user queries to the AI system and their corresponding responses
    - Fields:
        - User: Reference to the user making the query.
        - Document: Reference to the document queried.
        - Query_text: The natural language query submitted.
        - Response_text: AI-generated response.
        - Timestamp: Query execution time.
        - Metadata: Additional query details (e.g., study blueprint applied).
    - Relationships:
        - Each query is associated with a specific user and, optionally, a specific document.
        - Queries help refine the AI models based on user feedback.
- Study Blueprint Schema
    - Purpose: Captures user-defined study priorities.
    - Fields:
        - User: Reference to the user creating the blueprint.
        - Topics: Array prioritized topics with weightage (e.g., Math - 40%, Physics - 30%).
        - Preferences: Additional preferences like study duration, focus areas, or question difficulty.
    - Relationships:
        - Each user can have multiple study blueprints.
        - Blueprints dynamically influence query responses.

2.2.1. Vector Database for Search and Retrieval

A Vector Database is crucial for efficiently handling document embeddings and retrieving relevant content. For EduRAG, a Vector Database like FAISS or Pinecone stores and index embeddings are generated from user-uploaded documents.

- *Vector Storage Schema*
  - *Purpose:* Stores vector embeddings of document chunks for fast similarity search.
  - *Fields:*
    - Vector_id: Unique identifier for each embedding.
    - Document_id: Links the embedding to a document in MongoDB.
    - Embedding: High-dimensional vector representation of a document chunk.
    - Metadata: Additional details like chunk position, keywords, or topics.
  - *Relationships:*
    - Each embedding corresponds to a specific document chunk.
    - Embeddings are used to retrieve content relevant to user queries.
- *Integration with MongoDB*
  - Embedding Generation: Each document the user uploads is processed and split into chunks. Each chunk is converted into an embedding using a fine-tuned language model (e.g., OpenAI, Hugging Face).
  - Metadata Linking: MongoDB stores document metadata, while the vector database handles high-dimensional embeddings. The two systems communicate using the vector_id.

2.2.2. Considerations for Vector Databases

When designing the vector database integration, the following factors are critical:

- Dimensionality Reduction: Ensure embeddings are optimized for storage and search efficiency.
- Indexing Algorithm: Use Flat L2 or IVF in FAISS for efficient nearest-neighbor searches.
- Scalability: Ensure the system can handle millions of embeddings without degrading performance.
- Latency: Optimize query latency to maintain a seamless user experience.
- Data Updates: Implement batch updates to keep the vector database synchronized with new or modified documents.

2.2.3. Summary of Database Design

The combination of MongoDB and a Vector Database ensures EduRAG achieves:

- Scalability: Efficiently handles large volumes of user documents and queries.
- Personalization: Enables precise content retrieval tailored to individual study needs.
- Security: Ensures sensitive user data remains secure while leveraging advanced AI capabilities.

3. Deployment and Hosting Considerations

3.1. Methods of Deployment

3.1.1. Management of Codebases

Git maintains version control, and development, testing, and production branches are kept apart.

- Main Branch: Production-ready code.
- Development Branch: Integration of features and preliminary testing.
- Hotfix Branches: Fast fixes for pressing production problems

The Continuous Integration/Continuous Deployment (CI/CD) framework is built using Google Cloud Build.

- Build Triggers: These start tests and build automatically when code commits are made.
- Container images are safely stored for deployment in the Artifact Registry.
- Using Google Cloud Deploy, deployment stages are automated in production and staging environments.

3.1.2. Containerization

Docker is being used to containerize the EduRAG application to guarantee consistency between the development, testing, and production environments.

- Backend: Python-based API for RAG pipelines and AI operations.
- Frontend: Google App Engine (Flex) or Google Cloud Storage are used to serve React applications.
- Database: The user and metadata databases are stored in Firestore, whereas the vector embeddings are stored in FAISS or Pinecone.
- Docker Containers for Container Orchestration: o Google Kubernetes Engine (GKE) supports rolling updates, auto-scaling, and health checks for multi-container deployments.

3.1.3. Testing Environments

- Staging Environment: To verify functionality and performance, a staging environment imitates production.
  - ➢ Cloud Run was used for deployment, with limited IAM roles for testing access.
  - ➢ AI model outputs are assessed to guarantee answer relevance and query correctness.
- Automated Testing:
  - ➢ Unit tests verify database queries and AI processing modules as part of testing pipelines.
  - ➢ Integration Tests: Verify smooth communication between database, AI, and backend systems.
  - ➢ End-to-end Tests: Focus on document uploads and query creation while simulating actual user processes.

3.1.4. Security

- Secure Communication: Google Cloud Load Balancing facilitates all communications over HTTPS, guaranteeing end-to-end encryption.
- Secret Management:
  - ➢ Google Secret Manager stores and manages sensitive credentials, such as database credentials, API keys, and JWT signing keys.
  - ➢ The danger of key exposure is decreased by implementing automated key rotation rules.
  - ➢ In order to limit access to production resources and guarantee that only authorized individuals may communicate with vital systems, IAM policies employ fine-grained roles.

3.2. Hosting Considerations

3.2.1. Hosting Platforms

Google Cloud Platform (GCP) serves as the sole hosting platform for EduRAG in order to maximize scalability, availability, and integration.

- Frontend Hosting:
  - ➢ Google App Engine (Flexible Environment) is used to deploy React applications for simple scalability.
  - ➢ A globally distributed Content Delivery Network (CDN) used by Google Cloud Storage serves static assets (such as CSS and JavaScript).

- Backend Hosting:
  - Google Cloud Run, a serverless environment that adapts to user demand automatically, powers the Python backend.
  - Vertex AI hosts AI models for effective training and processing.
  - Database Management: Firestore, a serverless NoSQL database with scalability
  - Optimization houses user data and metadata.
  - Vertex AI Matching Engine or FAISS installed on Cloud Run is used to handle vector embeddings for document retrieval.

3.2.2. Load Balancing

- Google Cloud Load Balancing divides incoming traffic among several backend instances to provide high availability and responsiveness.
- Multi-region failover is configured to reduce downtime.
- Improves latency by integrating with Cloud CDN for static content caching.

3.2.3. Auto-Scaling

- GKE and Cloud Run offer auto-scaling according to memory and CPU use.
- Vertex AI's AI model services adapt automatically to changing query volumes.

3.2.4. Observation and Record Keeping

- For thorough monitoring and logging, Google Cloud Operations Suite (previously Stackdriver) is utilized:
- Monitoring: Keeps tabs on parameters, including query accuracy, AI model utilization, and API response times.
- Logging: Cloud Logging enables centralized logging for all services, facilitating effective analytics and troubleshooting.
- Alerts: Set up to provide proactive incident resolution by alerting users to abnormalities such as high mistake rates or resource constraints.

3.3. Environments for Operation

*3.3.1. Developmen*t

Local Development Configuration:

- Backend and frontend services are provided by developers using Docker containers.
- For testing, AI models are executed locally via a Firestore connection.
- When code changes, hot-reloading guarantees rapid iteration.

3.3.2. Staging

Staging Environment:

- Uses Google Identity and Access Management (IAM) rules to limit external access while replicating production settings.
- AI queries are recorded for performance verification, guaranteeing preparedness before production deployment.

3.3.3. Production

- Cloud-hosted production with high-performance settings and auto-scaling.
- Cloud CDN enhances user experience by speeding up query answers globally.
- Data Backup: For disaster recovery, regularly backup Firestore and embed data in Google Cloud Storage.

4. Conclusion

An important step forward in resolving the issues with current e-learning systems is the EduRAG platform. EduRAG offers customized learning experiences based on user-uploaded content by utilizing cutting-edge technologies such as Retrieval-Augmented Generation (RAG) and cutting-edge AI models. Students and teachers can interact with the platform across devices thanks to its cutting-edge features, which include fluid user interfaces created with Material UI, secure login methods using JWT, and adaptable study designs.

EduRAG's capacity to produce context-specific replies is one of its best qualities; it enables students to learn from their own study materials. By empowering users to take charge of their education, this feature not only improves the learning process but also helps users match the platform with their own academic objectives. Furthermore, in an expanding market for AI-driven educational solutions, EduRAG's emphasis on data privacy and moral AI practices guarantees that private user information is managed safely, fostering credibility and confidence.

References

[1] What is Retrieval-Augmented Generation?, Glossary Index, Retrieval Augmented Generation, nVIDIA. [Online]. Available: https://www.nvidia.com/en-in/glossary/retrieval-augmented-generation/

[2] Retrieval Augmented Generation and Generative AI on SAP BTP, SAP Discovery Center. [Online]. Available: https://discovery-center.cloud.sap/refArchDetail/ref-arch-open-ai

[3] Designing and Developing a RAG Solution, Learn. [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/rag/rag-solution-design-and-evaluation-guide

[4] Uğur Özker, Advanced RAG Architecture, Medium, 2024. [Online]. Available: https://ugurozker.medium.com/advanced-rag-architecture-b9f8a26e2608

[5] Harrison Hoffman, Build an LLM RAG Chatbot With LangChain, Real Python, 2024. [Online]. Available: https://realpython.com/build-llm-rag-chatbot-with-langchain/

[6] Adesoji Alu, 3 Proven Methods for Real-Time Voice Transcription Success: Balancing Precision and Performance in Critical Industries, Collabnix, 2024. [Online]. Available: https://collabnix.com/3-proven-methods-for-real-time-voice-transcription-success-balancing-precision-and-performance-in-critical-industries/

[7] Cem Dilmegani, How to Build a Chatbot: Components & Architecture, AI Multiple Research, 2024. [Online]. Available: https://research.aimultiple.com/chatbot-architecture/

[8] Retrieval Augmented Generation, Databricks. [Online]. Available: https://www.databricks.com/glossary/retrieval-augmented-generation-rag

[9] Retrieval Augmented Generation (RAG), Cloudflare Docs. [Online]. Available: https://developers.cloudflare.com/reference-architecture/diagrams/ai/ai-rag/