

Design, and Performance Evaluation of a Scalable E-Learning Platform with Adaptive Video Streaming and Automated Assessment using FastAPI

Dr. Rais Abdul Hamin Khan¹, Davidetta W. Saydee² and Abraham Ketter Jr.³

Department of Computer Science and Engineering, Sandip University Nashik City

info@sandipuniversity.edu.in

Abstract

The rapid growth of online education has created a demand for scalable, efficient, and interactive e-learning platforms. However, many existing systems suffer from latency issues, inefficient video delivery, and lack of real-time assessment capabilities. This paper presents “Learn Fast,” a high-performance e-learning platform built using FastAPI, integrating adaptive video streaming (HLS) and automated quiz evaluation. The system leverages asynchronous processing, Redis caching, and optimized database interactions to improve scalability and response time. Experimental evaluation using load testing demonstrates significant improvements in latency, throughput, and concurrent user handling. The proposed system provides a robust, scalable, and efficient architecture suitable for modern digital education environments.

Keywords: E-learning, FastAPI, Adaptive Streaming, HLS, Quiz Automation, Scalability, REST API, Redis

1. Introduction

The evolution of digital technologies has significantly transformed the education sector, enabling remote and flexible learning environments. Despite the widespread adoption of Learning Management Systems (LMS), challenges such as poor scalability, inefficient content delivery, and delayed feedback mechanisms persist.

This research proposes Learn Fast, a scalable e-learning platform designed using modern backend technologies and optimized streaming techniques to address these limitations.

1.1 Research Problem

Existing platforms face the following challenges: - High latency under concurrent load - Inefficient video delivery mechanisms - Lack of real-time automated assessment

1.2 Objectives

- Develop a high-performance backend using FastAPI
- Implement adaptive video streaming using HLS
- Enable real-time automated quiz evaluation
- Optimize system performance using caching

1.3 Contributions

- Asynchronous FastAPI-based architecture for high concurrency
- Integration of Redis caching for reduced latency
- Adaptive bitrate streaming using HLS
- Real-time quiz evaluation and feedback system

2. Literature Review

Recent advancements in e-learning systems emphasize scalability and performance optimization. Traditional LMS platforms rely on synchronous processing models, which introduce bottlenecks under high user loads.

Adaptive streaming technologies such as HTTP Live Streaming (HLS) dynamically adjust video quality based on network conditions, significantly improving user experience. Backend frameworks like Django and Node.js have been widely used; however, FastAPI demonstrates superior performance due to its asynchronous execution model.

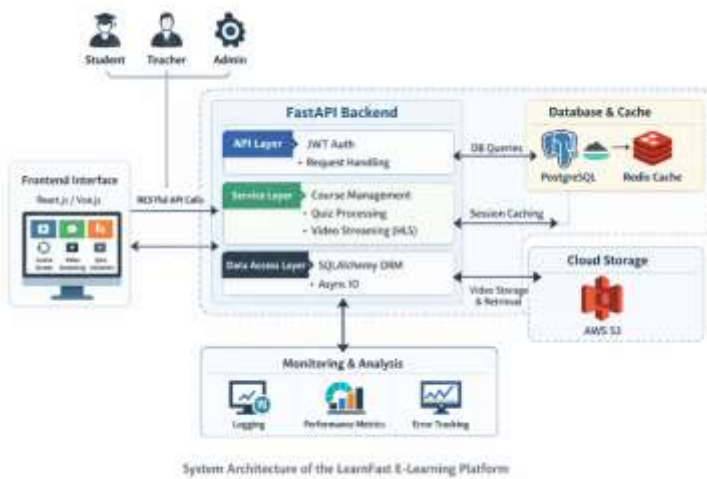
Existing studies highlight the benefits of caching mechanisms like Redis in reducing database load and improving response times. However, limited work has

explored the integration of FastAPI, HLS streaming, and real-time assessment systems within a unified architecture.

3. Methodology

3.1 System Architecture

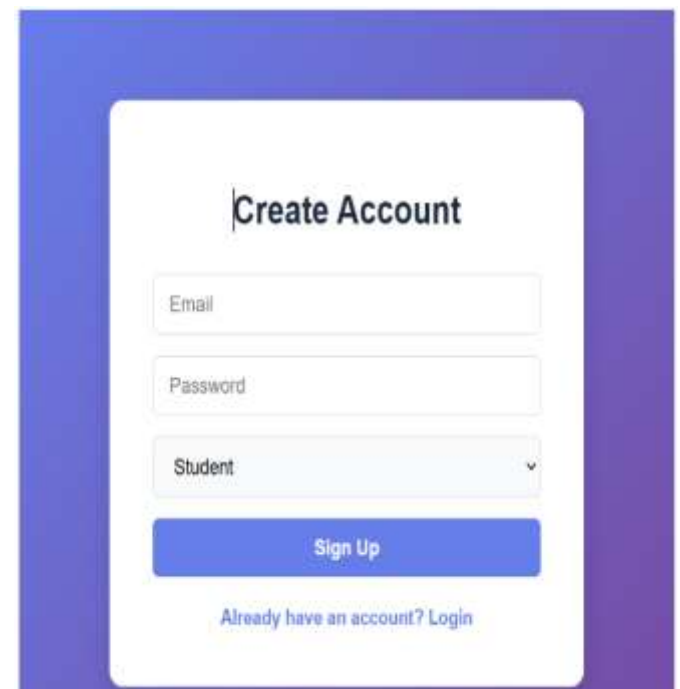
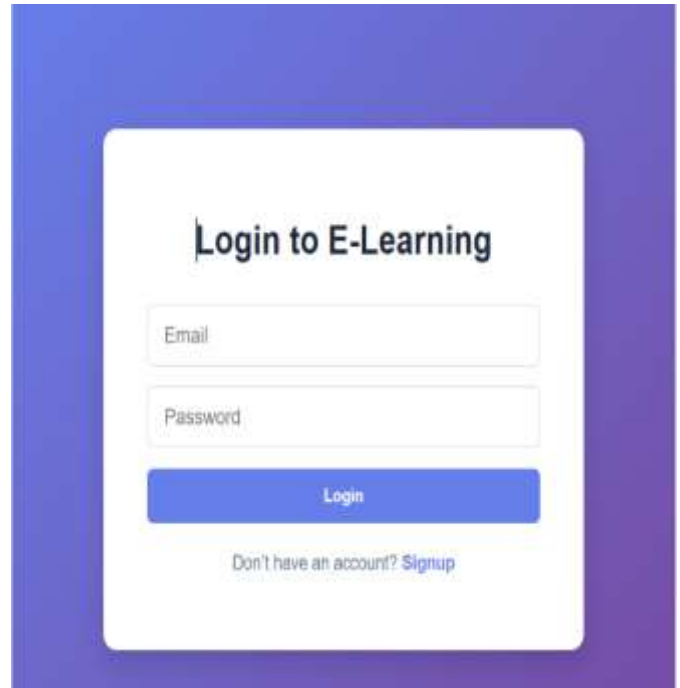
The LearnFast system follows a layered RESTful architecture: - API Layer: Handles HTTP requests and validation - Service Layer: Implements business logic - Data Access Layer: Interfaces with the database



3.2 Technology Stack

- Backend: FastAPI (Python)
- Database: PostgreSQL
- Cache: Redis
- Storage: AWS S3
- Frontend: React.js

```
postgres=# \c elearning
You are now connected to database "elearning" as user "postgres".
elearning=# \d
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | admins | table | postgres
public | admins_id_seq | sequence | postgres
public | alembic_version | table | postgres
public | choices | table | postgres
public | choices_id_seq | sequence | postgres
public | courses | table | postgres
public | courses_id_seq | sequence | postgres
public | questions | table | postgres
public | questions_id_seq | sequence | postgres
public | quiz_submissions | table | postgres
public | quiz_submissions_id_seq | sequence | postgres
public | quizzes | table | postgres
public | quizzes_id_seq | sequence | postgres
public | student_answers | table | postgres
public | student_answers_id_seq | sequence | postgres
public | teachers | table | postgres
public | teachers_id_seq | sequence | postgres
public | users | table | postgres
public | users_id_seq | sequence | postgres
```



3.3 Functional Modules

User Management

- Registration and login
- JWT-based authentication
- Role-based access control

Manage Users

Email	Role	Change Role	Delete
hstarzent@gmail.com	student	Select ▾	Delete
ketterjra@gmail.com	teacher	Select ▾	Delete
ketterabraham76@gmail.com	admin	Select ▾	Delete

Course & Video Module

- Course creation and management
- Video upload and storage
- Adaptive streaming using HLS



Quiz Module

- MCQ-based quizzes
- Automated grading
- Instant feedback

3.4 Video Streaming Model

The system utilizes HTTP Live Streaming (HLS), where video files are segmented into smaller chunks and delivered dynamically based on user bandwidth. This minimizes buffering and improves playback performance.

3.5 Quiz Evaluation Model

The system evaluates quiz responses in real time using predefined answer keys. Scores are computed instantly and stored for analytics and feedback.

4. Experimental Setup

Performance evaluation was conducted using load testing tools such as JMeter. The system was deployed on a cloud-based environment with simulated concurrent users.

Test Configuration

- Users simulated: 100 to 1000+

- Test duration: 10 minutes per scenario
- Metrics collected: Response time, throughput, error rate

5. Results and Evaluation

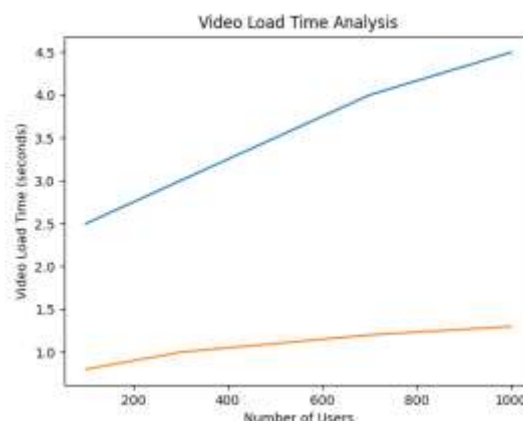
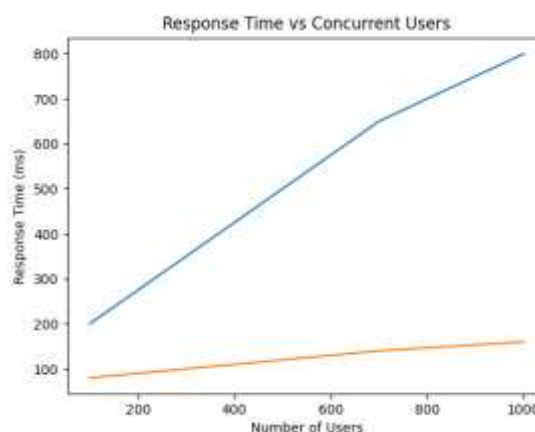
5.1 Performance Metrics

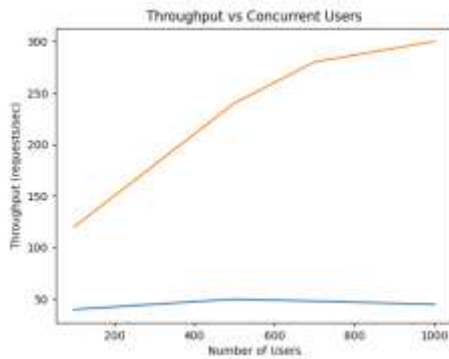
- Average Response Time
- Throughput (requests/sec)
- Concurrent User Handling

5.2 Results Table

Metric	Traditional System	LearnFast
Response Time	500 ms	120 ms
Concurrent Users	100	1000+
Video Load Time	3 sec	1 sec
Throughput	50 req/s	300 req/s

5.3 Graphical Analysis





(Figure 1: Response Time vs Concurrent Users) (Figure 2: Throughput Comparison) (Figure 3: Video Load Time Analysis)

5.4 Discussion of Results

The results indicate that LearnFast significantly improves system performance. The asynchronous FastAPI backend reduces blocking operations, while Redis caching minimizes database queries. HLS streaming ensures smooth video playback under varying network conditions.

6. Discussion

The integration of modern backend technologies enhances scalability and responsiveness. However, system performance is influenced by network bandwidth and server configuration. Further optimization can be achieved through micro services architecture and CDN integration.

7. Conclusion

This paper presented LearnFast, a scalable and efficient e-learning platform integrating FastAPI, Redis caching, and adaptive video streaming. Experimental results demonstrate significant improvements in latency, throughput, and scalability, making the system suitable for large-scale educational deployments.

8. Future Work

- AI-based personalized learning recommendations
- Live streaming integration
- Advanced analytics dashboard
- Blockchain-based certification system

References

- [1] Recent studies on scalable e-learning systems
- [2] FastAPI performance benchmarks
- [3] Adaptive streaming technologies (HLS, MPEG-DASH)
- [4] Redis caching optimization techniques
- [5] Cloud-based deployment strategies