**Abstract**

The COVID-19 pandemic changed the world in various aspects, especially in the education industry. Education expanded beyond the traditional classrooms, into the virtual world. The virtual learning management environment gained significant importance during this pandemic, which resulted in wider dissemination of education. People were introduced to tools and technologies which eventually became crucial not only in the education sector but also in the commercial industry, where companies can leverage such LMS platforms to enhance collaboration. We explore the current learning management systems used across different sectors, identify the shortcomings, and recommend a better system to address these vulnerabilities. The proposed model, mainly targeted towards the education sector, integrates video conference management, class segregation, role-based access control, class scheduling and user authentication as the core features. It aims to make the proposed web-based applications as accessible, dynamic, interactive and flexible as possible for both students and instructors. This model uses Python, HTML, JavaScript and CSS under Django’s framework. It stands out from other existing platforms by offering a highly personalized learning path to each student, along with the system of automatic assignment grading and curated library resources. By revolutionizing digital learning with personalization, automation and engagement, it enhances efficiency, understanding and transforms a still-standing education system into a much more accessible and interactive student-instructor experience.

**Introduction**

Education is the bedrock for the progress of an individual and a society as a whole. During the three year duration of the COVID-19 pandemic, from 2020 to 2023, the world witnessed a very rapid change in various aspects of regular life. The education sector came to a sudden standstill and then rapidly transformed into digital classrooms from the traditional setup. This sudden requirement for technical support was delivered by several LMS platforms at that time, by providing structure and interaction platform between students and instructors. However, with the rapid increase in technology, most of the LMS platforms only perform the basic requirements, and fail to offer advanced support. Majority of the LMS platforms in current usage have several shortcomings in various aspects. Their rigid and inflexible structure offers little to none customisation. UI/UX is outdated as well and most of them are not inclusive of users with cognitive disabilities, making them less engaging and accessible. Almost none of them offer student analytics reports to the instructors which makes it difficult for the instructors to identify the struggling students, especially in a virtual classroom where they cannot see a student. Additionally, high cost involved with certain platforms makes it difficult for certain institutions to access them on a large-scale to accommodate all students. Despite their several advantages, their limitations make the process of learning static and ineffective, being a hindrance to both students and instructors. To overcome these issues, we propose a modernized LMS platform which prioritises personalization, automation, inclusivity and flexibility as the top features supported by an optimised core model. Our model aims to foster a healthy and engaging learning environment in a virtual space. The proposed model has been made utilising Python frameworks for the backend logic, for their speed of execution, rapid development and scalability; native HTML, CSS and JavaScript for faster request processing, browser compatibility and very little overhead; Firebase database and cloud storage for scalable and real-time synchronisation paired with robust security; and pre-trained machine learning model APIs for faster development, improved accuracy and reduction in the amount of computational resources that would be used otherwise. Incorporation of Augmented Reality for user engagement and better conceptual understanding along with personalised content based on user activity. The video conference core model was predominantly built using WebRTC (Web Real-time Communication) APIs and protocols, including ICE candidates and RTCPeerConnection. The STUN/TURN server usage ensures a seamless network during the conference. We use pre-trained AI models to produce analytical reports of every individual student and present them to the instructors using which they can decide what they need to do eventually.

**Requirements**

| **REQUIREMENT** | **CATEGORY** | **PURPOSE** |
| --- | --- | --- |
| HTML | Markup Language  (Frontend Development) | To design the website layout. |
| CSS | Stylesheet Language  (Frontend Development) | To style the website. |
| JavaScript | Programming Language | To create dynamic web pages. |
| Django | Framework | To handle backend logic. |
| Firestore | NoSQL Database  (API) | To store the data in the backend. |
| Firebase Auth | API | User authentication |
| Python 3.13 | Programming Language | Primary language used for development. |
| OpenAI | API | To extract keywords from pdf, summarizing content, creation of flashcards,AI powered Q&A, and difficulty classification. |
| JSON | Data Formatting | Easy parsing of objects and data for human readability. |
| AsyncWebSocketConsumer | Protocol Libraries | To establish bidirectional real-time communication. |
| RTCPeerConnection | Protocol Libraries | To establish peer-to-peer connection. |
| RTCIceCandidates | Protocol Libraries | To represent network paths for peer-to-peer connections. |
| STUN/TURN servers | Server | Provides relay services to facilitate peer-to-peer communication. |
| REDIS server | NoSQL Database | To serve as an in-memory data store for session management. |
| VSCode | Integrated Development Environment (IDE) | For construction of the platform. |

**Literature Review**

Anaraki (2004) highlighted that an effective LMS must give priority to usability, interactivity, and scalability, but the majority of platforms fail to accommodate personalized learning styles and content suggestions. We will be integrating personalized learning paths and user activity-based content suggestions in the LMS [1].

Role-Based Access Control (RBAC) is a feature that will be used to control user permissions under this LMS. StudyGyaan (2023) gave a practical tutorial on how to create several user roles namely students, instructors, and administrators so that users would access only those features pertaining to their respective roles in Django [2]. Likewise, the Mozilla Developer Network (MDN) (2022) gave a full-featured tutorial on user authentication and permission management using Django [3].

For authenticating the users, we will be using Firebase Auth which is a simple-to-use API and facilitates the user login with their Google accounts in contrast to some LMS having fixed user credentials to log into their interface and Firebase also supports a real-time database known as Firestore which will be provide instant updates and is better than traditional SQL-Based LMS models [4][5].

Akçayır & Akçayır (2017) cites that AR-based learning will increase student involvement and cognitive understanding of complicated topics. Several LMS platforms do not support 3D models that could demand supplementary plugins or manual file uploads. We are going to incorporate a 3D model uploading repository and web scraping the existing models from public repositories [6].

Deshmukh et al. (2023) have discussed the application of WebRTC technology for video conferencing that employs real-time communications; it takes advantage of peer-to-peer communication to minimize latency and enhance interaction quality [7].

WebRTC (Web Real-Time Communication) enables real-time audio, video, and data communication in web applications without plugins. Supported by major browsers, WebRTC implements three main APIs: MediaStream (getUserMedia), RTCPeerConnection, and RTCDataChannel. WebRTC prioritizes security, accessibility, and decentralization, with applications in video chat, file transfer, gaming, and more. [8]

Django is a high-level Python web framework which is known for its rapid development and clean design features. It follows the Model-View-Template (MVT) architecture, offering admin interface and robust security features, amongst many such other qualities. Django is a highly scalable and secure framework, which is why it is ideal to develop data-driven applications on this platform. [9]

**References**

[1] Anaraki, F. Developing an Effective and Efficient eLearning Platform. International Journal of The Computer, The Internet and Management, 12(2), 57-63. (2004).

[2] StudyGyaan: How to create RBAC in Django with multiple user types. Retrieved from

<https://studygyaan.com/django/how-to-create-rbac-in-django-with-multiple-users-types> (2023).

[3] MDN Web Docs: Django Tutorial Part 8: User authentication and permissions. Retrieved from <https://developer.mozilla.org> (1998-current).

[4] Firebase. Firebase Authentication: Secure and Scalable User Authentication. Google Developers <https://firebase.google.com/docs/auth> . (2011-current).

[5] Firestore. Cloud Firestore: NoSQL Database for Serverless Applications. Google Cloud Documentation <https://firebase.google.com/docs/firestore> . (2011-current).

[6] Akçayır, G., & Akçayır, M. Advantages and Challenges Associated with Augmented Reality for Education: A Systematic Review of the Literature. Educational Research Review, 20, 1-11. (2017).

[7] Deshmukh, R., Nand, N., Pawar, A., Wagh, D., Kudale, A.: Video conferencing using WebRTC. In: 2023 International

Conference on Sustainable Computing and Data Communication Systems (ICSCDS), pp. 857–864 (2023).

[8] Dutton, Sam. "Getting started with WebRTC." *HTML5 Rocks* 23 (2012).

[9] Django Software Foundation. *Django Documentation.* <https://docs.djangoproject.com/en/5.1/> (2005-current).