



# Virtual Classroom on Cloud: An Integrated System for Course Management and Digital Learning

<sup>1</sup>Alan B Mohan,<sup>2</sup>Ghansyam A,<sup>3</sup>Karthik M, <sup>4</sup>Manav M Nambiar, <sup>5</sup>Prof. Rakesh Sharma

<sup>1</sup>Final year B.E. Student, <sup>2</sup> Final year B.E. Student, <sup>3</sup>Final year B.E. Student, <sup>4</sup>Final year B.E. Student, <sup>5</sup> Assistant Professor, CSE

<sup>1</sup>Department of Computer Science & Engineering,

<sup>1</sup>Srinivas Institute of Technology, Mangaluru, India

**Abstract:** Using this cloud-based "Virtual Classroom," the entire system tries to support digital education on an interactive and scalable platform. The system backend was developed in Flask with Supabase for cloud storage and database management. The system offers role-based access, secure file hosting, a modular system for managing content, creating teachers and courses, uploading notes, uploading submissions, grading online, scheduling online meetings, etc. Study material is provided to students who do assignments and check for their assessments. It lays emphasis on quick file management, real-time assignment status, and responsive UI. With user authentication and role-specific dashboards, it intends to create richer online learning experiences and bridge the gap between traditional and virtual education

**Keywords:** Cloud Computing, Virtual Learning, Flask, Supabase, Assignment Management, Online Education, Educational Platforms.

## I. INTRODUCTION

With technology flexibility rapidly emerging while being easily accessible, information nowadays necessitates a shift from the paradigm of the traditional classroom setting. Such environments need to be created where there is not only the availability of online books, but also the systematic and secure administration of the different learning resources. The core focus of this work, therefore, is to delineate a cloud supported educational system which incorporates course management and real time meetings alongside document cloud storage enabling seamless assignment submissions termed as the Virtual Classroom on Cloud.

In a conventional academic setup, the handling of course content, student submissions, and grading, are accompanied by a bulky administrative overhead. This system is inefficient and does not take full advantage of emerging technologies. Moreover, it has a potential to disengage students due to strict deadlines and increased administrative work which in turn impedes a teacher's ability to meet the learners' needs. The simplicity and elegance of contemporary web technologies, alongside a robust cloud infrastructure, sought to address the myriad concerns dealing with the education process as a whole. Virtual Classroom on Cloud employs auto-grading features, real-time monitoring of students, role-specific access control, and collaborative tools ensuring a step forward towards a more effective teaching and learning experience on digital environments.

## II. LITERATURE REVIEW

Existing Application of cloud computing in schools has been extensively researched and established in recent studies. Cloud learning environments offer high scalability, real-time collaboration, and remote access to learning materials. Virtual classrooms built on cloud infrastructure, as Ramos et al. (2018) argue, offer seamless communication between teachers and students using video conferencing, document sharing, and collaborative tools. The integration of cloud storage platforms such as Supabase, Google Drive, and Dropbox in educational systems enables learners to access materials at all times, allowing continuity and flexibility in learning.

Beyond content accessibility, modern education systems have embraced interactive functionalities such as live video, chat interfaces, and virtual whiteboards to provide an emulation of the physical classroom experience. Hughes et al. (2019) state that synchronous learning with the use of tools such as Zoom and Google Meet improves the level of student engagement and rates of retention. Moreover, the assessment tools in the online form are changing the way assignments are submitted and graded. Moodle and Google Classroom have been the standard-bearer for course management and giving formal feedback to students.

Security and privacy remain the topmost priority even in cloud classrooms. Yadav and Kumar (2018) emphasize the importance of employing multi-factor authentication, encrypted data transfer, and role-based access controls to safeguard sensitive information such as student grades and attendance records. Emerging technologies such as Artificial Intelligence (AI) and Machine Learning (ML) are being integrated into virtual classrooms to provide personalized feedback, automate grading, and offer

customized learning experiences. These findings form the basis of the proposed system, which informs its design in line with modern pedagogical standards.

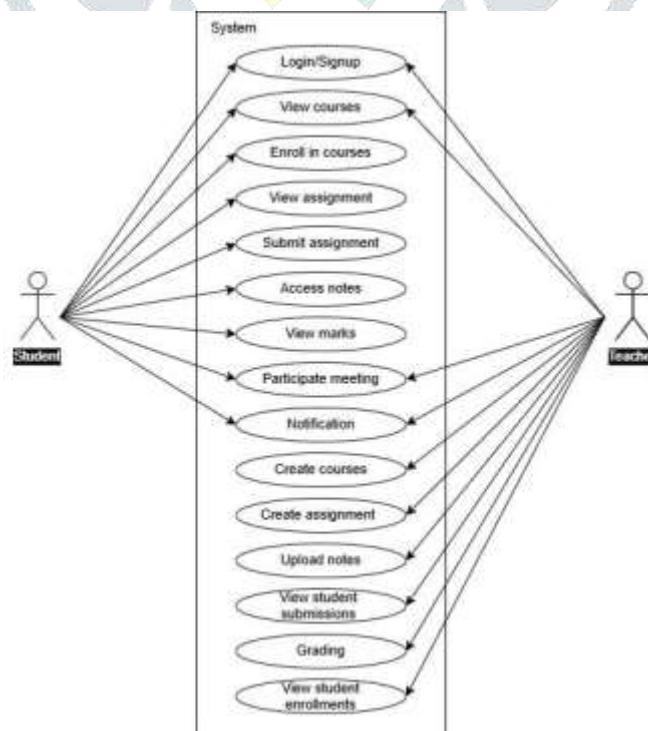
### III. PROPOSED METHODOLOGY

The Cloud Virtual Classroom is implemented as a full-stack, modular web application with emphasis on ease of use, accessibility, and scalability. Client-server architecture is employed in the core design, where client-side operations are handled through a responsive frontend built with Bootstrap, HTML, and CSS, and all application logic and data processing are in the backend, which is implemented using Python's Flask framework. Flask was employed for its lightweight nature, routing flexibility, and seamless integration with web development components. The backend communicates with a PostgreSQL database to save and fetch structured data with SQLAlchemy as the Object Relational Mapper (ORM) to manage complex relationships among entities such as users, courses, assignments, and submissions.

Role-based access control (RBAC) is utilized by the system to differentiate between two major types of users: teachers and students. This access control is embedded at frontend (via conditional rendering) and backend (via decorators and access checks) to support strict role adherence. Upon registration, each user is assigned a role that controls their interface and permissible actions in the system. Instructors have access to modules through which they can create courses, share lecture notes, generate assignments with deadlines, grade assignment submissions, and schedule live meetings. Students, on the other hand, can view and enroll in courses listed, download files, submit assignments, and join live virtual sessions.

For managing learning resources like assignments and notes, the system includes Supabase a horizontally scalable open-source cloud backend that has both file storage and relational database operations support. The files uploaded by teachers or students are retained in Supabase buckets and associated with their respective database records via signed URLs for secure and efficient access without explicit file system operation. The cloud-first design lessens server load, improves data resilience, and makes the application scale elastically by nature.

The flow of application data is structured and well-validated by well-defined endpoints. RESTful APIs handle form submission, user login and logout, data queries, and file uploads. Transactions are logged and tracked to obtain data integrity as well as traceability. Forms are validated by Flask-WTF, which not only ensures proper input but also protects against common web attacks such as CSRF (Cross-Site Request Forgery).



**Fig.1: use case diagram**

Figure 1 The user is redirected to the registration page where he enters his details like username, password and role. After registration process is done user is redirected to login page where he enters his login id and password. User then accesses the dashboard. Whenever the user of whenever application or admin tries to do any activity, it is validated and fetched from database.

### IV. IMPLEMENTATION DETAILS AND TECHNOLOGIES

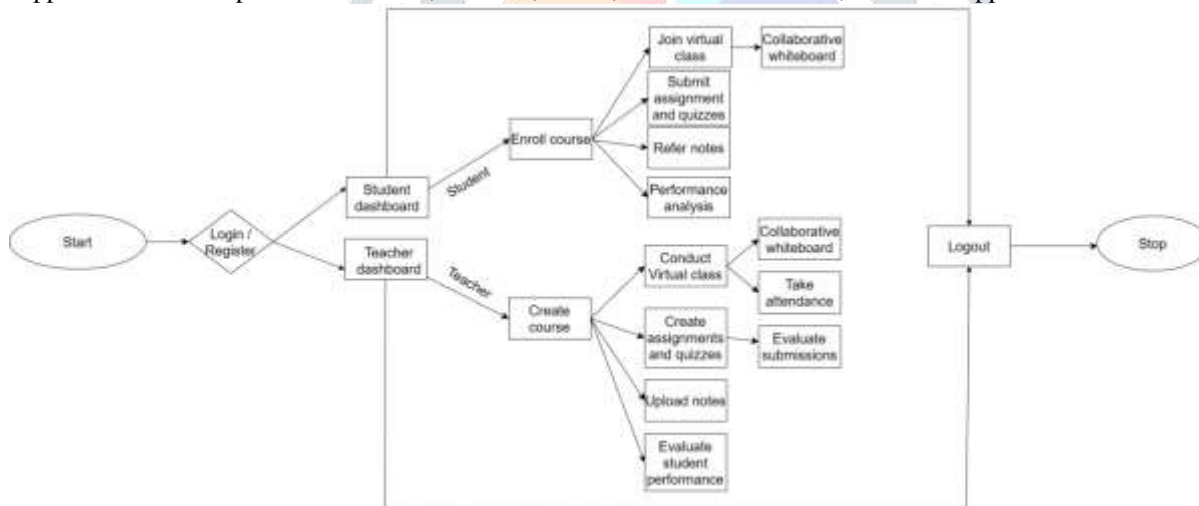
The deployment of Virtual Classroom on Cloud system involves the establishment of a fully functional, full-stack web platform wherein a number of backend and frontend technologies are brought together to create a collective virtual learning space. The application is deployed using Flask, a lightweight Python web framework in favor of rapid development and clean separation of concerns. Flask handles routing, request processing, sessions, and database interaction through the assistance of SQLAlchemy,

an ORM that expresses SQL commands in terms of Python class-based objects. The application follows the MVC (Model-View-Controller) architecture to encourage maintainability, code reusability, and extension modularity.

The system's backbone is a PostgreSQL database holding all the fundamental entities like users, courses, notes, assignments, submissions, and meeting sessions. Each table is structured with proper primary and foreign key relationships for referential integrity. For example, each assignment record has a specific course and instructor linked to it, and each submission belongs to both a student and an assignment. The relationships are specified and CRUD operations handled across the application using SQLAlchemy.

The UI is constructed using HTML5 and styled using CSS3 and Bootstrap framework. Bootstrap is responsive design that scales perfectly on various screen sizes, like mobile phones and tablets. The frontend is dynamic and role-sensitive, i.e., the dashboard and navigation menus are different depending on whether the logged-in user is a teacher or a student. Instructors get additional functionality like course creation, assignment uploads, grading interfaces, and meeting control, while students see bare-bones dashboards with note access, assignment view, submission view, and join links for live sessions. User login and session management are handled through Flask-Login and Flask-WTF. User sessions are stored securely by Flask-Login, and Flask-WTF integrates CSRF protection and form validation. These libraries help with secure login procedures, protect against injection attacks, and offer form integrity. During registration, each user inputs credentials and selects a role. These are stored securely and fetched to design custom dashboards upon login.

One of the strongest aspects of the implementation is integration with Supabase, an end-to-end backend-as-a-service platform that supports both file storage and database functionality. All assignment files and lecture notes that have been uploaded are stored in Supabase buckets. The files are accessed against related records in the database by using signed URLs, which provide access only to legitimate users. This avoids server-side local file management and improves scalability and redundancy of data. Files are supported to be uploaded in PDF, DOCX, TXT, and ZIP formats, which support diverse academic use cases.



**Fig.2: Data flow diagram for virtual classroom**

Figure 2 illustrates one of the unavoidable factors of understanding the internal operations of the Virtual Classroom on Cloud is the Data Flow Diagram (DFD), presented through Figure 2. This diagram presents a top-level abstraction of data moving through different modules of the system and focuses on the control flow between users, the system interface, and back-end procedures.

The DFD freezes the interaction amongst the key system players students, teachers, and the administrator as well as the key components of the application. Every user causes specific actions using the graphical frontend of the application. For instance, a student may choose to submit an assignment, while a teacher may post course notes or create a new assignment. All these inputs are passed to the Flask-based backend logic, validated, and executed.

## V. RESULTS AND DISCUSSION

The system deployed, Virtual Classroom on Cloud, was tested by simulating regular usage patterns for both teacher and student roles. The testing was to verify correctness of operations, responsiveness of interfaces, the performance of the system with regular load, and the overall user experience. The results were documented using live screen shots and verified using iterative test cycles.

Upon app startup, users are greeted with a registration page with support for secure role-based user registration. Users can sign in after they have been registered and get redirected to user dashboards determined by roles. The student dashboard provides access to completed courses, uploaded notes, submitted assignments, and current meetings. The instructor dashboard, however, allows course creation and management, class material uploads, assignment and deadline generation, tracking of student assignment submissions, and initiation of online meetings. The above dashboards are responsive and both desktop and browser-tested on mobile phones.

Performance-wise, the system had consistent response times across all modules that were subjected to testing. Back-end operations like form validation, file activity, and database lookup executed within acceptable latencies under typical user loads. Front-end responsiveness was preserved even when multiple operations (like file upload and dashboard refresh) were performed

simultaneously.

## VI. CONCLUSION

The Virtual Classroom on Cloud solution convincingly demonstrates the viability of web development and cloud technology being employed to create an effective digital learning platform. By introducing features like tracking assignments, management of cloud files, and instant communication, it offers a seamless learning experience to teachers as well as students. The system's architecture facilitates scalability, simplicity, and modular enhancement.

Future plans include the inclusion of a dedicated mobile app to maximize accessibility, implementation of AI-powered performance monitoring, multimedia support of video lectures, and integration with leading LMS platforms like Moodle. Offline support, language localization, and assistive technologies like screen readers will further maximize the accessibility of the system. These will further maximize the platform's potential as a reliable and innovative solution for modern education..

## VII. REFERENCES

- [1] Armbrust, M. et al. "A View of Cloud Computing," Communications of the ACM, 2010
- [2] Allen, I.E., Seaman, J. "Digital Learning Compass: Distance Education Enrollment Report 2017."
- [3] Yadav, S., Kumar, R., "Security in Online Classrooms," IJCSIS, 2018.
- [4] Müller, L., et al. "AI-Driven Learning Pathways in Education," Journal of AI in Education, 2021.
- [5] Flask Documentation: <https://flask.palletsprojects.com>
- [6] Supabase Docs: <https://supabase.com/docs>
- [7] Bootstrap: <https://getbootstrap.com>
- [8] SQLAlchemy ORM Docs: <https://docs.sqlalchemy.org>