

We Drive: The Simplest way to share learning materials with students

Durevash Marathe¹, Santosh Bongu², Meet Beewal³, Mukool Bhagwat⁴, Miss Rupali Pashte⁵

durvash.s.Marathe@slrtce.in,santosh.i.bongu@slrtce.in,meet.r.beewal@slrtce.in,

mukool.k.bhagwat@slrtce.in , rupali.pashte@slrtce.in

IT, MUMBAI UNIVERSITY

Abstract: The massively accelerated (and exponential) growth of digital education raised the bar for fast, robust, and secure file distribution systems, especially those that make seamless two-way transfers possible between Educators and their students. To address this urgent need, this work carefully presents the thorough design and solid implementation of a sophisticated web-based file transfer platform that enables teachers to acquire and intuitively send assignment files to students in an easy way via integration with popular cloud storage services. To this end, the system does not delight in advanced authentication that is absolutely based on institutional email accounts — and access to the platform can only be granted by well-established verification of appropriate authorization users — which also enhances general security while ensuring safe connection. Additionally, with authorization processes that are secure and standardized (like OAuth 2.0) and integration with the Google Drive API, there is very granular sharing functionality as well as optional delivery of files copied into each student's own secured space on the original file repository which provides both flexibility in use and security from unauthorized use or copying of educational materials. The platform allows teachers to upload assignments, select target recipients, and initiate file sharing operations, while students receive files directly in their cloud storage environment or through shared access links. This approach reduces manual distribution effort, enhances accessibility, and ensures data security and traceability. [1]

Keywords: Digital Education, File Distribution Systems, Google Drive API, Institutional Authentication, Automation

1. INTRODUCTION

With the rise of digital technology in education, the methods by which academic work is created,

disseminated and accessed have evolved. Handing out assignments via email or by requiring students to upload assignments directly to a learning portal can be long-winded and chaotic. Keeping track of file transmission and ensuring that the files are kept in an organized fashion becomes a chore. The more the students and digital files increase, the more a robust method of file sharing becomes necessary.[2]

To overcome these shortcomings, there is the demand for a light weight, minimal overhead, file transfer system that omits the use persistent storage systems. This work proposed an internet based file transfer system that is used in the academic context, where the file needs to be transmitted from the instructor to the learner without being stored on the server or on any cloud storage services. Removing the need for storage reduces the infrastructure associated overheads and security issues.[2]

This paper introduces a web based file transfer system which tries to make file transferring between teachers and students much more easier. Files are shared between the two users via the intermediate. Instead of saving to some server or using a cloud based infrastructure, the files are shared directly. The intermediary be simply described as a simple file sharing system, a simple but effective and simple way to transfer files. This kind of system is not expensive to setup and to maintain , as there is not need to any special storage infrastructure. The infrastructure itself is web based and should be scalable and easy to adapt to different and varying requirements of any education environment.[2][3]

2. PROBLEM STATEMENT

1.Storage Dependency Issue: Existing ways of sharing files in the academic environment are through cloud storage or server-based sites, where the file needs to be previously uploaded. This process causes a higher cost for file storage, redundancy of data and the

dependence of a third-party service. Furthermore, the fact that many of the files are sensitive academic material, the privacy of data is also at stake.[3]

2. Inefficient File Distribution: Managing all the assignments given to the students, by each class teacher in any of these methods separately, is a very tedious and time-consuming task, especially since the number of students is growing continuously and it is also a difficult task to verify that every student got the files without centralized control. So we need a system that manages this task automatically.[3]

3. Authentication and Access Control Challenges: But in most of the current system there is a problem in providing the file only to the known students. If the file is provided free of cost then there is a chance of losing the confidentiality of the required file and the others who have already accessed the file may misuse the required file. So the transfer should be made efficient with proper authentication.[3]

4. Lack of Lightweight Scalable Solutions: Most file-sharing architectures require heavy infrastructure such as large storage space and the continuous maintenance of the data. Over-comprehensive or heavy infrastructure won't be affordable enough for small institutes. Lack of an architecture that requires space and scalable file transfer system, no such architecture exists which can reduce the lot to facilitate rapid transferring of files as well as provide scalable and independent file transfer.[7][3]

3. LITERATURE SURVEY

1. Developing and Deploying Large-File Upload Services of Personal Cloud Storage (IEEE, 2025) Li et al designed a solution that addressed media-heavy collections of large-file upload services in personal storage cloud. The traditional form-based uploading was ill-suited for media-heavy collections. This paper also identified the difficulties of chunked uploading, progress reporting and backend handling across multiple vendors such as Google and Baidu, which are applicable to SLRTCE File Manager that uses Google Drive API to upload files of various sizes and types.[4]

2. Online Document Management System for Academic Institutes (IEEE, 2022) This paper has tackled the increasing problem of searching out documents on huge digital bodies of electronic documents in academic environments, by showing that students and teachers require arranged systems that preserve and retrieve important documents and transfer them securely. The researchers argued for an online document management system that would foster knowledge sharing within students and improve the effectiveness of the study process. The paper has

proved that web-based document management systems are vital tools for storing and organizing large quantities of digital documents in institutions.[4]

3. Document Management System (IEEE, 2024) : Introduction on a Web-based Document Management System (Electronic Archive) at the General Directorate of Personnel of the Ministry of National Education, and if the system allows efficient management, surveillance and delivery of official documents in this mode. The electronic archive solved the problem of storing official documents safely on electronic supports, managing them through role-based accesses, and channeling them through the different responsible organs of the organization. The paper has shown that management systems for web-based files do diminish the expenses linked to archiving and printing of documents and ease access to document contents by a factor of 10 in distributed organizations.[5]

4. Google Drive API for File Storage and Sharing (Google Developers Documentation, 2020): According to instructions made available by Google, the Google Drive allows developers to implement file storage and sharing features, and provides a listing of the relevant features such as file sharing. It also describes how a developer could implement a method that would otherwise be a massive task for users to distribute a file multiple times across an educational platform by automating file sharing using access rights allocations.[6]

5. On Automated Data Import using Spreadsheet Integration (Microsoft Documentation, 2018):

The Microsoft articles on spreadsheet integration described approaches to 'Import structured data [from MS Excel] into various databases'. This article describes how batch data transfer operations can be used to decrease errors and improve efficiency in large data management, and is applicable to this system in the context of creating many individual records of students, without using a form repeatedly.[5]

4. OBJECTIVES

The main motivation of the suggested system is to propose a convenient, speed, safe and scalable file transfer system for academia which will allow easy sharing of assignment files between lecturers and students. This system will ease the process of file sending and receiving with ensuring its availability.[7]

1. To create a web-based system that makes it easy for teachers to upload files and send to specific students in the class.

2.To provide secure log in by authenticating users by their institutional email account via OAuth 2.0 so that users can log in to the system.

3. To implement Google Drive API to provide file sharing function, allow controlled operation on files without saving a copy of files on the server.

4.To develop and implement a role based access control whether different functionally would be held by teachers or learners.

5.To develop the functionality of CRUD operation to manage student data more efficiently into the system.

6.To provide an Excel upload facility for mass student registration, so allowing for blank, form-less registration, and ease of administration.

7. To facilitate a secure and Direct transfer of filesystem with authentication for the privacy and to keep away from the unauthorized system.

5. SCOPE OF SYSTEM

The scope of the project involves the entire scenario of sharing files securely and efficiently between teachers and students in an educational setup. While the primary focus is on sharing between students and teachers in SLRTCE, the system can be extended for use in other educational setup to fulfill similar needs, given that the constraints and specifications are similar. The emphasis is on distributing assignments while maintaining security/authentication/occasional manual intervention.[7]

The teachers also have the ability to login securely through a login based on Google accounts. The login is limited to accounts with the university's domain. Once logged in teachers can administer their course data such as lists of students, subjects, and groups. They can also add, delete, or modify students without having to re input data, and be able to organize students based on subject or group. Teachers will be able to upload

thousands of users using an excel file and eliminate the tedious task of manually uploading each student. They will also be able to create secure, authorization links for students to grant the system access to their files.[7]

On the technical side, the project scope involves creating the backend (implemented by Node.js and Express.js) to create API requests, set the authentication flow, and implement file-sharing logic. Furthermore, MongoDB Atlas is the database managing the user information, the student records, and the metadata of the file transfers. In addition, OAuth 2.0 and Google Drive API are implemented into the system for secured authenticating and authorizing and file upload and sharing respectively. Finally, the frontend is also created with HTML, CSS, and JS.

In general, the project is covering secure, scalable and efficient sharing of files that can be used for academic purpose with desired simplicity and minimum dependency on storage.[1][2]

6.PROCESS OF SYSTEM: FLOWCHART

The system we proposed is a centralized platform for teachers to manage student information and feeding materials to students automatically. The flow can be categorized into four functional modules which are accessible through a main login page.

1. Authentication & Access Control

The entry point of the system utilizes a gatekeeper logic to ensure data security:

Identity Verification: The system initiates with a **Login** check.

2. Direct Database Integration (Batch Import): For large data entry counts, the system proposes an "Insert Student from CSV" flow. This flow enables teachers to avoid 1-1 data entry by uploading pre-structured data files directly into the system's Database. This help the teachers to insert bulk students .

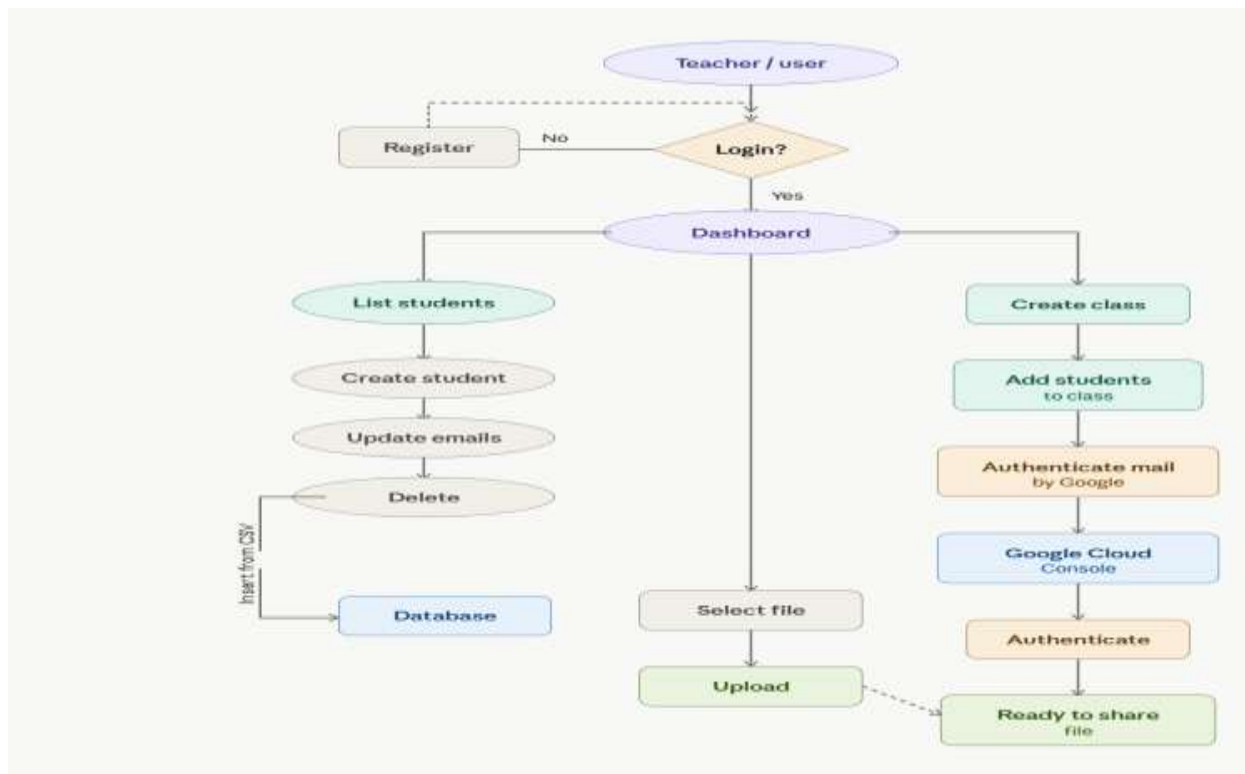


Fig 1

3.Resource Distribution Workflow

The final functional branch handles the actual dissemination of materials:

A. Selection & Staging: Through the "Select File" interface, the educator chooses the assignment or resource from their local or cloud storage.

B. Deployment: The "Upload" action triggers the distribution logic, placing the file into the secure cloud environment defined in the system's backend (as described in your research abstract).

C. Integrated Middleware Logic: The branch showing Dashboard → Google Cloud → Console → Authenticate represents Middleware Integration Theory.[5]

Flowchart (Figure 1) represents the operational flow of the system and has Gatekeeper Security and Module Independence as its first levels. The flow logic is

represented by a flow pattern called Centralized Controller (used in Teacher Dashboard), where in order to reach the 'Ready to Share' state it is necessary to go through an elaborate multi-step validation sequence, including local DB consistency and external OAuth based authentication by Google Cloud Console.[8]

7.SYSTEM OVERVIEW

The proposed system is such as an effective Middleware Integration System where the access is normalized between the institutional student managing and distributing the cloud file. The system is designed such workflow in which the main principal (teacher) is assisted by automation to administer and deliver the education contents safely using the federated identity.[6]

Operational Workflow:

- **Initialization:** The teacher authenticates the group and initializes the class group.
- **Validation:** Student emails are authenticated with the Google Cloud Directory.

- **Staging:** Target assignment file is selected and ingested into temporary staging area by instructor.
- **Execution:** When pressing the 'finalize' button, the system performs a distribution command that automatically applies sharing permissions for each student that registers into the chosen course.

Design Goals:

- **Security:** When using Google's infrastructure, only verified institutional accounts can access files.
- **Efficiency:** An educator cannot enjoy the O() time complexity of disseminating an assignment any more by replacing the time-consuming distribution method of sending a copy to each student by the method of creating and batch sharing a cloud folder.

System Architecture and Components:

- **User Interface & Access Layer:** The entry point is the web dashboard that employs a gateway logic with bi-line authentication requiring institutional id and password. Session states are maintained at this level to provide only credentialed teachers access to student database and file sharing modules.[5]
- **Data Management Layer (Local):** This layer is in charge of saving student records. To avoid too much admin overhead, the system allows Batch Data Ingestion by parsing a CSV file. After splitting out, student metadata (names, institutional emails) is saved in a relational database so to perform fine grain CRUD tasks, as well as to arrange students within specific 'Classes'.[6]
- **Cloud Integration & Distribution Layer:** The key advanced feature of the system is its integration with the Google Cloud Console. By

sharing the Google credentials through OAuth 2.0 and the Google Drive API, the system completes an authorized "handshake" that proves the work is completed by the student. This implemented layer enables the actual copying of the assignment file into students' cloud spaces, rather than students manually attaching the file to an email.[8]

8. INNOVATION IN SOLUTION:

The system proposed describes a new method of academic file distribution by dynamically composing together the industry-standard technologies like OAuth 2.0, Google Drive API and RESTful web services into an uncharacterized workflow directed for learning platforms. Although these technologies are mature and popular, their ensemble in an automated, structured educational file delivery system is a notable contribution.

It create the folders inside your drive and name the folders on the basis of type of file shared i.e if the file shared is assignment it will create a assignment folder and will put the file into that and all other assignment also will go into the same folder if the reason is same no need to create new folder every time.

An important new feature of the system is the automatic organization of circulated items. Files are not only circulated but stored into neatly structured folders in each recipients Google Drive; organized by course and subject. Not only does this provide a standard and workable method of file sharing, but it reduces the impact on the end user by removing the need for the student to file the circulated items themselves.

Another crucial feature is the per-user authorization scheme that adheres to the standards of OAuth 2.0. Each user (student) voluntarily authorizes the system with specific scope for using their Google Drive without disclosing the account name and password. This prevents showing sensitive data contents and mitigate the damages if the theft.

9.SCREENSHOTS

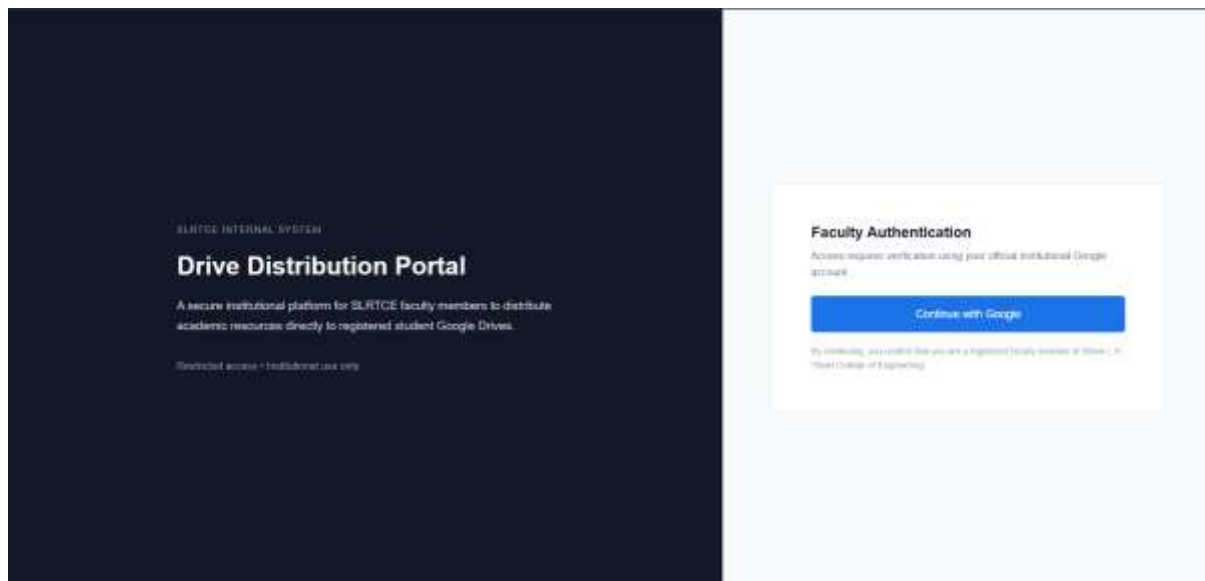


Fig 2: Login interface of the web based system

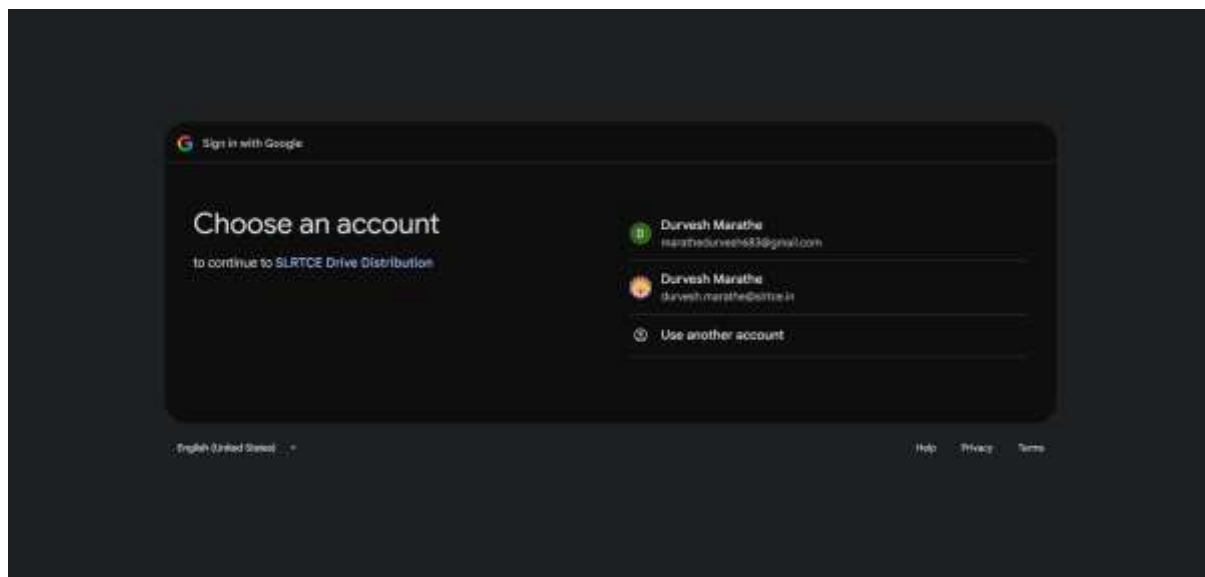
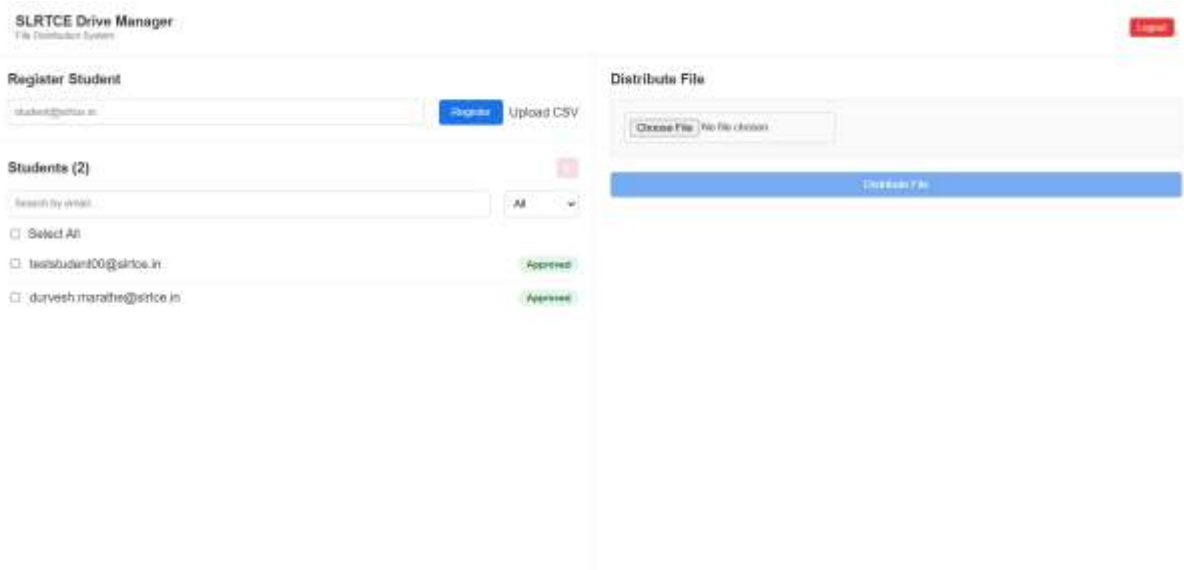


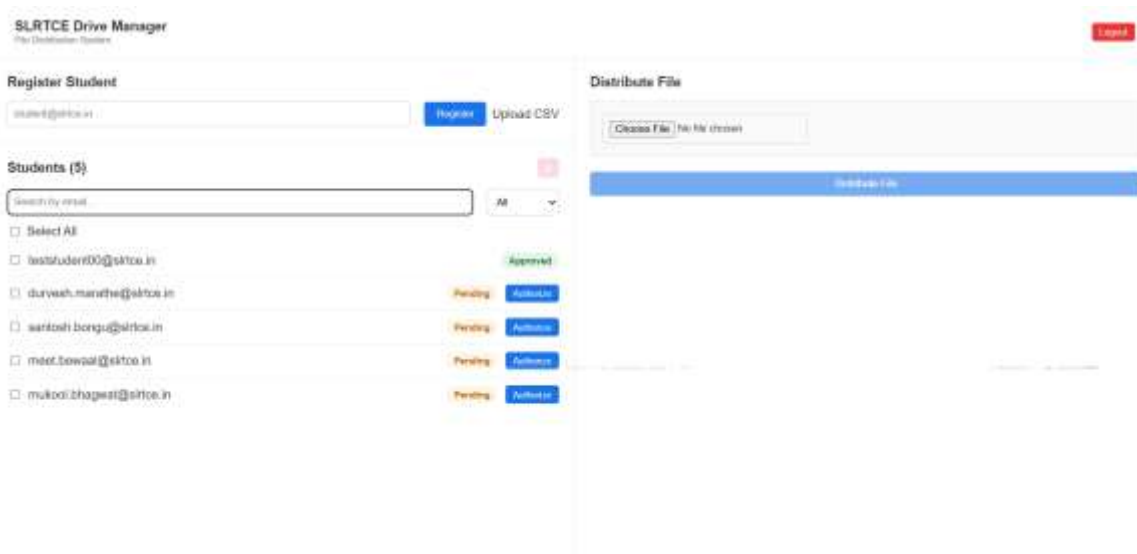
Fig 3: google authentication interface



The dashboard is titled "SLRTCE Drive Manager" with the subtitle "File Distribution System". It features a "Register Student" section on the left with an email input field (containing "student@slrtce.in"), a "Register" button, and an "Upload CSV" link. Below this is a "Students (2)" table with a search bar, a "Select All" checkbox, and two rows of student data. The first row shows "teststudent00@slrtce.in" with an "Approved" status. The second row shows "durvesh.marathe@slrtce.in" with an "Approved" status. On the right, there is a "Distribute File" section with a "Choose File" button (labeled "No file chosen") and a "Distribute File" button. A "Logout" button is located in the top right corner.

Students (2)	
teststudent00@slrtce.in	Approved
durvesh.marathe@slrtce.in	Approved

Fig 4: Successful login to dashboard



This dashboard is similar to the one in Fig 4 but shows a different state. The "Register Student" section remains the same. The "Students (5)" table now contains five rows. The first row, "teststudent00@slrtce.in", is marked as "Approved". The next three rows, "durvesh.marathe@slrtce.in", "santosh.bongu@slrtce.in", and "meet.sowal@slrtce.in", are marked as "Pending" and each has a blue "Approve" button next to it. The final row, "mukesh.bhagwat@slrtce.in", is also marked as "Pending" but does not have an "Approve" button. The "Distribute File" section and "Logout" button are also present.

Students (5)	
teststudent00@slrtce.in	Approved
durvesh.marathe@slrtce.in	Pending Approve
santosh.bongu@slrtce.in	Pending Approve
meet.sowal@slrtce.in	Pending Approve
mukesh.bhagwat@slrtce.in	Pending

Fig 5: Student Email registration [student adding interface]

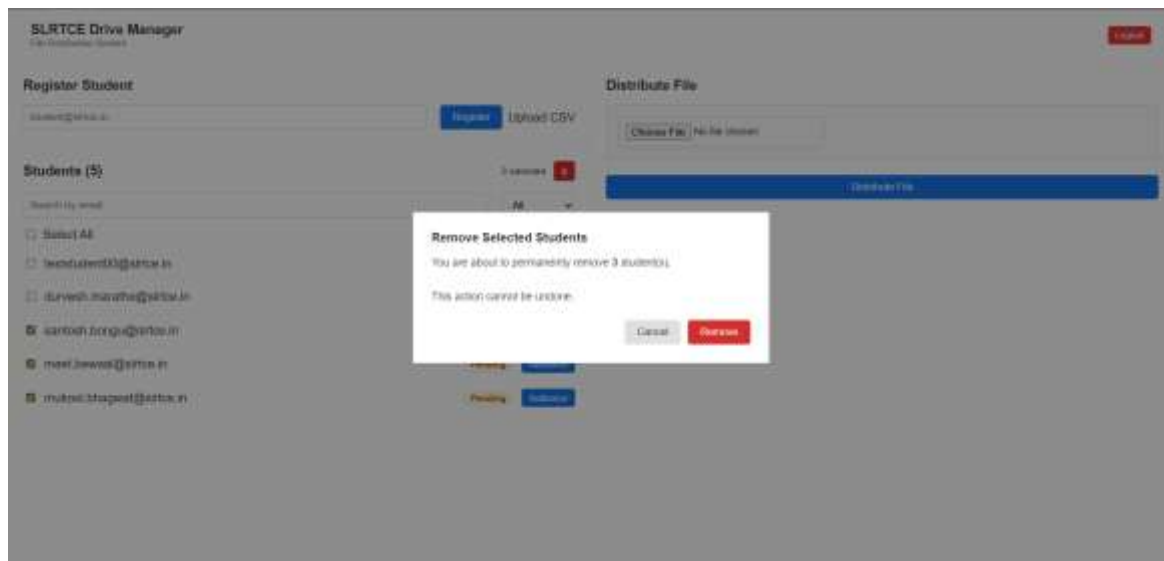


Fig 6:Error prevention functionality for undoable action

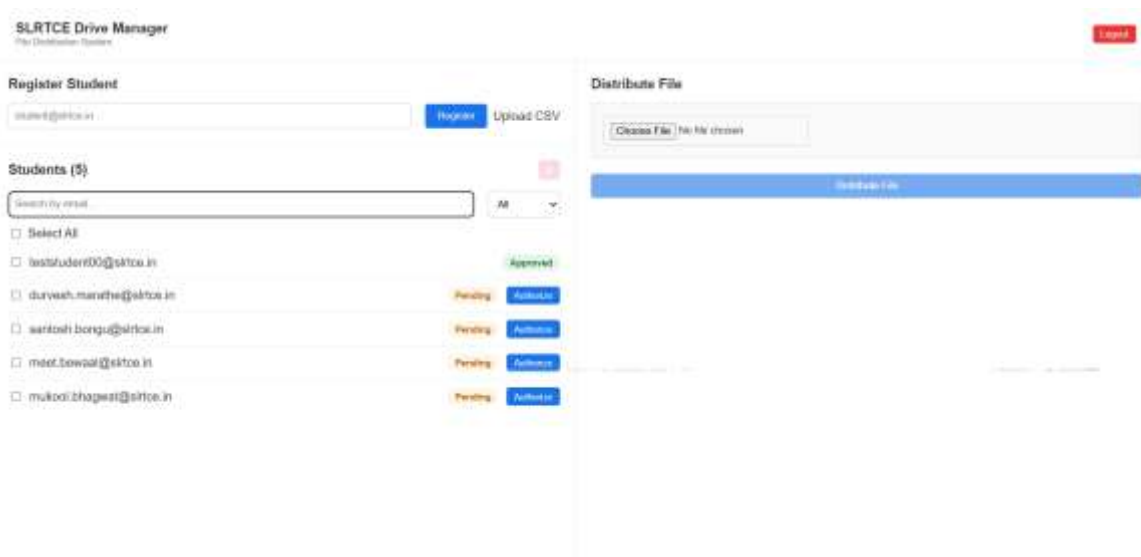


Fig 7: Dashboard of student with list of authorized and Unauthorized

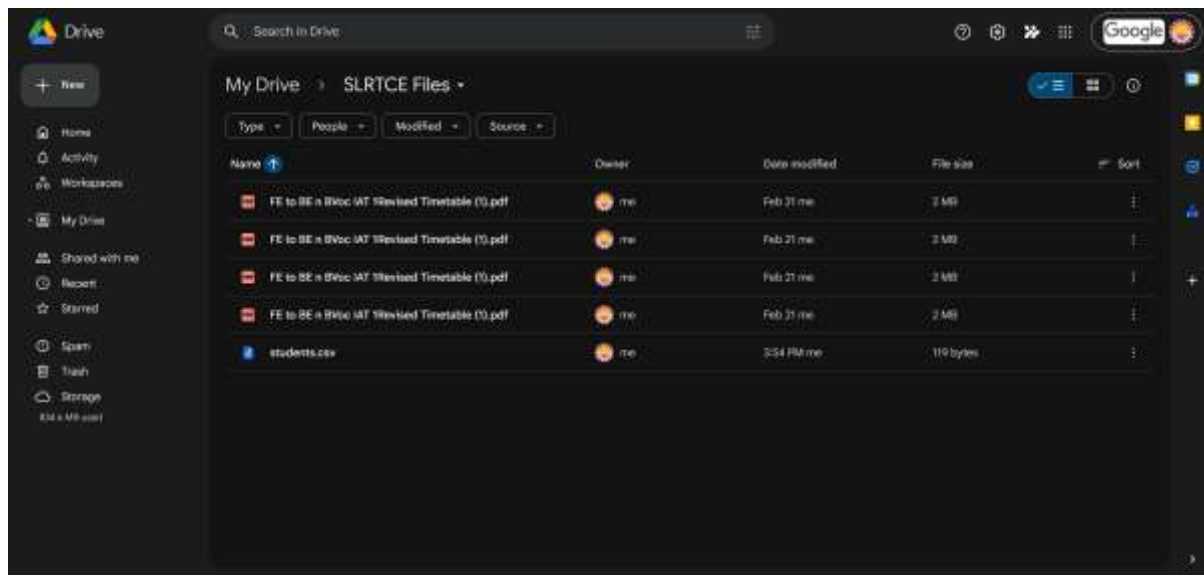


Fig 8: file transferred to student drive account

10. CONCLUSION

The suggested web-based file transfer system offers a robust and secure solution to the problems of distributing files in an academic setting. With the use of standard technologies like OAuth 2.0, Google Drive API and REST, the application does not require any centralized storage and remains an accessible means of file transfer for teachers and students.[2]

According to the system, the file delivery process is also the one of the system's strengths that. Teachers could upload and broadcast assignment files to students with ease and efficiency via a very few steps, it takes less time from teachers to deliver the documents to students, they don't need to handle the file management. On the other side, students has the advantage to receive all of their files in Google drive automatically, no need to log in repeatedly, and handle the task every hand to execute the file management and access.[5]

The system also has a good compliance with security and privacy guidelines: the use of OAuth based authentication provides constrains so that only authorized users can use the system and run operations. Also, the per-user authorization allows students to control access to their own Google Drive account, keeping privacy high while allowing automatically delivered files. Moreover, the system does not need to

store any files on own server, not risking a data breach on server side and keeping infrastructure overhead low.[5]

The system also provides support for administrative ease. This is in two ways bulk student registration using excel files in addition to the CRUD based student management. These features cater well to the practical issues faced by teachers and administrators, simplifying the lives of student management in large classrooms. The modular architecture enables a plan for scalable system extensions including analytics, role expansion and education services.[7]

11.FUTURE SCOPE

Although the system used is efficient for academic file sharing; there are still some improvements that can be added into the system in future in order to improve the usability, functionality and scalability of system, the future enhancement areas are:

A. Real-Time Notifications and Alerts:

In addition, we are able to improve this system by sending notification in live, such as email

notification or push notification, to inform the students that their new file are uploaded.

B. Multi-Institution Support (SaaS Model):

While being built for one institution, it could be used for multiple institutions if built on a multi-tenant architecture. Using multi-tenancy approach also provides a solution that is scalable and could be delivered as SaaS Platform.

C. Integration with Learning Management Systems (LMS):

The subsystems can be funneled into traditional LMS systems, calendaring applications, communication systems, or other online systems to form a complete educational environment. This would insure that LMS platform, file sharing, and everything else connects to promote an entire educational network.

D. Security Enhancements:

Future versions can include additional security measures such as multi-factor authentication (MFA), advanced token management, and audit logs to further strengthen data protection and system integrity.

In conclusion:

Future scope of the system is to extend the features provided by the system from simple file sharing utility to an entire academic platform. Improvements to the system in terms of scalability, ease of use and features that will be added will make the system more scalable, versatile and robust.[7]

Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICECCOT), 2017.

[5] B. Jose and S. Abraham, "Exploring the Merits of NoSQL: A Study Based on MongoDB," in *Proc. IEEE International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2017.

[6] Google Developers, "Google Drive API Documentation," 2020. [Online]

[7] M. Jones and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," IETF RFC 6750, Oct. 2012.

[8]] Google Developers, "Using OAuth 2.0 for Web Server Applications," 2020.

12. REFERENCES

[1] Z. Li et al., "On Developing and Deploying Large-File Upload Services of Personal Cloud Storage," in *Proc. IEEE International Conference on Cloud Computing (CLOUD)*, 2015. DOI: 10.1109/7207376.

[2] Authors et al., "Online Document Management System for Academic Institutes," in *Proc. IEEE Conference*, 2010. DOI: 10.1109/5694910.

[3] J. Singh and N. K. Chaudhary, "OAuth 2.0: Architectural Design Augmentation for Mitigation of Common Security Vulnerabilities," *Journal of Information Security and Applications*

[4] S. Binu, S. Misbahuddin, and P. Raj, "Secure Cloud Storage and File Sharing," in *Proc. IEEE International*