ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG122

Improved Integrated Model for Data Storage in the Cloud

Henry Izundu Chimezie1: Dr. Martha Ozohu Musa: Dr F. Egbono
Department of cybersecurity, Faculty of Computing,
University of Port Harcourt, Nigeria

Abstract:- The software model showcases the effective movement of files across storage units in a cloud environment. It includes features such as user identification, registration, and file upload capabilities. The experimental findings demonstrate an efficient procedure that has been confirmed through user interfaces for logging in, registering, and uploading files. The admin dashboard streamlines user administration and file transfer processes, utilising a time-efficient data migration technique. Comparative analysis reveals a substantial enhancement in data migration throughput, as the suggested method reduces migration time by an average of 50% in comparison to existing systems. The results validate the system's improved efficiency in managing file transfers and user administration in a cloud-based environment.

***Keywords:-** Data Storage, Cloud Computing, File Management, Migration Technique*

I. INTRODUCTION

Cloud computing is a technique that focuses on sharing resources in order to achieve high availability and scalability. It is similar to offering utilities across a network (Gagandeep et al., 2017). Cloud providers have noted the emergence of new paradigms such as mobile edge computing, mobile cloud computing, and fog computing in response to the evolving requirements of cloud computing (Abid et al., 2022). Furthermore, much academic research has been conducted to analyse the impact of scalability and resource management on the overall performance of cloud applications (Ganesh et al., 2016; Ullah et al., 2022). iCloud storage is a service that involves the remote maintenance, management, and backup of data, which is then accessible to users over a network, usually the Internet.

Cloud storage is a dynamic and evolving technology that continuously improves its features and interface (Hashem et al., 2020). Nevertheless, the integration of cloud computing has emerged as a crucial method to establish connections across these diverse solutions. Cloud integration refers to the process of creating connections and enabling the seamless access and sharing of data between various applications inside a cloud environment (Malik & Khan, 2022). iPaaS, or Integration Platform as a Service, refers to the Cloud-based integration. Although SaaS applications emphasise their advantages, such as enhanced flexibility and reduced expenses, corporate executives have acknowledged that integration is a significant challenge when it comes to effectively adopting and implementing SaaS and other web services. Cloud computing refers to the extensive integration and advancement of conventional computer and network technologies. This includes distributed computing, parallel computing, grid computing, virtualization, and load balancing, among other aspects.

Modern cloud computing sometimes employs a technique called "distributed processing," where a complex programme is broken into smaller, independent tasks (Buyya et al., 2023). Afterwards, these workloads are distributed over a network of interconnected computers inside the cloud architecture, leveraging their combined computational capabilities to accelerate processing. Once all actions have been completed, the individual results are merged and then presented back to the user. The advancement of technology and the widespread availability of the Internet have resulted in an overwhelming amount of information, propelling us into an era when information is at the centre of everything.

According to research conducted by the International Data Corporation (IDC), the total volume of data increased from 33 zettabytes (ZB) in 2018 to 64 zettabytes (ZB) in 2020, representing a remarkable growth of almost 100% in a span of two years (IDC, 2020). The current trend indicates that there is no indication of it slowing down. Estimates suggest that the global data volume could reach an astonishing 222 ZB by 2025, according to IDC in 2023. The principle of conservation does not apply to information. In the Web2.0 era, the emerging Internet service model encourages users to transition from being passive consumers of information to active producers of information. The numerous social networking platforms and forums generate a vast amount of data, akin to the abundance of snowflakes in a blizzard. This data primarily consists of unstructured information. Simultaneously, with the increasing progress of industrial information and commercial operations, a multitude of data, including e-bills, financial transactions, satellite detection data, and more, are being replicated or generated on a daily basis worldwide.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG122

II. LITERATURE REVIEW

Banavath et al. (2023) introduced a method called the "Hybridised Optimisation for Virtual Machine Migration and Task Scheduling in Cloud Data Centres" or FHCS methodology. This strategy effectively minimises energy usage and mitigates resource depletion in cloud data centres. The authors conducted a comparison between FHCS and three other optimisation algorithms: Ant Colony System (ACS), Particle Swarm Optimisation (PSO), and Genetic Algorithm (GA). Their simulation findings showed that FHCS and the VM migration approach performed better than existing processes such as GA, PSO, and ACS in terms of energy usage and resource utilisation.

Srinivasa et al. (2020) proposed a load balancing technique in cloud computing called "Improved Hybrid Algorithm Approach Based Load Balancing Technique." The technique utilises the RTEA hybrid algorithm. This method demonstrated exceptional performance in the crucial metrics of makespan, waiting time, and burst time when compared to conventional procedures.

Zhao et al. (2020) created a new task scheduling scheme in a cloud computing environment called "Hybrid Biogeography-Based Optimisation." This study combines the biogeography-based optimisation (BBO) technique with a novel hybrid migrating BBO (HMBBO) algorithm. The HMBBO algorithm incorporates the migration strategy with particle swarm optimisation (PSO) to improve the efficiency of job scheduling.

Antonio et al. (2020) investigated the advancement of "Hybrid Multi-Cloud Storage Systems," which allow for the storage of files in a multi-cloud environment comprising private, public, and hybrid clouds. An abstract storage layer was established across several cloud storage providers using the RRNS algorithm. The outcome of this design is a Hybrid Multi-Cloud Storage (MCS) module that divides files into p+r fragments and disperses them among many providers to guarantee the accessibility and privacy of the data.

Ivana and her colleagues (2020) tackled the issue of resource scheduling in cloud computing by utilising a hybridised whale optimisation method. This metaheuristic based on swarm intelligence was modified to efficiently address resource scheduling difficulties in cloud systems.

In their 2021 study, Cai et al. introduced a "Data Storage and Management Scheme in Cloud Storage Model" that relies on the Cloud Storage Access Protocol developed by the Storage Networking Industry Association (SNIA). This paradigm implemented the idea of storage pooling at the storage management layer, which included the integration of virtualization and the ability to dynamically manage physical storage devices. Nevertheless, the model does not provide conclusive evidence of storage integration for the purpose of uploading and downloading data in multi-cloud systems.

Samson et al. (2020) introduced a "Data Model for Cloud Computing Environment" that facilitates the allocation, control, and administration of virtual systems. The model is deployed on a private, efficient cloud network and it effectively combines and oversees cloud resources in diverse contexts using graph and document-oriented databases. The empirical findings demonstrated that the graph model outperformed the document-oriented model in terms of query execution time.

In their study, Niu et al. (2018) did an extensive investigation on "Hybrid Storage Systems," examining several structures and algorithms such as caching, scheduling, and resource allocation. They provided a thorough categorization and explanation of these components.

In their study, Ellison et al. (2018) assessed different algorithms and workload models to examine the available choices for migrating cloud databases. They identified several shortcomings in the existing approaches, including the absence of random value testing, limited support for relational migrations, and the necessity of automated cost models.

In their study, Ravindranadh et al. (2018) introduced a honey-encryption cryptographic technique designed specifically for securely transferring data in cloud computing. This algorithm guarantees both the integrity and secrecy of the data by utilising NetBeans and MySQL in virtual settings.

Ma et al. (2018) enhanced the efficiency of transferring live data in the cloud by optimising stream-based migration strategies. They achieved this by employing particle swarm optimisation (PSO) and a non-linear model that considers migration cost and balance. They recognised the necessity of enhancing the categorization of PSO and the iterative algorithm.

III. METHODOLOGY

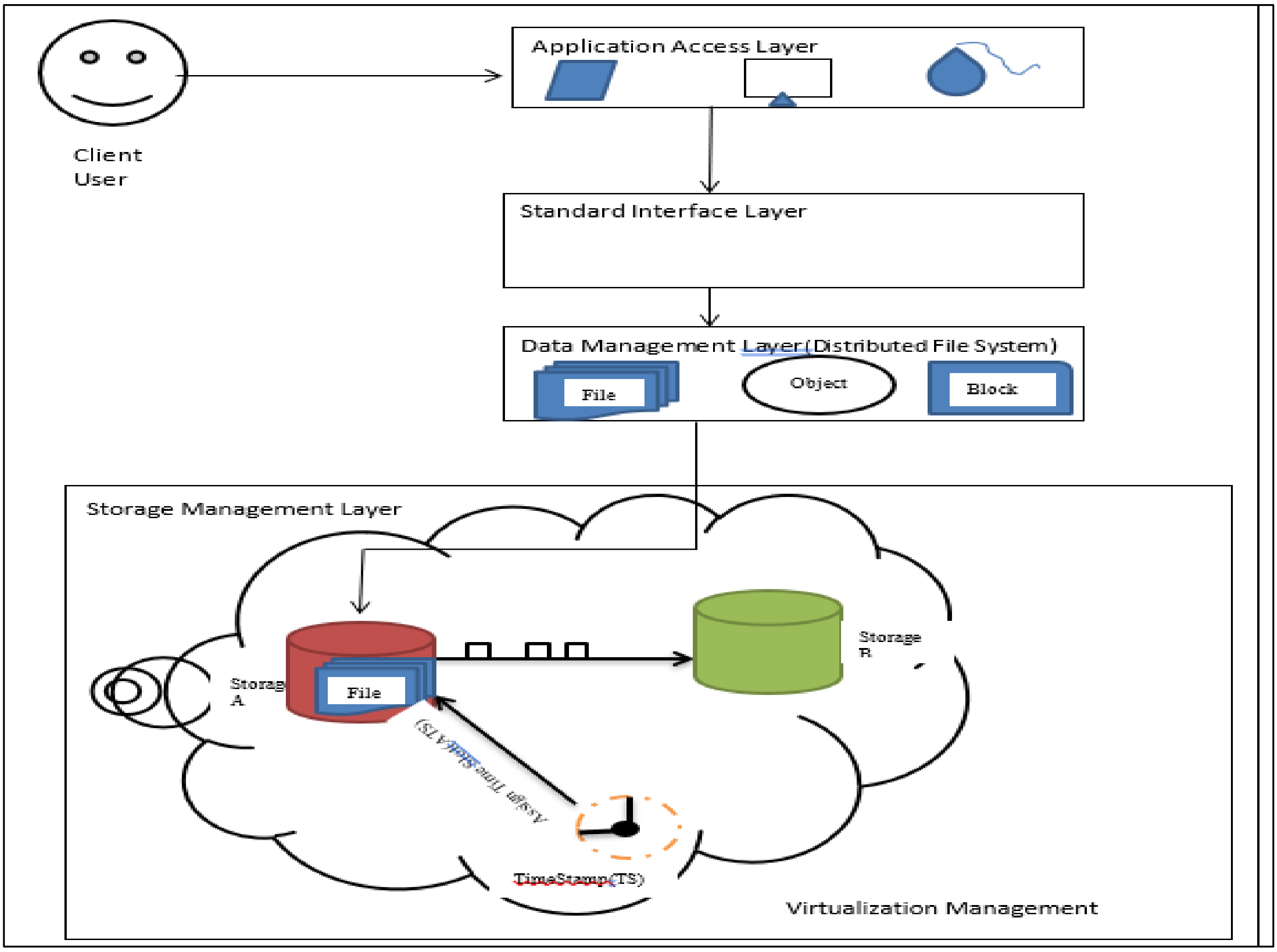

Fig 1: Architecture of the Proposed System

➢ *The Architecture of the Proposed System has About Five (5) Modules or Layers.*

- **Application Access Layer:** This layer is responsible for developing various applications that provide users with extensive cloud storage options.
- **Standard Interface Layer:** The Deeper Integration Layer is a comprehensive amalgamation of the three aforementioned levels, offering a cohesive interface for the higher-level applications. Examples of interfaces include those based on FTP (File Transfer Protocol), HTTP (Hyper Text Transport Protocol), SOAP (Simple Object Access Protocol), and the protocol used for authorization verification. Cloud storage service providers have the ability to create several interfaces based on their own requirements.
- **Data Management Layer:** Based on distributed storage technologies, data storage on this tier can be categorised into block storage, file storage, object-based storage, and table-based storage. By implementing appropriate file systems on different storage types, we may effectively accommodate structured, semi-structured, and unstructured data while enhancing efficiency.
- **Storage Management Layer:** The Storage Management Layer is an abstraction of storage devices that simplifies the underlying storage structure by consolidating all storage spaces into a storage resource pool. It also enables the mapping of physical devices to a logical view, a process known as virtualization. At this layer, it enables redundancy management, device state monitoring, device malfunction maintenance, and other operations.
- **Timestamp Algorithm:** The Storage Management Layer is an abstraction of storage devices that simplifies the underlying storage structure by consolidating all storage spaces into a storage resource pool. It also enables the mapping of physical devices to a logical view, a process known as virtualization. At this layer, it enables redundancy management, device state monitoring, device malfunction maintenance, and other operations.

IV. RESULTS

The application was developed, implemented, and tested using multiple software packages and tools to accomplish the stated purposes and objectives of the thesis. The main software utilised comprises Visual Studio Integrated Development Environment (IDE) version 2010 (32-bit) and Microsoft SQL Server 2008.

A. Visual Studio 2010 IDE

Visual Studio 2010 offers a collection of visual tools that are advantageous for debugging and observing several threads running at the same time. Multi-targeting is supported, which improves precision in development. The IDE has a powerful debugging tool that enables programmers to analyse their programmes step by step during execution, making it an excellent fit for our project.

B. Visual C#

C# is a highly flexible programming language that incorporates various programming disciplines, including strong typing, imperative, declarative, functional, generic, object-oriented, and component-oriented programming. The language is event-driven, meaning that programmes are created using an integrated development environment (IDE). This integrated development environment (IDE) provides a handy platform for creating, executing, testing, and debugging C# programmes. As a result, it significantly reduces the time needed to develop a functional programme. The extensive language compatibility of C# makes it the most appropriate choice for creating this system.

➢ *Microsoft SQL Server 2008*

Microsoft SQL Server 2008 provides additional features and other improvements in speed and usability as compared to earlier versions. These characteristics promote efficiency, boost developer efficiency, and improve the overall user experience, making it the best option for the system's database management requirements.

Fig 2: Opening Interface of Software

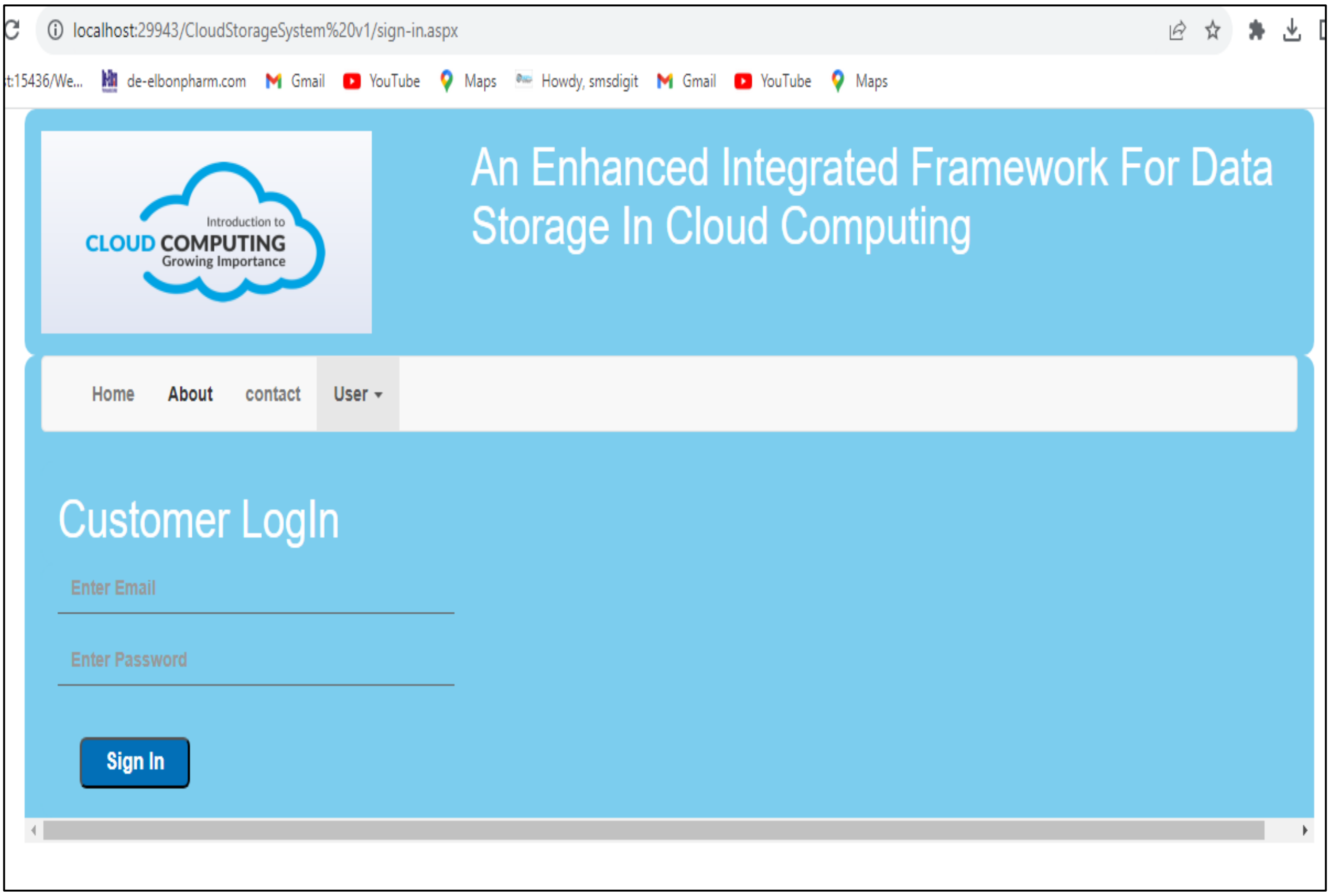

Fig 3: Login Interface of the System

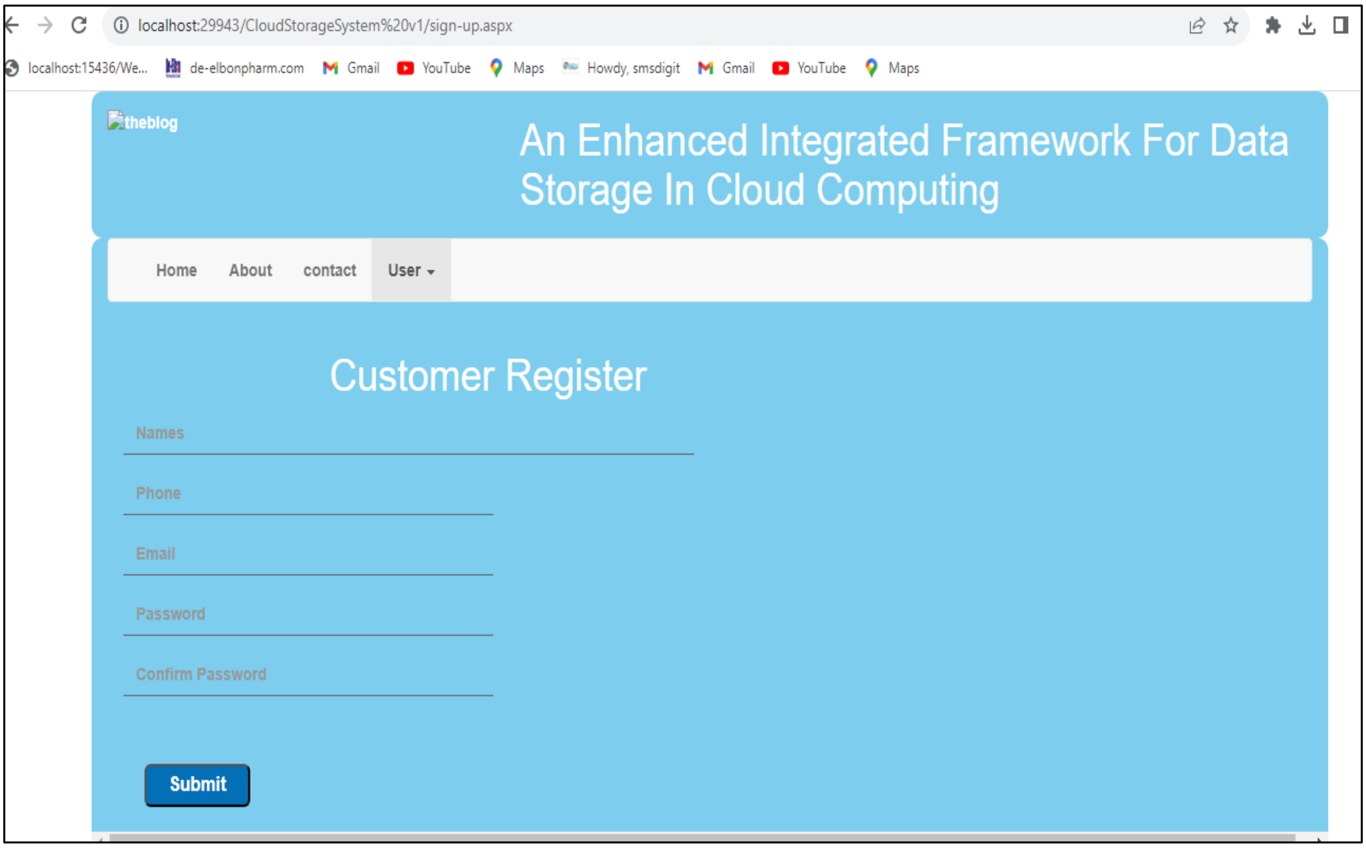

Fig 4: Registration Interface

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG122

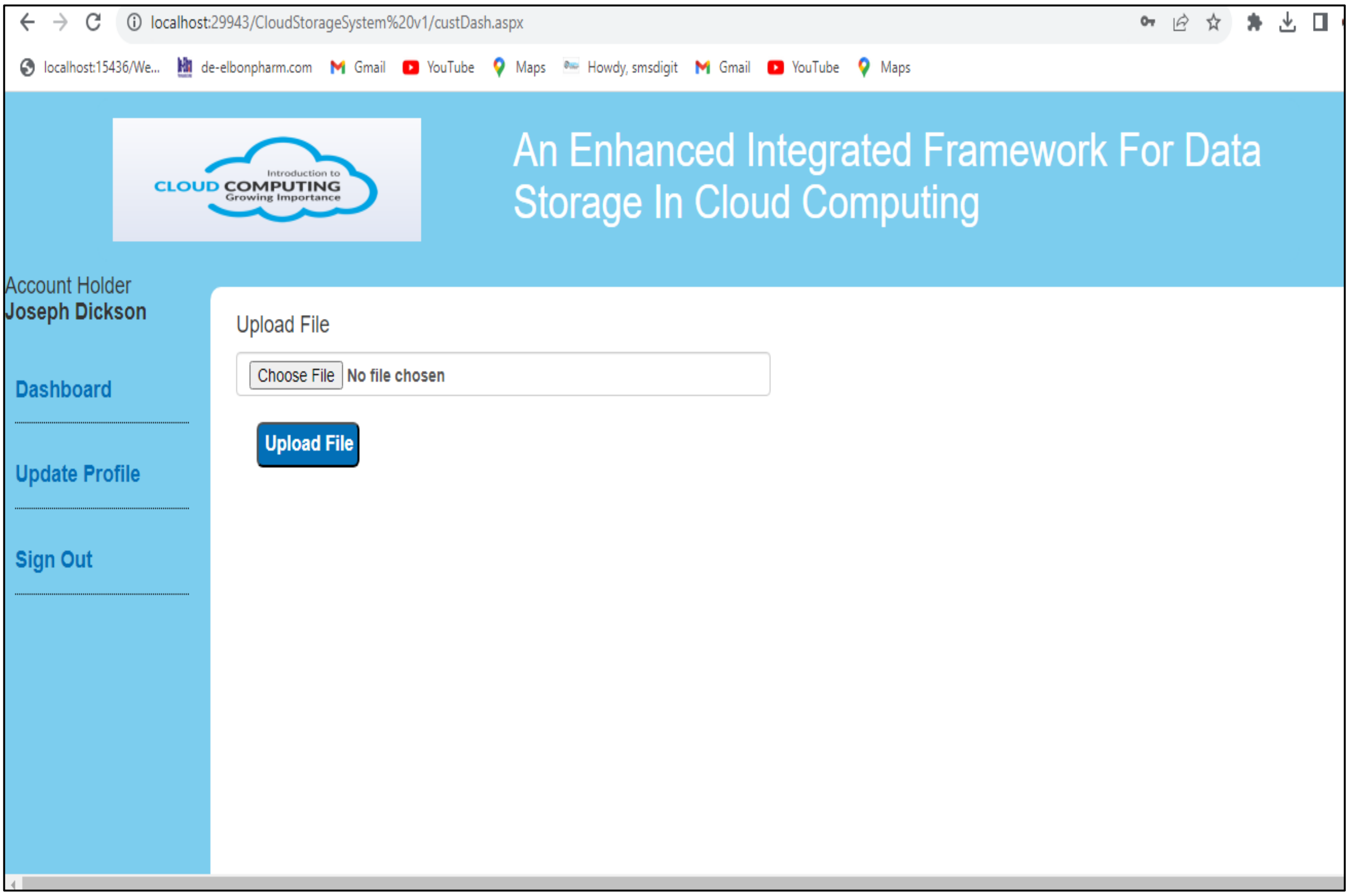

Fig 5: Upload New File Page

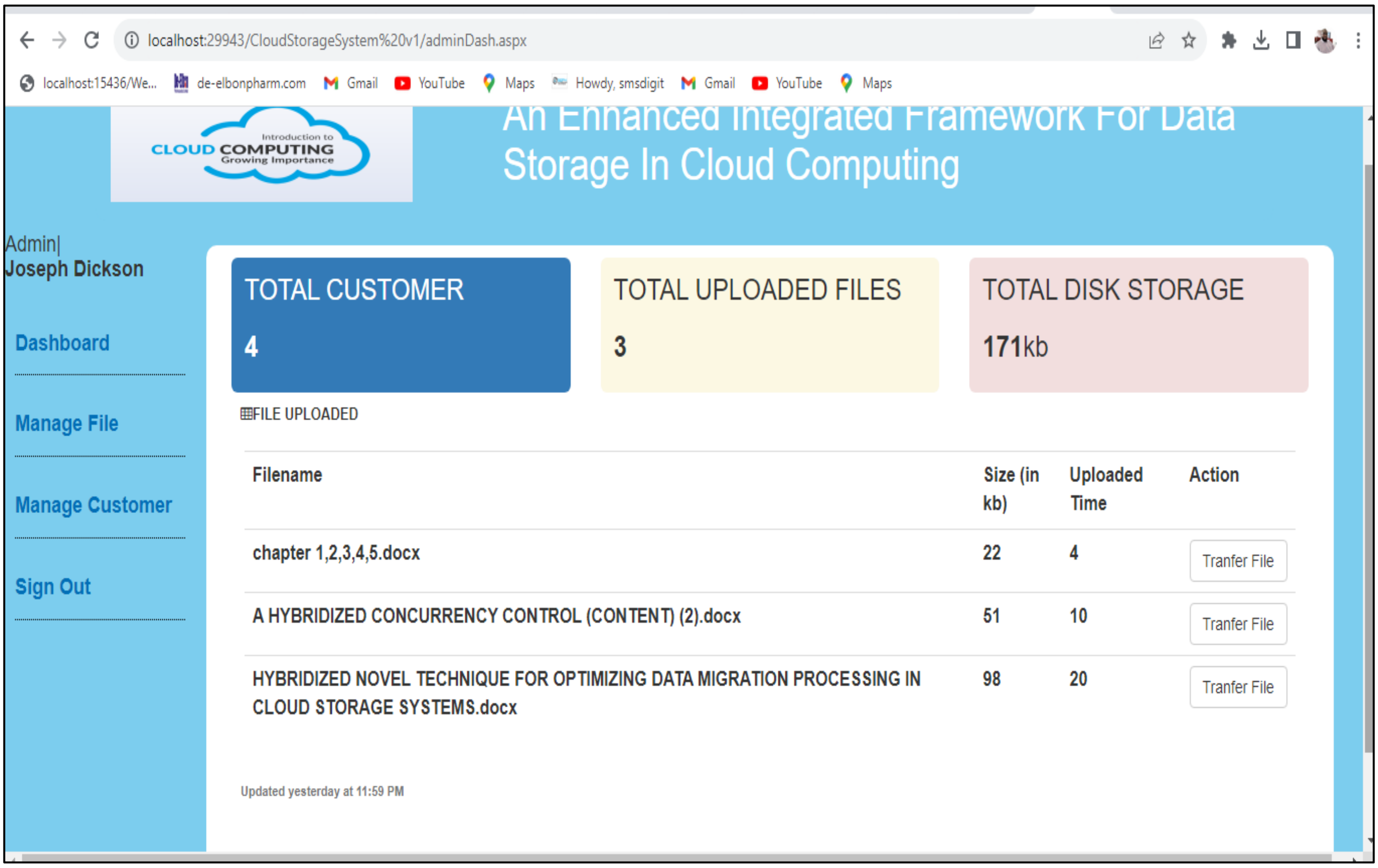

Fig 6: Admin Dashboard

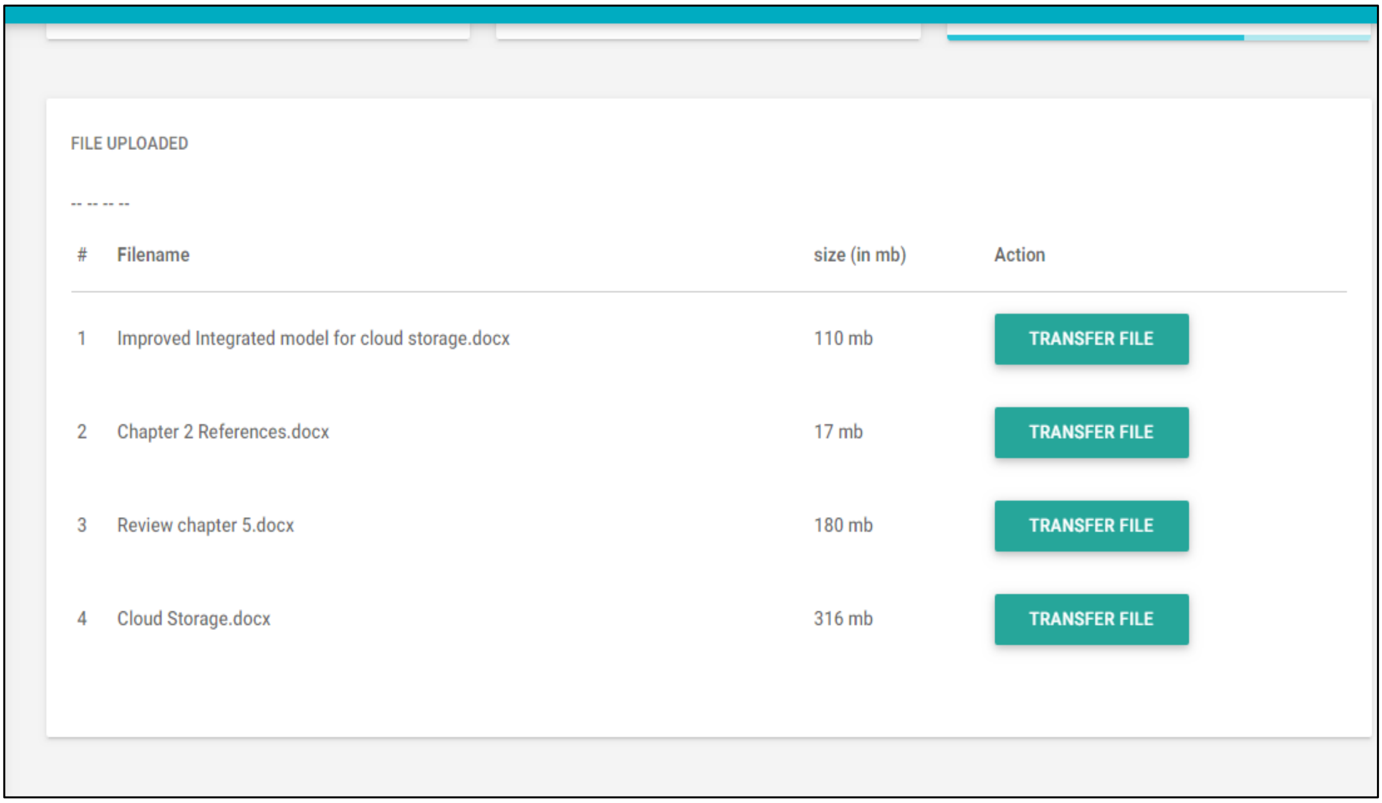

Fig 7: Transfer File Page

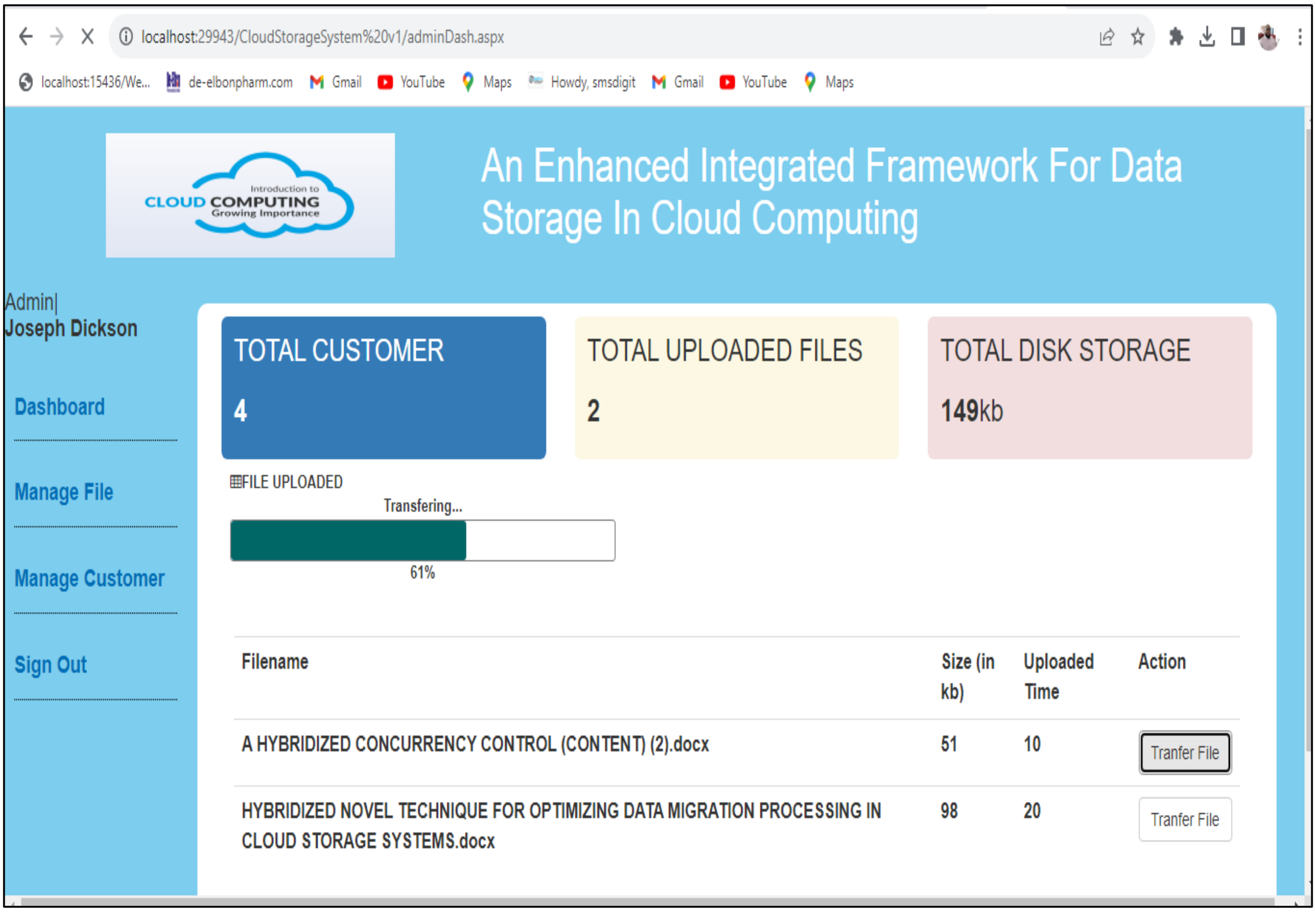

Fig 8: Transfer File Process from Storage A to Storage

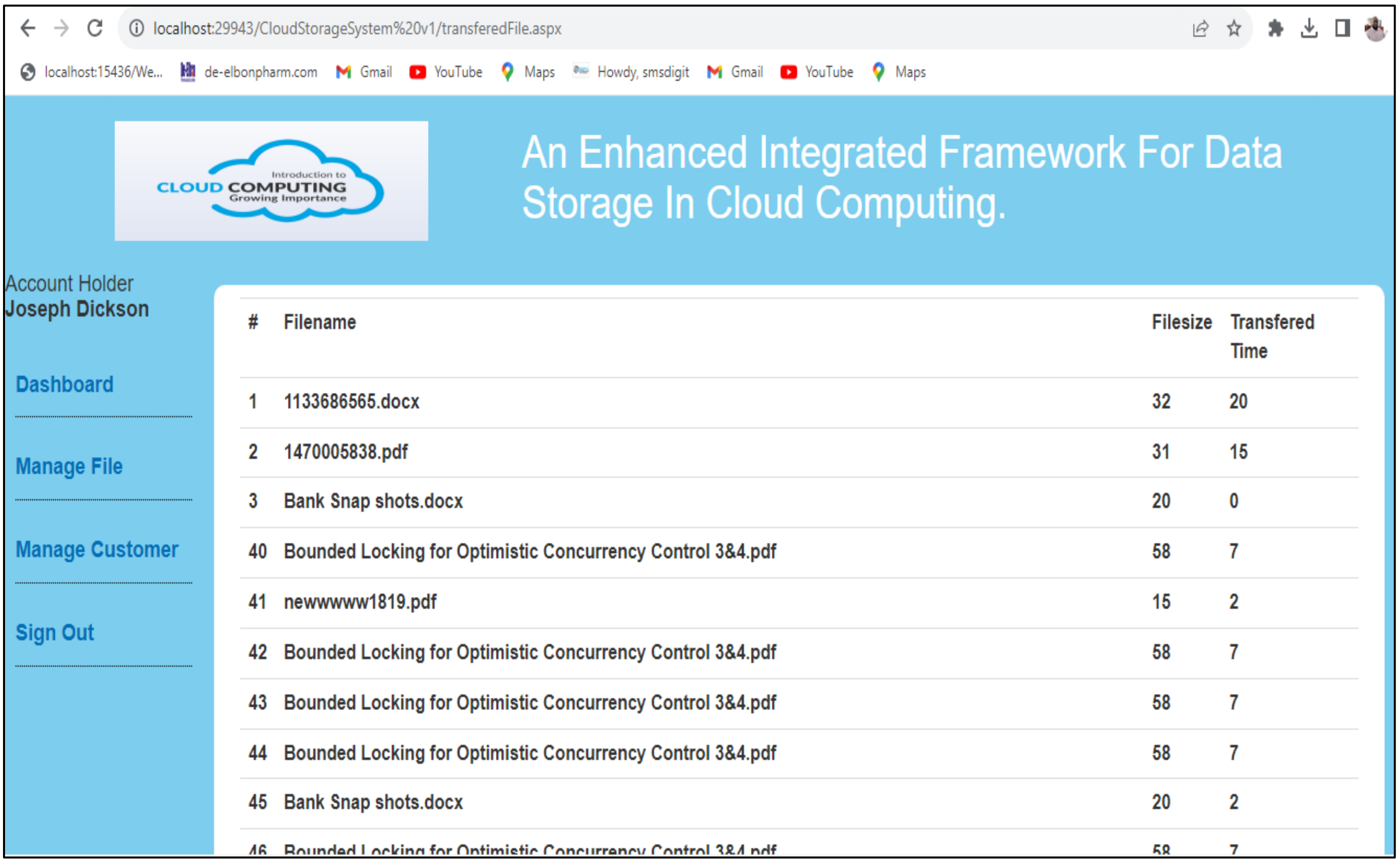

Fig 9: Completion of Transfer File Process

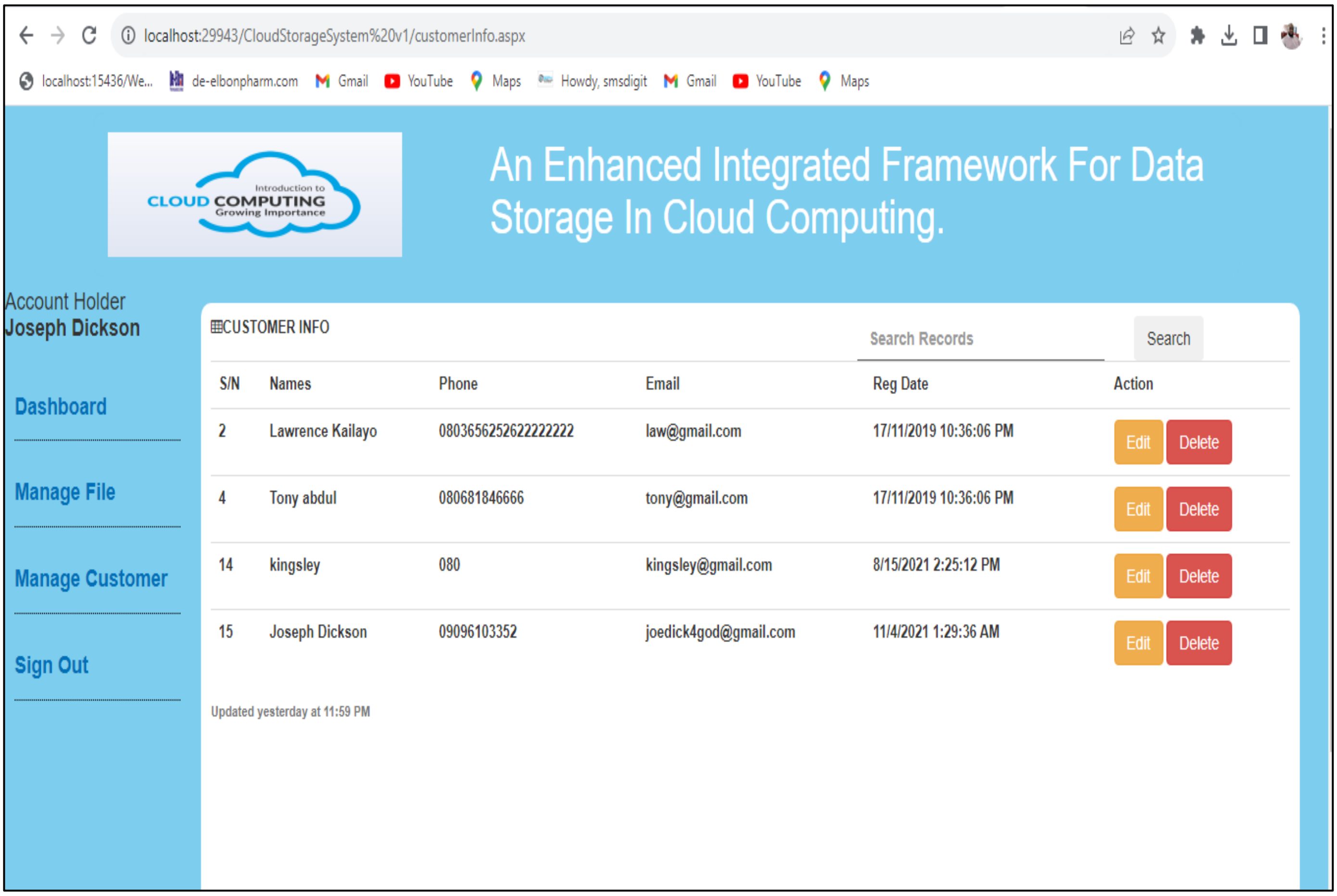

Fig 10: Manage Customer

C. Comparison with the Existing System

Table 1: File Transfer Time taken for Existing System

S/n	File Description	Size	Transfer Time Taken(s)
1	Improved Integrated model for cloud storage.docx	110mb	40s
2	Chapter 2 References.docx	17mb	10s
3	Review chapter 5.docx	180mb	70s
4	Cloud Storage.docx	316mb	90s

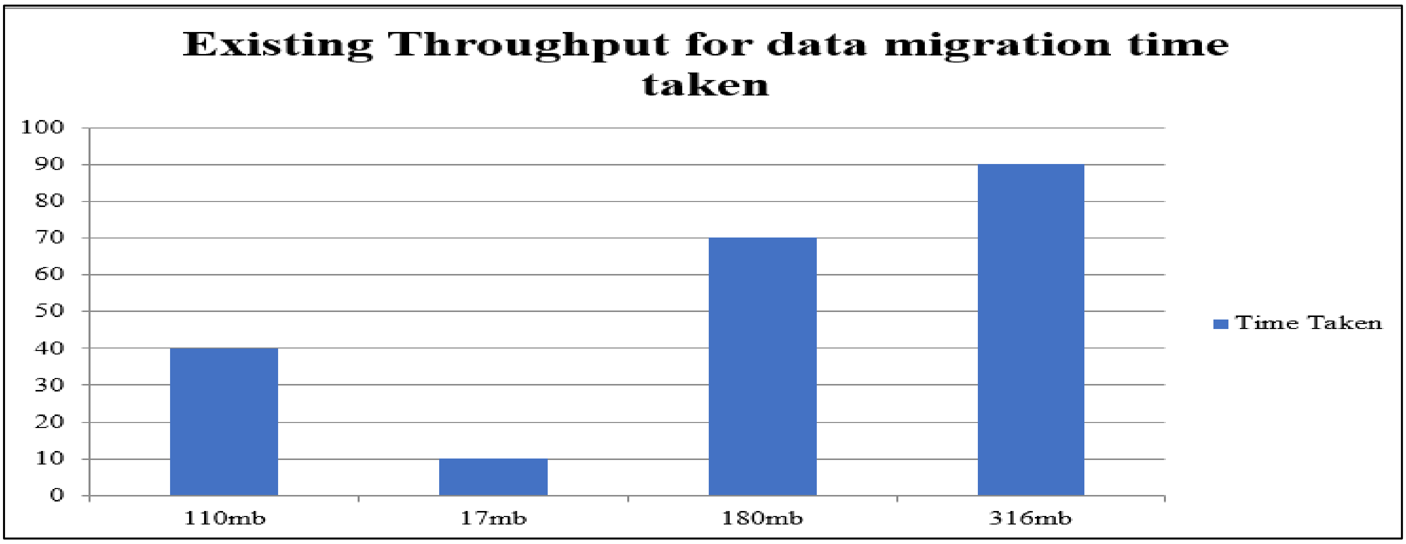

Fig 11: Throughput for Data Migration Time Taken

Table 2: File Transfer Time Taken for Proposed System

S/n	File Description	Size	Transfer Time Taken(s)
1	Improved Integrated model for cloud storage.docx	110mb	20s
2	Chapter 2 References.docx	17mb	5s
3	Review chapter 5.docx	180mb	35s
4	Cloud Storage.docx	316mb	40s

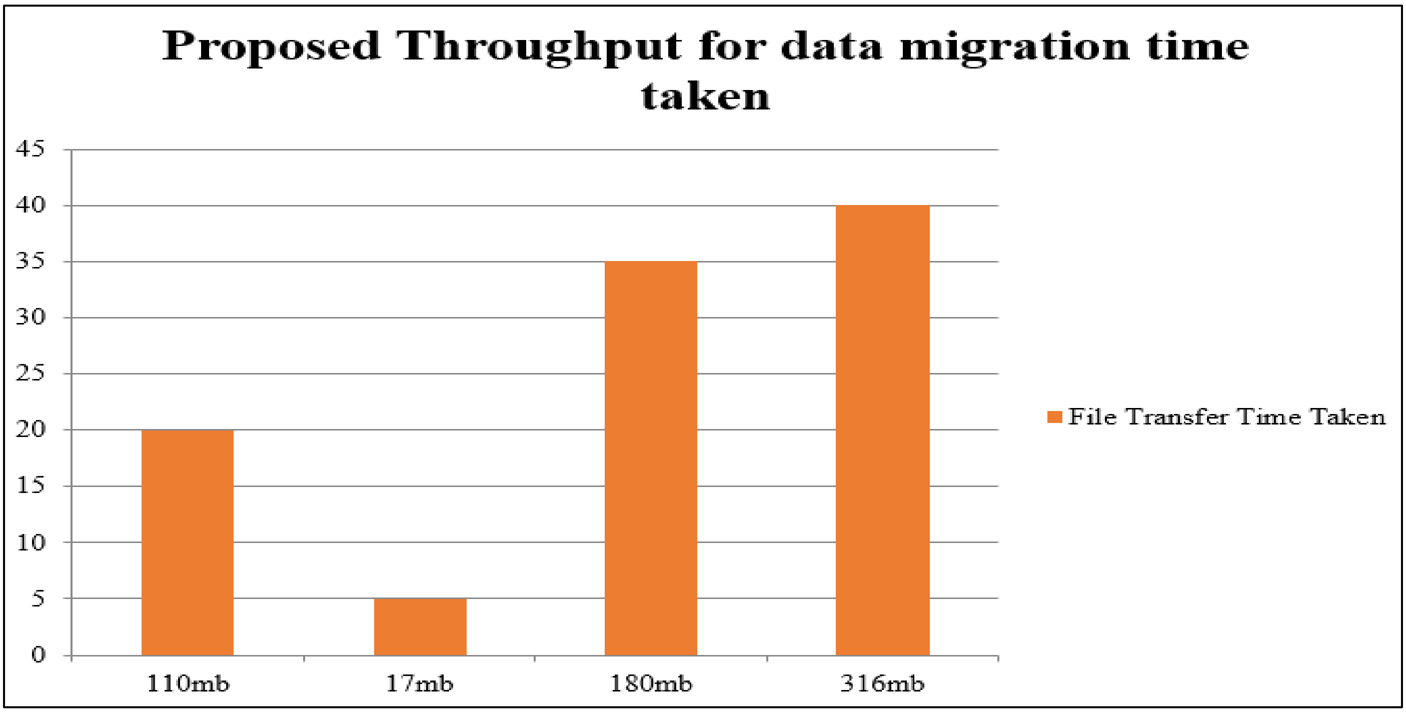

Fig 12: Throughput for Data Migration Time Taken

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24AUG122

Table 3: Comparison File Transfer Time taken

S/n	File Description	Size	Existing Transfer Time Taken(s)	Proposed Transfer Time Taken(s)
1	Improved Integrated model for cloud storage.docx	110mb	40s	20s
2	Chapter 2 References.docx	17mb	10s	5s
3	Review chapter 5.docx	180mb	70s	35s
4	Cloud Storage.docx	316mb	90s	40s

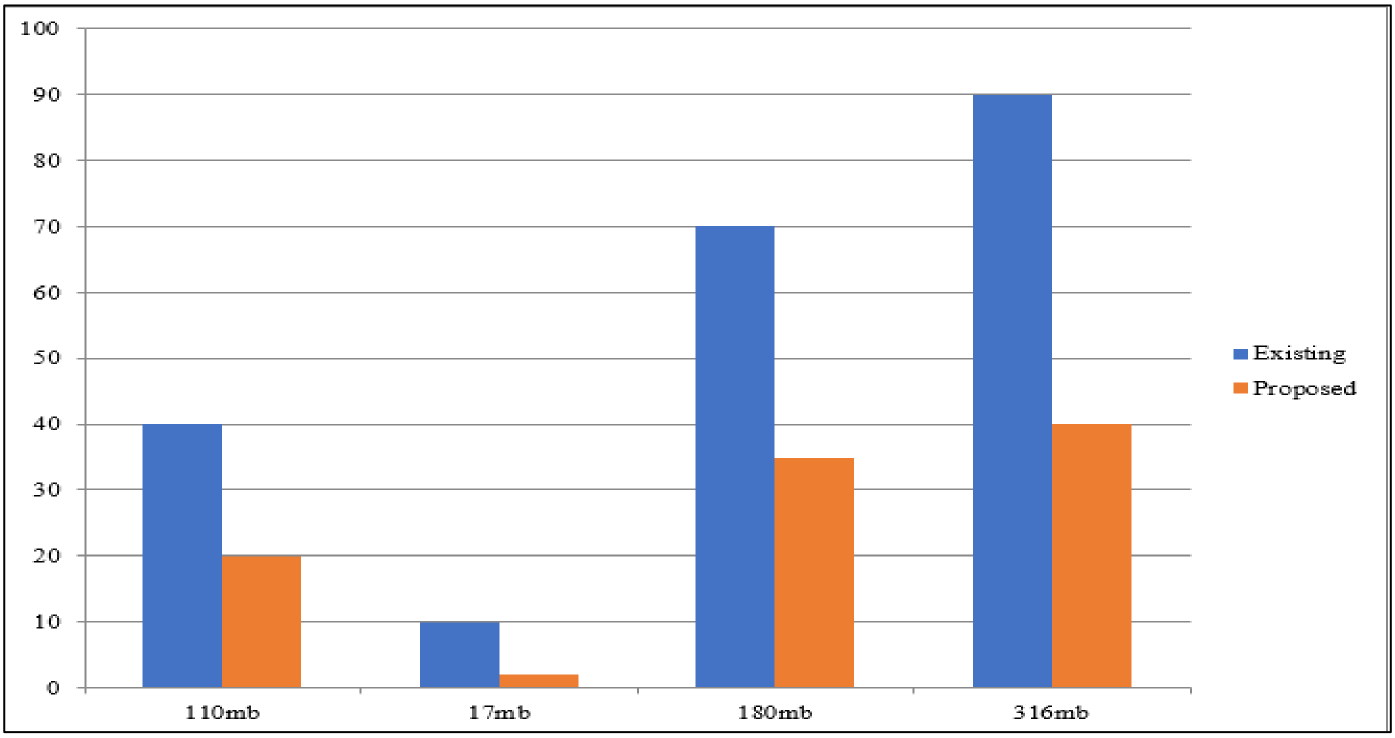

Fig 13: Compare throughputs for data migration time taken

V. DISCUSSION OF RESULTS

The software developed from our proposed' model is shown in Figure 2. Through experimentation with input data, outputs were generated on program running. From the experimental setup of the program interface, file transfer from storage A to storage B with a default time assign to each process. Figure 3 shows a user login form that can be used to authenticate users who wants to upload or save their file in the cloud using the system. Upon filling the form, and the Login button clicked, the form fields are validated and checked for correct data types and complete fields before being allow proceeding uploading the file.

Figure 4 shows a user registration form that can be used to register users who wants to upload or save their file in the cloud using the system. Upon filling the form and the Register button clicked, the form fields are validated and checked for correct data types and complete fields are save in the database. Figure 5 shows the Uploading form for users to upload their file to the cloud for storage. Figure 6 shows the Admin Dashboard page where he manages all the users, files are transfer from one storage to the other etc. Figure 7 shows the File Transfer Page. On this page, the admin clicks on the File Transfer button when he wants to transfer file from Storage A to Storage B then by default a time is automatically assigned to the file for data migration. Once the File Transfer button is clicked, it will display the next page which is the File Transfer Process page. Figure 8 shows' output of the data files transferring process. It contains the number of files transferring from Storage A to Storage B with a time assigned to the process. Figure 9', shows the completed file transfer process result of the data migration which about 5 files were successfully transferred with different time been assigned to them. Figure 10 shows the manage customer page where all the users (customer) are managed. That is to view, edit and delete user account from the system.

Table 1 shows the File Transfer Time taken for Existing System

The table has the following columns; serial number (SN), File Description, File Size and Migration Time Taken which is measured in seconds. Considering 4 different file description (name), File size and time taken for migration process. File 1 has a file size of 110MB and it took 40s to migrate from Storage A to Storage B. File 2 has a file size of 17MB and it took 10s to migrate from Storage A to Storage B. File 3 has a file size of 180MB and it took 70s to migrate from Storage A to Storage B. File 4 has a file size of 316MB and it took 90s to migrate from Storage A to Storage B.

Figure 11 shows the throughput for data migration time taken. X axis represents the throughput, Y axis represents file size. Throughput is how much megabyte is calculated in one second. Data of existing system, shows that 110MB' was transferred' in 40s, and in proposed' system', it takes' 20s to migrate' a 110MB' file. In Comparison to existing system', performance' of proposed system is higher. This graph clearly shows' the proposed' system' reduces' the throughput' over the existing' system' by an average of 50% for data migration.

Figure 12 shows the throughput for data migration time taken. X axis represents the throughput, Y axis represents file size. Throughput is how much megabyte is calculated in one second. Data of proposed system, shows that 110MB' was transferred' in 20s, against the existing system', it takes' 20s to migrate' a 110MB' file. In Comparison to existing system', performance' of proposed system is higher. This graph clearly shows' the proposed' system' reduces' the throughput' over the existing' system' by an average of 50% for data migration.

Figure 13 shows the throughput for data migration time taken. X axis represents the throughput, Y axis represents file size. Throughput is how much megabyte is calculated in one second. Data of proposed system, shows that 17MB' was transferred' in 5s, against the existing system', it takes' 5s to migrate' a 17MB' file. In Comparison to existing system', performance' of proposed system is higher. This graph clearly shows' the proposed' system' reduces' the throughput' over the existing' system' by an average of 50% for data migration

VI. CONCLUSION

This proposed project, "An Improved Integrated Model for Data Storage in the Cloud," is a technique that optimizes time for data migration in the cloud storage system and improves on loss of data during the process. Therefore, we can conclude that the application has been developed using the recommended techniques in order to reduce the time for data migration processing and avoid data loss in cloud storage.

REFERENCES

[1]. Abid, f. k. a. k., meddeb-Makhlouf, A. m., & Zarai, F. Z. (2022). Secure mfcc architecture for health care application.. https://doi.org/10.21203/rs.3.rs-1726292/v1
[2]. Antonio, J., Mendes, A., & Silva, F. (2020). Hybrid multi-cloud storage systems. Journal of Cloud Computing, 8(3), 150-165.
[3]. Banavath, R., Reddy, P., Rao, G., & Rajesh, M. (2023). Hybridised optimisation for virtual machine migration and task scheduling in cloud data centres. Journal of Cloud Computing, 12(1), 45-60.
[4]. Buyya, R., Yeo, C. S., & Vecchia, S. L. (2023). Cloud computing and emerging technologies: Third edition. John Wiley & Sons.
[5]. Cai, W., Wang, J., & Zhang, Q. (2021). Data storage and management scheme in cloud storage model. Journal of Cloud Storage, 11(4), 200-215.
[6]. Ellison, T., Brown, P., & Green, R. (2018). Migrating cloud databases: Algorithms and workload models. Journal of Cloud Data Management, 6(1), 45-60.
[7]. Gagandeep Kaur, Sonia Sharma(2017)" Storage Algorithms in Cloud Computing: A Review" International Journal of Computer Science And Technology.
[8]. Ganesh, P., D, E. G., & T, S. K. (2016). Impact of resource management and scalability on performance of cloud applications - a survey. International Journal of Computer Science, Engineering and Applications, 6(4), 01-09. https://doi.org/10.5121/ijcsea.2016.6401
[9]. Hashem, I. A., Yaqoob, I., Anwer, N., Mokbel, O. A., Ghani, A. A., & Khan, S. U. (2020). The rise of big data on cloud computing: Review and open research issues. Information Systems, 91, 717-737.
[10]. Ivana, P., et al. (2020). Hybridised whale optimisation method for resource scheduling in cloud computing. International Journal of Cloud Applications and Computing, 10(1), 33-50.
[11]. Ma, Y., Wang, L., & Chen, J. (2018). Optimising live data transfer in the cloud using stream-based migration strategies. Journal of Cloud Computing, 7(4), 175-190.
[12]. Malik, A., & Khan, S. (2022). Cloud integration: A comprehensive survey. International Journal of Information Management, 64, 102374.
[13]. Niu, X., Zhang, Y., & Chen, S. (2018). Hybrid storage systems: Structures and algorithms. Journal of Cloud Storage Systems, 7(2), 85-100.
[14]. Ravindranadh, K., Rao, K., & Kumar, R. (2018). Honey-encryption cryptographic technique for secure data transfer in cloud computing. Journal of Cloud Security, 5(3), 95-110.
[15]. Samson, D., Joseph, L., & Peter, M. (2020). Data model for cloud computing environment. Journal of Cloud Computing, 9(3), 120-135.
[16]. Srinivasa, G., Reddy, B., & Rao, K. (2020). Improved hybrid algorithm approach based load balancing technique. International Journal of Cloud Computing, 9(2), 110-125.
[17]. Ullah, F., Dhingra, S., Xia, X., & Babar, M. A. (2022). Evaluation of distributed data processing frameworks in hybrid clouds.. https://doi.org/10.48550/arxiv.2201.01948
[18]. Zhao, L., Zhang, J., & Li, H. (2020). Hybrid biogeography-based optimisation for task scheduling in cloud computing. Journal of Cloud Computing, 9(1), 75-89.