ACCELDEDUP

MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Computer Science and Engineering)

Submitted By:

Roban Singh (2104169)

Rupalli Devi (2104172)

Simran Tiwari (2104194)

Guided By:

Er. Shailja

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE

LUDHIANA, 141006

Nov, 2024

ABSTRACT

This project, AccelDedup, focuses on efficient file storage management by preventing duplicate file uploads to cloud storage. Leveraging technologies like Node.js for backend services, React.js for the user interface, and AWS S3 for file storage, the system allows users to securely upload, store, and retrieve files while optimizing storage costs and space. The deduplication process is implemented through file hashing techniques, which check for duplicates by comparing unique file hashes before storing any new file. This ensures that only unique files occupy storage, eliminating redundancy and reducing storage expenses. Additional features include real-time bucket status monitoring, file search and retrieval, and user authentication to secure access. The project demonstrates a scalable and cost-effective approach to cloud storage management, making it highly suitable for personal and organizational use.

ACKNOWLEDGEMENT

We are highly grateful to the Dr. Sehajpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the major project work.

The constant guidance and encouragement received from Dr. Kiran Jyoti H.O.D. CSE Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Project Guide Er. Shailja, without her wise counsel and able guidance, it would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of computer science and engineering department of GNDEC for their intellectual support throughout the course of this work.

Finally, we are indebted to all whosoever have contributed in this report work.

Roban Singh

Rupalli Devi

Simran Tiwari

LIST OF FIGURES

TABLE OF CONTENT

CHAPTER 1 INTRODUCTION

1.1 Introduction to the Project

In today’s statistics-centric international, organizations generate big volumes of records each second which necessitates enormous garage potential and green information management practices. With records being stored throughout diverse structures and systems, statistics redundancy-the accidental duplication of identical records throughout garage locations-has grow to be a massive and costly trouble. As groups extend their digital ecosystems, the need for optimized, price-effective storage solution is extra essential than ever.

The AccelDedup venture addresses this challenge by using presenting an advanced, cloud-primarily based records de-duplication system designed to streamline garage management, decrease redundant records, and decorate system performance

1.1.1 Understanding Data Re-duplication and its Impact

Data Redundancy occur while multiple copies of the equal records are stored in extraordinary places, both because of backup protocol, file duplication, or other garage inefficiencies. This redundancy can several drawbacks:

- Increased Storage Costs: Redundant records consumes valuable storage area, forcing agencies to buy additional capability or migrate to extra large and often greater high priced garage answer.
- Operational Inefficiencies: The Presence of useless information slows down gadget operations, particularly in statistics retrival , backup, and recuperation methods.
- Complex Data Management: Managing massive portions of replica data makes it tougher for companies to preserve a clear, centralized view of their records, leading to a loss of transparency and capability data control errors.

AccelDedup addresses these challenges via implementing intelligent de-duplication methods that now not only lessen storage requirements however also enhance information management efficiency, lower operational costs, and improve gadget overall performance. Through a completely unique aggregate of report, block, and application-level de-duplication. AccelDedup optimizes the data garage landscape, allowing organizations to save most effective precise copies of statistics without compromising accessibility or integrity.

1.1.2 Cores Features and Benefits of AccelDedup

AccelDedup stands out with its comprehensive suite of features, each designed to ensure that organizations can scale their storage solutions with efficiency and flexibility.

- Comprehensive De-duplication: AccelDedup conducts de-duplication at various levels- file, block and application- to identify and eliminate redundant data copies from across the storage infrastructure. By de-duplicating at these multiple layers, AccelDedup ensures optimal storage utilization and prevents a bottleneck.
- Cross-Platform Integration: The solution integrates seamlessly with major cloud storage platforms, such as AWS, Google Cloud, and Azzure. This Cross-platform compatibility ensures that de-duplication can occur across different environments, giving users a unified view of their storage while reducing across diverse data landscapes
- Real-Time De-duplication: Unlike traditional de-duplication methods that are performed in scheduled batches, AccelDedup continuously monitors incoming data and de-duplicates it in real-time. This means that the moment new data is ingested, it is analyzed for duplication, ensuring that storage is efficiently managed from the outset.
- Scalability: As organizations grow and thir storage demands increase, AccelDEdup's architecture is designed to scale seamlessly. The system can handle

high volumes of data without compromising on de-duplication speed or accuracy, making it suitable for enterprises of all sizes, from small startups to large corporations with substantial data footprints.

- Cost Efficiency: By reducing the volume of stored data, AccelDedup significantly lowers storage costs, particularly for organizations with heavy reliance on cloud storage. Reduced redundancy means that less physical and cloud storage is required, translating to considerable savings on storage expenses.
- Enhanced System Performance: With fewer duplicate data entries, systems experience faster data retrieval speeds, quicker backup and recovery times, and improved application performance. AccelDedup ensured that de-duplication optimizes not just storage but overall system functionality.

1.1.3 Impact And Advantages Of Acceldedup

AccelDedup represents a groundbreaking step forward in statistics garage managements, offering a pretty efficient and scalable solution for businesses seeking to minimize records redundancy and optimize garage utilization. By ensuring that handiest precise copies of information are retained, Accel Dedup not handiest reduce garage expenses but also permits quicker information retrieval and progressed machine overall performance. With greater support for cloud platforms and real-time de-duplication skills, AccelDedup positions itself as a useful device for contemporary groups navigating the complicated demands of large facts management.

AccelDeup's mixture of superior de-duplication generations, user-friendly interface and robust cloud compatibility makes it a unique and flexible solution in the landscape of facts storage and optimization. With this assignment, agencies can achieve more manage over their storage sources, lowering redundancy and improving overall operational efficiency.

1.2 Project Category

The AccelDedup task fall beneath the industrial classs, aimed toward addressing the challenges that huge-scale information-driven industries face in managing and optimizing their digital storage. In the era of huge information, industries which include finance, healthcare, telecommunications, and manufacturing handle tremendous volumes of information ever day. This information includes operational facts, customer statistics, transactional statistics, logs, backups, and extra-all of which contribute to huge garage demands. Industrial agencies often rely upon massive information storage structures, disbursed throughout cloud systems and on-premise answers. These complicated garage infrastructures are vulnerable to record redundancy, in which duplicate copies of statistics proliferate across databases, cloud environments and backup structures. This redundancy now not best will increase storage prices but also slows down machine overall performance, impacting operational performance, and scalability. AccelDedup addresses this particular industry ache point by means of supplying a cloud-based totally statistics de-duplication solution that removes needless facts duplicates, thereby allowing more efficient, fee-powerful, and reliable garage management.

Why AccelDedup is an industrial Solution

AccelDedup has been designed to integrate with agency-scale cloud offerings and assist excessive-extent statistics environments ordinary in business settings. The mission leverages scalable, actual-time de-duplication processes that align with necessities of large industries, which demand.

Scalability: The system can cope with large datasets that grow constantly as industrial operations make bigger. Accel Dedup's architecture is constructed to scale dynamically, making sure that it stays powerful at the same time as storage demands increase.

Cross-platform Compatibility: AccelDedup works seamlessly with multiple cloud storage providers, consisting of AWS, Google Cloud, and Microsoft Azure, making it appropriate for industries the usage of hybrid or multi-cloud environments. This flexibility permits industries to standardize their information de-duplication approaches across various systems.

Enhanced Security: Security is paramount in commercial environments in which statistics is touchy and frequently regulated. AccelDedup integrates safety features which include encryption, positions-based totally access manipulate, and compliance with cloud protection standards, making it a robust desire for sectors like healthcare and finance.

Cost Efficiency: For industries that control full-size facts, garage costs constitute a massive fee. By identifying and doing away with replica records, AccelDedup helps lessen these fees, presenting full-size value financial savings and an optimized go back on funding in cloud infrastructure.

Operational Efficiency: Industrial groups rely upon speedy statistics retrieval, real-time analytics, and green backup and recovery approaches. By minimizing garage redundancy, AccelDedup enhances basic system overall performance, allowing quicker records processing and decreased gadget downtime.

1.3 Problem Formulation

In today's data-intensive environment, organizations face the challenge of managing vast amounts of information across cloud platforms. Data Redundancy, or the presence of duplicate files and data blocks, consumes valuable storage resources, inflate operational costs, and slows down system performance. Existing de-duplication method like file-level and block-level de-duplication provide some relief but fall short in cloud environments with large, distributed datasets. For instances, file-level de-duplication may miss identical content saved under different names, while block-level de-duplication offer granular comparison but increases processing overhead, impacting performance. This scenario underscores the need for a more advanced solution that can handle these limitations effectively.

AccelDedup aims to address these issues by developing an intelligent cloud-based de-duplication system that eliminates redundant data across multiple platforms such as AWS, Google Cloud, and Azzure.

Designed to perform in real time and scale as data grows, AccelDEdup continuously Monitors and remove duplicate data, improving storage efficiency and reducing associated costs. The solution enhance overall system performance by reducing unnecessary storage demands, allowing organizations to handle large datasets with speed and efficiency. By combining the strengths of existing de-duplication techniques, AccelDedup seeks to offer a reliable, cost-effective, and highly scalable solution that supports seamless data management in cloud environments.

1.4 Identification/Recognition of Need

In today's facts-centric world, organizations are producing, storing and studying big volumes of information throughout a variety of platforms and environments. As businesses increasingly adopt cloud storage answers to manage this information, they face a mounting tasks: record redundancy. Redundant facts-duplicate files, information blocks, and objects unfold across storage places-consumes valuable storage area, inflates operational charges, and slows down device performance. Cloud storage is inherently luxurious, with prices scaling immediately with data usage, making data redundancy as even more urgent difficulty for groups seeking price-powerful, efficient garage control.

Industries with excessive data demands, consisting of healthcare, finance and telecommunications, and manufacturing, are in particular impacted through facts redundancy. For instances, healthcare organizations have to shop big amounts of patients information, medical imaging files, and research data, regularly with replica entries due to backups and regulatory compliance. Financial institutions, coping with touchy transactional and compliance statistics, additionally revel in substantial redundancy as information is replicated across system for operational continuity and chance management. In telecommunications, duplicate purchaser information, name logs, and provider data gather as companies extend and combine with numerous cloud services. In each of those sectors, redundant facts creates inefficiencies, drives up costs, and can even compromise system overall performance for the duration pf crucial records retrieval or evaluation operations.

1.5 Existing System

1.5.1 iDedup

- Purpose: iDedup is designed especially for primary garage systems, with a primary cognizance on garage capability savings.
- Approach: It selectively de-duplicates large i/o requests, at the same time as small I/O requests(like 5kb or 8kb) are commonly bypassed because their de-duplication is taken into consideration much less profitable due to excessive overhead and low capacity savings.
- Advantages: Reduce the storage area usage for huge blocks, and improves storage performance in number one storage systems.
- Limitations: iDedup doesn't cope with smaller I/O requests and can cause performance issues as it doesn't completely optimize study or write operations. The restrained dealing with of small requests also fails to fully leverage number one garage I/O traits for overall performance.

1.5.2 Offline Deduplication

- Purpose: This gadget is used ordinarily in backup and archiving answers, where deduplication is carried out after statistics is stored.
- Approach: Deduplication is processed offline by using analyzing the information repository and figuring out duplicate publish-garage.
- Advantages: High de-duplication performance and considerable storage potential financial savings.

- Limitations: Offline de-duplication isn't appropriate for actual-time primary garage systems since lacks on the spot de-duplication of incoming records, impacting real-time read/write performance.

1.5.3 Primary Data De-duplication(PDD)

- Purpose: PDD structures goal number one storage environments with a real-time focus on both capability financial savings and performance optimization.
- Approach: PDD answers frequently use inline de-duplication for larger files and offline de-duplication for smaller or much less often accessed documents.
- Advantages: Balances de-duplication and performance in primary storage setting by means of focusing on real-time processing and potential financial savings.
- Limitations: Real-time de-duplication can introduce overall performance bottlenecks and require careful reminiscence and resources management to keep away from I/O delays.

1.5.4 Content-Defined Chunking(CDC)

- Purpose: CDC is broadly used in report-level and block-level de-duplication, focusing on de-duplication for variable-sized chunks rather than fixed-length blocks.
- Approach: CDC divides information into chunks primarily based on content material traits, making it effective in detecting reproduction chunks inside and across documents.
- Advantages: Reduce redundancy extra efficaciously than fixed-size chunking, specifically while coping with files that undergo minor modification.
- Limitations: CDC may be computationally extensive, and indexing smaller chunks can require extra reminiscence, impacting overall performance in large-scale systems.

1.6 Objectives

1.6.1: To create an interface for storing data on the cloud.

The AccelDedup mission achieves its first goal of making an interface for storing data on cloud via a user-pleasant internet application built with React. This interface allows customers to without difficulty add documents, which are then processed to discover and cast off reproduction records the usage of sophisticated algorithms. The utility interacts with a cloud-based totally backend, using APIs to securely save specific facts and preserve metadata for monitoring functions. This streamlined method ensures that users can manipulate their data correctly without worrying approximately redundancy. Furthermore, the application integrates with numerous cloud garage services, providing customers with the power to pick their desired storage answer. This adaptability is essential for corporations dealing with numerous facts environments. By ensuring that most effective unique data is stored and offering customers with a clean view of their storage fame, AccelDedup now not most effective simplifies garage expenses and complements overall machine performance.

1.6.2: To perform De-duplication at the cloud Platform.

AccelDedup employs superior algorithms to perform de-duplication on cloud platforms by analyzing data before it is stored. When users upload files, the application scans for identical copies using hashing techniques that generate unique identifiers for each file. By comparing these hashes, the system can quickly identify duplicates and determine which data can be retained or discarded. This process significantly reduces the volume of data stored, leading to lower storage costs and improved system efficiency. Additionally the de-duplication process is seamlessly integrated into the user interface, allowing users to view their storage consumptions in real-time. Users receive notification about potential duplicate and can manage their data through intuitive options, such as reviewing duplicates and selecting which files to keep. This not only enhances user experience but also ensures optimal utilization of cloud resources, making data management more efficient and cost-effective.

1.6.3: To compare and analyze the existing de-duplication technique with proposed solution

AccelDedup aims to enhance data storage efficiency through its proprietary de-duplication techniques while comparing them with existing methods such as file-level, block-level, and byte-level de-duplication. Traditional file-level de-duplication identifies duplicates files by comparing file names and sizes, which can be limiting, as it fails to recognize identical files stored with different names. In contrast, block-level de-duplication breaks files into smaller 3 chunks, allowing for more granular comparisons, but can lead to increased processing overhead. AccelDedup leverages a hybrid approach that combines the strength of both techniques, utilizing advanced hashing algorithms to analyze data more effectively while minimizing resources usage. Moreover, the proposed solution not only focuses on improving de-duplication efficiency but also emphasizes data integrity and security. By analyzing existing techniques, AccelDedup incorporates best practices while addressing common drawback, such as high peocessing time and potential data, providing features such as selective retention of files, real time monitoring of storage usage, and comprehensive reporting on data redundancy. This holistic approach ensures that AccelDedup stands out in the crowded landscape of cloud storage solutions, offering superior performance and user satisfaction.

1.7 Proposed System

1.7.1 Overview of the Proposed System

The proposed system is a cloud-based deduplication service designed to optimize garage control for users. The system integrates with a cloud garage platform (including)

AWS S3) to locate and do away with replica documents, ensuring that only specific information is saved, thereby decreasing garage costs and improving efficiency.

The center functionality of the system revolves around:

- File Upload: Users can add documents to the cloud storage device.
- Deduplication: The system checks if the record is a replica through comparing its hash (e.G., SHA-256) in opposition to previously uploaded documents.
- Storage Optimization: Identified duplicates are either now not uploaded or replaced with a reference (pointer) to the unique file, thereby saving garage space.
- File Management: The system affords users with get entry to to view, manipulate, and retrieve files at the same time as making sure that handiest unique files are stored.

The proposed device will leverage cloud storage carriers' APIs (e.G., AWS SDK) to interact with cloud services, whilst adding a layer of deduplication on top to automatically discover and do away with duplicates.

1.7.2 Key Features

- The proposed cloud deduplication gadget consists of the subsequent key features:
- File Upload and Management: Users can add documents thru a web interface. The machine will take care of the importing process, music the files, and display them in a user-pleasant format.
- Deduplication Logic: Hashing: When a file is uploaded, the device computes its hash value (e.G., SHA-256). The hash is then in comparison towards current files inside the cloud garage.
- Duplicate Detection: If a document with the identical hash already exists, the system will either:
- Skip Upload: Avoid importing the duplicate record.

- Replace with Reference: Replace the replica document with a reference or pointer to the unique document. This approach reduces storage usage extensively.
- Efficient Search: The system will offer a seek characteristic that permits users to filter out and find files without difficulty, both with the aid of call or hash.
- File Management Interface: A person-pleasant interface to manipulate uploaded documents, take a look at file info (size, remaining modified date, and many others.), and look at or download documents from the cloud storage platform.
- Security: The device will ensure that uploaded files are securely treated. Sensitive information might be encrypted each in transit (the usage of SSL/TLS) and at rest (the use of cloud vendors' built-in encryption mechanisms).
- Scalable Architecture: The machine may be designed to deal with big volumes of documents, the usage of cloud garage's scalability functions to control developing records.

1.8 Unique features of the Proposed System

The proposed cloud deduplication device contains numerous specific capabilities to beautify garage optimization, efficiency, and person experience.

- Advanced Deduplication: Uses cryptographic hashing to hit upon and remove reproduction documents, optimizing storage with the aid of storing best unique files.
- Cloud-Native Integration: Seamlessly integrates with cloud offerings like AWS S3, ensuring scalability, reliability, and minimum setup.
- User-Friendly Interface: Provides an intuitive UI for easy document uploads, search, and document control, with clean feedback on add achievement or failure.
- File Versioning and Deduplication: Handles multiple variations of documents efficiently, storing handiest precise versions and decreasing storage usage. Real-Time Duplicate

Detection: Detects and prevents duplicate uploads right away, ensuring green storage usage.

- File Integrity Verification: Uses hashing to ensure uploaded documents aren't corrupted or altered, making sure data integrity.
- Data Encryption: Implements sturdy encryption both at some stage in file transfer and at the same time as at relaxation to stable person facts.
- Eight. Scalable Architecture: Built to scale with person wishes, handling increasing report sizes and facts volumes with out performance degradation.
- Cost Efficiency: Reduces storage charges via putting off replica documents and optimizing aid use inside the cloud.
- Customizable and Extensible: Allows customization of deduplication good judgment and integration with other cloud-based offerings for bendy expansion.
- These functions together offer a comprehensive, stable, and fee-efficient answer for optimizing cloud storage, distinguishing it from traditional cloud storage structures.

CHAPTER 2 REQUIRMENT ANALYSIS AND SYSTEM SPECFICATION

2.1 Feasibility study

2.1.1 Technical Feasibility

Because the AccelDedup Project makes use of well-established technologies for user authentication, cloud storage, and deduplication, its technological viability is encouraging. By using hashing techniques like MD5 or SHA-256, which create distinct file identities and enable the system to quickly identify duplicates by comparing hash values, deduplication can be accomplished. Choosing a relational or NoSQL database to manage metadata will facilitate quick file retrieval and storage management, and integrating with a scalable cloud storage provider, such as AWS S3 or Google Cloud Storage, guarantees that the solution can handle massive data volumes. Techniques like JSON Web Tokens (JWT) for session handling and password encryption to safeguard user credentials can be used to safely build the user authentication system.

2.1.2 Economic Feasibility

The AccelDedup Project, a cloud-based deduplication solution designed to eliminate redundant data storage, has high potential for long-term savings and sustainable value based on its economic viability. This project's main goal is to provide customers with an effective method of cloud file storage by removing duplicate copies, which eventually results in significant storage cost savings. Users of traditional storage techniques frequently upload several versions of the same content, thereby causing redundancy that costs more money and takes up space. The Accedl system, in contrast, will only save distinct copies of files, reducing the amount of storage space needed and, consequently, expenses.

Setting up the required infrastructure, writing code, and putting the deduplication algorithm, authentication, and user interface into practice are the main causes of AccelDedup's initial development costs. The project team will need to use hashing techniques, such as MD5 or SHA-256, which generate distinct digital signatures for every file, in order to construct a dependable deduplication system. Back-end engineering know-how and integration with a scalable cloud storage provider like Google Cloud Storage or Amazon Web Services (AWS) S3 that can manage large data volumes without compromising performance are needed to put this into practice. Because of the flexible, pay-as-you-go storage pricing offered by these cloud providers, AccelDedup can scale its storage consumption in response to real-time demand, keeping early costs under control.

Economically speaking, AccelDedup offers significant long-term value by skilfully striking a balance between initial development expenses, continuous storage savings, and income-generating prospects. Since data volumes are predicted to increase exponentially in an increasingly digital environment, the project's capacity to optimise storage through deduplication offers a forward-thinking answer. In a market where effective data management is essential, Accedl Dedup offers itself as a cutting-edge, reasonably priced platform by tackling a major cloud storage pain point: redundant data storage. In the end, the project's ability to save storage space, draw in a large user base, and offer long-term revenue potential demonstrate its economic viability and make it a wise investment for consumers and stakeholders alike.

2.1.3 Operational Feasibility

The AccelDedup Project has a high practical feasibility since it is made to be secure, easy to use, and flexible enough to meet different data management requirements while resolving a major storage issue: redundant data. With features like file upload, file bucket status, and account administration through login and logout, the platform appears to have a straightforward yet useful interface from the user's point of view. Particularly for people and companies who might not have a lot of technological know-how, these aspects are crucial for a satisfying user experience. For example, the upload process will be very user-friendly: users will only need to choose files, which the deduplication algorithm would handle automatically. retention..

The backend infrastructure of the system, which includes the deduplication algorithm and cloud storage integration, is set up to function smoothly and effectively behind the scenes. Once set up, the automated deduplication process powered by hashing algorithms like MD5 or SHA-256 requires little manual intervention. Depending on the usage load and file size, the platform's deduplication algorithm will either execute in real time or at predetermined intervals to keep the system responsive to user demands. When a user uploads a file, for example, the system creates a unique hash signature, compares it to records already in existence, and only stores the file if it is unique. lifespan of the platform and user satisfaction.

2.2 Software Requirement Specification Document

2.2.1 Data Requirements

The Accel Deduplication Project's data needs cover a wide range of information types that are necessary for user, file, storage, and system functionality management. The platform will be able to measure storage consumption, maintain user accounts, deduplicate files, and guarantee secure access with the help of this data. As explained below, each type of data has particular needs and is essential to the project's success.

- User Data

  The system will gather necessary user information to guarantee secure and customised access. As the main identifier throughout the database, each user will have a distinct User ID that is created by the system. Username and Password are inputs supplied by the user; usernames are distinct, while passwords are encrypted and saved safely to preserve user privacy. Email (optional) can also be utilised for notifications or account recovery, which improves security and usability. The system keeps track of the Last Login Timestamp and Account Creation Date for administrative purposes in order to keep tabs on user involvement and activity.

- File Data

Since processing files is the main function of the Accedl system, efficient file management necessitates a number of important data points. A distinct File ID is assigned to each uploaded file, aiding in file tracking and organisation within the system. While the File Path saves the location or URL in the cloud storage where the file is saved, the File Name keeps the original name for user reference. The system creates a File hash for every file using techniques like MD5 or SHA-256 to aid in deduplication and guarantee effective duplicate detection. Additional crucial information includes the File Type (e.g., image/png, application/pdf) to identify the file format, the Upload Timestamp to record the time each file was added, and the File Size (in bytes) to determine storage use.

- Deduplication Data

Deduplication-specific data is essential for storage optimisation. The system can effectively identify duplicates by comparing new uploads to existing records thanks to the Hash Index, which stores each file's unique hash value. The file duplicate status improves system organisation by indicating if a file is unique or contains duplicates. Each file's duplicate count is counted in a Duplicate Count field, which allows system reporting on storage optimisation and offers information on how effective space-saving measures.

- Bucket Status

Certain data points in the bucket status are necessary to give users a summary of their storage utilisation. A computed field called "Total Storage Used" gives users information about their data footprint by summing together the sizes of all the distinct files they have uploaded. Space Saved by Deduplication shows how much storage space is saved by storing just unique files, highlighting the platform's efficacy. Furthermore, File Count and Duplicate Count give customers an overview of their unique and duplicate files, which helps them comprehend their storage effectiveness and consumption trends in general.

2.2.2 Functional Requirements

The AccelDedup Project's functional requirements centre on the necessary functions that the system must have in order to fulfil its objectives of effective file management, deduplication, and easy-to-use access. The system should primarily enable users to register for accounts using a distinct username and password, log in safely, and then log out when finished. Encrypting passwords for privacy and safely storing user credentials are part of creating an account. Users should see a dashboard showing their current storage consumption, file count, duplicate status, and the space saved by deduplication after successfully logging in. Users may effectively control their data utilisation with the help of this interface, which gives them precise, real-time insights into their storage operations.

File uploading, where users can choose which files to save in the cloud, is a fundamental feature. These files must be processed by the system by creating a unique hash (such as MD5 or SHA-256) and comparing it to hashes that are already in the database. If the file is duplicated, the user will be informed of its duplicate status and it won't be stored again. If not, the file is saved in the cloud, and the system records pertinent metadata, including the file size, type, and upload date. To improve transparency and accessibility, users should be able to see the status of every file submitted, including if it is a duplicate and the precise location of the file saved in the cloud.

Bucket status tracking, which summarises each user's overall storage utilisation, duplicate files, and the space saved through deduplication, is another functional requirement. This is essential for providing users with information about how well their storage is optimised by the deduplication process. Session management, which uses safe, token-based login sessions (like JSON Web Tokens) to save user login states and protect access credentials, is also crucial. For a flawless user experience, the system must also manage error reporting and alerts. In the event that an error arises (such as an upload failure because of a file size limit), users should be notified and directed to a solution.

Lastly, for system auditing purposes, an activity log should be kept to record all user activities, including login times, file uploads, and file deletions. An additional layer of functionality for security and troubleshooting. In order for users and administrators to track and control costs, the system must also compute storage cost data in order to monitor operating expenses. Together, these functional requirements guarantee that Accel Dedup is a safe, effective, and intuitive platform for deduplication and cloud file management.

2.2.3 Performance Requirements

The AccelDedup Project's performance requirements are made to give users who manage files in the cloud a quick, effective, and scalable experience. While file upload and deduplication procedures, such as hash creation and duplicate checking, should finish in 5 seconds for files up to 10 MB and within 10 seconds for bigger files, key actions, including user login, logout, and dashboard loading should react in 2 seconds. This guarantees a responsive user experience and short wait times. At least 100 users must be able to access the system at once without experiencing any performance issues, and all essential features, such as file uploads and dashboard access, must operate at constant speeds.

Future expansion requires scalability, therefore the system's deduplication and storage infrastructure must be able to handle growing user data and file volumes while preserving quick response times. A minimum uptime of 99.5% is required by reliability requirements, guaranteeing high availability for consumers and avoiding downtime. Additionally, the system aims for a low error rate—less than 1% of file uploads fail—to guarantee consistent performance. The system should log errors for debugging and swiftly warn users in the event of an error, all without interfering with their experience.

2.2.4 Dependability Requirements

The AccedlDedup Project's dependability requirements are essential for guaranteeing the system's availability, fault tolerance, and dependability when managing user data and files. To ensure that users can access their accounts, upload files, and use deduplication functions without any disruptions, the system must provide a high level of availability, with an uptime of at least 99.5%. The system must include strong error-handling features that can identify, record, and recover from any problems that can happen during file uploads, deduplication procedures, or user authentication in order to satisfy these dependability requirements while reducing the negative effects on the user experience. To guarantee uninterrupted operation in the case of a system failure—such as a network problem or a storage failure—the system ought to automatically move to a backup or failover server.

Furthermore, user files' data integrity is crucial. It is imperative that the system guarantees the correct storage, deduplication, and corruption-free retrieval of uploaded files. Data loss or corruption should not result from system crashes or unplanned shutdowns, and the system should be able to successfully recover files and metadata from backups. Maintaining consistency is essential to ensure that the right files are stored and that duplicate files are correctly found and removed, particularly during the deduplication process. The system must be scalable without sacrificing availability or performance in order to accommodate growing data loads and provide steady service, which will further improve dependability.

Backend:

- **Node.js (v14+)**

  Based on Chrome's V8 engine, Node.js is a robust and effective JavaScript runtime environment that enables developers to execute JavaScript code server-side. Node.js uses an event-driven, non-blocking I/O model and functions asynchronously, in contrast to conventional server-side technologies that are synchronous and block other tasks. Because of this, Node.js is especially well-suited for applications like web servers, real-time apps, and APIs that need to be highly scalable and able to manage several concurrent connections. Building apps that handle frequent I/O operations, such as file uploads, database queries, and API requests, is made easier with Node.js's lightweight design and quick speed.
- **AWS SDK**: Developers can communicate with Amazon Web Services (AWS) directly from within their apps thanks to a collection of libraries called the AWS SDK (Software Development Kit). The main purpose of the AWS SDK for the Accedl Deduplication Project is to integrate with Amazon S3, a dependable and highly scalable cloud storage solution. Developers may upload, download, and manage files stored in S3 buckets programmatically using the SDK. By offering a user-friendly interface for file management, access control, and authentication, the SDK abstracts away the complexity of working with the S3 API. The Accedl Deduplication Project may safely upload user files to S3 and benefit from S3's worldwide scalability by using the AWS SDK, guaranteeing that data are dependably stored and available from any location. Furthermore, it offers functionalities like
- **Multer**: Multer is a Node.js middleware designed to manage application file uploads. It is based on the busboy library, which manages multipart file uploads and parses form input. Multer handles various content kinds (such photos, PDFs, or documents) and saves

them to the designated storage location (either in-memory or on disc), making the process of uploading data from a client to a server easier. File size restrictions, file filtering (to permit only particular kinds of files), and destination routes for storing the uploaded files are just a few of the configuration choices it offers. Multer is used to handle user file uploads in the context of the Accedl Deduplication Project, making sure that files are appropriately processed, saved, and prepared for deduplication tests.

- **S3 Hashing:** The practice of creating distinct hash values for files prior to their upload to Amazon S3 is known as S3 hashing. The system may create a distinct fingerprint (hash) for every file by utilising hashing methods like MD5 or SHA-256. This fingerprint is then used to identify duplicate files while they are being uploaded. The hash of a file is calculated upon upload and compared to the hashes of files that have already been uploaded. The system identifies a file as a duplicate and stops superfluous storage if its hash matches that of an already existing file. Optimising storage space requires this deduplication procedure, especially in cloud environments where file storage expenses can be high. The Accedl Deduplication Project guarantees effective file management and lowers errors by including hashing into the upload process.

Frontend:

- **React.js (v18+)**, **React Router**: For UI and navigation.
- **Material-UI/Bootstrap**: For UI components.
- **Axios**: For HTTP requests.

2.2.5 Maintainability Requirement

The AccelDedup Project's maintainability requirements are crucial to guaranteeing that the system can be effectively updated, debugged, and scaled over time. The system should have a modular architecture to facilitate easy maintenance, with each component (such as file upload, deduplication, and user authentication) being independent and able to be changed or replaced without impacting the system as a whole. To make it simpler for future developers to comprehend and alter, the codebase should be clear, thoroughly documented, and adhere to conventional coding practices. This entails creating thorough comments and upholding precise function definitions, which facilitate problem-solving and the introduction of new features or updates.

Additionally, version control (such as Git) should be incorporated into the system's design to facilitate collaboration among developers, track changes, and roll back to earlier iterations as necessary. To find problems early and guarantee the system's stability during updates, regular unit and integration testing should be used. To enable routine regression checks and make sure new features don't interfere with already existing functionality, tests should be automated. In order to facilitate prompt issue resolution, the system should also facilitate effective error logging and monitoring.

The system's components (such as the database and file storage) should be readily upgradeable or replaceable for scalability and performance. This calls for the use of adaptable and modular technologies, such as AWS services, which offer the ability to scale up or down in response to needs. Long-term system maintenance will also be aided by thorough documentation of data models, API endpoints, and system dependencies. To keep the system safe and compliant with best practices, regular security audits and updates should be carried out to fix any vulnerabilities. Overall, the Accedl Deduplication Project will be simple to maintain and modify to meet future requirements if modularity, testing, documentation, and scalability are prioritised.

2.2.5 Security Requirements

Ensuring secure file management, securing user data, and defending the system against potential assaults all depend on the AccelDedup Project's security criteria. The first important security need is authorisation and authentication of users. Strong password regulations must be enforced by the system, requiring users to generate complicated passwords that are then stored safely hashed using techniques. In order to prevent sensitive data from being revealed in session cookies or URLs, user sessions should be controlled using safe, token-based authentication To enhance security, the system ought to incorporate multi-factor authentication (MFA), which requires users to confirm their identity using multiple methods (such as an SMS code or authentication app) prior to gaining access to their account. This is particularly important for sensitive operations like deleting files or changing account settings.

2.3 SDLC model to be used

For the AccelDedup Project, the Agile Software Development Life Cycle (SDLC) model is the most suitable approach. Agile is a flexible and iterative development methodology that focuses on delivering small, incremental improvements over time rather than waiting for a final product. This model emphasises collaboration, customer feedback, and adaptability to change, making it ideal for projects that evolve based on user needs and requirements.

2.3.1. Requirement Gathering and Analysis

The development team works with stakeholders to establish high-level project needs at the start of the project, including the capacity to upload files, duplicate files, authenticate users, and integrate with cloud storage. The development process will be directed by these needs, which are recorded as user stories and product backlog items.

2.3.2 Planning

The development team chooses which items from the product backlog to focus on during the next sprint at each sprint planning meeting. For instance, the team may dedicate one sprint to the deduplication technique and another to the implementation of the file upload feature. Every sprint is designed to produce a working project increment that is prepared for evaluation and testing.

2.3.3. Development and Implementation

Developers collaborate with other team members (such as UI designers and testers) as needed to complete the tasks outlined in the sprint plan. In order to make sure that new code blends seamlessly with the current system, the project's frontend and backend features are created concurrently, with ongoing testing and code integration. Because Agile is iterative, every feature is created, tested, and improved in brief cycles.

2.3.4. Testing and Validation

Testing is done throughout the development process, with unit tests and integration tests running continuously to ensure each feature works as expected. At the end of each sprint, the increment (which includes the newly developed functionality) is tested more extensively, and user acceptance testing (UAT) may be performed to validate that the feature meets the user's needs.

2.3.5. Review and Retrospective

Each sprint concludes with a review meeting with stakeholders to present the features that have been produced and solicit input. This input is essential for determining whether the project is on schedule and whether any changes are required. To assess the sprint process itself what worked and what may be improved for the next sprint—the development team also holds a retrospective meeting in addition to the sprint review.

2.3.6. Maintenance and Support

The project is delivered in phases following the development, testing, and review of important features. As each sprint is completed, regular updates are released, with new features and improvements being made over time. The project is still available for continuous modifications even after the original release, fixing issues, adding new features, or enhancing performance in response to user feedback.

Why Agile SDLC is Suitable for This Project

The Agile model is particularly suited for this OBE platform because:

- **Iterative Development**: New features can be delivered and tested in small increments, allowing quick feedback and reducing risks.
- **Stakeholder Collaboration**: Agile involves stakeholders throughout, ensuring the platform meets evolving institutional needs.
- **Flexibility**: Agile's iterative nature allows for adjustments based on user feedback, enhancing the relevance and usability of the platform.
- **Focus on Continuous Improvement**: With retrospectives and regular reviews, Agile supports continuous enhancement, a critical aspect for a platform that must adapt to new academic requirements over time..

By following this Agile SDLC approach, the project team can adapt to changing requirements, deliver value incrementally, and ensure alignment with stakeholders' needs throughout the development process.

CHAPTER 3 SYSTEM DESIGN

3.1 System Design

In crafting the design approach for the AccelDedup, a deliberate and strategic approach has been made to embrace the object-orientated paradigm. The methodology is not merely a technical decision but a comprehensive strategy to empower Intellicourse with a design philosophy that transcends the conventional boundaries of function-orientated approaches

3.1.1 Foundation of Object-Orientated

Encapsulation is essential to the efficient organization and security of data components in the design of a deduplication system. For example, a `DataBlock` class contains each data block and associated metadata, including the timestamp, hash value, and file ID. By limiting direct access to the block's characteristics and permitting controlled modifications using specified methods, this structure improves data security by isolating and allowing separate management of each data block'scontents

A distinct `AccelDedup Engine` class encapsulates all deduplication-related functions, including managing a repository of unique blocks, comparing data blocks, and computing hash values, in order to expedite the deduplication process itself. This solution encourages code modularity and readability by keeping the deduplication logic in a single, central location. Coordinating is the responsibility of the `Deduplication Engine`.

3.1.2 Modularity for Sustainable Growth

AccelDedup's scalability and design are guided by the fundamental idea of modularity for sustainable expansion. Each part of AccelDedup, including data ingestion, duplicate detection, and storage optimization, is contained within separate, reusable modules thanks to the implementation of a modular architecture. This framework encourages lifespan and adaptability by allowing the addition of new features or platform integrations (such as moving from AWS to multi-cloud setups) without interfering with current procedures . AccelDedup's modularity makes use of OOP principles to guarantee maintainability and scalability. Each deduplication operation, such as file comparison or storage management, can function separately thanks to **encapsulation**, which allows for the improvement or debugging of certain system components without compromising others. As AccelDedup grows, specialized classes can expand the main framework created by inheritance, which further makes code

3.2 Design Approach

3.2.1 Modular Design

File uploads, file management, user authentication, and bucket status retrieval are just a few of the distinct components that make up the system's modular structure. By ensuring that each module can be independently built, tested, and maintained, this modularization reduces the interdependencies within the system. Furthermore, future additions of functionality can be made without compromising other modules. This method guarantees a more manageable codebase, enhances code reusability, and makes debugging easier.

3.2.2 Client-server Architecture

The system has a traditional client-server design, with the server side (Node.js with Express.js) managing the backend logic, processing requests, and interacting with the database and cloud storage, and the client side (React.js) serving as the user interface. Using RESTful APIs, the client sends HTTP queries to the server, which replies with information or success/error messages. By separating issues and facilitating separate server scaling, this division improves security, performance, and scalability.

3.2.3 Layer Architecture

Presentation Layer: Data display, user input processing, and view rendering are all handled by the user interface (UI) created using React.js.

Business Logic Layer: Essential business logic, including file uploads, deduplication checks, and connection with the AWS S3 bucket, is managed by the Node.js backend with Express.js. Data Layer: To ensure that all files are safely maintained and kept, the system uses cloud storage(AWS S3) as the data layer. A smooth experience is made possible by abstracting away data management and file retrieval from the user.

3.2.4 Cloud-Based Infrastructure

For file storage, the system makes use of cloud infrastructure, particularly AWS S3. This guarantees:

Scalability: AWS S3 can manage high file upload and data storage volumes without seeing a drop in performance.

Availability: Even during periods of high traffic, files are always accessible thanks to S3's high availability.

Reliability: The system guarantees that uploaded files are not lost and can be recovered even in the event of hardware failures thanks to S3's inherent redundancy and durability characteristics. It is feasible to integrate the system in the future with other cloud providers like Google Cloud or Azure, which will increase its flexibility and adaptability

Security is a critical design element. The system integrates various techniques to ensure that data is protected both at rest and during transmission:

- **Authentication**: User authentication is handled using hashing sh-256 algorithm ensuring secure user sessions and protecting sensitive endpoints.

- **Encryption**: User passwords are encrypted using hashing sh-256 before being stored in the database to ensure password safety.
- **Authorization**: Role-based access control (RBAC) can be implemented to restrict user access to specific features or data, ensuring that only authorized users can perform certain actions (e.g., file uploads, view uploaded files).
- **HTTPS**: All communication between the client and server is encrypted via HTTPS, ensuring that sensitive data like user credentials and files are protected during transmission.

ResponsiveDesign

The frontend will be designed to ensure accessibility and usability across a variety of devices using responsive web design principles:

- **Material-UI** or **Bootstrap** will be used for building a flexible and visually consistent UI that adjusts to different screen sizes.
- The system will ensure smooth navigation and interaction on desktops, tablets, and mobile phones, enhancing the overall user experience (UX).

ErrorHandlingandLogging:

To ensure smooth operation and provide transparency in case of errors, the system will include robust error handling mechanisms:

- **Centralized Error Handling**: All errors, both client-side and server-side, will be caught and logged with meaningful messages, which will allow quick diagnosis and resolution.
- **Logging**: Logs will be captured for both user actions (such as file uploads) and system errors (such as failed requests). This will enable efficient troubleshooting, performance monitoring, and auditing.

- **Notifications**: Administrators or users will be notified in case of critical issues, such as failed file uploads or system downtime.

Extensibility:

The system will be designed with future enhancements in mind:

- **New Features**: New features such as advanced file searching, file versioning, or user role management can be added with minimal disruption to the existing system.
- **Cloud Integrations**: Future cloud services or integrations, such as additional file storage providers (Google Cloud, Azure), can be incorporated without major changes to the overall architecture.

Microservices: As the system scales, it can be refactored into microservices, where each functionality (e.g., file handling, user management) is handled by independent services. This ensures that the system can handle increased traffic or complexity in the future.

3.2 Detail Design

3.2.1 Flowchart of the work

The high-level procedures of file upload, deduplication, and storage management in AccelDedup are depicted in the flowchart. When a user logs into the system, the flow starts. The system verifies the validity of the user's credentials upon login. An error notice is shown and the user's session ends if the login attempt is unsuccessful.

After successful authentication, the user can upload a file to cloud storage by accessing the dashboard. In order to detect duplicate files, the backend generates a unique hash for each file throughout the upload process, usually using the SHA-256 hashing technique. After that, the system determines whether a file with the same hash value is already stored.

The user is notified that the file already exists and the file upload is skipped if a duplicate is found. The hash and file metadata are saved to enable future deduplication checks if the file is unique and transferred to cloud storage. Lastly, the user may see their updated storage consumption, which gives them insight into the state of their files and storage efficiency.

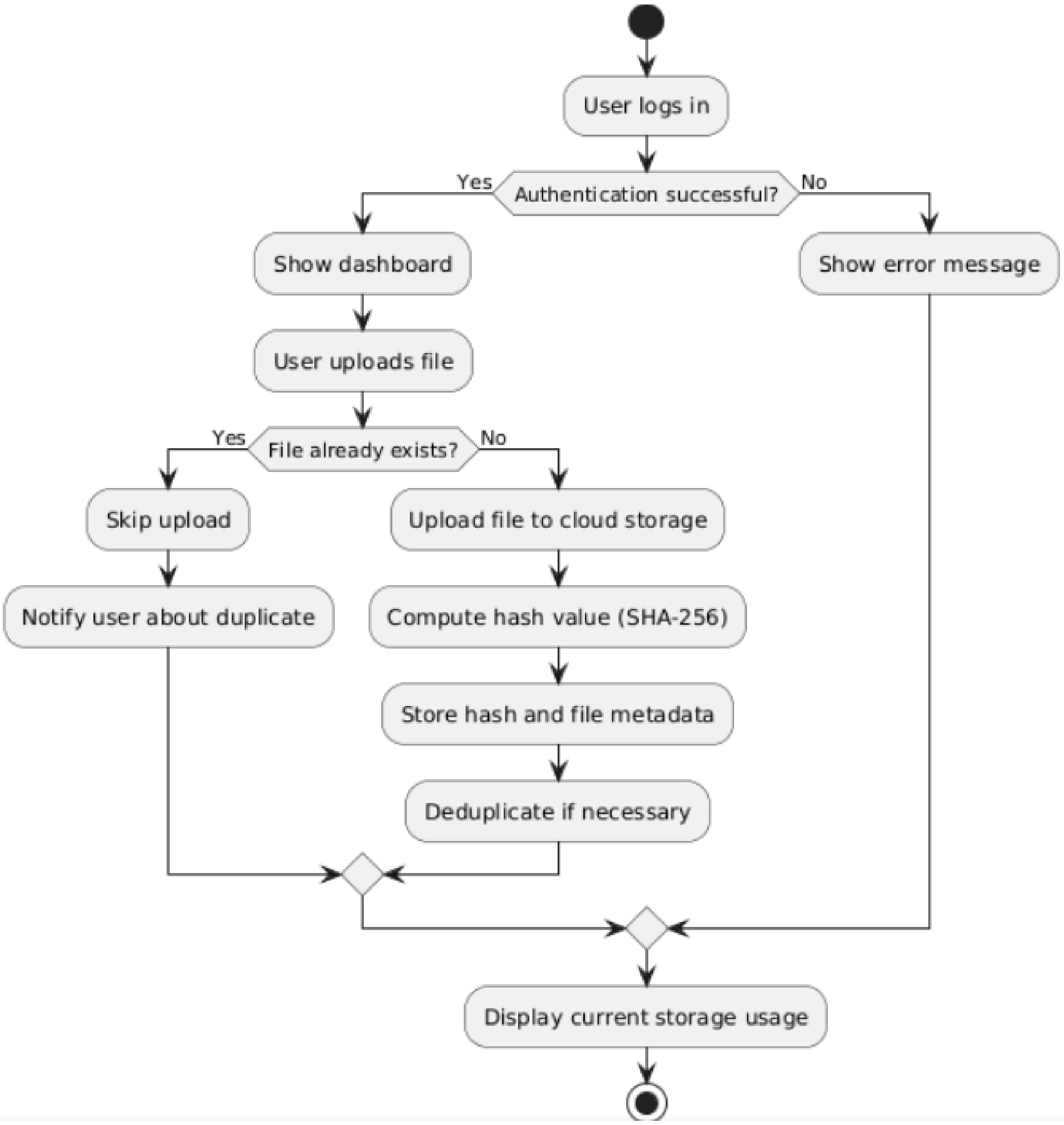

User logs in
Yes
Authentication successful?
No
Show dashboard
Show error message
User uploads file
Yes
File already exists?
No
Skip upload
Upload file to cloud storage
Notify user about duplicate
Compute hash value (SHA-256)
Store hash and file metadata
Deduplicate if necessary
Display current storage usage

Fig 3.1(Flowchart of work)

3.2.2 Sequence Diagram

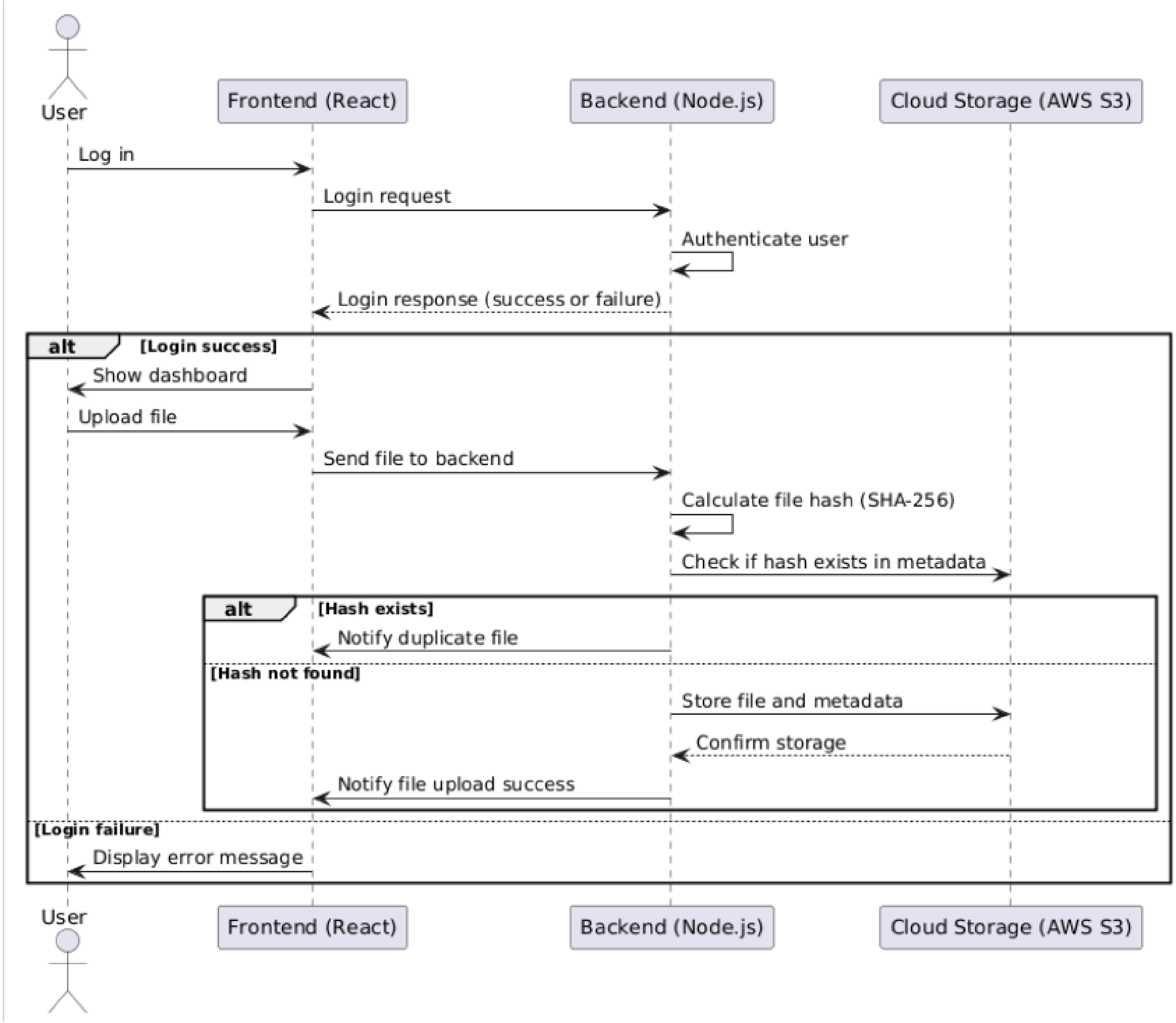


Fig 3.2(Sequence Diagram)

Explanatiom of Sequence Diagram

- **User Login Page**

  When the user enters their login information, including their username and password, the frontend interface (which was created using React) begins the login process. The backend (which is implemented in Node.js) receives these credentials from the frontend in a login request for authentication. The credentials are validated by the backend, which could entail comparing them to an external authentication service or a database. Following the completion of this authentication procedure, the backend notifies the frontend of whether or not the login was successful. The frontend shows the user's dashboard, providing access to further program functionalities, if the login is successful. But if the login doesn't work, the user is prompted to try again with the right credentials via the frontend's error message.

- **File Upload Process (If login is Successful)**

  Once the User has successfully logged in and gained access to the application, they are presented with the dashboard—the main interface for navigating available features. Among these options, the user can choose to upload a file. The file upload process begins when the user selects a file from their device. This action triggers the frontend (built in React) to handle the file selection and prepare it for upload**.**

  Then, by encapsulating the file data—which frequently includes crucial metadata like the filename, size, and type—into a structured format like a FormData object, the Frontend starts the upload. Because it makes it possible for binary data, such as files, to be safely exchanged over HTTP, this format is frequently used to handle file uploads in online application.

- **Backend Authentication Verification**

  In order to make sure that only authorized users are able to access the application, the Backend starts the authentication process as soon as it receives the login request from the frontend. Verifying that the entered login credentials (password and username) match those stored in the system or with a reliable third-party source is the first step in the authentication process. There are two main approaches to this process: using an authentication service or database verification.

  The backend strengthens the application's security and safeguards private user information by completing this authentication procedure, which guarantees that only users with legitimate credentials can access the system. This stage acts as a gateway, letting only authorized users into the program, whether via database verification or an service.

Returning the Authentication Verification

Upon finishing the authentication check, the backend generates a suitable answer according to the validity of the credentials that were submitted. This response is important since it dictates what happens next for the user and the frontend, either allowing access to the application or directing the user to submit their credentials again. Successful Authentication Process

If the backend confirms that the credentials are correct, it formulates a success response to send back to the frontend. This response typically includes an authentication token that serves as proof of the user's identity and authorization for accessing the system.

The backend creates a failure response and sends it to the frontend, usually with an error message, if it discovers that the credentials are invalid. For the user, this error message is essential since it explains why their attempt to log in was failed, such as "Invalid username or password" or "Account not found."

The user's access to the application is determined by the backend's response to the authentication attempt. While a failure answer keeps the system secure and gives the user feedback, a successful authentication response makes the session smooth and safe. By using tokens like session IDs or JWTs, the application can function more effectively and the frontend can authenticate requests without the user having to log in frequently, improving user experience and security.

3.2.3 Activity Diagram

Explanation of Activity Diagram

User Login Process

- The process begins when the User initiates the login by entering their credentials on the frontend. This data is then forwarded to the backend for validation.
- The backend checks the validity of the credentials. If correct, it generates a session token which is sent back to the frontend to manage the session. This token acts as an authentication mechanism, allowing the frontend to handle subsequent requests without requiring the user to log in repeatedly.
- If the credentials are invalid, the backend sends an error message, which is displayed on the login form, prompting the user to re-enter their credentials.

Displaying the Dashboard and Initiating File Upload

- Upon successful login, the user gains access to the dashboard, where they can choose to upload files to the cloud storage.
- The file upload process is initiated when the user selects a file and confirms the upload action. The frontend sends the file, along with any necessary metadata, to the backend for processing.

File Hashing and Duplicate Check

- Once the backend receives the file, it generates a SHA-256 hash of the file, which acts as a unique identifier for that file’s content.
- The backend then checks if this hash already exists in its records by comparing it with the hash metadata stored in AWS S3 or a dedicated hash index.

- If a file with the same hash already exists (indicating a duplicate), the backend notifies the frontend, which informs the user that the file has already been uploaded and doesn't store it again. This prevents redundant storage and optimizes storage usage.

Storing Unique Files in Cloud Storage

- If no existing hash matches the new file's hash, indicating that the file is unique, the backend proceeds to upload the file to AWS S3.
- Once the file is successfully stored, the backend sends a confirmation message to the frontend, where the user receives feedback that the file upload was successful

By ensuring that only authorized users may upload data and that duplicate files are found and handled efficiently, this procedure maximizes storage use and improves system performance. The activity diagram offers a clear flow of choices, actions, and feedback, demonstrating a practical and effective method for handling massive file uploads to cloud storage.

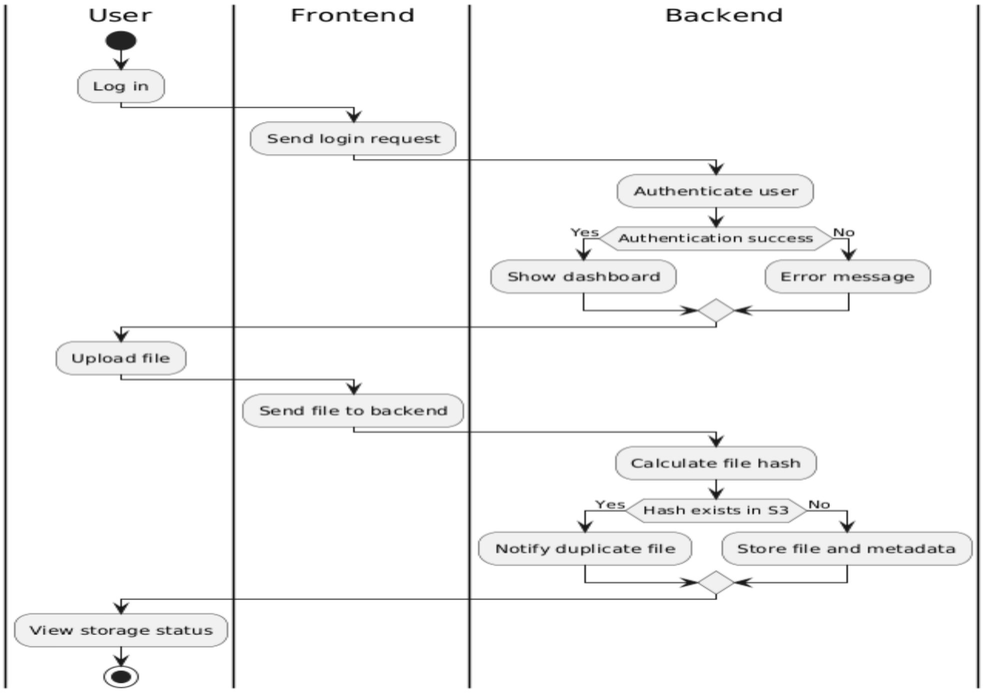


Fig 3.3 (Activity Diagram)

3.2.4 DFD (DATA FLOW DIAGRAM)

The Acceldedup system's Data Flow Diagram (DFD) shows the main procedures and relationships involved in controlling and maximizing data storage through deduplication. Fundamentally, the DFD illustrates how the user engages with the Acceldedup System, starting processes like requesting deduplication reports or uploading files. The user's data files are ingested via the File Upload procedure, which gets them ready for analysis. The Deduplication Process compares incoming files to existing records after files are uploaded in order to look for redundant data. In order to save storage space and guarantee that only unique data is kept, this procedure finds duplicate files, which are subsequently deleted or referenced.

After deduplication, the Storage Management feature effectively arranges and preserves the distinct data in a cloud-based system, maximizing storage capacity and preserving data availability. The system may provide comprehensive reports on deduplication actions through the Report Generation process, giving customers information about storage savings and the degree of redundancy removed. In order to confirm each user's credentials and guarantee that only authorized users have access to system resources, User Authentication is finally put into place. Through the reduction of redundancy and the proper handling of storage requirements, this DFD provides a thorough understanding of how the Acceldedup system functions to promote safe Economical data management.

LEVEL-0 DFD

The Acceldedup system's Level 0 Data Flow Diagram (DFD), represented by this code, provides a high-level overview of the main procedures and user interactions. The "User," who is the main actor in the process, engages with the "Acceldedup System" directly to start things like file uploads and requests for deduplication reports. Data and user orders enter the system through this interaction, which triggers a number of procedures meant to maximize storage and guarantee data security

Several processes in the Acceldedup System manage different tasks that work together to facilitate effective data management. Incoming data is processed by the File Upload component in order to get it ready for deduplication. To guarantee that only unique data is stored, the Deduplication Process finds and eliminates unnecessary files. Data organization is optimized using Storage Management optimizes data organization and storage by utilizing cloud infrastructure for effective storage utilization. Summaries of deduplication operations are produced by the Report Generation function, which gives users information on duplicate file statistics or storage savings. Last but not least, User Authentication verifies user credentials, protecting information and system operations from unwanted access. When combined, these elements offer a thorough description of how Acceldedup facilitates user access, safe storage management, and file deduplication in a smooth, effective way.

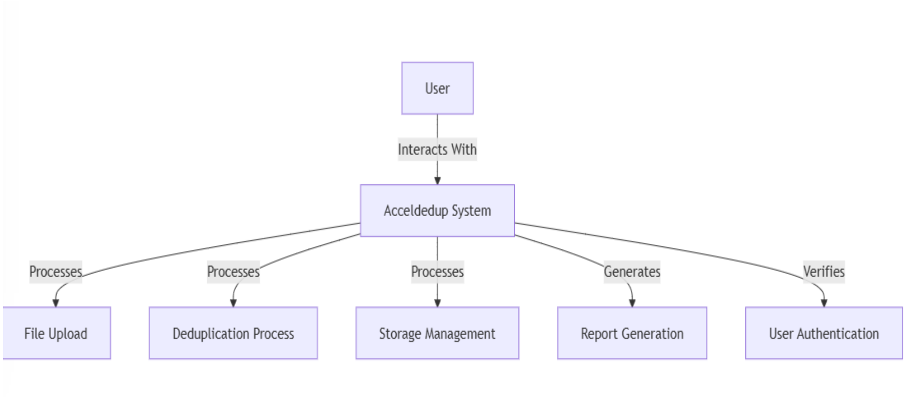


Fig 3.4 (Level-0-DFD)

LEVEL-1-DFD

The Acceldedup system's intricate process flow is depicted in this flowchart, which shows how it manages file uploads, deduplication, user authentication, and report production. The user interacts with the Acceldedup System to start the procedure. Before a file is uploaded, it must first pass through the File Upload procedure, which includes a hash calculation. In order to ascertain whether the file already exists in the system, the Check for Duplicates procedure uses the unique identifier (hash) that is created in this stage. To prevent redundancy, the system either skips the upload or makes a reference to the original file if a duplicate is found. The file is uploaded to the cloud and safely saved in the storage if it is unique.

The system has User Authentication in addition to the file upload and deduplication procedures. Users go through Login Validation here, where their login information is examined. A Session Token is created after validation is successful, giving the user safe access to the system's features. The technology also offers the ability to generate reports. After this procedure, the customer receives Deduplication Reports for inspection, which include information on storage utilization and redundancy elimination. Within the Acceldedup system, these interrelated procedures work together to provide effective data deduplication, safe user access, and thorough reporting.

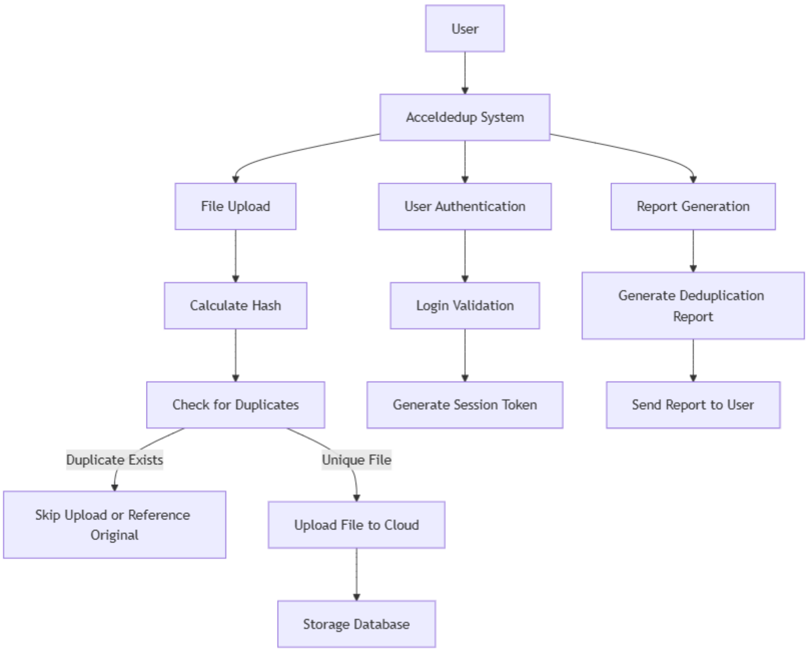


Fig 3.5(Level-1-DFD)

Level-2 DFD

This flowchart illustrates how the Acceldedup system handles file uploads, deduplication, report generating, and user authentication.

Uploading and Deduplicating Files

When the user starts a file upload, the process starts. In the Compute SHA-256 Hash stage, the system uses the File Metadata—such as the file name, size, and type—to create a unique hash value for the file. By acting as a distinct identification for the file's contents, this hash makes precise duplicate detection possible. After being calculated, the hash is routed to the Check Against Hash Index procedure, where it is contrasted with hashes that have already been recorded. No extra storage is utilized if a duplicate is discovered; instead, the system alerts the user of the duplicate status**.**

Creation of Reports

By submitting a Request Deduplication Report, users can also obtain information on storage and deduplication operations. After receiving this request, the Report Generator component starts the Fetch Deduplication Data process, which retrieves pertinent information like the number of duplicates and the storage space saved by deduplication. A report customized for the user's storage activity is created using this data in the Generate User Report function. Lastly, the system provides the user with a report that includes information on data management performance and storage efficiency.

The User Authentication procedure controls access control to ensure security. A user's login credentials are validated during the Validate Credentials step of the login process. In order to authenticate and track the user's session, the system creates a Session Token in the Generate Session Token field if it is successful. As indicated by Access Granted, this token enables the user to safely access the system.Users may manage their storage resources on the Acceldedup platform in an optimized and secure manner thanks to these interconnected workflows, which guarantee secure access, effective data deduplication, and insightful reporting.

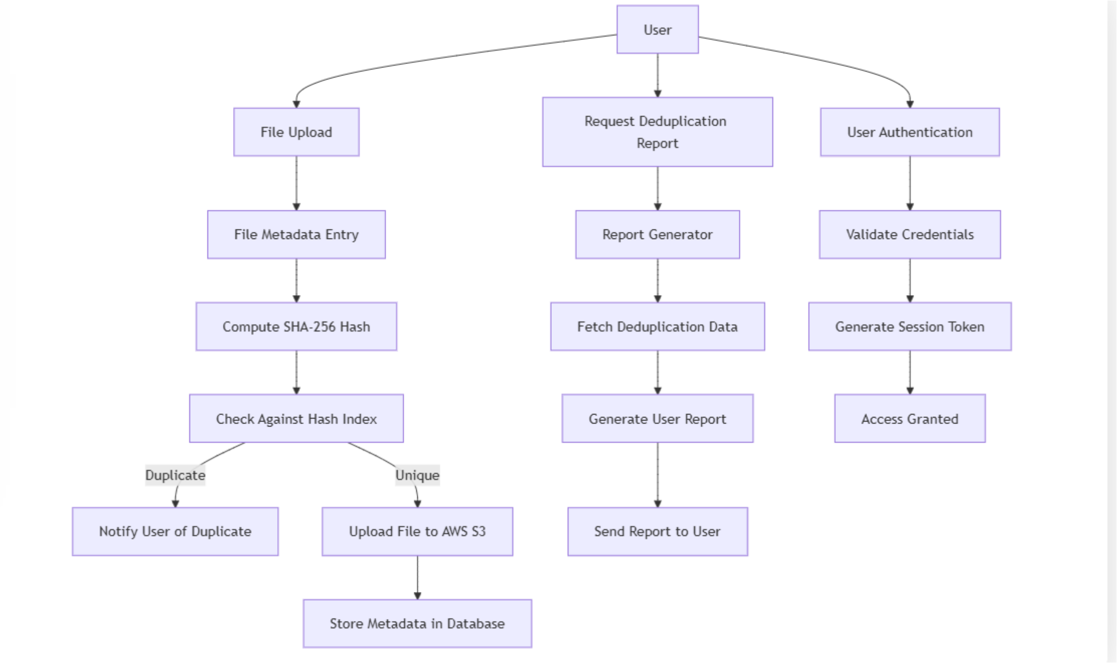


Fig 3.6 (Level-2-DFD)

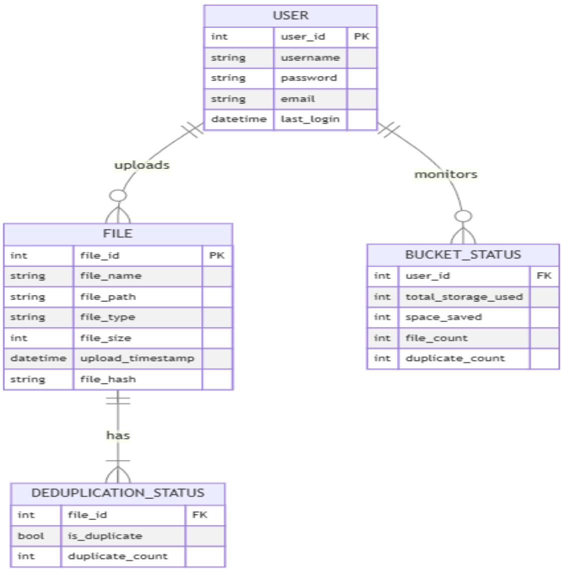
3.2.5 ER Diagram

With an emphasis on users, files, deduplication status, and storage monitoring, this Entity-Relationship (ER) diagram illustrates the main entities in the Acceldedup system and their connections.

Each user's unique user_id, username, password, email, and last_login timestamp are all stored in the USER entity. The USER to FILE relationship (USER ||--o{ FILE : "uploads") indicates that users in the system are associated with the files they upload. Multiple files can be uploaded by a single user, establishing a one-to-many relationship in which every file is linked to a distinct FILE record. Each uploaded file's file_id, file_name, file_path, file_type, file_size, upload_timestamp, and file_hash (a distinct hash for deduplication) are all contained in the FILE entity.

Deduplication-specific data, including whether a file is a duplicate (is_duplicate) and the number of duplicates (duplicate_count), are stored in the DEDUPLICATION_STATUS object. The system can identify which files are duplicates and how many copies have been deleted because this entity is related to FILE (FILE ||--|{ DEDUPLICATION_STATUS: "has"), which indicates that each file may have an associated deduplication status**.**

Last but not least, the BUCKET_STATUS entity has values like total_storage_used, space_saved (via deduplication), file_count, and duplicate_count that record each user's storage consumption. BUCKET_STATUS (USER ||--o{ BUCKET_STATUS : "monitors") is connected to the USER entity, indicating a one-to-one relationship in which each user keeps track of their storage statistics. Effective user account management and tracking, file storage, deduplication, and storage reductions are made possible by these entities and relationships working together.



USER		
int	user_id	PK
string	username	
string	password	
string	email	
datetime	last_login	

FILE		
int	file_id	PK
string	file_name	
string	file_path	
string	file_type	
int	file_size	
datetime	upload_timestamp	
string	file_hash	

BUCKET_STATUS		
int	user_id	FK
int	total_storage_used	
int	space_saved	
int	file_count	
int	duplicate_count	

DEDUPLICATION_STATUS		
int	file_id	FK
bool	is_duplicate	
int	duplicate_count	

Fig 3.7 (ER Diagram)

3.3 Methodology of work

The goal of this project's technique is to create a cloud-based file management and deduplication system that is safe, effective, and scalable. It entails a number of phases, such as architectural design, authentication implementation, file upload processing, and deduplication to remove redundant data. The technique is broken down as follows:

3.3.1 Requirement Analysis

A high-performance deduplication solution called Acceldedup was created to minimize redundancy in big datasets and maximize data storage. Finding the particular requirements and difficulties related to data duplication is part of the system's requirement analysis, particularly in settings with high data generation and storage volumes, such cloud-based services, data centers, and businesses managing big data. Making ensuring the deduplication procedure effectively finds and eliminates duplicate data blocks without sacrificing the system's speed or scalability is a key need. In order to prevent data loss or corruption throughout the deduplication and retrieval procedures, Acceldedup should also preserve data integrity. To keep up with high-speed data flows, it must enable compliance with a variety of data kinds and formats, connect smoothly with current storage systems, and offer real-time deduplication capabilities. Furthermore, the

3.3.2 Feasibility Study

Through an examination of technological, economic, operational, and legal elements, the feasibility study for Acceldedup evaluates its potential for practical implementation. Because Acceldedup is built on well-established deduplication algorithms, tailored for speed and scalability, and built to integrate with contemporary storage infrastructures, it is technically feasible. It is suitable for real-world enterprise application due to its adaptability to different data contexts; nonetheless, comprehensive performance testing is necessary to guarantee resource efficiency. Economically speaking, Acceldedup can save a lot of money by eliminating unnecessary data storage, which eventually lowers hardware and operating expenses. The project is financially feasible due to the long-term decrease in storage costs, even though the initial setup may be expensive. In terms of operation, Acceldedup is meant to be simple and unobtrusive, requiring little training for administrators while preserving end users' uninterrupted access to data. From a legal standpoint,

3.3.3 System Design

The foundation of Acceldedup's system design is an architecture that effectively locates and removes redundant data blocks in large-scale storage settings. Starting with a data intake module that filters incoming data streams and utilizes a fingerprinting technique to create unique identifiers for every data block, the system employs a multi-layered approach. Because these fingerprints are kept in an index database, they can be quickly compared to newly added data. The system minimizes storage utilization by replacing redundant data with references to the original data block when duplicates are found. Scalability and real-time processing are key components of Acceldedup, which uses parallel processing strategies and high-speed storage options like SSDs or NVMe to provide faster data access. Furthermore, a security layer guarantees data encryption and adherence to privacy guidelines

3.3.4 Implementation

Setting up the essential parts of Acceldedup, such as the index database, fingerprinting algorithm, data intake module, and storage management system, is the first step in the implementation process. The system may process files, blocks, or streams because the data intake module is set up to handle different kinds of data. Each data block is then given a unique identifier created by integrating the fingerprinting process, which is subsequently saved in the index database for easy comparison. Parallel processing techniques are used to guarantee scalability, allowing the system to manage massive data volumes concurrently without compromising speed. By connecting duplicate data blocks to the original references, data storage is therefore streamlined, hence saving storage space. To protect data privacy, a security layer with encryption techniques is included, and a dashboard for monitoring and reporting

3.3.5 Testing

In order to ensure that Acceldedup is efficient in high-volume data environments, load and performance testing are essential; these tests evaluate response times, deduplication rates, and overall system throughput; vulnerability and penetration testing are carried out to identify and mitigate any risks related to data breaches or unauthorized access; and user acceptance testing (UAT) is carried out with actual users to verify usability and functionality in live scenarios, allowing for any issues. Acceldedup testing includes a comprehensive approach to validate functionality, performance, security, and scalability.

3.3.6 Deployment

Acceldedup's deployment is set up to optimize user accessibility, scalability, and performance. First, the backend system is built up on Vercel, using its serverless architecture to manage indexing, deduplication logic, and data processing in a high-performance environment that can grow with the volume of data. Acceldedup's backend can effectively handle computationally demanding deduplication activities while preserving high availability and quick response times thanks to Vercel's capabilities. Render was selected for the frontend's deployment because to its dependable hosting and ease of setup. The user interface, hosted by Render, enables users to easily engage with system features, follow deduplication statistics, and keep an eye on storage savings. The system's flexibility and efficiency are increased by this deployment technique, which divides frontend and backend services. This enables both components to scale independently and be upgraded as necessary without impacting the other. The result is a seamless, reliable, and scalable deployment tailored to meet the needs of data-driven environments.

3.3.7 Updates and Maintenance

The goal of Acceldedup updates and maintenance is to keep the system safe, effective, and adaptable to changing data management requirements. To maintain excellent speed as data volume increases, Vercel's backend is regularly updated with changes to the deduplication algorithm, data processing optimizations, and index management. Any vulnerabilities are quickly fixed with security updates, protecting user privacy and adhering to legal requirements. Updates for the Render-hosted frontend could include UI/UX refinements, user-feedback-driven new features, and improvements for improved analytics and monitoring. To make sure the system can manage higher data loads without experiencing a performance deterioration, routine maintenance chores include load testing, server performance monitoring, and storage efficiency checks. Vercel and Render both permit for easier updates, allowing the frontend and backend to be controlled separately. Continuous improvement, little downtime, and a responsive system that seamlessly adjusts to user demands and technology breakthroughs are all made possible by this framework.

Flowchart of Methodology

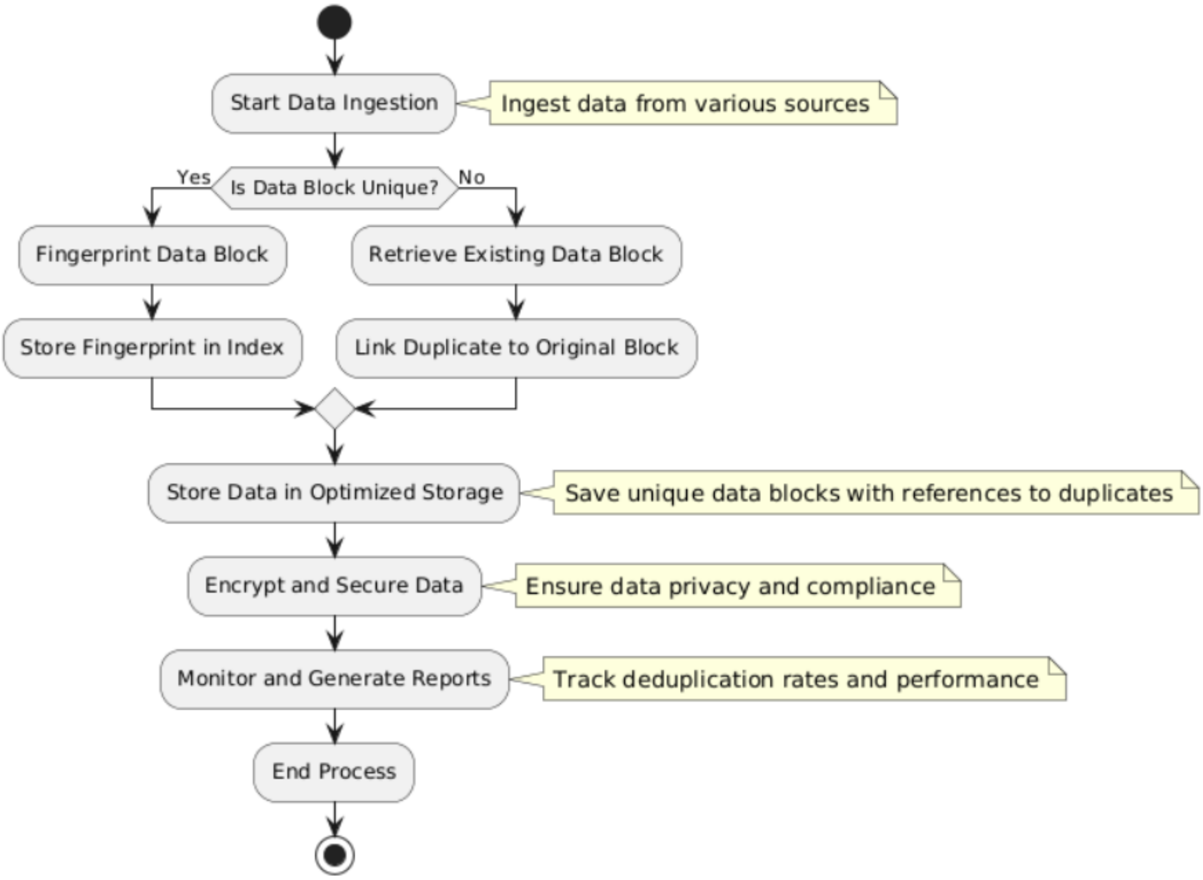

Start Data Ingestion
Ingest data from various sources
Yes
Is Data Block Unique?
No
Fingerprint Data Block
Retrieve Existing Data Block
Store Fingerprint in Index
Link Duplicate to Original Block
Store Data in Optimized Storage
Save unique data blocks with references to duplicates
Encrypt and Secure Data
Ensure data privacy and compliance
Monitor and Generate Reports
Track deduplication rates and performance
End Process

Fig 3.8 (Methodology of work)

CHAPTER 4 IMPLEMENTATION AND TESTING

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Project work

4.1.1 Languages and framework

The Cloud Deduplication System leverages modern and robust programming languages and frameworks to ensure scalability, performance, and ease of development. Below is an overview of the languages and frameworks used in both the backend and frontend development.

Backend

- **Node.js**:

  Node.js is a powerful and efficient JavaScript runtime environment built on Chrome's V8 engine, allowing developers to run JavaScript code server-side. Unlike traditional server-side technologies that are synchronous and block other operations, Node.js operates asynchronously and uses an event-driven, non-blocking I/O model. This makes Node.js particularly well-suited for applications that require high scalability and the ability to handle multiple concurrent connections, such as real-time applications, web servers, and APIs. With its lightweight and fast performance, Node.js excels in building applications that deal with frequent I/O operations, like file uploads, database queries, and API requests, all without compromising performance. The use of JavaScript on both the client and server side also simplifies the development process, as developers can work with a single programming language across the entire stack.

- **Express.js**:

  Express.js is a web application framework for Node.js that simplifies the process of building robust and scalable web applications. As a minimal, flexible framework, Express provides a range of built-in functionalities to handle HTTP requests, define API routes, and manage middleware, making it an essential tool for backend development. It simplifies tasks such as routing, request handling, and generating responses, allowing developers to focus more on application logic. Express also supports middlewares, which can be used to process requests, validate data, and handle authentication. With Express.. The simplicity and extensibility of Express make it a popular choice for building modern web servers and APIs

- **AWS SDK**:

  Amazon Web Services (AWS) from within their applications. For the Accel Dedup Project, the AWS SDK is primarily used to integrate with Amazon S3, a highly scalable and durable cloud storage service. Through the SDK, developers can programmatically upload, download, and manage files stored in S3 buckets. The SDK abstracts the complexities of interacting with the S3 API, providing an easy-to-use interface for managing files, setting access controls, and handling authentication. With AWS SDK, the Accedl Deduplication Project can securely upload user files to S3 and take advantage of S3's global scalability, ensuring that files are stored reliably and are accessible from anywhere. Additionally, it supports features such as versioning and file retrieval, providing flexibility in managing cloud-based file storage.

- **Multer**:

  Multer is Node.js middleware designed to manage application file uploads. It is based on the busboy library, which manages multipart file uploads and parses form input. Multer handles various content types (such as photos, PDFs, or documents) and saves them to the designated storage location (either in memory or on disc), making the process of uploading data from a client to a server easier. File size restrictions, file filtering (to permit only particular kinds of files), and destination routes for storing the uploaded files are just a few of the configuration choices it offers. Multer is used to handle user file uploads in the context of the Accedl Deduplication Project, making sure that files are appropriately processed, saved, and prepared for deduplication tests. Multer's integration with the backend

Frontend

- **React.js**:

  React.js is essential to the AccelDedup Project because it offers a quick, effective, and interactive user interface (UI) development experience. Smaller, reusable components, including file upload forms, file listings, and authentication sections, can be used to deconstruct the user interface (UI) using React's **component-based architecture**. With this modular approach, components may be changed or replaced separately without impacting the system as a whole, ensuring improved maintainability and scalability. Rendering is optimised by React's virtual DOM, guaranteeing smooth and effective user interface updates, including the status of file uploads or deduplication outcomes. It also manages state management monitoring uploaded files, deduplication results, and the user's authentication status to make sure the user interface (UI) adapts to changes in the data.

- **React Router**:

  The AccelDedup Project's incorporation of React Router, a popular tool for managing routing in React applications, is essential to improving the application's usability. In

React application React Router enables developers to specify several routes, each of which is connected to a distinct component or view. This is crucial for developing a smooth, single-page application (SPA) that allows users to switch between views, including bucket status, uploaded file list, and file upload, without requiring a full-page reload. A seamless navigating experience is made possible by React Router's ability to maintain URL pathways and dynamically render the necessary components based on the current path.

- **Axios**:

  In order to manage communication between the frontend (built with React.js) and the backend (powered by Node.js and Express), the Accedl Deduplication Project relies heavily on Axios, a well-known promise-based HTTP client for sending asynchronous requests from the frontend to external APIs. In order to facilitate communication with external services like the file upload API, the deduplication API, or even the cloud storage service (like AWS S3) for managing user files, Axios aims to streamline the process of sending HTTP requests and managing responses. Because it offers a reliable and simple method for working with JavaScript APIs, Axios is particularly useful in projects that involve a lot of API calls.

- **Material-UI/ Bootstrap**:

  The inclusion of Material-UI and Bootstrap, two well-known front-end frameworks for creating responsive, user-friendly, and visually appealing user interfaces, in the Accedl Deduplication Project offers a major benefit in terms of expediting the development process and guaranteeing a polished and consistent user interface. Buttons, forms, navigation bars, modals, and other pre-made, usable user interface elements are included in both frameworks and are simple to incorporate into an application. These elements guarantee that the user interface is consistent throughout the application's many pages and parts, in addition to assisting in accelerating the development process.

4.1.2 IDEs used in the project:

The Integrated Development Environment (IDE) used for this project is Visual Studio Code (VS Code).

Reasons for Choosing Visual Studio Code:

- **Lightweight and Fast**: VS Code is known for its speed and efficiency, making it ideal for both frontend and backend development.
- **Extensibility**: With a wide range of extensions, such as those for Node.js, React.js, and AWS, VS Code supports a variety of programming languages and tools, enabling a productive development environment.
- **Integrated Terminal**: It allows developers to run commands directly from the terminal within the editor, streamlining the development workflow.
- **Debugging**: VS Code offers built-in debugging tools for both client-side and server-side code, making it easier to find and fix issues.
- **Version Control Integration**: It integrates seamlessly with Git, allowing for easy source control management, commit history, and code collaboration.

VS Code is chosen because of its lightweight nature, powerful features, and extensive plugin ecosystem that enhances development productivity across various stages of the project.

4.1.3 Development Tools

- **Git**: A distributed version control system for tracking changes in the project’s source code.
- **GitHub/GitLab**: Platforms for hosting repositories and collaborating with other developers.

These tools and technologies provide the foundation for the development of a robust, scalable, and secure cloud-based file upload system with deduplication capabilities. They support various phases of development, including backend logic, frontend user interface, cloud infrastructure management, testing, and version control.

4.2 Algorithm/pseudocode used

Objective: To get the current status of the cloud storage (e.g., available space, used space).

Pseudo code:

```
function getBucketStatus():
    # Step 1: Retrieve bucket usage statistics from AWS S3
    stats = s3.getBucketStatistics({
        Bucket: "your-bucket-name"
    })

    # Step 2: Return the bucket's usage statistics
    return stats
```

Explanation:

- **AWS S3 Stats**: The system queries AWS S3 for statistics related to the storage bucket (such as the amount of storage used and available space).
- **Return Stats**: The retrieved stats are returned to the user.

4.3 Testing Techniques

4.3.1. Unit Testing in the Context of the Project

In software development, unit testing is a crucial technique that entails independently evaluating discrete code pieces, such as components, functions, or methods, to make sure they perform as intended. Unit testing is used in both the frontend and backend components of the AccelDedup Project to verify that each component of the program functions as intended, finds defects early, and increases system stability.

To ensure that individual React components render successfully and behave as expected in various contexts, unit testing is done on the frontend using the React Testing Library. It integrates easily with the React Testing Library, a small testing tool for user-perspective testing of React components. A more thorough and user-centred approach to testing is ensured by the React Testing Library, which promotes testing the components in a manner that resembles how users would interact with them in a real environment, in contrast to typical shallow rendering.

Unit tests, for instance, could be created for the AccelDedup Project to examine how UI elements like forms (for uploading files), buttons (for initiating the deduplication process), and navigation elements (for ensuring that users can move fluidly between the various application sections) behave. A test could confirm that the submit button initiates the intended upload functionality, that the file list shows appropriately after the user has uploaded files, or that a file upload form renders correctly with the required input fields. Additionally, by mimicking user behaviours and making sure the proper error messages or feedback are displayed, edge cases like improper file formats or uploading issues can be tested.

4.3.2. Integration Testing in the Context of the Project

Integration testing ensures that different modules of the system work together as expected. This is particularly important for testing the interaction between the frontend and backend, as well as backend interactions with cloud storage and the database. API Integration

4.3.3 End-to-End (E2E) Testing E2E: To make sure the application works as intended from beginning to end, end-to-end (E2E) testing is an essential step in the testing process that mimics real-world situations and user activities. E2E testing verifies the application's overall flow, guaranteeing that all systems and components operate in unison, in contrast to unit tests, which concentrate on evaluating specific parts or features separately. Testing the entire user journey from logging in, uploading files, processing deduplication, and showing the user the results is known as E2E testing in the context of the AccelDedup Project. This procedure helps guarantee that the program satisfies end users' expectations and operates accurately in real-world scenarios.

4.3.4 Performance Testing Performance To make sure the AccelDedup Project can manage massive data volumes effectively without experiencing performance degradation, performance testing is a crucial component. In order to make sure the system can grow to accommodate user demands, particularly when handling many files being uploaded and processed at once this kind of testing examines how the system responds to different loads and pressures. In order to assess various facets of the system's performance, load testing and stress testing are the two main forms of performance testing that are employed in the project.

takes the testing a step further, however, by testing the system to its breaking point in order to determine the maximum capacity it can support. To evaluate how the system responds to tremendous stress, this kind of testing entails uploading an excessive amount of files or very huge files. Stress testing assists in identifying any performance issues, such as memory leaks, server failures, or delayed response times, that may compromise scalability or stability. Developers can identify the constraints of the deduplication engine and cloud storage interaction by testing the system's performance under stress. This ensures that the system can manage scenarios where the data load surpasses normal usage with grace. Understanding how the application responds to unforeseen spikes requires this testing.

4.3.5. Security Testing: To make sure that sensitive user data is handled safely and that the application is shielded from typical vulnerabilities, security testing is essential to the Accedl Deduplication Project. Authentication testing, which examines the strength of the application's authentication method, is a crucial component of security testing in this project. Only authorised users can access certain resources and carry out tasks, like uploading files or checking deduplication results, thanks to the project's usage of JWT-based authentication (JSON Web Tokens), a secure user verification technique. Before allowing access to the user's data, authentication testing makes sure that the JWT tokens are generated correctly, sent securely, and verified.

To guarantee the security of user sessions, a variety of situations are examined during authentication testing. Testing, for instance, entails making sure that access is refused and appropriate error messages are sent without disclosing private information in the event that a login or password is entered incorrectly. In order to confirm that the system appropriately rejects access in such circumstances, testers will also mimic attempts to get around authentication, such as by using counterfeit or expired tokens. To avoid unwanted access, the system's capacity to manage user sessions including token expiration, safe token storage, and invalidation upon logout is also examined. This extensive authentication testing helps guarantee that sensitive user data, including file uploads and personal information, is shielded from unwanted access and that the AccelDedup Project complies with security best practices.

4.4 Test Cases designed for the project work

Here are the test cases for each of the algorithms, including inputs and expected outputs:

Test Case 1: Verify File Existence in S3 Bucket

Test Case ID: TC001

Test Title: Verify that uploaded files are correctly listed in the S3 bucket. Preconditions: The user has successfully uploaded files using the web interface. The AWS S3 bucket is properly configured, and the user has access to it.

Test Steps: Navigate to the AWS S3 console and open the specific bucket (e.g., "inpbucket"). List the contents of the bucket.

Expected Results: Files such as download.jpg, html1.html, ques4.html, ques5.html, and Webfile.docx should be present in the bucket. 15 Each file should display the correct file name, type, and last modified timestamp (as seen in the image).

Validation Criteria: The files displayed on the AWS S3 console should match the files uploaded from the client interface. The "Last modified" timestamps should accurately reflect the upload times. The file types should be correctly identified (e.g., jpg, html, docx).

Test Case 2: Verify File Accessibility via URL

Test Case ID: TC002 Test Title: Verify that each file can be accessed using its URL. Preconditions: Files are successfully uploaded and listed in the S3 bucket.

Test Steps: Generate the public URL for a file (e.g., html1.html). Attempt to access the file by opening the URL in a web browser. Expected Results: The file should be accessible via its URL. The content of the file should load properly (e.g., HTML content should render if viewed in a browser).

Validation Criteria: 16 Files should load successfully using their URLs without returning errors like 404 (Not Found) or 403 (Access Denied).

Test Case 3: Duplicate File Upload

Test Case ID: TC003

Test Title: Verify that duplicate file uploads are handled correctly. Preconditions: Deduplication logic is implemented in the server to check for file content hash. Test Steps: Attempt to upload the same file (e.g., html1.html) that already exists in the S3 bucket. Expected Results: The server should identify the duplicate file by its hash. The system should display a message indicating that the file has already been uploaded. Validation Criteria: The system should prevent the re-upload of the same file content and notify the user appropriately.

Test Case 4: Unsupported File Type Upload

Test Case ID: TC004

Test Title: Verify that unsupported file types are rejected. Preconditions: 17 The server has file type restrictions configured (if applicable)

Test Steps: Attempt to upload a file with an unsupported type (e.g.,.exe or.bat). Expected Results: The server should reject the file upload and return an appropriate error message to the user. Validation Criteria: The file should not be uploaded, and an error should be displayed

The expected outputs align with the objectives of the algorithms, ensuring data is accurately processed and stored in the database. These tests provide confidence in the application's performance and its ability to handle different use cases effectively.

CHAPTER 5 RESULTS AND DISCUSSIONS

5.1 User Interface Representation (of Respective Project)

This section presents a visual representation of the user interface (UI) within the AI course generation platform. The UI plays a pivotal role in user interaction and experience. In this analysis, we showcase key UI components and their functionality, emphasizing the platform's design principles.

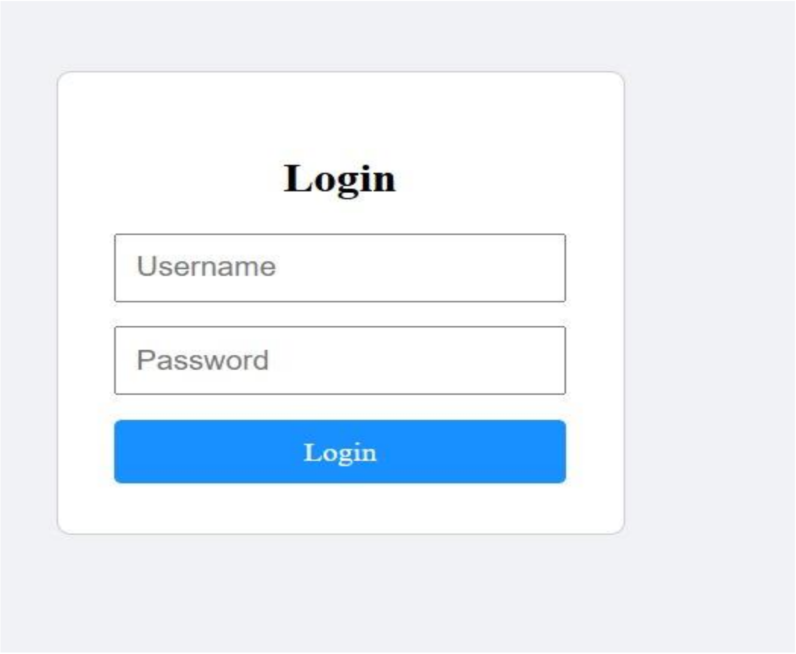


Fig 5.1(User Interface)

5.1.1 Brief Description of Various Modules of the system

The system is divided into several modules, each performing a specific function. Below is a brief description of each module:

Authentication Module

- **Purpose**: Handles user login, registration, and authentication.

- **Functionality**: Validates user credentials, manages user sessions, and ensures secure access to the system.

File Upload Module

- **Purpose**: Facilitates uploading files to the cloud storage system.
- **Functionality**: Accepts file uploads from the user, processes them using the Multer library, and stores them securely in cloud storage (AWS S3). It also provides feedback about the success or failure of the upload.

File Management Module

- **Purpose**: Manages uploaded files and performs deduplication.
- **Functionality**: Ensures that duplicate files are identified and removed from the cloud storage. It uses various algorithms to compare files and determine uniqueness. It also tracks metadata such as file names, sizes, and upload timestamps.

Bucket Status Module

- **Purpose**: Provides the status of the cloud storage bucket (AWS S3).
- **Functionality**: Fetches data related to the usage of the cloud storage, such as total space used, remaining space, and file count. It helps users monitor the storage capacity and performance of the system.

File Listing Module

- **Purpose**: Allows users to view and manage their uploaded files.
- **Functionality**: Displays a list of all uploaded files, supports searching for files by name, and provides links to view or download the files. It also provides information on file size and upload date.

Deduplication Module

- **Purpose**: Identifies and removes duplicate files from the storage system.
- **Functionality**: Compares uploaded files using hashing techniques and removes duplicates, ensuring that only unique files are stored, thus saving space.

Notification Module

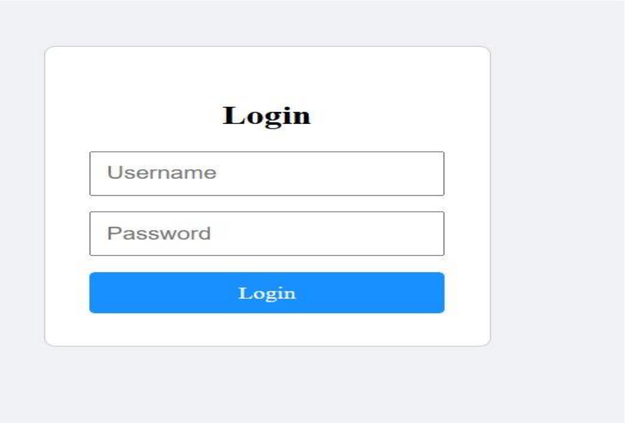
- **Purpose**: Notifies users about the status of their file uploads and any errors.
- **Functionality**: Sends real-time alerts or error messages about file uploads, system status, or issues related to file deduplication.

User Interface Module

- **Purpose**: Provides a user-friendly interface for interaction.
- **Functionality**: Displays all the aforementioned features in an intuitive manner. Ensures that the user experience is smooth, with clear buttons, forms, and messages for easier navigation.

5.2 Snapshots of system with brief detail of each and discussion

5.2.1 Login Page



Login

Username

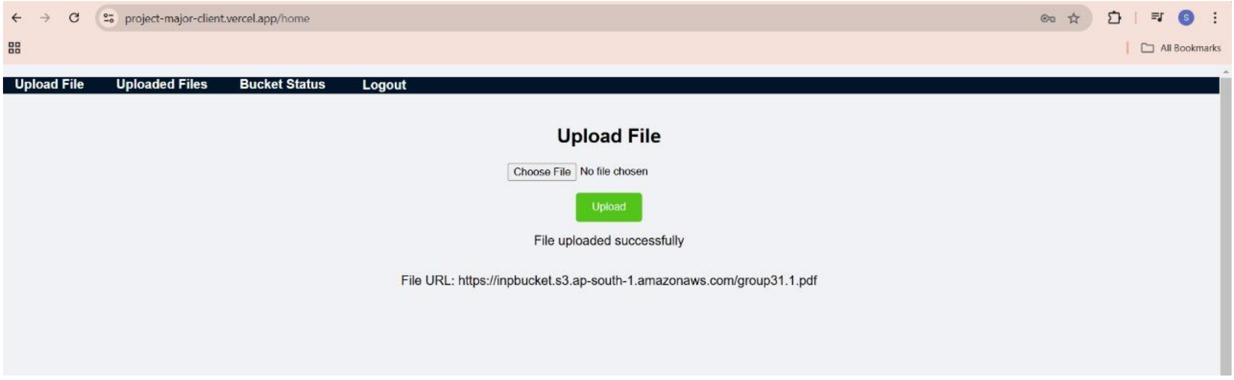
Password

Fig 5.2(Login Page)

The login feature in AccelDedup allows for safe access to user accounts for file management in the cloud. Users are required to enter their login information, usually a username (or email address) and password, when they first use the application. The backend checks these credentials against the MongoDB database when the login request is transmitted to the server. Because MongoDB saves hashed passwords for security purposes, the user-entered password is hashed and compared to the hash that is saved. To manage the user session and ensure safe access to the application, a session token—typically a JSON Web Token (JWT)—is generated upon authentication.

5.2.2 Uploading a file

In AccelDedup, the file upload functionality allows users to securely store files in the cloud. When a user uploads a file, the application first checks if a file with the same content already exists to prevent duplication. This is achieved by generating a unique hash for each file and comparing it with existing hashes in the database. If the file is unique, it's uploaded to AWS S3, where it's stored and managed. The file metadata, including its hash and location, is then saved in MongoDB for easy access and future verification. This approach ensures efficient storage by avoiding duplicate files and allowing users to manage their cloud-stored data effectively.



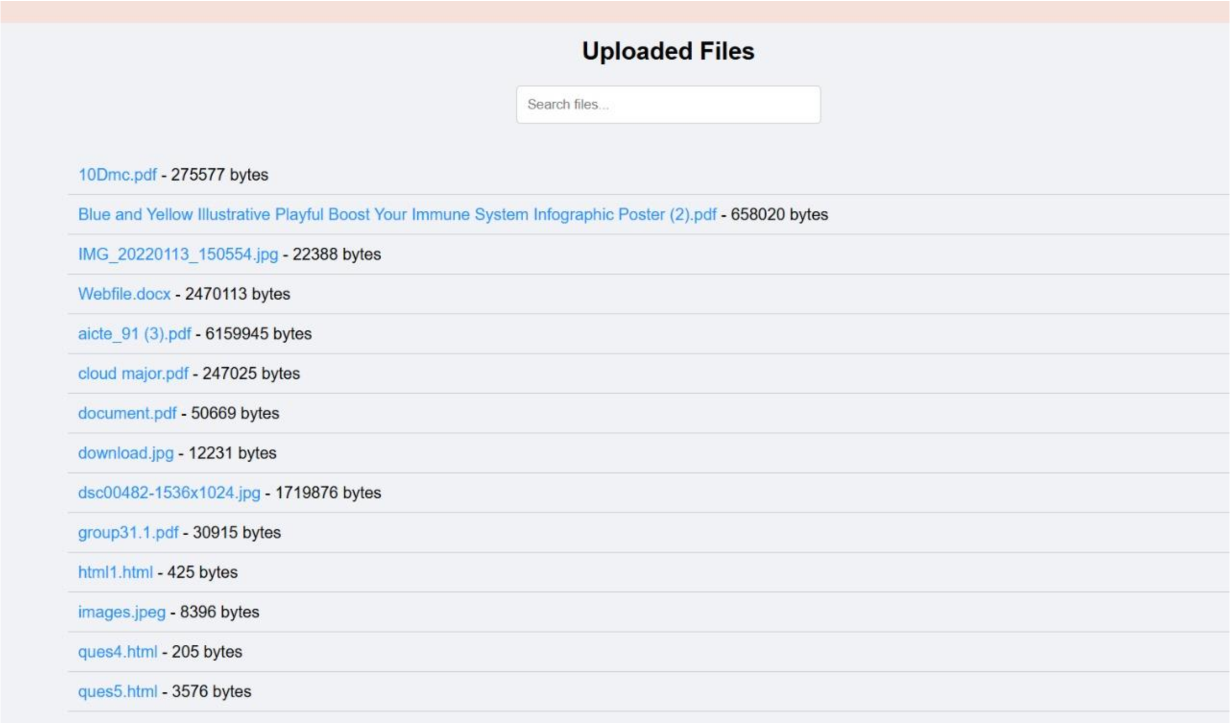
File uploaded successfully

File URL: https://inpbucket.s3.ap-south-1.amazonaws.com/group31.1.pdf

Fig 5.3(Uploading a File)

5.2.3 Uploaded Files

In AccelDedup, uploaded files are stored securely in AWS S3. When a file is successfully uploaded, its metadata—such as file name, size, unique hash, and storage location—is saved in MongoDB. This metadata helps the system keep track of each file and enables features like deduplication by preventing the same content from being stored multiple times. Users can easily access and manage these uploaded files through the application, leveraging efficient storage management to ensure each file in the cloud is unique.



Uploaded Files

10Dmc.pdf - 275577 bytes

Blue and Yellow Illustrative Playful Boost Your Immune System Infographic Poster (2).pdf - 658020 bytes

IMG_20220113_150554.jpg - 22388 bytes

Webfile.docx - 2470113 bytes

aicte_91 (3).pdf - 6159945 bytes

cloud major.pdf - 247025 bytes

document.pdf - 50669 bytes

download.jpg - 12231 bytes

dsc00482-1536x1024.jpg - 1719876 bytes

group31.1.pdf - 30915 bytes

html1.html - 425 bytes

images.jpeg - 8396 bytes

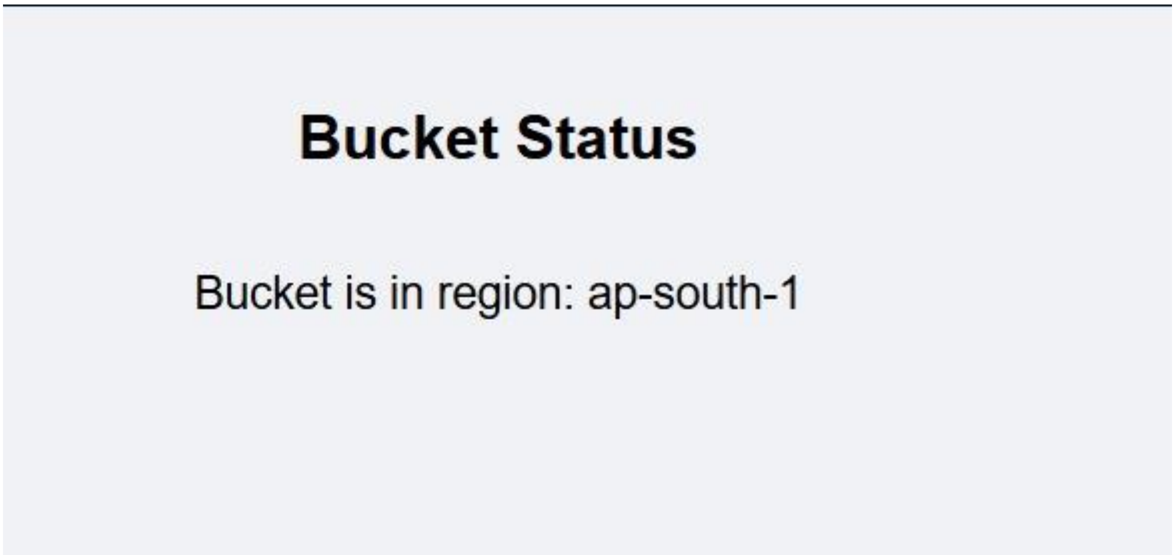
ques4.html - 205 bytes

ques5.html - 3576 bytes

Fig 5.4(Uploaded Files)

5.2.4 Bucket Status

In AccelDedup, AWS S3 buckets serve as the primary storage location for all uploaded files. Each bucket provides a secure, scalable solution to store and manage files in the cloud, ensuring data durability and availability. When a user uploads a file, it is stored in an S3 bucket, and its metadata (like file name, size, and unique hash) is saved in MongoDB to enable efficient file management and deduplication. AWS S3 also offers features like versioning, lifecycle management, and access controls, which help AccelDedup efficiently organize, protect, and manage stored data.



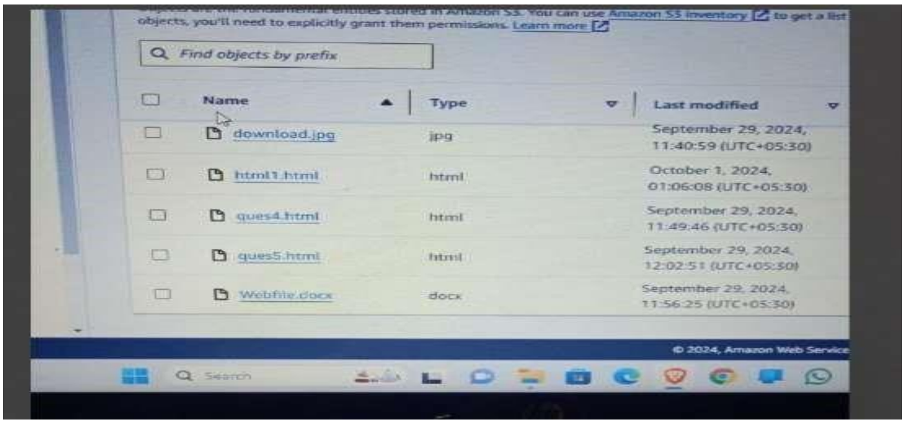
Bucket Status

Bucket is in region: ap-south-1

Fig 5.5(Bucket Status)

5.2.5 Aws S3 Bucket Status

In AccelDedup, AWS S3 buckets serve as the primary storage location for all uploaded files. Each bucket provides a secure, scalable solution to store and manage files in the cloud, ensuring data durability and availability. When a user uploads a file, it is stored in an S3 bucket, and its metadata (like file name, size, and unique hash) is saved in MongoDB to enable efficient file management and deduplication. AWS S3 also offers features like versioning, lifecycle management, and access controls, which help AccelDedup efficiently organize, protect, and manage stored data.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list
objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified
□	download.jpg	jpg	September 29, 2024, 11:40:59 (UTC+05:30)
□	html1.html	html	October 1, 2024, 01:06:08 (UTC+05:30)
□	ques4.html	html	September 29, 2024, 11:49:46 (UTC+05:30)
□	ques5.html	html	September 29, 2024, 12:02:51 (UTC+05:30)
□	Webfile.docx	docx	September 29, 2024, 11:56:25 (UTC+05:30)

Fig 5.6 (Aws S3 Bucket Status)

CHAPTER 6 CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

The deduplication-enabled cloud-based file upload and management system offers a reliable and effective way to handle massive data sets. AWS S3, Multer, React.js, and Node.js are just a few of the technologies that are integrated to make the system scalable, safe, and easy to use.

We addressed important issues like security, performance, and maintainability while concentrating on producing a flawless user experience throughout the development process. Utilising AWS services for file management and cloud storage guarantees that the system can manage big datasets and give users dependable file access.

Additionally, by incorporating deduplication features, the system helps reduce unnecessary storage usage and ensures that users do not face redundancy issues, improving overall system efficiency and reducing operational costs.

The project also demonstrates the value of implementing contemporary development techniques, such as version control with Git, API testing with Postman, and component-based architecture with React.js. These procedures help to build a system that is organised and manageable.

Overall, this project achieves its objective of providing a functional and efficient file management system with integrated deduplication, positioning it as a valuable solution for handling large-scale file uploads and ensuring optimal use of storage resources.

6.2 Future Scope:

While the current system successfully addresses file management and deduplication, there are several opportunities for further enhancement and expansion. Some potential areas for future development include:

6.2.1AI-basedDeduplication

To further optimise storage and provide more precise deduplication, machine learning or artificial intelligence (AI) algorithms that detect near duplicates may be based on file content analysis rather than merely hashing.

6.2.2Multi-cloudIntegration:

Adding support for different cloud providers (such as Google Cloud and Azure) to the system will improve scalability and redundancy while allowing customers to select the cloud storage option that best suits their needs.

6.2.30FileVersioning:

introducing file version control, which improves tracking and rollback capabilities by enabling users to keep several versions of a file.

6.2.4MobileApplication:

creating an iOS and Android mobile version of the system that will allow users to upload, manage, and access their files from any device.

6.2.5EnhancedSecurityFeatures:

For increased security and compliance, multi-factor authentication (MFA), enhanced auditing and monitoring capabilities, and the use of sophisticated encryption techniques for file storage are all recommended.

6.2.6BulkFileUploadandDownload:

Implementing bulk file upload and download functionality to streamline the process of managing large datasets is especially useful for enterprises with extensive file repositories.

6.2.7Real-timeCollaboration:

The addition of real-time collaboration facilities, which enable users to share and work on files simultaneously, further enhances the system's usefulness for team and business contexts.

6.2.8PerformanceOptimisations:

Continuously improving system performance, especially for high-volume file uploads, downloads, and file searching, by optimising backend infrastructure and incorporating caching mechanisms.

References

1]Author(s), "De-Duplication Over Cloud Data to Enhance the Storage Systems," JP Infotech, [Online]. Available: https://jpinfotech.org/de-duplication-over-cloud-data-toenhance the-storage-systems/

[2] Author(s), " De-Duplication Over Cloud Data to Enhance the Storage Systems " International Arab Journal of Information Technology, vol.16, no. 5,Sep. 2019. [Online]. Available: https://iajit.org/portal/PDF/September%202019,%20No.%205/15822.pdf.

[3]"Data Deduplication," Data Intell, Feb. 2023. [Online]. https://dataintell.io/2023/02/data-deduplication/. Available: https://dataintell.io/2023/02/data-deduplication/

[4] Author(s), " De-Duplication of Data in Cloud Storage," International Journal of Advanced Networking and https://www.ijana.in/papers/84.pdf.