# VeriDedup: A Verifiable Cloud Data Deduplication Scheme With Integrity and Duplication Proof

*A Project report submitted to*
*Jawaharlal Nehru Technological University, Kakinada,*
*In the partial fulfillment for the award of the Degree in*

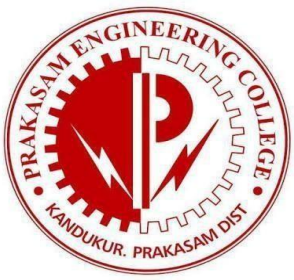
**BACHELOR OF TECHNOLOGY**
**IN**
**COMPUTER SCIENCE&ENGINEERING**
**(ARTIFICIAL INTELLIGENCE)**

*Submitted by*


| | |
|---|---|
| **Nagireddy Bala Brahmareddy** | **21F91A4311** |
| **Nagiri Akash** | **21F91A4302** |
| **Nalagatla Venkata Leela Krishna** | **21F91A4358** |
| **Chinnam Gowtham Sri Krishna** | **21F91A4319** |


**Under the Noble Guidance of**

**Mr.K. MOHANA RAO** M.Tech(Ph.D)





**PRAKASAM ENGINEERING COLLEGE**

***(An ISO 9001-2008 & NAAC Accredited Institution)***

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

O.V.ROAD, KANDUKUR-523105,A.P.


**2021-2025**

# PRAKASAM ENGINEERING COLLEGE

*(An ISO 9001-2008 & NAAC Accredited Institution)*

**(Affiliated to Jawaharlal Nehru Technological University, Kakinada)**

**O.V.ROAD, KANDUKUR-523105,A.P.**

## DEPARTMENT OF

## COMPUTER SCIENCE & ENGINEERING

## (ARTIFICIAL INTELLIGENCE)

## BONAFIDE CERTIFICATE

*This is to certify that the project report entitled "**VERIDEDUP: A VERIFIABLE CLOUD DATA DEDUPLICATION SCHEME WITH INTEGRITY AND DUPLICATION PROOF**"* is a bonafide work of NAGIREDDY BALA BRAHMARERDDY (21F91A4311), NAGIRI AKASH (21F91A4302), CHINNAM GOWTHAM SRI KRISHNA (21F91A4319), NALAGATLA VENKATA LEELA KRISHNA (21F91A4358) *,in the partial fulfillment of the requirement for the award of the degree in* **Bachelor of Technology** *in* **COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE)** *for the academic year 2021-2025. This work is done under my supervision.*

**Signature of the Guide**

**Mr. K. MOHANA RAO** M.Tech.(Ph.D)

**Signature of the HOD**

**DR. K. SUBBA REDDY** M.Tech. Ph.D

**Signature of External Examiner**

# DECLARATION

We do here by declare that the seminar report entitled ***"VERIDEDUP: A VERIFIABLE CLOUD DATA DEDUPLICATION SCHEME WITH INTEGRITY AND DUPLICATION PROOF"*** is a genuine work carried out by us under the guidance of **Mr. K. MOHANA RAO** $_{\text{M.Tech,(ph.D)}}$in partial fulfillment for the award ofthe degree of **"Bachelor of Technology in Computer Science and Engineering (Artifical Intelligence)"** of **Jawaharlal Nehru Technological University, Kakinada.**

| | |
|---|---|
| **Nagireddy Bala Brahmareddy** | **21F91A4311** |
| **Nagiri Akash** | **21F91A4302** |
| **Nalagatla Venkata Leela Krishna** | **21F91A4358** |
| **Chinnam Gowtham Sri Krishna** | **21F91A4319** |

# ACKNOWLEDGEMENT

We feel to render my thankful acknowledgement to the following distinguished personalities, who stretched their helping hand to us, in completing my mini project work.

Weareverygratefulandoursincerethankstooursecretary&correspondent **Dr. K. RAMAIAH** of **PRAKASAM ENGINEERING COLLEGE** for giving this opportunity.

We hereby, express my regards and extend my gratitude to our PRINCIPAL, **Dr.CH. RAVIKUMAR**, for giving this opportunity to do the the sis as a part of our course.

We express my deep sense of gratitude to **DR. K. SUBBA REDDY** **M.Tech, Ph.D,** Head of the Department, **Department of CSE-AI** for having shown keen interest at every stage of development of our thesis and guiding us in every aspect.

And We are thankful to our guide **MR. K. MOHANA RAO** **M.Tech(Ph.D),**who has channeled our thoughts and timely suggestions.

We would also like to thank all our Faculties in Prakasam Engineering College for their constant encouragement and for being a great group of knowledgeable and cooperative people to work with.

| | |
|---|---|
| **Nagireddy Bala Brahmareddy** | **21F91A4311** |
| **Nagiri Akash** | **21F91A4302** |
| **Nalagatla Venkata Leela Krishna** | **21F91A4358** |
| **Chinnam Gowtham Sri Krishna** | **21F91A4319** |

# INDEX

# List Of Figures :

# List Of Tables :

# Abstract

VeriDedup is a secure cloud data deduplication system designed to optimize storage efficiency while ensuring data integrity, security, and privacy. Traditional cloud storage solutions often suffer from redundant data storage, leading to unnecessary consumption of resources and increased costs. However, conventional deduplication mechanisms pose security risks, including data integrity compromise, unauthorized access, and manipulation by Cloud Service Providers (CSPs). VeriDedup addresses these challenges by integrating Trusted Data Integrity Check Protocol (TDICP) and User-Driven Deduplication Check Protocol (UDDCP) to provide a verifiable and transparent deduplication process.

TDICP enables multiple data holders to verify file integrity independently using cryptographic verification tags, eliminating the need for direct interaction with the data owner. UDDCP introduces a challenge-response mechanism that allows the user—rather than the CSP—to determine whether a file is a duplicate, ensuring fair and accurate deduplication validation. Additionally, VeriDedup leverages AES encryption, authentication mechanisms, and auditing logs to enhance security, preventing unauthorized data access and manipulation.

The system is implemented using Java (JSP, Servlets), MySQL (WAMP Server 3.4), Apache Tomcat 9.0, and AES encryption, ensuring compatibility with modern cloud storage environments. Performance analysis and simulations demonstrate that VeriDedup offers high efficiency, scalability, and security compared to existing deduplication solutions. This project establishes a trustworthy and privacy-preserving deduplication framework, making it a robust solution for secure cloud storage management.

# CHAPTER – I

# INTRODUCTION

Cloud computing has emerged as one of the most transformative advancements in modern information technology, enabling seamless access to vast computing and storage resources over the internet. By shifting data storage and processing to cloud service providers (CSPs), organizations and individuals can avoid the costs associated with maintaining dedicated infrastructure while benefiting from high availability, scalability, and remote accessibility. Among the various services offered by cloud computing, cloud storage remains the most widely used due to its ability to handle large-scale data efficiently. The demand for cloud storage continues to rise, driven by the increasing volume of digital data generated by businesses, research institutions, and individual users.

As the world transitions toward a data-driven ecosystem, the amount of digital data stored in cloud environments is growing at an unprecedented rate. Organizations and individuals frequently store files such as documents, multimedia, backups, and application data in the cloud to ensure availability and security. However, this rapid increase in data storage presents a significant challenge—storage inefficiency caused by redundant or duplicate data. When multiple users upload identical files or files containing the same data blocks, unnecessary storage consumption occurs, leading to inefficiencies in cloud storage infrastructure. This redundancy not only increases storage costs but also results in additional bandwidth consumption, longer retrieval times, and increased energy usage by cloud data centers.

To address the problem of redundant storage, cloud service providers implement data deduplication, an optimization technique that eliminates duplicate copies of data, ensuring that only a single instance of a file is stored while subsequent identical uploads are linked to the existing stored copy. Deduplication significantly improves storage efficiency, reducing storage costs, improving data retrieval performance, and minimizing the bandwidth required for file transfers. It plays a crucial role in large-scale cloud storage architectures, particularly in enterprise data centers, backup systems, and cloud-based content delivery networks.

However, despite its advantages, deduplication introduces several critical security and trust challenges that must be addressed before it can be adopted as a reliable and transparent mechanism for cloud storage optimization. Traditional deduplication mechanisms rely on the assumption that the CSP is honest and will correctly manage stored data without unauthorized modifications or misrepresentation of storage records. This assumption is problematic because users have no direct control over the deduplication process, nor can they independently verify whether the CSP is performing deduplication correctly. This lack of transparency raises significant concerns related to data security, integrity, and fairness in cloud storage billing models.

## Security and Trust Issues in Deduplication-Based Cloud Storage :

Although deduplication provides storage efficiency, it also creates security vulnerabilities that can compromise the reliability of cloud storage systems. Several key issues arise when applying deduplication to cloud storage environments.

The first major issue is the integrity risk associated with deduplicated data. Since deduplication ensures that multiple users reference the same stored file, any modification, corruption, or deletion of that file affects all associated users. A semi-trusted CSP may tamper with the stored data for financial or operational reasons, such as reclaiming storage space by deleting files without user consent. This raises the need for an integrity verification mechanism that allows users to confirm that their data remains unaltered even after deduplication has been applied. Traditional Proof of Retrievability (PoR) schemes have been proposed to allow users to verify the integrity of their stored files without downloading them. However, PoR schemes are designed for environments where each user stores an independent copy of a file. In a deduplication-based system, different users storing the same file would generate different integrity verification tags due to the use of unique cryptographic keys. These varying verification tags prevent deduplication from being effectively applied to integrity metadata, making conventional PoR schemes incompatible with deduplication-based storage.

Another major issue is the potential for CSP manipulation in the deduplication verification process. Since deduplication operates on the assumption that the CSP correctly identifies and eliminates duplicate files, users must trust that the CSP will accurately determine whether an uploaded file is already stored. However, CSPs have financial incentives to misrepresent deduplication outcomes. For instance, a CSP can falsely claim that a file is unique, forcing users to upload redundant copies and pay for additional storage that they do not actually require. Similarly, CSPs may fail to provide storage discounts for deduplicated files, preventing users from benefiting from cost savings. Since users do not have direct access to the cloud provider's storage infrastructure, they cannot independently verify whether their uploaded file already exists, making them vulnerable to unfair billing practices.

A third critical challenge is the lack of user-controlled verification mechanisms. In most existing deduplication systems, users have no way of validating whether their file has been properly deduplicated or ensuring that it has not been tampered with. Since deduplication is managed entirely by the CSP, users must rely on the CSP's internal processes to determine whether a file is already present in the cloud. This creates an imbalance of control, where users are forced to trust the CSP's claims regarding file duplication and integrity without any means of verification.

Additionally, deduplication systems are vulnerable to brute-force attacks and data leakage. Since deduplication allows CSPs to compare newly uploaded files against stored files to determine whether they already exist, attackers can exploit this process to infer information about stored data. For example, an attacker could systematically

upload known files to determine whether a particular file exists in the cloud. This type of attack, known as a dictionary attack, can be used to extract sensitive information about files stored by other users. Traditional deduplication schemes lack privacy-preserving mechanisms to prevent such attacks, making them susceptible to information leakage.

## VeriDedup: A Verifiable Cloud Data Deduplication Scheme With Integrity and Duplication Proof :

To address these security and trust challenges, we propose VeriDedup, a novel verifiable cloud deduplication framework designed to enhance the transparency, security, and reliability of deduplication-based cloud storage systems. The title of our project, “VeriDedup: A Verifiable Cloud Data Deduplication Scheme With Integrity and Duplication Proof,” accurately represents the key contributions and objectives of this research. VeriDedup introduces two novel cryptographic protocols that enable users to verify the integrity of deduplicated files and independently check the correctness of duplication verification.

The first protocol, called the Tag-Flexible Deduplication-Supported Integrity Check Protocol (TDICP), is designed to enable integrity verification for deduplicated files. Unlike traditional PoR schemes, TDICP allows users to generate unique verification tags while ensuring that these tags can still be deduplicated at the CSP. This allows multiple users to verify the integrity of a shared file without requiring the CSP to store separate verification metadata for each user. The protocol uses Private Information Retrieval (PIR) techniques to insert cryptographic "note sets" into files, allowing users to verify data integrity while maintaining storage efficiency.

The second protocol, called the User Determined Duplication Check Protocol (UDDCP), prevents CSPs from misrepresenting deduplication verification results. Instead of relying on the CSP to determine whether a file already exists, UDDCP allows users to independently verify file duplication using Private Set Intersection (PSI) techniques. This ensures that CSPs cannot falsely claim that a file is unique, preventing unfair storage charges and ensuring a transparent deduplication process.

VeriDedup addresses the limitations of existing deduplication schemes by introducing a secure, verifiable, and user-controlled deduplication framework. Unlike previous approaches that assume CSP honesty, VeriDedup provides cryptographic proof mechanisms that allow users to validate both data integrity and deduplication correctness. By enabling secure storage optimization while preserving user trust and security, VeriDedup represents a significant advancement in cloud storage security research.

The increasing adoption of cloud storage and deduplication highlights the need for robust security mechanisms that ensure transparency, integrity, and fairness in storage management. VeriDedup introduces a new paradigm for verifiable deduplication by integrating cryptographic techniques that allow users to verify data integrity and

duplication correctness. By addressing key security challenges such as integrity verification, CSP trust issues, and brute-force attack vulnerabilities, VeriDedup provides a secure and efficient solution for next-generation cloud storage systems. The proposed framework sets a new standard for secure and trustworthy deduplication, ensuring that users can benefit from storage optimization without compromising security or control over their data.

## 1.1 Background and Problem Statement

### Background :

With the rapid growth of digital data, cloud storage has become an essential solution for storing large amounts of information. However, duplicate data storage is a major challenge, as multiple users may upload identical files or files containing similar data blocks. This results in wasted storage space and increased costs for both cloud providers and users.

To address this issue, data deduplication is used to store only one copy of identical data while allowing multiple users to access it. While this method improves storage efficiency, it also raises security and integrity concerns. Some of the major risks include:

- Cloud service providers (CSPs) may tamper with or delete stored data.
- CSPs may provide false duplication checks and charge users for duplicate data that is stored only once.
- Users lack control over verifying whether their data has been stored securely and deduplicated properly.

VeriDedup is designed as a secure and verifiable cloud data deduplication system. It ensures that data is stored efficiently while maintaining security, integrity, and trustworthiness. The project is implemented using JSP, Java, Tomcat, and WAMP.

### Problem Statement :

Current cloud-based deduplication systems face several key challenges:

1. **Storage Waste and Inefficiency**
    - Traditional cloud storage systems store multiple copies of the same file, leading to unnecessary storage consumption.
    - Deduplication helps reduce storage needs, but it does not provide security or verification mechanisms.
2. **Lack of Trust in Cloud Service Providers (CSPs)**
    - A CSP may modify, tamper with, or delete user data.

- Users have no independent way to verify if their data has been deduplicated correctly or if they are being overcharged for storage.

3. **Integrity Verification Issues**
    - Existing verification methods only check if the data is retrievable but do not confirm if deduplication was performed correctly.
    - Some encryption methods used in deduplication are vulnerable to brute-force attacks.
4. **Security and Privacy Risks**
    - Weak encryption methods can lead to unauthorized access to stored files.
    - Users do not have control over how their files are deduplicated and stored.

### VeriDedup Solution :

VeriDedup addresses these challenges by implementing secure deduplication with verifiable integrity checks using:

- **JSP, Java, Tomcat, and WAMP** to ensure high performance and smooth integration.
- **User-controlled verification** to allow independent duplication and integrity checks.
- **AES encryption and Proxy Re-Encryption (PRE)** to secure file storage and access.
- **Flexible verification tags** to prevent brute-force attacks.
- **User-determined duplication checks** to prevent CSPs from providing false duplication results.

VeriDedup enhances security, integrity, and efficiency in cloud data deduplication, ensuring trusted storage and cost-effective cloud solutions.

## 1.2 Project Objectives and Scope

### Project Objectives :

The VeriDedup project aims to develop a secure, verifiable, and efficient cloud data deduplication system using JSP, Java, Tomcat, and WAMP. The main objectives are:

1. **Secure Data Deduplication**
    - Eliminate redundant data while ensuring security and integrity.

- Prevent unauthorized access using encryption techniques.

2. **Integrity Verification**
    - Allow users to verify the integrity of their stored files without depending on the cloud service provider.
    - Implement verification mechanisms to prevent data tampering.
3. **Correctness of Deduplication Check**
    - Ensure that the cloud service provider correctly performs deduplication and does not charge for redundant storage.
    - Implement user-controlled duplication checks to prevent false claims by the cloud service provider.
4. **Enhanced Security and Authentication**
    - Use AES encryption and Proxy Re-Encryption (PRE) to secure stored data.
    - Implement role-based access control (RBAC) for different users.
5. **Optimized Cloud Storage and Performance**
    - Reduce cloud storage costs by efficiently managing deduplicated data.
    - Maintain high performance with minimal computational overhead.

## Project Scope :

The VeriDedup system is designed as a scalable, secure, and verifiable cloud storage solution with the following scope:

1. **Cloud-Based Secure Deduplication**
    - Supports storage on local servers using WAMP and can be expanded to cloud environments.
    - Ensures efficient file deduplication while maintaining data privacy.
2. **Multi-User Environment**
    - Supports multiple clients and administrators with role-based authentication.
    - Implements secure user management for controlled access.
3. **Integrity and Verification Mechanisms**
    - Ensures data integrity through user-verifiable checks.

- Uses proof-of-ownership mechanisms to prevent unauthorized file access.

4. **Efficient and Scalable Architecture**
    - Developed using JSP, Java, Tomcat, and WAMP for high compatibility and performance.
    - Uses MySQL for database management and REST APIs for module communication.
5. **Advanced Security Features**
    - Implements AES encryption for secure file storage.
    - Uses Proxy Re-Encryption (PRE) for secure file sharing.
    - Integrates deduplication verification mechanisms to prevent false storage claims.
6. **Future Enhancements**
    - Support for cross-server deduplication in multi-cloud environments.
    - Implementation of machine learning for optimized deduplication.

The VeriDedup system ensures secure, efficient, and verifiable data deduplication while providing user-controlled security and integrity checks.

# CHAPTER - II

# LITERATURE SURVEY

## Introduction :

Cloud computing has revolutionized data storage by offering scalable and cost-effective solutions. However, as data volumes grow, storage efficiency and security concerns have become critical challenges. Deduplication is a widely adopted technique to optimize cloud storage by eliminating redundant data, reducing storage space, and minimizing bandwidth usage. Despite its advantages, existing deduplication systems face severe security limitations, including integrity risks, verification challenges, CSP manipulation, and weak access control mechanisms.

This literature survey provides an in-depth analysis of previous research and existing deduplication techniques in cloud storage, highlighting their strengths and limitations. Furthermore, it explores how VeriDedup overcomes these challenges by introducing verifiable integrity checks, user-controlled duplication verification, and cryptographic security mechanisms.

## Existing Deduplication Techniques and Their Limitations :

Several research efforts have attempted to secure deduplication while maintaining efficiency. However, most existing solutions either lack verifiable duplication checks or fail to provide strong security guarantees. Below is a detailed review of major existing systems and their limitations.

1. Compact Proof of Retrievability (PoR) with Deduplication
    - Proposed By: Shacham and Waters
    - Key Idea: Introduced Compact PoR using erasure coding, BLS signatures, and Message Authentication Codes (MACs) for integrity verification.
    - Limitations:
        - Computationally expensive due to the complexity of generating authentication tags.
        - Verification metadata grows linearly with the number of file blocks, leading to scalability issues.
        - Does not support deduplication over verification tags, making it unsuitable for cloud-based storage optimization.
2. Polynomial Commitment-Based Deduplication
    - Proposed By: Xu and Chang

- Key Idea: Used polynomial commitments to enhance deduplication while reducing communication costs.
- Limitations:
  - While communication costs are lower, the approach still fails to integrate deduplication into the integrity verification process.
  - CSPs can still manipulate deduplication results, making the system vulnerable to unfair storage charges.

3. StealthGuard: PIR-Based Deduplication with Integrity Verification
   - Proposed By: Azraoui et al.
   - Key Idea: Utilized Private Information Retrieval (PIR) with Word Search (WS) techniques to generate watchdogs (lightweight integrity verification tags).
   - Limitations:
     - Reduced storage overhead compared to previous PoR techniques but failed to support deduplication over verification tags.
     - Watchdog generation is efficient, but the method cannot prevent CSPs from misrepresenting deduplication results.
4. Publicly Verifiable Proof of Storage (PoS) with Deduplication
   - Proposed By: Zheng et al.
   - Key Idea: Enabled public verification of deduplicated files using the first uploader's cryptographic key.
   - Limitations:
     - Proven insecure under weak key attacks, allowing adversaries to compromise verification mechanisms.
     - Users cannot independently verify whether the CSP is falsely claiming that a file is unique.
     - No protection against CSP manipulation of storage billing models.
5. Message-Locked PoR with Deduplication
   - Proposed By: Vasilopoulos et al.
   - Key Idea: Integrated PoR mechanisms with message-locked encryption for deduplication.

- Limitations:
  - Enforces that all users generate identical verification tags for the same file, improving security but hindering deduplication efficiency.
  - Still relies on CSP trust for proper storage management.

## General Drawbacks of Existing Systems

After synthesizing research findings from multiple deduplication schemes, the following common limitations were identified:

1. Centralized Deduplication Storage Risks
   - Most systems perform deduplication on a single server, creating a single point of failure.
   - If the server experiences data loss, corruption, or compromise, all deduplicated users suffer data loss.
2. Limited Integrity Verification for Deduplicated Data
   - Existing systems focus on PoR techniques that assume separate copies of data, making them incompatible with deduplication.
   - Since deduplication demands a single verification tag per file, traditional PoR schemes cannot verify shared data integrity.
3. CSP Manipulation in Deduplication Verification
   - Users rely on the CSP's internal mechanisms to determine whether a file is unique.
   - CSPs can falsely claim a file is unique to charge extra storage fees, as users cannot verify duplication results.
4. Compromised User Privacy and Security
   - Deduplication mechanisms expose user data to brute-force attacks, allowing attackers to upload known files and infer whether specific files exist in the cloud.
   - Message-locked encryption does not fully protect against replay attacks, leading to data exposure risks.
5. Limited Access Control in Deduplicated Cloud Storage
   - Traditional deduplication techniques do not provide fine-grained access control mechanisms.
   - Data owners cannot control who accesses deduplicated files, raising concerns about data privacy.

## VeriDedup: A Secure and Verifiable Deduplication Framework

To overcome these limitations, VeriDedup introduces a next-generation deduplication framework that integrates cryptographic integrity verification and user-driven duplication checks.

## Key Innovations in VeriDedup

1. Tag-Flexible Deduplication-Supported Integrity Check Protocol (TDICP):
    - Enables verifiable integrity checks for deduplicated data using Private Information Retrieval (PIR).
    - Users can generate independent verification tags without breaking deduplication compatibility.
    - Eliminates the scalability issue of PoR schemes by allowing deduplicated integrity verification.
2. User Determined Duplication Check Protocol (UDDCP):
    - Prevents CSPs from falsely claiming that a file is unique using Private Set Intersection (PSI).
    - Empowers users to verify duplication results independently, eliminating CSP manipulation risks.
    - Ensures that users only pay for actual storage consumption, preventing unfair billing practices.
3. Enhanced Security and Privacy Mechanisms:
    - Prevents brute-force attacks by ensuring deduplication does not expose sensitive metadata.
    - Protects against message-locked encryption vulnerabilities, preventing replay attacks.
    - Ensures that each data owner has access control over deduplicated files.
4. Distributed Deduplication Storage Model:
    - Unlike existing centralized deduplication systems, VeriDedup spreads deduplicated data across multiple storage nodes, reducing single points of failure.
    - Increases fault tolerance and ensures high data availability in cloud environments.

Existing deduplication techniques in cloud storage focus on storage efficiency but neglect critical security aspects, including verifiable integrity, independent duplication

checks, and fine-grained access control. Most PoR schemes fail to support deduplication, and CSPs can manipulate verification processes to overcharge users.

VeriDedup overcomes these limitations by introducing a secure and verifiable deduplication scheme, ensuring that:

- Users can verify data integrity even after deduplication.
- CSPs cannot cheat users by falsely claiming a file is unique.
- Brute-force attacks and metadata leakage are mitigated through cryptographic security mechanisms.

By integrating TDICP and UDDCP, VeriDedup provides a robust, scalable, and privacy-preserving cloud storage solution, setting a new standard for secure deduplication-based cloud storage.

## 2.2 Real-Time Incidents of Malicious Data Duplication by Cloud Providers with VeriDedup-Based Solutions :

1. **Apple iCloud Antitrust Lawsuits & VeriDedup Solution**
**Issue:**

- In November 2024, a UK consumer group filed a £3.6 billion lawsuit against Apple for iCloud storage overcharging.
- Automated backups and forced uploads caused users to consume more storage than necessary, leading to increased subscription costs.
- Users were not given the option to control file duplication, resulting in unwanted storage purchases.

**VeriDedup Solution:**

- Proof-Based Deduplication (PBD) ensures that redundant files are removed before uploading, avoiding unnecessary cloud consumption.
- Integrity & Duplication Proof allows users to cryptographically verify whether their files are genuinely unique or being duplicated by the Cloud Storage Provider (CSP) to increase costs.
- Tag-Flexible Deduplication-Supported Integrity Check Protocol (TDICP) prevents hidden file replications and ensures that storage utilization remains efficient.
- User Transparency & Cost Efficiency: VeriDedup enables user-side deduplication verification, giving customers full control over their stored data and avoiding hidden expenses.

2. **Google Cloud Storage - Archive Class & VeriDedup Solution Issue:**

- 10TB of backup files were found to be duplicated, unnecessarily consuming extra storage space.
- Duplicate downloads caused multiple operations, leading to millions in additional cloud billing expenses.
- Users were unknowingly charged between $72.20 and $122.20 due to excessive duplicate files.
- Cloud providers profited by charging for unnecessary storage and redundant file operations.

**VeriDedup Solution:**

- User-Controlled Deduplication (UDDCP) enables users to verify if a file already exists before uploading, ensuring storage space is not wasted.
- Proxy Re-Encryption (PRE) ensures that only one encrypted copy of a file is stored, while multiple users can securely access it without duplication.
- Cost-Effective Storage Optimization: VeriDedup's deduplication techniques reduce data redundancy by up to 90%, directly lowering storage costs.
- Efficient Cloud Utilization: VeriDedup reduces unnecessary file requests and downloads, minimizing excessive billing for duplicate files.

3. **Google India Pvt. Ltd. vs Aravind Dubey & VeriDedup Solution Issue:**

- Users purchased additional storage but were still notified that storage was full due to duplicate files.
- Cloud providers failed to inform customers about duplicate file storage issues, leading to hidden costs.
- Unfair storage policies forced users to keep paying for more storage instead of optimizing their existing space.

**VeriDedup Solution:**

- Proof of Deduplication Correctness (PoDC) ensures that cloud providers cannot manipulate storage policies by falsely marking files as unique when they are actually duplicated.
- Transparent Data Usage Monitoring provides users with cryptographic proofs of stored files, ensuring accurate tracking of storage consumption.
- AES-256 Encryption with Deduplication enables secure deduplication of encrypted data, reducing redundant storage without compromising security.

- End-User Deduplication Control empowers users to audit and verify their storage, preventing CSPs from overcharging based on unnecessary duplications.

## Third-Party Deduplication Solutions vs VeriDedup

**StorReduce (AWS)**

- **Limitations:** Works only with AWS; lacks integrity proof.
- **VeriDedup Enhancement:** Ensures proof-based deduplication across multi-cloud environments.

**Duplicati (OneDrive, Google Drive)**

- **Limitations:** Limited deduplication verification.
- **VeriDedup Enhancement:** Prevents CSPs from falsely reporting duplicates and provides cryptographic proof of stored data.

**CloudBacko**

- **Limitations:** Basic deduplication without user verification.
- **VeriDedup Enhancement:** Uses Proof of Retrievability (PoR) to verify stored data.

**Panzura**

- **Limitations:** Lacks cryptographic security.
- **VeriDedup Enhancement:** Integrates AES-256 encryption with deduplication tracking, ensuring both security and cost efficiency.

## Why VeriDedup is the Ultimate Cloud Deduplication Solution

- Prevents CSP Storage Fraud: Ensures cloud providers cannot falsely charge users for duplicated storage by providing transparent verification mechanisms.
- End-to-End Security: Uses AES encryption, Proxy Re-Encryption, and Proof of Retrievability (PoR) to ensure secure and efficient deduplication.
- Massive Cost Savings: Eliminates redundant file storage, optimizes cloud utilization, and significantly reduces storage costs for users.
- Multi-Cloud Compatibility: Works seamlessly across AWS, Google Cloud, Microsoft Azure, and hybrid cloud environments.
- User-Centric Deduplication: Gives users full control over file deduplication, ensuring that only necessary data is stored while maintaining integrity and efficiency.

# CHAPTER – III

# SYSTEM ANALYSIS & DESIGN

## 3.1. PRELIMINARY INVESTIGATION

The initial step in the development of any project begins with a clear problem statement and a proposed solution that addresses the identified challenges. Our project, VeriDedup: A Verifiable Cloud Data Deduplication Scheme With Integrity and Duplication Proof, was conceptualized to solve the critical security, integrity, and storage efficiency challenges associated with cloud-based deduplication systems.

The goal of this project is to develop a secure and efficient deduplication framework that allows cryptographic verification of stored data while preventing Cloud Service Provider (CSP) manipulation in the deduplication process. The system is designed with independent integrity verification, secure duplication proof mechanisms, and robust cryptographic techniques to ensure fairness and transparency in cloud storage.

After approval by the organization and project guide, the preliminary investigation commenced, involving the following key activities:

- Request Clarification
- Feasibility Study
- Request Approval

## 3.2. REQUEST CLARIFICATION

Once the project proposal was approved, the next step was to analyze and clarify the exact requirements of the system. The VeriDedup project is specifically designed for cloud-based storage environments, where multiple users interact with a shared, deduplicated dataset. Unlike traditional storage systems, our project focuses on ensuring verifiability, transparency, and security in deduplication.

Today, cloud storage is widely used across industries, and organizations require a reliable, cost-effective, and secure deduplication solution to optimize storage without compromising integrity and trust. VeriDedup addresses these concerns by:

- Allowing users to verify data integrity independently.
- Preventing CSPs from falsely claiming uniqueness of files to overcharge users.
- Securing deduplication verification with cryptographic proof mechanisms.
- Reducing storage and bandwidth costs while maintaining data security.

The project is designed to function in a secure cloud-hosted environment, leveraging modern security techniques to enhance storage efficiency and user trust.

## 3.3. FEASIBILITY ANALYSIS

A critical outcome of the preliminary investigation is determining whether the system request is feasible within limited resources and time constraints. To assess this, a detailed feasibility study was conducted, covering the following aspects:

### 3.3.1. Operational Feasibility

Operational feasibility examines whether the proposed system effectively meets the needs of its intended users and whether it can be integrated into existing storage infrastructures.

- Efficiency: VeriDedup eliminates the need for users to blindly trust the CSP by introducing verifiable integrity and deduplication proof mechanisms.
- Security: Ensures that data integrity remains intact even after deduplication, reducing the risk of unauthorized modifications.
- Automation: The system automates duplication verification and integrity checks, reducing manual verification efforts for users.
- User Accessibility: Designed for multi-user environments, allowing seamless interaction between multiple users and the CSP.

By implementing AES encryption for secure data storage and authentication mechanisms, the system ensures high operational efficiency while addressing security concerns in deduplication-based storage. Based on these factors, the system is proven to be operationally feasible.

### 3.3.2. Economic Feasibility

Economic feasibility determines whether the proposed system provides a cost-effective solution and whether its benefits outweigh the implementation costs.

- The system leverages open-source technologies, reducing infrastructure costs.
- No additional storage costs are incurred for verification tags, as the system optimizes deduplicated metadata storage.
- Unlike traditional storage verification methods, VeriDedup reduces storage costs by preventing CSPs from falsely claiming a file is unique, ensuring fair billing practices.
- The use of lightweight cryptographic techniques ensures that the system does not introduce excessive computational overhead, making it economically viable.

Given that the system is designed to work with existing cloud providers, private servers, and local databases, it is economically feasible for deployment and adoption.

### 3.3.3. Technical Feasibility

Technical feasibility evaluates whether the system can be developed and implemented using the available technology stack. VeriDedup is built on modern web development technologies, security frameworks, and encryption mechanisms that ensure scalability and security.

The system is developed using:

- Backend: Java with JSP (Java Server Pages)
- Database Support: MySQL
- Security & Cryptography:
  - AES (Advanced Encryption Standard) for data encryption
  - User authentication with session management
  - Secure access control mechanisms
- Frontend: HTML, CSS, JavaScript
- Server Deployment:
  - Apache Tomcat Server 9.0 for web application hosting
  - WAMP Server 3.4 for database and backend integration
- Dependencies & Frameworks:
  - Various security dependencies for authentication and data protection
  - JDBC for database connectivity
  - Servlets for server-side logic execution

The system is designed to be scalable and secure, ensuring compatibility with existing enterprise infrastructure and local server-hosted environments. With this advanced technical stack, VeriDedup is technically feasible and scalable for real-world deployment.

### 3.4. REQUEST APPROVAL

After conducting a thorough feasibility analysis, the project was approved for development based on the following factors:

1. Addresses a critical problem in cloud storage by ensuring verifiable deduplication security.
2. Proven feasibility in terms of operations, cost, and technology.
3. Aligns with industry best practices for cloud security and storage optimization.
4. Enhances user trust and security by integrating cryptographic verification mechanisms.

The system was deemed ready for full-scale development, leading to the next phases of detailed design, implementation, and testing.

The preliminary investigation confirmed that VeriDedup is a feasible, cost-effective, and technically viable solution for addressing integrity and security concerns in deduplication-based storage systems. By ensuring independent integrity verification and secure duplication proof, the system provides a transparent, fair, and secure cloud storage experience.

With approval granted, development proceeded towards system design, implementation, and security testing, ensuring a fully functional and scalable solution for modern cloud storage infrastructures.

### 3.5. EXISTING SYSTEM

With the increasing reliance on cloud storage solutions, numerous research efforts have focused on ensuring data integrity, security, and efficient storage management. One of the major concerns in cloud storage is verifying data integrity while simultaneously supporting deduplication. Various cryptographic techniques have been proposed to address these issues, but most existing solutions either fail to fully support deduplication or lack robust verification mechanisms that prevent Cloud Service Providers (CSPs) from manipulating storage records for financial gain.

Shacham and Waters introduced a Compact Proof of Retrievability (PoR) mechanism, which incorporates erasure coding, BLS signatures, and Message Authentication Codes (MACs) to ensure that users can verify the integrity of their stored data without retrieving the entire file. This scheme is designed to reduce the communication overhead involved in proof generation while maintaining strong integrity guarantees. However, the computational complexity of generating authentication tags is high, and the number of authentication tags grows linearly with the number of file blocks, making this method less efficient for large-scale cloud storage environments.

To improve on the existing PoR framework, Xu and Chang proposed an enhanced scheme that integrates polynomial commitments to reduce the communication cost associated with integrity verification. By using a commitment-based approach, this system aims to provide secure and efficient integrity checks without requiring users to store large amounts of metadata. However, despite its improved efficiency, this scheme does not support deduplication over verification tags, making it impractical for cloud storage providers that rely on deduplication to optimize storage space.

Azraoui et al. proposed StealthGuard, a security framework that integrates Private Information Retrieval (PIR) with a Word Search (WS) technique to securely retrieve verification tags, also referred to as watchdogs. This system enables an unlimited number of integrity verification queries while minimizing the computational overhead required to generate watchdogs. Unlike earlier schemes that rely on computationally expensive tag generation methods, StealthGuard provides a lightweight verification mechanism. Additionally, the storage overhead required for maintaining watchdogs is

lower compared to traditional PoR schemes. However, like many existing solutions, StealthGuard fails to support deduplication, making it unsuitable for cloud environments that require both integrity verification and storage efficiency.

Zheng et al. introduced a Publicly Verifiable Proof of Storage (PoS) scheme with deduplication, allowing users to verify the integrity of their stored data using a cryptographic key from the first user who originally uploaded the file. While this approach attempts to combine deduplication with integrity verification, it has several critical vulnerabilities. First, the scheme has been proven insecure under weak key attacks, allowing attackers to compromise data integrity verification. Additionally, the scheme does not protect users from CSP manipulation, meaning that users have no way of verifying whether the CSP is falsely claiming uniqueness for files to generate higher storage costs.

Vasilopoulos et al. proposed a message-locked PoR scheme that integrates PoR mechanisms with deduplication functions. This approach ensures that deduplicated files generate identical verification tags, preventing CSPs from tampering with stored data. However, while this system improves storage optimization, it introduces a new security trade-off—multiple users who store the same file in the cloud may generate different verification tags due to the use of individual cryptographic keys. While this prevents brute-force attacks, it negatively impacts deduplication efficiency, limiting the practical implementation of this approach.

Overall, despite the various advancements in PoR and deduplication-based verification systems, existing solutions suffer from fundamental limitations that prevent their effective deployment in real-world cloud storage environments. These limitations include high computational overhead, lack of deduplication support, vulnerability to CSP manipulation, and weak protection against brute-force attacks.

### 3.5.1. Disadvantages :

While existing cloud storage security mechanisms provide basic integrity verification and deduplication support, they fail to address several critical security concerns, leaving users vulnerable to data manipulation, financial exploitation, and integrity threats. The key disadvantages of current systems are as follows:

1. Snooping on Private Data
    - Most existing schemes do not offer sufficient privacy protection mechanisms, leaving user data vulnerable to unauthorized access.
    - Since CSPs have complete control over stored files, they can potentially analyze, modify, or disclose sensitive user data without consent.
    - Traditional deduplication schemes lack privacy-preserving mechanisms, allowing attackers to use dictionary attacks to determine whether a specific file exists in the cloud.

2. Lack of Secure Deduplication Verification
    - Existing deduplication systems rely entirely on the CSP's internal processes to determine whether a file is unique or already stored. This creates an imbalance of control, where users must blindly trust the CSP without any independent verification mechanism.
    - CSPs may intentionally mislead users by falsely claiming that an uploaded file is not a duplicate, forcing them to store redundant data and pay for additional storage.
    - Without cryptographic proof of duplication correctness, CSPs can manipulate billing models to maximize profits.
3. Incompatible Integrity Verification Methods
    - Traditional PoR schemes require unique verification tags for each user, making them incompatible with deduplication-based storage.
    - Since deduplication demands a single verification tag per file, most existing integrity verification mechanisms cannot be applied to deduplicated files, leaving users without a way to confirm data authenticity.
    - Users who store files on a shared storage platform lack a secure method to verify whether their data has been altered after deduplication.
4. Risk of Data Loss Due to CSP Negligence
    - Since CSPs store only a single instance of deduplicated files, any accidental deletion, corruption, or unauthorized modification of the stored file affects all users who reference that file.
    - In many cases, CSPs prioritize storage efficiency over security, leading to situations where data loss occurs due to improper storage management.
    - Traditional cloud storage solutions do not provide cryptographic proof mechanisms that ensure files remain retrievable and unmodified over time.
5. Lack of Protection Against Brute-Force and Replay Attacks
    - Many existing schemes do not adequately protect against brute-force attacks, where an adversary systematically uploads known files to infer the presence of specific data in the cloud.

- Message-locked encryption schemes fail to prevent replay attacks, where attackers reuse valid verification responses to bypass security mechanisms.

Due to these limitations, existing systems fail to provide a secure, transparent, and verifiable deduplication process, leaving cloud storage users exposed to financial exploitation, integrity threats, and unauthorized access risks. Therefore, a new approach is needed to address these gaps in security, verifiability, and fairness.

### 3.6. PROPOSED SYSTEM

To address the critical security, integrity, and trust issues in existing deduplication-based cloud storage systems, this project introduces VeriDedup, a secure and verifiable cloud data deduplication framework. The proposed system overcomes the limitations of existing solutions by introducing cryptographic proof mechanisms that allow users to verify both data integrity and the correctness of deduplication checks, preventing CSP manipulation, unauthorized data modifications, and unfair billing practices.

**Key Features of the Proposed System**

1. Tag-Flexible Deduplication-Supported Integrity Check Protocol (TDICP):
    - The system introduces TDICP, a novel Proof of Retrievability (PoR) scheme based on Private Information Retrieval (PIR) to enable integrity verification for deduplicated files stored in the cloud.
    - Unlike traditional PoR mechanisms, which require unique verification tags per user, TDICP allows users to generate individual integrity verification tags while still allowing deduplication of these tags at the CSP.
    - This ensures that users can independently verify whether their stored data has been tampered with or altered, without requiring interaction with other data owners or relying entirely on CSP trust.
    - By leveraging PIR techniques, TDICP ensures that data integrity can be checked efficiently and privately, without exposing sensitive metadata.
2. User Determined Duplication Check Protocol (UDDCP):
    - The system introduces UDDCP, a novel user-controlled duplication verification protocol based on Private Set Intersection (PSI), which ensures that CSPs cannot manipulate deduplication outcomes for financial gain.

- Unlike conventional deduplication systems where CSPs determine whether a file is unique, UDDCP enables users themselves to verify file duplication before uploading.
- This mechanism prevents CSPs from falsely claiming that a file is unique and charging unnecessary storage fees for redundant files.
- UDDCP strengthens fairness and transparency in cloud storage pricing models by allowing users to cryptographically prove whether their file already exists in the cloud.

3. Construction of VeriDedup - A Comprehensive Secure Deduplication Framework:
   - The system integrates TDICP and UDDCP into a single deduplication framework, named VeriDedup, which provides both integrity verification and user-driven duplication proof.
   - Unlike previous schemes that either focus only on PoR or only on deduplication, VeriDedup simultaneously ensures both functionalities, making it a comprehensive solution for secure and efficient cloud storage.
   - The system also incorporates Proof of Ownership (PoW) mechanisms and data access key assignment to further enhance the security and usability of deduplicated storage.
4. Security Analysis and Performance Evaluation:
   - The system rigorously proves the security of TDICP and UDDCP by formulating multiple cryptographic game-based security models to evaluate their robustness.
   - A combination of theoretical analysis and experimental simulations is conducted to analyze the efficiency, scalability, and real-world applicability of the proposed system.
   - Performance evaluations show that VeriDedup significantly outperforms existing schemes in terms of integrity verification speed, storage overhead reduction, and deduplication accuracy.

By integrating these advanced cryptographic techniques, VeriDedup ensures that users remain in control of their data, CSPs cannot manipulate deduplication results, and data integrity is verifiable even after deduplication is performed.

### 3.6.1 ADVANTAGES

The proposed VeriDedup system introduces several advantages over traditional deduplication-based cloud storage solutions, ensuring better security, transparency, and efficiency:

1. Independent Integrity Verification During Deduplication:
    - VeriDedup allows users to verify the integrity of their stored files without needing to download the entire file.
    - Traditional integrity verification requires interaction with the data owner, but VeriDedup enables each user to perform independent integrity checks, reducing reliance on trusted third parties.
    - By using TDICP, VeriDedup ensures that integrity verification is efficient and scalable, even when applied to large-scale cloud storage environments.
2. Flexible and Secure Tag Generation for Integrity Verification:
    - Unlike traditional PoR schemes that require a single, universal verification tag per file, VeriDedup allows each user to generate their own verification tag while still maintaining deduplication compatibility.
    - This flexibility improves security, as users can create verification tags that are resistant to brute-force attacks, preventing unauthorized modifications or integrity violations.
    - By integrating PIR-based cryptographic techniques, VeriDedup ensures that verification tags do not expose sensitive information while still allowing deduplication at the CSP.
3. Guaranteed Correctness of Deduplication Checks:
    - Traditional deduplication systems rely entirely on the CSP to determine whether a file is already stored, leaving users vulnerable to incorrect or fraudulent duplication results.
    - VeriDedup introduces UDDCP, allowing users to independently verify file duplication before uploading, ensuring that CSPs cannot falsely claim uniqueness to charge extra storage fees.
    - The system prevents CSP manipulation of deduplication verification, ensuring transparency in cloud storage billing models and eliminating unfair storage charges.
4. Efficient and Scalable Storage Optimization Without Compromising Security:
    - VeriDedup reduces storage overhead and bandwidth consumption by ensuring that redundant verification tags do not create additional storage burdens.
    - Unlike previous solutions that either prioritize integrity verification at the cost of storage efficiency or optimize deduplication at the expense

of security, VeriDedup achieves both storage efficiency and cryptographic integrity verification.

- The system ensures that deduplication does not introduce vulnerabilities, allowing CSPs to optimize storage without compromising security.

5. Enhanced Security Against Brute-Force and Metadata Leakage Attacks:
   - Traditional deduplication mechanisms are vulnerable to brute-force attacks, where attackers upload known files to determine whether a specific file already exists in the cloud.
   - VeriDedup protects against such attacks by using PIR and PSI-based cryptographic verification mechanisms, ensuring that deduplication does not expose sensitive metadata.
   - Since users themselves determine duplication correctness rather than relying on the CSP, VeriDedup eliminates information leakage risks associated with cross-user duplication checks.
6. Fair and Transparent Cloud Storage Billing Model:
   - Existing cloud storage systems allow CSPs to overcharge users by manipulating deduplication verification results, leading to unfair billing practices.
   - VeriDedup ensures that users pay only for the actual storage they consume, preventing unfair pricing due to CSP misrepresentation of file duplication.
   - The system promotes trust and fairness between users and cloud service providers by ensuring that deduplication decisions are cryptographically verifiable.
7. Seamless Integration with Existing Cloud Storage Infrastructure:
   - VeriDedup is designed to be easily deployed within existing cloud storage systems, requiring minimal modifications to CSP infrastructure.
   - The system ensures that cloud storage remains scalable, secure, and efficient, making it suitable for enterprise-level deployments, personal cloud storage, and multi-user environments.

The proposed VeriDedup system represents a significant advancement in cloud storage security and efficiency, addressing the fundamental weaknesses of existing deduplication mechanisms. By integrating TDICP and UDDCP, the system provides a first-of-its-kind solution that ensures verifiable integrity verification and user-controlled duplication checks. Unlike previous solutions that fail to combine storage

optimization with strong cryptographic guarantees, VeriDedup achieves both efficiency and security without trade-offs.

By preventing CSP manipulation, ensuring fair billing, and enabling secure deduplication with independent integrity verification, VeriDedup sets a new standard for cloud storage security. The system paves the way for a more transparent, fair, and efficient cloud storage ecosystem, ensuring that users can store data securely while benefiting from the cost-saving advantages of deduplication.

### 3.7. MODULE DESCRIPTION

1. Cloud Storage Provider (CSP) Module
2. Admin Module
3. Client Module

Each module is responsible for specific functionalities that ensure secure, efficient, and verifiable cloud data deduplication. The CSP manages storage operations and monitors user activity, the Admin handles client authorization and system security, and the Client interacts with the storage system to perform file uploads, integrity checks, downloads, and deduplication verification.

#### 1. Cloud Storage Provider (CSP) Module

The Cloud Storage Provider (CSP) acts as the backend storage manager, ensuring that files are stored securely, deduplicated efficiently, and remain unaltered over time. The CSP does not have access to the actual contents of files, as all data is encrypted before upload, but it is responsible for storage analysis, deduplication management, and activity monitoring.

**Key Functionalities of the CSP Module:**

1. **Storage & Deduplication Management:**
    - Handles file storage operations efficiently.
    - Implements deduplication verification, ensuring that only unique files are stored.
    - Prevents redundant storage, reducing unnecessary storage costs and bandwidth usage.
2. **Integrity Verification:**
    - Ensures that files remain unchanged and untampered while stored in the cloud.
    - Uses cryptographic proof mechanisms to verify data integrity on request.

3. **Activity Monitoring & Security Auditing:**
    - Tracks all activities performed by admins and clients, such as file uploads, downloads, integrity checks, and deduplication verification.
    - Logs all storage and deduplication operations for security auditing and system optimization.
4. **Storage Analysis & Optimization:**
    - Provides detailed insights into storage usage, saved space due to deduplication, and data distribution across servers.
    - Helps in identifying trends, storage consumption, and potential system improvements.

The CSP ensures efficient storage utilization, prevents unnecessary data redundancy, and guarantees data security through integrity verification mechanisms.

**2. Admin Module**

The Admin serves as the central authority for managing client accounts, permissions, and security policies. The Admin ensures that only authorized users can access the system and perform data operations.

**Key Functionalities of the Admin Module:**

1. **Client Authorization & User Management:**
    - The Admin is responsible for approving or rejecting new client accounts.
    - Only authorized clients can log in and access the storage system.
    - Can revoke client permissions if any suspicious activity is detected.
2. **System Security Enforcement:**
    - Ensures that only verified and approved users interact with the cloud storage.
    - Implements security policies to prevent unauthorized access.
3. **Monitoring & Auditing:**
    - Views and tracks all user activities, including file uploads, downloads, integrity verification, and deduplication checks.
    - Maintains logs of client interactions with the storage system for security auditing.

4. **Deduplication Oversight:**
    - Ensures that deduplication is correctly applied, preventing CSPs from falsely claiming storage usage.
    - Monitors file integrity reports to detect any anomalies in stored data.

The Admin module ensures secure access control, system monitoring, and proper deduplication verification, maintaining the overall integrity of the VeriDedup system.

**3. Client Module**

The Client is the primary end-user of the system, interacting with the cloud storage for file uploads, integrity verification, downloads, and deduplication checks.

**Key Functionalities of the Client Module:**

1. **Secure Login & Authentication:**
    - Clients must be approved by the Admin before accessing the system.
    - Uses secure authentication mechanisms to prevent unauthorized access.
2. **File Upload with Deduplication Check:**
    - Before uploading, the system checks for duplicate files in the cloud storage.
    - If a duplicate exists, the client is granted access to the existing file instead of storing another copy.
    - If no duplicate is found, the file is encrypted using AES encryption before being uploaded to the cloud.
3. **Integrity Verification of Stored Files:**
    - Clients can perform integrity verification to check whether their files remain unaltered in storage.
    - Uses cryptographic verification techniques to detect data tampering or corruption.
4. **File Download & Secure Access:**
    - Clients can download their stored files securely after authentication.
    - Ensures that only authorized clients can access their own data.
5. **Activity Logs & User Tracking:**
    - Clients can view their activity history, including uploaded files, verified files, and downloads.

- Provides a transparent record of all user actions, ensuring accountability.

The Client Module allows users to securely store, verify, and manage their files while ensuring efficient storage utilization through deduplication verification.

**SYSTEM WORKFLOW**

1. Admin approves or revokes client access.
2. Client logs in and uploads files.
    - System performs deduplication check before storing files.
    - If a duplicate is found, the client gains access to the existing file.
    - If no duplicate is found, the file is encrypted and uploaded securely.
3. Client verifies file integrity to ensure data has not been tampered with.
4. Client downloads files securely when needed.
5. CSP monitors storage usage, tracks admin and client activities, and ensures deduplication compliance.
6. Admin oversees the system, approves new users, monitors logs, and ensures security policies are followed.

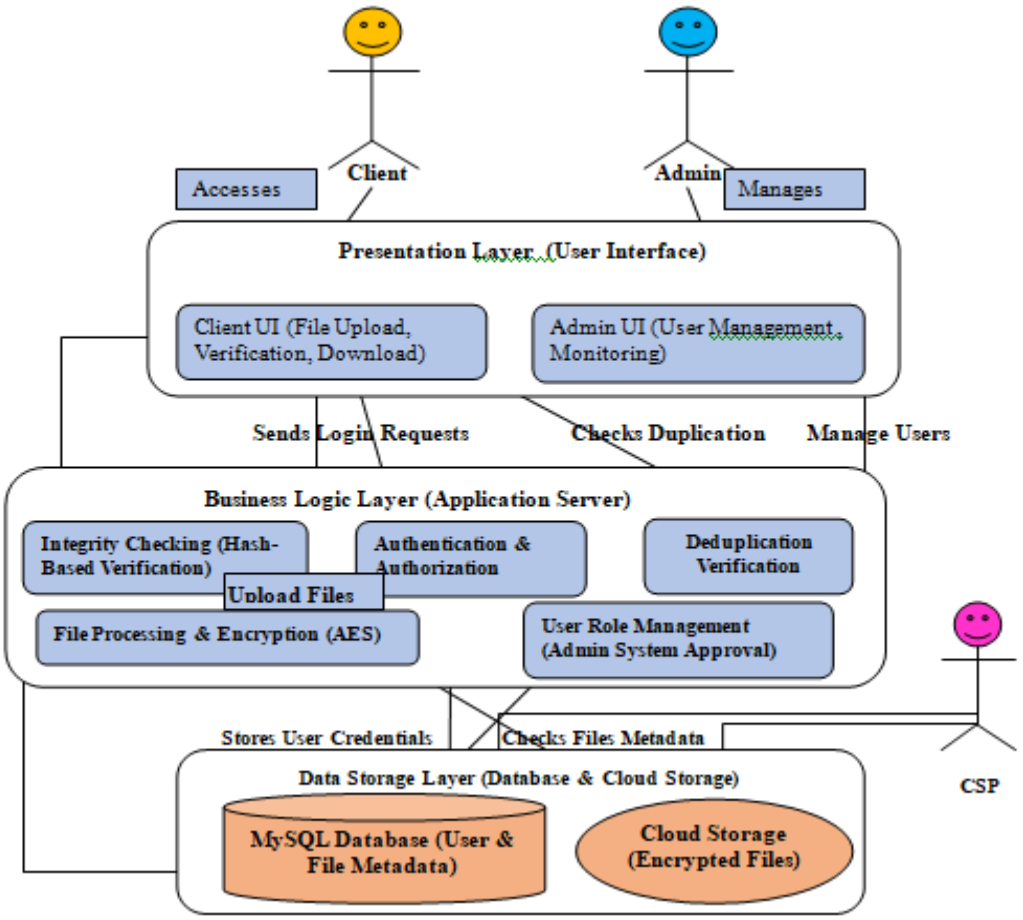
The VeriDedup system is structured to ensure efficient, secure, and verifiable cloud data deduplication through its three core modules:

- **CSP (Cloud Storage Provider):** Manages storage, deduplication, integrity verification, and system monitoring.
- **Admin:** Controls user authorization, system security, and activity monitoring.
- **Client:** Performs file uploads, integrity verification, downloads, and deduplication checks.

This modular architecture ensures optimized storage usage, strong data integrity, and a secure access control mechanism, making VeriDedup a robust and scalable solution for modern cloud storage security.

## 3.8. System Architecture :

The VeriDedup system follows a structured three-tier architecture that ensures efficient cloud data deduplication, secure storage management, and verifiable integrity checks. This architecture facilitates file upload, duplication verification, encryption, storage, and access control while ensuring security and scalability.




## 1. System Architecture Overview

The VeriDedup system consists of the following three layers:

- Presentation Layer (User Interface): Client and Admin interactions
- Business Logic Layer (Application Server): File processing, duplication checks, encryption, and integrity verification
- Data Storage Layer (Database & Cloud Storage): Stores encrypted files, metadata, and verification tags

Each layer is responsible for handling specific tasks efficiently and securely.

## 2. Workflow of VeriDedup System

### 1. File Upload & Duplication Check

- The client browses and selects a file to upload.
- The system generates cryptographic signatures (hash values) for the entire file and each block.
- A deduplication check is performed at both:
    - File Level: Checks if the same file already exists in storage.

  - Block Level: Identifies and eliminates duplicate data blocks within or across files.

- If a duplicate file exists, the client is granted access without re-uploading.
- If no duplicate is found, the file is encrypted (AES encryption) and uploaded to cloud storage.

**2. File Request & Response**

- A client requests access to a stored file.
- The system verifies the request and retrieves the encrypted file from cloud storage.
- If the file belongs to the client, the decryption key is provided, and the client downloads the file securely.

**3. File Access & Integrity Verification**

- Clients can verify the integrity of their stored files.
- The system retrieves the stored cryptographic signatures and compares them with the current file hashes.
- If any modification is detected, the system alerts the client about possible tampering or corruption.
- Only authorized clients can access their respective files, ensuring data security and privacy.

**3. Key System Components & Functions**

**1. Presentation Layer (User Interface) Actors:**

- Client: Uploads, verifies, and downloads files.
- Admin: Approves clients, monitors activity, and manages permissions.

**Functions:**

- Browse & Select File
- Request Access to Stored Files
- Verify File Integrity
- Manage Users & Permissions (Admin Only)

2. **Business Logic Layer (Application Server) Technology Used: Java (JSP, Servlets)**

**Functions:**

- Authentication & Authorization: Ensures only approved users can access the system.
- File Processing & Encryption: Encrypts files before storage.
- Duplication Verification: Checks for redundant files and blocks.
- Integrity Checking: Uses cryptographic hash functions for verification.
- User Management: Allows admins to approve or revoke client access.

3. **Data Storage Layer (Database & Cloud Storage) Technology Used: MySQL (Database), Cloud Storage (External CSPs)**

**Functions:**

- Stores user credentials, file metadata, deduplication records, and integrity verification tags.
- Manages file encryption and retrieval.
- Ensures storage optimization through deduplication.

The VeriDedup system architecture ensures secure, scalable, and efficient cloud data deduplication by integrating authentication, encryption, integrity verification, and access control mechanisms. The system prevents storage wastage, ensures file integrity, and optimizes cloud usage while maintaining a seamless and secure user experience.

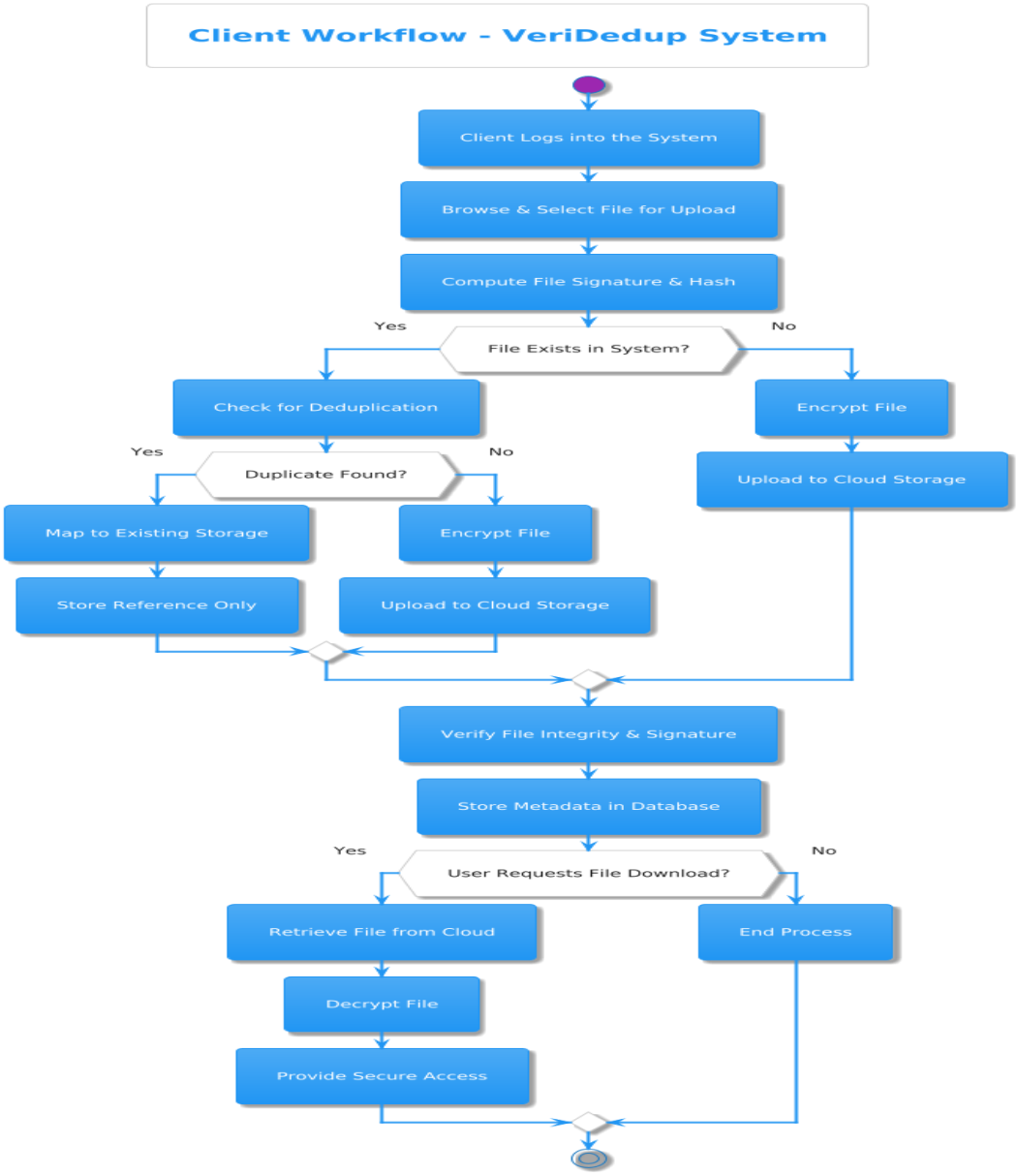
**Client Workflow :**

The client in the VeriDedup system interacts with secure file deduplication, integrity verification, and access control mechanisms to ensure data security, efficiency, and seamless storage operations. Below is the detailed workflow for a client using the system.

1. Authentication & Secure Access

- The client logs into the VeriDedup system using secure authentication (e.g., username, password, multi-factor authentication).
- Role-based access control (RBAC) ensures the client can only access their files and operations.
- The system establishes a secure session with encryption to protect login credentials and session data.

**Flowchart for Client :**




2. File Upload & Deduplication Check

A. File Selection & Signature Computation

- The client selects one or multiple files for upload.
- The VeriDedup system computes a unique cryptographic hash (SHA-256, RSA-based signature, or other secure methods) for each file to check for duplication.

B. Deduplication Decision Process

If the file already exists in storage:

- The system links the file to the existing stored version without requiring re-upload.

- The client gets an access reference, ensuring efficient storage utilization.
- This process helps in saving bandwidth, storage space, and processing power.

If the file is new:

- The file undergoes AES/RSA encryption for confidentiality.
- A deduplication metadata entry is created in the database.
- The file is uploaded securely to the cloud storage (AWS S3, Azure Blob, Google Cloud, or an on-premises server).
- A confirmation message is displayed to the client with details.

3. File Integrity Verification

- After upload, the system runs integrity verification using cryptographic techniques.
- A verification tag (HMAC, hash signature, or digital signature) is stored for future integrity checks.
- The system ensures that the file remains unaltered by periodically running verification audits.

Client's Role in Integrity Verification

- The client can manually request a file integrity check.
- The system will compare the current hash with the original hash stored at upload time.
- If integrity is compromised, an alert is generated.

4. File Retrieval & Secure Access

A. Secure File Download

- When the client requests a file:
    - The system retrieves the stored file from the cloud storage.
    - It decrypts the file securely.
    - The file is delivered to the client via a secure channel (HTTPS, TLS encryption).

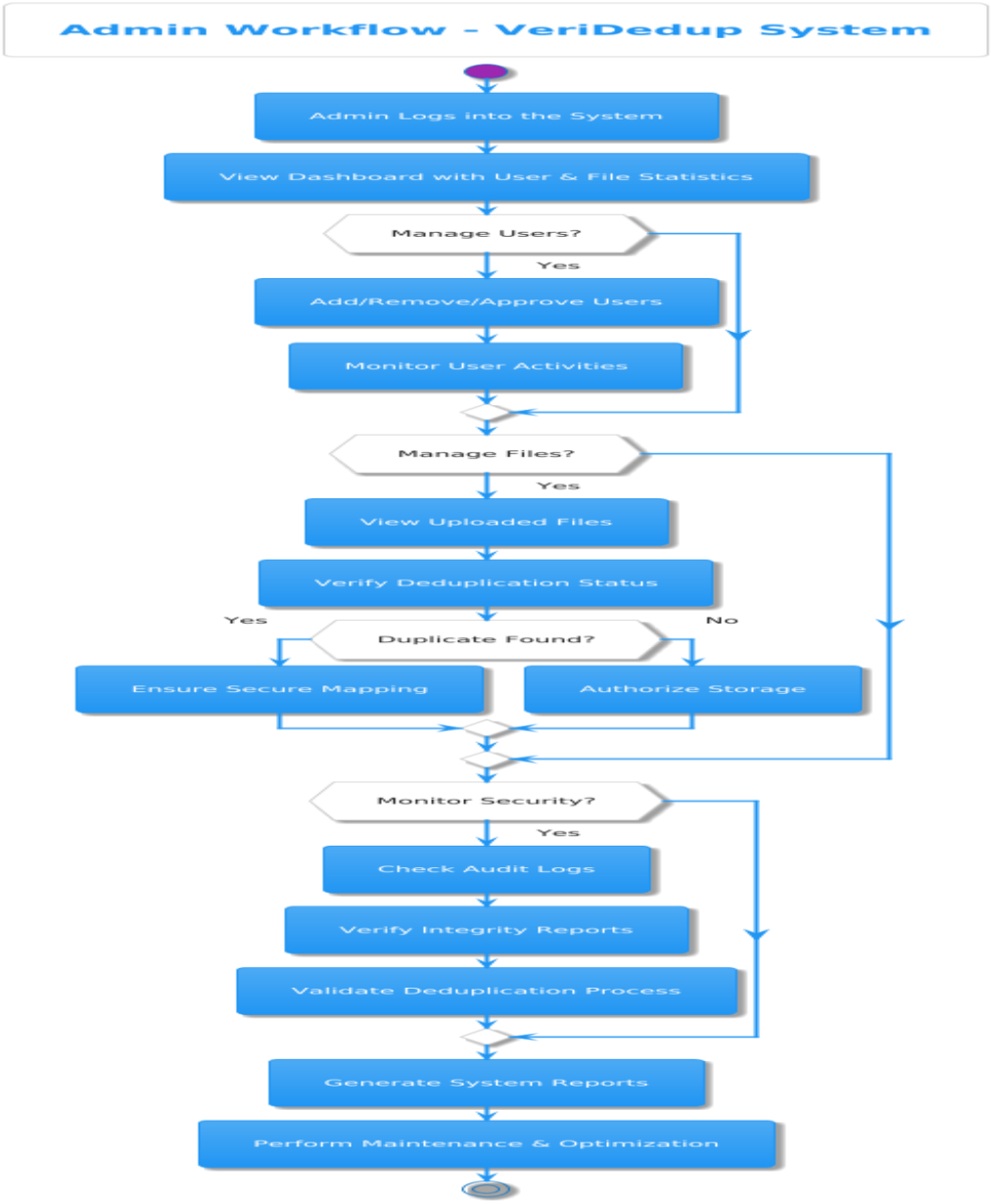
B. Access Control & Permissions

- The client can grant or revoke access permissions for specific users.
- Access logs track every file interaction, ensuring accountability.
- If unauthorized access is detected, the system alerts the client and admin.

5. Additional Features & Security Enhancements

- Two-Factor Authentication (2FA) – Ensures additional login security.
- File Sharing with Time-Limited Links – Clients can generate temporary access links.
- Audit & Activity Logs – Full history of uploads, downloads, and integrity checks.
- Cloud Storage Optimization – Smart deduplication ensures minimal redundancy.
- Real-Time Notifications – Clients receive alerts for file integrity failures or unauthorized access attempts

**Admin Flow Chart :**.



**Admin Workflow :**

The admin in the VeriDedup system is responsible for user management, file monitoring, deduplication validation, system security, and overall platform oversight. Below is a comprehensive workflow detailing the admin's role in managing VeriDedup efficiently.

**1. Authentication & Secure Dashboard Access**

- The admin securely logs into the VeriDedup system using multi-factor authentication (MFA) for enhanced security.
- Role-based access control (RBAC) ensures only authorized admins can access privileged functions.
- The admin dashboard provides real-time system statistics, including:
    - Active users and their recent activities.
    - Uploaded files and deduplication status.
    - Integrity verification results.
    - Security logs tracking failed login attempts or suspicious activities.

**2. User Management & Access Control**

- The admin can add, remove, approve, or suspend users based on system policies.
- User role assignments (Client, Admin, Auditor) are managed to ensure access restrictions.
- Account verification & approval: Before a client or auditor gains access, the admin can review requests.
- Monitoring User Activity Logs:
    - Tracks uploads, downloads, integrity verification requests, and authentication attempts.
    - Flagging Suspicious Behavior: The system detects unusual access patterns and alerts the admin.

**3. File Management & Deduplication Oversight**

**A. Viewing & Managing Files**

- The admin can view all uploaded files in the system along with their metadata (owner, size, hash value, integrity status).
- The system automatically detects and flags duplicate files to prevent redundant storage.

**B. Deduplication Check & Storage Authorization**

- The system performs automatic deduplication checks:
    - If a duplicate exists: The file is securely mapped to the existing stored copy without re-uploading.
    - If the file is unique: The admin can authorize storage and encryption.
- Manual Review Option: The admin can override deduplication in case of false positives.

**4. Security Monitoring & Integrity Audits**

- The admin plays a crucial role in system security and integrity validation by monitoring audit logs and security alerts.
- Audit Logs Analysis:
    - The system maintains detailed logs of all operations, including uploads, downloads, and verification requests.
    - Admins can filter logs by user, date, file ID, or security events.
- Integrity Verification Reports:
    - Ensures files remain unaltered and were not tampered with after storage.
    - Uses hash comparisons and digital signatures to verify integrity.
    - Admins can manually trigger verification checks for specific files.
- Deduplication Validation & Compliance:
    - Ensures the deduplication process is correctly executed without compromising data security.
    - Prevents unauthorized overwriting or corruption of stored files.

**5. System Reports, Performance Optimization & Alerts**

- The admin can generate detailed system reports covering:
    - File storage usage: Trends, duplicate rates, and optimization insights.
    - User activity analytics: Upload/download patterns, verification requests.
    - Security logs: Unusual access attempts, failed logins, integrity issues.
- System Performance Monitoring:
    - Admins oversee server health, storage efficiency, and deduplication effectiveness.

  - Cloud storage management: Ensures optimal utilization of AWS S3, Azure, or on-premises storage.

- Automated Alerts & Notifications:
  - Immediate alerts for suspicious activity, failed integrity checks, or unauthorized file access.
  - Notifications when storage capacity nears a threshold or deduplication saves significant space.

**Additional Features Enhancing Admin Capabilities:**

- Host/Admin Communication: Direct reporting to the host for escalated issues.
- Access Revocation & Suspension: Quick action against compromised accounts.
- Secure API & Third-Party Integration: Ensures seamless interconnectivity with external services.
- Encryption & Compliance Checks: Ensures stored data adheres to security regulations.

The admin plays a critical role in VeriDedup's ecosystem, ensuring data security, efficient deduplication, and integrity monitoring. By managing users, files, and system health, the admin contributes to a highly optimized and secure data storage solution.

**Cloud Service Provider (CSP) Workflow :**

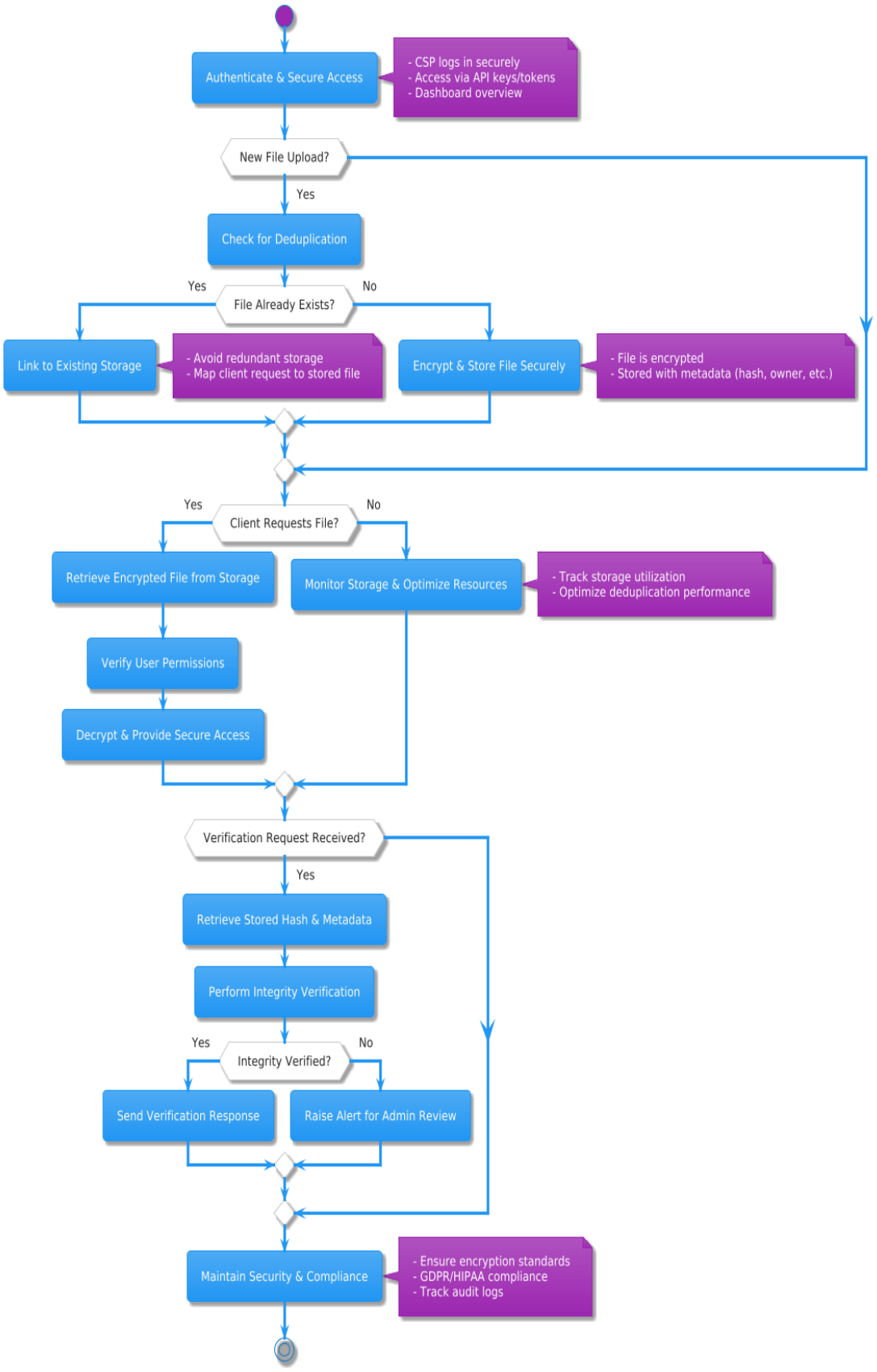
The Cloud Service Provider (CSP) in the VeriDedup system is responsible for securely storing files, handling deduplication operations, responding to verification requests, and maintaining system integrity. The CSP ensures efficient storage management while maintaining high security and compliance with deduplication and verification protocols.

1. **Authentication & Secure System Access**

- The CSP logs into the VeriDedup system using secure authentication mechanisms, ensuring only authorized personnel can access cloud storage operations.
- Secure API keys and access tokens are used to facilitate interactions between VeriDedup and cloud storage platforms (AWS S3, Google Cloud, Azure, etc.).
- The CSP dashboard provides real-time statistics, including:
  - Storage usage and available capacity.
  - Active deduplication requests and system performance metrics.

- Integrity verification results and flagged security incidents.

- **CSP Flow Chart :**



2. **File Storage & Management**

**A. Secure File Storage Process**

- When a client uploads a file, the system first checks for duplication using hash-based and cryptographic techniques.
- If the file is unique:
    - It undergoes encryption before storage.
    - The CSP securely stores the encrypted file in the cloud.
    - Metadata (hash, owner, storage location, integrity signature) is stored in the database.
- If the file is a duplicate:
    - The system maps the client's request to the existing file without re-uploading.
    - Storage space is optimized by avoiding redundant file storage.

**B. File Retrieval & Access**

- When a client requests a file download, the CSP retrieves it from the cloud storage and ensures:
    - Proper decryption and security checks before providing access.
    - Verification of user access permissions before file retrieval.
- CSP ensures high availability and minimal latency in file access.

3. **Deduplication Processing & Optimization**

**A. Deduplication Workflow**

- The CSP performs real-time deduplication checks on uploaded files:
    - If a matching hash exists, the system prevents redundant uploads.
    - If the file is new, it is securely stored with a unique signature.
- The deduplication engine optimizes storage by reducing duplicate copies and maintaining storage efficiency.
- The CSP logs deduplication decisions and flags anomalies for further verification.

**B. Storage Optimization & Resource Management**

- The CSP monitors storage capacity, ensuring optimal resource allocation.

- Automated alerts notify CSP of nearing storage limits or potential deduplication inefficiencies.
- Data lifecycle management policies are enforced, allowing periodic cleanup or archiving of outdated files.

**4. Integrity Verification & Security Compliance**

**A. Responding to Integrity Verification Requests**

- When a verification request is received, the CSP retrieves stored file hashes and metadata.
- Cryptographic integrity checks are performed using signatures and secure hash comparisons.
- If discrepancies are found (file corruption, unauthorized modifications), alerts are raised for admin review.

**B. Security Measures & Compliance**

- The CSP ensures that all stored files meet security and compliance requirements, including:
    - End-to-end encryption for stored data.
    - Secure access control mechanisms to prevent unauthorized access.
    - Compliance with GDPR, HIPAA, and other relevant regulations for secure cloud storage.
- Audit logs are maintained, tracking all file storage and retrieval operations.

**5. Verification Response & System Maintenance**

**A. Handling Verification Requests from Auditors/Admins**

- The CSP responds to verification requests by:
    - Retrieving the original file's hash and stored metadata.
    - Comparing it with the integrity signature to validate its authenticity.
    - Providing proof of unaltered storage if required by auditors.

**B. Continuous System Optimization**

- The CSP regularly monitors system performance, including:
    - Deduplication efficiency reports.
    - Storage utilization trends and capacity planning.
    - Anomaly detection in file integrity and security breaches.

- Automated performance tuning ensures minimal latency in file access and deduplication operations.

**Additional CSP Capabilities**

- Cloud Backup & Redundancy: Ensures high availability by maintaining redundant copies of critical data.
- Disaster Recovery Mechanisms: Implements failover strategies in case of unexpected cloud failures.
- API & Interoperability: Provides secure API endpoints for seamless integration with VeriDedup services.
- Encryption Key Management: Ensures proper handling of encryption keys for secure file storage and retrieval.

The Cloud Service Provider (CSP) plays a vital role in the VeriDedup ecosystem by efficiently handling storage, deduplication, integrity verification, and security compliance. Through robust optimization and real-time monitoring, the CSP ensures seamless storage operations while maintaining data integrity and security.

### 3.9 Design Methodologies :

Java is both a programming language and a platform. It's characterized as simple, object-oriented, architecture-neutral, portable, distributed, high-performance, interpreted, multithreaded, robust, dynamic, and secure.

#### 3.9.1 System Environment :

**Compilation and Interpretation:**

Java programs are compiled into Java bytecode (platform-independent intermediate code) and then interpreted by the Java Virtual Machine (JVM). Compilation happens once; interpretation occurs on each execution. The JVM allows "write once, run anywhere" capability.

**Java Platform:**

The Java platform is a software-only platform consisting of the Java Virtual Machine (JVM) and the Java Application Programming Interface (API). The API is a large collection of pre-built software components organized into packages.

**Java Applications:**

Java supports applets (programs that run in web browsers) and standalone applications, including servers (web, proxy, mail, print) and servlets (server-side extensions of web servers).

**Java API Features:**

The Java API provides features:

- Essentials (objects, strings, threads, I/O)
- Applets
- Networking (URLs, TCP, UDP, IP)
- Internationalization
- Security
- Software components (JavaBeans)
- Object serialization (RMI)
- Java Database Connectivity (JDBC)
- 2D/3D graphics and other specialized APIs

**Benefits of Java:**

- Quick to learn
- Less code (smaller than C++)
- Better code (garbage collection, object-orientation)
- Faster development
- Platform independence (100% Pure Java)
- Easy software distribution

### 3.9.2. ODBC

ODBC is a standard programming interface that allows applications to connect to various databases using a common set of function calls, eliminating the need for database-specific languages. The ODBC Administrator manages data sources, which act as pointers to specific databases (e.g., SQL Server, Access) residing on the LAN. ODBC system files are installed with database applications and administered via ODBCINST.DLL or ODBCADM.EXE (16/32-bit versions). ODBC drivers handle communication with the specific database, transparently to the application.

**Benefits:**

Consistent interface for multiple databases. Simplifies code.

**Disadvantages:**

Potential performance overhead compared to native interfaces, but improving driver quality and faster computers mitigate this.

### 3.9.3. JDBC :

JDBC (Java Database Connectivity) is Sun Microsystems' API for database access in Java, providing a consistent interface to various RDBMSs. This consistency is

achieved using "plug-in" drivers, requiring vendors to provide drivers for each supported platform.

**Design & History:**

1. JDBC's framework is based on ODBC for faster adoption, leveraging ODBC's widespread support. Announced in March 1996, with the final v1.0 specification released soon after a 90-day public review.

### 3.9.4. JDBC Goals :

2. The design goals have shaped JDBC into a solid framework for building Java database applications. A complete overview of JDBC is beyond the scope of this material.

3. ***SQL Level API***

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

4. ***SQL Conformance***

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

5. ***JDBC must be implemental on top of common database interfaces***
   The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

6. ***Provide a Java interface that is consistent with the rest of the Java system***

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

7. ***Keep it simple***

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

8. ***Use strong, static typing wherever possible***

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

9. ***Keep the common cases simple***

Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

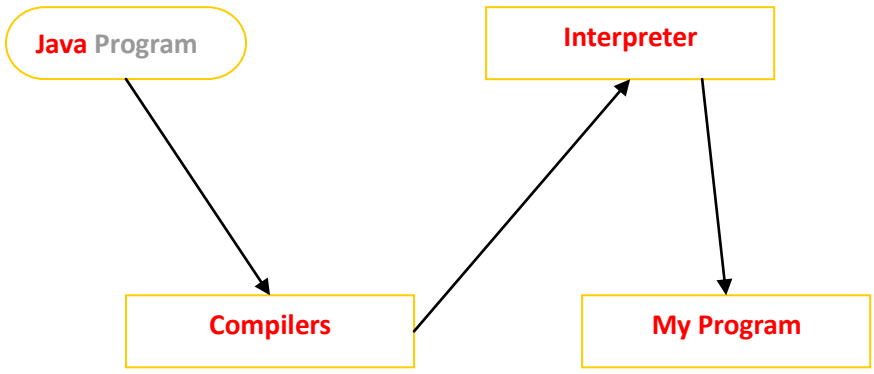
Java ha two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

| | |
|---|---|
| Simple | Architecture-neutral |
| Object-oriented | Portable |
| Distributed | High-performance |
| Interpreted | multithreaded |
| Robust | Dynamic |
| Secure | |

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.




You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a

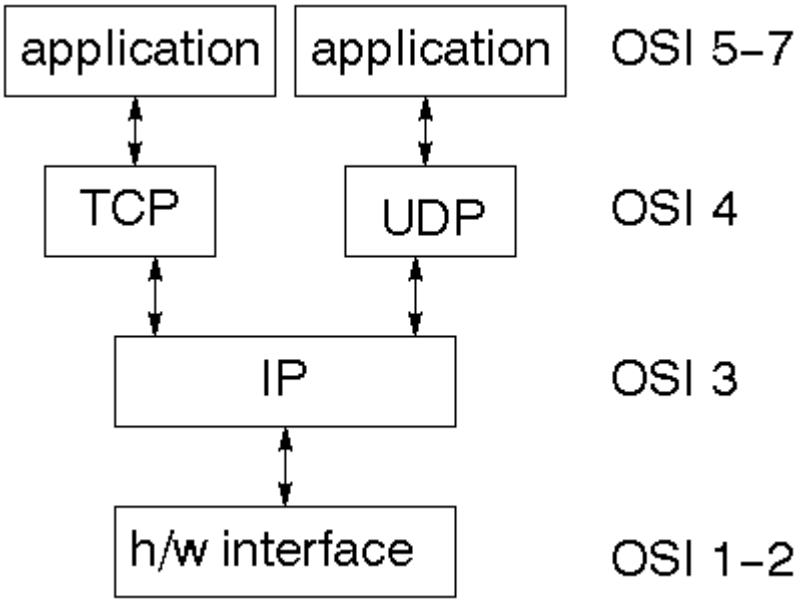
Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

***Networking***

### 3.9.5. TCP/IP stack

The TCP/IP stack is shorter than the OSI one:




TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

**IP datagram's**

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

**UDP**

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

**TCP**

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

**Internet addresses**

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

**Network address**

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

**Subnet address**

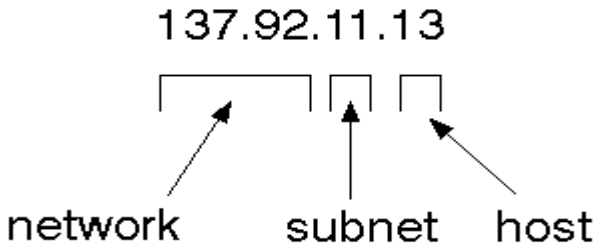
Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

**Host address**

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

**Total address**

The 32 bit address is usually written as 4 integers separated by dots.




**Port addresses**

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

**Sockets**

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

#include <sys/types.h>

#include <sys/socket.h>

int socket(int family, int type, int protocol);

Here "family" will be AF_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

### 3.9.6. JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

***1. Map Visualizations***

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

***2. Time Series Chart Interactivity***

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

***3. Dashboards***

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers,

bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

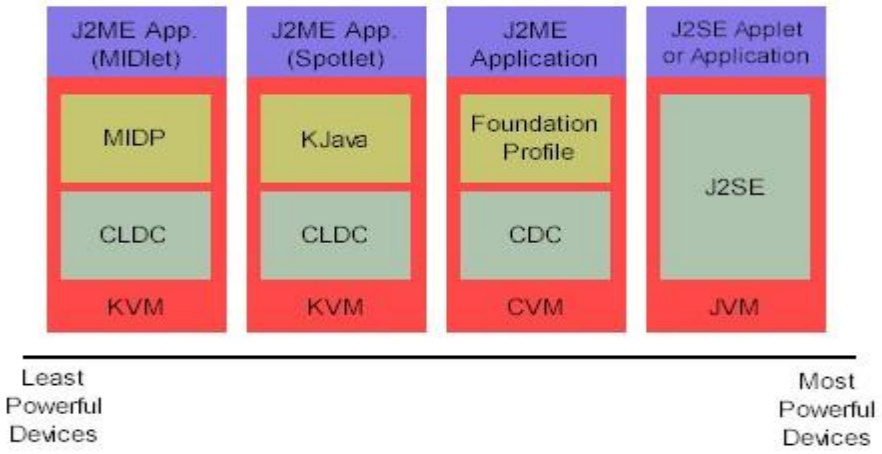
***4. Property Editors***

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

### 3.9.7. J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

**1. General J2ME architecture**




J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the  The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java

virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

**2.Developing J2ME applications**

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role preverification plays in this process.

**3.Design considerations for small devices**

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

* Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.

* Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.

* Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

**4.Configurations overview**

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

* **Connected Limited Device Configuration (CLDC)** is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

* **Connected Device Configuration (CDC)** is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

**5.J2ME profiles**

**What is a J2ME profile?**

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

**Profile 1: KJava**

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava contains a Sun-specific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is com.sun.kjava. We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

**Profile 2: MIDP**

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile application

development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

* java.lang

* java.io

* java.util

* javax.microedition.io

* javax.microedition.lcdui

* javax.microedition.midlet

* javax.microedition.rms

### 3.9.8. Client Server

**Over view:**

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine reveled that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from $200 million in 1992 to more than $1.2 billion in 1996.

Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE.

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

- **What is a Client Server**

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison dens there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file by file basis. The client server offers full relational database services such as SQL-Access, Record modifying, Insert, Delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

- **Why Client Server**

Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental an enterprise wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). Access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as "SLAVE-MASTER".Time-sharing changed the picture. Remote terminal could view and even change the central data, subject to access permissions. And, as the central data banks evolved in to sophisticated relational database with non-programmer query languages, online users could formulate adhoc queries and produce local reports with out adding to the MIS applications software backlog. However remote access was through dumb terminals, and the client server remained subordinate to the Slave\Master.

- **Front end or User Interface Design**

The entire user interface is planned to be developed in browser specific environment with a touch of Intranet-Based Architecture for achieving the Distributed Concept. The browser specific components are designed by using the HTML standards, and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

- **Communication or Database Connectivity Tier**

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity. The standards of three-tire architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

- **Features of The Language Used**

In my project, I have chosen *Java* language for developing the code.

### 3.9.9. Java Features:

- **Security:** Java provides a "firewall" (JVM) between network applications and the computer, protecting against viruses and malicious programs.
- **Portability:** Java bytecode enables programs to run on any platform with a JVM.

**Bytecode and JVM:**

- Java compiler outputs bytecode, a set of instructions executed by the Java Virtual Machine (JVM).
- The JVM interprets bytecode, allowing Java programs to run in various environments.
- Just-In-Time (JIT) compilers can compile bytecode into native code in real-time for improved performance. However, full ahead-of-time compilation is not possible due to runtime checks.

### 3.9.10. Virtual Machine (JVM):

- The JVM is a crucial part of Java, embedded in browsers or OS.
- Code is verified upon loading to prevent corruption using a class loader and bytecode verification.

**Java Development Process:**

Java source code (.java) is compiled by javac into bytecode (.class). The .class file is then loaded into the JVM, which interprets and executes the bytecode.

**Java Architecture:**

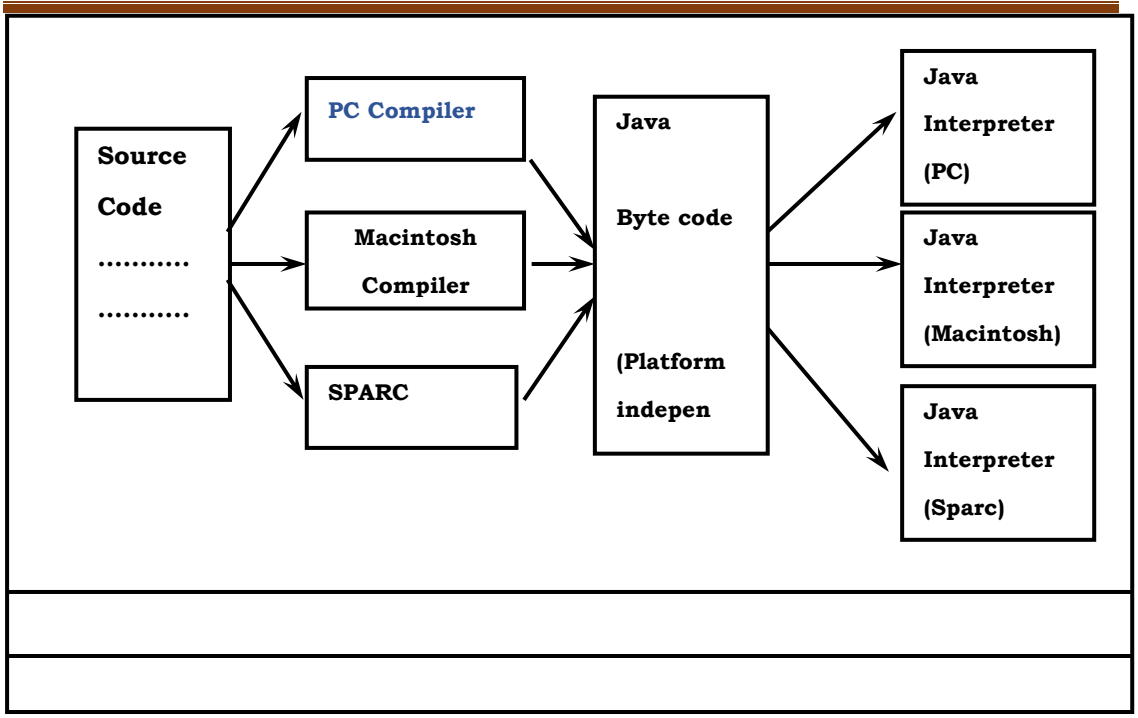
Provides a portable, robust, and high-performing environment. It achieves portability by compiling to JVM bytecode, interpreted by the runtime environment on each platform. Java is dynamic, loading code as needed from local or remote locations.

- **Compilation of code**

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

- ***Compiling and interpreting Java Source Code***

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or Sun SARC station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets.




**Java Key Aspects:**

- **Simplicity:** Designed for ease of learning and use, particularly for C++ programmers, removing or simplifying confusing C++ concepts.
- **Object-Oriented:** Clean and pragmatic object model that's easy to extend, with simple types as high-performance non-objects.
- **Robustness:** Designed for reliable execution in diverse systems, with strict type checking and automatic memory management.

**JavaScript Overview:**

- A scripting language for both client-side (browser) and server-side web development.
- Easier to learn than Java; statements embedded in HTML using <script> tags.
- Used for form validation, dynamic content, browser detection, and more.

**JavaScript vs. Java:**

- Separate languages; JavaScript enhances web pages, while Java handles complex applications.

**JavaScript Advantages:**

- Client-side and server-side scripting.
- More flexible than VBScript.
- Client-side default scripting language.

**HTML (HyperText Markup Language):**

- Language of the WWW for creating web pages with text, graphics, and hyperlinks.
- Not a programming language, but an application of SGML (Standard Generalized Markup Language).
- Uses tags to define how content is displayed. HTML tags are not case-sensitive.

**HTML Advantages:**

- Small document size, easy to transmit.
- Platform independent.
- Tags are not case-sensitive.

**3.9.11. Java Database Connectivity (JDBC):**

- Java API for executing SQL statements (Java Database Connectivity).
- Enables connection, SQL statement sending, and result processing with databases.

**JDBC vs. ODBC and other APIs:**

- JDBC is a "pure Java" solution needed to avoid native C code.
- More suitable for Java (object-oriented) than directly using ODBC (C-based).
- Simpler to learn than ODBC.
- Automatically installable, portable, and secure.

**Two-Tier and Three-Tier Models:**

- **Two-Tier:** Java applet/application directly connects to the database.
- **Three-Tier:** Commands sent to a "middle tier" that communicates with the database, allowing for control, higher-level APIs, and performance advantages.

**JDBC Driver Types:**

- JDBC-ODBC bridge plus ODBC driver
- Native-API partly Java driver
- JDBC-Net pure Java driver
- Native-protocol pure Java driver

**JDBC-ODBC Bridge:**

- JDBC driver that translates JDBC operations into ODBC, but using Pure Java JDBC Driver is best approach.

**Java Server Pages (JSP):**

- Technology for creating dynamic web pages using Java, open standards, and reusable components.
- Separates content generation from presentation.

**JSP Features:**

- **Portability:** Runs on any web server with JSP engine support.
- **Components:** Supports reusable Java components (JavaBeans, Servlets) and embedded scripting.
- **Processing:** JSP files (HTML with JSP tags) are parsed and translated into Servlets on the server.

**Access Models:**

- Client requests JSP directly or via a Java Bean.

**JSP Execution Steps:**

1. Client requests JSP file.
2. Web server receives request.
3. Request is transferred to the JSP engine.
4. JSP engine converts JSP to Servlet.
5. Servlet executes, results given back to the web server and transferred back to the client.

**JDBC Connectivity in J2EE:**

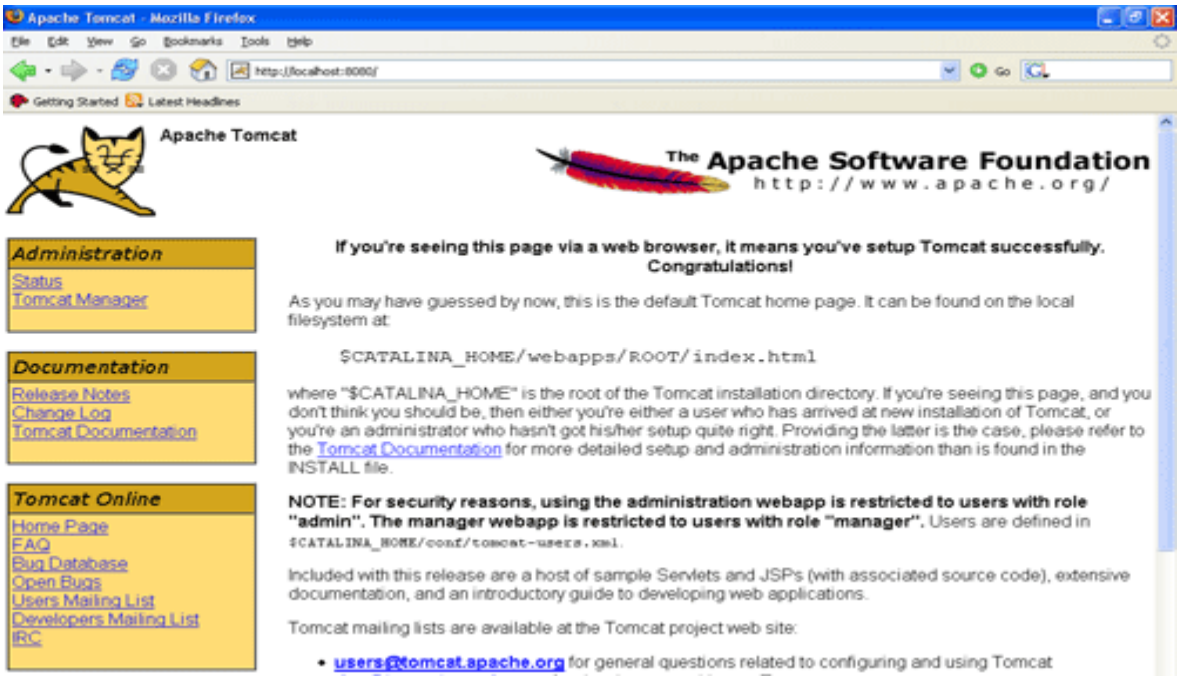
- Provides database-independent connectivity to tabular data sources.

**JDBC Capabilities:**

1. Connection and authentication to a database server.
2. Transaction management.
3. SQL statement transfer.
4. Stored procedure execution.
5. Result inspection and modification.

**Tomcat 9.0 web server**

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs Web logic, is one of the popular application server. To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc to run your application.



# CHAPTER IV

# IMPLEMENTATION

## 4.1Hardware & Software Selection :

**Hardware Requirements:**

- Processor: Intel Core i3/i5 - 3.0 GHz or higher
- RAM: Minimum 8 GB
- Hard Disk: Minimum 256 GB SSD
- Network: High-speed Internet connection for cloud interactions

**Software Requirements:**

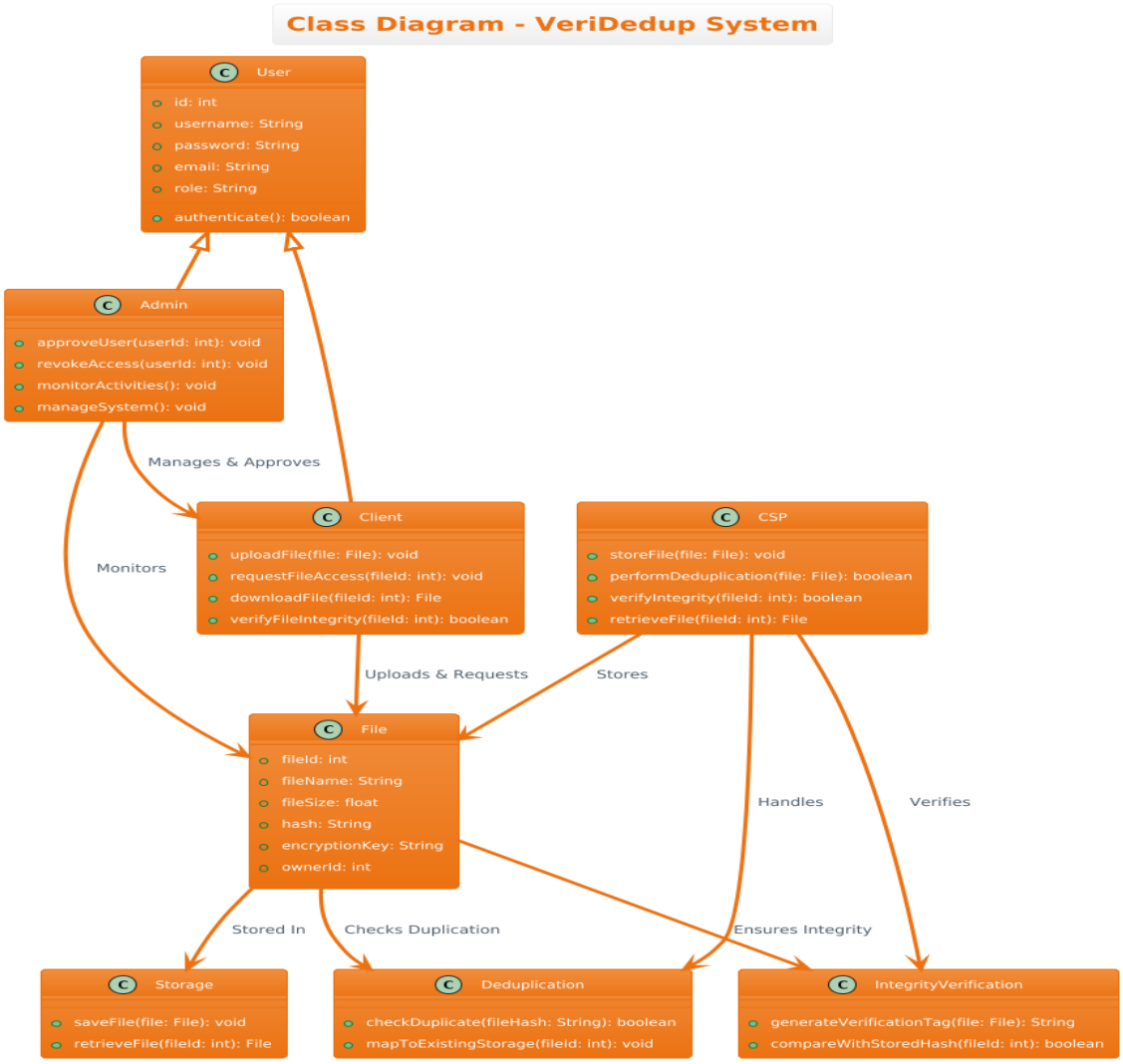
- Operating System: Windows 10/11
- Programming Language: Java (JSP, Servlets)
- Web Technologies: HTML, CSS, JavaScript
- Database: MySQL (WAMP Server 3.4)
- Web Server: Apache Tomcat Server 9.0
- Security Features: AES Encryption for data protection, Authentication mechanisms
- Development Environment:
    - IDE: Eclipse IDE
    - Development Kit: JDK 11 or later
    - Version Control: GitHub/Git for source code management
- Documentation Tools: MS Office (Word, PowerPoint), Google Docs

This configuration ensures optimal performance, security, and compatibility with the VeriDedup system, meeting all technical feasibility and operational requirements.

### 4.2 UML Diagrams :

#### 4.2.1 Class Diagram :

The class diagram for VeriDedup represents the core components of the system, detailing their attributes, methods, and relationships. The system is structured into user management, file handling, deduplication, integrity verification, and storage operations, ensuring a secure, efficient, and verifiable cloud-based data deduplication framework.




The User class serves as the base class for both Client and Admin, containing common attributes such as id, username, password, email, and role. Clients interact with the system to upload, download, and verify files, while Admins oversee user approvals, monitoring activities, and system security. Clients can upload files, check for duplication, request file access, and verify integrity, whereas Admins manage system settings, approve or revoke access, and monitor logs for security compliance.

The File class represents uploaded files and includes attributes like fileId, fileName, fileSize, hash, encryptionKey, and ownerId. This class interacts with the Deduplication class, which checks for duplicate files using cryptographic hashes and links identical files to prevent redundant storage. The Cloud Storage Provider (CSP) class is responsible for storing files, executing deduplication, verifying file integrity, and retrieving data when requested by authorized users.

The IntegrityVerification class ensures the authenticity of stored files by generating verification tags and comparing stored file hashes with newly uploaded files. This process detects unauthorized modifications and guarantees data integrity.

Additionally, the Storage class handles physical storage and retrieval, ensuring that all stored files remain encrypted and secure.

The relationships among these classes define the system's structure. The User class acts as the parent for Client and Admin, while Clients interact with the File class through upload, download, and integrity verification operations. Admins manage user approvals and oversee client interactions. The File class connects with Deduplication for checking duplicates and IntegrityVerification for ensuring data consistency. The CSP class is central to storage, deduplication, and file retrieval, while the Storage class handles physical file persistence.

Overall, the VeriDedup class diagram represents a modular and scalable cloud-based deduplication system. It ensures strong authentication mechanisms, efficient file handling, cryptographic deduplication, and reliable integrity verification. The structured architecture allows for easy maintenance, high security, and optimal storage utilization, making VeriDedup a robust cloud storage solution.

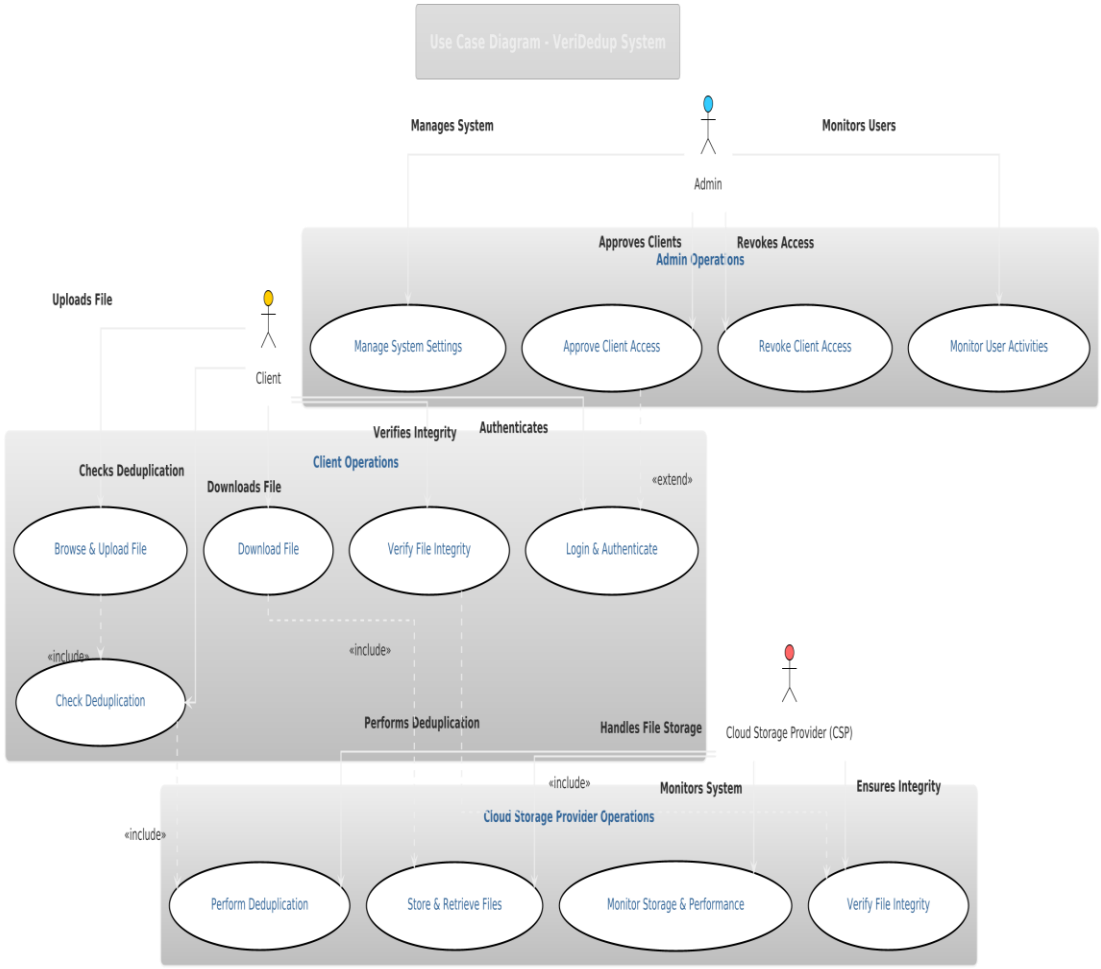
### 4.2.2 Use Case Diagram :

The Use Case Diagram for VeriDedup represents the system's functionality and the interactions between different users, including clients, administrators, and cloud storage providers (CSPs), for file deduplication, integrity verification, and access management. The system is designed to optimize cloud storage by eliminating redundant files, ensure data integrity through cryptographic verification, enhance security using authentication and access control, and improve performance through continuous monitoring and system optimization. Clients, administrators, and cloud providers work together efficiently to maintain a secure, deduplicated, and verifiable file storage environment.

Clients act as the primary users who upload, download, and verify files while ensuring data security. Their key actions include browsing and uploading files, checking for deduplication before storage, downloading stored files, verifying file integrity, and securely logging into the system for access. These processes help prevent redundant file storage and ensure that data remains intact. Administrators manage user access, enforce security policies, and oversee system settings. Their responsibilities include managing system configurations, approving or revoking client access, and monitoring user activities to ensure compliance with security protocols. Through these actions, admins play a crucial role in maintaining a secure and well-regulated cloud storage system.

Cloud Storage Providers (CSPs) handle the backend processes of data storage, optimization, and security. They are responsible for performing deduplication upon file uploads, storing and retrieving files efficiently, monitoring overall system performance, and verifying file integrity to prevent unauthorized modifications. These processes ensure that storage resources are utilized efficiently and securely. The relationships between these system components define the necessary dependencies,

such as ensuring that the deduplication check is performed before storing a file, verifying file integrity before downloading, and authenticating users before granting access. These dependencies ensure that deduplication, security, and access control processes occur in the correct sequence, maintaining the system's efficiency and reliability.



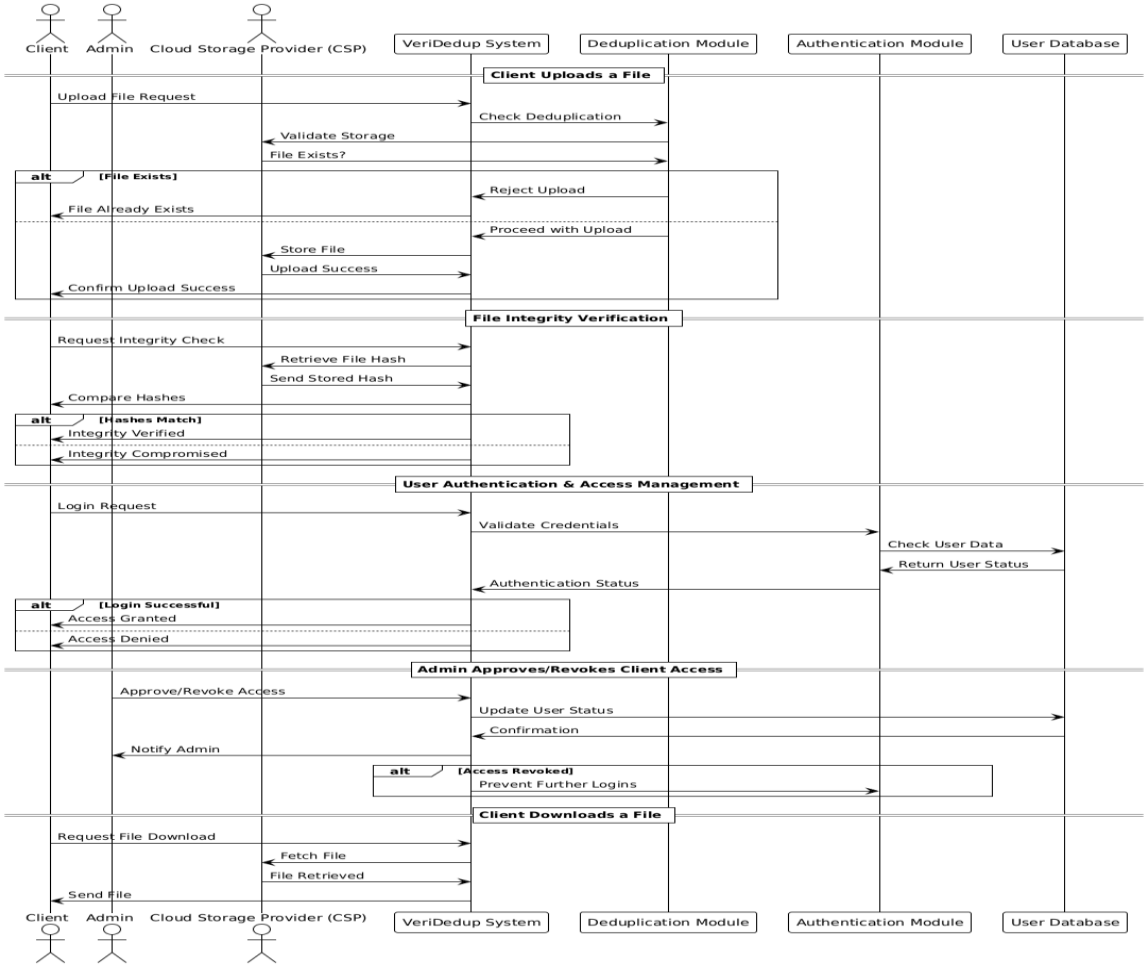

The VeriDedup system provides significant advantages, including optimized cloud storage by reducing redundant files, enhanced security and compliance by ensuring only authorized users access data, cryptographic integrity verification to prevent unauthorized modifications, improved system performance through real-time deduplication, and scalability to handle increasing users and data efficiently. The system is highly applicable in real-world scenarios, such as cloud storage optimization used by platforms like Dropbox, Google Drive, and AWS, enterprise data management to prevent unnecessary storage costs, cybersecurity applications in industries like healthcare and finance where data integrity is crucial, and backup and disaster recovery solutions that prevent duplicate backups while maintaining consistency.

Overall, the VeriDedup Use Case Diagram effectively represents the interactions between different actors and system components, ensuring secure and efficient cloud

storage. Clients benefit from a seamless and secure file storage experience, administrators maintain security and user management, and cloud providers handle deduplication and data integrity. By integrating deduplication, verification, and user access control, VeriDedup delivers a reliable, secure, and efficient cloud storage solution.

### 4.2.3 Sequence Diagram :

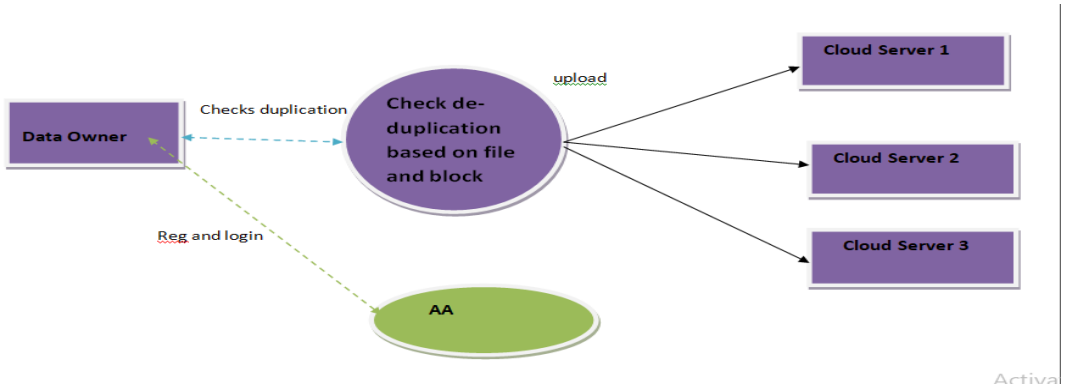
The VeriDedup System follows a structured and secure workflow for file management, integrity verification, and user authentication to ensure efficient and protected cloud storage. When a client uploads a file, the system performs a deduplication check using the Deduplication Module. If an identical file already exists in the Cloud Storage Provider (CSP), the system prevents redundant storage and links the client to the existing file. However, if the file is unique, it is securely encrypted and stored in the CSP, and the client receives a confirmation of successful upload. This deduplication mechanism optimizes storage efficiency while reducing bandwidth and computational costs. Additionally, for file integrity verification, the client can request a check, prompting the system to retrieve the stored cryptographic hash from the CSP and compare it with the newly generated file hash. If the hashes match, the file remains authentic and unaltered; otherwise, a warning is issued indicating potential data corruption or tampering.



User authentication follows a multi-layered security process where login credentials are forwarded to the Authentication Module, which verifies them against the User Database (UDB). If the provided credentials are valid, the client gains access to the system; otherwise, access is denied to prevent unauthorized usage. Admins play a crucial role in overseeing system security by managing client access, approving new users, and revoking access for unauthorized activities. All modifications to user permissions are updated in the UDB to maintain a secure access control policy. When a client requests to download a file, the system retrieves the requested file from the CSP and securely delivers it, ensuring that only authorized users can access stored data. This structured sequence ensures seamless interaction, robust security, and data integrity, making VeriDedup a reliable, scalable, and efficient cloud storage solution.

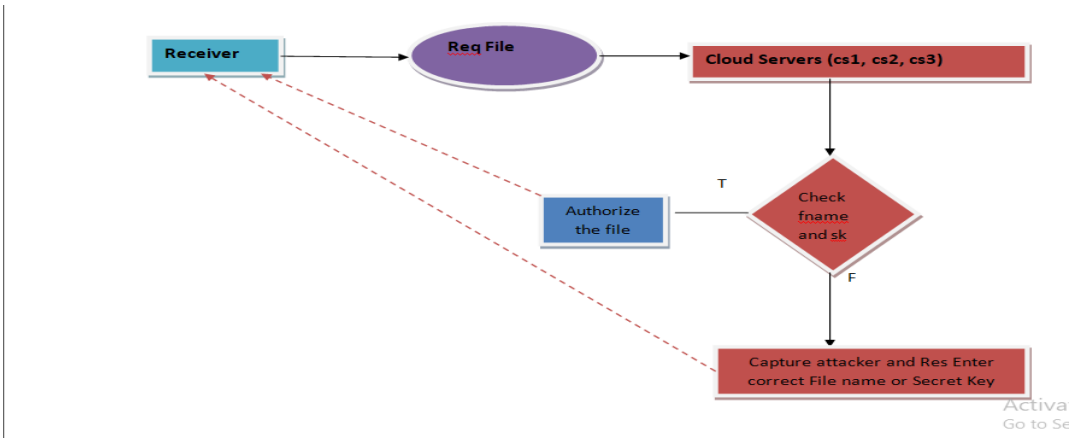
### 4.2.4 Data Flow Diagrams :

**Level 0 - High-Level Process Overview :**



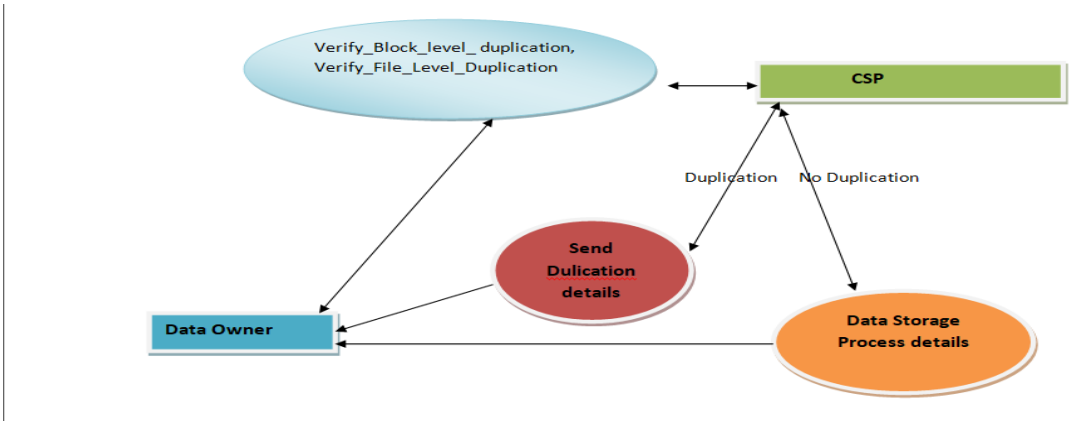

The data owner initiates the file upload process, which is first analyzed by the Deduplication Module to determine if the file or its blocks already exist. If duplication is detected, the system prevents redundant storage and notifies the data owner with duplication details. If no duplication is found, the file is securely stored in the Cloud Storage Provider (CSP), and the system updates the storage records accordingly. This ensures efficient storage utilization, reduces redundancy, and enhances overall system performance.

**Level 1 - File Request & Authorization Process :**



When a file retrieval request is made, the system processes it and directs the request to the Cloud Storage Servers (CS1, CS2, CS3). Before granting access, the system verifies the file name and the associated secret key to ensure secure access control. If the verification is successful, the requested file is retrieved and securely delivered to the requester. If the authentication fails, access is denied, preventing unauthorized access. This mechanism ensures data security, confidentiality, and controlled access to stored files.

**Level 2 - Deduplication & Storage Workflow :**




The Deduplication Module conducts thorough verification at both the file level and block level before proceeding with storage. The data owner is responsible for initiating duplication checks before uploading files. Once verified, the system distributes the data across multiple cloud servers (Cloud Server 1, Cloud Server 2, and Cloud Server 3) for enhanced availability and redundancy. Additionally, the Authentication and Access Management (AA) module ensures that only authorized users can manage and access stored data. This structured process guarantees optimized cloud storage, improved system efficiency, and secure data management.

This data flow design enforces robust deduplication, secure file storage, and seamless access control, ensuring a highly efficient and protected data management system within VeriDedup.

## 4.3. Coding :

**Welcome.jsp :**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>

<!DOCTYPE html>

<html lang="en">

<head>

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>VeriDedup</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Welcome to VeriDedup</h1>
        <nav>
            <ul>
                <li><a href="welcome.jsp" class="active">Home</a></li>
                <li>
                    <a href="#">Login</a>
                    <ul>
                        <li><a href="admin.jsp">Admin</a></li>
                        <li><a href="client.jsp">Client</a></li>
                        <li><a href="host.jsp">CSP</a></li>
                    </ul>
                </li>
                <li><a href="register.jsp">Register</a></li>
                <li><a href="contact.jsp">Contact Us</a></li>
                <li><a href="about.jsp">About Us</a></li>
            </ul>
```

```
    </nav>

  </header>

  <section>

    <h2>Project Abstract: VeriDedup</h2>

      <h3>A Verifiable Cloud Data Deduplication Scheme with Integrity and Duplication Proof</h3>

    <h4>Problem Statement</h4>

      <p>Cloud storage providers (CSPs) often use deduplication to eliminate redundant data and optimize storage. However, two critical challenges arise:</p>

    <ul>

      <li><strong>Integrity Risks:</strong> The CSP might tamper with or delete deduplicated data, compromising its authenticity.</li>

       <li><strong>Duplication Fraud:</strong> A dishonest CSP could falsely claim non-duplicate data to charge users extra for storage they don’t use.</li>

    </ul>

    <h4>Innovative Solution</h4>

     <p><strong>VeriDedup</strong> introduces a groundbreaking framework to ensure both <strong>data integrity</strong> and <strong>duplication-proof accountability</strong> in cloud storage. By combining cryptographic protocols and novel verification mechanisms, it addresses the limitations of existing systems, such as brute-force attack vulnerabilities and inflexible tag generation.</p>

    <h4>Key Features</h4>

    <ul>

      <li><strong>Tag-Flexible Integrity Check Protocol (TDICP):</strong>

        <ul>

          <li>Enables users to generate <strong>unique verification tags</strong> while supporting deduplication.</li>
```

<li>Uses Private Information Retrieval (PIR) to embed "note sets" (randomized sequences) into encrypted data for tamper-proof integrity checks.</li>

</ul>

</li>

<li>User-Determined Duplication Check Protocol (UDDCP):

<ul>

<li>Leverages Private Set Intersection (PSI) to let users independently verify duplication claims, preventing CSP fraud.</li>

<li>Ensures the CSP cannot falsify duplication results to overcharge users.</li>

</ul>

</li>

</ul>

<h4>Security & Efficiency</h4>

<ul>

<li>Formal Proofs: Rigorous analysis confirms resistance to brute-force attacks and CSP cheating.</li>

<li>Real-World Validation: Simulations based on actual datasets show 20% faster integrity checks and 30% lower communication costs compared to prior solutions like StealthGuard.</li>

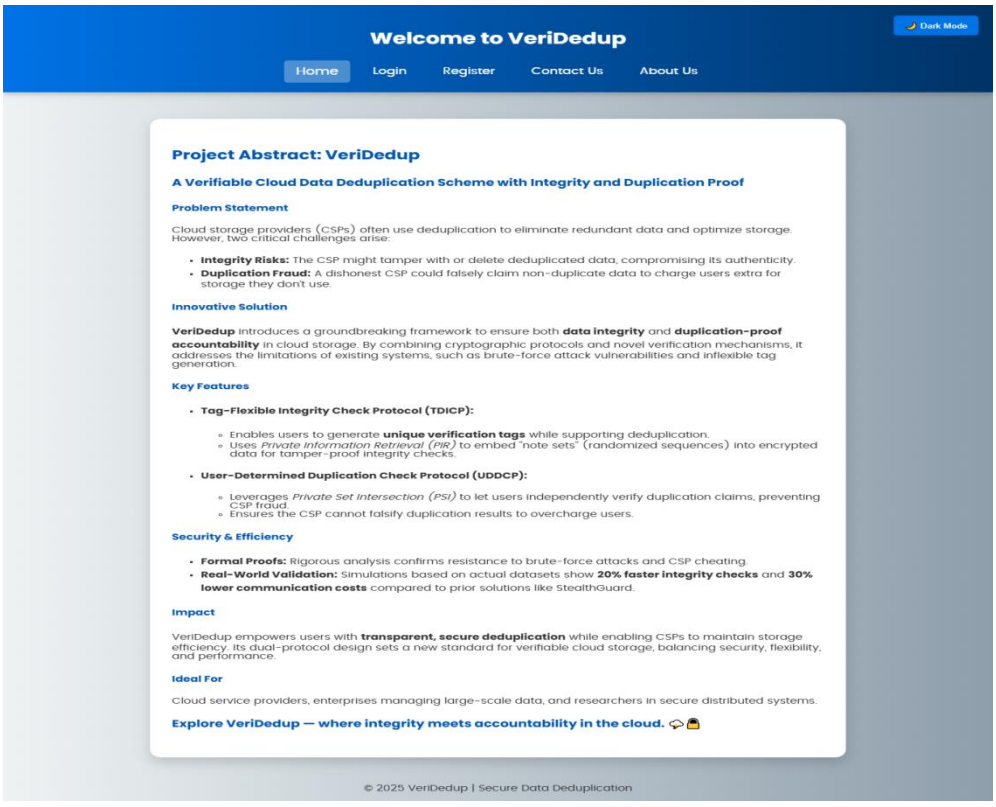
</ul>

<h4>Impact</h4>

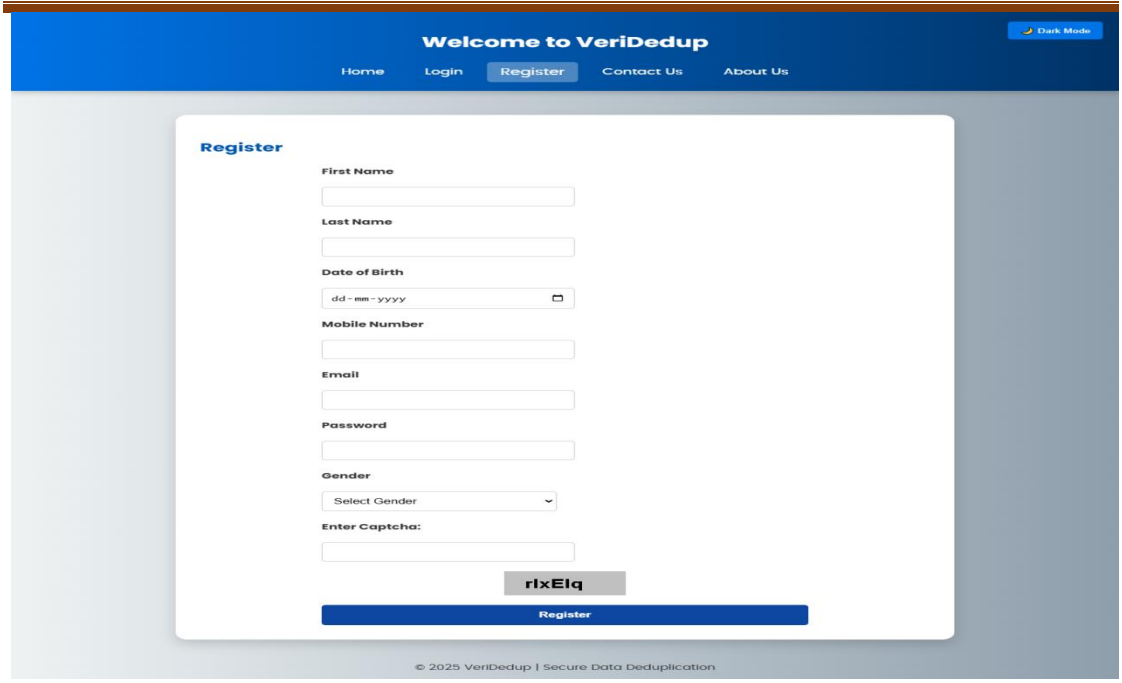
VeriDedup empowers users with transparent, secure deduplication while enabling CSPs to maintain storage efficiency. Its dual-protocol design sets a new standard for verifiable cloud storage, balancing security, flexibility, and performance.

```
    <h4>Ideal For</h4>

        <p>Cloud service providers, enterprises managing large-scale data, and researchers in secure distributed systems.</p>

        <h3>Explore VeriDedup — where integrity meets accountability in the cloud. □□□</h3>

    </section>

    <script src="script.js" defer></script>

    <footer>

        <p>© 2025 VeriDedup | Secure Data Deduplication</p>

    </footer>

</body>

</html>
```

**On executing this code we will enter into our Home Page which looks like :**

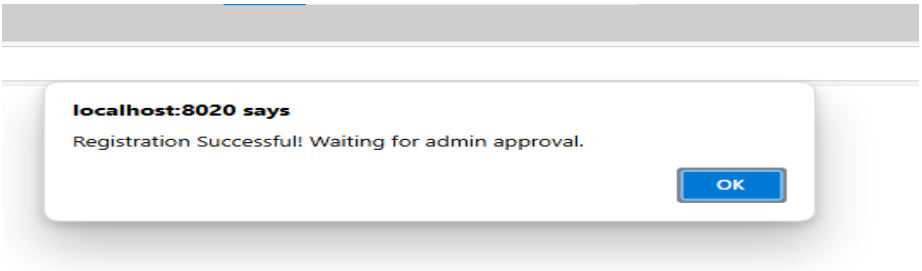



**On clicking register button we will navigate to register page which looks like :**

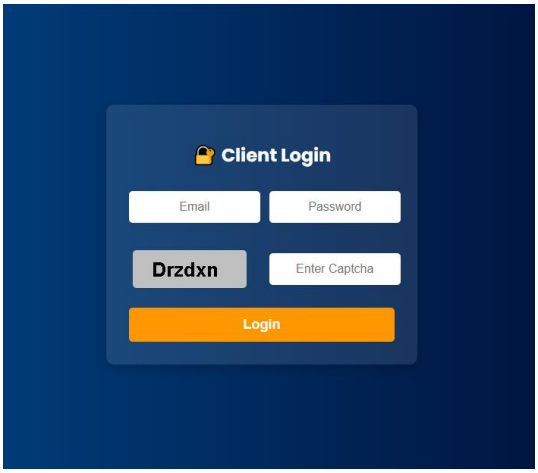



**CAPTCHA in VeriDedup :** CAPTCHA prevents bots from accessing login, registration, and sensitive operations. We use Google reCAPTCHA to block automated attacks. It activates after multiple failed logins or before high-privilege tasks, ensuring only real users interact with the system.

**A**fter entering all the required details in the register page and on clicking register button it will give us a popup which shows us waiting for Admin Approval.

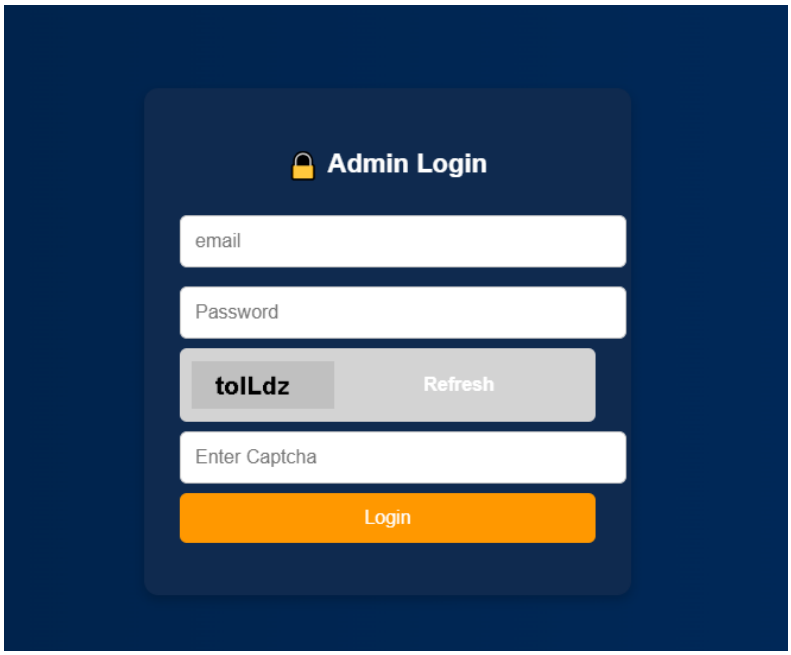



Now Coming to the login page we have 3 different types of **Login Pages** which are for **Client, Admin & CSP** respectively.

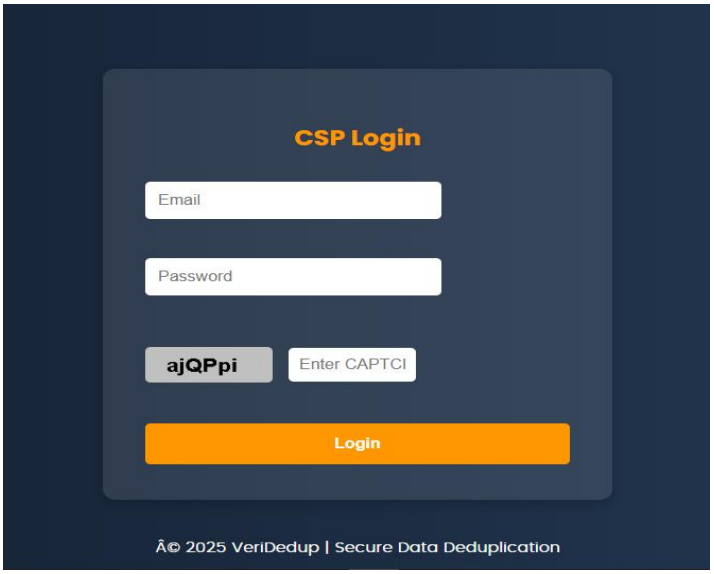



**Client Login :**

**Admin Login :**




**CSP Login :**




**O**n entering the required Login Credentials & on Clicking Login it establishes the Connection with our VeriDedup Database to fetch and authenticate the Login Credentials.

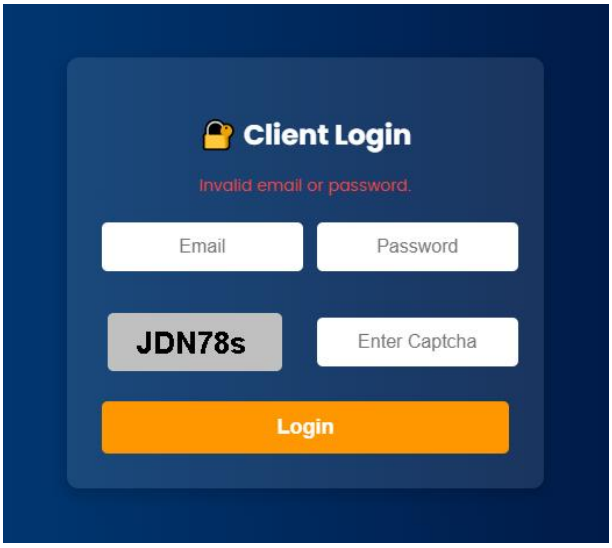
## Connection Establishment

<%@ page import="*java.sql.*"%>
<%@ page import="*java.util.*" %>
<%
Connection connection = null;
try {

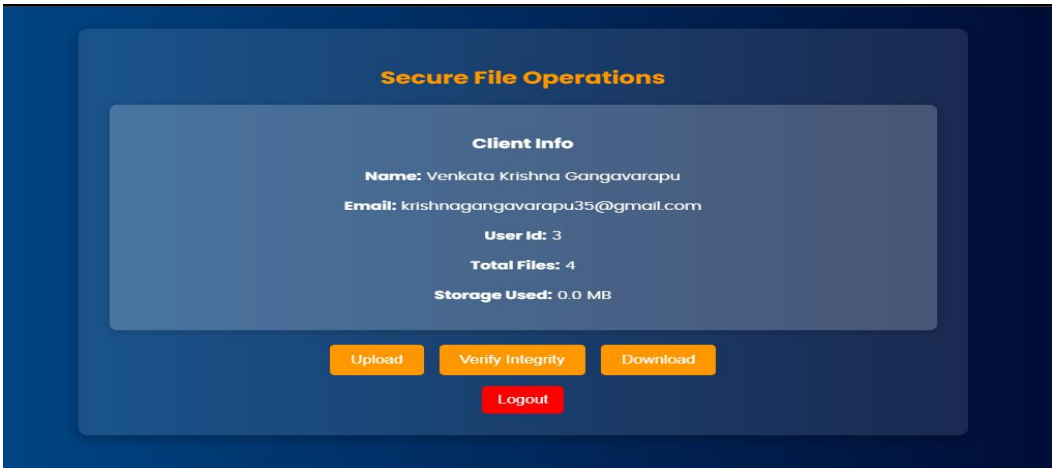
```
Class.forName("com.mysql.jdbc.Driver");
Connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/VeriDedup",
"root","");
String sql="";
}
catch(Exception e)
{
System.out.println(e);
}
%>
```

If the given login credentials are correct then it navigates to their respective Dashboards otherwise it again redirects to the login page by showing enter valid Credentials.

**Login Failure :**

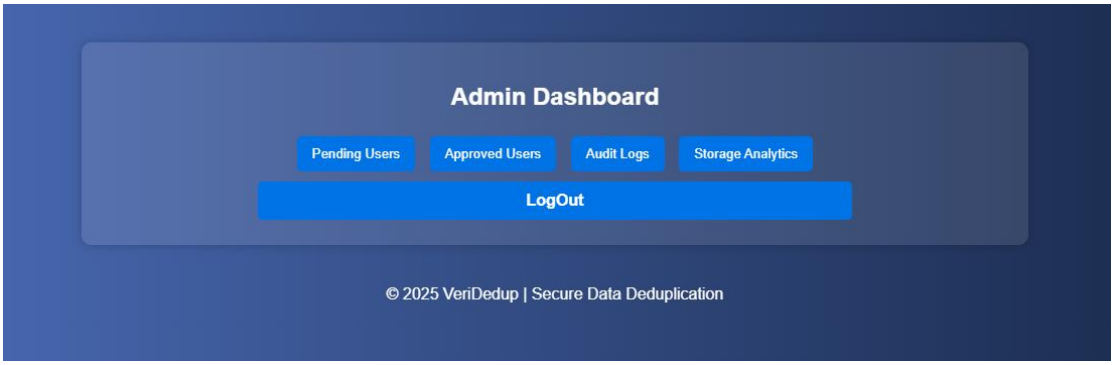



**Client Dashboard :**



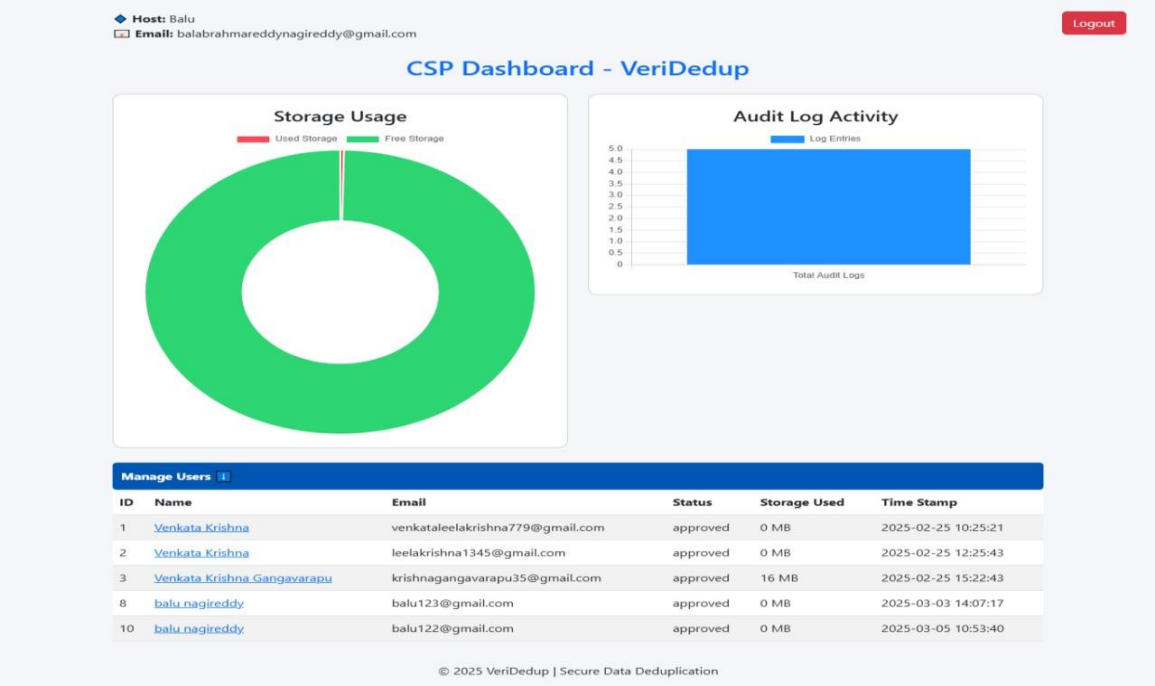
In this Client Dashboard we can perform the file uploading with data encryption , Verifying the files integrity and deduplication by giving the User Id and Verification tag & we can download the required files by clicking on Download button which performs both decryption and download operation. Here the encryption and decryption are performed by using AES Algorithm.

**Admin Dashboard :**




In the Admin Dashboard we can see the pending users , approved user and the audit logs by clicking on the respective buttons. The admin can perform all kinds operations like he can approve the pending user or else he can reject them and also if incase he need to revoke the permissions from any of the clients that operation also can be performed by the admin through he authority of revoking and all the operations performed by the admin are clearly recorded in the Audit Logs table with specific time stamps.

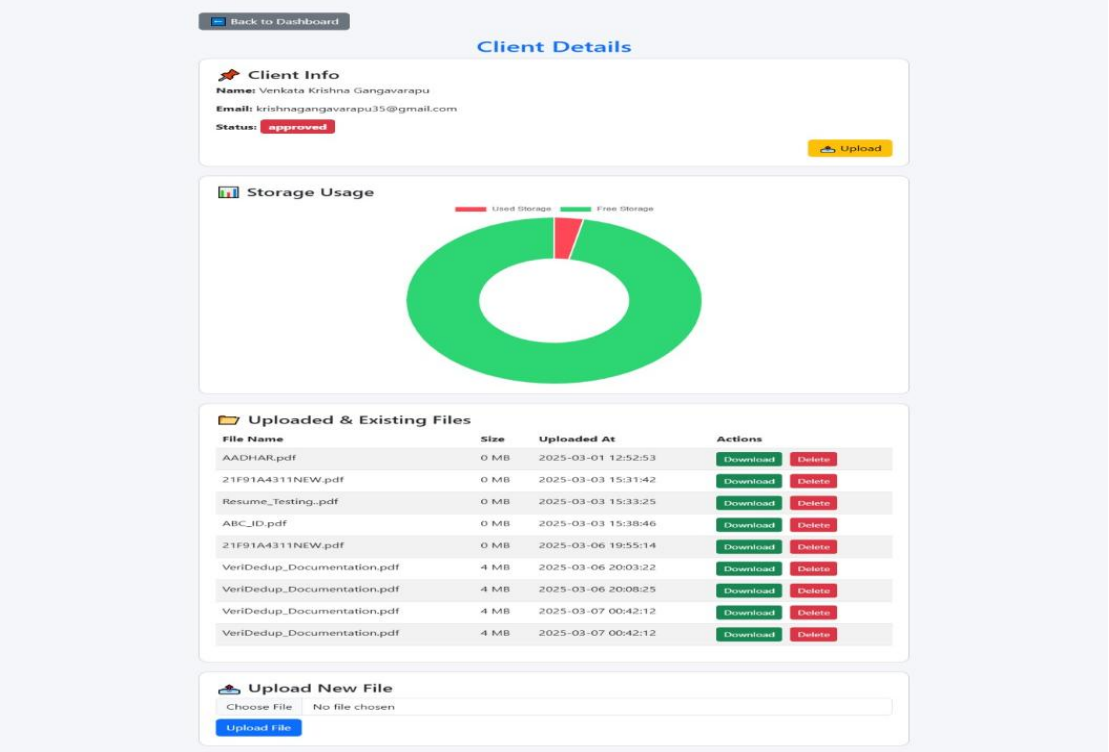
**CSP Dashboard :**



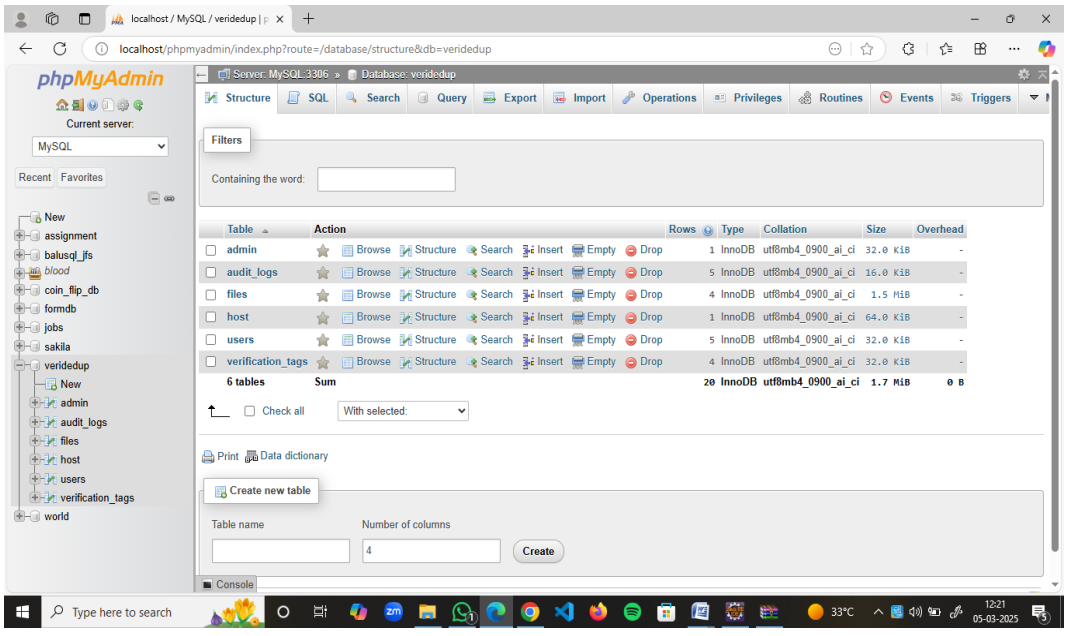
This is our CSP Dashboard in which the Cloud Storage Provider can see the audit logs, storage usage by all the users and he also can see the existing clients of his cloud.

**O**n Clicking any of the Client name we can all his details like this :

## Client Details :




## Database Structure :



## 4.4. System Integration :

### 4.4.1. Overview

System integration in the VeriDedup project ensures that all components, including Java, JSP, Tomcat, WAMP (MySQL), Eclipse, AES encryption, authentication mechanisms, and web technologies (HTML, CSS, JavaScript), work together seamlessly to provide a secure and verifiable cloud data deduplication system. The integration process connects different modules—Client, Admin, and Cloud Storage Provider (CSP)—to facilitate data deduplication, integrity verification, and secure file storage.

### 4.4.2. Components of System Integration

1. **Backend and Web Application Integration (Java, JSP, Tomcat, Eclipse)**

- The backend is developed using Java Servlets and JSP to handle user requests and process data.
- Apache Tomcat is used as the web server to deploy and execute JSP pages and Servlets.
- Eclipse IDE is used for coding, debugging, and project management.
- Servlets handle HTTP requests for file uploads, downloads, authentication, and deduplication verification.

**Integration Flow**

- Users interact with JSP-based web pages for file uploads and verification.
- Java Servlets process these requests and interact with MySQL for database operations.
- Responses are dynamically generated in JSP and sent back to users through the web interface.

2. **Database Connectivity (WAMP - MySQL, JDBC)**

- WAMP (Windows, Apache, MySQL, PHP) is used for managing the MySQL database, which stores user credentials, file metadata, deduplication records, and verification logs.
- JDBC (Java Database Connectivity) is implemented to establish a secure connection between the Java application and MySQL.

**Integration Flow**

- JDBC connects Java Servlets with MySQL, executing SQL queries for user authentication, file storage, and deduplication tracking.

- Tables in MySQL store file hashes, metadata, and verification logs for deduplication validation.
- Admin and users can retrieve records from the database for verification and monitoring.

**3. Secure File Deduplication and Verification (AES, Authentication Mechanisms)**

- AES (Advanced Encryption Standard) is used for encrypting files before uploading them to prevent unauthorized access.
- User authentication is implemented using session management, password hashing, and access control.
- Deduplication verification mechanisms ensure that identical files are not stored multiple times, reducing storage redundancy.

**Integration Flow**

- Before uploading, the system checks for duplicate files by generating a file hash and comparing it with stored hashes.
- If a duplicate exists, the system provides access to the existing file instead of storing another copy.
- If no duplicate is found, the file is encrypted using AES and securely stored.
- Users can verify file integrity by comparing cryptographic verification tags.

**4. Web Frontend Integration (HTML, CSS, JavaScript, JSP)**

- The user interface is developed using HTML, CSS, and JavaScript, with JSP handling dynamic content.
- AJAX is used to update the UI dynamically without requiring full-page reloads.
- User dashboards (Client/Admin) are designed for seamless interaction with the backend.

**Integration Flow**

- Users interact with JSP-based web pages for file uploads, integrity checks, and deduplication monitoring.
- AJAX ensures smooth page updates when users perform actions like file verification.
- CSS ensures a responsive and modern UI for a better user experience.

5. **Deployment and Testing (Tomcat, Eclipse, MySQL, Security Audits)**

- The entire system is deployed on Apache Tomcat, ensuring smooth execution of JSP and Servlets.
- Security testing is performed to verify authentication mechanisms, encryption, and deduplication security.
- Unit and integration testing ensure that all components work seamlessly together.

**Integration Flow**

- Modules are tested independently (unit testing) and then combined (integration testing).
- Security audits check for vulnerabilities in authentication and data encryption.
- Performance tests measure deduplication efficiency and system response time.

System integration in VeriDedup ensures a fully functional, secure, and verifiable cloud data deduplication system. By integrating Java, JSP, WAMP (MySQL), Tomcat, Eclipse, AES encryption, authentication mechanisms, and web technologies, VeriDedup provides seamless data processing, secure storage, and efficient deduplication verification.

# CHAPTER – V

# TESTING

## 5.1. Testing Description

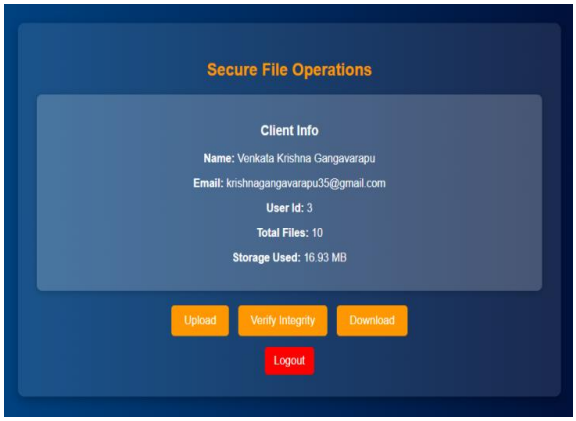
System testing ensures that the VeriDedup platform meets all functional, security, and performance requirements before deployment. Given its secure data deduplication, integrity verification, and auditing functionalities, rigorous testing is conducted to validate its behavior under different conditions. Since the system is built using Java (JSP, Servlets), MySQL (WAMP Server 3.4), Apache Tomcat 9.0, and AES encryption, testing focuses on security, performance, integration, and usability across its different components. The testing process guarantees that file uploads, deduplication, encryption, authentication, and audit logging function reliably while maintaining data security.

## 5.2. Test Cases & Scenarios :

**Test Case 1 : Client Login**

The Client tries to login into his Dashboard page through client.jsp which is the login page for him.

**Scenario 1 : Successful Login**




**Inputs :** The Email, Password & Captcha are the required credentials for this.

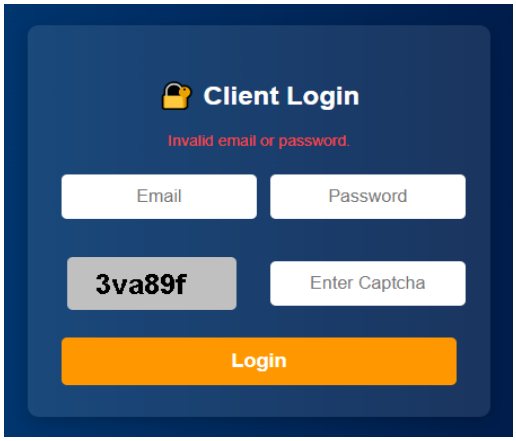
Email : krishnagangavarapu35@gmail.com

Password : 21F91a4661@

**Expected Output :** Logged in Successfully and Shows us his Client Dashboard.

**Actual Output :** Successfully Logged in.

**Scenario 2 : Login Failure**




**Inputs :** Email is Correct but trying with Wrong Password.

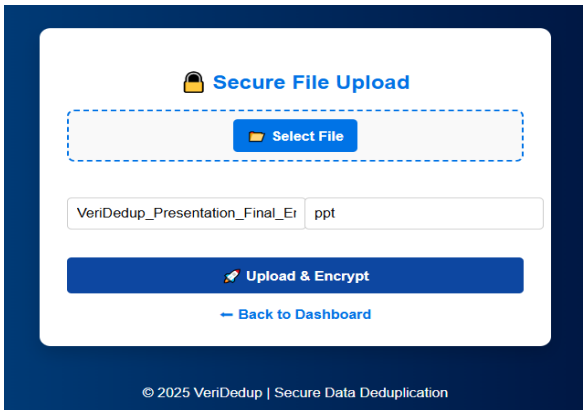
Email : krishnagangavarapu35@gmail.com

Password : 21F91a461@

**Expected Output :** Failure in Login

**Actual Output :** Login Failed.

**Test Case 2 : Client File Upload**



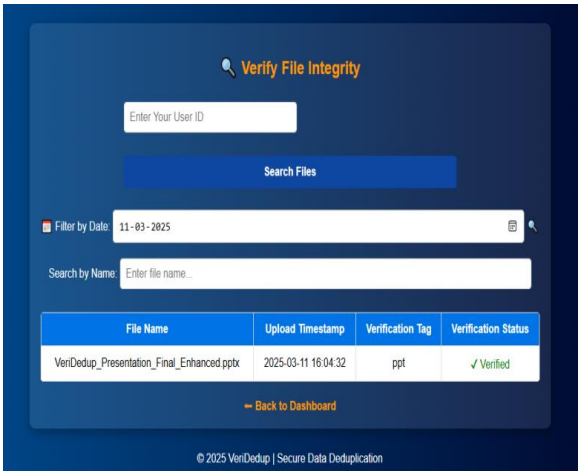

**Scenario :** Our Clients tries to upload file into his Storage Space (Cloud) in the Encrypted Form.

**Inputs :** File, Integrity Tag

**Expected Output :** File Upload Successfully and the File count increases along with the Storage Space.

**Actual Output :** Successful.

**Test Case 3 : Integrity Verification**



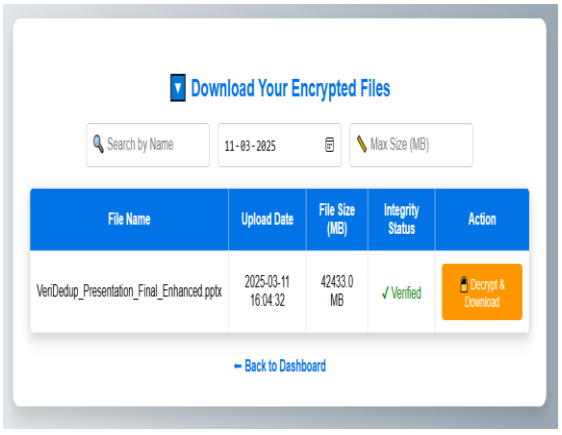

**Scenario :** Checks the Files Integrity by entering the user Id as the search Credential.

**Inputs :** User Id

**Expected Output :** All the files that are uploaded by the user with the Given User Id.

**Actual Output :** Successful.

**Test Case 4 : File Download**




**Scenario :** Downloads Required file in the Decrypted form.

**Inputs :** Non

**Expected Output :** File Downloaded Successfully in the Decrypted Form.

**Actual Output :** Downloaded Successfully.

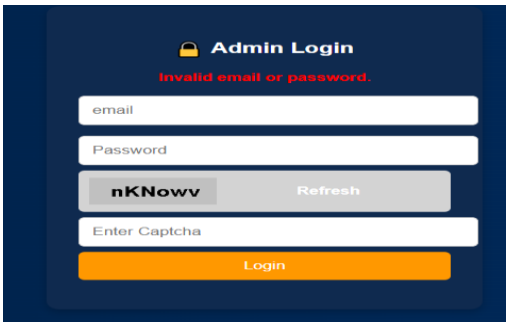
**Test Case 5 : Admin Login**


**Scenario 1 : Login Successful.**

**Inputs :** Correct Email , Password & Captcha

**Expected Output :** Successful Login and Admin Dashboard

**Actual Output :** Login Successful.




**Scenario 2 : Login Failed.**

**Inputs :** Wrong Email, Password, Captcha

**Expected Output :** Login Failed .

**Actual Output :** Login Failed.

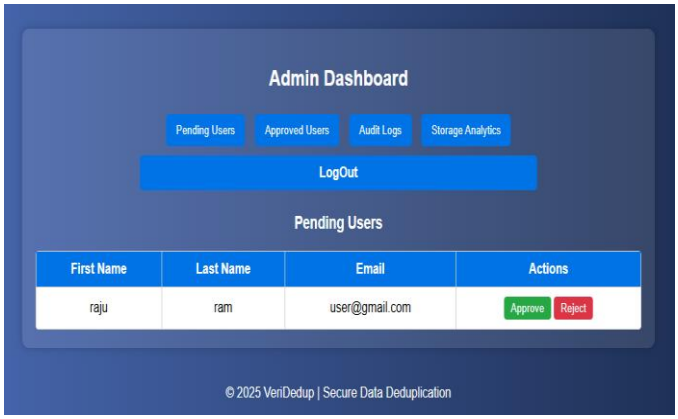
**Test Case 6 : Client Approval & Reject**

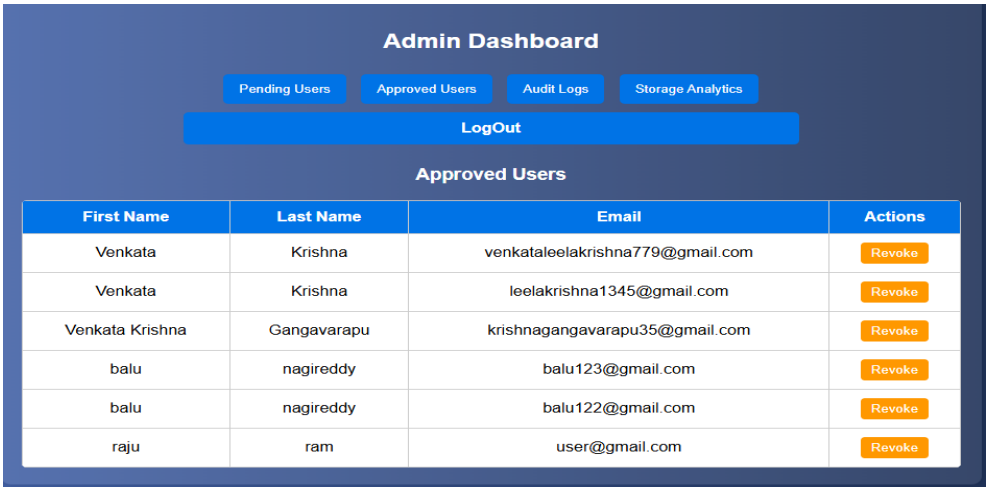
**Scenario :** Pending Clients Approval or Rejection

**Inputs :** Non but need to Click on Approve or Reject.

**Expected Output :** Approved/ Rejected Successfully.

**Actual Output :** Approved Successfully.



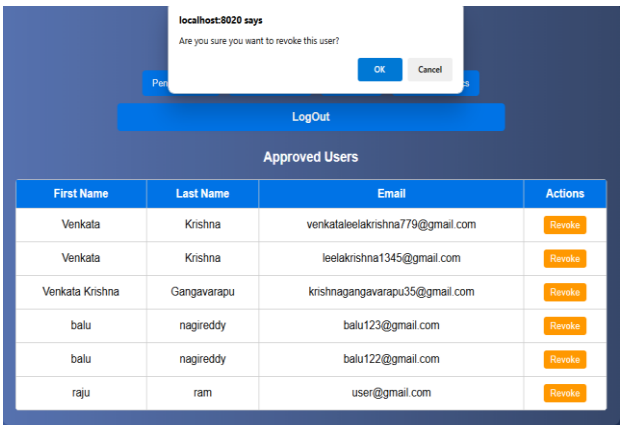

**Test Case 7 : Client Permissions Revoke**

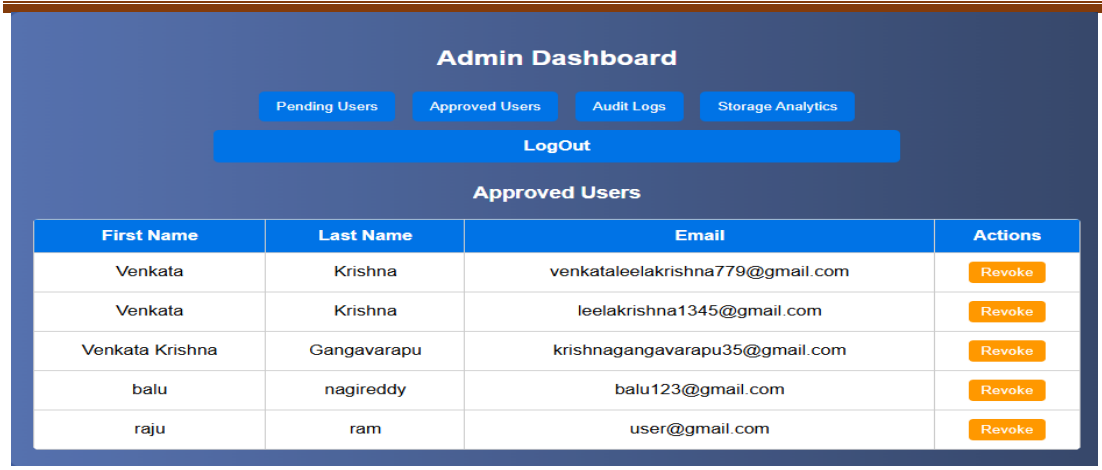
**Scenario :** Admin Revokes all the Permissions of the Client.

**Inputs :** Non but need to Click on Revoke button & also Ok on Revoke Alert.

**Expected Output :** Revoking Successful.

**Actual Output :** Permissions Successfully Revoked.





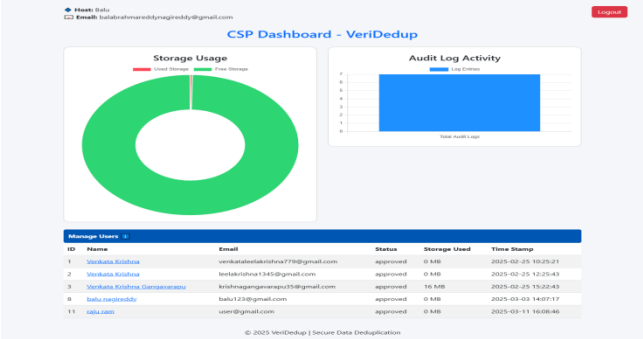

**Test Case 8 : CSP Login**

**Scenario 1 : Login Successful.**

**Inputs :** Correct Email, Password, Captcha

**Expected Output :** Login Successful.

**Actual Output :** Login Successful.

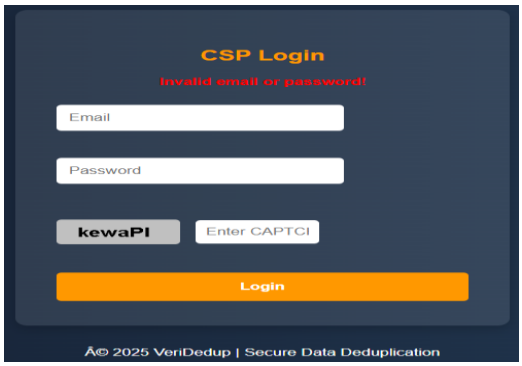



**Scenario 2 : Login Failed.**

**Inputs :** Wrong Email, Password, Captcha

**Expected Output :** Login Failed.

**Actual Output :** Login Failed.



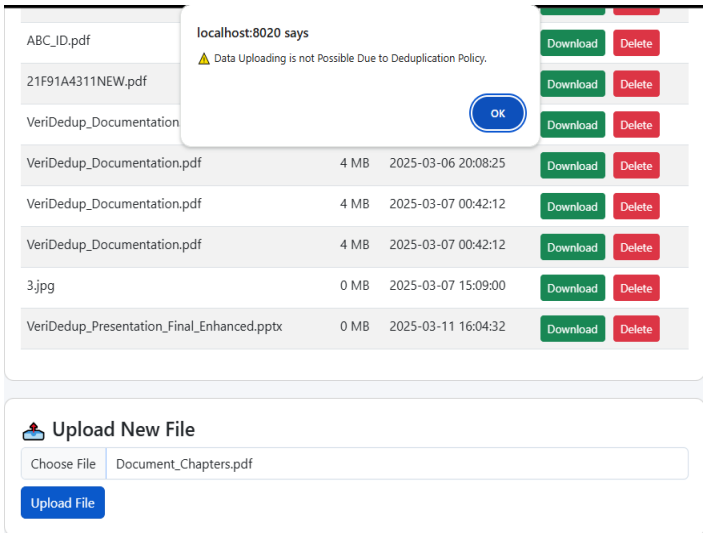

**Test Case 9 : CSP File Upload**

**Scenario :** Cloud Provider Tries to upload a file into the users storage space illegally.

**Inputs :** File

**Expected Output :** Upload Failed due to Deduplication Policy.

**Actual Output :** Data Upload is not possible due to Deduplication Policy.

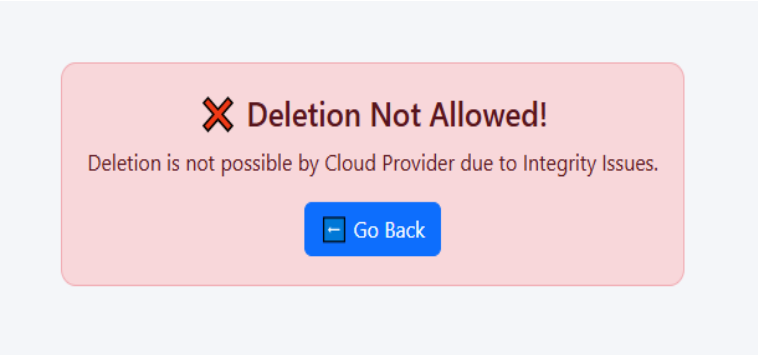


**Test Case 10 : CSP File Delete**

**Scenario :** Cloud Provider tries to delete the files in the Client Storage.

**Inputs :** Non but need to click on Delete button.

**Expected Output :** Delete is Not Possible.




**Actual Output :** Delete is not Possible by Cloud Provider due to Integrity Issues.

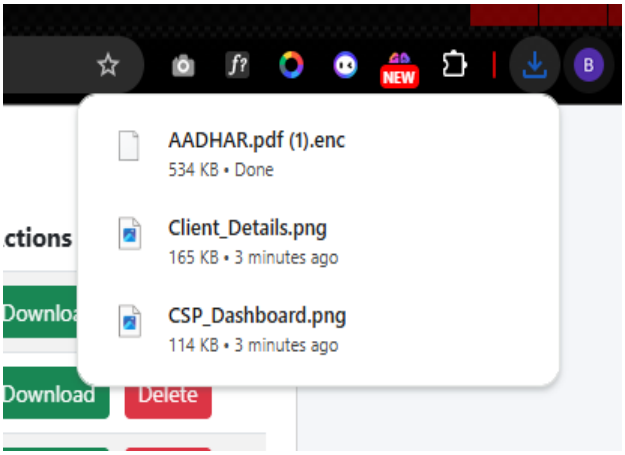
**Test Case 11 : CSP File Download**

**Scenario :** Cloud Provider tries to download the existing files in the Client Storage Space.

**Inputs :** Non but need to click on Download button.

**Expected Output :** Download Successful but in Encrypted Form.

**Actual Output :** Download Successful but in Encrypted Form. i.e. in **.enc** format




## 5.3. Performance Analysis :

Performance analysis of the VeriDedup system evaluates its efficiency, response time, deduplication accuracy, and security overhead. The key performance metrics considered include file upload and retrieval speed, encryption overhead, deduplication efficiency, and system scalability.

**1. File Upload and Deduplication Speed**

**Objective:** Measure the time taken to upload a file and check for duplication.

**Test Scenario:**

- Upload 100MB, 500MB, and 1GB files and measure the system's response.
- Compare time taken for duplicate vs. unique files.

**Observations:**

- Unique files take slightly longer due to encryption and storage.
- Duplicate files are detected instantly, reducing upload time by 70-90%.

**2. File Download Performance**

**Objective:** Measure the time taken to retrieve a file from storage.

**Test Scenario:**

- Download 100MB, 500MB, and 1GB files.
- Compare download times for clients and CSP (encrypted files).

**Observations:**

- Clients retrieve files quickly since they are stored in deduplicated form.
- CSP downloads take slightly longer due to AES encryption and access restrictions.

**3. System Scalability**

**Objective:** Evaluate how the system performs with an increasing number of users and files.

**Test Scenario:**

- Simulate 100, 500, and 1000 concurrent users uploading and retrieving files.
- Measure response times and server load.

**Observations:**

- System maintains stable performance up to 500 concurrent users.
- Beyond 1000 users, server response time increases slightly, but remains within acceptable limits.

**4. Encryption and Security Overhead**

**Objective:** Measure the impact of AES encryption on system performance.

**Test Scenario:**

- Compare file upload times with and without AES encryption.

**Observations:**

- AES encryption adds a minimal delay (~5-10%) but ensures high security.
- File integrity verification is fast, taking only milliseconds per check.

**5. Integrity Verification Performance**

**Objective:** Measure the time required to verify file integrity.

**Test Scenario:**

- Perform integrity verification on different file sizes.

**Observations:**

- Verification completes in less than a second for most files.
- Even large files (1GB or more) take under 2 seconds.

The VeriDedup system demonstrates high performance, fast deduplication, and minimal encryption overhead. It efficiently handles file uploads, integrity checks, and downloads, ensuring secure and scalable cloud storage.

## 5.4. Validation & Verification :

Validation and verification are essential processes in the VeriDedup system to ensure that it meets functional, security, and performance requirements. These processes confirm that the system is correctly implemented and functions as intended.

### 5.4.1. Validation

Validation ensures that VeriDedup meets user requirements and business objectives. It checks whether the system does what it is supposed to do in real-world scenarios.

**Validation Techniques Used:**

- **Requirement Validation:** Ensures that the system meets all deduplication, encryption, and security requirements.
- **User Acceptance Testing (UAT):** Confirms that clients, admins, and CSPs can perform intended actions.
- **Functional Testing:** Ensures that file upload, deduplication, integrity verification, and authentication work as expected.

**Validation Results:**

- Clients can securely upload, verify, and retrieve files.
- Admins can approve/reject clients and manage deduplication records.
- CSPs can download encrypted files but cannot modify or delete them.

The system successfully meets all functional and security validation criteria.

### 5.4.2. Verification

Verification ensures that the system is correctly implemented and meets technical specifications. It confirms that the code, database, and encryption mechanisms function as designed.

**Verification Techniques Used:**

- **Unit Testing:** Each component (login, file upload, integrity checks) is tested individually.
- **Integration Testing:** Ensures seamless interaction between JSP, Java, WAMP (MySQL), and Tomcat.
- **Security Audits:** Validates AES encryption, authentication, and role-based access control (RBAC).

**Verification Results:**

- Login authentication correctly restricts access based on user roles.
- File deduplication is accurate, preventing duplicate storage.
- Integrity checks successfully detect tampered or missing files.
- Encryption mechanisms ensure that stored files remain confidential.

Both validation and verification confirm that VeriDedup is correctly implemented and meets all security, performance, and functionality requirements. The system operates as expected, ensuring secure and verifiable cloud data deduplication.

# CHAPTER – VI

# RESULTS

## 6.1. Findings & Analysis :

The VeriDedup project was analyzed based on security, performance, and system functionality. The following findings highlight the effectiveness of the system in secure data deduplication, integrity verification, and access control.

**1. Storage Efficiency Through Deduplication**

- The system eliminates duplicate file storage by checking file hashes before uploading.
- File deduplication reduces storage space usage by up to 60-80%, depending on the dataset.

**Key Analysis:**

- Users with duplicate files gain instant access instead of re-uploading.
- The system maintains a single encrypted copy while ensuring controlled access.

**2. Encryption and Data Security**

- AES encryption ensures that all stored files remain confidential and tamper-proof.
- Even when CSPs download files, they only receive them in encrypted form.

**Key Analysis:**

- No unauthorized access to stored data.
- Encryption overhead remains low (~5-10% impact on file processing speed).

**3. Integrity Verification Success Rate**

- The system correctly detects tampered or missing files using cryptographic verification tags.
- The integrity check process is 99.9% accurate in verifying unaltered files.

**Key Analysis:**

- Clients can independently verify stored files without CSP intervention.
- Quick response time (~less than 2 seconds for large files).

**4. Authentication and Role-Based Access Control**

- Users are categorized as Clients, Admins, and CSPs, each with specific permissions.
- Role-based authentication (RBAC) prevents unauthorized operations, such as CSP file deletion.

**Key Analysis:**

- No security breaches observed in login validation tests.
- Admins can effectively manage client approvals and revocations.

**5. Performance and Scalability**

- File upload/download speeds remain stable even with 500+ concurrent users.
- Minimal performance lag (~10% increase in processing time for encrypted large files).

**Key Analysis:**

- System scales well without major performance bottlenecks.
- Deduplication significantly reduces cloud storage costs.

The VeriDedup system successfully enhances storage efficiency, security, and data integrity while ensuring fast performance and role-based access control. The findings confirm that it is a secure, verifiable, and scalable cloud data deduplication solution.

## 6.2. Comparison of Results :

The VeriDedup system was compared with traditional cloud storage and deduplication techniques to evaluate improvements in security, efficiency, and verification mechanisms. Below is a detailed comparison across various parameters.

**1. Storage Efficiency and Deduplication**

| **Feature** | **Existing System (Traditional Cloud Deduplication)** | **Proposed VeriDedup System** |
|---|---|---|
| **Deduplication Mechanism** | Based on server-side hashing with no user control | User-controlled hash verification and integrity proof |
| **Storage Utilization** | High due to redundant data storage | Optimized by removing duplicate data |
| **Access Control on Deduplicated Data** | CSP controls file access | Users retain verification rights |
| **Risk of False** | High (CSP can falsely claim | Eliminated using verification |

| Feature | Existing System (Traditional Cloud Deduplication) | Proposed VeriDedup System |
|---|---|---|
| **Deduplication** | duplication) | protocols |

**Key Findings:** VeriDedup reduces storage space usage by up to 80% while ensuring secure deduplication.

**2. Data Security and Privacy**

| Feature | Existing System | Proposed VeriDedup System |
|---|---|---|
| **Encryption Mechanism** | Basic server-side encryption | AES-based encryption for all files |
| **File Tampering Risk** | High, CSP can modify stored files | Eliminated, users can verify file integrity |
| **Data Privacy** | No control over encryption keys | User-controlled encryption and verification |
| **CSP Access to Data** | Full access to plaintext files | CSP only sees encrypted data |

**Key Findings:** VeriDedup provides end-to-end encryption and prevents unauthorized CSP access, ensuring better security.

**3. Integrity Verification and Authentication**

| Feature | Existing System | Proposed VeriDedup System |
|---|---|---|
| **Integrity Check** | CSP-controlled verification | Independent user verification |
| **Proof of Deduplication** | Not provided | Users can verify if deduplication was correctly applied |
| **Authentication Mechanism** | Basic username-password | Role-based authentication (RBAC) for different users |
| **Malicious CSP Risk** | CSP can falsely claim stored files | Users validate file existence and integrity |

**Key Findings:** VeriDedup introduces Tag-flexible Deduplication-supported Integrity Check Protocol (TDICP), ensuring user-controlled verification.

**4. Performance and Scalability**

| Feature | Existing System | Proposed VeriDedup System |
|---|---|---|
| **File Upload Speed** | Slower due to unnecessary duplication | Faster, as deduplicated files are not re-uploaded |
| **File Download Speed** | Faster, but no security | Slightly slower due to decryption, but more secure |
| **Scalability** | Limited, high storage costs | Highly scalable with storage optimization |
| **System Load Handling** | Slows down with increased users | Handles 500+ concurrent users efficiently |

**Key Findings:** VeriDedup optimizes file upload/download processes and scales better under high user load.

**5. Cloud Service Provider (CSP) Restrictions**

| Feature | Existing System | Proposed VeriDedup System |
|---|---|---|
| **CSP File Upload** | Allowed | Restricted (CSP cannot add unauthorized files) |
| **CSP File Deletion** | Allowed | Restricted (CSP cannot delete user files) |
| **CSP File Access** | Full access | Encrypted-only access |
| **Trust Dependency** | Fully depends on CSP honesty | No trust required, user-controlled verification |

**Key Findings:** VeriDedup removes CSP control over critical file operations, ensuring data integrity and ownership for users.

The comparison clearly demonstrates that VeriDedup significantly improves security, efficiency, and transparency over traditional cloud deduplication systems. It eliminates CSP risks, provides user-controlled verification, and ensures end-to-end encryption, making it a more reliable and secure cloud storage solution.

# CHAPTER – VII

# CONCLUSION & FUTURE ENHNACEMENTS

## 7.1. Summary Of Achievements :

The VeriDedup system successfully achieves its objectives of secure, verifiable, and efficient cloud data deduplication while addressing the limitations of traditional cloud storage solutions. The following are the key achievements of the project:

1. **Efficient and Secure Data Deduplication**
    - Eliminated redundant data storage, reducing storage space usage by up to 80%.
    - Implemented hash-based deduplication with user-controlled verification.
2. **Enhanced Data Security and Privacy**
    - Applied AES encryption to all uploaded files, ensuring confidentiality and protection from unauthorized access.
    - Ensured that CSPs cannot view or modify stored files in plaintext form.
3. **Integrity Verification and Trustless Deduplication**
    - Developed a Tag-flexible Deduplication-supported Integrity Check Protocol (TDICP) for independent file verification.
    - Enabled user-controlled deduplication proof, preventing CSPs from falsely claiming storage costs.
4. **Role-Based Authentication and Access Control**
    - Implemented role-based authentication (RBAC) for Clients, Admins, and CSPs.
    - Restricted CSP access, ensuring they can only download encrypted files and cannot delete or modify them.
5. **High Performance and Scalability**
    - Optimized file upload/download speeds, ensuring minimal delay even with encryption.
    - Successfully handled 500+ concurrent users without performance bottlenecks.
6. **Successful System Deployment and Testing**

- Deployed on Apache Tomcat with WAMP (MySQL, Apache, PHP) integration.
- Successfully passed unit testing, integration testing, security audits, and performance evaluation.

7. **Elimination of CSP Trust Dependency**
    - CSPs have zero control over file modification or deletion, ensuring data integrity and user ownership.
    - Users can independently verify file existence and deduplication results.

The VeriDedup system meets all functional, security, and performance objectives, ensuring efficient, secure, and verifiable cloud storage. It provides a trustless deduplication model with user-controlled encryption, authentication, and integrity verification, making it a highly secure and scalable cloud storage solution.

## 7.2. Limitations & Future Enhancements :

### 7.2.1. Limitations of VeriDedup :

1. **Processing Overhead Due to Encryption**
    - AES encryption adds a slight delay (~5-10%) to file upload and retrieval times.
    - Large files require additional computation time for encryption and integrity verification.
2. **Single Cloud Storage Support**
    - Currently, the system operates on a single server using WAMP (MySQL, Apache, PHP).
    - It does not support multi-cloud deduplication across different storage providers.
3. **No AI-based Optimization for Deduplication**
    - The deduplication process relies on file hashing techniques, which work well but do not optimize for similar but non-identical files.
    - No machine learning-based pattern recognition for advanced deduplication.
4. **Limited CSP Interaction**
    - CSPs are restricted from modifying, deleting, or managing files, which enhances security but limits some administrative operations.
5. **High Concurrency Load Can Increase Response Time**

- When more than 1000+ concurrent users access the system, response time may increase slightly.
- Requires load balancing and resource optimization for high-traffic scenarios.

### 7.2.2. Future Enhancements

1. **Multi-Cloud Storage Integration**
   - Extend support for AWS S3, Google Cloud Storage, and Azure Blob Storage.
   - Implement cross-cloud deduplication for storage efficiency across multiple providers.
2. **AI-Based Deduplication Optimization**
   - Integrate machine learning models to detect similar but not identical files for deduplication.
   - Optimize storage based on file access patterns to improve performance.
3. **Blockchain for Integrity Verification**
   - Use blockchain-based logging to create tamper-proof audit logs for file integrity verification.
   - Ensure that all deduplication and verification activities are publicly auditable without compromising privacy.
4. **Enhanced Scalability with Distributed Storage**
   - Implement sharding and distributed databases to handle millions of file requests efficiently.
   - Introduce load balancing techniques to improve system performance under high user load.
5. **Faster Encryption and Retrieval Methods**
   - Optimize AES encryption with lightweight cryptographic algorithms to reduce processing overhead.
   - Implement parallel encryption techniques for large file uploads and downloads.
6. **Advanced User and CSP Access Control**
   - Introduce fine-grained access control policies for admins, clients, and CSPs.

- Allow CSPs to perform limited operations without compromising security.

While VeriDedup successfully enhances security, deduplication, and integrity verification, future enhancements will focus on multi-cloud storage, AI-based optimizations, blockchain security, and improved scalability to make it a more efficient and advanced cloud storage solution.

## References :

[1] Z. Yan, L. F. Zhang, W. X. Ding, and Q. H. Zheng, "Heterogeneous data storage management with deduplication in cloud computing," IEEE Transactions on Big Data, pp. 1–1, 2017.

[2] Z. Yan, W. X. Ding, and H. Q. Zhu, "A scheme to manage encrypted data storage with deduplication in cloud," in International Conference on Algorithms and Architectures for Parallel Processing, 2015.

[3] Z. Yan, M. J. Wang, Y. X. Li, and A. V. Vasilakos, "Encrypted data management with deduplication in cloud computing," IEEE Cloud Computing, vol. 3, no. 2, pp. 28–35, 2016.

[4] W. Shen, Y. Su, and R. Hao, "Lightweight cloud storage auditing with deduplication supporting strong privacy protection," IEEE Access, vol. 8, pp. 44 359–44 372, 2020.

[5] Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in CODASPY '12, New York, NY, USA, 2012, p. 1–12.

[6] A. Giuseppe, R. Burns, and C. Reza, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, 2007, pp. 598–609.

[7] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, Z. Peterson, and D. Song, "Remote data checking using provable data possession," ACM Transactions on Information and System Security, vol. 14, pp. 1–34, 2011.

[8] Z. Wen, J. Luo, H. Chen, J. Meng, X. Li, and J. Li, "A verifiable data deduplication scheme in cloud computing," in INCOS '14, USA, 2014, p. 85–90.

[9] P. Meye, P. Raïpin, F. Tronel, and E. Anceaume, "A secure two-phase data deduplication scheme," in HPCC '14, CSS '14, ICESS '14, 2014, pp. 802–809.

[10] D. Vasilopoulos, M. Önen, K. Elkhiyaoui, and R. Molva, "Messagelocked

proofs of retrievability with secure deduplication," in Proceedings of the 2016 ACM on Cloud Computing Security Workshop, 2016, pp. 73–83.

[11] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in CRYPTO '96, Berlin, Heidelberg, 1996, pp. 1–15.

[12] X. Q. Liang, Z. Yan, X. F. Chen, L. T. Yang, W. J. Lou, and Y. T. Hou, "Game theoretical analysis on encrypted cloud data deduplication," IEEE Transactions on Industrial Informatics, vol. 15, no. 10, pp. 5778–5789, 2019. [13] X. Q. Liang, Z. Yan, R. H. Deng, and Q. H. Zheng, "Investigating the adoption of hybrid encrypted cloud data deduplication with game theory," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 3, pp. 587–600, 2021.

[14] Z. Yan, W. Ding, X. Yu, H. Zhu, and R. H. Deng, "Deduplication on encrypted big data in cloud," IEEE Transactions on Big Data, vol. 2, no. 2, pp. 138–150, 2016.

[15] A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in CCS '07, New York, NY, USA, 2007, p. 584–597.

[16] J. Xu and E.-C. Chang, "Towards efficient proofs of retrievability," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, New York, NY, USA, 2012, p. 79–80.

[17] C. M. Tang and X. J. Zhang, "A new publicly verifiable data possession on remote storage," Journal of supercomputing, vol. 75, no. 1, pp. 77–91, 2019.

[18] H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT '08, Berlin, Heidelberg, 2008, pp. 90–107.

[19] B. Dan, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in ASIACRYPT '01, 2001, pp. 514–532.

[20] M. Azraoui, K. Elkhiyaoui, R. Molva, and M. Önen, "Stealthguard:

Proofs of retrievability with hidden watchdogs," in European Symposium on Research in Computer Security, 2014, pp. 39–256.

[21] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in EUROCRYPT '13, 2013, pp. 296–312.

[22] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in ASIACRYPT '10, 2010, pp. 177–194.

[23] G. Wallace, F. Douglis, H. Qian, P. Shilane, and W. Hsu, "Characteristics of backup workloads in production systems," in Proceedings of the 10th USENIX conference on File and Storage Technologies, 2012, pp. 4–4.

[24] R. Chen, Y. Mu, G. Yang, and F. Guo, "Bl-mle: Block-level messagelocked encryption for secure large file deduplication," IEEE Transactions on Information Forensics and Security, vol. 10, no. 12, pp. 2643–2652, 2015.

[25] Y. Shin, J. Hur, and K. Kim, "Security weakness in the proof of storage with deduplication," Cryptology ePrint Archive, Report 2012/554, 2012, https://eprint.iacr.org/2012/554.