A Midterm Progress Report

on

AccelDedup

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

Computer Science and Engineering

SUBMITTED BY

ROBAN SINGH(2104169)

RUPALLI DEVI (2104172)

SIMRAN TIWARI (2104194)

UNDER THE GUIDANCE OF

Er. Shailja
(September-2024)

Department of Computer Science and Engineering
GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA, (141006)

LIST OF FIGURES

INDEX

1. INTRODUCTION

The Accel Dedup system is an advanced, cloud-based data optimization solution focused on data deduplication, which is crucial for organizations grappling with large-scale data storage. As companies generate and store enormous amounts of information, the problem of data redundancy becomes a significant issue. This redundancy arises when multiple copies of the same data are stored in different locations, unnecessarily consuming precious storage space, increasing operational costs, and potentially degrading overall system performance.

Accel Dedup tackles this problem through intelligent data deduplication, which identifies and eliminates redundant copies of data. This is achieved by examining data at a granular level and identifying duplicate blocks, files, or objects stored across various cloud platforms. It can analyze stored content in databases, virtual machines, backups, or any digital environment where duplicate data can accumulate. By ensuring that only unique copies of data are retained, the system significantly optimizes the storage infrastructure.

Key Features:

1. **Comprehensive Deduplication**: Accel Dedup examines data across multiple layers, including block-level, file-level, and application-level storage systems, to ensure that all redundant copies are removed.
2. **Cross-Platform Integration:** It supports various cloud storage providers, enabling seamless deduplication across different environments such as AWS, Google Cloud, and Azure.
3. **Real-Time Deduplication:** The system continuously monitors and deduplicates data as it's ingested, ensuring storage efficiency from the moment new data is added.

4. **Scalability:** As organizations grow and their storage demands increase, Accel Dedup scales effortlessly to handle large volumes of data without compromising performance.

5. **Cost Efficiency:** By eliminating duplicate data, Accel Dedup helps organizations reduce storage costs, making it a valuable tool for enterprises with significant cloud storage requirements.

6. **Enhanced System Performance:** With reduced data volumes, systems experience faster backup and recovery times, more efficient data retrieval, and improved application performance.

How the Cloud Deduplication Process Works:

The app uses algorithms to analyze the data stored on cloud platforms. It identifies redundant files and data chunks by comparing data signatures, such as hash values or fingerprints. Once duplicates are identified, the app consolidates them, ensuring that only one unique copy of each data element is stored while referencing it appropriately wherever needed. This approach ensures that no essential data is lost, but unnecessary storage overhead is eliminated.

Objectives of the AccelDedup:

1. **To create an interface for storing data on the cloud.**

The Accel Dedup project achieves its first objective of creating an interface for storing data on the cloud through a user-friendly web application built with React. This interface allows users to easily upload files, which are then processed to identify and eliminate duplicate data using sophisticated algorithms. The application interacts with a cloud-based backend, utilizing APIs to securely store unique data and maintain metadata for tracking purposes. This streamlined process ensures that users can manage their data efficiently without worrying about redundancy.

Furthermore, the application integrates with various cloud storage services, providing users with the flexibility to choose their preferred storage solution. This adaptability is crucial for organizations dealing with diverse data environments. By ensuring that only unique data is stored and providing users with a clear view of their storage status, Accel Dedup not only simplifies data management but also optimizes storage costs and enhances overall system performance

2. **To perform de-duplication on the cloud platform.**

Accel Dedup employs advanced algorithms to perform de-duplication on cloud platforms by analyzing data before it is stored. When users upload files, the application scans for identical copies using hashing techniques that generate unique identifiers for each file. By comparing these hashes, the system can quickly identify duplicates and determine which data can be retained or discarded. This process significantly reduces the volume of data stored, leading to lower storage costs and improved system efficiency.

Additionally, the de-duplication process is seamlessly integrated into the user interface, allowing users to view their storage consumption in real-time. Users receive notifications about potential duplicates and can manage their data through intuitive options, such as reviewing duplicates and selecting which files to keep. This not only enhances user experience but also ensures optimal utilization of cloud resources, making data management more efficient and cost-effective.

3. **To compare and analyze existing de-duplication techniques with the proposed solution.**

Accel Dedup aims to enhance data storage efficiency through its proprietary deduplication techniques while comparing them with existing methods such as filelevel, block-level, and byte-level de-duplication. Traditional file-level de-duplication identifies duplicate files by comparing file names and sizes, which can be limiting, as it fails to recognize identical files stored with different names. In contrast, block-level de-duplication breaks files into smaller

chunks, allowing for more granular comparisons, but can lead to increased processing overhead. Accel Dedup leverages a hybrid approach that combines the strengths of both techniques, utilizing advanced hashing algorithms to analyze data more effectively while minimizing resource usage

.

Moreover, the proposed solution not only focuses on improving de-duplication efficiency but also emphasizes data integrity and security. By analyzing existing techniques, Accel Dedup incorporates best practices while addressing common drawbacks, such as high processing time and potential data loss during the deduplication process. The solution enables users to maintain complete control over their data, providing features such as selective retention of files, real-time monitoring of storage usage, and comprehensive reporting on data redundancy. This holistic approach ensures that Accel Dedup stands out in the crowded landscape of cloud storage solutions, offering superior performance and user satisfaction.

2. SYSTEM REQUIREMENTS

2.1 Software Requirements

2.1.1. Operating System

- Development: Windows
- Server: Windows Server

2.1.2. Backend

- Language: Node.js (v14+)
- Framework: Express.js (v4+)
- Cloud SDKs: AWS,
- File Management: Multer (for uploads), Bcrypt (for encryption)
- API Testing: Postman

2.1.3. Frontend

- Language: JavaScript/TypeScript
- Framework: React.js (v18+),
- Routing: React Routern ReactDom
- Middleware: Cros and Json.

HTTP Client: Axios

- UI: Material-UI or Bootstrap
- Testing: Jest, React Testing Library

2.1.4. Cloud Infrastructure

- Cloud Providers: AWS (S3), Google Cloud, Azure
- Hosting: EC2 Instances, Lambda/Cloud Functions (optional)

2.1.6. Tools

Version Control: Git (GitHub/GitLab)

- IDE: Visual Studio Code
- Package Manager: npm/Yarn

2.2 Hardware Requirement

2.2.1.Development Environment

- Processor: Intel Core i5 or AMD equivalent (minimum), i7 or higher recommended
- RAM: 8 GB (minimum), 16 GB or more recommended for smooth multitasking
- Storage: 500 GB SSD (minimum), 1 TB or higher recommended for fast read/write operations
- Graphics: Integrated GPU (minimum), dedicated GPU for better rendering performance

2.2.2. Backend Server (for Production)

- Processor: Intel Xeon or AMD EPYC (multi-core) recommended for handling concurrent

users

- RAM: 16 GB (minimum), 32 GB or more for high traffic applications Storage:
  Primary Storage: 500 GB SSD (minimum), 1 TB SSD recommended for faster
- read/write operations
  Additional Storage: Optional scalable cloud storage (e.g., AWS S3, Google
- Cloud Storage)

Network: High-speed internet connection (1 Gbps recommended) for seamless cloud communication

2.2.3. Frontend Client Requirements

- Processor: Intel Core i3 or higher (for client machines)
- RAM: 4 GB (minimum), 8 GB or more for intensive tasks
- Storage: 128 GB SSD (minimum), 256 GB or more for performance
- Browser: Latest versions of Chrome, Firefox, or Edge

2.2.4. Cloud Infrastructure

- Compute Power: AWS EC2 t2.large instance (or equivalent in GCP/Azure) with scalable options
- Storage: Cloud-based storage (AWS S3, Google Cloud Storage, Azure Blob) with dynamic scaling

3. SOFTWARE DESIGN

The software design of Accel Dedup follows a modular, multi-tier architecture to ensure scalability, maintainability, and performance. The application is designed with distinct layers for the frontend,

backend, database, and cloud storage interaction, allowing each component to perform specific tasks efficiently. The design integrates modern web technologies, cloud platforms, and a robust API structure to support data deduplication operations.

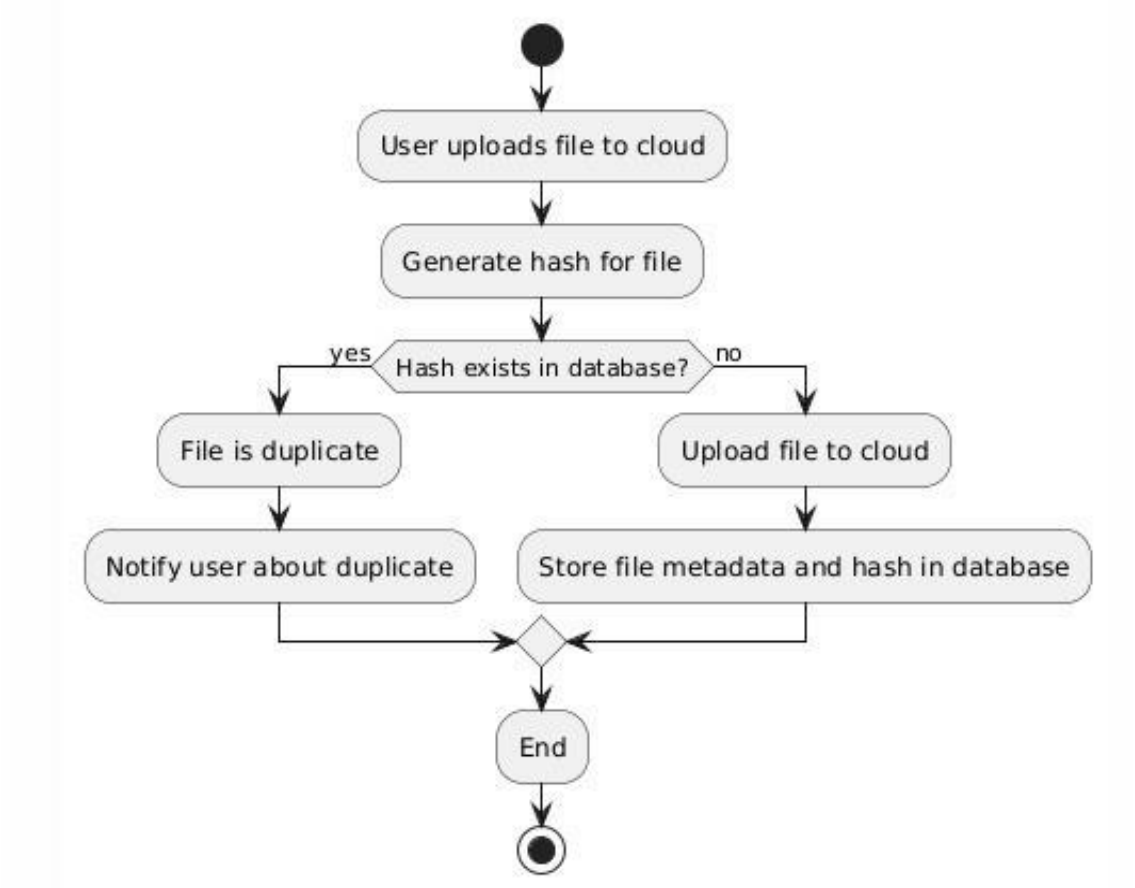

Figure 3.1 Flowchart

3.1. Frontend Design (Client-Side)

The client-side is a responsive web application built using React.js, a popular JavaScript library for creating interactive user interfaces. The frontend is responsible for user interactions, data input, and visualization of deduplication results.

- **Components:**

- Login & Authentication: Handles secure user login via JWT (JSON Web Tokens) and validates credentials. o Dashboard: Displays data deduplication statistics, progress, and visual graphs to monitor storage usage and savings.
- File Upload Interface: Enables users to upload files for deduplication.
- Deduplication Status: Provides real-time updates on deduplication processes.

- **Routing:** Managed using React Router, enabling smooth navigation between pages (e.g., login, dashboard, upload files).
- State Management: Utilizes React's useState and useContext for managing application state, authentication tokens, and session data.

3.2. Backend Design (Server-Side)

The backend is built using Node.js with Express.js, providing the logic to handle requests, process data, and interact with external services. The backend is responsible for data deduplication, cloud interaction, and storage management.

- **API Layer:**
  - RESTful API: Exposes endpoints to handle file uploads, trigger deduplication processes, and manage user data.
  - Endpoints:
    - /upload: Handles file uploads from the frontend.
    - /dedup: Initiates the deduplication process.
    - /status: Returns the deduplication status and results.

- **Authentication:** Implements JWT for secure authentication, ensuring that only authorized users can access the deduplication service.
- **Deduplication Logic:**

  o The deduplication logic leverages file hashing algorithms (e.g., SHA-256) to identify duplicate files. o Duplicate files are mapped and stored efficiently to minimize redundant data storage.

Cloud Storage Integration: o AWS SDK or equivalent libraries are used to communicate with cloud storage (e.g., AWS S3, Google Cloud Storage). o File metadata and deduplication results are stored in a cloud database.

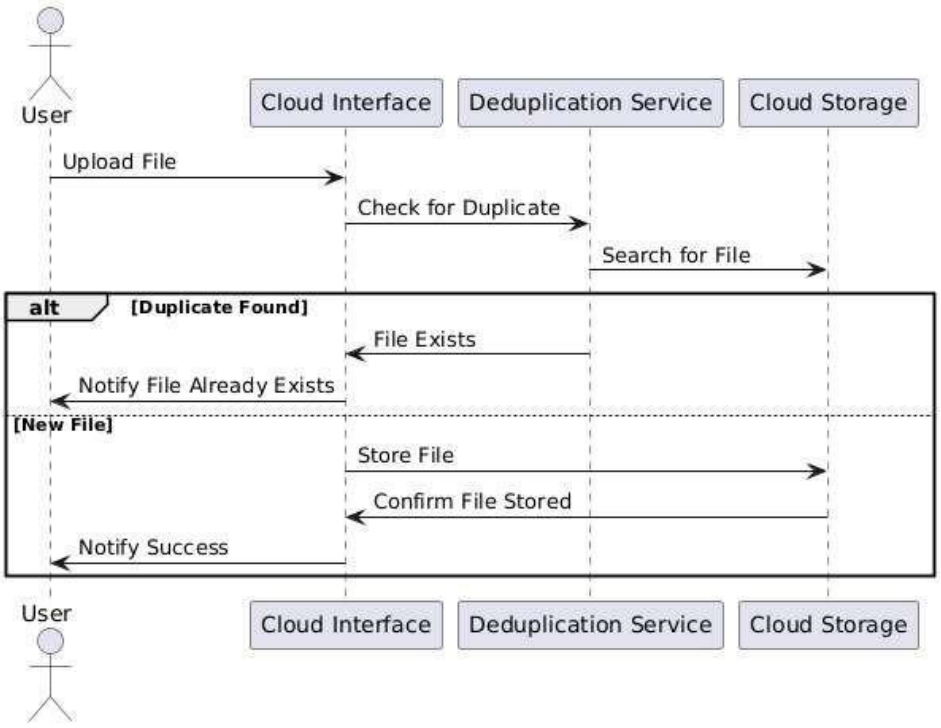

Figure 3.2Sequence Diagram

3.3. Database Design

The database is designed to store user data, file metadata, deduplication results, and logs. MongoDB is used for its flexibility and ability to handle large volumes of unstructured data.

- **Collections:**
  - Users: Stores user details, credentials, and authentication tokens. of Files: Stores file metadata (hashes, size, storage location) and deduplication status. o Deduplication Logs: Maintains a history of deduplication activities and performance metrics.
- **Indexes:** Optimized indexes on file hashes to quickly identify duplicates during the deduplication process.

3.4. Deduplication Engine

At the core of Accel Dedup is the deduplication engine, responsible for analyzing uploaded data and identifying duplicates.

- **Hashing Algorithm:**
  - Each file is processed through a hashing algorithm (e.g., SHA-256), generating a unique hash for each file. If a file's hash already exists in the system, it is flagged as a duplicate.
- **Storage Optimization:**
  - If duplicates are found, only a reference to the original file is stored, reducing overall storage consumption.
  - Metadata related to deduplication (e.g., file path, storage location) is stored in the database.

3.5. Cloud Infrastructure Integration

The application integrates with cloud storage services to provide scalable and secure file storage solutions. It leverages services like AWS S3 or Google Cloud Storage for storing files and performing data deduplication operations.

- **File Uploads:** Files are uploaded to cloud storage, where deduplication is applied.
- **Storage Management:** Once deduplication is completed, the unique data is retained, while redundant copies are discarded or replaced with reference pointers.
- **Scaling:** The system can scale horizontally by adding more instances to handle larger data volumes or increased user traffic. 6. Error Handling and Logging
- **Error Handling:** The application implements robust error handling, ensuring that any issues during the deduplication process, file uploads, or cloud communication are logged and handled gracefully.

  Logging: Logs are maintained for both server-side and client-side errors, providing insight into system performance, user activities, and deduplication efficiency.

3.6. Security

- **Data Encryption:** All file transfers between the client, server, and cloud storage are encrypted using HTTPS and secured with SSL/TLS.
- **Access Control**: The system uses role-based access control (RBAC) to manage user permissions, ensuring only authorized users can upload or manage data.
- **Cloud Security**: Cloud storage follows best practices for data encryption and secure access, such as using IAM roles and access control lists (ACLs).

Architectural Diagram

- Frontend (React) → Backend (Node.js/Express) → Database (MongoDB) and Cloud Storage (AWS S3/Google Cloud Storage)

This modular software design ensures that Accel Dedup is efficient, scalable, and secure, providing a seamless solution for eliminating redundant data in cloud storage envir

4.TESTING MODULE

Testing is a crucial aspect of ensuring the reliability, performance, and security of the Accel Dedup application. Various testing techniques are employed to validate different components of the system, including the frontend, backend, deduplication engine, and cloud integration. **4.1. Unit Testing**

Unit testing involves testing individual components or functions to ensure they work as expected. It is applied across both frontend and backend components.

- Frontend (React) Unit Testing: Using Jest and React Testing Library to test individual UI components like forms, buttons, and navigation elements.
- Backend (Node.js) Unit Testing: Using Mocha or Jest to test API endpoints, deduplication logic, and cloud integration modules.

4.2. Integration Testing

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

API Integration: Testing REST API endpoints to ensure correct responses for file uploads, deduplication requests, and result fetching.

4.3. End-to-End (E2E) Testing

E2E testing simulates real-world scenarios and user workflows, testing the application from start to finish. This includes the full flow from file upload to deduplication and result display.

- Tools: E2E testing is performed using Cypress or Selenium, simulating user interactions like logging in, uploading files, viewing deduplication results, and managing data.

4.4. Performance Testing

Performance testing is conducted to measure how the system behaves under load. It ensures that the deduplication engine can handle large data volumes without performance degradation and that cloud storage interaction is efficient.

- Load Testing: Using Apache JMeter to simulate multiple users uploading files and running deduplication processes simultaneously.
- Stress Testing: Testing the system’s limits by uploading large files or a high volume of files to identify performance bottlenecks.

4.5. Security Testing

Security testing ensures that the application is safe from common vulnerabilities and that sensitive data is protected.

- Authentication Testing: Ensuring that JWT-based authentication is secure and users can only access data they're authorized for.

- Data Encryption: Testing file transfer security to ensure data is encrypted during transmission to and from the cloud.
- Penetration Testing: Simulating attacks to check for vulnerabilities in the API, user authentication, and cloud storage access.

4.6. User Acceptance Testing (UAT)

UAT involves testing the application with real users to ensure that it meets their expectations and requirements. Feedback from this stage helps refine the final product

Sample Test Cases for Accel Dedup

Test Case 1: Verify File Existence in S3 Bucket Test

Case ID: TC001

Test Title: Verify that uploaded files are correctly listed in the S3 bucket.

Preconditions:

The user has successfully uploaded files using the web interface.

The AWS S3 bucket is properly configured, and the user has access to it.

Test Steps:

Navigate to the AWS S3 console and open the specific bucket (e.g., "inpbucket").

List the contents of the bucket.

Expected Results:

Files such as download.jpg, html1.html, ques4.html, ques5.html, and Webfile.docx should be present in the bucket.

Each file should display the correct file name, type, and last modified timestamp (as seen in the image).

Validation Criteria:

The files displayed on the AWS S3 console should match the files uploaded from the client interface.

The "Last modified" timestamps should accurately reflect the upload times.

The file types should be correctly identified (e.g., jpg, html, docx).

Test Case 2: Verify File Accessibility via URL

Test Case ID: TC002

Test Title: Verify that each file can be accessed using its URL.

Preconditions:

Files are successfully uploaded and listed in the S3 bucket.

Test Steps:

Generate the public URL for a file (e.g., html1.html).

Attempt to access the file by opening the URL in a web browser.

Expected Results:

The file should be accessible via its URL.

The content of the file should load properly (e.g., HTML content should render if viewed in a browser).

Validation Criteria:

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

Test Case 3: Duplicate File Upload

Test Case ID: TC003

Test Title: Verify that duplicate file uploads are handled correctly.

Preconditions:

Deduplication logic is implemented in the server to check for file content hash.

Test Steps:

Attempt to upload the same file (e.g., html1.html) that already exists in the S3 bucket.

Expected Results:

The server should identify the duplicate file by its hash.

The system should display a message indicating that the file has already been uploaded.

Validation Criteria:

The system should prevent the re-upload of the same file content and notify the user appropriately.

Test Case 4: **Unsupported File Type Upload**

Test Case ID: TC004

Test Title: Verify that unsupported file types are rejected.

Preconditions:

The server has file type restrictions configured (if applicable).

Test Steps:

Attempt to upload a file with an unsupported type (e.g., .exe or .bat).

Expected Results:

The server should reject the file upload and return an appropriate error message to the user.

Validation Criteria:

The file should not be uploaded, and an error should be displayed (e.g., "Unsupported file type").

Test Case 5: Verify File Deletion from S3 Bucket

6.PERFORMANCE OF THE PROJECT DEVELOPED

6.1 Performance Aspects of the UI Page:

Design Simplicity: The interface has a clean and minimalist design with only essential elements (username, password fields, and a login button), making it easy for users to navigate.

Responsiveness: To evaluate its performance, ensure the UI works across various devices (mobile, tablet, desktop) and different screen sizes.

Load Speed: If the page loads quickly without significant delays, this improves the user experience. Fast response time when entering credentials and submitting them is important.

Accessibility: Assess if the design is accessible to all users, including those with disabilities (e.g., adding proper labels, keyboard navigation support, etc.).

Security Considerations: For login pages, ensure encryption (e.g., SSL) for secure data handling. This is crucial for safeguarding user credentials.

6.2 Performance of Deduplication Process:

File Comparison and Detection: The deduplication service likely compares incoming files (e.g., by file hash or content analysis) to detect if the file already exists in the cloud storage.

Storage Efficiency: By eliminating duplicate files, the cloud storage becomes more efficient, reducing redundant data storage and saving space.

Metadata Management: The interface clearly shows file names, types, and modification times, making it easier to track which files are unique and which have been processed for deduplication.

User Notification: When duplicate files are found, the system may alert the user, preventing them from uploading the same file again.

7.OUTPUT SCREENS

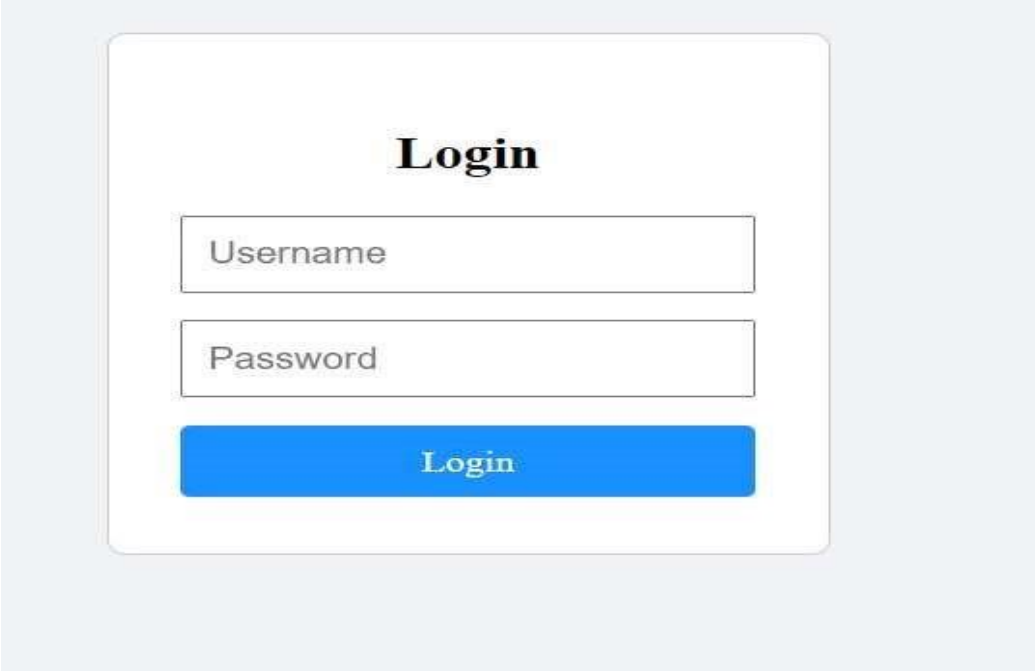

Figure 7.1 Login Page

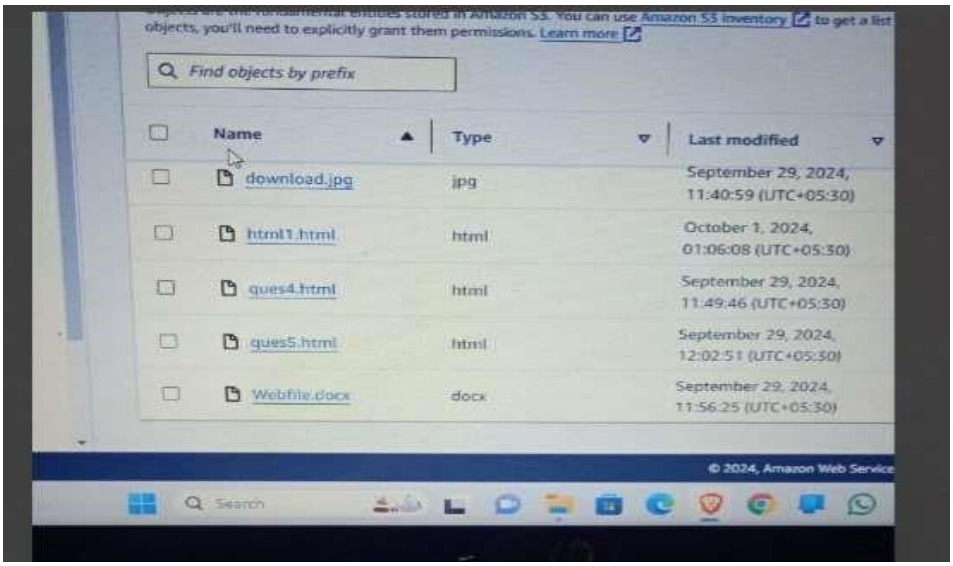

Figure 7.2 Bucket Uploaded Status

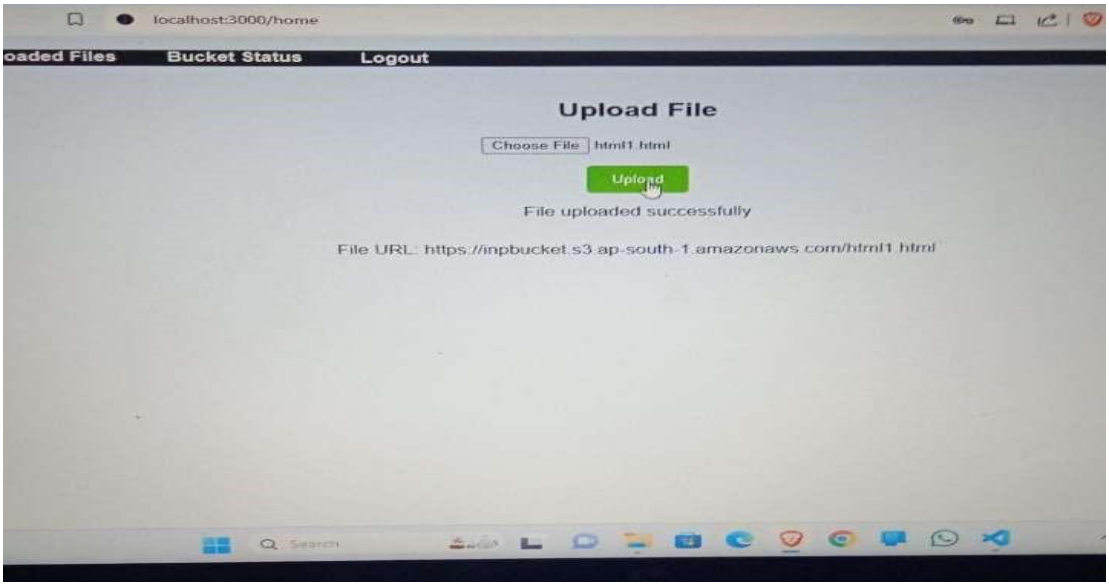

Figure 7.3 Uploading the File

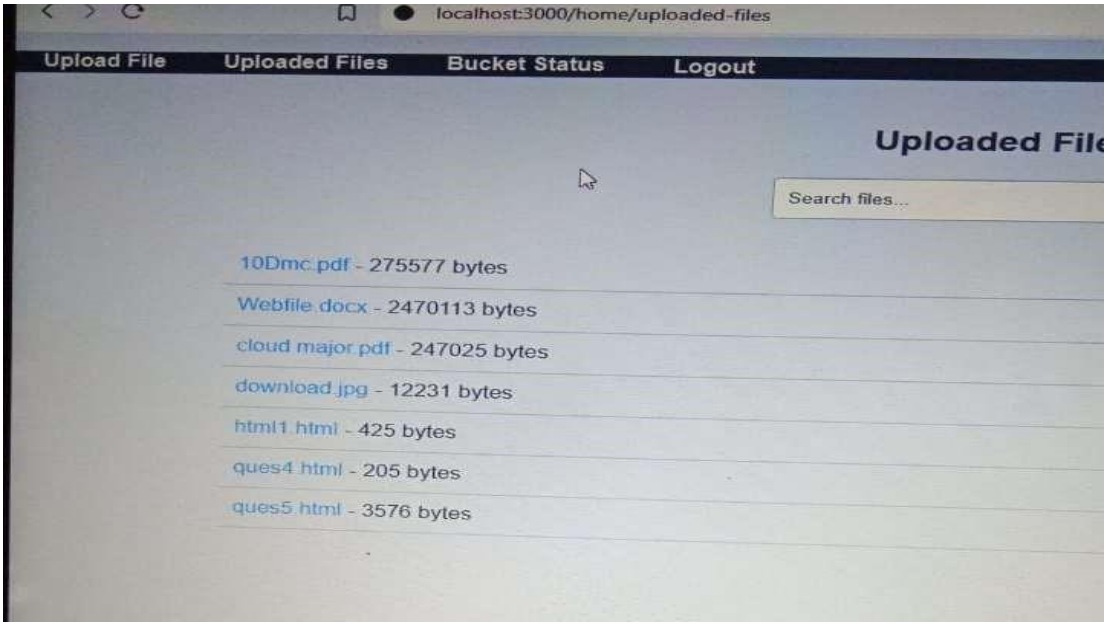

Figure 7.4 Uploaded File Status

8. REFERENCES

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

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

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

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