CLOUD DATA BACKUP & RECOVERY

# A PROJECT REPORT

|  |  |
| --- | --- |
| ***Submitted by*** | |
| **ASHWIN.V** | **(311619104301)** |
| **NAVEEN KUMAR.M** | **(311619104045)** |
| **SIVABALAN.J** | **(311619104067)** |

***in partial fulfilment for the award of degree of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE THORAIPAKKAM, CHENNAI-600097**

**ANNA UNIVERSITY: CHENNAI 600025 MAY 2023**

ANNA UNIVERSITY: CHENNAI 600 025

**BONAFIDE CERTIFICATE**

## Certified that this project report “ CLOUD DATA BACKUP & RECOVERY ” is the bonafide work of “ NAVEEN KUMAR.M , SIVABALAN.J

## ASHWIN.V ” who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| Dr. N.Savaranan B.E., ME., Ph.D | Ms. A. Nivashini B.E., ME |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR**  **ASSISTANT PROFESSOR** |
| Department of Computer Science and Engineering  Misrimal Navajee Munoth Jain Engineering College, Thoraipakkam,  Chennai-600097 | Department of Computer Science and Engineering  Misrimal Navajee Munoth Jain Engineering College, Thoraipakkam,  Chennai-600097 |

Submitted for the Project Viva-Voce Examination held on

# INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We express our sincere gratitude to our honorable **Secretary** (Administration) **Dr. Harish L Metha** and Secretary (Academic) **Shri L. Jaswant S Munoth** for providing the infrastructure facilities to do this project during our course period.

We thank our **Principal**, **Dr. C.C.Christopher** for his support and motivation for the development and completion of this project.

We express profound sense of gratitude and thanks to our **Head Of The Department**, **Dr.N.Saravanan** for his valuable suggestions and guidance for the development and completion of this project.

We thank our project coordinator, **Mrs.D.Yashwanthini , Assistant Professor**, for her valuable suggestions and constant encouragement that led us to the successful completion of our project.

We thank our supervisor **Ms.A.Nivashini , Assistant Professor** for her constant help and valuable suggestions.

We thank all the Teaching and Non-Teaching Staff members of our Department who helped us to complete our project.

Above all we thank the Almighty, and our parents and for their constant support and encouragement for completing this project.

# ABSTRACT

Cloud storage services have demonstrated significant strength and widespread acceptance, providing fundamental support for the rapid development of cloud computing. However, due to management incompetence and deliberate attacks, massive security events still result in large amounts of sensitive data leaking at the cloud storage level. This paper proposes a Cloud Secure Storage Mechanism (CSSM) to secure the secrecy of cloud data.

CSSM merged data dispersion and distributed storage to realize encrypted, chucked, and dispersed storage to avoid data breaches at the storage layer. CSSM also used a hierarchical management strategy and integrated user passwords with secret sharing to prevent the leakage of cryptographic materials.

The experimental findings show that the proposed method is not only suitable for maintaining data security at the storage layer against leaks, but it can also effectively store large amounts of cloud data without imposing too much overhead. For example, when users utilize CSSM to upload or download a 5G file, it only takes 646seconds/269seconds which is acceptable for users.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | **5** |
|  | **LIST OF FIGURES** | **9** |
| **1**  1.1 | **INTRODUCTION**  COMPUTER VISION | **11**  **12** |
| **2** | **LITERATURE SURVEY** | **13** |
| **3** | **SYSTEM OVERVIEW** | **15** |
| 3.1 | EXISTING SYSTEM | 15 |
| 3.2 | PROPOSED SYSTEM | 16 |
| 3.3 | REQUIREMENT ANALYSIS | 18 |
|  | 3.3.1 SOFTWARE REQUIREMENT | 18 |
|  | 3.3.2 HARDWARE REQUIREMENT | 18 |
| 3.4 | TECHNOLOGIES USED | 19 |
|  | 3.4.1 BOOTSTRAP | 19 |
|  | 3.4.2 JAVA | 20 |
|  | 3.4.3 POSRGRESQL  3.4.4 TOMCAT | 20    21 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4** | **SYSTEM DESIGN** | **22** |  |  |
| 4.1 | OVERVIEW OF ARCHITECTURE | 22 |  |  |
| 4.2 | USE CASE DIAGRAM | 23 |  |  |
| 4.3 | CLASS DIAGRAM | 24 |  |  |
| 4.4 | ACTIVITY DIAGRAM | 25 |  |  |
| 4.5 | DEPLOYMENT DIAGRAM | 27 |  |  |
| **5**  5.1  5.1.1  5.1.2  5.1.3  5.1.4  5.1.5    5.1.6  5.1.7  5.1.8  5.1.9    **6**  6.1    7  7.1  7.2  **FIGURE NO**  4.1  4.2  4.3  4.4  4.5 | **IMPLEMENTATION**  MODULES  PREREQUISITES  CHOOSING THE DEVELOPMENT TOOLS  CHOOSING CORE LIBRARIES AND FRAMEWORK  DECOY FILE ALGORITHM  FILE ENCRYPTION AND DISPERSION TECHNIQUE  BUILDING USER AUTHENTICATION  BUILDING FINAL APPLICATION  DEVELOPING UI  BUILDING SERVER  **SYSTEM TESTING**  TESTING OBJECTIVE  **CONCLUSION AND FUTURE**  **ENHANCEMENT**  CONCLUSION  FUTURE ENHANCEMENT  **APPENDICES**  **APPENDICES 1 –** SAMPLE CODING    **APPENDICES 2 –** SCREENSHOTS  **REFERENCES**  **LIST OF FIGURE**  **NAME OF THE FIGURE**  OVERVIEW OF ARCHITECTURE  USECASE DIAGRAM  CLASS DIAGRAM  ACTIVITY DIAGRAM  DEPLOYMENT DIAGRAM   |  |  |  | | --- | --- | --- | | **SNO** | **ABBREVIATION** | **EXPANSION** | | 1 | BaaS | Backup as a Service | | 2 | DLP | Data Loss Prevention | | 3 | VM | Visual Machine | | 4 | AWS | Amazon Web Services | | 5 | GCP | Google Cloud Platform | | 6 | AZURE | Microsoft Azure | | 7  8  9 | CDN  VTL  CSSM | Content Delivery Network  Virtual Tape Library  Cloud Secure Storage Mechanism |   **LIST OF ABBREVIATIONS** | **28**  28  28  30  31  33  34  35  36  38  39  41  41  44  44  45  47  47  52  56    **PAGE NO**  21  22  23  24  26 |  |  |

**CHAPTER 1**

**INTRODUCTION**

Cloud storage has become an essential component of modern information technology systems. It offers a convenient and cost-effective way of storing, sharing, and accessing data across different devices and locations. Cloud storage providers offer a wide range of services that allow users to store and manage various types of data, such as documents, images, videos, and music. However, the security of cloud storage remains a significant concern for many users. The data stored in the cloud is vulnerable to various security threats, such as hacking, data breaches, and unauthorized access. These threats can result in the loss or theft of sensitive and confidential data, which can have severe consequences for individuals and organizations. To address these security concerns, many cloud storage providers offer various security features, such as data encryption, access control, and data backups.

However, these features are not foolproof, and they may not provide adequate protection against sophisticated security threats. Therefore, there is a growing need for more robust and secure cloud storage mechanisms that can provide a high level of security for data stored in the cloud. The Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption is one such mechanism that offers a high level of security for data stored in the cloud. This mechanism is based on two key concepts: data dispersion and encryption. The data dispersion technique splits the data into several smaller data fragments and stores them in different locations in the cloud.

The encryption technique ensures that the data fragments are encrypted using a strong encryption algorithm before storing them in the cloud. The mechanism is scalable and can be easily implemented in a cloud environment. However, the mechanism requires additional overhead for data dispersion and encryption, which can affect the performance of the cloud storage system.

# 1.1 COMPUTER VISION:

Computer vision may not be the primary technology for local server backup and recovery, it may have some applications in this context. One potential use case for computer vision in local server backup and recovery is to help automate the backup and recovery process. For example, computer vision algorithms could be used to automatically identify files and folders that need to be backed up or recovered based on their content, without requiring manual intervention. Another potential use case for computer vision in local server backup and recovery is to help identify potential issues with backup and recovery processes.

For example, computer vision algorithms could be used to analyze images or videos of backup or recovery processes to identify any errors or anomalies that may need to be addressed. Overall, while computer vision may not be the primary technology for local server backup and recovery, it may have some applications in automating and improving these processes. However, it is important to note that backup and recovery software specifically designed for local servers is still likely to be the most appropriate solution for these tasks.

# CHAPTER 2 LITERATURE SURVEY

**A Cloud Secure Storage Mechanism Based On Data Dispersion And Encryption:**

CSSM(Cloud Secure Storage Mechanism), CSSM Description, Data Dispersion, Data Encryption The issue of cloud data leakage caused by management negligence and malicious attack at storage layer, CSSM, a cloud secure storage mechanism. CSSM adopted a combined approach of data dispersal and encryption. [4]

**An Efficient Secure System for Fetching Data from the Outsourced Encrypted Databases:**

Homomorphic Encryption, Randomized Encryption, Onion Encryption, Order-preserving Encryption which works as an inter-mediate between the cloud server and user(s) and performs all the crypto processes. In addition, we design a novel indexing technique based on predefining partitions for each sensitive attribute, and then encode each tuple to bits.[11]

**DropStore :- A Secure Backup System Using Multi-Cloud and Fog Computing:**

Data Backup On Multi-cloud, Droplet-cloud Side, Data Retrieval. The solution is based on Multi-Cloud and Fog Computing paradigms. Data security and user privacy are maintained by encryption and data partitioning on Multi-Cloud Storage. [10]

**Duplicacy:- A New Generation of Cloud Backup Tool Based on Lock-Free Deduplication:**

Chunking, Naming Chunks, Lock-free Deduplication, Full Snapshot They have propose a new cross-client deduplication solution called Lock-Free Deduplication to improve backup speeds at the cloud storage era. Its key idea can be summa- Rized Use a fixed-size or variable-size chunking algorithm to split files into chunks. [14]

**Dependable Data Outsourcing Scheme Based on Cloud-of-Clouds Approach with Fast Recovery:**

Hail, Cleversafe , CDStore , DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds Our scheme not only overcomes the four limitations to cloud storage, but also provides three special detection algorithms for different situations including a feature for determining whether an error exists and then, if one does localizing it. [3]

**CHAPTER 3**

**SYSTEM OVERVIEW**

* 1. **EXISTING SYSTEM**

Cloud Secure Storage Mechanism named CSSM to avoid data breach at the storage layer, CSSM integrated data dispersion and distributed storage to realize encrypted chucked and distributed storage . CSSM adopted a hierarchical management approaches and combined user password with secret sharing to prevent cryptographic materials leakage.

* A secure distributed storage system based on Hadoop system , which keep the confidentiality of cloud data dispersion and encryption . It performs the data decryption and assembly tasks before reading data .
* To prevent the keys from being stolen, this method requires key cache server and all keys should be stored in memory only.

**DRAWBACKS:**

The use of a decoy method for cloud data backup and recovery on a local server has some potential drawbacks, such as:

* **Increased maintenance requirements**: The use of a local server for cloud data backup and recovery requires additional maintenance tasks, such as hardware maintenance, software updates, and security patches. These tasks can be time-consuming and may require specialized skills and resources, increasing the total cost of ownership.
* **Single point of failure**: Using a single local server for cloud data backup and recovery creates a single point of failure, meaning that if the server fails, all backup and recovery processes will be impacted. This can lead to data loss and downtime, causing significant disruptions to business operations.
* **Limited disaster recovery options**: In the event of a disaster, such as a fire or flood, the local server may be damaged, leading to data loss. A cloud-based backup solution would provide more comprehensive disaster recovery options, including data redundancy and geographic replication.

# PROPOSED SYSTEM

In this project , we try improve the security even after the data gets stolen, with a set of decoy files imitating the original file in terms of size and other factors. The decoy file then goes to similar processing as the original file and gets stored in the system. This improves the possibility of attacker unable to understand and decipher the original data from the set of encrypted data he stole.

* **Decoy Creation :** A unreadable/usable random file with similar size and factors to the original file is created.
* **Encryption:** Each fragment is encrypted using a secure encryption algorithm to ensure that the data is protected even if one of the fragments is compromised.
* **Data Dispersion:** The data is divided into multiple parts, called fragments, and each fragment is stored in a different location within the cloud.
* **Storage:** The encrypted fragments are stored in different storage locations within the cloud to provide redundancy and improve the availability of the data.
* **Data Retrieval:** When the user wants to retrieve the data, the fragments are retrieved from their storage locations, decrypted and reassembled to provide the original data.

**ADVANTAGES:**

Some potential advantages of using a decoy method for cloud data backup and recovery on a local server:

* **Improved data security**: By using a decoy method, the actual data is hidden and protected from unauthorized access. This provides an additional layer of security, reducing the risk of data breaches and cyber attacks.
* **Increased data privacy**: By keeping the actual data encrypted and dispersed, the privacy of the data is maintained. This is particularly important for organizations that deal with sensitive or confidential information, such as financial or healthcare data.
* **Cost savings**: Using a local server for cloud data backup and recovery may be more cost-effective than using a cloud-based solution, particularly for smaller organizations with limited budgets. This is because cloud-based solutions often require ongoing subscription fees or usage charges.
* **Faster recovery times**: With a local server, data recovery times may be faster than with a cloud-based solution, particularly if the backup data is stored on-site. This can minimize downtime and disruption to business operations in the event of data loss.

# REQUIREMENT ANALYSIS

The requirement specification is a technical specification of requirements for the software products. The purpose of the software requirement specification is to provide a detailed overview of the software project, its parameter and goal. It describes the project target audience and its user interface, hardware and software requirements.

# SOFTWARE REQUIREMENT

The software requirements give a detailed description of the system and all its features.

* Eclipse
* Bootstrap
* PosrgreSQL
* Tomcat

# HARDWARE REQUIREMENT

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete engineer as the starting point for the system design.

Ram : 8GB Ram or more

Processor : Any Intel Processor

GPU : 4GB or more

Hard Disk : 10GB or more

Speed : 1.4GHZ or more

# TECHNOLOGIES USED

**3.4.1 BOOTSTRAP**

Bootstrap is a free and open-source front-end development framework used for creating responsive and mobile-first websites and web applications. It was developed by Twitter and is now maintained by the Bootstrap Core Team and a large community of developers.

Bootstrap provides a range of pre-built CSS, JavaScript, and HTML components that can be used to create responsive and mobile-first websites and web applications. These components include typography, forms, buttons, navigation, modals, and more. Bootstrap also includes a grid system, which allows developers to easily create responsive layouts for their websites.

Here are some of the key features of Bootstrap:

* **Responsive design**
* **Cross-browser compatibility**
* **Customizable**
* **Easy to use**
* **Large community**

**3.4.2 JAVA**

Java is a high-level, class-based, object-oriented programming language developed by Sun Microsystems (now owned by Oracle Corporation) in 1995. Java is designed to be platform-independent, meaning that it can run on any platform that has a Java Virtual Machine (JVM) installed, making it a popular language for developing cross-platform applications.

Here are some of the key features of Java:

* **Object-oriented**
* **Platform-independent**
* **Memory management**
* **Security**

**3.4.3 PostgreSQL**

PostgreSQL is a powerful open-source object-relational database management system (ORDBMS) that is widely used for storing and managing large amounts of data.

It was first released in 1989 by the PostgreSQL Global Development Group, and it is available under the PostgreSQL License, which is a permissive open-source license.

Here are some of the key features of PostgreSQL:

* **Object-relational database**
* **Open-source**
* **ACID compliance**
* **Extensible**
* **Scalable**

**3.4.4 TOMCAT**

Apache Tomcat, often referred to simply as Tomcat, is an open-source web server and servlet container that is used to run Java-based web applications. It was first released in 1999 by the Apache Software Foundation and is available under the Apache License.

Here are some of the key features of Tomcat:

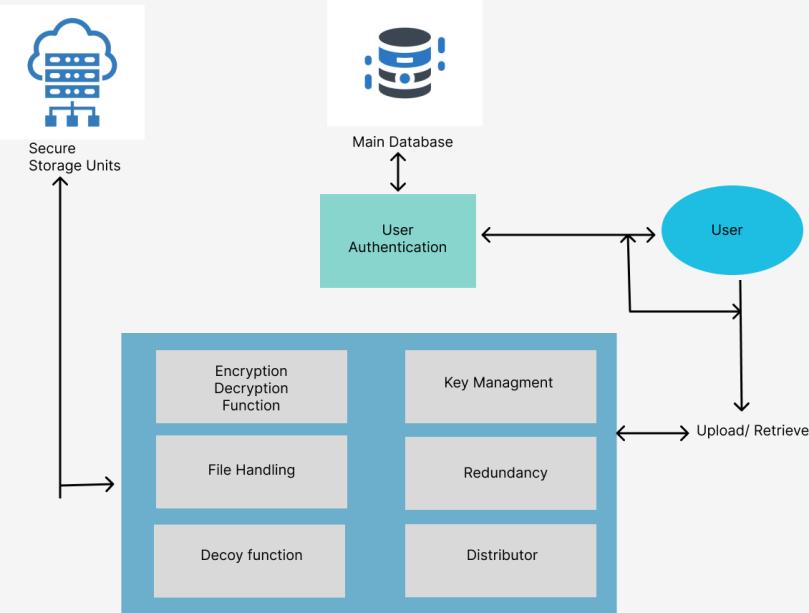
* **Servlet container**
* **Web server**
* **Cross-platform**
* **Lightweight**
* **Extensible**

# CHAPTER 4

# SYSTEM DESIGN

* 1. **OVERVIEW OF ARCHITECTURE**

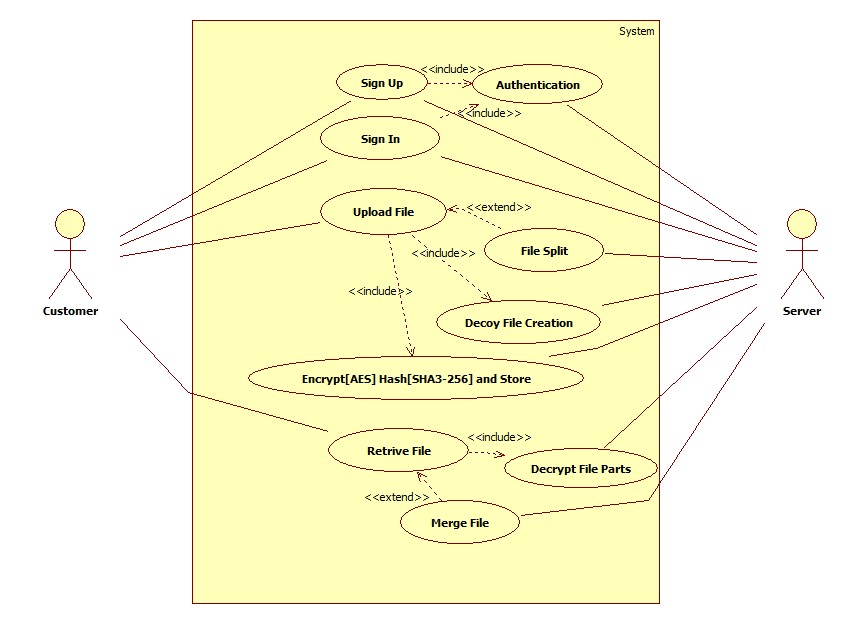
The diagram shows the overall architecture of the project which consists of the web pages and server block. Each block shows their structural and functional components of the project.



**Figure 4.1 System Architecture**

# USE CASE DIAGRAM

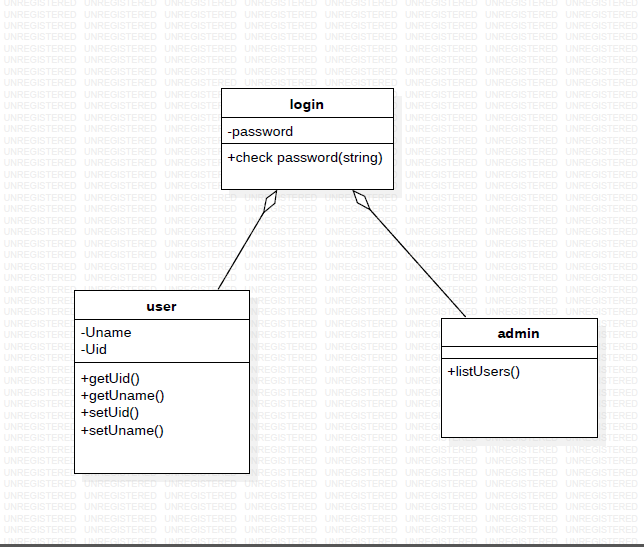
This diagram shows the user such as actor, system and the role of developer in this project. This behavior diagram models the functionality of the system using use cases.



**Figure 4.2 Use Case Diagram**

# CLASS DIAGRAM

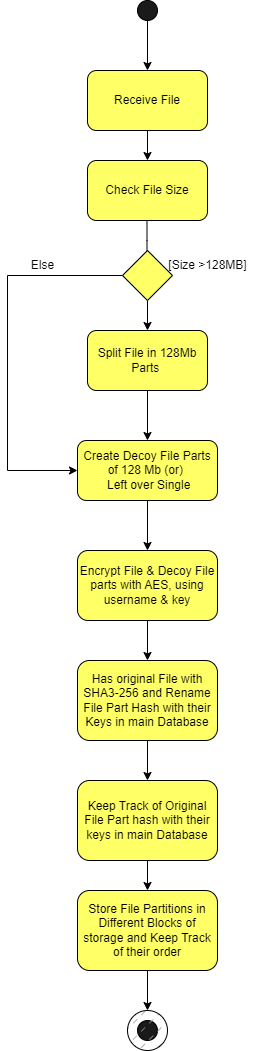
The structure of the application id described in the class diagram by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects. Each class has their attributes and the operations defined with the relationship between the classes.



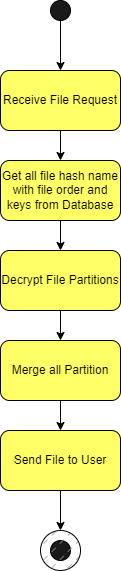
**Figure 4.3 CLASS DIAGRAM**

# ACTIVITY DIAGRAM

This module diagram represents the flow from one activity to another activity. The activity can be described as an operation of the system. Some activities are based on conditions satisfied by the actor/object.

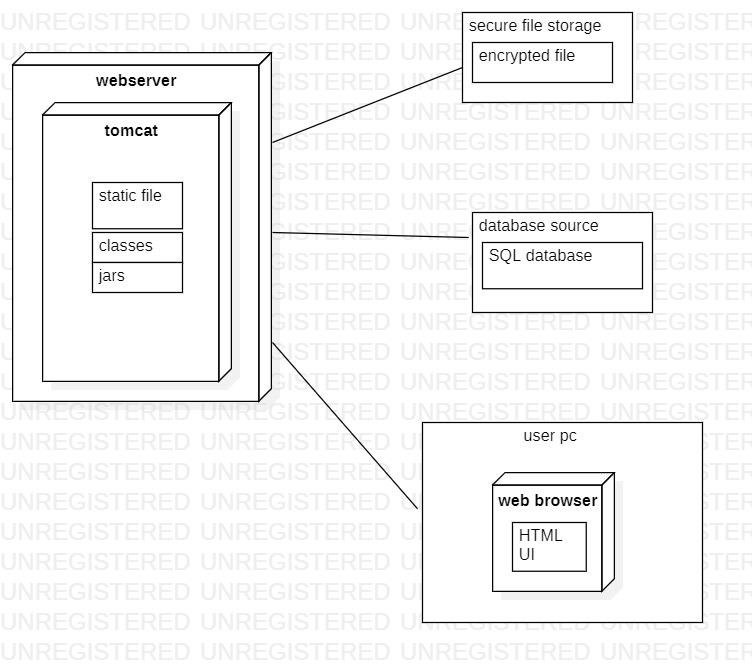


**Figure 4.4.1 ACTIVITY DIAGRAM**



**Figure 4.4.2 ACTIVITY DIAGRAM**

**4.5 DEPLOYMENT DIAGRAM**

****

**Figure 4.5 DEPLOYMENT DIAGRAM**

# CHAPTER 5 IMPLEMENTATION

* 1. **MODULES**
* Prerequisites
* Choosing the development tools.
* Choosing core libraries and framework
* Decoy file algorithm
* File encryption and dispersion technique
* Building user authentication.
* Building final application.
* Developing UI
* Building server
  + 1. **PREREQUITES**

The project "A Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption" requires numerous prerequisites to ensure successful implementation.These prerequisites include: Cloud infrastructure: The challenge requires a cloud infrastructure that can guide the storage and retrieval of data. This infrastructure should provide high scalability, availability, and reliability to ensure that the data is always accessible.

* **Data dispersion technique:** The project requires the implementation of a data dispersion technique that could disperse the data across more than one cloud server. This technique needs to ensure that even if one server is compromised, the data stays safe.
* **Encryption algorithm:** The project requires the implementation of a strong encryption algorithm to ensure the confidentiality and integrity of the data. The encryption set of rules needs to be able to protect the data from unauthorized access and malicious attacks.
* **Key management system:** The project requires the implementation of a key management system that may manipulate the encryption keys used to encrypt and decrypt the data. The key management system needs to ensure that the keys are restricted and only accessible to authorized employees.
* **Authentication and access control:** The project requires the implementation of an authentication and access control mechanism to ensure that only authorized employees can access the facts. The mechanism has to provide secure personal authentication and authorization to prevent unauthorized access to the data.
* **Data backup and recovery:** The project requires the implementation of a data backup and recovery mechanism to make certain that the data is always available, even in the event of a disaster or hardware failure.

Overall, those prerequisites are important to ensure the safety and reliability of the cloud storage mechanism and protect the records from unauthorized access and malicious attacks.

* + 1. **CHOOSING THE DEVELOPMENT TOOLS:**

The choice of development tools for the project "A Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption" will depend on several factors such as the programming language, the cloud infrastructure used, and the specific requirements of the project. Here are some tools that could be considered:

* **Programming language:** The programming language chosen for the project will depend on the developer's experience and expertise. Common languages used for cloud-based applications include Java, Python, and C#. Language: JAVA well trusted by enterprises. Well developed. Java has a vast library of pre-built classes and APIs. Java has built-in security features that make it a good choice for projects.
* **Cloud infrastructure:** The development tools chosen should be compatible with the cloud infrastructure used for the project. For example, if the project is hosted on Amazon Web Services (AWS), the developer might consider using AWS SDKs and APIs for development.
* **Integrated Development Environment (IDE):** An IDE is a software application that provides a comprehensive development environment for coding, testing, and debugging. Popular IDEs for cloud-based applications include Eclipse, Visual Studio, and IntelliJ IDEA.
* **Version control:** A version control system such as Git can help manage the source code and track changes made by multiple developers.
* **Encryption libraries:** The developer might consider using encryption libraries such as Bouncy Castle or OpenSSL to implement encryption algorithms.
* **Key management:** For key management, the developer might consider using open-source tools such as Hashicorp Vault or AWS Key Management Service (KMS).
* **Authentication and access control:** The developer might consider using Identity and Access Management (IAM) services provided by the cloud infrastructure provider or implementing a third-party authentication service such as OAuth.
  + 1. **CHOOSING CORE LIBRARIES AND FRAMEWORK**

When choosing core libraries and frameworks for a cloud data backup and recovery project, there are a few key factors to consider. Here are some options to consider:

* **Cloud Storage Providers**: One of the key components of a cloud backup and recovery solution is cloud storage. Consider using a cloud storage provider like Amazon S3, Google Cloud Storage, or Microsoft Azure Blob Storage to store the backed-up data. These cloud storage providers provide scalable, reliable, and secure storage options, making them a popular choice for cloud backup solutions.
* **Backup and Recovery Libraries**: There are several open-source libraries available that can be used for backup and recovery purposes, including Bacula, Amanda, and BackupPC . These libraries provide a range of backup and recovery features, including incremental backups, compression, encryption, and scheduling.
* **Monitoring and Alerting Libraries**: To ensure that backups are running correctly and to receive notifications in case of any issues, it's important to choose monitoring and alerting libraries. Some popular options include Nagios, Zabbix, and Prometheus. These libraries provide real-time monitoring of the backup process and send alerts in case of any errors or issues.
* **Logging and Analytics Frameworks**: It's important to track backup and recovery performance and analyze backup data for insights. Choose a logging and analytics framework such as Elasticsearch, Logstash, and Kibana (ELK stack) or Apache Hadoop and Apache Spark for analyzing and visualizing the backup data.
* **Encryption Libraries**: To ensure the security of the backed-up data, encryption libraries should be used. Some popular encryption libraries include OpenSSL, Cryptlib, and Bouncy Castle.
* **API Frameworks**: If you want to expose backup and recovery functionality through an API, you can choose API frameworks such as Spring Boot or Django REST Framework to create RESTful APIs. These frameworks provide a range of features for building secure and scalable APIs.

**5.1.4 DECOY FILE ALGORITHM:**

The decoy file algorithm is a method used in the project "A Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption" to enhance the safety of the stored data. It involves developing decoy documents that might be similar in length and format to the actual documents, but contain random statistics. These decoy files are saved in the cloud storage along with the actual files and are used to make it difficult for an attacker to identify the real records.

The decoy file algorithm works by means of creating a set of decoy files with the same size and layout as the real files. Those dummy files are then encrypted using the same encryption algorithm and key as the actual files. The encrypted decoy files are then dispersed across multiple cloud servers using the same data dispersion technique used for the actual files.

When an attacker tries to access the stored data, they will be unable to distinguish between the real files and the decoy files for the reason that they have the same size and layout. Even if the attacker manages to get access to the decoy files, they will not be able to decrypt them without the correct encryption key. This technique can help prevent unauthorized access and provide a further layer of protection for the stored data.

In conclusion, the decoy file set of rules is a technique used within the project "A Cloud Secure Storage Mechanism Based on Data Dispersion and Encryption" to enhance the security of the stored data. It involves creating decoy files that are similar in length and format to the real files but include random data. This approach can help prevent unauthorized access and offer an extra layer of protection for the saved data.

* + 1. **FILE ENCRYPTION AND DISPERSION TECHNIQUE:**

File encryption and dispersion techniques are critical components of a secure storage mechanism. Here's a rundown of how you can put them into action:

**File Encryption:**

The encryption technique involves converting plain-text content or data into an unreadable format using a cryptographic algorithm. The encryption algorithm generates a key that is used to fasten and unlock the encrypted data. There are several encryption algorithms available, which include the advanced encryption standard (AES), the data encryption standard (DES), and RSA.

To enforce file encryption for your project, you can use one of these algorithms or a combination of them to encrypt the data before storing it in the cloud. This will ensure that even if unauthorized people access your data, they won't be able to read it without the decryption key.

**Data Dispersion:**

The data dispersion technique involves splitting a file into more than one smaller part and storing them in different locations. This makes it more difficult for an attacker to gain access to the entire report, even if they have control over a portion of it.To implement data dispersion in your project, you could split the files into several smaller parts with the use of a dispersal algorithm. The algorithm needs to ensure that the dispersed data is distributed across different cloud storage places. This will provide an extra layer of protection, as an attacker will want to access a couple of storage locations to reconstruct the file.

* + 1. **BUILDING USER AUTHENTICATION**

To implement user authentication for your cloud data backup and recovery project, you could follow these steps:

* **Choose an authentication method**: There are several authentication methods available, such as username and password, multi-factor authentication (MFA), and biometric authentication. Select the one that best suits your project's requirements.
* **Implement the authentication method**: Once you have chosen the authentication method, you need to implement it in your project. This may involve adding new code to your backup and recovery script or integrating with an existing authentication service.
* **Create user accounts**: Create user accounts for each user who needs access to the backup and recovery system. Each user account should have a unique username and password or other authentication credentials.
* **Define user roles and permissions**: Define the roles and permissions for each user account. For example, some users may only be allowed to view backups, while others may be allowed to restore data.
* **Test the authentication system**: Perform thorough testing of the authentication system to ensure that it is working correctly. Test for common security vulnerabilities such as password guessing and brute force attacks.
* **Monitor and maintain the authentication system**: Monitor the authentication system to ensure that it is running smoothly and that user accounts are up-to-date. Also, perform periodic maintenance to ensure that the system is up-to-date with the latest security patches and updates.

By following these steps, you should be able to build user authentication for your cloud data backup and recovery project, helping to ensure that only authorized users can access and interact with your backups.

* + 1. **BUILDING FINAL APPLICATION**

To build the final application for your cloud data backup and recovery project, you could follow these steps:

* **Define the application requirements**: Define the functional and non-functional requirements of the application. This includes the features, performance, scalability, security, and user experience.
* **Choose a programming language and framework**: Choose a programming language and framework that best suits your project's requirements. Some popular choices for cloud-based applications include Java, Python, Node.js, and Ruby on Rails.
* **Design the user interface**: Design a user interface that is intuitive and easy to use. The user interface should provide access to all the application's features, including backup and recovery options.
* **Implement the backup and recovery features**: Implement the backup and recovery features using the cloud service provider's APIs and tools. This may involve writing custom code to transfer data to and from the cloud, as well as managing backups and restores.
* **Implement user authentication and authorization**: Implement user authentication and authorization to ensure that only authorized users can access and interact with the backups. This may involve integrating with an existing authentication service or building your own authentication system.
* **Test the application**: Test the application thoroughly to ensure that it meets all the requirements and works as expected. This includes functional testing, performance testing, and security testing.
* **Deploy the application to the cloud**: Deploy the application to the cloud service provider of your choice. This may involve setting up the necessary infrastructure, such as virtual machines, databases, and load balancers.
* **Monitor and maintain the application**: Monitor the application to ensure that it is running smoothly and that backups are being performed as expected. Also, perform periodic maintenance to ensure that the system is up-to-date with the latest security patches and updates.

By following these steps, you should be able to build the final application for your cloud data backup and recovery project. The application should be able to backup and recover data to and from the cloud, while providing a secure and user-friendly interface for users to interact with the backups.

* + 1. **DEVELOPING UI**

To develop the user interface (UI) for your cloud data backup and recovery project, you could follow these steps:

* **Define the user interface requirements**: Define the requirements of the user interface, including the functionality, design, and user experience. This should include a list of features that the user interface should have, such as backup and recovery options, search functionality, and user authentication.
* **Choose a UI framework**: Choose a UI framework that best suits your project's requirements. Some popular choices for web-based applications include React, Angular, and Vue.js.
* **Design the user interface**: Design the user interface based on the requirements and the chosen UI framework. This may involve wireframing and prototyping to ensure that the UI meets the functional and design requirements.
* **Implement the user interface**: Implement the user interface using the chosen UI framework and other web development technologies such as HTML, CSS, and JavaScript. This may involve writing custom code to integrate with the cloud service provider's APIs and tools.
* **Test the user interface**: Test the user interface to ensure that it meets the requirements and works as expected. This includes functional testing, usability testing, and performance testing.
* **Incorporate user feedback**: Incorporate feedback from users and stakeholders to improve the user interface. This may involve making changes to the design or functionality based on user feedback.
* **Deploy the user interface to the cloud:** Deploy the user interface to the cloud service provider of your choice. This may involve setting up the necessary infrastructure, such as virtual machines, databases, and load balancers.
* **Monitor and maintain the user interface**: Monitor the user interface to ensure that it is running smoothly and that backups are being performed as expected. Also, perform periodic maintenance to ensure that the system is up-to-date with the latest security patches and updates.
  + 1. **BUILDING SERVER**
* **Choose the hardware**: The first step is to select the hardware on which you will build your server. You can use an old computer or build a new one with high-speed processors, large storage capacity, and a reliable power supply.
* **Choose the operating system**: The next step is to select the operating system that you will use to run the server. You can choose any operating system that supports the required software and has a good reputation for security and stability.
* **Install the necessary software**: The server will require software to manage the backup and recovery process, such as a backup software, a decoy data generator, and data dispersion system. You will also need a key management system to store and manage encryption keys.
* **Configure security settings**: To ensure the security of your server, you will need to configure security settings such as firewalls, intrusion detection, and prevention systems. You can also use encryption and other security measures to secure your server and backup data.
* **Configure backup and recovery settings**: Configure your backup software to create regular backups of your data and store them in dispersed locations with decoy data sets. You will also need to configure recovery software to restore your data from the backups in case of any data loss.
* **Test the server**: Before deploying the server, you should test it thoroughly to ensure that it is functioning correctly and that your backup and recovery systems are working as expected.
* **Deploy the server**: Once you have tested your server and confirmed that it is working correctly, you can deploy it in your organization.

By following these steps, you can build a reliable and secure server for your cloud data backup and recovery project using the decoy method and a local server.

# CHAPTER 6 SYSTEM TESTING

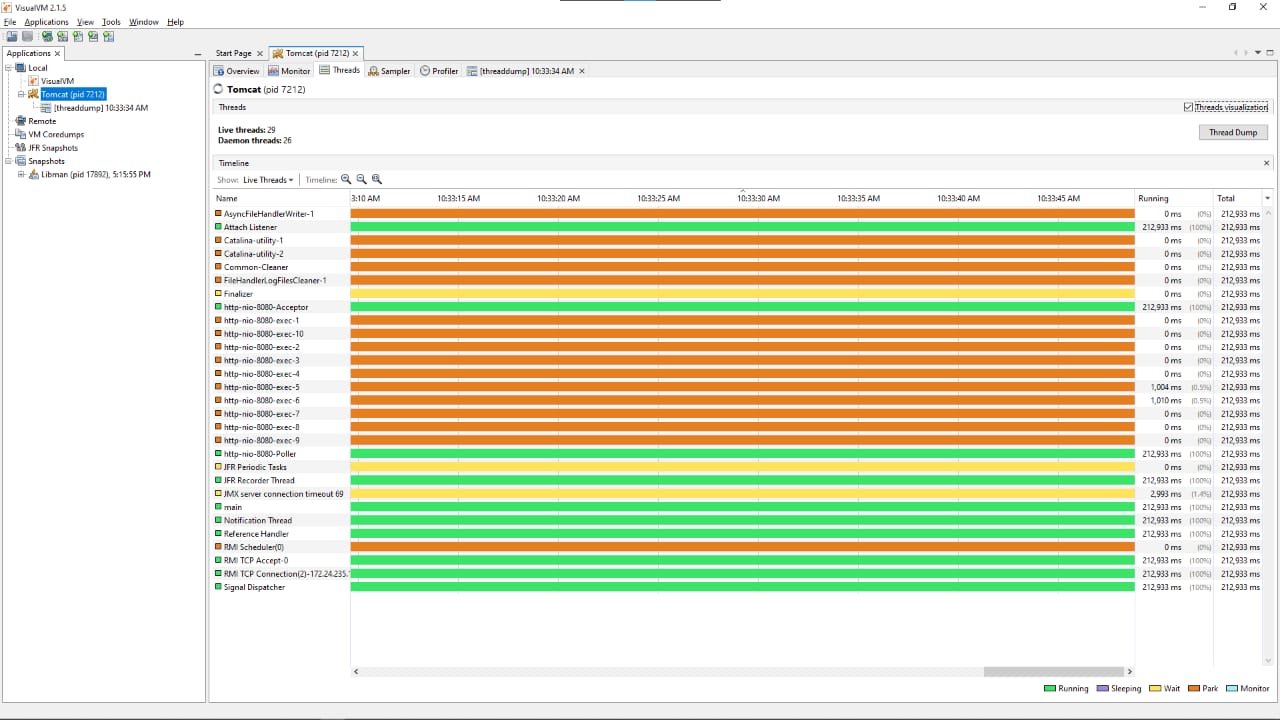
* 1. **TESTING OBJECTIVES**

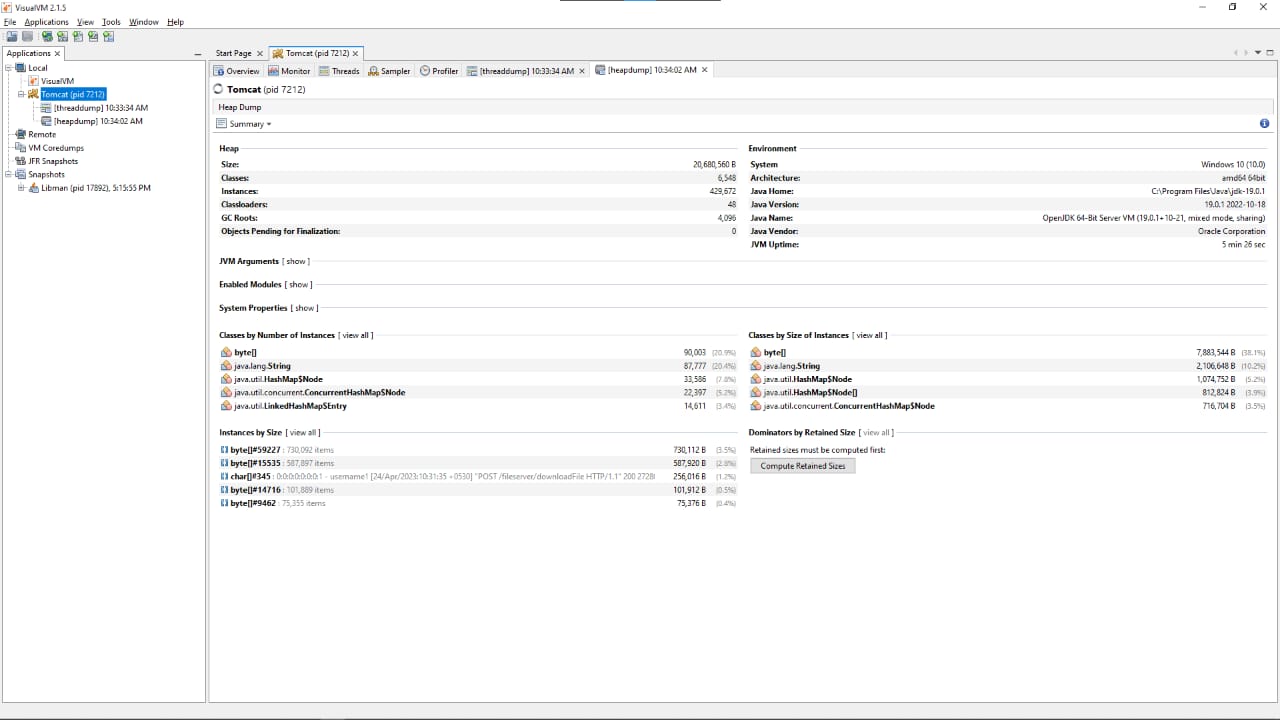
The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

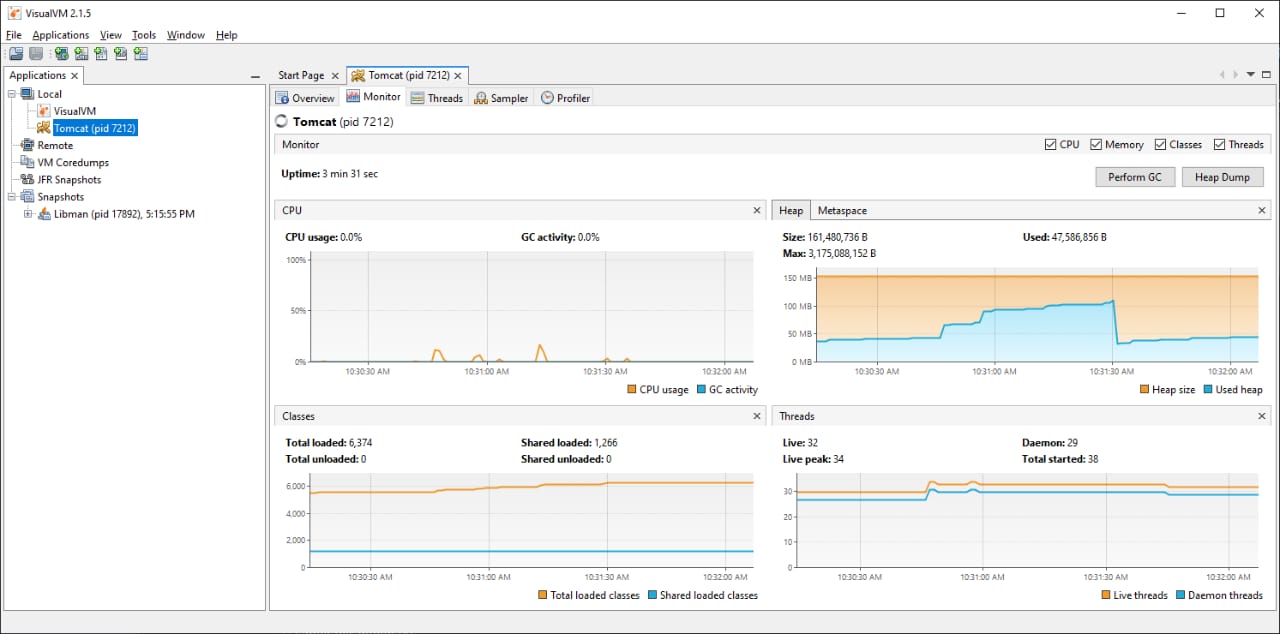
VisualVM is a Java profiling and monitoring tool, which is included in the Java Development Kit (JDK) distribution. It provides a graphical user interface (GUI) for monitoring and analyzing Java applications. Some of the key features of VisualVM include:

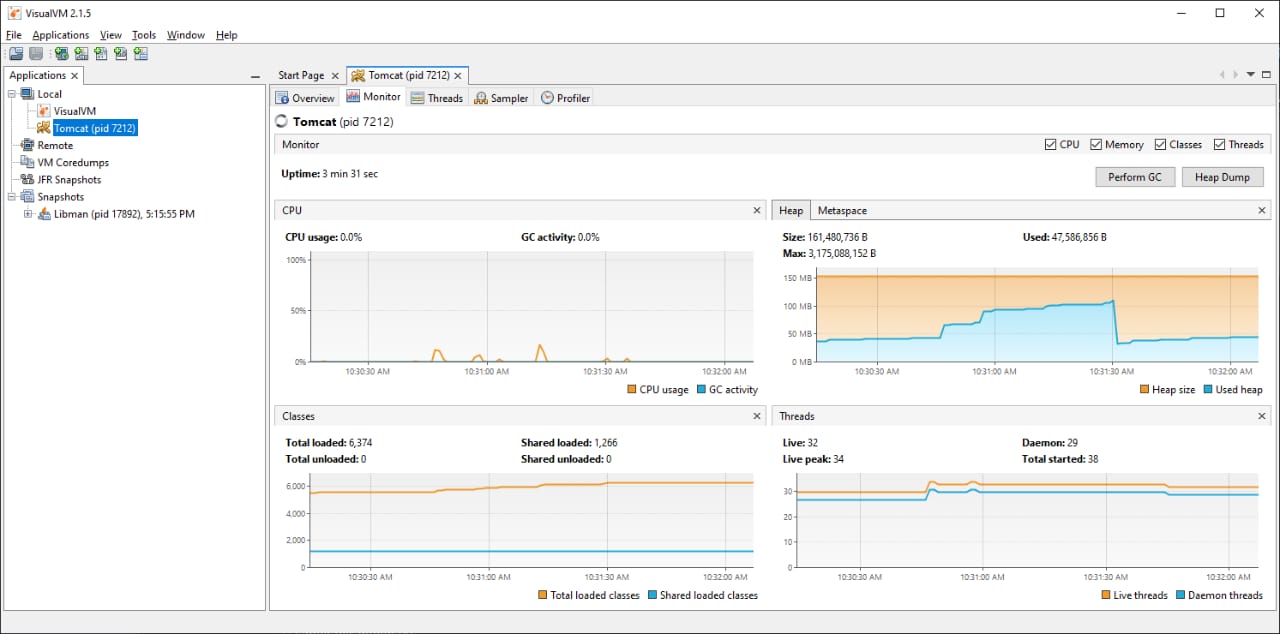
* **Profiling**
* **Heap dump analysis**
* **Thread dump analysis**
* **JMX monitoring**

VisualVM supports a variety of plugins that extend its functionality. For example, the Visual GC plugin provides a visual representation of the Java heap and garbage collection activity, while the Sampler plugin can be used to sample method execution times. Overall, VisualVM is a powerful tool for profiling and monitoring Java applications, and it can help developers identify and resolve performance and memory-related issues.









**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**7.1 CONCLUSION**

In conclusion, cloud data backup and recovery is an essential component of any organization's disaster recovery and business continuity plan. With the increasing amount of data being generated and stored, organizations need a reliable and secure way to backup and recover their data to ensure continuity of their operations. Cloud-based backup and recovery solutions offer several advantages over traditional on-premises solutions, including scalability, cost-effectiveness, and ease of use.

However, in addition to traditional backup and recovery methods, organizations may also consider using decoy methods as a supplementary approach to protect their data. Decoy methods involve using fake data to mislead attackers, thereby reducing the risk of successful attacks and data breaches. Decoy methods can be used in conjunction with other security measures to provide an additional layer of protection for sensitive data.

Overall, implementing a cloud data backup and recovery solution with decoy methods can help organizations better protect their data and ensure business continuity in the face of disasters or security breaches. It is important to carefully assess the organization's requirements and choose the appropriate backup and recovery solution that best meets their needs. Additionally, regular testing, monitoring, and maintenance of the backup and recovery system are critical to ensure its effectiveness and reliability.

**7.2 FUTURE ENHANCEMENT**

In the future, cloud data backup and recovery solutions are likely to continue evolving and improving to meet the ever-increasing demands of organizations. Here are some potential future enhancements that can be considered for cloud data backup and recovery solutions:

* **Advanced encryption techniques**: While most cloud backup solutions provide encryption capabilities, advanced encryption techniques such as homomorphic encryption, quantum-resistant encryption, and multi-party computation can provide even greater levels of data security.
* **Artificial Intelligence (AI) & Machine Learning (ML) integration**: By integrating AI and ML into cloud backup and recovery solutions, organizations can automate several aspects of the backup and recovery process, such as identifying critical data, predicting potential issues, and optimizing backup schedules.
* **Blockchain technology integration**: Blockchain technology can provide a tamper-proof record of all backup and recovery activities, providing an additional layer of security and ensuring data integrity.
* **Enhanced Decoy Methods**: As attackers become more sophisticated, decoy methods can be enhanced by using advanced deception technologies such as honeypots, honeynets, and honeytokens.
* **Integration with hybrid cloud environments**: Organizations are increasingly using hybrid cloud environments to store and process data. Cloud backup and recovery solutions can be enhanced by providing seamless integration with hybrid cloud environments, ensuring data consistency and reducing data loss risks.

Incorporating decoy methods with these future enhancements could further improve the security posture of cloud data backup and recovery solutions. Organizations must continuously monitor and evaluate their backup and recovery systems to ensure that they are up-to-date with the latest technology and security practices.

# APPENDICES APPENDIX 1 SAMPLE CODING

index.jsp

<%@ page session="false" %>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.2.0/css/bootstrap.min.css"

integrity="sha512-XWTTruHZEYJsxV3W/lSXG1n3Q39YIWOstqvmFsdNEEQfHoZ6vm6E9GK2OrF6DSJSpIbRbi+Nn0WDPID9O7xB2Q=="

crossorigin="anonymous" referrerpolicy="no-referrer" />

<link rel="stylesheet" href="./css/main.css">

<link rel="stylesheet" href="./css/login.css">

<script src="https://code.jquery.com/jquery-1.10.2.js" type="text/javascript"></script>

</head>

<body>

<body class="bg-dark">

<div class="main-container">

<div class="login-container p-5">

<a href="./home.jsp"><button type="button" class="btn btn-dark butt mt-3">User</button></a>

<a href="./admin.jsp"><button type="button" class="btn btn-dark butt mt-3">Admin</button></a>

<a href="./signup.jsp" class="link-text mt-2">new user?</a>

</div>

</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.2.0/js/bootstrap.bundle.min.js"

integrity="sha512-9GacT4119eY3AcosfWtHMsT5JyZudrexyEVzTBWV3viP/YfB9e2pEy3N7WXL3SV6ASXpTU0vzzSxsbfsuUH4sQ=="

crossorigin="anonymous" referrerpolicy="no-referrer"></script>

<script>

function runtoast() {

const toasts = document.getElementsByClassName('toast')

for (i = 0; i < toasts.length; i++) {

let elem = toasts.item(i)

let elem\_tost = new bootstrap.Toast(elem)

elem\_tost.show()

}

}

runtoast();

</script>

</body>

</html>

</body>

</html>

ArgonCredentialHandler.java

import org.apache.catalina.CredentialHandler;

import org.springframework.security.crypto.argon2.Argon2PasswordEncoder;

public class ArgonCredentialHandler implements CredentialHandler{

public boolean matches(java.lang.String arg0,java.lang.String arg1){

Argon2PasswordEncoder enc = new Argon2PasswordEncoder(16, 32, 1, 65536, 10);

return enc.matches(arg0, arg1);

}

public java.lang.String mutate(java.lang.String arg0){

Argon2PasswordEncoder encoder=new Argon2PasswordEncoder(16,32,1,65536,10);

return encoder.encode(arg0);

}

}

FolderView.java

import java.io.File;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import com.fasterxml.jackson.annotation.JsonProperty;

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.SerializationFeature;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

public class FolderView extends HttpServlet {

@Override

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

ObjectMapper mapper = new ObjectMapper().enable(SerializationFeature.INDENT\_OUTPUT);

File folder1 = new File("C:\\sivabalan\\dev\\fserver\\f1\\");

File folder2 = new File("C:\\sivabalan\\dev\\fserver\\f2\\");

File folder3 = new File("C:\\sivabalan\\dev\\fserver\\f3\\");

folder1.mkdirs();

folder2.mkdirs();

folder3.mkdirs();

File[] files1 = folder1.listFiles();

File[] files2 = folder2.listFiles();

File[] files3 = folder3.listFiles();

ArrayList<String> filesArr1=new ArrayList<String>();

ArrayList<String> filesArr2=new ArrayList<String>();

ArrayList<String> filesArr3=new ArrayList<String>();

for (File file : files1) {

if (file.isFile()) {

filesArr1.add(file.getName());

}

}

for (File file : files2) {

if (file.isFile()) {

filesArr2.add(file.getName());

}

}

for (File file : files3) {

if (file.isFile()) {

filesArr3.add(file.getName());

}

}

FolderViewResponse fvr=new FolderViewResponse(filesArr1, filesArr2, filesArr3);

String outstr=mapper.writeValueAsString(fvr);

resp.setContentType("application/json");

resp.setCharacterEncoding("UTF-8");

PrintWriter out=resp.getWriter();

out.print(outstr);

out.flush();

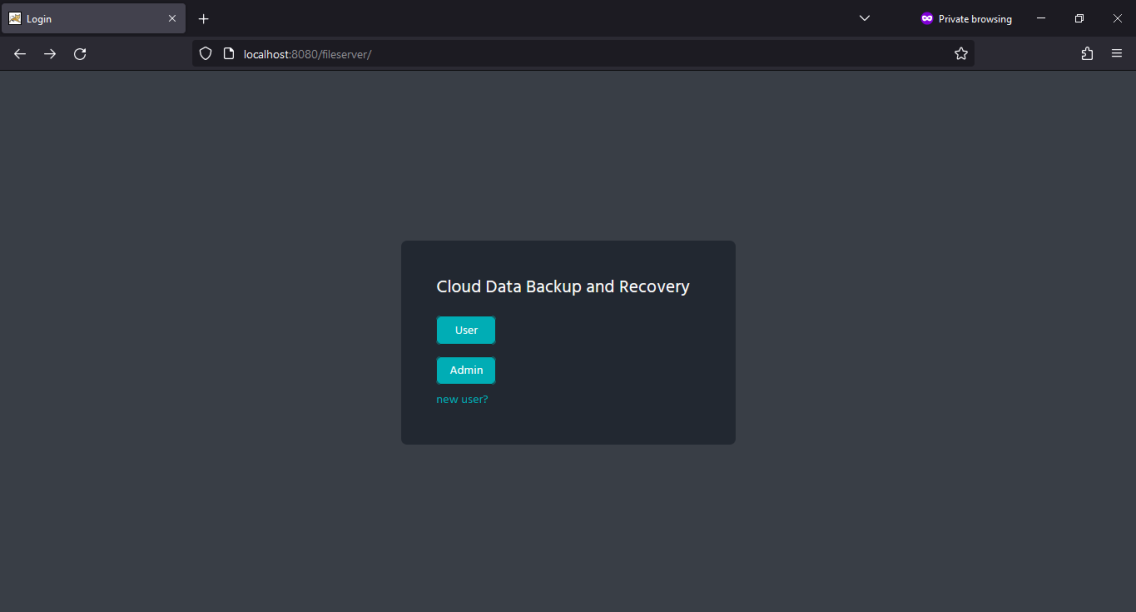
}

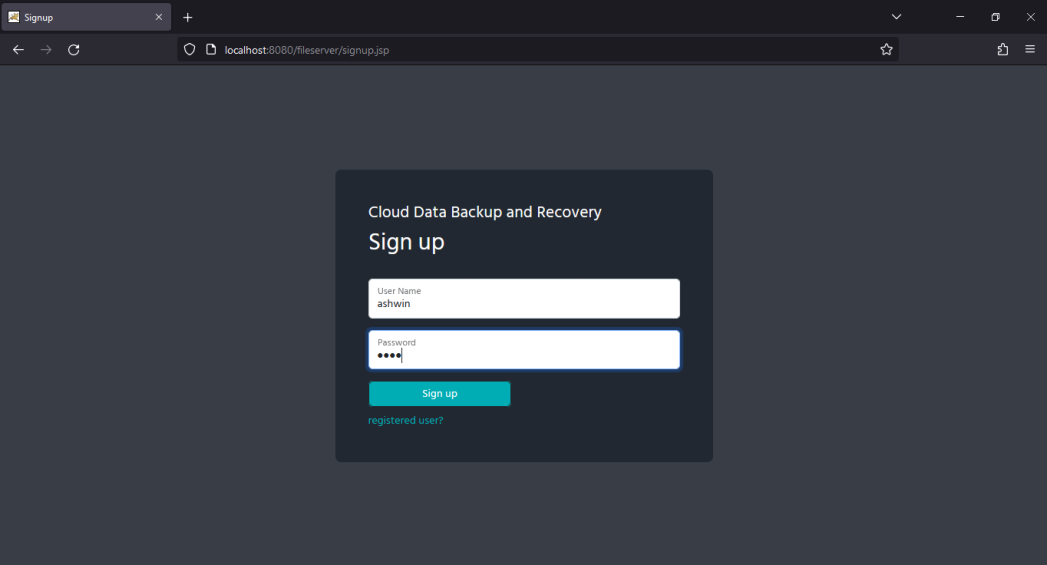
}

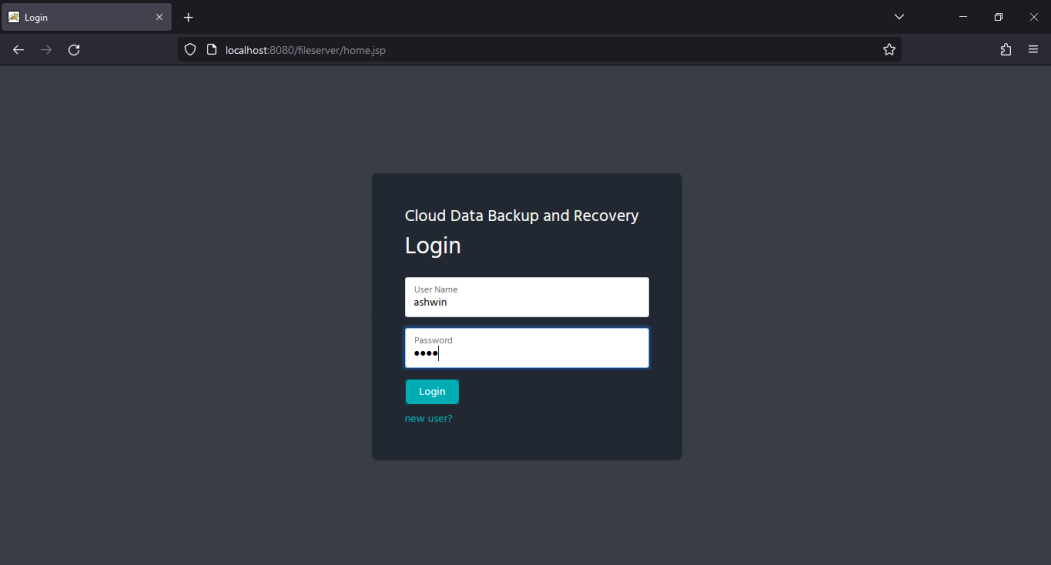
# APPENDIX 2

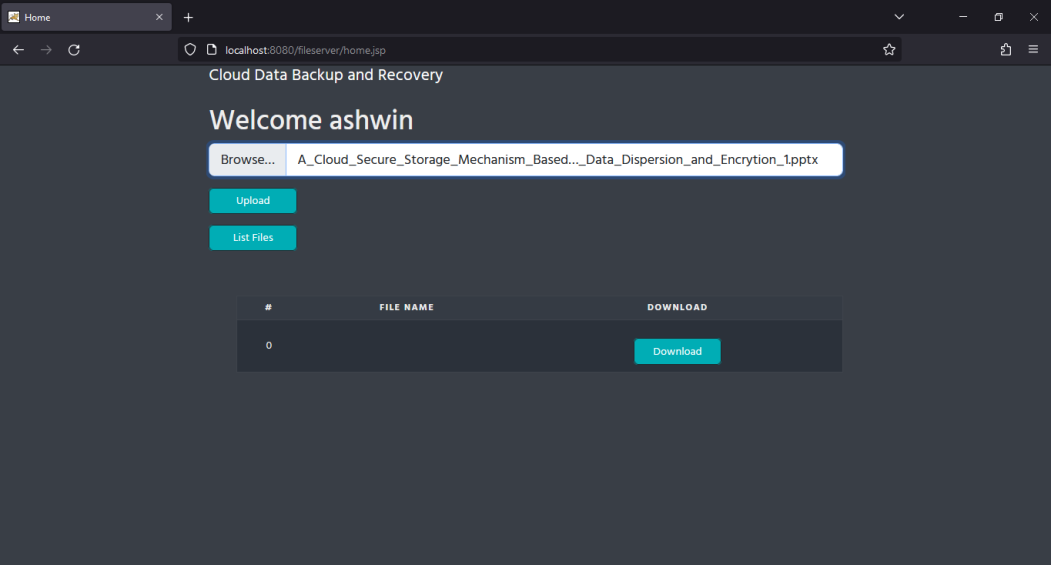
# SCREEN SHOTS

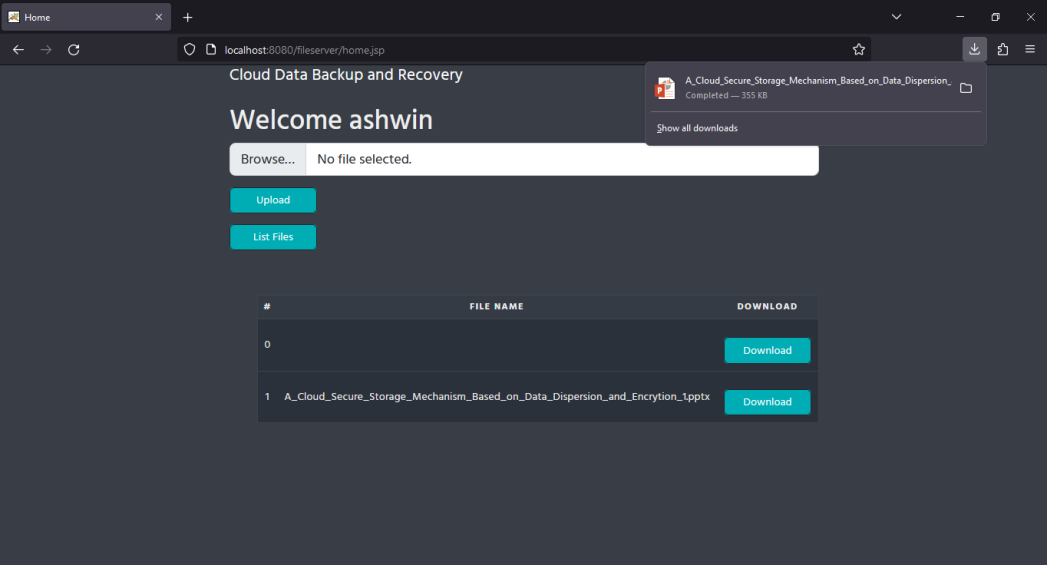
USER SIDE



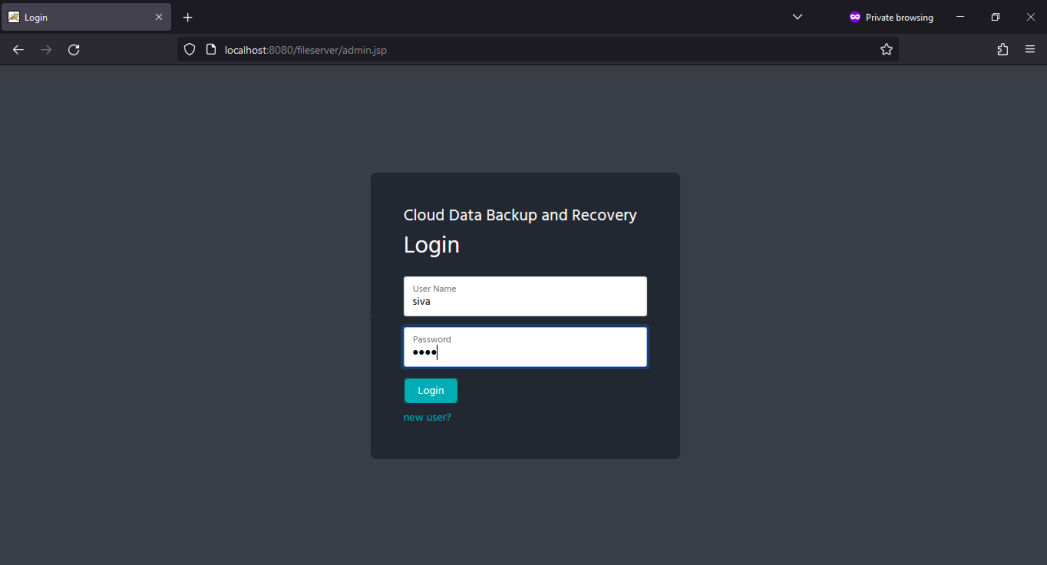


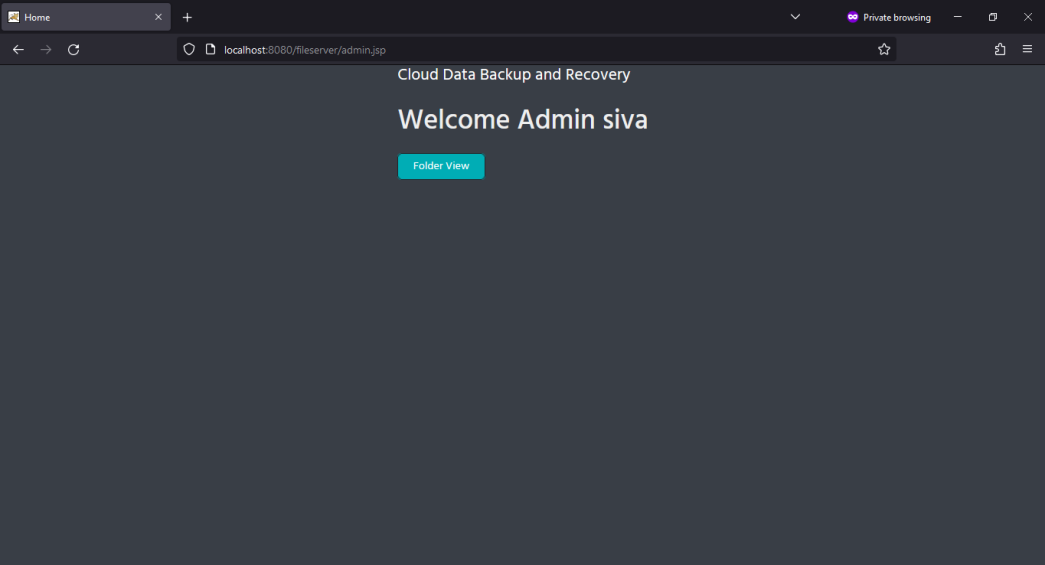


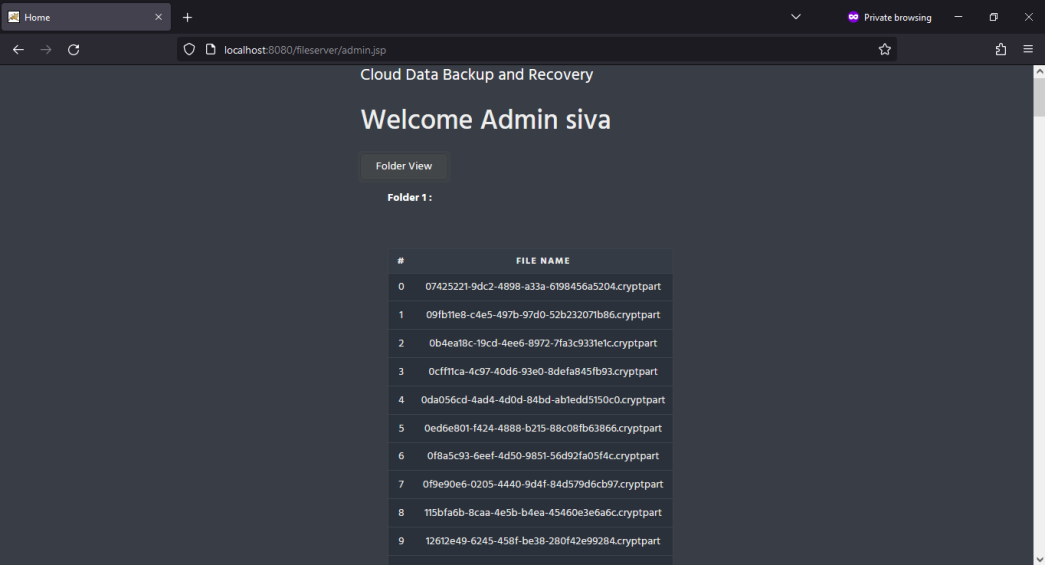




ADMIN SIDE







**REFERENCES**

1. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda ,``Capturing-the-invisible (CTI): Behavior-based attacks recognition in IoT-oriented industrial control systems,'' *IEEE Access*, vol. 8 , pp. 104956104966, 2020.
2. AlBelooshi, E. Damiani, K. Salah, and T. Martin, ‘‘Securing crypto- graphic keys in the cloud: A survey,’’ IEEE Cloud Computer., vol. 3, no. 4, pp.42–56, Jul. 2016.
3. Chun-I Fan “Dependable Data Outsourcing Scheme Based on Cloud-of-Clouds Approach with Fast Recovery”, IEEE Transactions on Cloud Computing Journals & Magazines’ 19 September 2018.
4. Heqing Song “A Cloud Secure Storage Mechanism Based On Data Dispersion And Encryption” IEEE Journal Article, 10 May 2021.
5. J. Li, Y. Zhang, X. Chen, and Y. Xiang, ``Secure attribute-based data sharing for resource-limited users in cloud computing,'' *Computer. Security.* , vol. 72, pp. 112, Jan. 2018.
6. J. Zhou, H. Duan, K. Liang, Q. Yan, F. Chen, F. R. Yu, J. Wu, and J. Chen,‘‘Securing outsourced data in the multi-authority cloud with fine-grained access control and efficient attribute revocation,’’ Computer. J., vol. 60, no. 8, pp. 1210–1222, Feb. 2017
7. M. Kumar, A. Rani, and S. Srivastava, ``Image forensics based on lighting estimation,'' *Int. J. Image Graph.*, vol. 19, no. 3, Jul. 2019 , Art. no. 1950014.
8. M. O. Rabin, ``Efcient dispersal of information for security, load balancing, and fault tolerance,'' *J. ACM*, vol. 36, no. 2, pp. 335348, Apr. 1989.[29] L. S. Jie and H. Y. Ping, ``A privacy-preserving integrity measurement P. Zerfos, H. Yeo, B. D. Paulovicks, and V. Sheinin, ``SDFS: Secure distributed le system for data-at-rest security for Hadoop-as-a-service, ‘in *Proc. IEEE Int. Conf. Big Data*, Santa Clara, CA, USA, Oct. 2015,pp. 12621271.
9. P. Li, T. Li, H. Ye, J. Li, X. Chen, and Y. Xiang, ‘‘Privacy-preserving machine learning with multiple data providers,’’ Future Gener. Computer. System ., vol. 87, pp. 341–350, Oct. 2018.
10. Reda Maher “DropStore: A Secure Backup System Using Multi-Cloud and Fog Computing’, IEEE Access Journals & Magazines , 10 May 2021.
11. Sultan Almakdi “An Efficient Secure System for Fetching Data From the Outsourced Encrypted Databases”, IEEE Access Journals & Magazines , 20 May 2021.
12. S. Y. Shah, B. Paulovicks, and P. Zerfos, ``Data-at-rest security for spark,''in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Washington DC, USA,Dec. 2016, pp. 14641473.
13. W. Wang, Z. Li, R. Owens, and B. K. Bhargava, ``Secure and effcient access to outsourced data,'' in *Proc. 1st ACM Cloud Comput. Secur.Workshop*, vol. 2009, Chicago, IL, USA, Nov. 2009, pp. 5566.
14. Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, ``DivORAM: Towards a practical oblivious RAM with variable block size,‘’*Inf. Sci.*, vol. 447,pp. 111, Jun. 2018

**COs, PO Mapping, PSO Mapping**

Course Code &Name : CS8811 Project Work

REGULATION: R2017 YEAR/SEM: IV/VIII

**COURSE OUTCOMES**

|  |  |
| --- | --- |
| CS8811.1 | Identify the problem by applying acquired knowledge. |
| CS8811.2 | Analyze and categorize executable project modules after considering risks. |
| CS8811.3 | Choose efficient tools for designing project modules. |
| CS8811.4 | Combine all the modules through effective team work after efficient testing. |
| CS8811.5 | Elaborate the completed task and compile the project report. |

**CORRELATION LEVELS**

|  |  |
| --- | --- |
| Substantial/ High | 3 |
| Moderate/ Medium | 2 |
| Slight/ Low | 1 |
| No correlation |  |

**CO – PSO CORRELATION LEVEL MATRIX**

|  |  |  |
| --- | --- | --- |
| COs | PSO1 | PSO2 |
| CS8811.1 | 3 | 1 |
| CS8811.2 |  | 2 |
| CS8811.3 |  | 3 |
| CS8811.4 |  | 3 |
| CS8811.5 |  |  |

**CO-PO CORRELATION LEVEL MATRIX**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COs | POs | | | | | | | | | | | |
| PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CS8811.1 | 3 | 3 |  | 1 |  |  | 1 | 2 | 3 |  |  |  |
| CS8811.2 |  | 3 | 2 | 3 |  | 3 | 1 |  | 3 | 3 |  | 2 |
| CS8811.3 |  |  | 3 | 3 | 3 |  |  | 3 | 3 | 3 | 1 | 2 |
| CS8811.4 |  |  |  |  |  |  | 3 | 3 | 3 | 3 | 2 | 3 |
| CS8811.5 |  |  |  |  |  |  |  |  | 3 | 3 |  | 3 |