

RAPID FAST PUBLIC AUDITS AND DYNAMIC DATA HANDLING IN CLOUD ENVIRONMENTS

A Project Report submitted for the partial fulfilment of the requirement for the
award of the degree of

MASTER OF COMPUTER APPLICATIONS

SUBMITTED BY

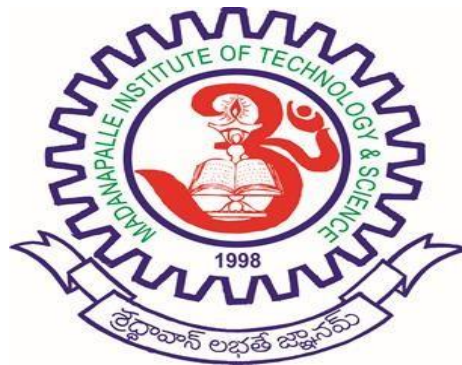
PASALA ANUJA
ROLL NO.: 23691F0009

Under the Guidance of

Dr. C. SIVARAJ

Associate Professor

Department of Computer Applications



DEPARTMENT OF COMPUTER APPLICATIONS
MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
UGC AUTONOMOUS

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

www.mits.ac.in

2024-2025

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapur)

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

This is to certify that the project work entitled “**RAPID FAST PUBLIC AUDITS AND DYNAMIC DATA HANDLING IN CLOUD ENVIRONMENTS**” is a bonafide work carried out by **PASALA ANUJA (Roll No.: 23691F0009)**, submitted in the partial fulfilment of the requirements for the award of the degree of Master of Computer Applications in Madanapalle Institute of Technology & Science, Madanapalle, affiliated to Jawaharlal Nehru Technological University Anantapur, Anantapuram the academic year 2024-2025.

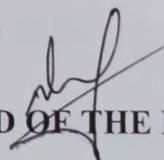


PROJECT GUIDE

Dr. C. SIVARAJ, Ph.D.,

Associate Professor

Department of Computer Applications



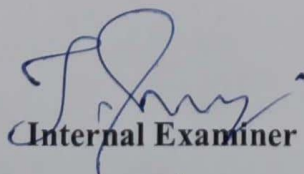
HEAD OF THE DEPARTMENT

Dr. N. NaveenKumar, M.C.A., Ph.D.,

Professor

Department of Computer Applications

Submitted for viva voce examination held on 07/07/2025



Internal Examiner

Date:

7/7/2025



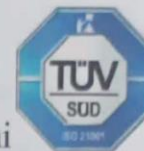
External Examiner

Date:

7/7/2025



MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC-AUTONOMOUS INSTITUTION)



Affiliated to JNTUA, Ananthapuramu & Approved by AICTE, New Delhi

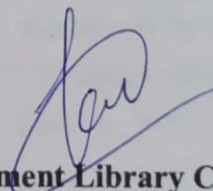
NAAC Accredited with A+ Grade, NIRF India Rankings 2021 - Band: 201-250 (Engg.)

NBA Accredited - B.Tech. (CIVIL, CSE, ECE, EEE, MECH), MBA & MCA

Plagiarism Verification Certificate

This is to certify that the MCA Major Project entitled as **“Rapid Fast Public Audits And Dynamic Data Handling In Cloud Environments”** submitted by **Ms. PASALA ANUJA (Reg.No.23691F0009)** has been evaluated using **Anti-Plagiarism** Software, **TURNITIN** and based on the analysis report generated by the software, the similarity index is found to be **19%**.

The following is the **TURNITIN** report for the Major Project consisting of **72** pages.


Department Library Coordinator

DECLARATION

I, PASALA ANUJA (Roll No.: 23691F0009) hereby declare that the project entitled “**RAPID FAST PUBLIC AUDITS AND DYNAMIC DATA HANDLING IN CLOUD ENVIRONMENTS**” is done by me under the guidance of **Dr.C. SIVARAJ, Ph.D.**, submitted in partial fulfilment of the requirements for the award of degree of Master of Computer Applications at **MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE**, Madanapalle affiliated to Jawaharlal Nehru Technological University Anantapur, Anantapuram during the academic year 2024-2025. This work has not been submitted by anybody towards the award of any degree.

Date: 30/06/2025
Place: Madanapalle

P. Anuja
Signature of the Student
(PASALA ANUJA)
(Roll No.: 23691F0009)

ACKNOWLEDGEMENT

First, I must thank the almighty who granted me knowledge, wisdom, strength and courage to serve in this world and to carry out my project work in a successfully way.

Next, I have to thank my parents who encouraged me to undergo this MCA course and do this project in a proper way.

I express my sincere thanks to **Dr. N. Vijaya Bhaskar Chowdary, Ph.D.**, Secretary & Correspondent, Madanapalle Institute of Technology & Science for his continuous encouragement to wards practical education and constant support in all aspects which includes the provision of very good infrastructure facilities in the institute.

It is my duty to thank our Principal, **Dr. C. Yuvaraj M.E, Ph.D.**, Madanapalle Institute of technology & science, for his guidance and support at the time of my course and project.

It is my duty to thank vice Principal Academics, **Dr. P. Ramanathan, Ph.D.**, for his continuous support though out our MCA career.

I express my sincere thanks to vice Principal PG programs **Dr. R. Kalpana, Ph.D.**, for her constant support and encouragement during the project.

It is my foremost duty to thank Head of the Department **Dr. N. Naveen Kumar, M.C.A, Ph.D.** who gave me constant support during the project time and continuous encouragement to words the completion of the project successfully.

I thank my faculty guide **Dr. C. Sivaraj, Ph.D.**, for his continuous support by conducting periodical reviews and guidance until the completion of the project in as successful manner.

PASALA ANUJA
(23691F0009)

ABSTRACT

Cloud storage offers flexibility and scalability by enabling customers to store their data remotely and access it at any time, from any location. However, users no longer have direct control over data after it is transferred to the cloud, which can result in risks including data loss, unauthorized alterations, or integrity breaches. To address these issues, a number of auditing techniques have been created to confirm the accuracy and security of data that has been saved.

A public auditing system designed for shared cloud data is introduced via an existing method, allowing several users to access data collectively while maintaining the confidentiality of their identities and facilitating traceability when necessary. This method still has security flaws even though it increases privacy and usability. Even after deleting the original data, it is still vulnerable to threats like tag forging and false proof generation, where an unreliable cloud provider could create phony verification findings. These issues highlight the need for a stronger and more secure auditing system.

This work offers a novel public auditing approach that effectively supports dynamic data management while enhancing security in order to get over these drawbacks. Our strategy guarantees that data integrity may be checked without disclosing user identities by utilizing cryptographic techniques. Additionally, it permits safe data operations, such as additions, removals, and updates, without interfering with the auditing process. This updated plan lowers the possibility of data loss or alteration while promoting safe and reliable public auditing in cloud environments.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO.
I	INTRODUCTION	1
1.1	About the project	1
II	SYSTEM ANALYSIS	2-5
2.1	Existing System	2
2.2	Proposed System	2
2.3	Hardware and Software Requirements	3
2.4	Feasibility Study	4-5
III	SYSTEM DESIGN	6-13
3.1	Module Description	6
3.2	ER Diagram	7-8
3.3	UML Diagrams	8-14
IV	SYSTEM IMPLEMENTATION	15-55
4.1	Language Selection	15-24
4.2	Screenshots	25-33
4.3	Sample Code	34-54
V	TESTING	55-59
5.1	Testing	55
5.2	Types of Tests	55-58
5.3	Test Strategy and Approach	59
VI	CONCLUSION AND FUTURE ENCHANCEMENTS	60-61
6.1	Conclusion	60
6.2	Future Enhancements	61
	REFERENCES	62-63

LIST OF FIGURES

S. N O.	FIGURE NO.	TITLE	PAGE NO.
1	3.2.1	ER Diagram	7
2	3.3.1	Use Case Diagram	10
3	3.3.2	Class Diagram	11
4	3.3.3	Sequence Diagram	12
5	3.3.4	Activity Diagram	13
8	3.3.5	Deployment Diagram	14
9	4.1.1	Working of java program	16
10	4.1.2	Implementation of JVM	16
11	4.1.3	Program running on the java platform	17
12	4.1.4	Structure of cloud computing	17
13	4.1.5	Structure of service models	18
14	4.2.1	Home Page	25
15	4.2.2	Login page of Data User	26
16	4.2.3	Welcome page of Data User	26
17	4.2.4	Login page of Data Owner	27
18	4.2.5	Welcome page of Data Owner	28
19	4.2.6	Data Owner Encrypt the data and upload the data	28
20	4.2.7	Data Owner can delete the files	29
21	4.2.8	Welcome page of CSP	30
22	4.2.9	View all files in cloud	31
23	4.2.10	View all files with decrypted permission	31
24	4.2.11	Generating the Secret Key	32

25	4.2.12	Decrypt the file for downloading	32
26	4.2.13	Downloading the file	33

LIST OF ABBREVAITIONS

S.NO.	ABBREVIATED TERM	EXPANSION
1	UML	Unified Modeling Language
2	ER Diagram	Entity Relationship Diagram

CHAPTER-I

INTRODUCTION

1.1 About The Project

With the increasing reliance users can save and retrieve their data on cloud storage services at any time and from any location. However, this convenience comes with security risks, particularly the loss of direct control over stored data, which may lead to unauthorized modifications or deletions. Ensuring the integrity and availability of outsourced data has therefore become a critical concern, prompting extensive research into cloud storage auditing techniques.

To overcome these obstacles, a number of public auditing programs have been put out that let outside auditors confirm data accuracy without jeopardizing user privacy. One such scheme claims to support key security features, including data privacy, identity traceability, and group dynamics. However, upon closer examination, we identify vulnerabilities in this scheme, specifically its susceptibility to tag forgery and proof forgery attacks. These weaknesses allow a malicious cloud server to generate valid proofs even if it has deleted part of the stored data, thereby deceiving auditors.

CHAPTER-II

SYSTEM ANALYSIS

2.1 Existing System

Current setup cannot be used directly in a shared multi-owner configuration because it is made to support unshared multi-owner settings. The current system lacks server data that allows for multi-keyword and multi-owner searches. When multiple data users have the same subset of characteristics, it becomes challenging to detect hostile users who leak the secret keys.

When businesses and people commit their data to cloud storage providers, data privacy is a major problem. By encrypting data at the packet level, secure network coding techniques provide increased confidentiality and guarantee that private data is safe even in the event that it is intercepted during storage or transit.

Disadvantages

- single Owner Sharing
- It takes high time.
- Complexity.
- Uncontrollable.

2.2 Proposed System

Furthermore, for that we introduce a multiparty access control system for ciphertext that enables data co-owners to modify the ciphertext in accordance with their own references to privacy. To address issue of privacy distributes brought on by disparate access regulations, three policy aggregation solutions are offered: majority permission, owner priority, and complete permit.

Advantages

- More Effective
- Data Cleaning and Visualization
- Scalability
- Less time

2.3 Hardware and Software Requirements

2.3.1 Hardware Requirements

- Processor : Intel Core2Duo
- RAM : 2GB (min)
- Hard Disk : 180GB
- Seed : 2.4GHz

2.3.2 Software Requirements

- Operating System : Windows 11
- Application Server : Tomcat 5.0
- Front End : HTML, Java, Jsp
- Scripts : JavaScript
- Server side Script : Java Server Pages
- Database : MySQL 5.0
- Database Connectivity : JDBC

2.4 Feasibility Study

Whether the system will be advantageous to the association and the design viability are the primary topics of discussion. The primary goal of the feasibility study is to assess technical, functional, financial viability implementing new modules and fixing outdated handling systems. If there are infinite funds and time, then any system can be implemented. The primary discussion's feasibility research section contains certain

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

2.4.1 Economic Feasibility:

The suggested project is very cost-effective because it makes use of open-source technology and doesn't call for pricey hardware or proprietary software. The open-source nature of tools like Apache Tomcat, MySQL, and Java drastically lowers development expenses. Because the application is web-based, it doesn't require sophisticated infrastructure to function well on simple systems. Additionally, by enhancing cloud security, the technology reduces the possibility of data breaches, which can assist businesses in preventing possible financial losses. It also reduces operational costs and the requirement for manual supervision by automating auditing procedures. In conclusion, this project offers a safe yet affordable solution, making it useful and economical for cloud customers of various sizes.

2.4.2 Technical Feasibility:

Technically, the suggested system can be implemented with easily accessible tools and technology. It is created with open-source, extensively supported MySQL for data storage, Apache Tomcat for the web server, and Java for application logic. These technologies are well-known for being cross-platform compatible, stable, and simple to use. No specialized or expensive hardware is needed for this project. Because it can function well on common computer systems, development and deployment are simple and reasonably priced. Furthermore, the system is built to provide safe data exchange, encryption. All things considered, the tools and techniques employed are appropriate for the project's needs, making

it technically feasible and simple to execute with few significant difficulties.

2.4.3 Social Feasibility:

The system proves to be effective in real-world scenarios and offers significant social value. Its user-friendly design ensures that even individuals without technical expertise can understand and operate it with ease. The simple interface and intuitive features allow users to interact with the system confidently and comfortably. Since it directly addresses key user concerns such as data privacy and secure cloud access, its adoption is expected to face minimal resistance. With basic guidance and support, users can quickly adapt to the system and make full use of its capabilities. By protecting both personal and organizational data, the system builds trust and encourages user acceptance.

CHAPTER-III

SYSTEM DESIGN

3.1. Module Description:

- **Data Owner**

The module allows the owner of the data to upload their information to the cloud server. For protection purposes, the owner of the data encrypts the information before putting it in the cloud.

- **CSP**

The main discussion looks at design viability and if the system will benefit the association and cloud can monitor all the actions

- **Private key generator**

According to the user's request, the module's private key generator will create a private key for them, which they can use to retrieve data.

- **Data disseminator**

When someone wants to share their data with others, such as their friends, family, or coworkers, they are known as a data disseminator. Since the data owner can decrypt the original ciphertexts, the data disseminator must be one of the intended receivers listed for security and access control purposes. To distribute the data owner's information to third parties, the data distributor can create reencryption keys and utilize them to make re-encryption requests for data sent to the CSP. Only when the data disseminator's characteristics meet access policy requirements and the pre-established period has passed may CSP successfully perform a data re-encryption request.

- **Data Consumer/End User**

Only if the user is authorized to get the data information in this module can they use encrypted key to access it. Malicious users may conspire to obtain sensitive files that are beyond their rights because users could attempt in order to view the data files inside access advantages. He is requesting data access from the data disseminator and private key generator; if they grant permission, the user can download the encrypted data.

3.2 ER Diagram

The ER model for short, maps out a database structure with an image known as an ER Diagram. You can picture the model as a developer's initial sketch that the team looks over before they start to build the actual database. The core parts of the ER picture are sets of entities and sets of relationships that tie them together. Inside the diagram, lines link the groups of entities, making clear how they interact.

Each entity set collects similar items, and those items carry their own features called attributes. From the database manager's view, an entity roughly equals a table, while the table's properties are its columns. or even columns nested inside it. By mapping tables and their links this way, the diagram reveals the databases logical shape before any software is built. To make things clearer, imagine a simple ER sketch that shows, say, customers linked to orders it quickly shows how each customer can place many orders.

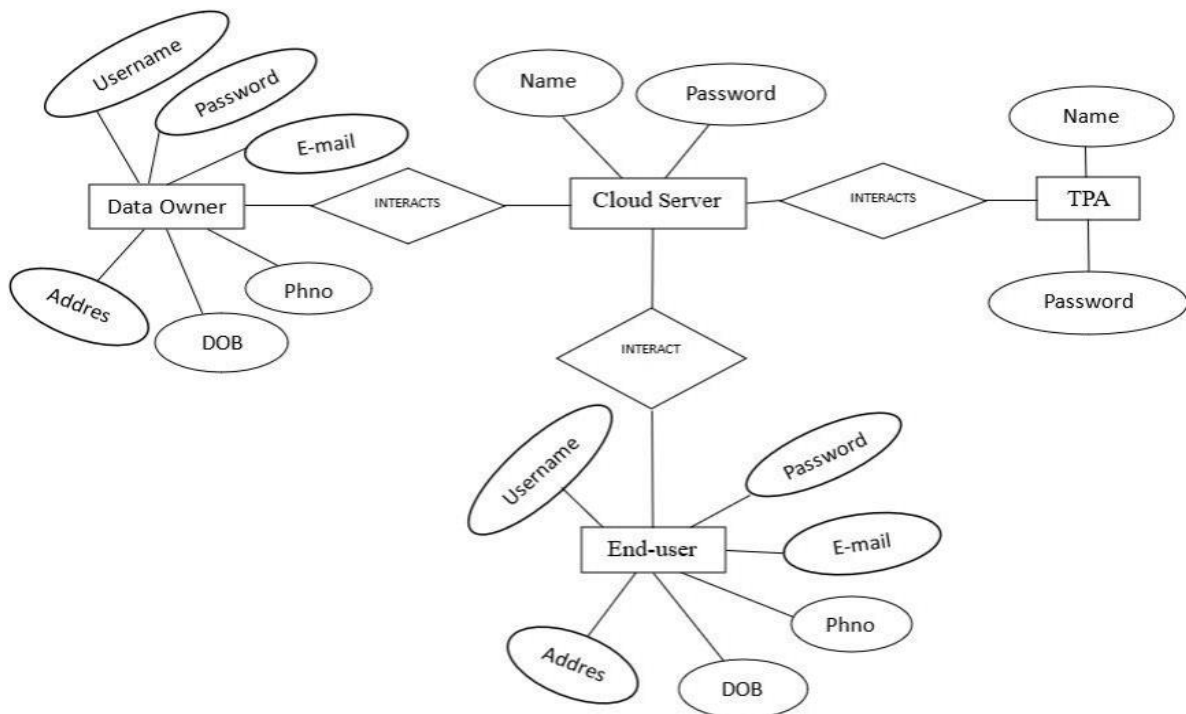


Figure 3.2.1: ER Diagram

Description: The Figure 3.2.1 illustrates the interaction between Data Owner, Cloud Server, End-user, and TPA (Third Party Auditor), each with their own authentication details and personal attributes. It highlights how these entities securely communicate and manage data access within a cloud environment.

3.3 UML Diagrams

UML stands for Unified Modelling Language. UML is a common resource for everyone who works with object-oriented software. It provides a way for engineers and designers to communicate the intentions for the work they wish to build before writing code. The OMG created UML to reduce the confusion and miscommunication when working with the models of object-oriented software.

Key Components of UML

UML is composed of two essential parts:

- **Meta-Model**

The meta-model defines the abstract structure and rules that govern UML. It outlines how UML elements are organized and how they relate to one another.

- **Notation**

This represents the visual symbols and diagrams used to express models. These graphical elements help in conveying system architecture, workflows, and object interactions effectively.

In the future, the framework may evolve to include formal methods or lightweight processes that would give the user even more control to develop and validate their models. At this time, users can utilize UML for defining, showing, building, and documenting all aspects of a software system; the techniques are also directly applicable to business processes and hardware design.

3.3.1 Use Case Diagram:

This method focuses on the people or outside systems—referred to as actors that communicate with the program. It also draws attention to the particular steps known as use cases—that the system takes in reaction to those exchanges.

Purpose of a Use Case Diagram

Use case diagrams are useful because they:

- Show who will be using the system
- Explain what the system will do for each user
- Show how different actions are connected to each other

These diagrams are most helpful during the early planning stages of a project. They give a clear idea of what the system is expected to do and help define its boundaries and functions.

Why Use Case Diagrams Matter

- They define what the system should do based on user needs.
- They improve understanding between team members and clients.
- They help identify who does what in the system.
- They show how tasks are linked together.
- They help prioritize features during development.
- They represent how the system behaves when someone uses it.

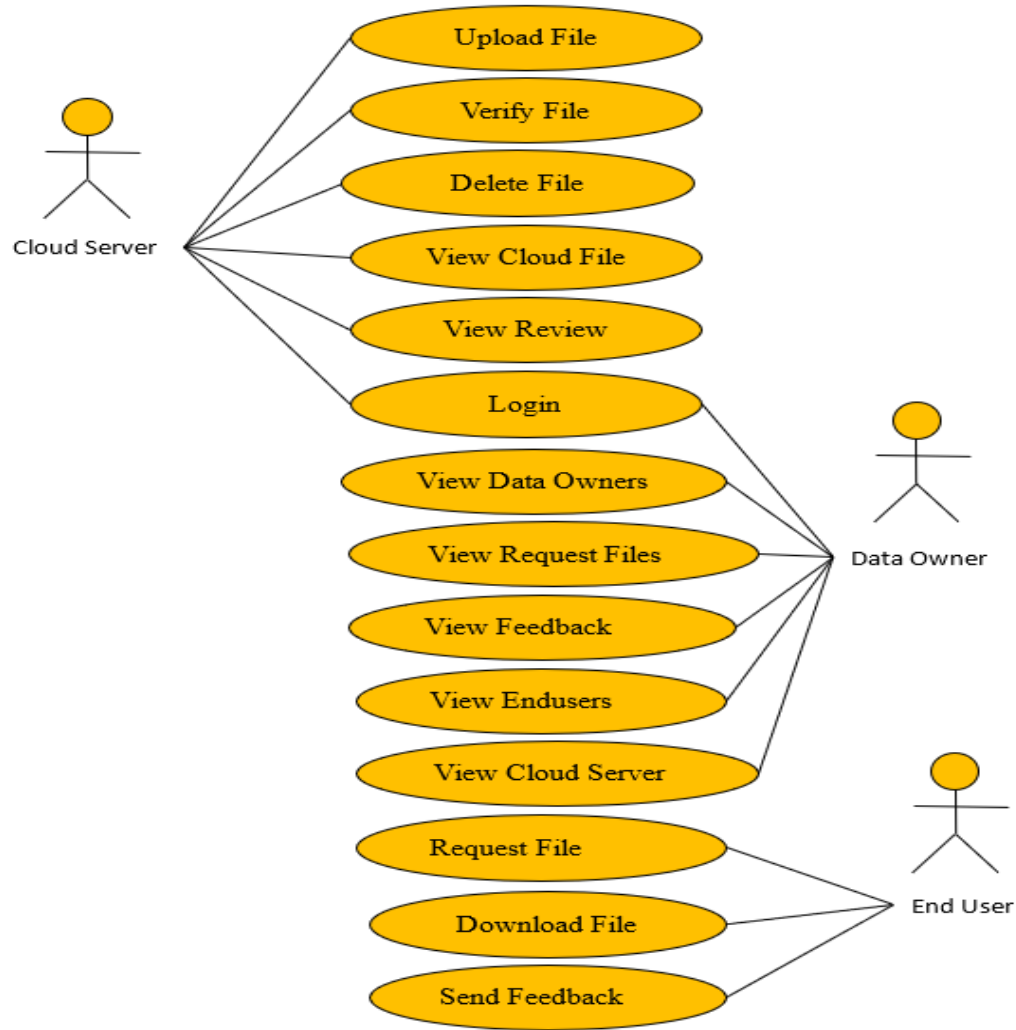


Figure 3.3.1: Use Case Diagram

Description: Figure 3.3.1 represents the interactions between Cloud Server, Data Owner, and End User, highlighting various file-related operations like upload, verify, request, download, and feedback. Each actor performs specific tasks to manage, access, and review cloud-stored data securely.

3.3.2 Class Diagram:

A class diagram is a type of visual representation used in software development to show the structure of a system. It highlights the system's classes, the data they hold (attributes), the operations they perform (methods), and the connections or relationships between different classes.

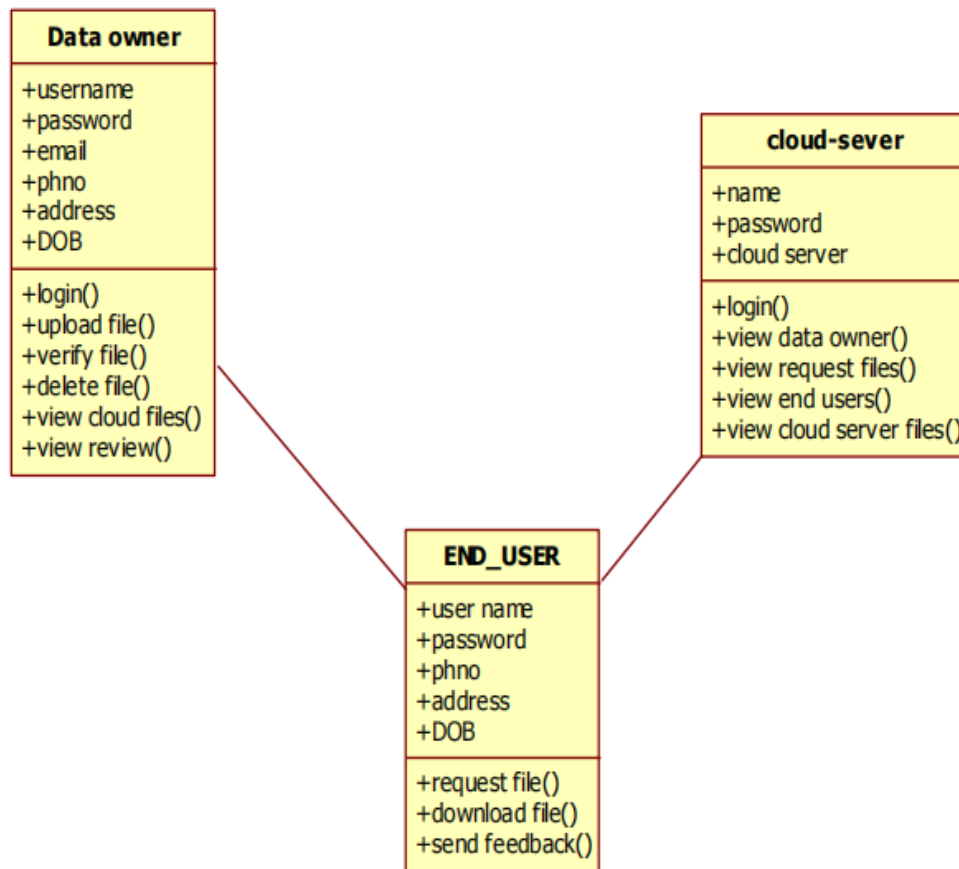


Figure 3.3.2: Class Diagram

Description: The Figure 3.3.2 illustrates the roles of Data Owner, End User, and Cloud Server, along with their attributes and associated operations. It shows how each entity performs specific tasks like file handling, feedback, and data management within the cloud system.

3.3.3 Sequence Diagram:

In Unified Modelling Language (UML), a sequence diagram is an interaction diagram that illustrates how different parts of a system communicate by exchanging messages in a specific order. It visually represents the flow of events between objects over time. These diagrams are sometimes also referred to as message sequence charts, timing diagrams, or event scenarios.

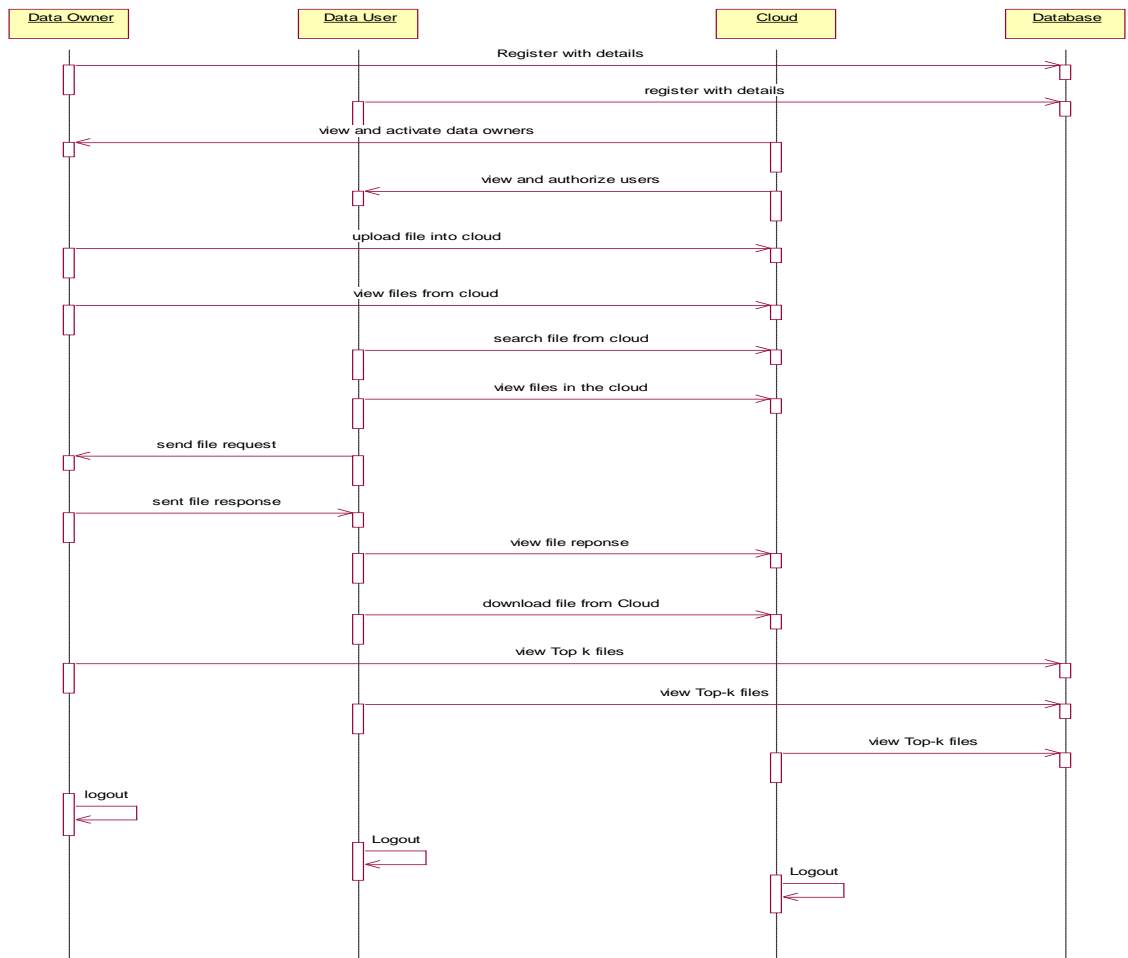


Figure 3.3.3: Sequence Diagram

Description: The Figure 3.3.3 shows the interactions between Data Owner, Data User, Cloud, and Database for activities like registration, file upload/download, and viewing top-k files. It outlines the step-by-step communication flow for file sharing and access in a secure cloud environment.

3.3.4 Activity Diagram:

In UML, activity diagrams are used to visually represent the flow of actions within a system, supporting features like decision-making, looping, and parallel processing. They help in detailing how different parts of a business process or system component operate. While activity diagrams are similar to flowcharts in showing the sequence of actions, they also allow for displaying multiple paths and actions happening at the same time. These diagrams use nodes to represent activities and edges to show the flow of control and data between them.

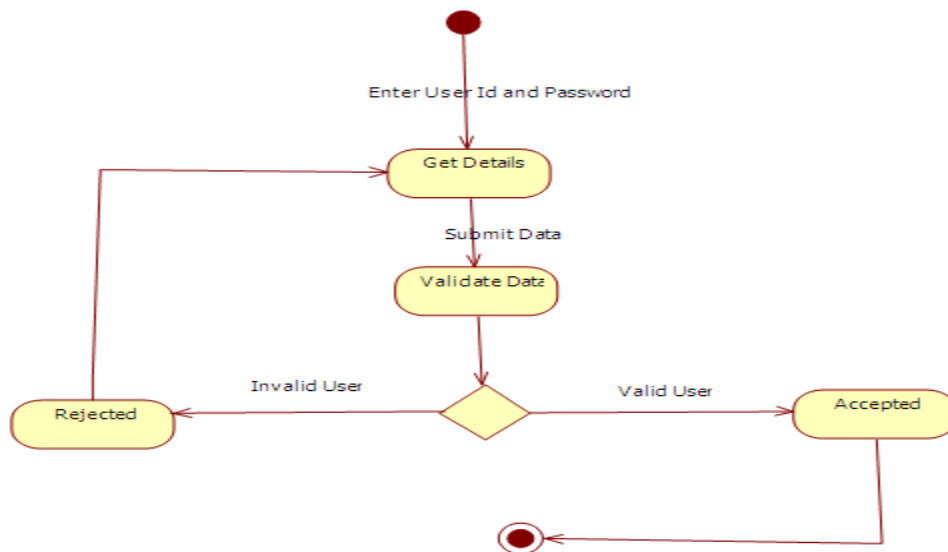


Figure 3.3.4: Activity Diagram

Description: The Figure 3.3.4 illustrates the login process where a user enters credentials, which are then validated. Based on the validation, access is either accepted for valid users or rejected for invalid ones.

3.3.5 Deployment Diagram:

In UML, a deployment diagram is used to represent the physical setup of a system. It shows the hardware elements, known as nodes (like web servers, application servers, and database servers), and the software, known as artifacts (such as web applications and databases), that run on those nodes. It also displays how these parts are connected, using communication methods like JDBC or REST. Deployment diagrams are useful for understanding how software components are physically distributed and how they interact with the system's hardware.



Figure 3.3.5: Deployment Diagram

Description: Based on the given figure 3.3.5, it shows the interaction between two nodes: user and admin. It represents how software components are deployed across these two physical or virtual devices, indicating communication between the user system and the admin system.

CHAPTER-IV

SYSTEM IMPLEMENTATION

4.1 Language Selection

Introduction to Java

High-level programming languages like Java are renowned for their strength and adaptability. Because of its simple and beginner-friendly design, developers can write and comprehend code with ease. Java's platform independence—the ability of Java programs to operate on several computers without requiring changes—is one of its main advantages. Additionally, it is ideally suited for distributed computing, which enables cross-network applications.

Java utilizes an interpreter to run programs efficiently and offers good performance. Java also facilitates multithreading, which allows applications to manage several tasks at once and enhances performance. The language can easily adjust to changes while being executed because it is designed to be safe, reliable, and flexible. Java stands out from most programming languages because it uses both compilation and interpretation to run programs. When you write a Java program, it is first compiled into an intermediate form called **bytecode**. This bytecode is not tied to any specific machine and can be run on any system that has a **Java Virtual Machine (JVM)**.

The compilation step happens only once, turning your source code into bytecode. After that, every time you run the program, the JVM interprets the bytecode and executes it on your device. This unique process allows Java to be platform-independent. Since bytecode is designed for the JVM, any system with a compatible JVM—whether it's Windows, macOS, Linux, or another platform—can run the same Java program. This is what enables Java's key principle: **“write once, run anywhere.”**

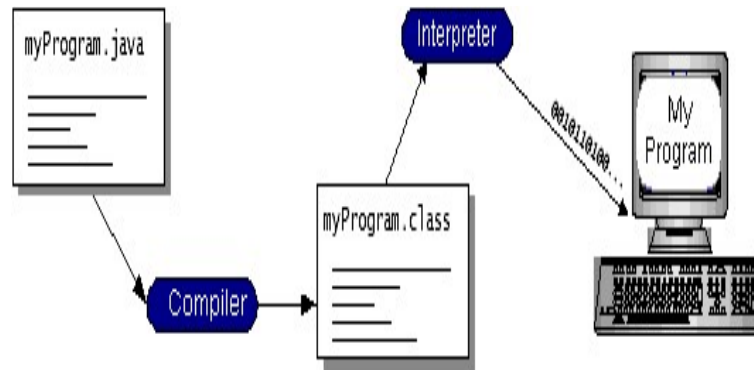


Figure 4.1.1: Working of Java Program



Figure 4.1.2: Implementation of Java Virtual Machine

The Java Platform:

A platform is the environment where software runs, typically combining an operating system and hardware. Common examples include Windows, Linux, and macOS.

Java is different because it offers a **software-only platform** that operates above any existing hardware-based system. This means Java programs aren't tied to one specific device or operating system.

The Java platform has two key parts:

- **Java Virtual Machine (JVM):** This component runs Java programs. It reads and executes Java bytecode, making sure the same program can work across different devices with a compatible JVM.
- **Java Application Programming Interface (Java API):** A collection of prebuilt classes and

tools that simplify application development in Java.

Thanks to this setup, Java follows the principle of “**write once, run anywhere,**” allowing developers to create programs that can run on any device with a Java environment.

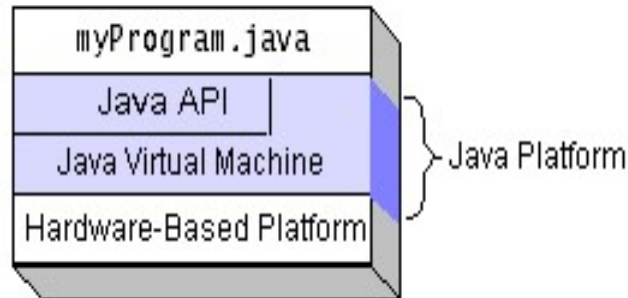


Figure 4.1.3: Program Running on the Java Platform

Technology used in this project:

➤ What is Cloud Computing?

Cloud computing is the practice of accessing computing services—such as storage, processing power, and software—over a network, most commonly the Internet. Instead of relying on local computers, users depend on remote servers to manage data, run applications, and perform tasks. These services are provided by third-party companies and are designed to be easily accessible online. Cloud computing allows individuals and organizations to use advanced tools and powerful server networks without having to maintain their own infrastructure.

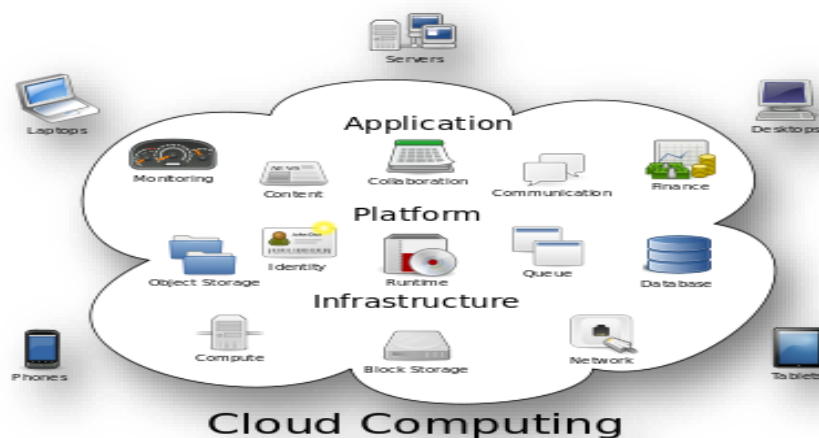


Figure 4.1.4: Structure of cloud computing

➤ Services Models:

The **end-user layer** represents the user's perspective when interacting with cloud services and complements the three core service models. This layer helps describe how users experience and utilize cloud-based solutions.

For example, if someone uses services at the **infrastructure level**, they gain access to computing resources like servers and storage. However, they are responsible for managing their applications, including maintenance, security, and support. In contrast, when using services at the **application level**, the cloud provider takes care of those responsibilities, allowing the user to simply use the application without worrying about the backend management.

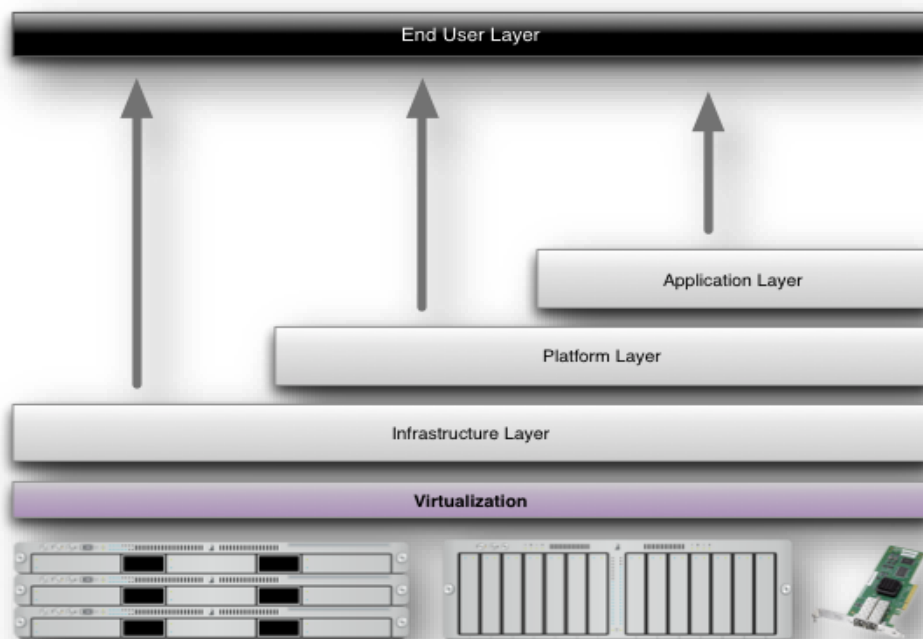


Figure 4.1.5: Structure of service models

Benefits of cloud computing:

- **Cost-Effective:** You are charged only for the computing resources you use, which helps save money by avoiding upfront hardware investments.
- **Improved Security:** Each user's cloud environment is isolated from others, offering better protection and reducing the risk of unauthorized access.
- **Strong Performance:** Cloud platforms give users access to powerful computing resources, ensuring smooth and efficient operation of applications.
- **Scalability:** Cloud systems can automatically adjust by adding or removing resources based on the current demand, making it easy to handle workload changes.
- **High Availability:** Cloud services use multiple servers to ensure reliability. If one server goes down, the system can quickly switch to another, minimizing interruptions.
- **Remote Control and Flexibility:** Users can access their systems from anywhere. The cloud also allows for easy deployment of customized setups using saved configurations and tools.

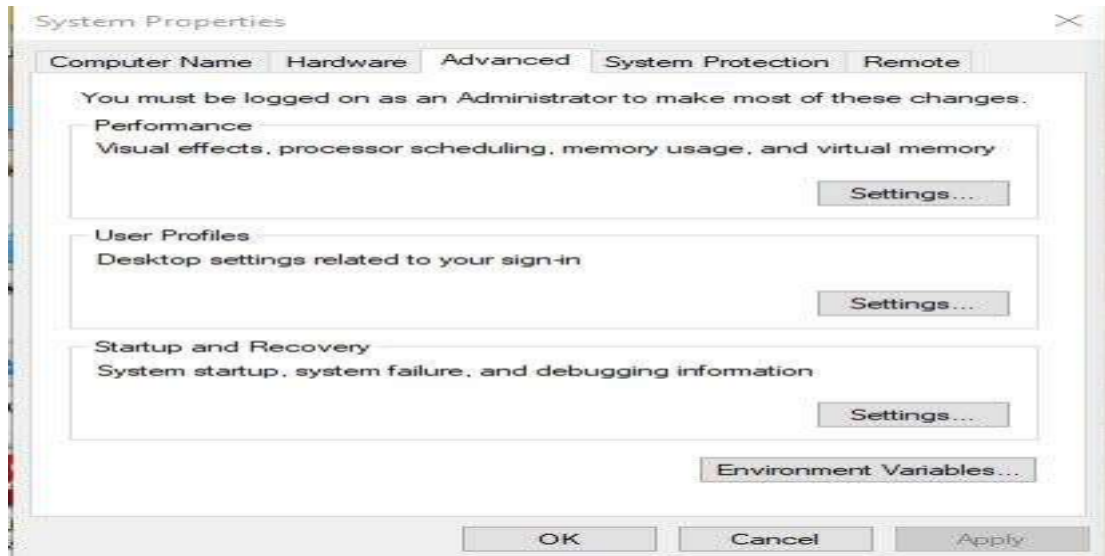
Working Model and installation procedure:

Step1: Installation of java

- Go to <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Click on JDK DOWNLOAD button. run the exe file and then follow the instruction given in wizard.

To set up the path

- Right click on my pc and then go to my properties



Screen 4.1.6: Properties wizard

Now check the latest and correct version for JDK installation to the instructions of latest wizard.

Step1.2: Path setting for java

- Go to advanced settings and then click on environment variables
- Create a class path and copy the path of the java folder where it is located in program files

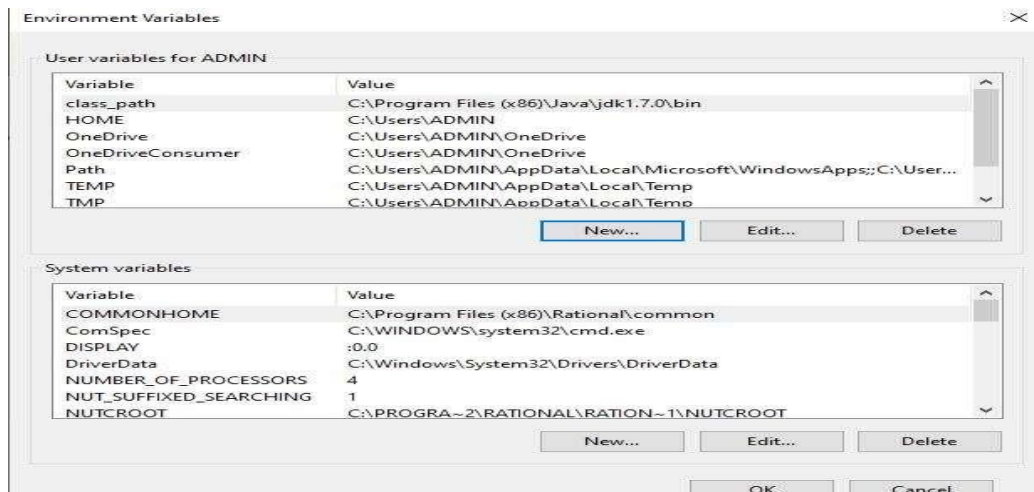


Figure 4.1.7: Path setting for java

Select the class _ path which is suitable to the project then click new button and click ok button.

Installation and setup of Apache Tomcat:

Step2: Go to <http://tomcat.apache.org/index.html> and click on download latest versions. Run the exe file and click on next and follow the wizard instructions.



Figure 4.1.8: Welcome page of Tomcat

Click on install with port number 8090 with username and password as aits and aits. Mention the connection port as 8090 and then click on next and finally click on Finish.

Step2.1: Enter the port number correctly and click to the next button.

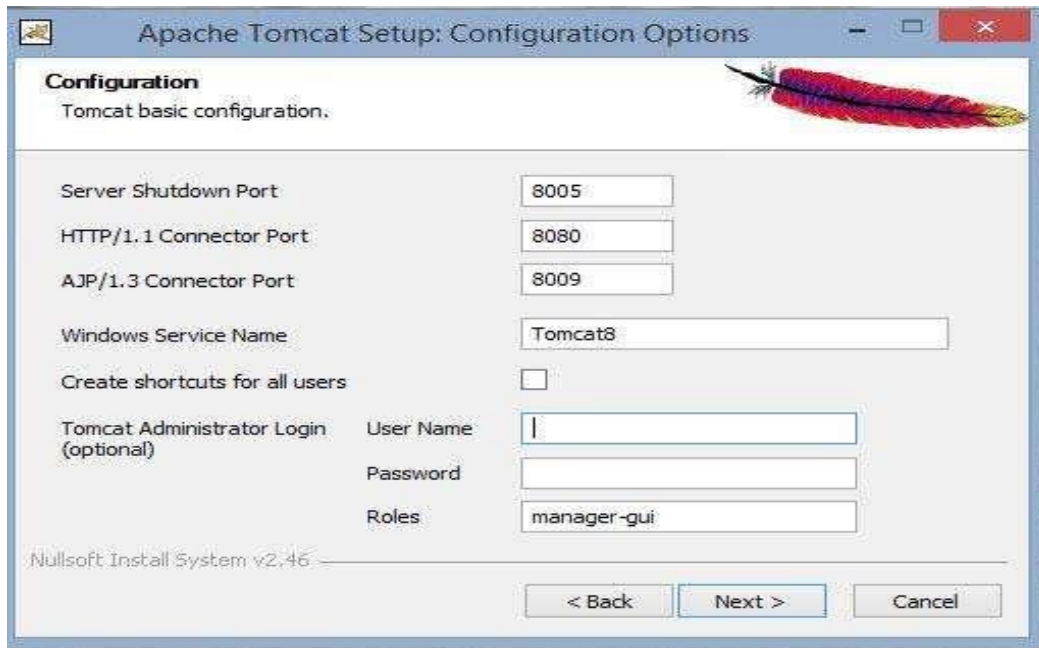


Figure 4.1.9: Tomcat Configuration Options Page

Description: Click on I agree button in. license agreement in order to accept the terms and condition.

Step2.2: Agree the Tomcat license agreement

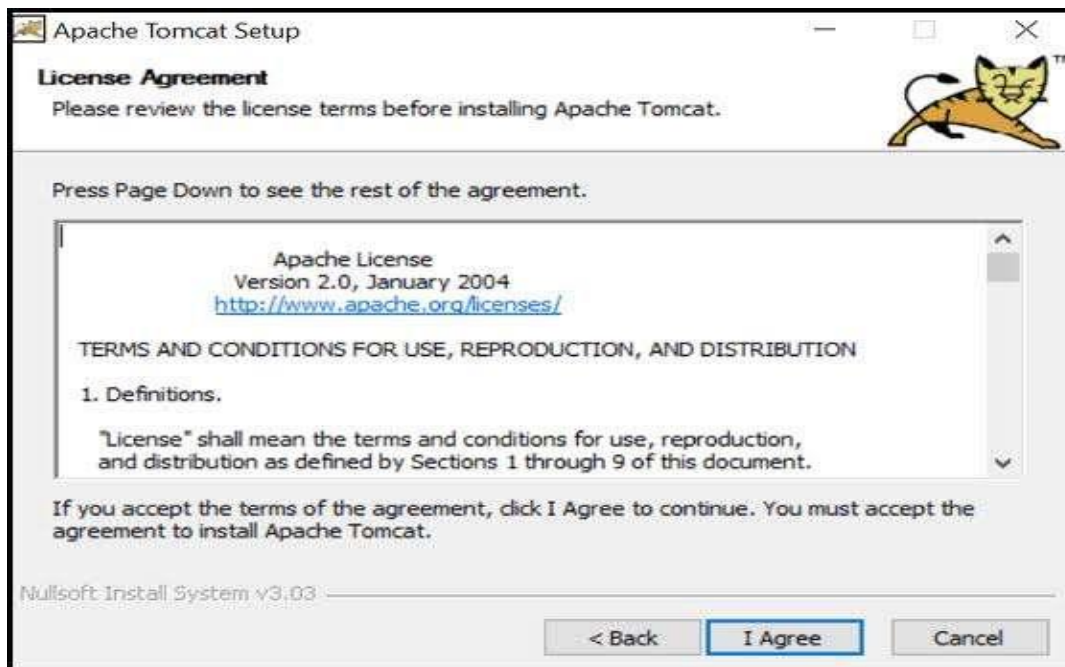


Figure 4.1.10: Tomcat license agreement

Description: check the license and agree the terms and conditions for go to the next Level.

Step3: Installation and setup of MYSQL

Go to <http://dev.mywql.com/downloads/> . and click on install button. After completion of installation, click on exe file and then click on next.

Run the MySQL setup and click on next and follow the instruction in wizard.



Figure 4.1.11: Welcome wizard of MYSQL

Step3.1: Conform the type as typical and then click on next and follow the Instructions.

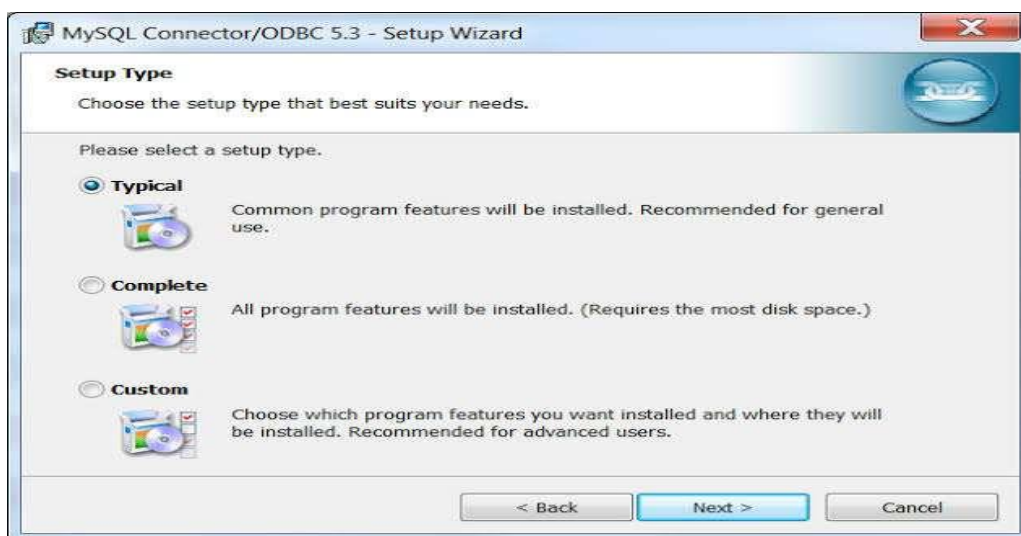


Figure 4.1.12: SQL setup Wizard

Description: Now confirm the password as root in system settings field and then click on finish.

Step 3.1: Enter the password correctly and click to the next button

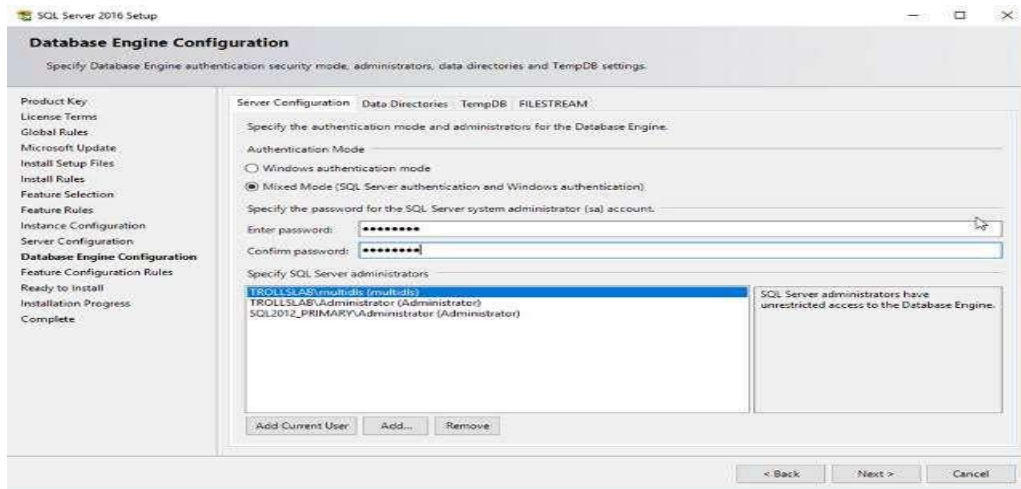


Figure 4.1.13: Database Configuration Engine

Description: Submit the data successfully in the database engine configuration

4.2 SCREEN SHOTS

All the screen shots of the project are given below which shows the execution for better understanding of the project.

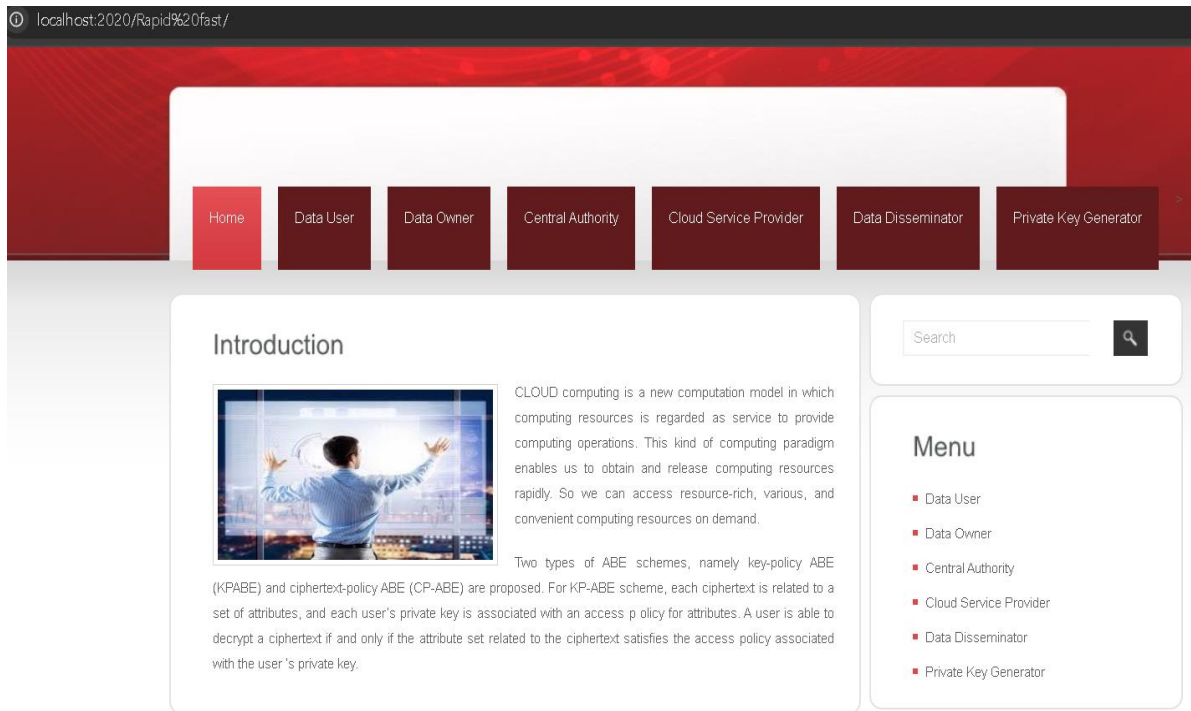


Figure 4.2.1 Home Page

Description: The webpage interface in the figure 4.2.1 shown is part of a cloud-based application featuring roles like Data User, Data Owner, and Central Authority. It introduces cloud computing concepts and ABE (Attribute-Based Encryption) schemes for secure and efficient data access control.

Figure 4.2.2: Login Page of Data User

Description: The figure 4.2.2 shows the **Data User Login** page of a cloud security system, where users can enter their credentials to access services. It also provides options for new user registration and credential reset.

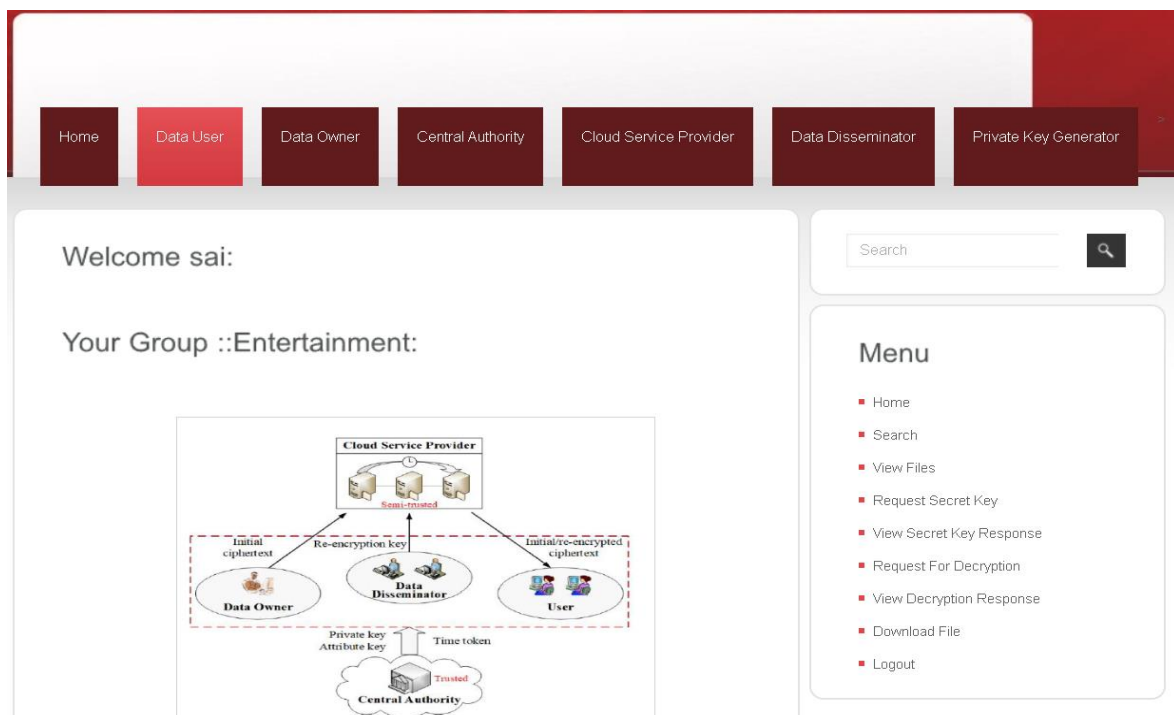


Figure 4.2.3: Welcome page of Data User

Description: The figure 4.2.3 shows a web-based cloud data management system with role-based access. The interface includes navigation tabs for various entities (like Data User, Data Owner, Central Authority) and a menu for key and file operations like search, decryption, and downloads.

The screenshot displays the 'Data Owner Login' page. At the top, there is a navigation bar with seven tabs: 'Home', 'Data User', 'Data Owner' (which is highlighted in red), 'Central Authority', 'Cloud Service Provider', 'Data Disseminator', and 'Private Key Generator'. Below the navigation bar, the main content area is divided into two sections. The left section, titled 'Data Owner Login :', contains two input fields labeled 'Data Owner Name (required)' and 'Data Owner Password (required)'. Below these fields are 'Login' and 'Reset' buttons. At the bottom of this section, there is a link that says 'New Data Owner? click here to [Register](#)' and a red '[Back](#)' link. The right section, titled 'Menu', contains a list of roles with red square bullet points: 'Data User', 'Data Owner', 'Central Authority', 'Cloud Service Provider', 'Data Disseminator', and 'Private Key Generator'. Above the 'Menu' section is a search bar with the placeholder text 'Search' and a magnifying glass icon.

Figure 4.2.4: Login Page of Data Owner

Description: The figure 4.2.4 shows a web-based login interface for a "Data Owner" within a cloud data management system. It includes navigation tabs for various roles such as Data User, Central Authority, and Private Key Generator, along with options to log in or register as a new data owner.

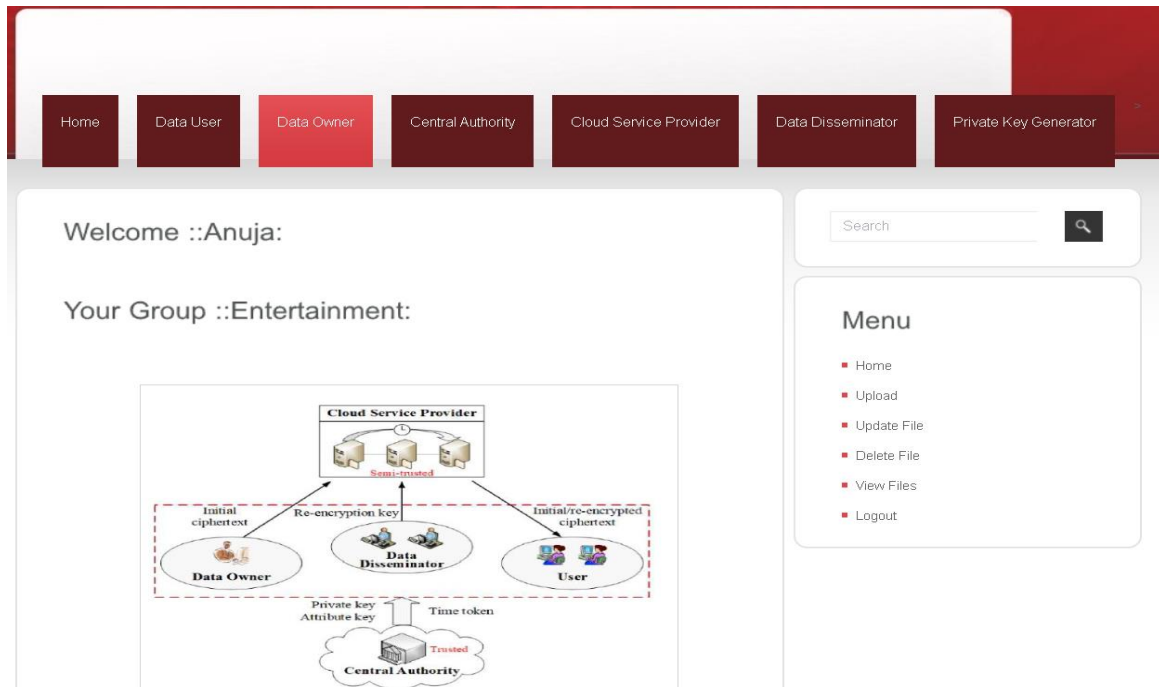


Figure 4.2.5: Welcome page of Data Owner

Description: The figure 4.2.5 shows a cloud data management portal where the user logged in as a Data Owner. The interface includes modules like Data User, Central Authority, and Cloud Service Provider, with options to upload, update, delete, and view files.

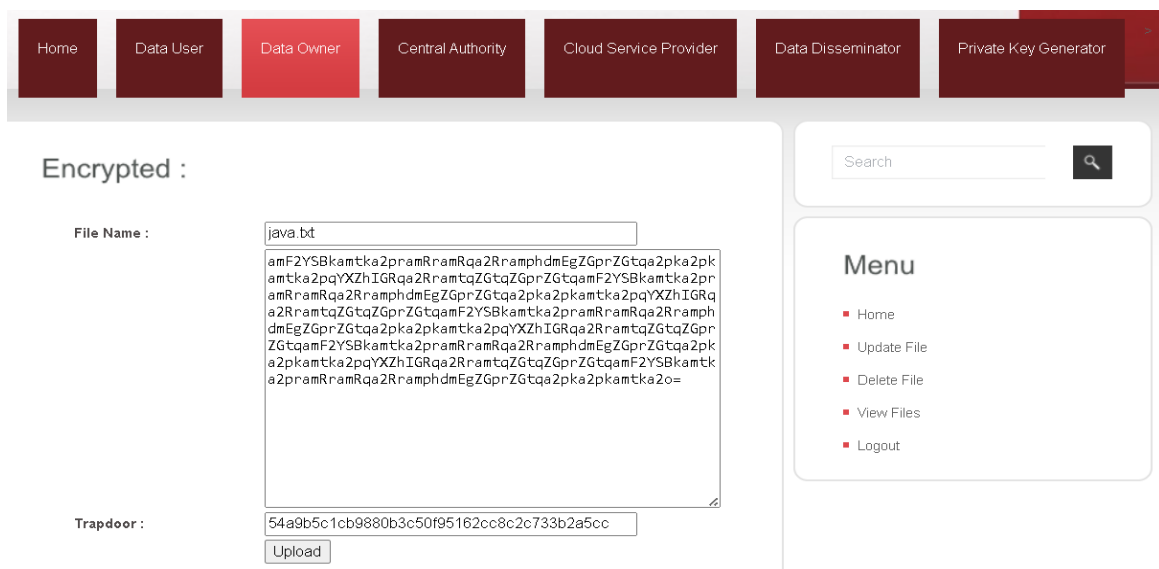


Figure 4.2.6: Data Owner Encrypt the data and upload the data

Description: The figure 4.2.6 shows a web interface for a cloud data management system where the "Data Owner" uploads an encrypted file (java.txt) along with a trapdoor key. The interface includes navigation tabs for various roles and file management options like update, delete, and view.

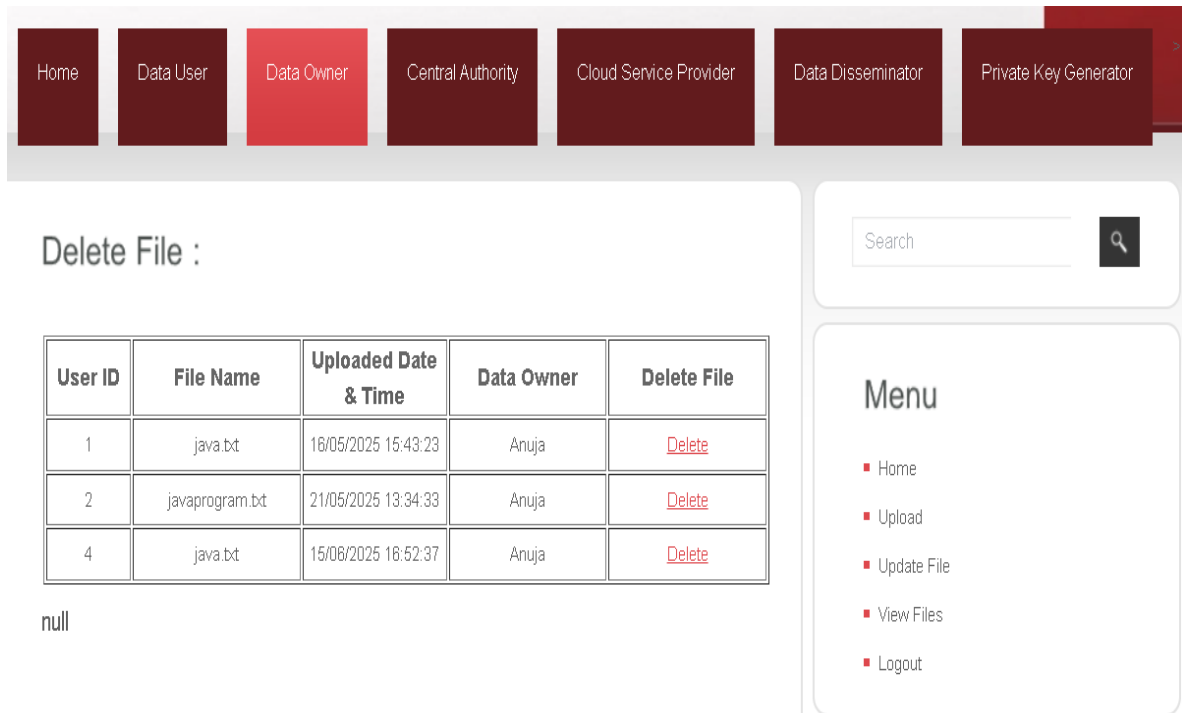


Figure 4.2.7: Data Owner can delete the files

Description: The figure 4.2.7 shows a "Data Owner" dashboard interface for deleting files in a cloud system. It lists uploaded files with details like User ID, File Name, Upload Date & Time, and provides a "Delete" option for each.

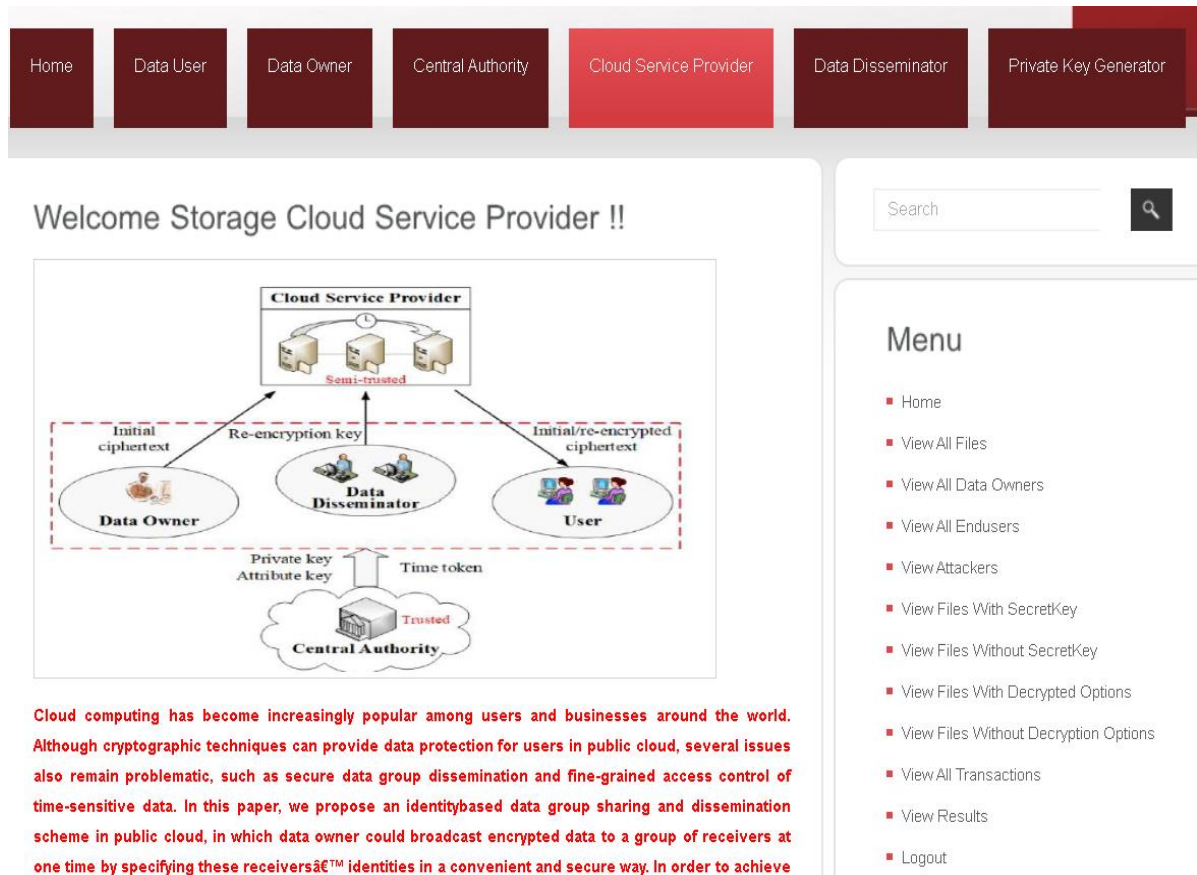


Figure 4.2.8: Welcome page of CSP

Description: The figure 4.2.8 shows a cloud-based data sharing system interface for a storage service provider, highlighting roles like Data Owner, User, Central Authority, and Data Disseminator. It illustrates a secure data dissemination process using re-encryption and attribute-based access in a semi-trusted cloud environment.

Home
Data User
Data Owner
Central Authority
Cloud Service Provider
Data Disseminator
Private Key Generator

View All Files !!

User ID	File Name	Data Owner	Date & Time
1	java.bt	Anuja	18/05/2025 15:43:23
2	javaprogram.bt	Anuja	21/05/2025 13:34:33
3	Elections	Ranjith	11/08/2025 21:15:35
4	java.bt	Anuja	15/08/2025 18:52:37

Back

Search

Menu

- Home
- View Attackers
- View Files With SecretKey
- View Files Without SecretKey
- View Files With Decrypted Options
- View Files Without Decryption Options
- View All Transactions
- Logout

Figure 4.2.9: View all files in the cloud

Description: The figure 4.2.9 shows a "Cloud Service Provider" interface displaying a list of uploaded files with details like user ID, file name, data owner, and upload date & time. The right-side menu offers navigation options such as viewing files with or without keys, viewing attackers, and logging out.

Home
Data User
Data Owner
Central Authority
Cloud Service Provider
Data Disseminator
Private Key Generator

View All Files With Decrypted Permission !!

ID	User Name	Owner Name	File Name	Decrypt per
1	sai	Anuja	java.txt	Permitted
2	sai	Anuja	javaprogram.txt	Permitted

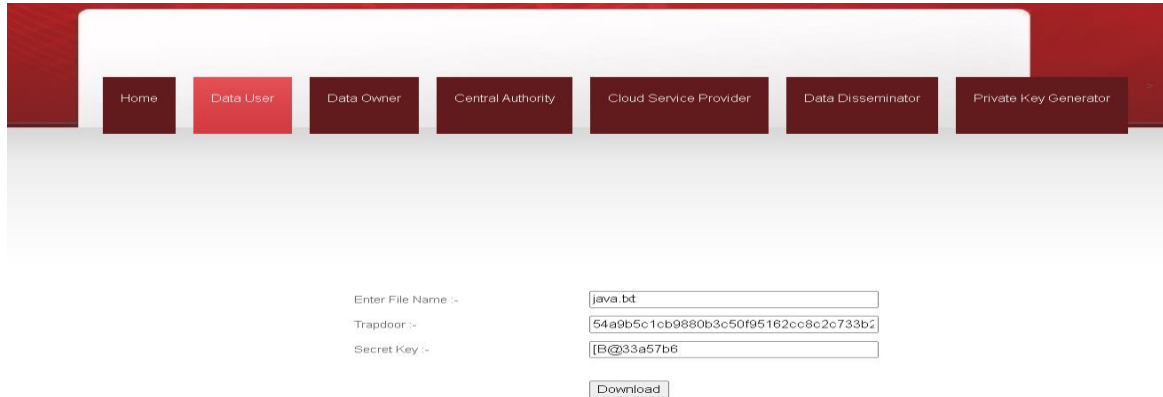
Search

Menu

- Home
- View All Files Without Decrypt Permission
- Provide Decrypt Permission
- Logout

Figure 4.2.10: View all files with Decrypted Permission

Description: This figure 4.2.12 illustrates the "Decrypt the File for Downloading" interface, where the Data User can view the decrypted contents of a file before downloading. It features a text area displaying the file content, a download button, and a side menu with related actions like requesting a secret key and viewing decryption responses.



The screenshot shows a web interface with a red header bar containing a navigation menu with the following items: Home, Data User (highlighted in red), Data Owner, Central Authority, Cloud Service Provider, Data Disseminator, and Private Key Generator. Below the header, the main content area is white. It contains a form with three input fields and a button. The first field is labeled 'Enter File Name :-' and contains the text 'java.txt'. The second field is labeled 'Trapdoor :-' and contains a long alphanumeric string: '54a9b5c1cb9880b3c50f95162cc8c2c733b2'. The third field is labeled 'Secret Key :-' and contains the text '[B@33a57b6'. Below these fields is a button labeled 'Download'.

Figure 4.2.13: Download the file

Description: The figure 4.2.13 displays the "Data User" interface where the user inputs a file name to retrieve its associated Trapdoor and Secret Key. It includes fields for entering the file name, viewing the generated trapdoor, and downloading the secret key.

4.3. SAMPLE CODE

Sample code has been given to show how the system has been implemented. The methodology used to create the screens and the sample java script code are given here.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
    <%@page import="java.util.*"%>
    <%@ include file="connect.jsp"%>
    <%@page import
    ="java.util.*, java.security.Key, java.util.Random, javax.crypto.Cipher, javax.crypto.spec.SecretKeySpec, org.bouncycastle.util.encoders.Base64"%>
    <%@ page
    import="java.sql.*, java.util.Random, java.io.PrintStream, java.io.FileOutputStream, java.io.FileInputStream, java.security.DigestInputStream, java.math.BigInteger, java.security.MessageDigest, java.io.BufferedReader, java.io.InputStream"%>
    <%@ page import
    ="java.security.Key, java.security.KeyPair, java.security.KeyPairGenerator, javax.crypto.Cipher"%>
    <%@page import
    ="java.util.*, java.text.SimpleDateFormat, java.util.Date, java.io.FileInputStream, java.io.FileOutputStream, java.io.PrintStream"%>
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <title>CLOUD SERVICE PROVIDERS </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link href="css/style.css" rel="stylesheet" type="text/css" />
    <link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
    <script type="text/javascript" src="js/cufon-yui.js"></script>
    <script type="text/javascript" src="js/cufon-titillium-250.js"></script>
    <script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
    <script type="text/javascript" src="js/script.js"></script>
```

```

<style type="text/css">
<!--
.style32 {
    font-size: 12px;
    font-weight: bold;
    color: #FF0000;
}
.style33 {font-size: 12px; color: #6d6d6d; }
.style34 {
    font-size: 24px;
    font-weight: bold;
}
.style35 {
    color: #FF0000;
    font-weight: bold;
}
.style36 {font-weight: bold}
-->

</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html" class="style34">A Distributed Trust Evaluation Protocol<br />
with Privacy Protection for Intercloud</a></h1>
</div>
<div class="menu_nav">
<ul>
<li><a href="C_Main.jsp">CLOUD SERVICE PROVIDERS</a></li>

```

```

</li></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"><a href="#"></a><a href="#"></a><a
href="#"></a></div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2> View File Details !!!</h2>
<p>&nbsp;</p>
<%
String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8="",s9="",s10,s11,s12,s13;
int i=0,j=0,k=0;
int uid = Integer.parseInt(request.getParameter("usid"));
try
{
String query="Select *from cloudserver where id='"+uid+"'";
Statement st=connection.createStatement();
ResultSet rs=st.executeQuery(query);
while ( rs.next() )
| {
i=rs.getInt(1);

```

```

s2=rs.getString(2);
s3=rs.getString(3);
s4=rs.getString(4);
s5=rs.getString(5);
s6=rs.getString(6);
s7=rs.getString(7);
s8=rs.getString(8);
s9=rs.getString(9);
}

```

```

%>

```

```

<table width="549" border="1" style="border-collapse:collapse" cellpadding="0" cellspacing="0"
align="center">
<tr>
<td width="142" height="36" bgcolor="#FFFF00"><div align="center" class="style29 style32">
<div align="left">ID :</div>
</div></td>
<td width="394"><label>
<input readonly="readonly" name="textfield" type="text" value="<%=i%>" size="49" />
</label></td>
</tr>
<tr>
<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">
<div align="left">Index : </div>
</div></td>
<td><input readonly="readonly" name="textfield22" type="text" size="49" value="<%=s3%>"
/></td>
</tr>
<tr>

```

```

<td height="186" bgcolor="#FFFF00"><div align="center" class="style32">
<div align="left">Contents :</div>
</div></td>
<td><label>
<textarea readonly="readonly" rows="10" cols="50" name="textarea"><%=s4%></textarea>
</label></td>
</tr>
<tr>
<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">
<div align="left">MAC :</div>
</div></td>
<td><input readonly="readonly" name="textfield3" type="text" value="<%=s5%>" size="49"
/></td>
</tr>
<tr>
<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">
<div align="left">Rank :</div>
</div></td>
<td><input readonly="readonly" name="textfield32" type="text" value="<%=s7%>" size="49"
/></td>
</tr>
<tr>
<td height="36" bgcolor="#FFFF00"><div align="center" class="style32">
<div align="left">Date & Time :</div>
</div></td>
<td><input readonly="readonly" name="textfield33" type="text" value="<%=s8%>" size="49"
/></td>
</tr>
<tr>
<td height="36" bgcolor="#FFFF00"><div align="center" class="style33">

```

```

<div align="left" class="style35">USERS: </div>
</div></td>
<td><input readonly="readonly" name="textfield34" type="text" value="<%=s9%>" size="49"
/></td>
<%

connection.close();

    }

    catch(Exception e)
    {
        out.println(e.getMessage());
    }
%>
</tr>
</table>
<p align="right"><a href="C_ViewFiles.jsp">Back</a></p>
</div>
</div>
<div class="sidebar">
<div class="gadget">
<h2 class="star">Menu</h2>
<div class="clr"></div>
<ul class="sb_menu style36">

<li><a href="C_Main.jsp">Home</a></li>
<li><a href="C_Login.jsp">Logout</a></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>

```



```

</div>
<div class="footer">
<div class="footer_resize">
<p class="lf">&nbsp;</p>
<p class="rt">&nbsp;</p>
<div style="clear:both;"></div>
</div>
</div>
</div>
<div align=center></div>
</body>
</html>

```

```

<%@page import="java.util.*"%>

```

```

<%@ include file="connect.jsp"%>

```

```

<%@page

```

```

    import="java.util.*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.
    SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>

```

```

<%@ page

```

```

    import="java.sql.*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.File
    InputStream,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.i
    o.BufferedInputStream"%>

```

```

<%@ page

```

```

    import="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.
    Cipher"%>

```

```

<%@page

```

```

    import="java.util.*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.
    FileOutputStream,java.io.PrintStream"%>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Attack</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-aller.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {font-size: 24px}
.style2 {color: #0000FF}
.style3 {color: #5E6D4E}
.style4 {font-weight: bold}
-->
</style>

</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">|
<div class="logo">
<h1><a href="index.html" class="style1">An Efficient Multi-User Searchable Encryption Scheme
without Query Transformation Over Outsourced Encrypted Data
</a></h1>
</div>
<div class="menu_nav">
<ul>

```

```

<li class="active"><a href="index.html"><span>Home</span> Page </a></li>
<li><a href="DataOwnerLogin.html">Data Owner </a></li>
<li><a href="EndUserLogin.html">End User</a></li>
<li><a href="CloudServerLogin.html">Cloud Server</a>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"><a href="#"></a><a href="#"></a><a
href="#"></a></div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2><span>WELCOME TO HOME PAGE </span></h2>
<p class="infopost"><a href="#" class="com"><span>11</span></a></p>
<div class="clr"></div>
<div class="clr"></div>
</div>
<div class="article">
<h2><span></span><form action="Attack1.jsp" method="post" name="form1" id="form1">
<table width="616" border="0" align="left">
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>

```

```

</tr>
<tr>
  <td width="333"><span class="style1">Enter File Name :-</span></td>
  <td width="273"><label><input required name="t1"
    type="text" size="40" /></label></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td><span class="style1">Ur Name :-</span></td>
  <td><input name="ot" type="text" size="40" /></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td>
    <div align="right"><input type="submit" name="Submit"
      value="Attack" /></div>
  </td>

```

```

        </tr>
    </table>
</form></h2>
<p class="infopost">&nbsp;</p>
<div class="clr"></div>
<div class="clr"></div>
</div>
</div>
<div class="sidebar">
<div class="searchform">
<form id="formsearch" name="formsearch" method="post" action="#">
<span>
<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80"
value="Search our ste:" type="text" />
</span>
<input name="button_search" src="images/search.gif" class="button_search" type="image" />
</form>
</div>
<div class="clr"></div>

<div class="gadget">
<h2 class="star"><span>Home</span> Menu</h2>
<div class="clr"></div>
<ul class="sb_menu">
<li><a href="index.html">Home Page </a></li>
<li><a href="DataOwnerLogin.html">Data Owner </a></li>
<li><a href="EndUserLogin.html">End User</a></li>
<li><a href="CloudServerLogin.html">Cloud Server</a></li>
</ul>
</div>
<div class="gadget">

```

```

<h2 class="star">&nbsp;</h2>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg">
<div class="fbg_resize">
<div class="clr"></div>
</div>
</div>
<div class="footer">
<div class="footer_resize">
<div style="clear:both;"></div>
</div>
</div>
</div>
</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@page import="java.util.*"%>
<%@page import="java.sql.*"%>
<%@page import
="java.util.*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>
<%@ page
import="java.sql.*,java.util.Random,java.io.PrintStream,java.io.FileOutputStream,java.io.FileInputStream,java.security.DigestInputStream,java.math.BigInteger,java.security.MessageDigest,java.io.BufferedInputStream"%>
<%@ page import

```

```

="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>
<%@page import
="java.util.*,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutputStream
eam,java.io.PrintStream"%>
<%@ include file="connect.jsp"%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CLOUD SERVICE PROVIDERS </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-titillium-250.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style21 {font-size: 14px}
.style24 {color: #FF0000}
.style28 {font-size: 24px}
.style29 {font-size: 14px; color: #FF0000; font-weight: bold; }
-->
</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">
<div class="logo">
<h1><a href="index.html" class="style28">A Distributed Trust Evaluation Protocol<br />

```

```

with Privacy Protection for Intercloud</a></h1>
</div>
<div class="menu_nav">
<ul>
<li><a href="C_Main.jsp">CLOUD SERVICE PROVIDERS</a></li>
<li><a href="C_Login.jsp"><span>Logout </span></a></li>
<li></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"><a href="#"></a><a href="#"></a><a
href="#"></a></div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2><span>Grant Search Control </span></h2>
<p>&nbsp;</p>
<table width="565" border="2" cellpadding="0" style="border-collapse:collapse" cellspacing="0"
align="center">
<tr>
<td width="99" height="30" bgcolor="#FFFF00"><div align="center" class="style21
style24"><strong> ID </strong></div></td>
<td width="174" bgcolor="#FFFF00"><div align="center" class="style29">User Name </div></td>

```



```
<td width="252" bgcolor="#FFFF00"><div align="center" class="style29">Permission</div></td>
<%
```

```
|
```

```
String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8,s9="",s10,s11,s12,s13;
int i=0,j=0,k=0;
```

```
try
{
    String query="select * from request";
    Statement st=connection.createStatement();
    ResultSet rs=st.executeQuery(query);
    while ( rs.next() )
    {
        i=rs.getInt(1);
        s2=rs.getString(2);
        s3=rs.getString(3);

        %>
```

```
</tr>
```

```
<tr>
```

```
<td height="28"><div align="center" class="style24"><%=i%></div></td>
```

```
<td><div align="center" class="style24"><a
```

```
href="C_UserSGDetails.jsp?uname=<%=s2%>"><%=s2%></a></div></td>
```

```
<%
```

```
if(s3.equalsIgnoreCase("Requested"))
{
```

```
%>
```

```
<td><div class="style24">
```

```
<div align="center"><a href="C_GrantSearch1.jsp?usid=<%=i%>"><%=s3%></a></div>
```

```

<li><strong><a href="C_Main.jsp">Home</a></strong></li>
<li><strong><a href="C_Login.jsp">Logout</a></strong></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg"></div>
<div class="footer">
<div class="footer_resize">
<p class="lf">&nbsp;</p>
<p class="rf">&nbsp;</p>
<div style="clear:both;"></div>
</div>
</div>
</div>
<div align=center></div>
</body>
</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@page import="java.util.*"%>
<%@page import="java.sql.*"%>
<%@page import
="java.util.* ,java.security.Key ,java.util.Random ,javax.crypto.Cipher ,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>
<%@ page
import="java.sql.* ,java.util.Random ,java.io.PrintStream ,java.io.FileOutputStream ,java.io.FileInputStream
ream ,java.security.DigestInputStream ,java.math.BigInteger ,java.security.MessageDigest ,java.io.Buff
eredInputStream"%>

```

```

<%@ page import
="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>
<%@page import
="java.util.* ,java.text.SimpleDateFormat,java.util.Date,java.io.FileInputStream,java.io.FileOutputStr
eam,java.io.PrintStream"%>
<%@ include file="connect.jsp"%>

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CLOUD SERVICE PROVIDERS </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-titillium-250.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--

</style>
</head>
<body>
<div class="main">
<div class="header">
<div class="header_resize">

```

```

<div class="logo">
<h1><a href="index.html" class="style28">A Distributed Trust Evaluation Protocol<br />
with Privacy Protection for Intercloud</a></h1>
</div>
<div class="menu_nav">
<ul>
<li><a href="C_Main.jsp">CLOUD SERVICE PROVIDERS</a></li>
<li><a href="C_Login.jsp"><span>Logout </span></a></li>
<li></li>
</ul>
</div>
<div class="clr"></div>
<div class="slider">
<div id="coin-slider"><a href="#"></a><a href="#"></a><a
href="#"></a></div>
<div class="clr"></div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="content">
<div class="content_resize">
<div class="mainbar">
<div class="article">
<h2><span>Grant Search Control </span></h2>
<p>&nbsp;</p>
<table width="565" border="2" cellpadding="0" style="border-collapse:collapse" cellspacing="0"
align="center">
<tr>
<td width="99" height="30" bgcolor="#FFFF00"><div align="center" class="style21

```

```

style24"><strong> ID </strong></div></td>
<td width="174" bgcolor="#FFFF00"><div align="center" class="style29">User Name </div></td>
<td width="252" bgcolor="#FFFF00"><div align="center" class="style29">Permission</div></td>
<%

```

```

String s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8,s9="",s10,s11,s12,s13;
int i=0,j=0,k=0;

```

```

try
{
    String query="select * from request";
    Statement st=connection.createStatement();
    ResultSet rs=st.executeQuery(query);
    while ( rs.next() )
    {
        i=rs.getInt(1);
        s2=rs.getString(2);
        s3=rs.getString(3);

        %>

```

```

</tr>

```

```

<tr>

```

```

<td height="28"><div align="center" class="style24"><%=i%></div></td>
<td><div align="center" class="style24"><a
href="C_UserSGDetails.jsp?uname=<%=s2%>"><%=s2%></a></div></td>
<%

```

```

        if(s3.equalsIgnoreCase("Requested"))
        {

```

```

        %>

```

```

<td><div class="style24">
<div align="center"><a href="C_GrantSearch1.jsp?usid=<%=i%>"><%=s3%></a></div>
</div></td>
<%

                                }else
                                {
                                %>
<td width="28"><div class="style24">
<div align="center"><%=s3%></div>
</div></td>
</tr>
<%

                                }

                                }
connection.close();
    }

catch(Exception e)
    {
    out.println(e.getMessage());
    }
%>
</table>
<p>&nbsp;</p>
<p align="right"><a href="C_Main.jsp">Back</a></p>
</div>
</div>
<div class="sidebar">
<div class="gadget">
<h2 class="star">Menu</h2>

```

```

<div class="clr"></div>
<ul class="sb_menu">
<li><strong><a href="C_Main.jsp">Home</a></strong></li>
<li><strong><a href="C_Login.jsp">Logout</a></strong></li>
</ul>
</div>
</div>
<div class="clr"></div>
</div>
</div>
<div class="fbg"></div>
<div class="footer">
<div class="footer_resize">
<p class="lf">&nbsp;</p>
<p class="rf">&nbsp;</p>
<div style="clear:both;"></div>
</div>
</div>
</div>
<div align=center></div>
</body>
</html>

```

CHAPTER-V

SYSTEM TESTING

5.1 Testing

Testing is carried out to detect errors and ensure a product works as expected. It involves checking different components, either individually or as part of a larger system, to confirm they function correctly.

In software, testing helps verify that the system meets user needs and performs reliably without serious issues. It ensures the software behaves as intended under different conditions. There are various types of tests, each designed to check specific features or requirements, contributing to the overall quality and dependability of the software.

5.2 Types of Testing

Unit testing:

Unit testing is a type of structural testing that focuses on checking individual components of a program. It requires understanding how the code is built and often involves directly interacting with the internal parts of the software.

This type of testing ensures that each specific business process, system setup, or application function behaves as expected. Test cases are carefully created to verify that the logic inside the code works correctly and that given inputs lead to the correct outputs. Unit tests are designed to cover every possible decision point and internal path within the code, helping to confirm that each part performs according to the defined requirements.

Integration testing:

Integrated software components are tested to see if they truly function as a single program using integration tests. The primary focus of event-driven testing is the fundamental results of fields or screens. Unit testing successfully shows that the components were satisfied independently, but integration tests indicate that the components are correctly and consistently

combined. The specific goal of integration testing is to identify issues that result from the merging of different components.

Functional testing:

Functional tests provide effective challenges to ensure that the capabilities tested are available based on the specific needs of the business, framework documentation, and client guides.

Output: The specified application output kinds must be exercised.

Systems/processes: Interacting systems or processes must be called.

Prerequisites, key competencies, or unique experiments are used to group and arrange practical examinations. Information fields, preset procedures, progressive procedures, and an effective scope for detecting business procedure streams must all be considered in testing.

Before utilitarian testing is finished, other tests are found and the powerful estimation of the current tests is fixed.

System Test:

System testing confirms that the integrated software system as a whole satisfies all requirements. It evaluates a configuration to ensure known and expected results. Process flows and descriptions, with an emphasis on integration points and pre-driven process links, form the basis of system testing.

White Box Testing:

White box testing is a method of software testing where the tester has knowledge of the program's internal structure, logic, and source code. This approach allows the tester to examine how the software is built and how it should behave in different situations.

The main purpose of white box testing is to check parts of the code that aren't visible or testable from the outside, which black box testing cannot reach. By understanding the inner workings, testers can create more detailed test cases and ensure that all logical paths, conditions, and code branches function correctly.

Test cases table for White Box Testing:

Code Area	Test Focus	Logic Covered	Expected Outcome
Encryption Module	Check correct key is used and data is changed properly	If/else: key validity, file encoding	Properly encrypted output
Re-encryption Logic	Check if file re-encrypts only for users with correct attributes and time	Condition: attribute match and time check	Re-encryption occurs only on valid match
Database Query Handling	Check SQL query execution. Make sure database queries run without errors	Try-catch and loops for DB connection	Data fetched without errors
File Permission Evaluation	Test if only the right users can open the file	Multiple branches for user roles	Only authorized roles allowed to proceed
Exception Handling	Checks how the system reacts to errors	Test with wrong inputs, null values, missing attributes	Show proper error without system crash

Black Box Testing:

A software testing technique known as "black box" testing involves the tester not knowing anything about the core code, architecture, or programming language of the system under test. Rather, the emphasis is entirely on the software's behavior as perceived by the user. This method treats the software as a "black box" with its core operations concealed. Without being aware of the internal workings of the system, testers supply inputs and watch the results. These tests, which make that the program behaves as predicted without examining how it does so, are typically reliant on outside sources such as requirement documents or specifications.

Test cases table for Black Box Testing:

Feature/Function	Input	Expected Output	Pass Criteria
Upload File	File, attributes	File uploaded and stored in cloud	System displays "Upload successful"
Re-encryption Request	File ID, policy, re-encryption key	File shared to permitted user	Access control verified
File Access by End User	Encrypted file + valid key	File content displayed	Content shown only if access is valid
Search File Details (CSP)	File ID or name	File metadata shown	All file details retrieved correctly
Permission Request	Request by unauthorized user	Request sent, access denied or pending	Unauthorized user restricted

Unit Testing:

In this type of testing, the software is viewed as a "black box," meaning its internal structure or code is not considered. Test cases are created based on a clear source document, such as a requirements or specification document. The tester provides inputs and observes the outputs to verify correct behaviour, without needing to understand how the system processes those inputs internally.

Test cases table for Unit Testing:

Module Name	Functionality Tested	Test Case Description	Expected Output
Data Owner	Encrypt and upload file	Upload file with valid content	Encrypted file saved successfully
Private Key Generator	Generate private key based on request	Generate key for a specific user	Unique private key generated
Data Disseminator	Create re-encryption request	Re-encrypt file for new user with valid attributes	Re-encryption successful
End User	Access and decrypt file	Decrypt file using private key	Original content displayed
CSP	Grant or deny search/data access	Evaluate request based on role and policy	Access granted or denied

5.3 Test Strategy and Approach

Functional tests will be meticulously planned, and field testing will be done by hand.

The examination's goals

- All field entries must work properly.
- To make the pages active, click on the assigned link.
- Responses, communications, and the input screen shouldn't be delayed.

The following are some testable features

- All links must direct users to the appropriate page
- Entries must be in the correct format
- No duplicate entries may be accepted.

Integration Testing:

Software integration testing is the practice of gradually integrating many software components on the same platform to identify any possible issues that may come from their interaction. Finding interface-related problems that may lead to component communication failures is the aim of this type of testing. The main goal is to ensure that different parts of a system, or different business applications, work together smoothly and without errors.

Test Outcomes: Every scheduled test case was successfully finished, and no flaws or problems were discovered throughout the testing procedure.

Acceptance Testing:

End users actively participate in user acceptance testing to ensure that the system operates in accordance with the established business requirements. It assists in verifying that the finished product satisfies user requirements and is prepared for practical application.

Test Results: Every test scenario was carried out satisfactorily, and no flaws were found during the procedure.

CHAPTER-VI

CONCLUSION & FUTURE ENCHANCEMENTS

6.1 Conclusion

Data security and privacy remain major concerns in cloud computing, especially when handling data shared by multiple owners. Ensuring confidentiality while managing diverse privacy preferences is a significant challenge. . A secure group data sharing and conditional dissemination model designed for cloud contexts is presented in this research. Using Identity-Based Broadcast Encryption (IBBE), a data owner can encrypt and share data with multiple users efficiently. The system integrates Ciphertext-Policy Re-Encryption (CPRE), allowing only users with matching attributes to re-encrypt and access data. A multiparty access control framework is proposed, enabling co-owners to attach their access policies. To resolve policy conflicts, three aggregation strategies—full permit, owner priority, and majority permit—are implemented, ensuring flexible and privacy-aware data sharing in multi-owner scenarios.

6.2 Future Enhancements

- Even stronger encryption methods can be adopted to store data more securely.
- Real-time tracking capabilities could be included to observe deposited feeders behaviour and any unexpected activities concurrently.
- Multi-factor authentication can also be implemented for improving security level while logging into the applications to stop the unauthorized access.
- The platform can be extended to cover larger numbers of users and amount of data especially in big organizations.
- automatic security threat detection and response through artificial intelligence possible.
- A more modern user interface can be designed and made accessible to all, included those with lesser technical background.
- A mobile application could also be created to enable users to securely access and administer their information remotely.

REFERENCES

1. C. Delerablée, “Identity-based Broadcast Encryption with Constant Size Ciphertexts and Private Keys,” Proc. the 13th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2007), pp. 200-215, 2007.
2. J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy Attributebased Encryption,” Proc. the 28th IEEE Symposium on Security and Privacy (S&P 2007), pp. 321-334, 2007.
3. Z. Wan, J. Liu, and R. Deng, “HASBE: A Hierarchical Attribute-based Solution for Flexible and Scalable Access Control in Cloud Computing,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 743-754, 2012.
4. M. Blaze, G. Bleumer, and M. Strauss, “Divertible Protocols and Atomic Proxy Cryptography,” Proc. Advances in Cryptology-EUROCRYPT 1998 (EUROCRYPT '98), pp.127-144, 1998.
5. Y. Yang, H. Lu, J. Weng, Y. Zhang, and K. Sakurai, “Fine-grained Conditional Proxy Re-encryption and Application,” Proc. the 8th International Conference on Provable Security (ProvSec 2014), pp. 206-222, 2014.
6. J. Hong, K. Xue, W. Li, and Y. Xue, “TAFC: Time and Attribute Factors Combined Access Control on Time-Sensitive Data in Public Cloud,” Proc. 2015 IEEE Global Communications Conference (GLOBECOM 2015), pp. 1-6, 2015.
7. R. Rivest, A. Shamir, and D. Wagner, “Time Lock Puzzles and Timed-release Crypto,” Massachusetts Institute of Technology, MA, USA, 1996.
8. J. Zhang, Z. Zhang, H. Guo, “Towards Secure Data Distribution Systems in Mobile Cloud Computing,” IEEE Transactions on Mobile Computing, 2017, doi: 10.1109/TMC.2017.2687931
9. J. Zhang, Z. Zhang, H. Guo, “Towards Secure Data Distribution Systems in Mobile Cloud Computing,” IEEE Transactions on Mobile Computing, 2017, doi: 10.1109/TMC.2017.2687931

10. Y. Yang, H. Lu, J. Weng, Y. Zhang, and K. Sakurai, "Fine-grained Conditional Proxy Re-encryption and Application," Proc. the 8th International Conference on Provable Security (ProvSec 2014), pp. 206-222, 2014.
11. K. Graves, Ceh: Official certified ethical hacker review guide: Exam 312-50. John Wiley & Sons, 2007.
12. R. Christopher, "Port scanning techniques and the defense against them," SANS Institute, 2001.
13. M. Baykara, R. Das , and I. Karado ğan, "Bilgi g üvenli ği sistemlerinde kullanılan arac ın incelenmesi," in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.
14. S. Robertson, E. V. Siegel, M. Miller, and S. J. Stolfo, "Surveillance detection in high bandwidth environments," in DARPA Information Survivability Conference and Exposition, 2003. Proceedings, vol. 1. IEEE, 2003
15. M. Armbrust et al. (2010) – Provide a clear overview of challenges and solutions in cloud computing.
16. H. Shacham and B. Waters (2008) – Present compact proofs for ensuring data is stored correctly.
17. Q. Jiang et al. (2019) – Propose lightweight public auditing suitable for mobile cloud systems.
18. J. Zhang, Z. Zhang, H. Guo, "Towards Secure Data Distribution Systems in Mobile Cloud Computing," IEEE Transactions on Mobile Computing, 2017, doi: 10.1109/TMC.2017.2687931
19. S. Kamara and K. Lauter (2010) – Discuss searchable encryption for secure cloud data handling.
20. C. Erway et al. (2009) – Support secure updates and audits for dynamic cloud data.
21. K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.