

A Major -Project(Stage- II) Report
on
ENCRYPTED FILE SHARING SYSTEM THROUGH
SECURED CLOUD STORAGE

Submitted in partial fulfillment of the requirements
for the award of degree of
BACHELOR OF TECHNOLOGY

in
Information Technology

by
Kundhi Ravalika (19WH1A1289)
Anusha Reddy(19WH1A1291)
Bushra Begum(19WH1A1292)
Ramya Sree(19WH1A12A7)

under the esteemed guidance of

Dr. J. Kavitha
Associate Professor



Department of Information Technology

BVRIT HYDERABAD College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)
(NAAC ‘A’ Grade & NBA Accredited- ECE, EEE, CSE, IT)

June, 2023

DECLARATION

We hereby declare that the work presented in this project entitled “ENCRYPTED FILE SHARING SYSTEM THROUGH SECURED CLOUD STORAGE” submitted towards completion of the Major Project(Stage - II) of the Project in IV year I sem of B.Tech IT at “BVRIT HYDERABAD College of Engineering for Women”, Hyderabad is an authentic record of our original work carried out under the esteem guidance of Dr. J. Kavitha, Associate Professor, Department of Information Technology.

Kundhi Ravalika (19WH1A1289)

Anusha Reddy (19WH1A1291)

Bushra Begum(19WH1A1292)

Ramya Sree (19WH1A12A7)



BVRIT HYDERABAD

College of Engineering for Women

Rajiv Gandhi Nagar, Nizampet Road, Bachupally, Hyderabad – 500090

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

(NAAC 'A' Grade & NBA Accredited- ECE, EEE, CSE IT)

CERTIFICATE

This is to certify that the major-project report on “ENCRYPTED FILE SHARING SYSTEM THROUGH SECURED CLOUD STORAGE” is a bonafide work carried out by **Kundhi Ravalika (19WH1A1289)**, **Anusha Reddy (19WH1A1291)**, **Bushra Begum (19WH1A1292)** and **Ramya Sree (19WH1A12A7)** in the partial fulfillment for the award of B.Tech degree in **Information Technology**, **BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad** affiliated to **Jawaharlal Nehru Technological University, Hyderabad** under my guidance and supervision.

The results embodied in the project work have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide

Dr. J. Kavitha

Associate Professor

Department of IT

Head of the Department

Dr.Dr. Aruna Rao S L

Professor & HoD

Department of IT

External Examiner

ACKNOWLEDGEMENT

We would like to express our profound gratitude and thanks to **Dr. K. V. N. Sunitha, Principal**, BVRIT HYDERABAD for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. Aruna Rao S L, Professor & Head**, Department of IT, BVRIT HYDERABAD for all the timely support, constant guidance and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Dr. J. Kavitha, Associate Professor**, Department of IT, BVRIT HYDERABAD for her constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinator **Ms. K. Niraja , Assistant Professor** and **Dr. P Kayal, Associate Professor**, all the faculty and staff of Department of IT who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

Kundhi Ravalika (19WH1A1289)

Anusha Reddy (19WH1A1291)

Bushra Begum(19WH1A1292)

Ramya Sree (19WH1A12A7)

ABSTRACT

It is possible to share data using the cloud on a massive scale because it is location independent. Customers can more easily access or share documents by sharing digital records in public cloud storage. However, issues related to data privacy and security, simple data exchange, effective delegation of authority and optimization of computing speed remain in the direction of achieving practical access control within the framework of electronic document exchange. The proposed system is an access control system and data exchange procedure for a system that fulfills all the main characteristics in parallel and is suitable for resource-based devices. Mail notifications for One Time Password are part of this project (One Time Password) to ensure more security and data privacy. In addition, this system offers the possibility of automatic deletion. The file is automatically deleted based on the expiration date specified by the user, so that memory is optimized. The access control method will be used in a real-world environment, including public cloud storage, a laptop, and a resource-constrained local server that is affordable. Practical results show that the procedure is effective, practical and affordable.

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.2.1	Java	7
3.2.2	Mysql	8
3.2.3	Tomcat Apache Server	9
3.3.1	Usecase of admin	10
3.3.2	Usecase of Employee	10
4.1	Architecture	13
5.1	Home Page	27
5.2	Admin loginPage	28
5.3	View Users	29
5.4	Cloud Uploads	30
5.5	Response status	31
5.6	Change Password	32
5.7	Change Secret Key	33
5.8	View request	34
5.9	Admin Table	35
5.10	Uploaded files in System	36
5.11	View Transactions	37
5.12	File RequestStatus	38
5.13	Employees Table	39

CONTENTS

TOPIC	PAGE NO.
DECLARATION	I
CERTIFICATE	II
ACKNOWLEDGEMENT	III
ABSTRACT	IV
LIST OF FIGURES	V
TABLE OF CONTENTS	VI
1. Introduction	1
1.1 Objective	3
1.2 Problem Definition	3
2. Literature Survey	4
3. System Design	6
3.1 Project flow	6
3.2 Technologies	7
3.2.1 Java	7
3.2.2 MySQL	8
3.2.3 Apache Tomcat Server	9
3.3 UML Diagram	10
3.4 S/W and H/W Requirements	12
4. Methodology	13
4.1 Architecture	13
4.2 Modules	14
5. Implementation & Coding	15
6. Implementation & Results	27
References	40

1. Introduction

1.1 Cloud Computing Definition

Cloud computing is that the utilization of registering assets (equipment and programming) that are conveyed as an assistance over a system (normally the Internet). The name originates from the traditional utilization of a cloud-formed image as a mirrored image for the unpredictable foundation it contains in framework charts. Cloud computing endows far away administrations with a client's information, programming, and calculation. Cloud computing comprises of kit and programming assets made accessible on the web as over saw outsider administrations. These administrations regularly give access to leading edge programming applications and top of the road systems server PCs.

The objective of cloud computing is to use conventional super figuring, or elite registering power, typically utilized by military and exploration offices, to perform many trillions of calculations for every second, in buyer arranged applications, for instance, monetary portfolios, to convey customized data, to offer information stock piling or to regulate enormous, vivid PC games

1.1.2 Cloud Computing Benefits

Accomplish economies of scale – increment volume yield or efficiency with less individuals. Your expense per unit, undertaking or item dives.

Decrease spending on innovation framework. continue simple access to your data with negligible forthright spending. Pay more only as costs arise (week by week, quarterly or yearly), in sight of interest.

Globalize your workforce for barely anything. Individuals worldwide can get to the cloud if they need an online association.

1.1.3 Information DeDuplication

Information deduplication implies diminish the copy information in distributed storage. Information deduplication has been displayed to be a viable strategy in Cloud reinforcement and documenting applications to decrease the reinforcement window, improve the additional room effectiveness and system transfer speed usage. On-going investigations uncover that moderate to high information repetition obviously exists in VM (Virtual Machine), endeavour and High-Performance Computing (HPC) stockpiling frameworks.

1.1.4 Encryption and Decryption

Encryption is that the way toward interpreting plain content information (plaintext) into something that provides off an impact of being arbitrary and unimportant (ciphertext). Decoding is that the way toward changing over ciphertext back to plaintext.

To encode quite a modest quantity of data, symmetric encryption is employed. A symmetric key's utilized during both the encryption and decoding forms. To unscramble a touch of ciphertext, the key that was utilized to scramble the knowledge must be utilized.

The objective of every encryption calculation is to form it as troublesome as conceivable to unscramble the produced ciphertext without utilizing the key. Within the event that an honest encryption calculation is employed, there's no strategy essentially better than deliberately attempting each conceivable key. For such a calculation, the more drawn out the key, the more troublesome it's to decode a touch of ciphertext without having the key.

It is hard to make a decision the character of an encryption calculation. Calculations that watch promising a number of the time find you being quite simple to interrupt, given the simplest possible assault.

1.2 OBJECTIVE OF THE PROJECT

The main objective of the project is to provide more efficient and secured cloud storage for employees to store and share their data one another easily. The main objectives are providing security, user friendliness, sharing environment, memory optimization and finally improve the performance. This project developed for the company employees to manage their data and files globally.

1.3 PROBLEM DEFINITION

The problems faced in cloud computing are security, ease of data sharing from one another, information privacy, efficient authority management, computation speed optimization, storage and they are remaining towards achieving practical access control in the Electronic Document Sharing. Proposed application is an innovative access control system and a fine-grained data sharing process for data sharing system, which paralelly achieves the all required features and is suitable for resource-based devices for employees in a company.

2. Literature Survey

Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud

AUTHORS: Miguel Morales-Sandoval; Melissa Hinojosa Cabello; Heidy Marisol Marin-Castro; Jose Luis Gonzalez Compean, In this paper we address the security concerns of cloud storage under the scenario where users encrypt-then-outsource data, share their outsourced data with other users, and the service provider can be queried for searching and retrieval of encrypted data. As main distinctive, we propose a security approach for storage, sharing and retrieval of encrypted data in the cloud fully constructed on the basis of attribute-based encryption (ABE) thus enabling access control mechanisms over both the Encrypted data and also for the information retrieval task through search access control. Compared to related works, our approach considers efficient encryption at three different levels: i) bulk encryption of data outsourced to the cloud, ii) keys management for access control over encrypted data by means of digital envelopes from attribute based encryption, and iii) novel construction for attribute based searchable encryption (ABSE). Our underlying ABE algorithms are carefully selected from the body of knowledge and novel constructions for ABSE are provided over the asymmetric setting (Type-III pairings) to support security levels of 128-bits or greater.

Advanced Encryption Standard Algorithm for File Security **AUTHORS:** : Olasunkanmi Felix Oyadokun ; Danjuma Shadrach Sunday ; Haruna Bege ; Kolawole S.F The efficacy of the Advanced Encryption Standard (AES) algorithm has made it very attractive for data encryption. As a result of this, it has been employed by many large organizations for safeguarding files in any binary format. It makes use of a symmetric key to achieve successful data encoding. To encrypt a file, it uses different key sizes. Although a bigger key size increases the degree of unpredictability, it also increases the encryption time. AES utilizes Add round key, Byte substitution, Shift rows and Shift column matrix operations together with a Galois Field computation in Modulus two (MOD 2). This paper shows the step by step process of how data is Encrypted using the AES.

Data Security in Cloud using AES

AUTHORS: Smitha Nisha Mendonca

One of the important services provided by the cloud is storage where users can keep their data as per the requirement. It is a challenging issue for the user as all the data are stored in some interconnected resource pool but this resource pool is situated over different places of the world. An unauthorized user may be access this data through the virtual machines. So, it is the dark side of cloud data storage. This insecurity creates a big problem for users. Therefore data security in cloud computing is a major problem. Currently, AES is regarded as the most popular symmetric cryptographic algorithm. It is very significant to develop high performance AES to further broaden its widespread application.

3. System Design

This project was developed using java technology and the main reason to select the java technology is, it is open source and programmer friendly. MySQL is used as database due to its simple organization and free cost availability. The apache tomcat server was implemented to deploy and run the web application and it is also open source.

Below mentioned is the System Design of the project explained in detail.

3.1 Project Flow

In this system we have designed a system which is simple, secure, and more efficient system to share the document among the employees in an organization. Proposed algorithm is AES256 algorithm to encrypt and decrypt the data.

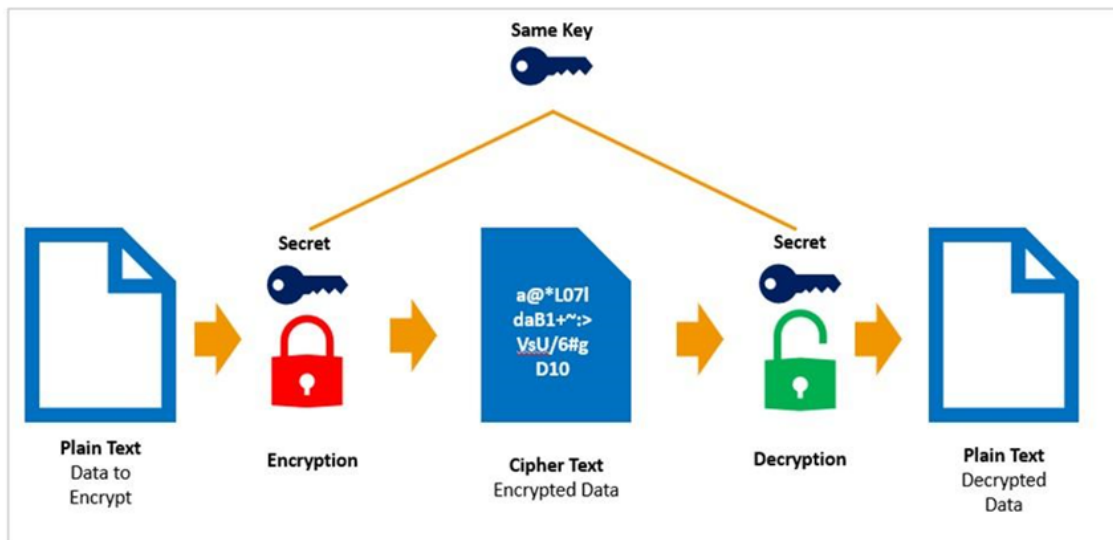


Figure 3.1: Project Flow

3.2 Technologies

This system uses HTML as frontend, Java as backend and MySQL to store the data from the system.

3.2.1 Java

JAVA was developed by James Gosling at Sun Microsystems Inc in the year 1995, later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs. Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to c/c++. The principles for creating java were simple, robust, secured, high performance, portable, multi-threaded, interpreted, dynamic, etc. In 1995 Java was developed by James Gosling, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.



Figure 3.2.1: Java

3.2.2 MySQL

MySQL is the world's most popular open source database. According to DB-Engines, MySQL ranks as the second-most-popular database, behind Oracle Database. MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

Since MySQL is open source, it includes numerous features developed in close cooperation with users over more than 25 years. So it's very likely that your favourite application or programming language is supported by MySQL Database.

MySQL Database is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application-programming interfaces (APIs). We also provide MySQL as an embedded multithreaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.



Figure 3.2.2: MySQL

3.2.3 Tomcat Server

Apache Tomcat is open source web server software for Java programming developed and maintained by the Apache software foundation. The initial idea of Apache tomcat software was to host and deploy the Java servlet that is the server-side Java code that manages HTTP results from client applications build using Java. It acts as a web server rather than a full-fledged application server that includes data persistence and load balancing capabilities. Apache Tomcat provides the basic feature of web server processing for the relevant servlets. It supports the java servlet lifecycle that are `init()`, `service()` and `destroy()` phases. It is the preferred web server software for Java implementations The latest stable release of a tomcat version 9.0.21, was released on June 7th, 2019. Apache tomcat may be defined as a web server (that is also referred to as a web container/ servlet container), which processes the servlets, JSP's (by internally converting your JSP's to servlets internally), and also render JSP's. If we want to send dynamic data or to make our website dynamic, we need to use the servlet. Hence, we need an HTTP server and what else we need is a container where we will run or servlet, so when we combine the HTTP server and the servlet (or we can say servlet container), they both combine to become a single server know as tomcat server. In simple words, we can say that The Apache Tomcat is actually a server and a servlet container.



Figure 3.2.3: Apache Tomcat

3.3 UML Diagrams

3.3.1 Use Case Diagrams

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

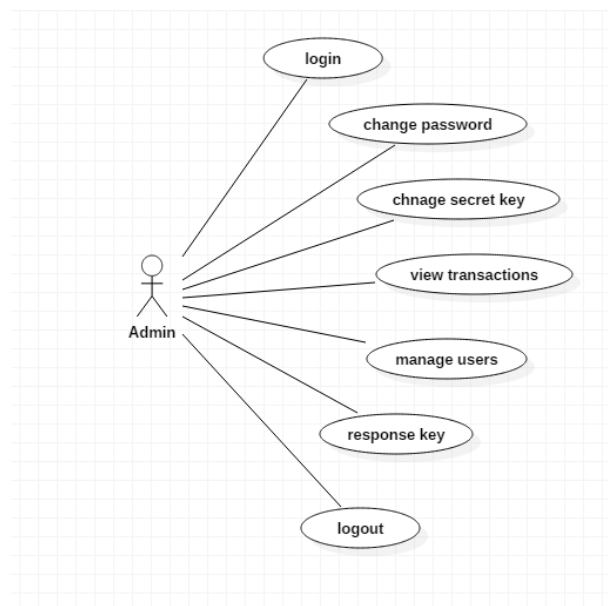


Figure 3.3.1: Use Case Of Admin

The Use case of the Admin diagram says that admin can login with his credentials. After logging in he can manage users and also access all the files in the system.;

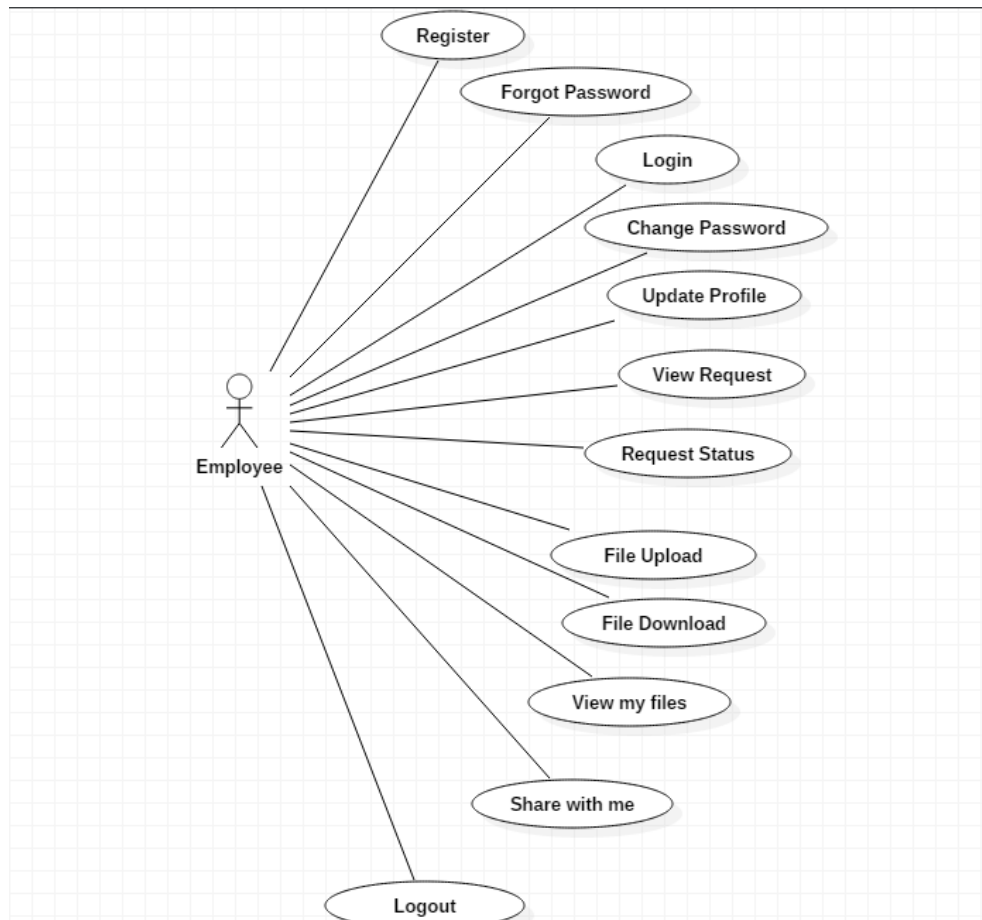


Figure 3.3.2: Use Case Of Employee

The Use case diagram of user says that user can login and enter his details, and view his/her profile. User can request for the access of various files in the system. He can also upload files into the system. Employee can also view the status of his requested files .

3.4 Software and Hardware Requirements

3.4.1 Software Requirements

Operating System	-	Windows
Coding Language	-	Java
Front End	-	HTML, CSS, JavaScript
Server	-	Apache Tomcat
Back End	-	MySQL

3.4.2 Hardware Requirements

Processor - Intel

RAM - 4 GB(min)

Hard Disk - 160 GB

4. Methodology

4.1 Architecture

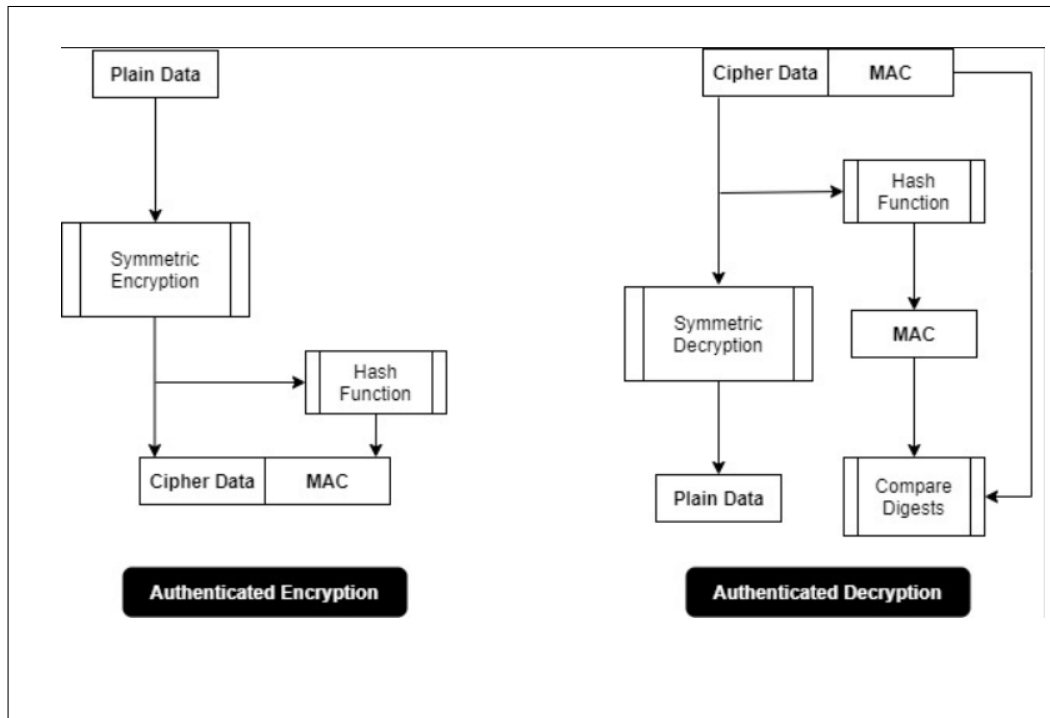


Figure 4.1: Architectures

In the Figure 4.1 a system is displayed with backend as Java. Java retrieves the data from database with MySQL.

4.2 Modules

Modules are important to have a precise overview on the development of the project process so that while execution clarity of the next step is maintained. The encrypted file system has following Modules.

- i) Admin Module
- ii) Employee Module
- iii) Data Auto Deletion

i) 4.2.1 Administration

This module consists administration activities like view users, authorize users, key generation, view transactions, view uploaded files etc. The admin acts as the master of the system. He has control over all the files of the system. He can accept or reject any user at any point of time. He can view the transactions in the system and can share secret key to the employees.

ii) 4.2.2 Employee

In this module employee can register, login and upload, download the files and share the files. A new employee must register and create his new profile. He can upload files on his own and share access with his fellow employees

iii) 4.2.3 Data Auto Deletion

In this module the main goal is auto deletion of files once expiry date is reached. Once the specified date of expiry is set the system automatically deletes the file. The default date of expiry set for a file is 24 hours i.e. 1 day.

5. Implementation

The Admin Module that we implemented has a welcome to Admin page, here the admin can login using the password and will be able to access all the files and Transactions that are being made in the systems.

5.1 Code

DBCON

```
package databa seconnection;
import java.sql.Conne ction;
import java.sql.Drive rManager;

public class DBCon {
    static Connecti on con=null ;
    public static Connecti on getConnec tion()
    {
        try{
            Class.forName("com.m ysql.jdbc.Driver");
            con=DriverManage r.getConnection("jdbc:mysql:// localhost:3306/ twofactor",
                "root","root") ;

        }catch(Exception e){
            System.out.pritln(e);
        }
    }
}
```

AES ALGORITHM

```
package action;

import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.spec.AlgorithmParameterSpec;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.KeySpec;
import java.util.Arrays;

import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
import javax.print.DocFlavor.STRING;

import org.apache.commons.codec.binary.Base64;

public class AES
{

// private static SecretKeySpec secretKey ;
    private static byte[] key ;
```

```
public static SecretKeySpec setKey(String myKey){
    MessageDigest sha = null;
    SecretKeySpec secretKey=null;
    try {
        key = myKey.getBytes("UTF-8");
        System.out.println(key.length);
        sha = MessageDigest.getInstance("SHA-1");
        key = sha.digest(key);
        key = Arrays.copyOf(key, 16); // use only first 128 bit
        System.out.println(key.length);
        System.out.println(new String(key,"UTF-8"));
        secretKey = new SecretKeySpec (key, "AES" );

    } catch (NoSuchAlgorithmException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return secretKey;
}

public static String getDecryptedString() {
    return decrypte dString;
}
```



```
        return EncryptedString;
    }

    public static void setEncryptedString(String EncryptedString) {
        AES.EncryptedString = EncryptedString;
    }

    public static String encrypt(String strToEncrypt, String myKey)
    {
        SecretKeySpec secretKey=null;
        String edata=null;
        MessageDigest sha = null;
        try {
            key = myKey.getBytes("UTF-8");
            System.out.println(key.length);
            sha = MessageDigest.getInstance("SHA-1");
            key = sha.digest(key);
            key = Arrays.copyOf(key, 16); // use only first 128 bits
            System.out.println(key.length);
            System.out.println(new String(key,"UTF-8"));
            secretKey= new SecretKeySpec (key, "AES" );

        } catch (NoSuchAlgorithmException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (UnsupportedEncodingException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

```
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);

        edata = Base64.encodeBase64String(cipher.doFinal
(strToEncrypt.getBytes("UTF-8")));

    }
    catch (Exception e)
    {

        System.out.println("Error while encrypting: "+e.toString());
    }
    return edata;

}

public static String decrypt(String strToDecrypt, String myKey)
{
    SecretKeySpec secretKey = null;
    MessageDigest sha = null;
    String fdata = null;
    try {
        key = myKey.getBytes("UTF-8");
        System.out.println(key.length);
        sha = MessageDigest.getInstance("SHA-1");
        key = sha.digest(key);
        key = Arrays.copyOf(key, 16); // use only first 128 bits
        System.out.println(key.length);
        System.out.println(new String(key, "UTF-8"));
    }
```

```
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

try
{
    Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");

    cipher.init(Cipher.DECRYPT_MODE, secretKey);
    fdata=new String(cipher.doFinal(Base64.decode(Base64(strToDecrypt))));

}
catch (Exception e)
{

    System.out.println("Error while decrypting: "+e.toString());

}

return fdata;
}

public static void main(String args[])
{

    final String strToEncrypt = "venugudluri";
    final String strPassword = "encryptorkey";
```

```
        System.out.println("Encrypted : " + AES.getEncryptedString());

        final String strToDecrypt = AES.getEncryptedString();
        AES.decrypt(strToDecrypt.trim(), strPssword);

        System.out.println("String To Decrypt : " + strToDecrypt);
        System.out.println("Decrypted : " + AES.getDecryptedString());

    }

}
```

EMAIL CONNECTIVITY

```
package action;

import java.util.Properties;
import javax.mail.*;
import javax.mail.internet.*;

public class Email {

    public static boolean sendMail(String sub, String msg, String userid, String to) {
        Properties props = new Properties();
        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.socketFactory.port", "465");
        props.put("mail.smtp.socketFactory.class",
            "javax.net.ssl.SSLSocketFactory");
    }
}
```

```
        return new PasswordAuthentication("bushrasai raaz@gmail.com", "143");
    }
});
System.out.println("Message " + msg);
try {
    Message message = new MimeMessage(session);
    message.setFrom(new InternetAddress("bushrasairaz@gmail.com"));

    message.setRecipients(Message.RecipientType.TO,
        InternetAddress.parse(to));
    message.setSubject(sub);
    message.setText(msg);
    Transport.send(message);
    return true;
}
catch (MessagingException e) {
    System.out.println(e);
    e.printStackTrace();
    return false;
    // throw new RuntimeException(e);
}
}

public static void main(String[] args) {

    Email.sendMail("check","hii", "Bush","bushrasairaz1@gmail.com");
}
```

VIEW TRANSACTIONS

```
<%@ include file="cback.jsp"%>
<%@page import="java.sql.*,databaseconnection.*"%>

<%
Connection con=DBCon.getConnection();
Statement st=con.createStatement();
ResultSetr s=st.executeQuery("select* f romnoti" );%>

<div class="container">
<div class="h3 text-center">VIEW TRANSACTION</div>
<div class="table-responsive">

<table class="table table-bordered">

<tr><th>NotiId</th><th>Name</th><th>Task</th><th>Tby</th>
<th>Tdata</th><th>Role</th><th>UploadedBy</th>
<th>ExpireDate</th></tr>

<%
while(rs.next())
{ %>
<tr><td><%=rs.getInt("notiId")%></td><td>
<%=rs.getString("fname")%></td>
<td><%=rs.getString("task") %></td>
<td><%=rs.getString("tby")%></td>
<td><%=rs.getString("tdate")%></td>
<td><%=rs.getString("role")%></td>
<td><%=rs.getString("uploadedby")%></td>
<td><%=rs.getString("expiredate")%></td>
</tr><% }
%>
```

ACCEPT REQUEST

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1 "
    pageEncoding="ISO-8859-1" import="java.sql.*"%>
<%

String userid=request.getParameter("userId");

try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/twofactor", "root", "root");
    PreparedStatement ps = con.prepareStatement("update users set status=1 where userId=?");
    ps.setString(1, userid);

    int n=ps.executeUpdate();

    //out.println("sucess");
    response.sendRedirect("viewusers.jsp");
    ps.close();
    con.close();
}
catch(Exception e)
{
    e.printStackTrace();
}

%>
```

CHANGE PASSWORD

<%

```
String username=request.getParameter("username");
```

```
String oldpwd=request.getParameter("oldpwd");
```

```
String newpwd=request.getParameter("newpwd");
```

```
try
```

```
{
```

```
    Class.forName("com.mysql.jdbc.Driver");
```

```
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/twofactor", "root", "root");
```

```
    PreparedStatement ps = con.prepareStatement("update admintbl set password=?
```

```
where username      =? and password=?");
```

```
    ps.setString(1, newpwd);
```

```
    ps.setString(2, username);
```

```
    ps.setString(3,oldpwd);
```

```
    int n = ps.executeUpdate();
```

```
    response.sendRedirect("clog.jsp");
```

```
    ps.close();
```

```
    con.close();
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
    e.printStackTrace();
```

```
}
```


Results

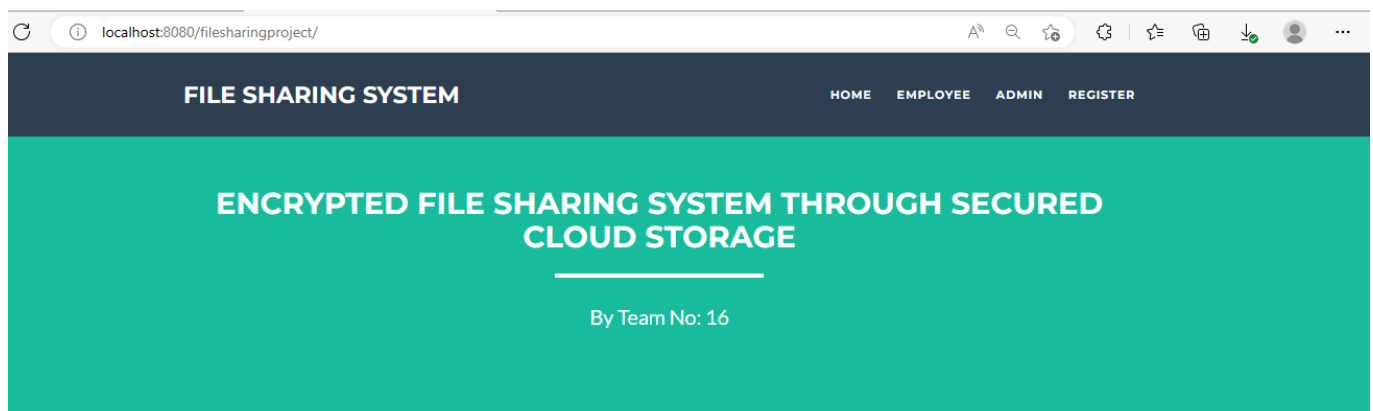


Figure 5.1: Home Page

ADMIN LOGIN PAGE

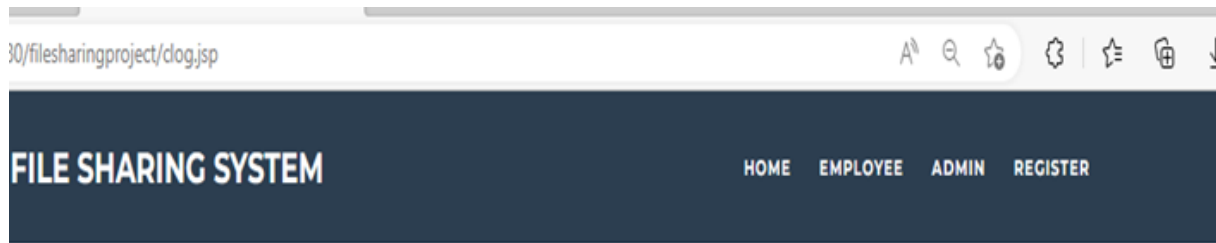
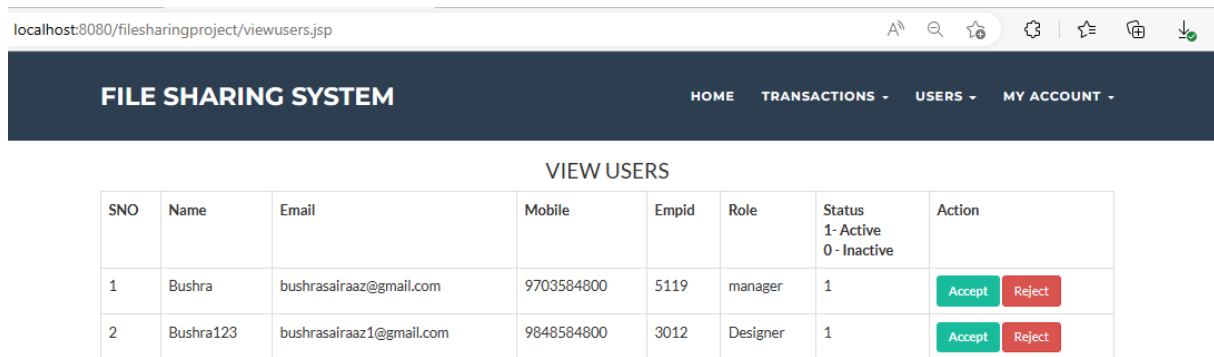


Figure 5.2 Admin login page

VIEW USERS AND MANAGE USERS



localhost:8080/filessharingproject/viewusers.jsp

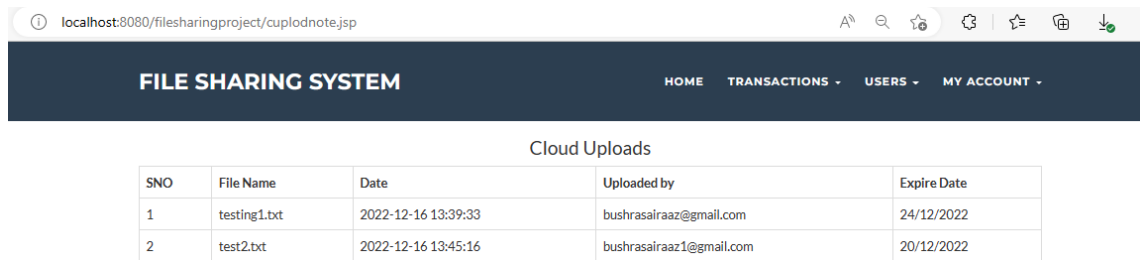
FILE SHARING SYSTEM HOME TRANSACTIONS - USERS - MY ACCOUNT -

VIEW USERS

SNO	Name	Email	Mobile	Empid	Role	Status 1- Active 0 - Inactive	Action
1	Bushra	bushrasairaaz@gmail.com	9703584800	5119	manager	1	<button>Accept</button> <button>Reject</button>
2	Bushra123	bushrasairaaz1@gmail.com	9848584800	3012	Designer	1	<button>Accept</button> <button>Reject</button>

Figure 5.3 View Users

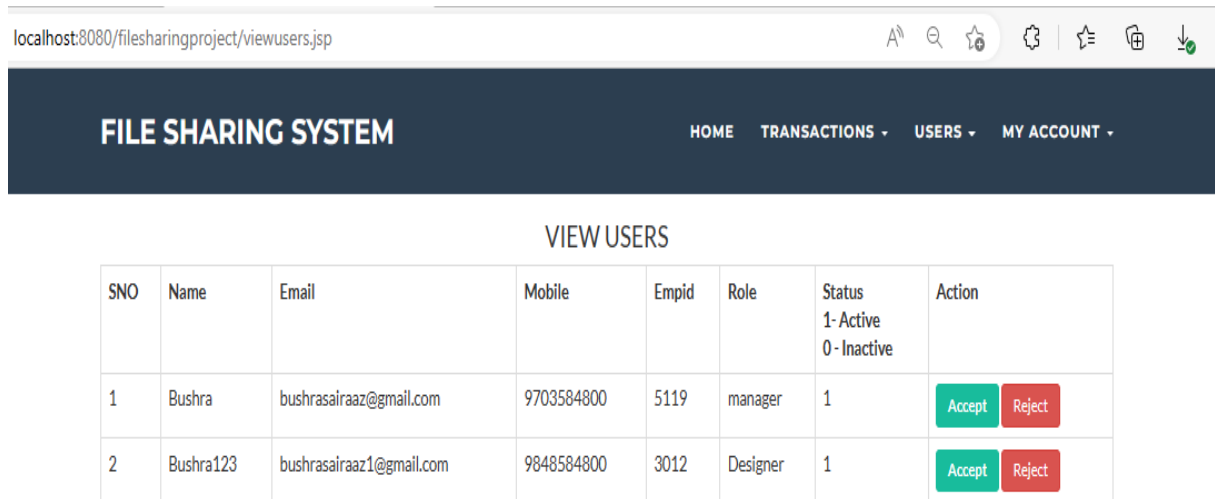
CLOUD UPLOADS



SNO	File Name	Date	Uploaded by	Expire Date
1	testing1.txt	2022-12-16 13:39:33	bushrasairaaz@gmail.com	24/12/2022
2	test2.txt	2022-12-16 13:45:16	bushrasairaaz1@gmail.com	20/12/2022

Figure 5.4 Cloud uploads with their Expiry Dates

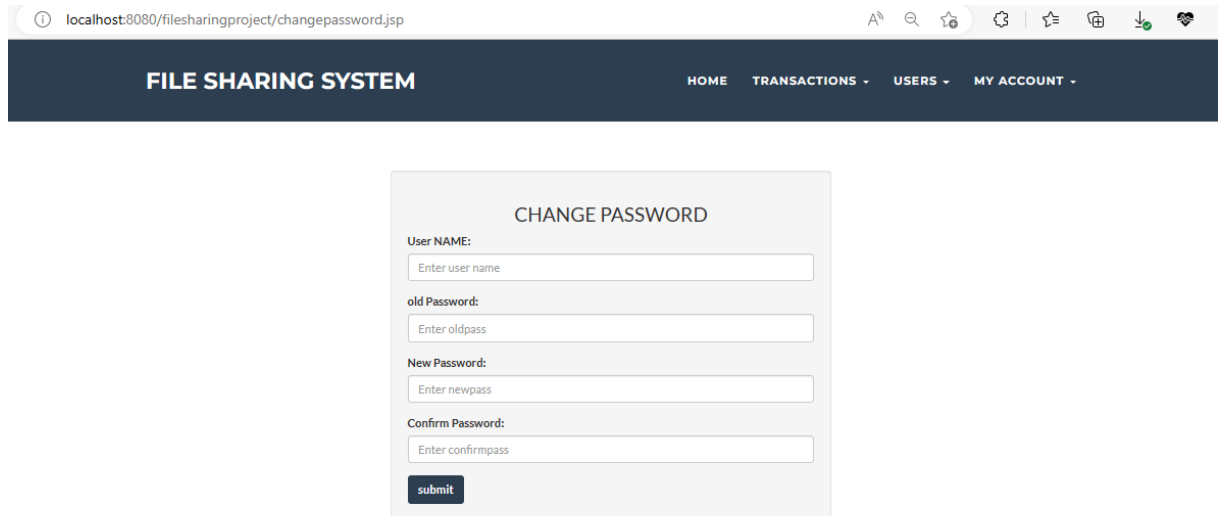
FILE SECURITY RESPONSES



SNO	Name	Email	Mobile	Empid	Role	Status 1- Active 0 - Inactive	Action
1	Bushra	bushrasairaaz@gmail.com	9703584800	5119	manager	1	<button>Accept</button> <button>Reject</button>
2	Bushra123	bushrasairaaz1@gmail.com	9848584800	3012	Designer	1	<button>Accept</button> <button>Reject</button>

Figure 5.5 View and Authorize Employees

CHANGE PASSWORD



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/filessharingproject/changepassword.jsp'. The browser's toolbar includes icons for back, forward, search, and other standard functions. Below the address bar is a dark blue navigation bar with the text 'FILE SHARING SYSTEM' on the left and a list of menu items: 'HOME', 'TRANSACTIONS -', 'USERS -', and 'MY ACCOUNT -'. The main content area is white and features a light gray rectangular form titled 'CHANGE PASSWORD'. Inside the form, there are four labeled input fields: 'User NAME:' with a placeholder 'Enter user name', 'old Password:' with a placeholder 'Enter oldpass', 'New Password:' with a placeholder 'Enter newpass', and 'Confirm Password:' with a placeholder 'Enter confirmpass'. At the bottom of the form is a dark blue button with the text 'submit' in white.

Figure 5.6 Change and Update Password

CHANGE SECRET KEY

8080/filessharingproject/skey.jsp

FILE SHARING SYSTEM

HOME TRANSACTIONS ▾ USERS ▾ MY ACCOUNT ▾

CHANGE SECERTKEY

User NAME:

Old Secertkey:

New Secertkey:

Confirm Secertkey:

submit

Figure 5.7 Change secret Key

REQUEST ACCEPT AND SEND KEY

FILE SHARING SYSTEM

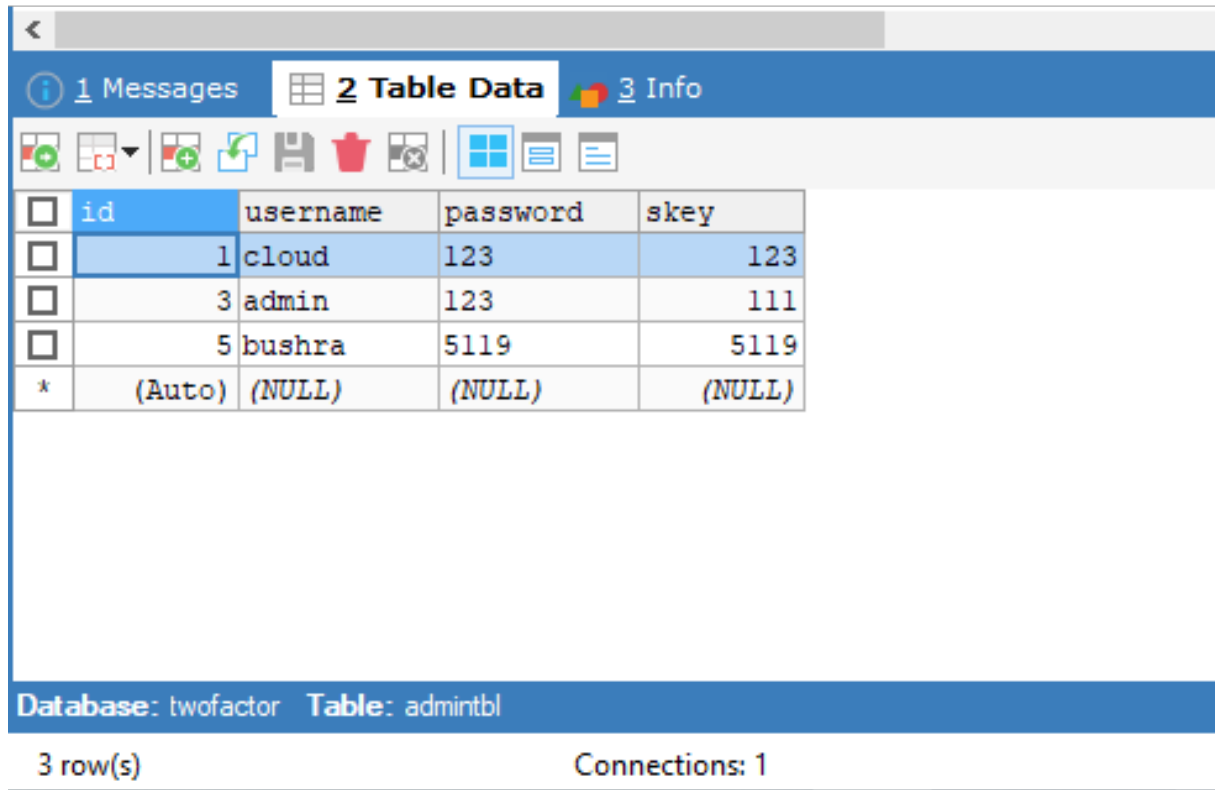
HOME TRANSECTIONS + USERS + MY ACCOUNT +

REQUEST DETAILS

SNO	File Name	Date	Request By	Send Key
1	cloud.txt	2022-11-16 19:44:20	p@gmail.com	Send key

Figure 5.8 View details and Send Key

ADMIN TABLE



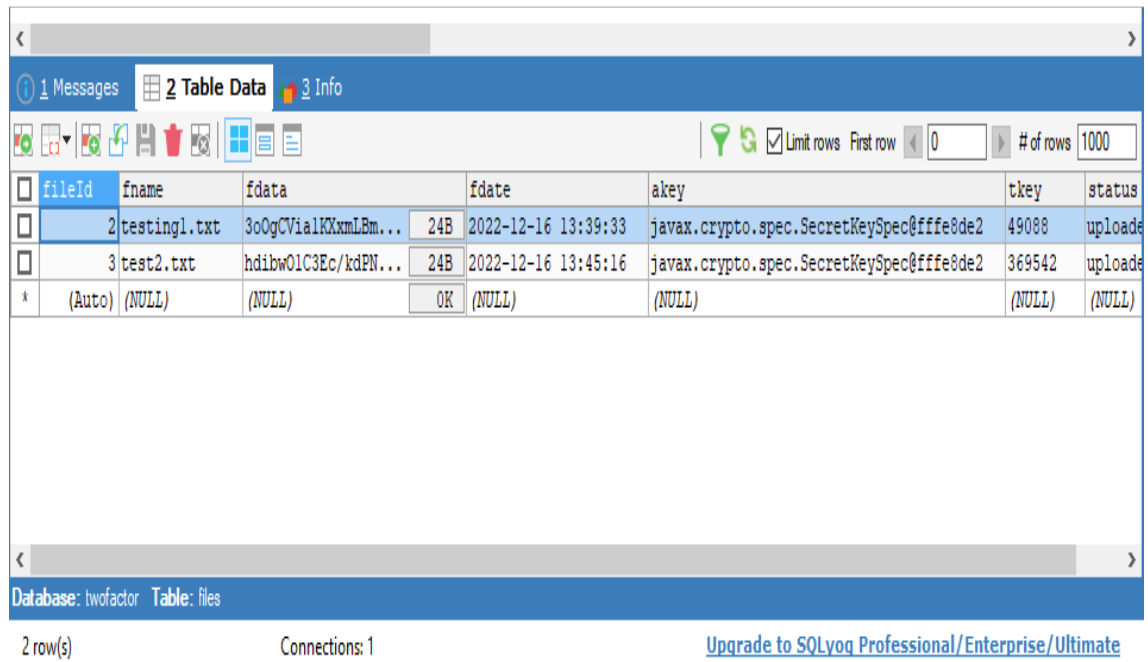
<input type="checkbox"/>	id	username	password	skey
<input type="checkbox"/>	1	cloud	123	123
<input type="checkbox"/>	3	admin	123	111
<input type="checkbox"/>	5	bushra	5119	5119
*	(Auto)	(NULL)	(NULL)	(NULL)

Database: twofactor Table: admintbl

3 row(s) Connections: 1

Figure 5.9 ADMIN TABLE

UPLOADED FILES IN SYSTEM



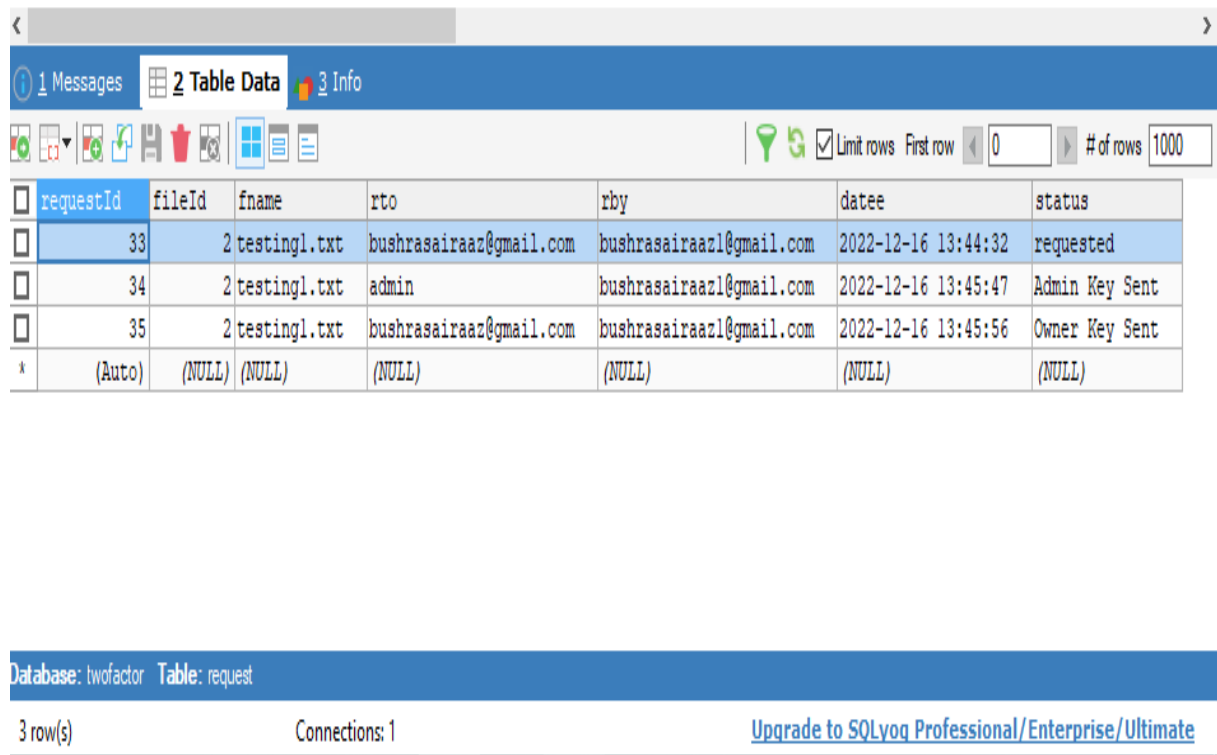
fileId	fname	fdata	fdate	akey	tkey	status
2	testing1.txt	3oOgCVialKXxmlBm...	24B 2022-12-16 13:39:33	javax.crypto.spec.SecretKeySpec@fffe8de2	49088	uploade
3	test2.txt	hdibw0lC3Ec/kdPN...	24B 2022-12-16 13:45:16	javax.crypto.spec.SecretKeySpec@fffe8de2	369542	uploade
*	(Auto)	(NULL)	OK (NULL)	(NULL)	(NULL)	(NULL)

Database: twofactor Table: files

2 row(s) Connections: 1 [Upgrade to SQLyog Professional/Enterprise/Ulimate](#)

Figure 5.10 Uploaded Files

VIEW TRANSACTIONS



The screenshot displays a database management interface with a table named 'request' containing three rows of transaction data. The interface includes a toolbar with various icons for navigation and editing, and a status bar at the bottom showing '3 row(s)' and 'Connections: 1'.

requestId	fileId	fname	rto	rby	datee	status
33	2	testing1.txt	bushrasairaz@gmail.com	bushrasairaz1@gmail.com	2022-12-16 13:44:32	requested
34	2	testing1.txt	admin	bushrasairaz1@gmail.com	2022-12-16 13:45:47	Admin Key Sent
35	2	testing1.txt	bushrasairaz@gmail.com	bushrasairaz1@gmail.com	2022-12-16 13:45:56	Owner Key Sent
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Database: twofactor Table: request

3 row(s) Connections: 1 [Upgrade to SQLyog Professional/Enterprise/Ulimate](#)

Figure 5.11 All Transactions

FILE REQUEST STATUS

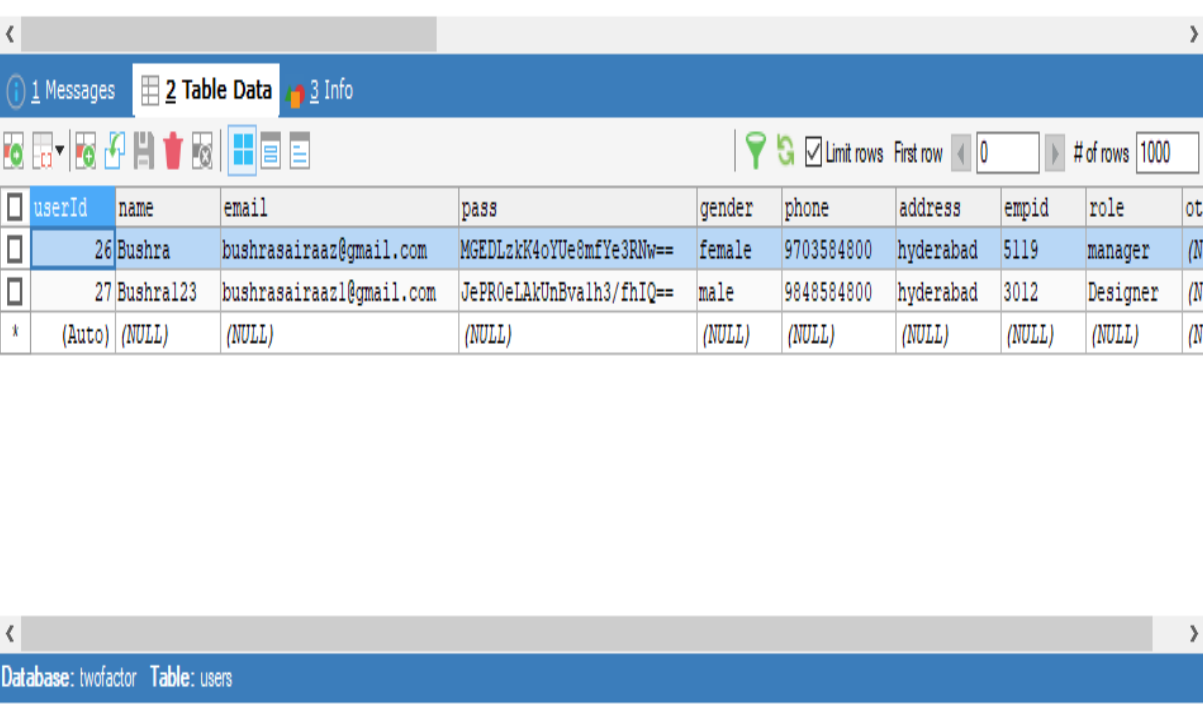
userId	name	email	pass	gender	phone	address	empid	role	ot
26	Bushra	bushrasairaaz@gmail.com	MGEDLzkk4oYUe8mfYe3RNw==	female	9703584800	hyderabad	5119	manager	(N
27	Bushra123	bushrasairaaz1@gmail.com	JePR0eLakUnBvalh3/fhIQ==	male	9848584800	hyderabad	3012	Designer	(N
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(N

Database: twofactor Table: users

2 row(s) Connections: 1 [Upgrade to SQLyog Professional/Enterprise/Ultimeate](#)

Figure 5.12 Request Status of Employees

EMPLOYEES



Database: twofactor Table: users

2 row(s) Connections: 1 [Upgrade to SQLyog Professional/Enterprise/Ulimate](#)

userId	name	email	pass	gender	phone	address	empid	role	otp
26	Bushra	bushrasairaaz@gmail.com	MGEDLzkK4oYUe8mfYe3RNw==	female	9703584800	hyderabad	5119	manager	(N
27	Bushra123	bushrasairaaz1@gmail.com	JePR0eLAKUnBvalh3/fhIQ==	male	9848584800	hyderabad	3012	Designer	(N
*	(Auto)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(N

Figure 5.13 Employee Table

FUTURE EXTENSION PLAN

To further extend this for future developments, there are few directions among which the one is to implement the same concept for various applications such as image sharing, digital wallets.

Another enhancement can be to extend the OTP verification with mobile instead of email to provide better identification of user. Further it can be implemented in public domain also.

REFERENCES

- [1] Olasunkanmi Felix Oyadokun , Danjuma Shadrach Sunday , Haruna Bege , Kolawole S .F "Advanced Encryption Standard Algorithm for File Security" Iconic Research And Engineering Journals, 6(3) IEEE 2022
- [2] Miguel Morales Sandoval, Melissa Hinojosa Cabello, Heidy Marisol Marin Castro, Jose Luis Gonzalez Compean, Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud IEEE 2022
- [3] M. Zeng, H. - F. Qian, J. Chen and K. Zhang, "Forward secure public key encryption with keyword search for outsourced cloud storage", IEEE Trans. Cloud Comput. early access, Sep. 2019
- [4] Peng Zeng, Kim-Kwang Raymond Choo, "A New Kind of Conditional Proxy Re-Encryption for Secure Cloud Storage", Access IEEE, vol. 6, pp. 70017-70024, 2018
- [5] S Smitha Nisha Mendonca "Data Security in Cloud using AES" International Journal of Engineering Research & Technology (IJERT) Vol. 7 Issue 01, January-2018.

