

TO USING HYBRID CRYPTOGRAPHY, THE CLOUD STORE FILES SECURELY

Submitted in partial fulfillment of the requirements for the award of
Master of Engineering degree in Computer Science and Engineering

By

AVANDHIKASRI M (Reg. No - 42310002)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

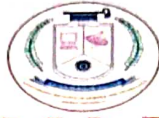
SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI -600119**

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **AVANDHIKASRI M (42310002)** who carried out the Project entitled **"TO USING HYBRID CRYPTOGRAPHY, THE CLOUD STORE FILES SECURELY"** under my supervision from January 2024 to April 2024.

Internal Guide

Dr. L. LAKSHMANAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Dr. L. LAKSHMANAN, M.E., Ph.D.
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600 119.

Submitted for Viva voce Examination held on 20/04/24

Internal Examiner

External Examiner

DECLARATION

I, AVANDHIKASRI M (42310002), hereby declare that the Project Report entitled "TO USING HYBRID CRYPTOGRAPHY, THE CLOUD STORE FILES SECURELY" done by me under the guidance of Dr. L. LAKSHMANAN, M.E., Ph.D. is submitted in partial fulfilment of the requirements for the award of Master of Engineering degree in **Computer Science and Engineering**.

DATE: 20/04/24

PLACE: CHENNAI

M. Avandhi Kasri

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. L. Lakshmanan M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Many industries, including the military, higher education, and others, use the cloud for a variety of services and the massive data storage that it offers. Without having direct access to the server computer, users can request to see or retrieve data stored in this cloud. The various organizations and IT industries are using cloud computing at a fast-increasing rate, which makes new software more affordable. Cloud computing is helpful since it's affordable and makes knowledge easily accessible. Numerous benefits of cloud computing include minimal costs and online information accessibility. Considering that customers frequently save sensitive data with cloud storage providers that may not be trustworthy ensuring cloud computing security may be a key consideration for the cloud computing ecosystem. Therefore, maintaining data from an untrusted cloud while exchanging it securely continues to be a difficult problem. By using AES and the RSA algorithm to store data over a single cloud, my method guarantees the security and privacy of important customer information.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	V
	LIST OF FIGURES	Viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
	2.1 Existing System	4
	2.2 Proposed System	5
	2.3 System Specification	6
3	REQUIREMENTS ANALYSIS	7
	3.1 Software Specification	7
	3.2 Backend	9
4	SOFTWARE DEVELOPMENT	11
	4.1 Methodology	11
	4.2 Modules Description	12
	4.2.1 Data Owner (DO)	12
	4.2.2 Data User (DU)	12
	4.2.3 Admin	12
	4.2.4 Cloud	12

5	SYSTEM DESIGN AND ANALYSIS	13
5.1	Architecture Diagram	13
5.2	Data Flow Diagram	14
5.3	UML Diagram	15
5.4	Data Base Design	19
6	SYSTEM TESTING AND IMPLEMENTATION	20
6.1	System Testing	20
6.1.1	Unit Testing Integration Testing	20
6.1.2	Integration Testing Functional Testing	21
6.1.3	Functional Testing Performance Testing	21
6.1.4	Performance Testing White Box Testing	22
6.1.5	White Box Testing Black Box Testing	22
6.1.6	Black Box Testing Acceptance Testing	22
6.1.7	Acceptance Testing	22
6.2	Input Design and Output Design	23
7	CONSLUSION	25
	REFERENCES	25
	APPENDIX	29
A.	SOURCE CODE	29
B.	SCREENSHOTS	38
C.	RESEARCH PAPER	47
D.	CERTIFICATE	56

LIST OF FIGURES

FIGURE NO .	FIGURE NAME	Page No.
5.1	Architecture Diagram	13
5.2	Data Flow Diagram	14
5.3	Use Case Diagram	15
5.3.1	Class Diagram	16
5.3.2	Activity Diagram	17
5.3.3	Sequence Diagram	18

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
AES	Advanced Encryption Standard
ECC	Elliptic Curve Encryption
RSA	Rivest, Shamir, Adleman
VPN	Virtual Private Network

CHAPTER 1

INTRODUCTION

A hybrid cryptography scheme is offered as a means of safely storing files on the cloud. Because the data are encrypted and only authorised users may access them, users can securely save their files in online cloud storage with this method. The files that the user uploads to the cloud will be securely stored there after being encrypted with a user-specific key, as seen in the above image. Security issues have grown more pressing as the digital environment continues to evolve. These dangers primarily target cloud computing vendors because of an organization's overall lack of insight into data access and movement. Regardless of where client information is stored, organisations handling data may face significant governance and compliance issues if they don't take aggressive steps to strengthen their cloud security. Regardless of the size of your business, cloud security has to be a hot topic of discussion. Cloud infrastructure supports almost every aspect of modern computing and is present across all industries and verticals. The proposed paradigm has the ability to meet cloud data centre security needs. Data security and privacy protection meet the fundamental challenge of keeping sensitive data apart from access control. Cryptography is used to convert the original data into an unintelligible representation. There are two forms of cryptography: public key cryptography and symmetric key cryptography. Using keys, this approach transforms data into a format that is not readable. As a result, data on cloud servers can only be accessed by authorised individuals. Information that has been ciphered is accessible to everybody. Compared to other symmetric algorithms, blowfish provides the quickest encryption and decryption rates when used to encrypt file slices. The principles of splitting and merging are added to satisfy the idea of data security. The hybrid approach boosts the security of the remote server in a cloud environment, hence increasing user confidence in cloud providers. Data security is a primary concern. This technology may be used in the banking and corporate sectors to securely transmit secret data.

CHAPTER 2

LITERATURE SURVEY

Performance Analysis Systems constitute the foundation of every mentorship programme. A person will undoubtedly advance with the support and direction of statistical data, which will also contribute to the success of the company that employs him. To examine comparable current methods that are used to analyse student performance, a background investigation is conducted. We first get a knowledge of three current systems that bear similarities to the proposed system before beginning our literature review.

[11]"Enhanced Security for Cloud Storage Using Hybrid Cryptography" by S.S.S. Sharma and S. Kumari [2021]

This paper proposes a hybrid cryptography-based enhanced cloud storage security paradigm that combines the AES and RSA encryption and decryption techniques. The proposed paradigm also includes a secure key management method that ensures the confidentiality and integrity of data kept in cloud storage. The study evaluates the proposed architecture's performance and shows how safe cloud storage it can be through testing and simulations.

[7]"A Hybrid Encryption Scheme for Cloud Storage Security" by M.A. Khan, M.A. Rahman, and M. Islam [2020]

For the security of cloud storage, this paper proposes a hybrid encryption technique that combines symmetric and asymmetric encryption. The suggested method makes use of the RSA and AES algorithms for decryption and encryption, respectively. The research evaluates the performance and security of the proposed system through simulations and testing, demonstrating how effectively it may allay security worries in cloud storage.

[5]"Secure File Storage and Sharing in Cloud Computing Using Hybrid Cryptography" by P. Kumar and P. Singh [2021]

This research proposes a hybrid cryptography strategy for file sharing and storage in cloud computing. The model uses the encryption techniques of RSA and AES to ensure the integrity and secrecy of the data. The study demonstrates the effectiveness and security of the recommended strategy using tests and simulations.

^[2] **Rohit Barvekar, Shrajal Behere, Yash Pounikar, and Ms. Anushka Gulhane, "Secure File Storage and Sharing in Cloud Computing using Hybrid Cryptography" (2021)**

The recommended security measures that will prevent sensitive data from being used unlawfully will make the system more trustworthy. Keys will be used in the proposed method for both encryption and decryption.

The work ^[5] **"Secure File Storage and Sharing in Cloud Computing using Hybrid Cryptography" by A. Sharma and K. Sharma. [2020]**

This research proposes a hybrid cryptography-based approach for cloud computing and secure file storage that combines AES and ECC encryption. The recommended method ensures confidentiality, integrity, and availability by utilising encryption and decryption methods. The study demonstrates the effectiveness and security of the recommended strategy using tests and simulations.

^[7] **"File Storage and Sharing in Cloud utilizing Hybrid Cryptography" by A. Shashank and K. mishra. [2020]**

This research proposes a hybrid cryptography strategy for file sharing and storage in cloud computing. The model uses the encryption techniques of RSA and AES to ensure the integrity and secrecy of the data. The study demonstrates the effectiveness and security of the recommended strategy using tests and simulations.

2.1 Existing System

Within the current framework, cloud computing offers a number of advantages, such as low prices and convenient access to information on the Internet. Ensuring cloud computing security may be a crucial factor for the cloud computing environment as users commonly save sensitive information with cloud storage providers, yet these providers may not be trusted. As such, securely transmitting data from an unreliable cloud while ensuring data integrity remains a challenging issue.

Disadvantage

- Digital document creation and analysis were not completed.
- Data Loss: This approach carries the risk of both document and data loss.

2.2 Proposed System

The recommended system describes the proposed system, which offers a method for securely storing information in the cloud using a hybrid cryptography algorithm. Thanks to this technology, users may safely store data in online cloud storage since the files are encrypted and only authorised users can access them. As seen in the above graphic, the data that the user uploads to the cloud will be safely kept there after being encrypted with a user-specific key.

Advantage

- To use hybrid cryptography to provide a secure cloud storage platform for files.
- Data Confidentiality: fully shielded from anyone without special access
- To use hybrid cryptography to shield data from crypto assaults.
- The process of cryptography transforms raw data into ciphertext.
- To provide an example of how to encrypt and decode a file and obtain the key.

2.3 System Specification

Hardware Requirements

Hard disk	:	1 TB
RAM	:	4 GB
Processor	:	Core i3
Monitor	:	15''Color Monitor

Software Requirements

Front-End	:	HTML, CSS, and JS
Back end	:	Python Flask, SQLite
Operating System	:	Windows 10
IDE	:	Visual studio code

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 SOFTWARE SPECIFICATION

Front end – HTML, CSS, JS

- HTML provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.
- CSS is used to control presentation, formatting, and layout.
- JavaScript is used to control the behavior of different elements.

Now, let's go over each one individually to help you understand the roles each plays on a website and then we'll cover how they fit together. Let's start with good of HTML.

HTML

HTML is at the core of every web page, regardless the complexity of a site or number of technologies involved. It's an essential skill for any web professional. It's the starting point for anyone learning how to create content for the web. And, luckily for us, it's surprisingly easy to learn.

Markup languages work in the same way as you just did when you labeled those content types, except they use code to do it -- specifically, they use HTML tags, also known as "elements." These tags have pretty intuitive names: Header tags, paragraph tags, image tags, and so on.

Every web page is made up of a bunch of these HTML tags denoting each type of content on the page. Each type of content on the page is "wrapped" in, i.e. surrounded by, HTML tags.

For example, the words you're reading right now are part of a paragraph. If I were coding this web page from scratch (instead of using the WYSIWG editor in HubSpot's COS), I would have started this paragraph with an opening paragraph tag: `<p>`. The "tag" part is denoted by open brackets, and the letter "p" tells the computer that we're opening a paragraph instead of some other type of content.

Once a tag has been opened, all of the content that follows is assumed to be part of that tag until you "close" the tag. When the paragraph ends, I'd put a closing paragraph tag: `</p>`.

Notice that closing tags look exactly the same as opening tags, except there is a forward slash after the left angle bracket. Here's an example:

```
<p>This is a paragraph.</p>
```

Using HTML, you can add headings, format paragraphs, control line breaks, make lists, emphasize text, create special characters, insert images, create links, build tables, control some styling, and much more.

To learn more about coding in HTML, I recommend checking out our guide to basic HTML, and using the free classes and resources on codecademy -- but for now, let's move on to CSS.

CSS

CSS stands for Cascading Style Sheets. This programming language dictates how the HTML elements of a website should actually appear on the frontend of the page.

If HTML is the drywall, CSS is the paint.

Whereas HTML was the basic structure of your website, CSS is what gives your entire website its style. Those slick colors, interesting fonts, and background images? All thanks to CSS. This language affects the entire mood and tone of a web page, making it an incredibly powerful tool -- and an important skill for web developers to learn. It's also what allows websites to adapt to different screen sizes and device types.

To show you what CSS does to a website, look at the following two screenshots. The first screenshot is my colleague's blog post, but shown in Basic HTML, and the second screenshot is that same blog post with HTML and CSS.

JavaScript

JavaScript is a more complicated language than HTML or CSS, and it wasn't released in beta form until 1995. Nowadays, JavaScript is supported by all modern web browsers and is used on almost every site on the web for more powerful and complex functionality.

JavaScript is particularly useful for assigning new identities to existing website elements, according to the decisions the user makes while visiting the page. For example, let's say you're building a landing page with a form you'd like to generate leads from by capturing information about a website visitor. You might have a "string" of JavaScript dedicated to the user's first name. That string might look something like this:

Then, after the website visitor enters his or her first name -- and any other information you require on the landing page -- and submits the form, this action updates the identity of the initially undefined "Firstname" element in your code. Here's how you might thank your website visitor by name in JavaScript:

```
para.textContent = 'Thanks, ' + First name + '! You can now download your ebook.'
```

In the string of JavaScript above, the "First name" element has been assigned the first name of the website visitor, and will therefore produce his or her actual first name on the frontend of the webpage.

3.2 Back end

Python

Python is a powerful and globally used programming language. Python is often used in as a scripting language. JavaScript embedded in a webpage can be used to control how a web browser such as Firefox displays web content, so JavaScript is a good example of a scripting language. Python can be used as a scripting language for various applications, and is ranked in the top 5-10 worldwide in terms of popularity. Python is fun to use. In fact, the origin of the name comes from the television comedy series Monty Python's Flying Circus.

Flask

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where preexisting third-party libraries provide common function

SQLite

SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI (American National Standards Institute) standard language, but there are many different versions of the SQL language.

What is SQL?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as –

- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

Why SQL?

SQL is widely popular because it offers the following advantages –

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows embedding within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.

CHAPTER 4

SOFTWARE DEVELOPMENT

4.1 METHODOLOGY

Advanced Encryption Standard (AES) with RSA algorithm

To get the security of RSA with the speed of AES, combine symmetric encryption with RSA encryption in this approach. Usually, to do this, a temporary AES key, also known as a session key, is generated and secured using RSA encryption. There are other symmetric encryption techniques than AES.

The Advanced Encryption Standard, or AES, is widely used globally. It belongs to a group of encryption techniques known as "symmetric" encryption. In other words, data is encrypted and decrypted using the same secret, which is an encryption key. Probably the most used encryption technique for safeguarding data when it's at rest is AES encryption. It is utilised in storage encryption, database encryption, self-encrypting disc drives, and other applications. It is a global standard that has existed since roughly 2002. Basically, data and the secret encryption key are entered into software that carries out AES encryption, and the result is encrypted data. When you wish to utilise that data, you just enter the encrypted data into the programme together with the same encryption key, and the original data appears that you may use.

We'll talk about other symmetric key encryption techniques in more detail below. The three people who invented the encryption technique—Adi Shamir, Leonard Adleman, and Ron Rivest—are honoured in the name of RSA encryption. RSA is a member of the "asymmetric" encryption class of encryption techniques. Due to the usage of two linked secrets, or keys, for encryption, the term "asymmetric" was born. The terms "public key" and "private key" refer to the two types of keys. The relationship between the keys is that only the corresponding private key may be used to decode data encrypted with the public key. Additionally, the opposite is also true: only the corresponding public key may be used to decrypt data that has been encrypted using the private key. Factorization and very huge prime integers are used in this fascinating math problem. When you have two physically distinct endpoints, you often

utilise RSA keys. RSA encryption is frequently utilised in VPN connections, web browser connections to your favourite websites, and several more applications. Every day, we employ asymmetric encryption.

4.2 MODULES DESCRIPTION

4.2.1 Data Owner (DO): The owner uploads the data to a cloud server. The file is divided into octet segments, and utilising multithreading, each segment is encoded concurrently. The encoded file is kept on a cloud server. The cover picture stores the encryption keys. The multiuser environment is known as cloud computing.

4.2.2 Data User (DU): File request from a Cloud User. Upon requesting the file, the user also receives a key via email that includes important details. The file is decoded using a reverse technique.

4.2.3 Admin: After entering the home page, admin acts as the owner of the application and can activate and deactivate users as well as view all uploaded file details and request details. If the username and password are entered correctly, only admin can access the home page; if the details entered are incorrect, admin cannot access the home page.

4.2.4 Cloud: With all registered users' information, owners' uploaded file information, and user-downloaded information, the admin of the cloud module may manage it.

CHAPTER 5

SYSTEM DESIGN AND ANALYSIS

5.1 Architecture Diagram

An architecture diagram is a graphic representation that shows how various software system components will be physically implemented. It displays the overall architecture of the software system along with the relationships, constraints, and divisions among its many components.

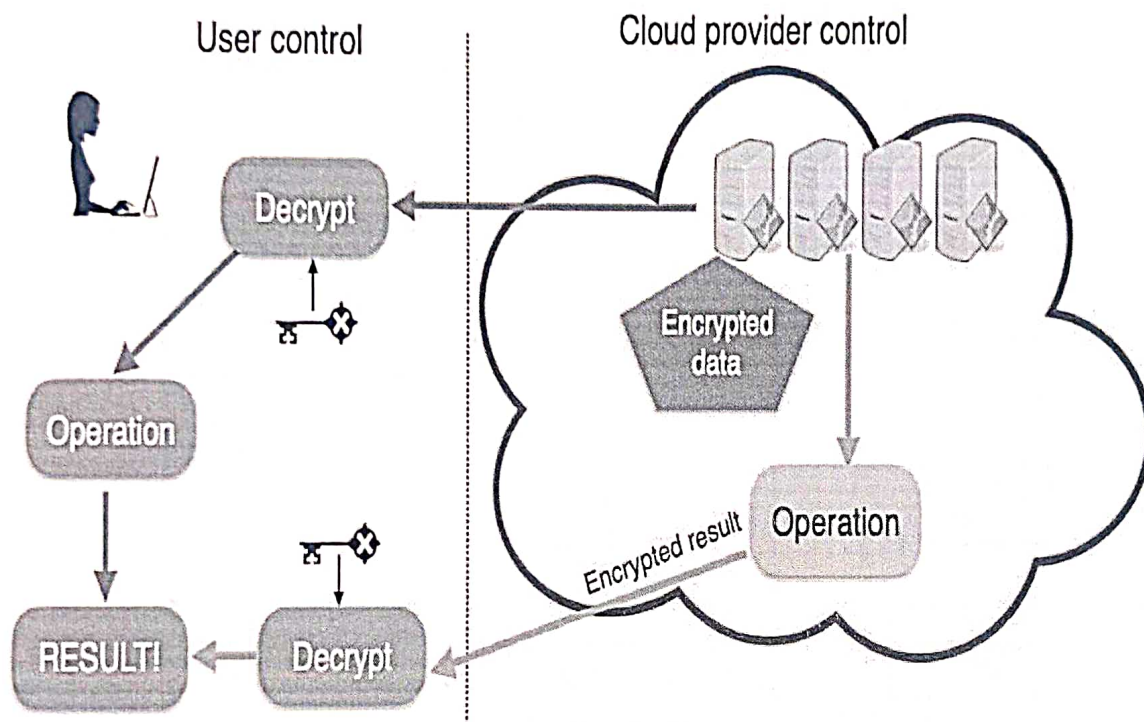


Fig 5.1 Architecture Diagram

5.2 Data Flow Diagram

A data-flow diagram (DFD) is a visual aid for illustrating how data moves through a system or process. Information on each entity's inputs and outputs as well as the process itself are also provided by the DFD. A data-flow diagram lacks decision rules, loops, and control flow. A flowchart can be used to illustrate certain operations based on the data.

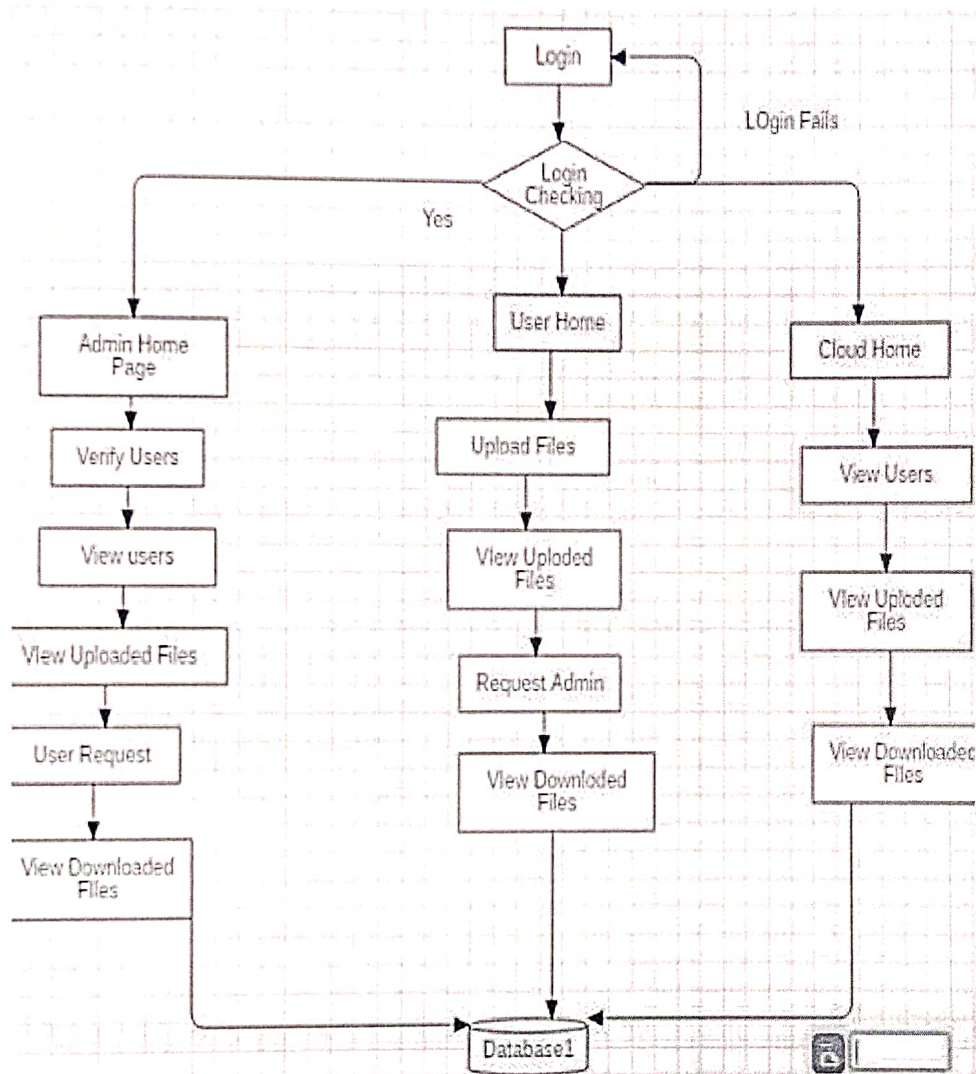


Fig 5.2 Data Flow Diagram.

5.3 UML Diagrams

Use Case Diagram

A use case diagram is a visual representation of how a system's components interact with one another. In system analysis, a use case diagram is a technique for organising, defining, and identifying system needs. the actor, who are often those whose duties within the system are established.

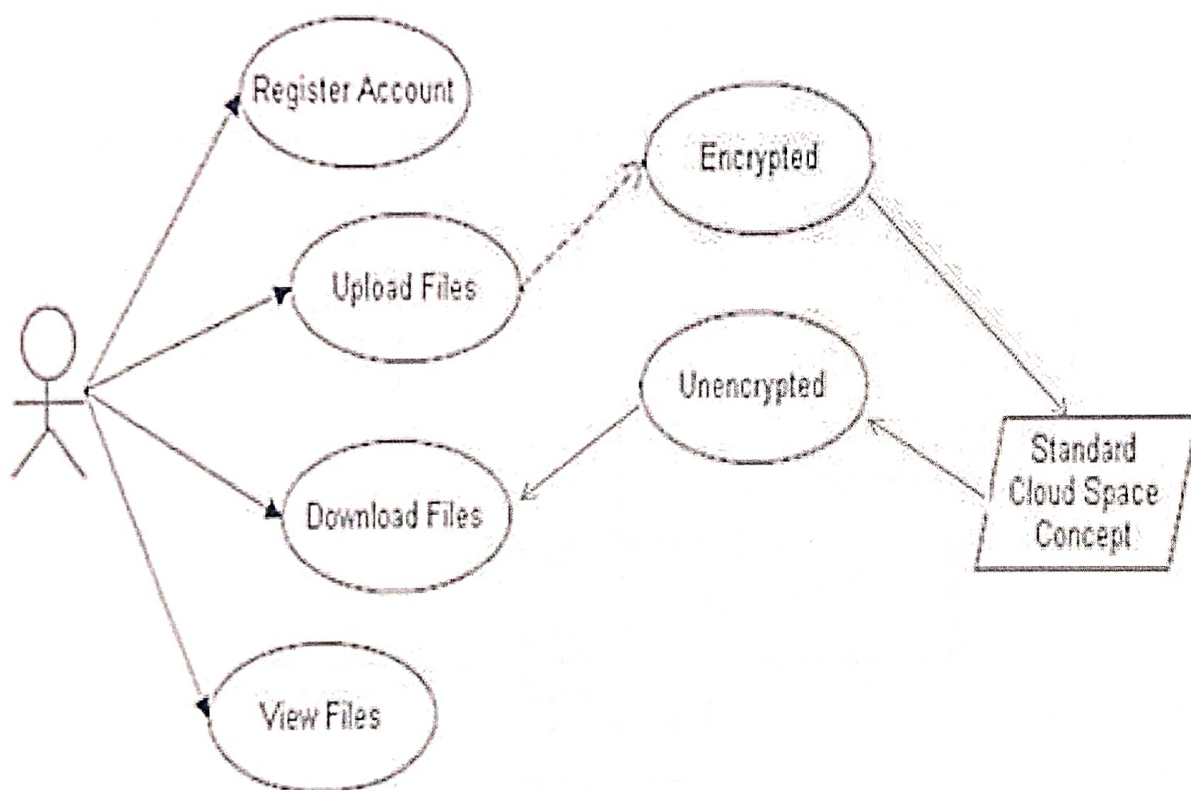


Fig 5.3 Use Case Diagram

Class Diagram

A class diagram is a kind of static structural diagram that illustrates a system's classes, properties, methods, and relationships between objects to depict the structure of the system.

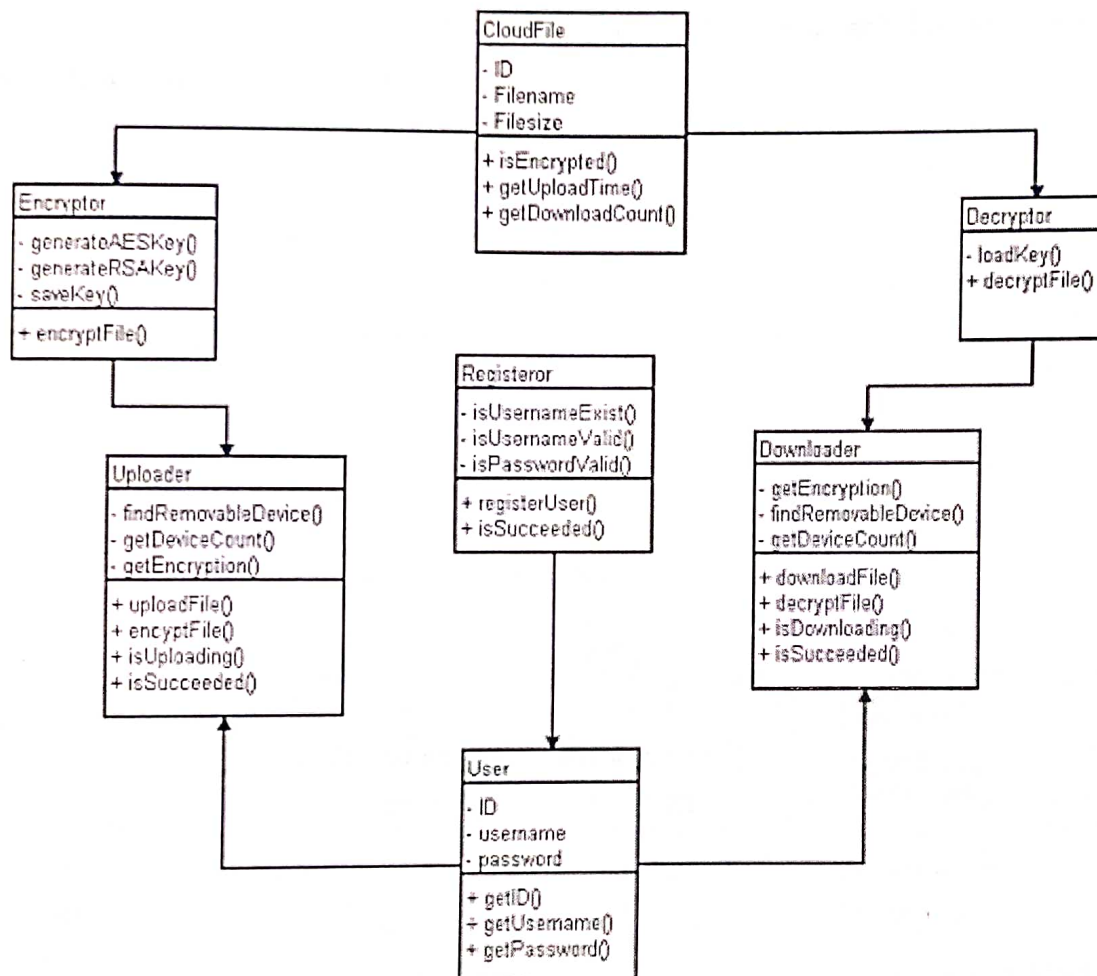


Fig 5.3.1 Class Diagram.

Activity Diagram

An additional crucial UML diagram for describing the system's dynamic elements is the activity diagram. An activity diagram is essentially a flowchart used to show how an activity flows from one to the next. One may refer to the action as a system operation. One action to another is brought into the control flow. This flow may occur concurrently, forked, or sequentially. Activity diagrams include a variety of components, including join, fork, and others, to handle various forms of flow control.

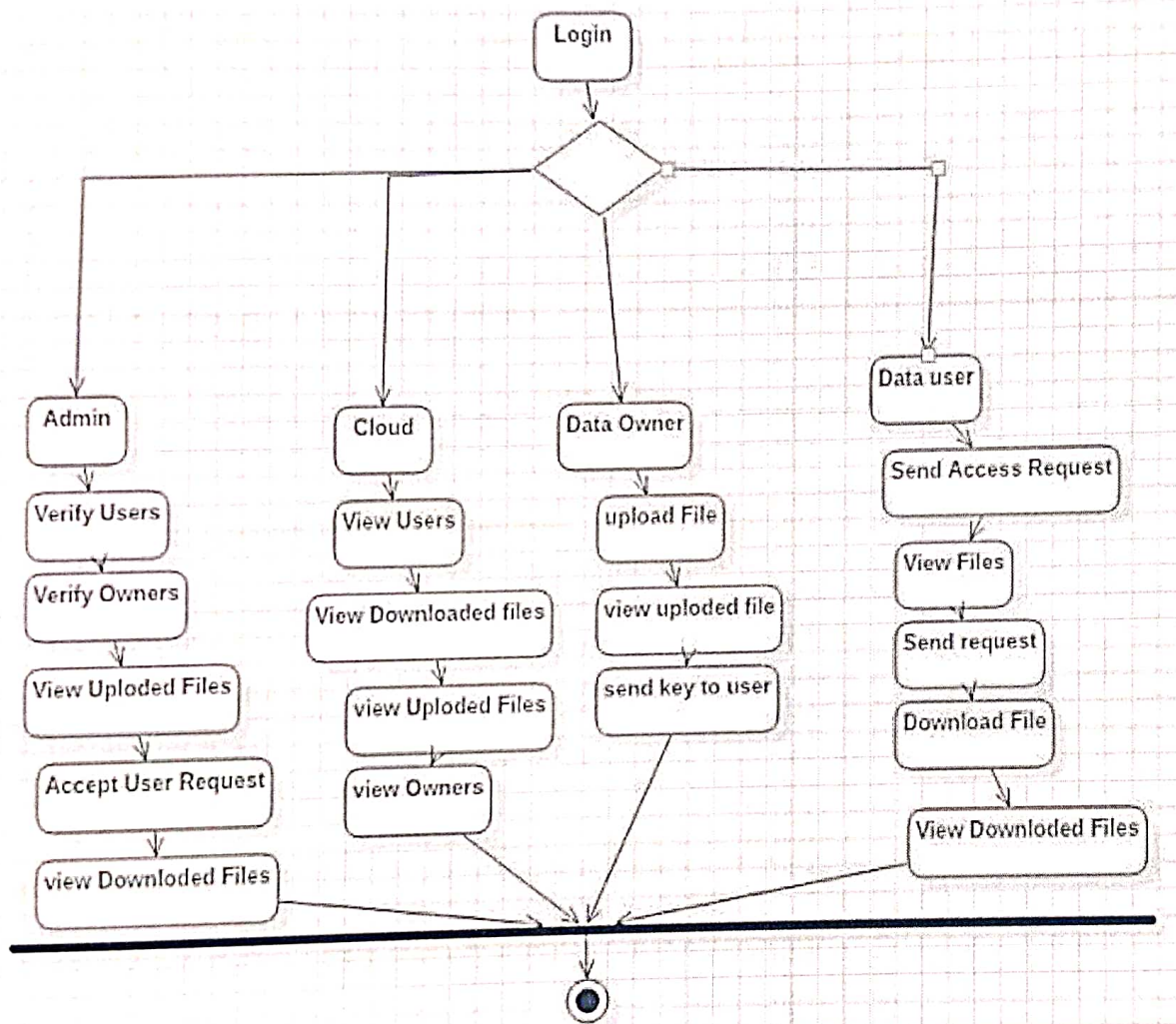


Fig 5.3.2 Activity Diagram.

Sequence Diagram

An example of an interaction diagram is a sequence diagram, which shows the relationship and sequence in which a set of items operate together. Software engineers and business experts use these diagrams to record current processes or to comprehend needs for new systems.

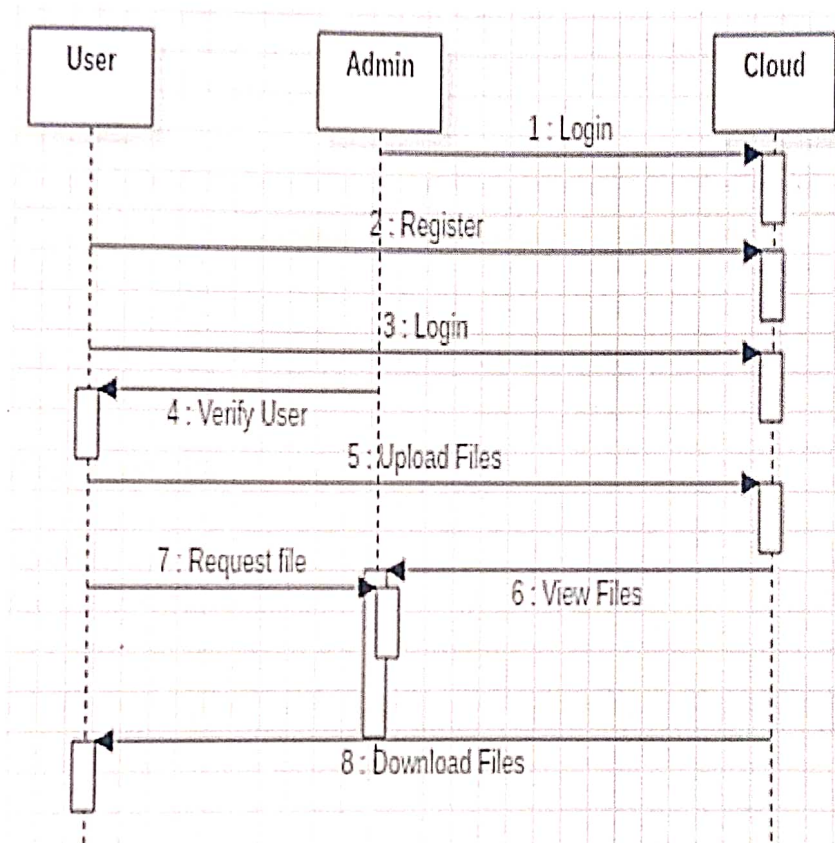


Fig 5.3.3 Sequence Diagram.

5.4 Data Base Design

```
CREATE TABLE users (  
  id INTEGER PRIMARY KEY  
  AUTOINCREMENT,  
  name TEXT,  
  email TEXT,  
  password TEXT,  
  created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
)
```

```
CREATE TABLE files (  
  id INTEGER PRIMARY KEY  
  AUTOINCREMENT,  
  title TEXT,  
  data TEXT,  
  type TEXT,  
  password TEXT  
  user INTEGER,  
  created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
)
```

CHAPTER 6

SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

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

TYPES OF TESTS:

Testing is the process of trying to discover every conceivable fault or weakness in a work product. The different type of testing is given below:

6.1.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.1.2 INTEGRATION TESTING:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.1.3 FUNCTIONAL TEST:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/ Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4 PERFORMANCE TESTING:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.1.5 WHITE BOX TESTING:

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

6.1.6 BLACK BOX TESTING:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1.7 ACCEPTANCE TESTING:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered

6.2 INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:'

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action. Confirm an action.

CHAPTER 7

CONCLUSION

"The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers. Cloud provide Vast storage to user also organizations like Hospital, MNCs, Military uses cloud to store, manage their data which is very convenient to use where ever and whenever user wants to access. But anyone can hack the files or document. To Provide High Security and Privacy for the document which stored in the cloud the project helps user.

SUMMARY

This aims to provide users with some unique tool to Share or store Files Securely and keep the file Private and Safeguard with Hackers. Cloud computing is helpful since it's affordable and makes knowledge easily accessible. Numerous benefits of cloud computing include minimal costs and online information accessibility. Considering that customers frequently save sensitive data with cloud storage providers that may not be trustworthy ensuring cloud computing security may be a key consideration for the cloud computing ecosystem. Therefore, maintaining data from an untrusted cloud while exchanging it securely continues to be a difficult problem. By using AES and the RSA algorithm to store data over a single cloud, my method guarantees the security and privacy of important customer information.

CONCLUSION

Many large staggered organisations have moved to storing and sharing their information on the cloud because to its many benefits. The first step in this effort is to raise awareness of the serious security risks that owners of data face while sharing it on the cloud. The most popular information sharing schemes are then quickly reviewed to identify any shortcomings in each of them. Divides a document containing information into many sections based on the benefits to the client and the material's impact. Then, each information file portion is common, depending on the benefits of the information client. Assuming that the DBDH presumption is true, we formally show that is safe against adaptively selected plaintext attack. This system's primary goal is to safely store and retrieve data on the cloud that is under the exclusive control of the data owner. Data security concerns with cloud storage are resolved through the use of steganography and cryptography methods. In order to secure data, the AES and RSA algorithms are used. Key data is securely preserved with the use of LSB technology (Steganography). Using the multithreading technology, the encryption and decryption procedure takes less time. We have achieved improved data integrity, high security, low latency, authentication, and secrecy with the aid of the suggested security mechanism.

1. **Accuracy:** Striving for accurate predictions remains paramount. Employing advanced Cloud computing security, refining feature engineering, and continuously updating datasets are essential for improving accuracy.
2. **User Experience:** Prioritizing a seamless and intuitive user experience is crucial for ensuring the usability and adoption of the application. This involves designing clear interfaces, providing relevant information, and possibly integrating feedback mechanisms.
3. **Real-time Updates:** Incorporating real-time data updates allows users to access the most current information, empowering them to make timely decisions and store files securely.
4. **Ethical Considerations:** Upholding ethical standards, particularly regarding data privacy, fairness, and transparency, is non-negotiable. Users must trust that their data is handled responsibly and that are made without bias.

REFERENCES

- [1] Uttam Kumar, Mr. Jay Prakash "Secure File Storage on Cloud Using Hybrid Cryptography Algorithm," International Journal Of Creative Research Thoughts (IJCRT), 2021. 2320-2882 is the ISSN. [Prime Document].
- [2] M. Malarvizhi, J. Angela Jennifa Sujana, and T. Revathi."Secure File Sharing Using Cryptographic Techniques In Cloud" (IEE) 2020
- [3] Punam V. Maitri and Aruna Verma present "Secure File Storage In Cloud Computing Using Hybrid Cryptography Algorithm." In IEEE Wispnet 2020,
- [4] Rashidhagat and Purvi Joshi, "Hybrid Cryptography for Secure File Storage," International Journal of Innovative Research in Technology (IJIRT), 2020.
- [5] Bilal Habib, Bertrand Cambou, Duane Booher, Christopher Philabaum, Public Key Exchange Scheme That Is Addressable (PKA), 2019 IEEE CNS.
- [6]. Tulip Dutta, Amarjyoti Pathak "Secure Data Sharing In Cloud Storage Using Key Aggregation Cryptography," Advanced Networks and Telecommunication System (ANTS) 2019.
- [7]. Bhale Pradeep Kumar Gajendra, Vinay Kumar Singh, and More Sujeet, "Achieving Cloud Security Through Identity-Based Encryption, MD5, and Third Party Auditor," ICCCA2020.
- [8] Shakeeba S. Khan, Professor R.R. Tuteja, "Cryptographic Algorithms and Security in Cloud Computing," 2017.
- [9] J. Jenifer Sanjana, and T.M. Reuka. "To Secure File Sharing Using RSATechniques In Cloud" in International Conference On Green Computing Communication And Electrical Engineering (ICGCCEE) 2021.
- [10] Sugumar, Sarvesh and Vijay Verma present "Secure File Storage on Cloud Computing Utilizing AES Algorithm." IJIRT In 2020.
- [11] Kushdeep Kaur and Kashvi Jain, "Hybrid Cryptography to Secure File Storage," International Conference of Innovative Research in Technology (ICIRT), 2020.

[12] Jumana, Vinston Jacob, Sheena Paarek, Christopher, Public Key Exchange Scheme That Is Addressable (PKA), IEEE CNS.2019.

[13] Gajendra Shenoi Singh, Kishore Agarwal, and Haseena, "On Achieving Cloud Security to Encryption, and Third-Party Auditor," IEE 2020.

[14] Pranav S. Govindh, Kathrin Moses, Christopher Rancy "Cryptographic Algorithms and Security in Cloud Computing," 2018.

[15] Bello SA, Oyedele LO, Akinde OO, Bilal M, Delgado JM, LA Ajayi (2020) "Cloud Computing in Industry: To Store file Securely" 122.1.3441

[16] Putta Bharathi, Gayathri Annam, Jaya Bindu Kandi, Vamsi Krishna Duggana, Anjali T"Secure File Storage Using Hybrid Cryptography" IEEE 2021

[17] Jiang Lai, Swee-Huay Heng , "Secure File Storage on Cloud Computing using Hybrid Cryptography",Journal of Informatics and Web Engineering, 2022.

APPENDIX

A.SOURCE CODE

```
App.py:
from flask import Flask, session, render_template, request, redirect, url_for
app = Flask(__name__)
app.secret_key = '23826b72637tskx86'
import sqlite3
import base64
@app.route('/logout')
def logout():
    session.clear()
    return redirect("/")
@app.route('/', methods = ['POST', 'GET'])
def index():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        conn = sqlite3.connect("database.db")
        conn.row_factory = sqlite3.Row
        cur = conn.cursor()
        cur.execute("select * from users WHERE email=? AND password=?", (email, password))
        rows = cur.fetchall()
        if len(rows) > 0:
            for row in rows:
                session['id'] = row['id']
                session['name'] = row['name']
                session['user'] = 'user'
            return redirect("/home")
        else:
            return redirect("/?err=Email or Password is Wrong!")
        else:
            return render_template('index.html')

@app.route('/signup', methods = ['POST', 'GET'])
def signup():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        password = request.form['password']
        conn = sqlite3.connect("database.db")
        conn.execute('INSERT INTO
        users(name,email,password)VALUES(?,?,?)',(name,email,password))
        conn.commit()
        return redirect("/?msg=Signup Successfully!")
    else:
        return render_template('signup.html')

@app.route('/home')
```

```

defHome():
    if 'user' in session and session['user'] == 'user':
        userid = str(session['id'])
        con = sqlite3.connect("database.db")
        con.row_factory = sqlite3.Row
        cur = con.cursor()
        cur.execute("select * FROM files where user=?", (userid))
        files = cur.fetchall()
        return render_template('user/home.html', files=files, user=session['name'])
    else:
        return redirect('/?err=Illegal Access')

@app.route('/files')
defFiles():
    if 'user' in session and session['user'] == 'user':
        userid = str(session['id'])
        con = sqlite3.connect("database.db")
        con.row_factory = sqlite3.Row
        cur = con.cursor()
        cur.execute("select * FROM files where user=?", (userid))
        files = cur.fetchall()
        return render_template('user/files.html', files=files, user=session['name'])
    else:
        return redirect('/?err=Illegal Access')

@app.route('/files/view')
defFilesView():
    attid = request.args.get('id')
    con = sqlite3.connect("database.db")
    con.row_factory = sqlite3.Row
    cur = con.cursor()
    cur.execute("select * FROM files where id=?", (attid))
    file = cur.fetchall()
    return render_template('password.html', file=file[0])

@app.route('/files/open', methods = ['POST', 'GET'])
defFilesOpen():
    if request.method == 'POST':
        fileid = request.form['id']
        password = request.form['password']
        conn = sqlite3.connect("database.db")
        conn.row_factory = sqlite3.Row
        cur = conn.cursor()
        cur.execute("select * from files WHERE id=? AND password=?", (fileid, password))
        file = cur.fetchall()
        if len(file) > 0:
            return render_template('file.html', type=file[0]['type'], file=file[0]['data'].decode())
        else:
            return redirect("/files/view?err=File Password is Wrong!&id="+fileid)

```

```

<divclass="avatar avatar-online">
<imgsrc="/static/img/avatars/1.png"altclass="w-px-40 h-auto rounded-circle"/>
</div>
</div>
<divclass="flex-grow-1">
<spanclass="fw-semibold d-block">{{user}}</span>
<!--<small class="text-muted">Admin</small> -->
</div>
</div>
</a>
</li>
<!--<li>
<div class="dropdown-divider"></div>
</li> -->
<!--<li>
<a class="dropdown-item" href="#">
<i class="bx bx-user me-2"></i>
<span class="align-middle">My Profile</span>
</a>
</li> -->
<li>
<divclass="dropdown-divider"></div>
</li>
<li>
<aclass="dropdown-item"href="/logout">
<iclass="bx bx-power-off me-2"></i>
<spanclass="align-middle">Log Out</span>
</a>
</li>
</ul>
</li>
<!--/ User -->
</ul>
</div>
</nav>

<!-- / Navbar -->

<!-- Content wrapper -->
<divclass="content-wrapper">
<!-- Content -->

<divclass="container-fluid flex-grow-1 container-p-y">
<!-- Layout Demo -->
<divclass="row">
<divclass="col-4">
<divclass="card p-3">
<h4>Total Files</h4>
<h2>{{files}}</h2>

```



```

</div>
</div>
</div>
<!-- / Layout Demo -->
</div>
<!-- / Content -->
<!-- / Footer -->

<divclass="content-backdrop fade"></div>
</div>
<!-- Content wrapper -->
</div>
<!-- / Layout page -->
</div>

<!-- Overlay -->
<divclass="layout-overlay layout-menu-toggle"></div>
</div>
<!-- / Layout wrapper -->
<!-- Core JS -->
<!-- build:js assets/vendor/js/core.js -->
<scriptsrc="/static/vendor/libs/jquery/jquery.js"></script>
<scriptsrc="/static/vendor/libs/popper/popper.js"></script>
<scriptsrc="/static/vendor/js/bootstrap.js"></script>
<scriptsrc="/static/vendor/libs/perfect-scrollbar/perfect-scrollbar.js"></script>

<scriptsrc="/static/vendor/js/menu.js"></script>
<!-- endbuild -->

<!-- Vendors JS -->

<!-- Main JS -->
<scriptsrc="/static/js/main.js"></script>

<!-- Page JS -->

<!-- Place this tag in your head or just before your close body tag. -->
<scriptasyncdefersrc="https://buttons.github.io/buttons.js"></script>
</body>
</html>

```

User/profile.html:
<!DOCTYPEhtml>

```

<!-- =====
Sneat - Bootstrap 5 HTML Admin Template - Pro | v1.0.0
=====

```

Product Page: <https://themeselection.com/products/sneat-bootstrap-html-admin-template/>
Created by: ThemeSelection
License: You must have a valid license purchased in order to legally use the theme for your project.
Copyright ThemeSelection (<https://themeselection.com>)

```
=====
-->
<!-- beautify ignore:start -->
<html
lang="en"
class="light-style layout-menu-fixed"
dir="ltr"
data-theme="theme-default"
data-assets-path="/static/"
data-template="vertical-menu-template-free"
>
<head>
<metacharset="utf-8"/>
<meta
name="viewport"
content="width=device-width, initial-scale=1.0, user-scalable=no, minimum-scale=1.0,
maximum-scale=1.0"
/>

<title>User</title>

<metaname="description"content=""/>

<!-- Favicon -->
<linkrel="icon" type="image/x-icon" href="/static/img/favicon/favicon.ico"/>

<!-- Fonts -->
<linkrel="preconnect" href="https://fonts.googleapis.com"/>
<linkrel="preconnect" href="https://fonts.gstatic.com" crossorigin/>
<link
href="https://fonts.googleapis.com/css2?family=Public+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;1,300;1,400;1,500;1,600;1,700&display=swap"
rel="stylesheet"
/>

<!-- Icons. Uncomment required icon fonts -->
<linkrel="stylesheet" href="/static/vendor/fonts/boxicons.css"/>

<!-- Core CSS -->
<linkrel="stylesheet" href="/static/vendor/css/core.css" class="template-customizer-core-css"/>
<linkrel="stylesheet" href="/static/vendor/css/theme-default.css" class="template-
```

```

customizer-theme-css"/>
<linkrel="stylesheet"href="/static/css/demo.css"/>

<!-- Vendors CSS -->
<linkrel="stylesheet"href="/static/vendor/libs/perfect-scrollbar/perfect-scrollbar.css"/>

<!-- Page CSS -->

<!-- Helpers -->
<scriptsrc="/static/vendor/js/helpers.js"></script>

<!--! Template customizer & Theme config files MUST be included after core stylesheets
and helpers.js in the <head> section -->
<!--? Config: Mandatory theme config file contain global vars & default theme options,
Set your preferred theme option in this file. -->
<scriptsrc="/static/js/config.js"></script>
</head>
<script>
constqueryString = window.location.search;
consturlParams = newURLSearchParams(queryString);
if(urlParams.get('err')){
document.write("<div id='err' style='position:fixed;bottom:30px; right:30px;background-
color:tomato;padding:10px;border-radius:10px;box-shadow:2px 2px 4px
#aaa;color:white;font-weight:600'">"+urlParams.get('err')+"</div>")
setTimeout(()=>{
document.getElementById("err").style.display="none"
}, 5000)
}
if(urlParams.get('msg')){
document.write("<div id='msg' style='position:fixed;bottom:30px; right:30px;background-
color:green;padding:10px;border-radius:10px;box-shadow:2px 2px 4px
#aaa;color:white;font-weight:600'">"+urlParams.get('msg')+"</div>")
setTimeout(()=>{
document.getElementById("msg").style.display="none"
}, 5000)
}
</script>
<body>
<!-- Layout wrapper -->
<divclass="layout-wrapper layout-content-navbar">
<divclass="layout-container">
<!-- Menu -->

<asideid="layout-menu"class="layout-menu menu-vertical menu bg-menu-theme">
<divclass="app-brand demo">
<a href=""class="app-brand-link ">
<spanclass="h3 text-primary demo menu-text fw-bolder ms-2">User</span>
</a>

```

```

<a href="javascript:void(0);" class="layout-menu-toggle menu-link text-large ms-auto d-block d-xl-none">
<i class="bx bx-chevron-left bx-sm align-middle"></i>
</a>
</div>

<div class="menu-inner-shadow"></div>

<ul class="menu-inner py-1">
<!-- Dashboard -->
<li class="menu-item">
<a href="/home" class="menu-link">
<i class="menu-icon tf-icons bx bx-home-circle"></i>
<div data-i18n="Analytics">Dashboard</div>
</a>
</li>
<li class="menu-item">
<a href="/files" class="menu-link">
<i class="menu-icon tf-icons bx bx-detail"></i>
<div data-i18n="Analytics">Files</div>
</a>
</li>
<li class="menu-item active">
<a href="/profile" class="menu-link">
<i class="menu-icon tf-icons bx bx-user"></i>
<div data-i18n="Analytics">Profile</div>
</a>
</li>
</ul>
</aside>
<!-- Layout container -->
<div class="layout-page">
<!-- Navbar -->

<nav
class="layout-navbar container-fluid navbar navbar-expand-xl navbar-detached align-items-center bg-navbar-theme"
id="layout-navbar"
>

<div class="navbar-nav-right d-flex align-items-center" id="navbar-collapse">
<!-- Search -->
<span class="fw-bolder display-6">Profile</span>
<!-- /Search -->

<ul class="navbar-nav flex-row align-items-center ms-auto">

<!-- User -->
<li class="nav-item navbar-dropdown dropdown-user dropdown">

```



```
<div class="layout-overlay layout-menu-toggle"></div>
</div>
```

```
<div
  class="modal fade"
  id="modalToggle"
  aria-labelledby="modalToggleLabel"
  tabindex="-1"
  style="display: none"
  aria-hidden="true"
>
  <div class="modal-dialog modal-dialog-centered">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="modalToggleLabel">Add Manager Attendance Details</h5>
        <button
          type="button"
          class="btn-close"
          data-bs-dismiss="modal"
          aria-label="Close"
        ></button>
      </div>
      <div class="modal-body">
        <form action="/admin/manager/add" method="post">
          <div class="mb-3">
            <label for="email" class="form-label">Staff</label>
            <select name="user" id="" class="form-select">
              <option value="" selected disabled hidden>Select Staff</option>
              {% for user in users %}
                <option value="{{user['id']}}">{{user['name']}}</option>
              {% endfor %}
            </select>
          </div>
          <div class="mb-3">
            <label for="email" class="form-label">Attendance</label>
            <select name="attendance" class="form-select" id="">
              <option value="" selected disabled hidden>Select Attendance</option>
              <option value="Present">Present</option>
              <option value="Absent">Absent</option>
            </select>
          </div>
          <div class="mb-3">
            <label for="email" class="form-label">Worked Hours</label>
            <input
              type="number"
              class="form-control"
              id="email"
              name="hour"
              placeholder="Enter Worked Hours"
            >
          </div>
        </form>
      </div>
    </div>
  </div>
```

```

autofocus
required
/>
</div>
<divclass="mb-3">
<labelfor="email"class="form-label">Date</label>
<input
type="date"
class="form-control"
id="email"
name="date"
placeholder="Enter date"
autofocus
required
/>
</div>
<divclass="text-end">
<buttonclass="btn btn-primary">Add</button>
</div>
</form>
</div>
</div>
</div>
</div>
<!-- / Layout wrapper -->
<!-- Core JS -->
<!-- build:js assets/vendor/js/core.js -->
<scriptsrc="/static/vendor/libs/jquery/jquery.js"></script>
<scriptsrc="/static/vendor/libs/popper/popper.js"></script>
<scriptsrc="/static/vendor/js/bootstrap.js"></script>
<scriptsrc="/static/vendor/libs/perfect-scrollbar/perfect-scrollbar.js"></script>

<scriptsrc="/static/vendor/js/menu.js"></script>
<!-- endbuild -->

<!-- Vendors JS -->

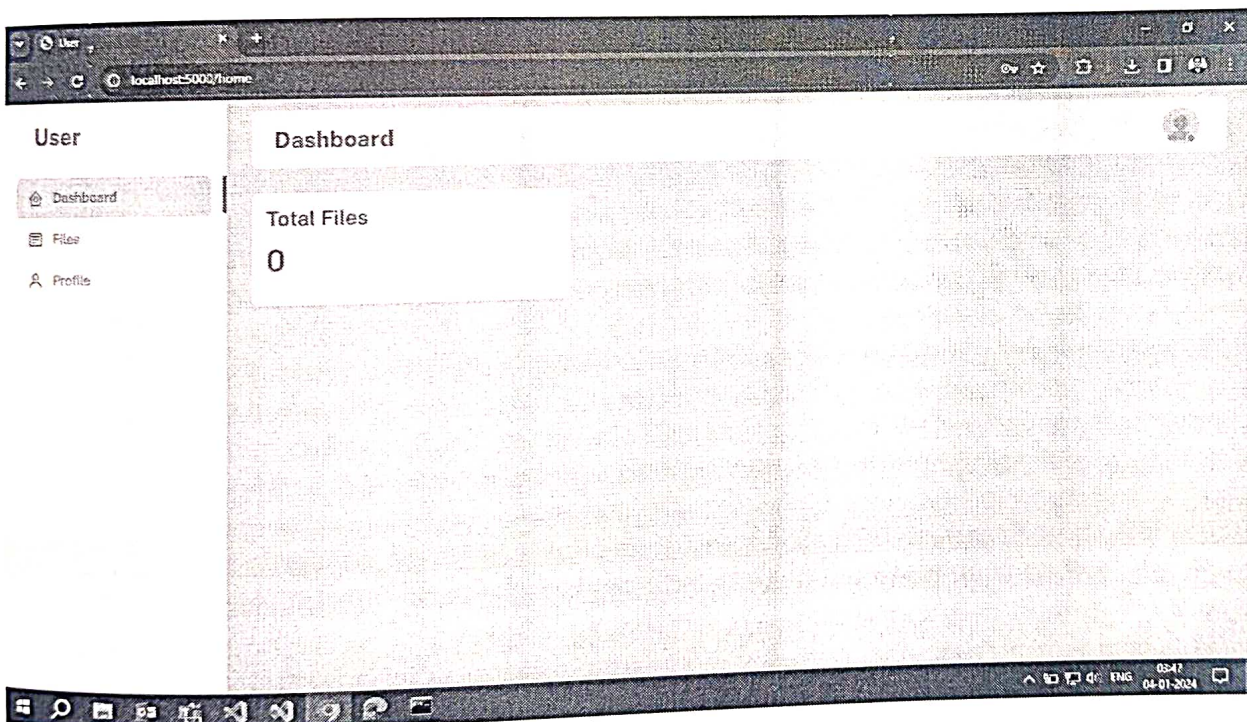
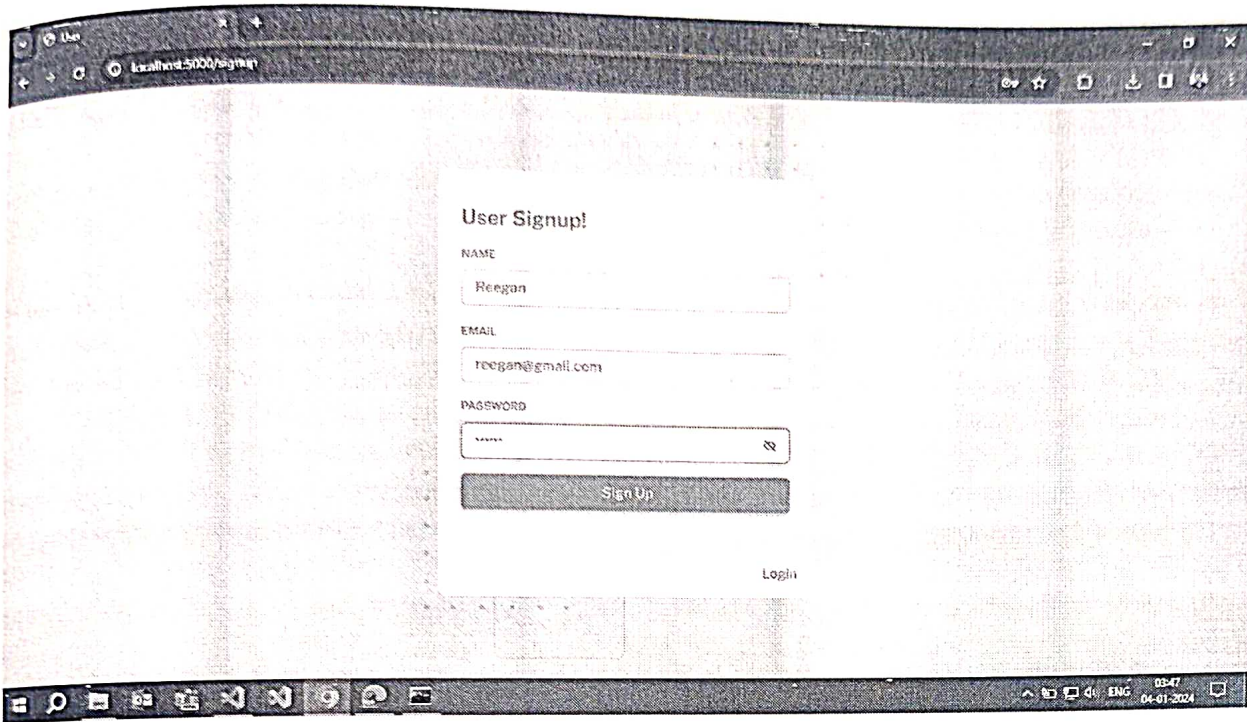
<!-- Main JS -->
<scriptsrc="/static/js/main.js"></script>

<!-- Page JS -->

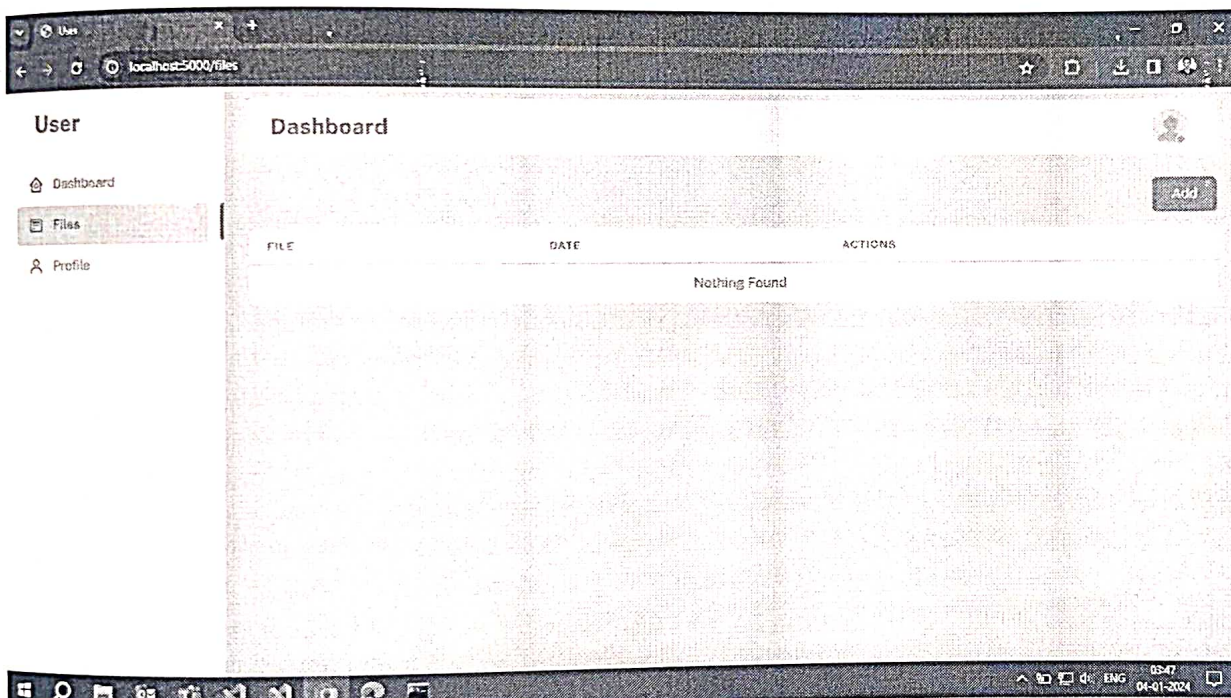
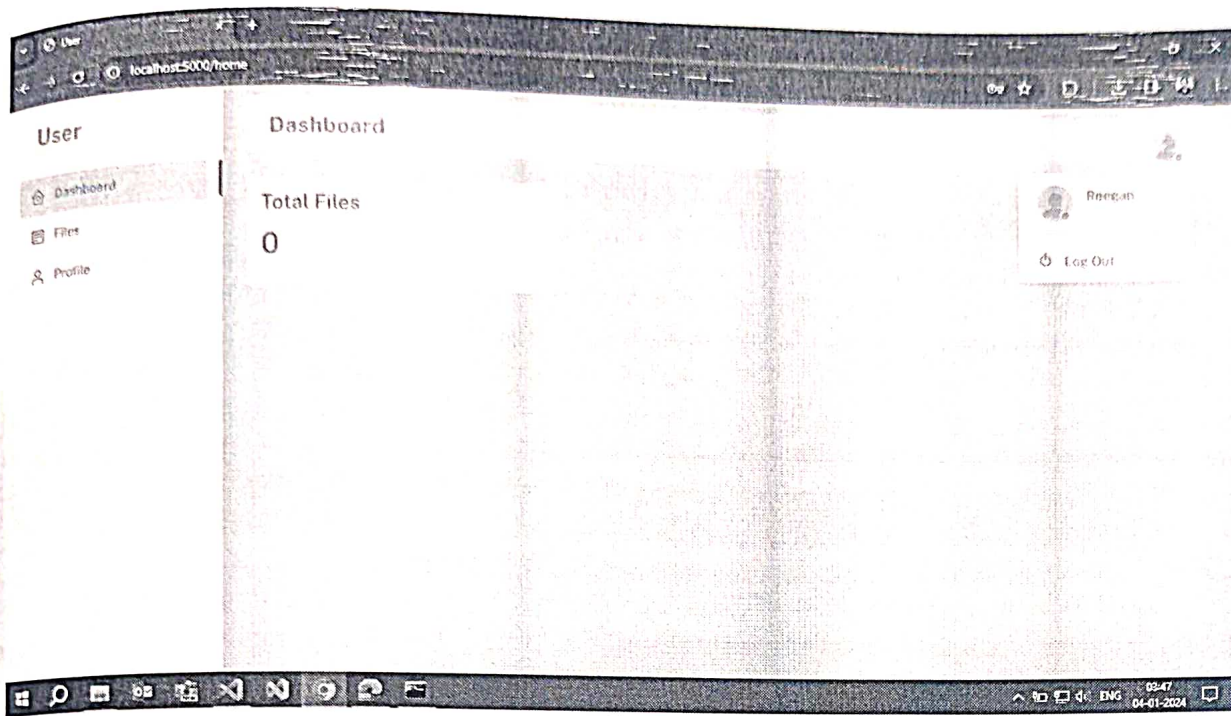
<!-- Place this tag in your head or just before your close body tag. -->
<scriptasyncdefersrc="https://buttons.github.io/buttons.js"></script>
</body>
</html>

```

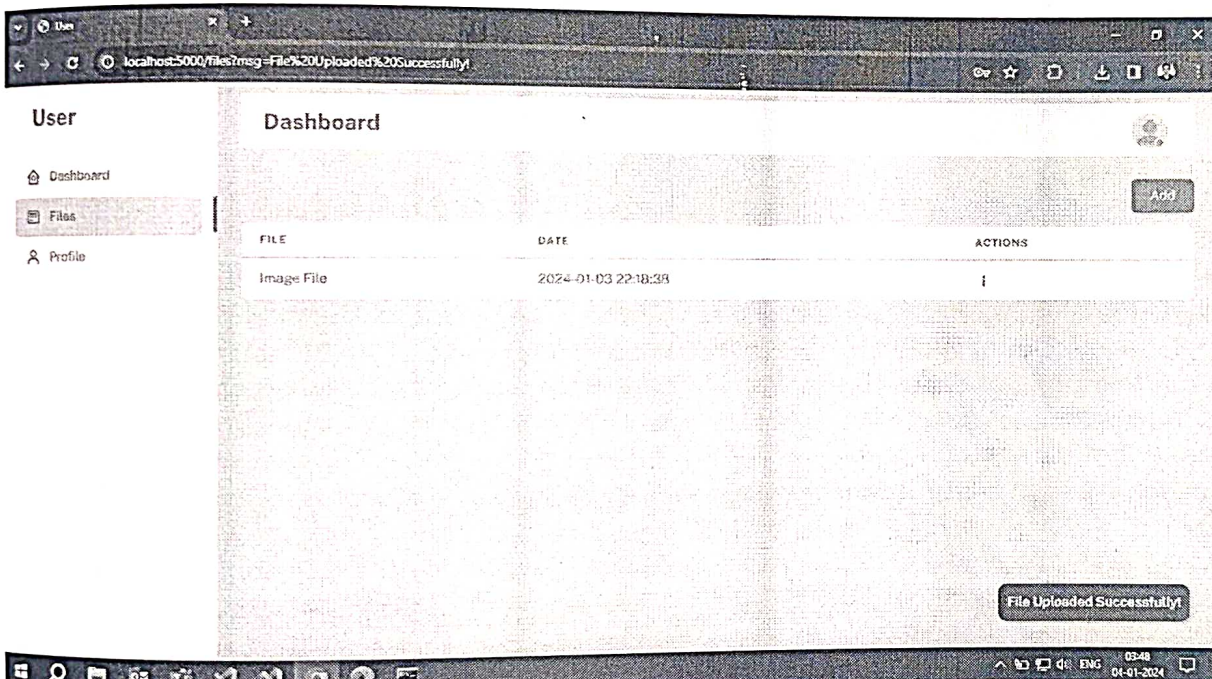
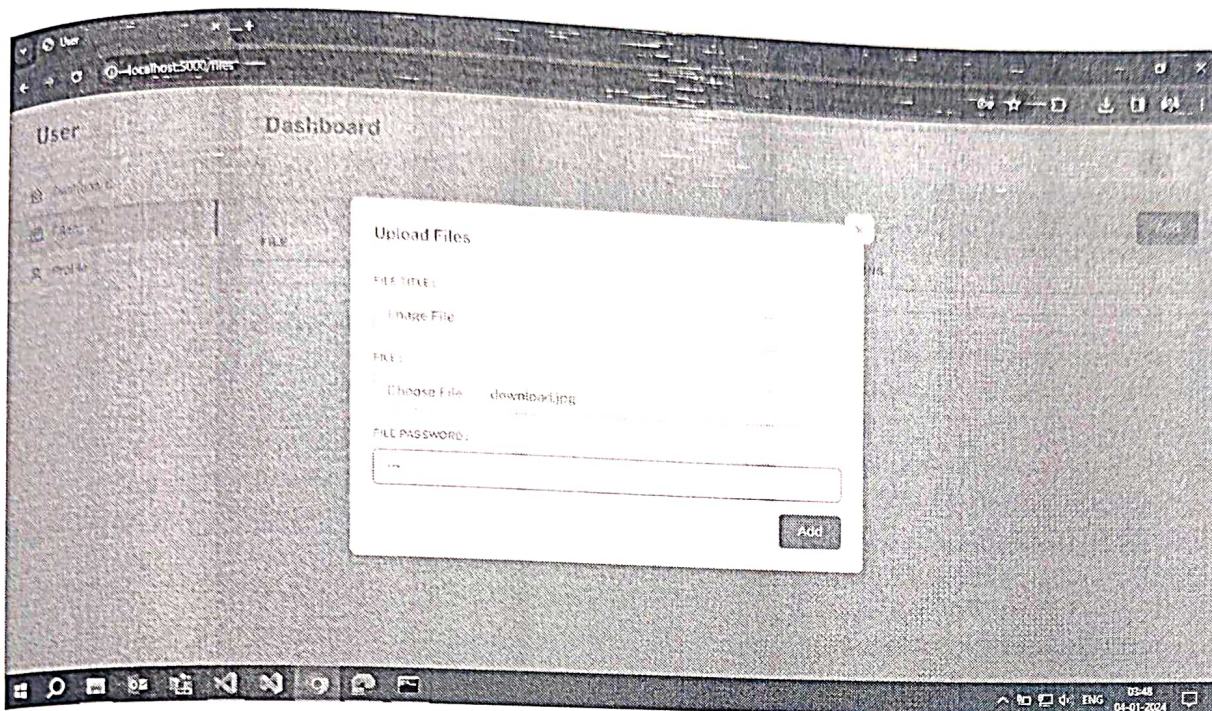
B. SCREEN SHOTS



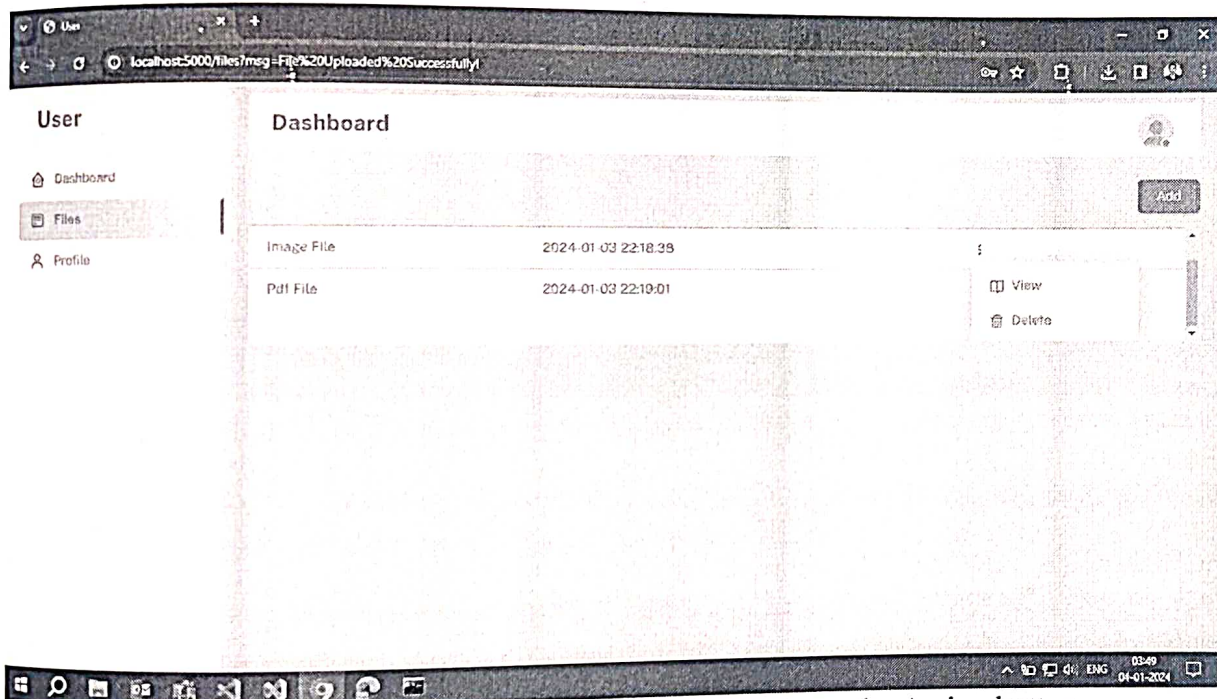
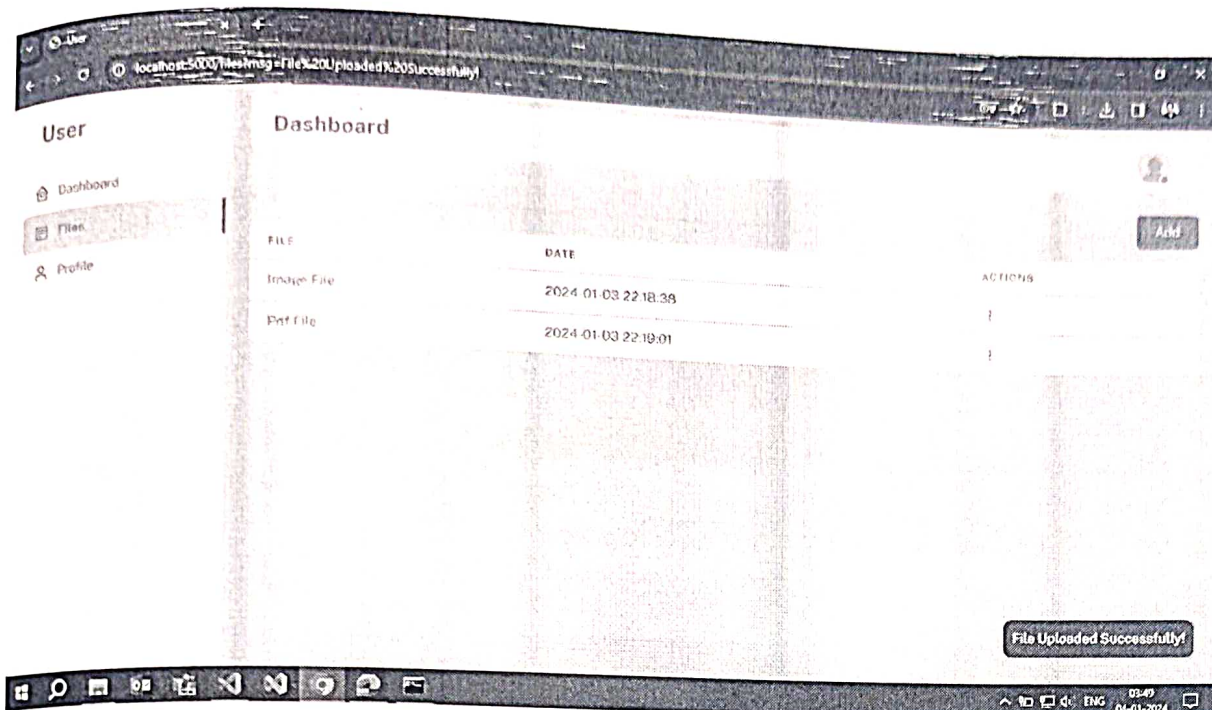
- From the above screenshot, it shows that login page of **USER**. Where it contains 3 types of option: **Dashboard, Files, Profile**.
- In Dashboard, User can see the total Files attach in their login.



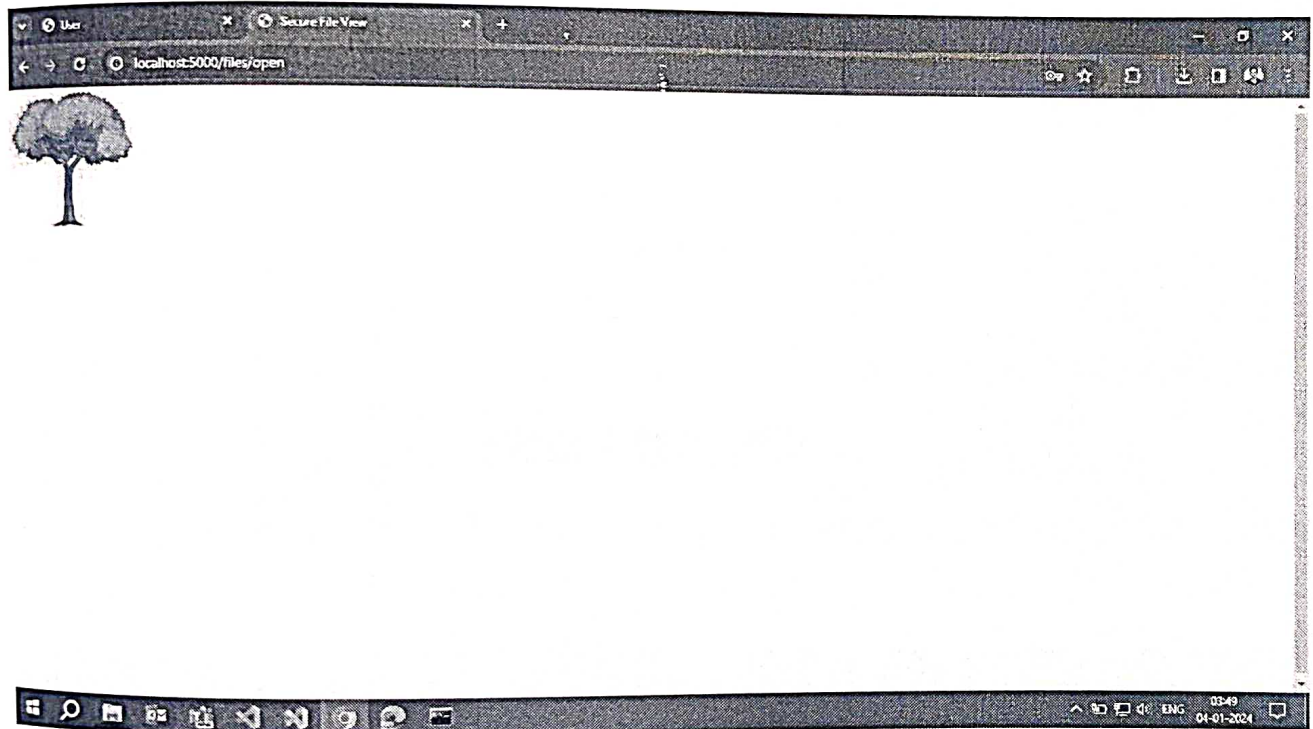
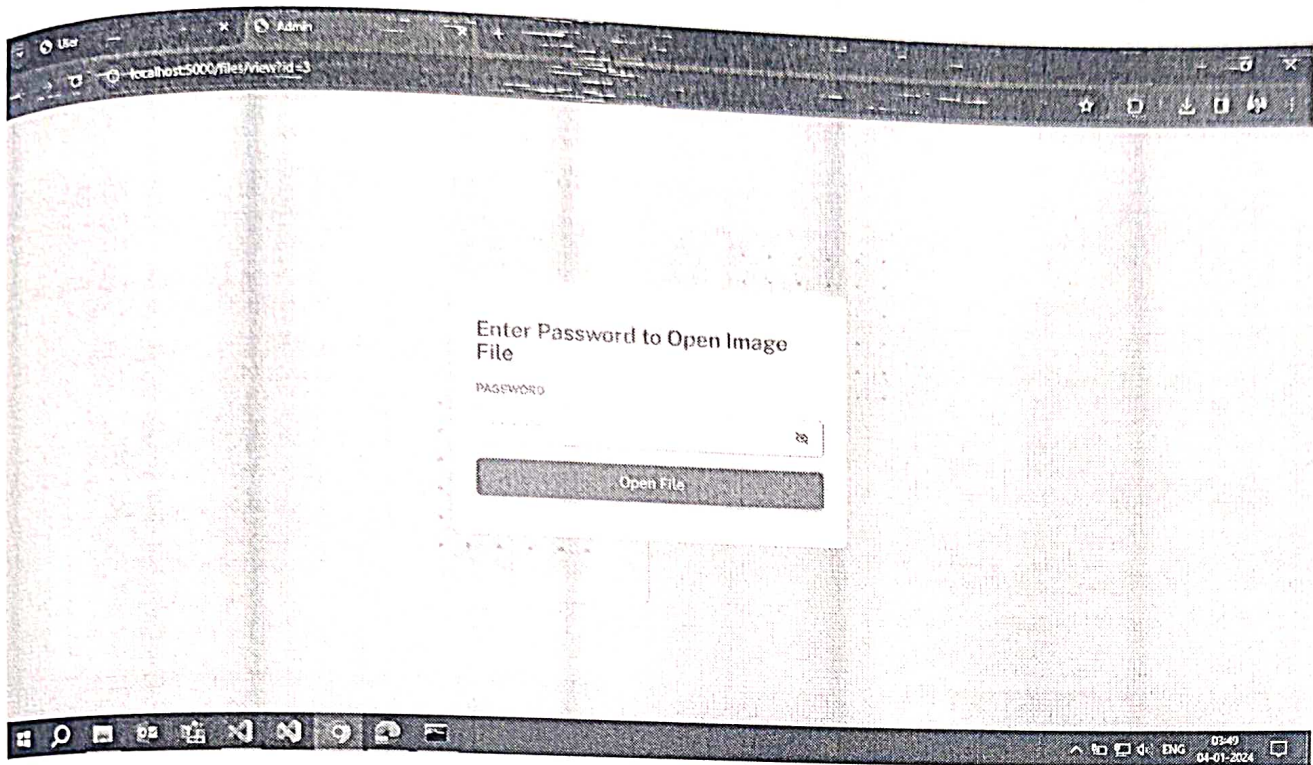
- Also user can logout from the dashboard page which is in the Right most corner.
- From Files option user can see FILES, when it upload and Action.



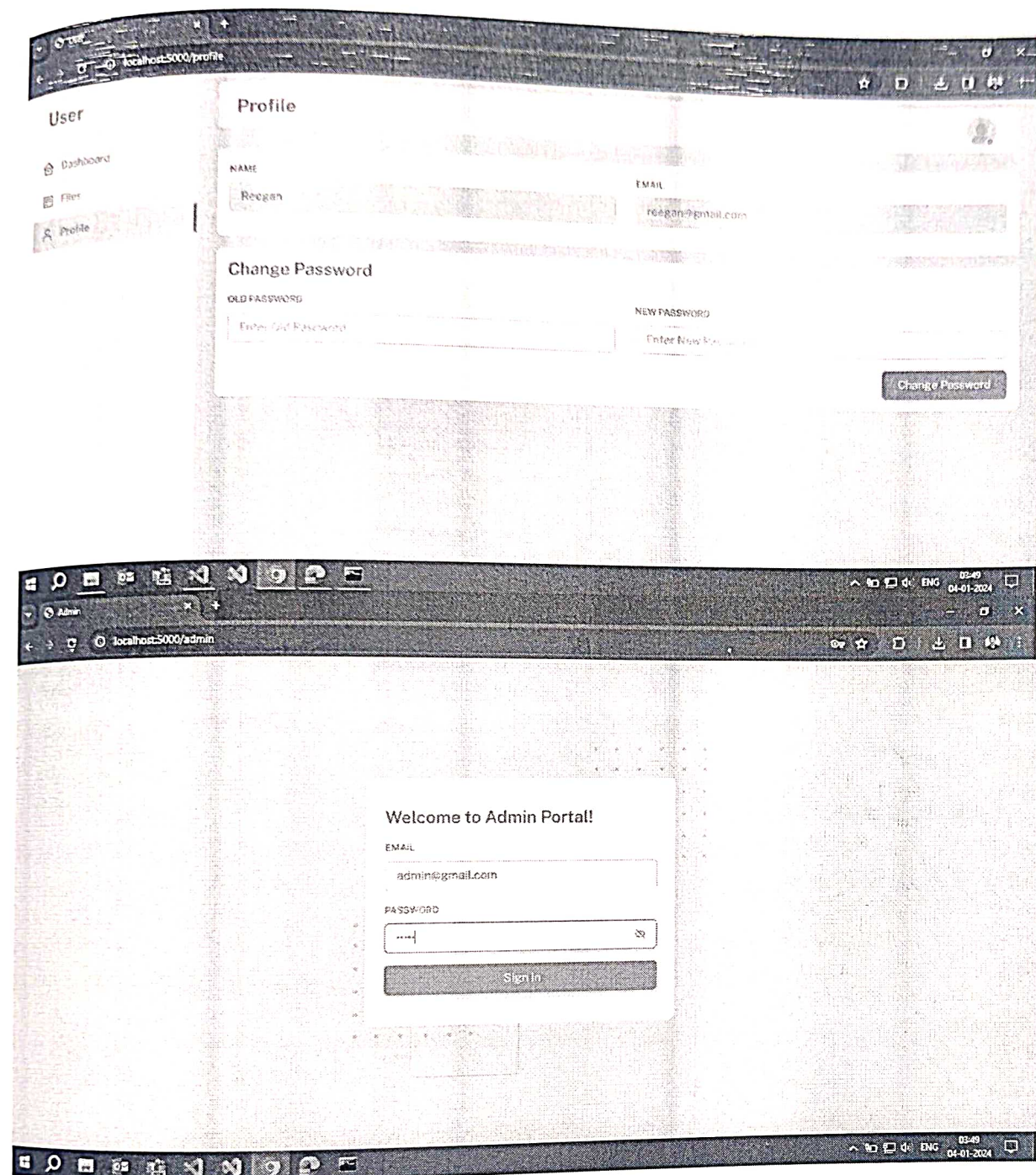
- From files section user can upload any type of file and secure the file with Strong password which the file can only with the password.
- Also, status of the file which uploaded suessfull or not.



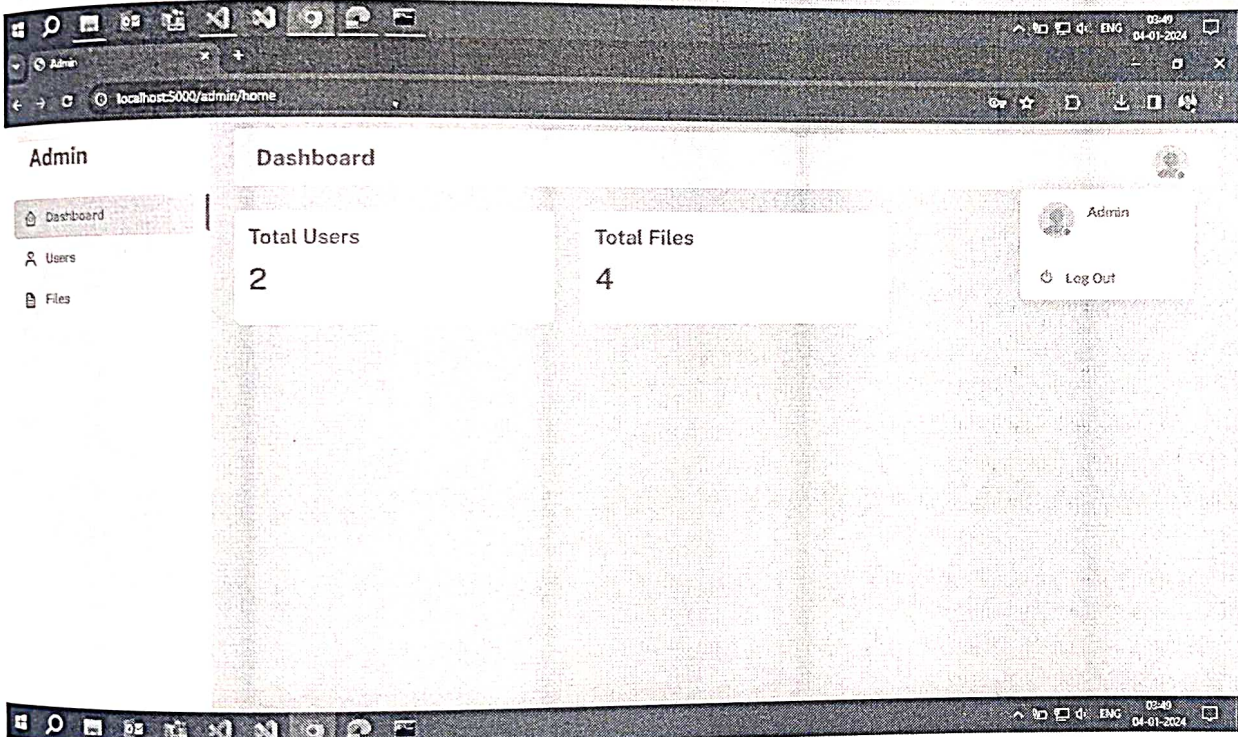
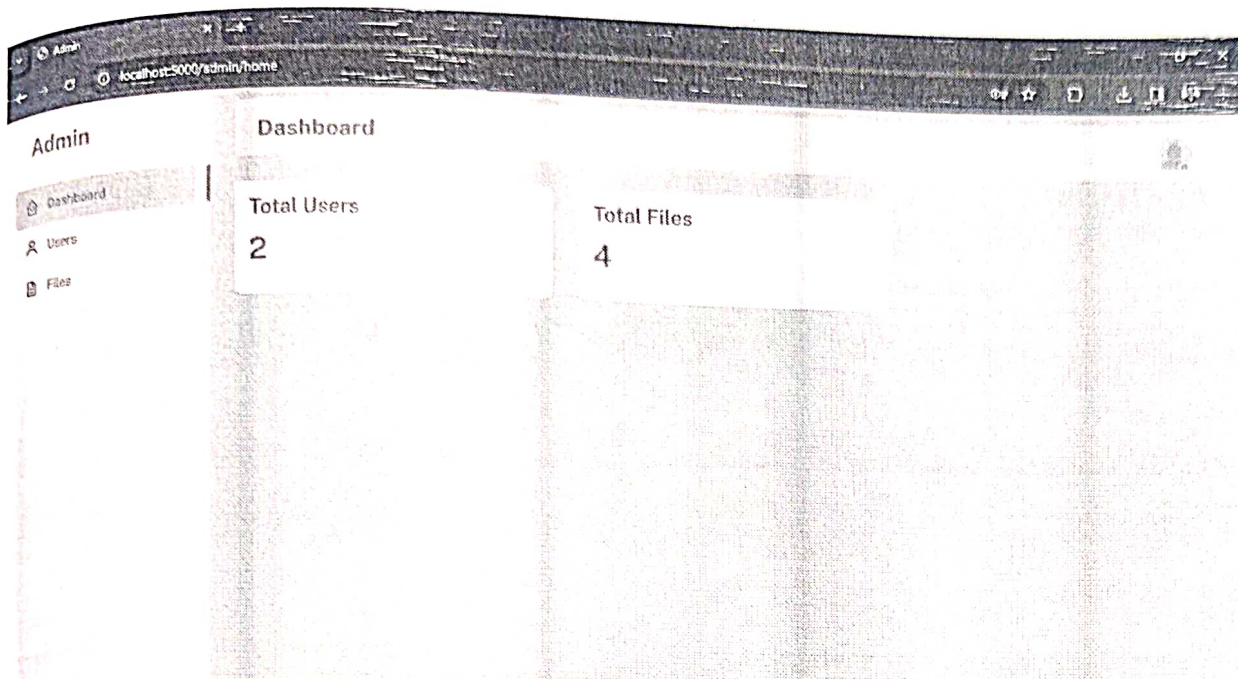
- Once the files uploaded, User can view or delete the file from the Action button.
- Where the deleted file Will be Delete permanently.



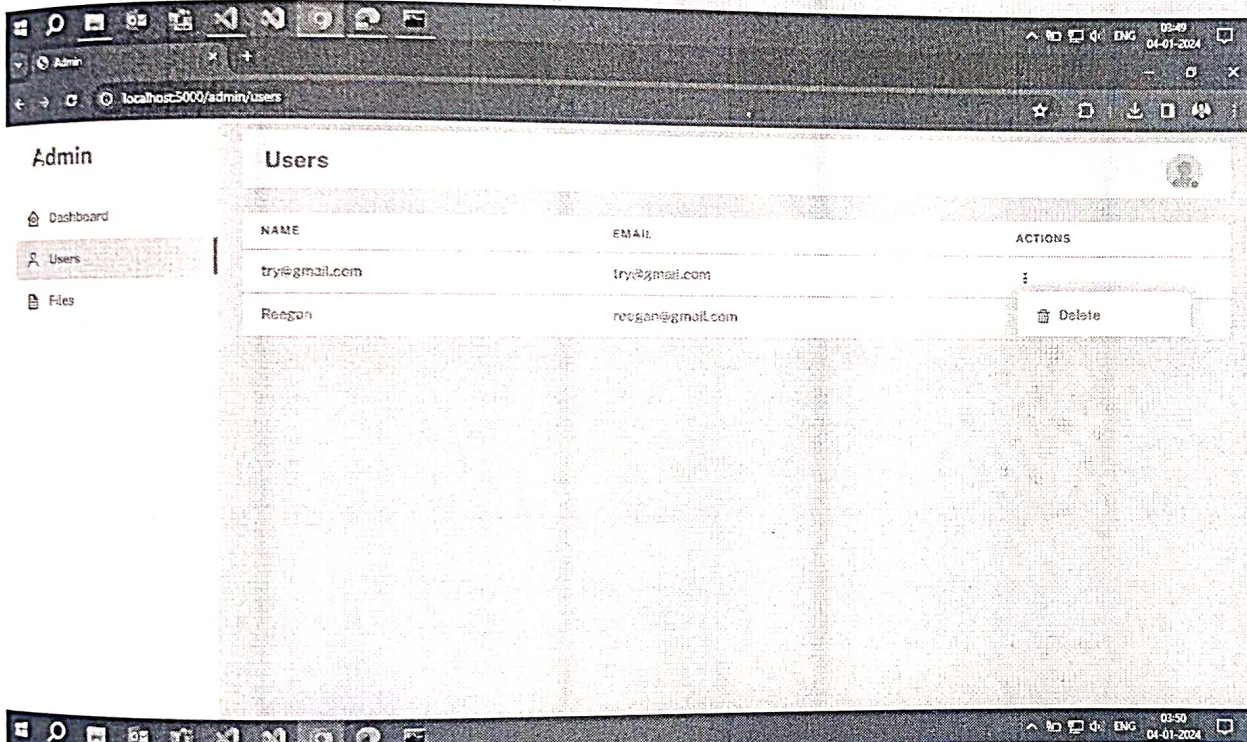
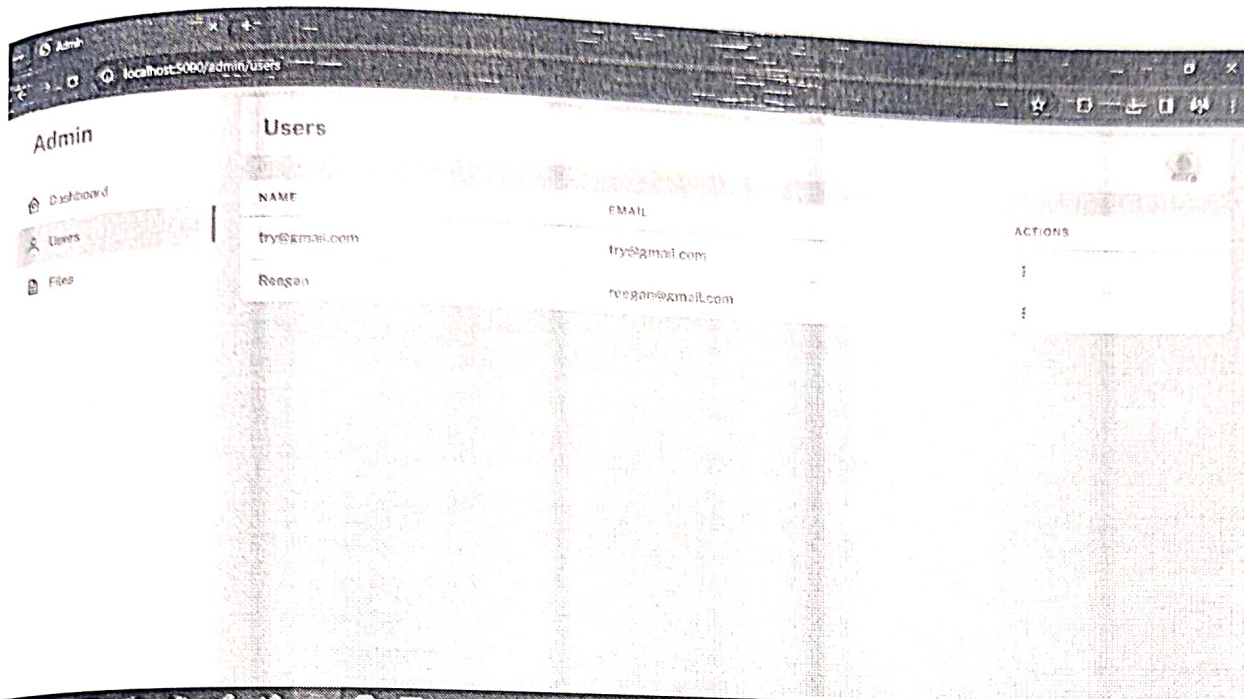
- Above attached screenshot shows the private open image which can open only by entering password by user.



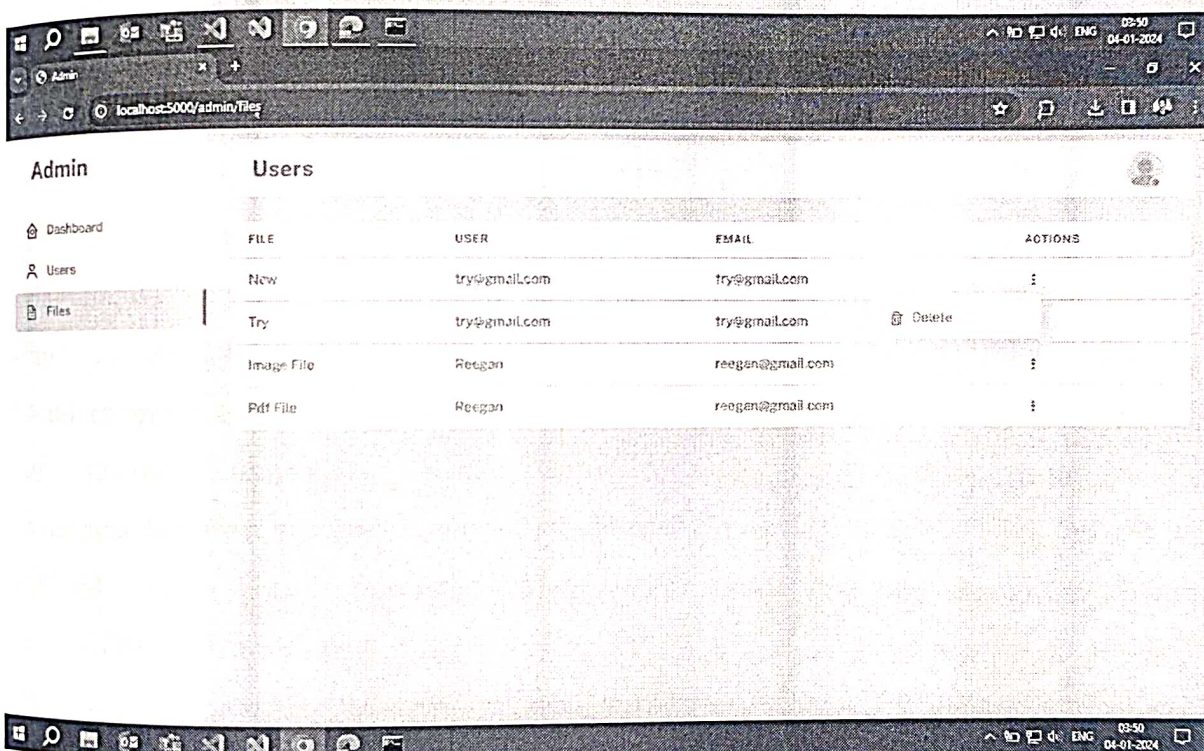
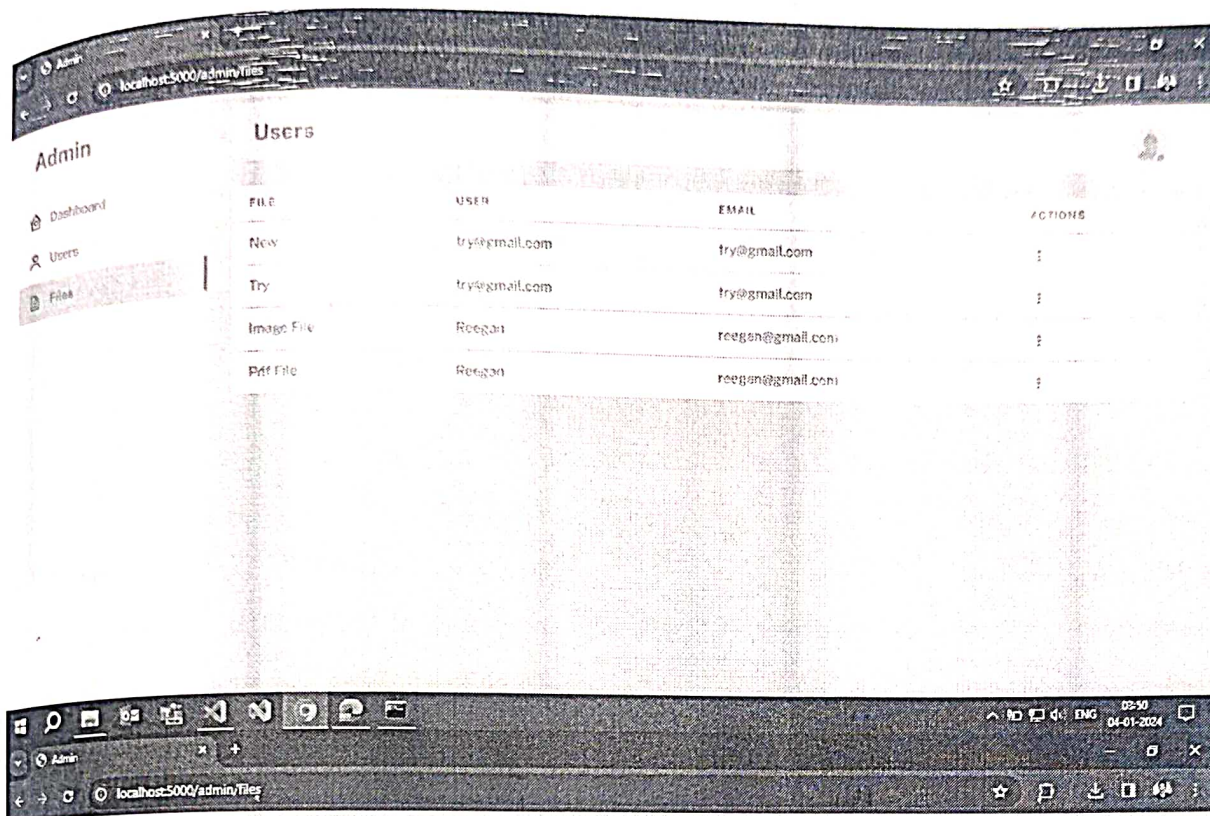
- Above is the Image showing the admin portal of Admin where it is only for Admin.



- From the Dashboard page Admin can logout.
- In users page Admin can see the users detail: Name and mail-id



- Also admin can delete or remove the file and person to view the file.
- Admin can also have these option in action button.



- Admin can delete the file and secure the file with new password, which the file will be store in the cloud securely.

C. RESEARCH PAPER

TO USING HYBRID CRYPTOGRAPHY, THE CLOUD STORE FILES SECURELY

¹ PG Scholar, Avandhikasri M (Department of CSE)

Sathyabama Institute of Science and Technology

avandhikas1@gmail.com

² Dr. L. Lakshmanan (Head, Department of CSE),

Sathyabama Institute of Science and Technology

lakshmanan.cse@sathyabama.ac.in

Abstract— many industries, including the military, higher education, and others, use the cloud for a variety of services and the massive data storage that it offers. Without having direct access to the server computer, users can request to see or retrieve data stored in this cloud. The various organisations and IT industries are using cloud computing at a fast-increasing rate, which makes new software more affordable. Cloud computing is helpful since it's affordable and makes knowledge easily accessible. Numerous benefits of cloud computing include minimal costs and online information accessibility. Considering that customers frequently save sensitive data with cloud storage providers that may not be trustworthy ensuring cloud computing security may be a key consideration for the cloud computing ecosystem. Therefore, maintaining data from an untrusted cloud while exchanging it securely continues to be a difficult problem. By using AES and the RSA algorithm to store data over a

single cloud, my method guarantees the security and privacy of important customer information.

Keywords— cloud computing, information sharing, gratitude. AES and RSA algorithm

I. INTRODUCTION

A hybrid cryptography scheme is offered as a means of safely storing files on the cloud. Because the data are encrypted and only authorised users may access them, users can securely save their files in online cloud storage with this method. The files that the user uploads to the cloud will be securely stored there after being encrypted with a user-specific key, as seen in the above image. Security issues have grown more pressing as the digital environment continues to evolve. These dangers primarily target cloud computing vendors because of an organization's overall lack of insight into data access and movement. Regardless of where client information is stored, organisations handling data may face significant governance and compliance issues if they don't take aggressive steps to strengthen their cloud security. Regardless of the

performance and security of the proposed system through simulations and testing, demonstrating how effectively it may allay security worries in cloud storage.

2.3 "Secure File Storage and Sharing in Cloud Computing Using Hybrid Cryptography" by P. Kumar and P. Singh [2019]

This research proposes a hybrid cryptography strategy for file sharing and storage in cloud computing. The model uses the encryption techniques of RSA and AES to ensure the integrity and secrecy of the data. The study demonstrates the effectiveness and security of the recommended strategy using tests and simulations.

2.4 Rohit Barvekar, Shrajal Behere, Yash Pounikar, and Ms. Anushka Gulhane, "Secure File Storage and Sharing in Cloud Computing using Hybrid Cryptography" (2018)

The recommended security measures that will prevent sensitive data from being used unlawfully will make the system more trustworthy. Keys will be used in the proposed method for both encryption and decryption.

2.5 The work "Secure File Storage and Sharing in Cloud Computing using Hybrid Cryptography" by A. Sharma and K. Sharma. [2017]

This research proposes a hybrid cryptography-based approach for cloud computing

and secure file storage that combines AES and ECC encryption. The recommended method ensures confidentiality, integrity, and availability by utilising encryption and decryption methods. The study demonstrates the effectiveness and security of the recommended strategy using tests and simulations.

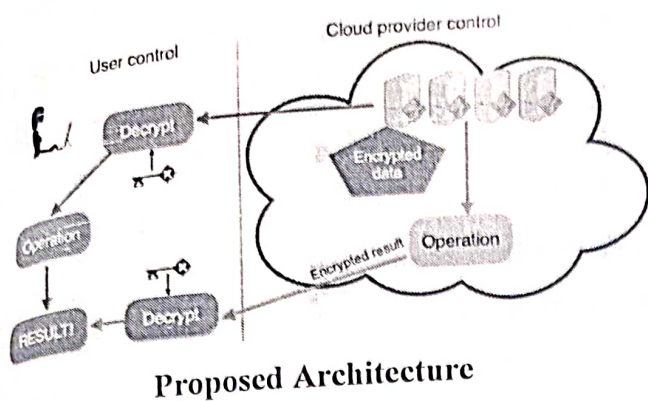
III. SYSTEM IMPLEMENTATION

Existing System

Within the current framework, cloud computing offers a number of advantages, such as low prices and convenient access to information on the Internet. Ensuring cloud computing security may be a crucial factor for the cloud computing environment as users commonly save sensitive information with cloud storage providers, yet these providers may not be trusted. As such, securely transmitting data from an unreliable cloud while ensuring data integrity remains a challenging issue.

Proposed System

The recommended system describes the proposed system, which offers a method for securely storing information in the cloud using a hybrid cryptography algorithm. Thanks to this technology, users may safely store data in online cloud storage since the files are encrypted and only authorised users can access them. As seen in the above graphic, the data that the user uploads to the cloud will be safely kept there after being encrypted with a user-specific key.



Methodology

Advanced Encryption Standard (AES) with RSA algorithm

To get the security of RSA with the speed of AES, combine symmetric encryption with RSA encryption in this approach. Usually, to do this, a temporary AES key, also known as a session key, is generated and secured using RSA encryption. There are other symmetric encryption techniques than AES.

The Advanced Encryption Standard, or AES, is widely used globally. It belongs to a group of encryption techniques known as "symmetric" encryption. In other words, data is encrypted and decrypted using the same secret, which is an encryption key. Probably the most used encryption technique for safeguarding data when it's at rest is AES encryption. It is utilised in storage encryption, database encryption, self-encrypting disc drives, and other applications. It is a global standard that has existed since roughly 2002. Basically, data and the secret encryption key are entered into software that carries out AES encryption, and the result is

encrypted data. When you wish to utilise that data, you just enter the encrypted data into the programme together with the same encryption key, and the original data appears that you may use. We'll talk about other symmetric key encryption techniques in more detail below. The three people who invented the encryption technique—Adi Shamir, Leonard Adleman, and Ron Rivest—are honoured in the name of RSA encryption. RSA is a member of the "asymmetric" encryption class of encryption techniques. Due to the usage of two linked secrets, or keys, for encryption, the term "asymmetric" was born. The terms "public key" and "private key" refer to the two types of keys. The relationship between the keys is that only the corresponding private key may be used to decode data encrypted with the public key. Additionally, the opposite is also true: only the corresponding public key may be used to decrypt data that has been encrypted using the private key. Factorization and very huge prime integers are used in this fascinating math problem. When you have two physically distinct endpoints, you often utilise RSA keys. RSA encryption is frequently utilised in VPN connections, web browser connections to your favourite websites, and several more applications. Every day, we employ asymmetric encryption.

Implementation

Data Owner (DO): The owner uploads the data to a cloud server. The file is divided into octet segments, and utilising multithreading, each segment is encoded concurrently. The encoded file

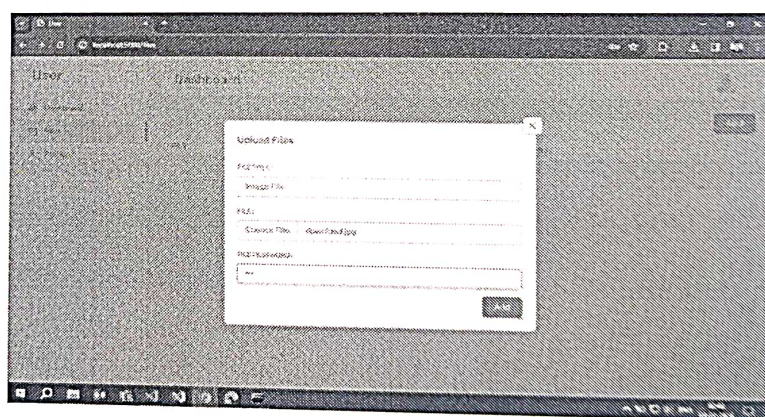
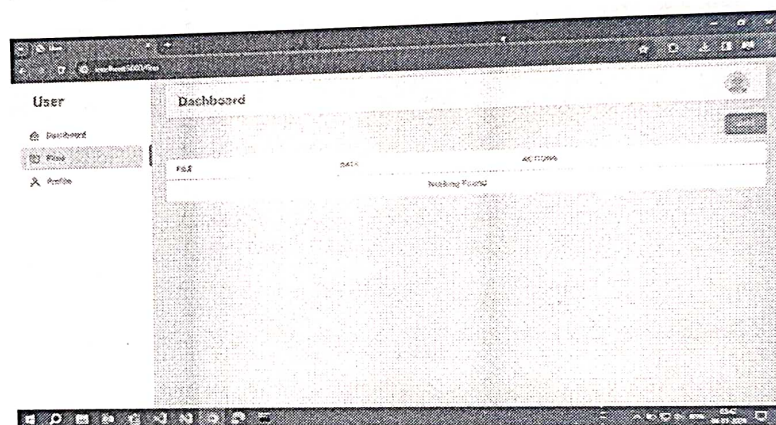
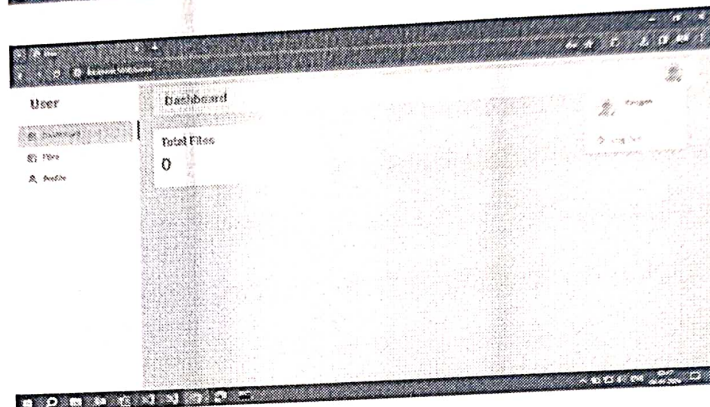
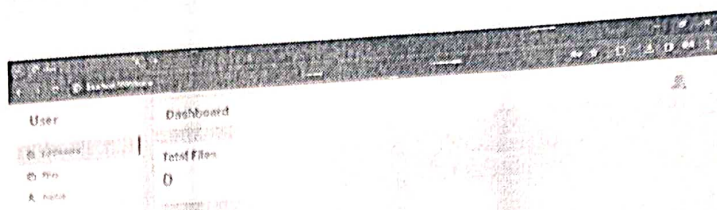
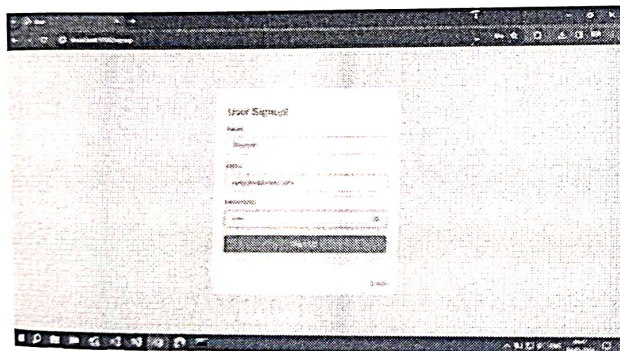
is kept on a cloud server. The cover picture stores the encryption keys. The multiuser environment is known as cloud computing.

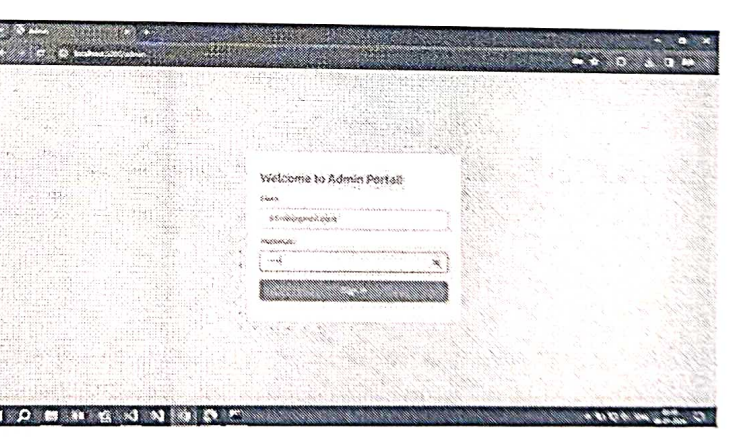
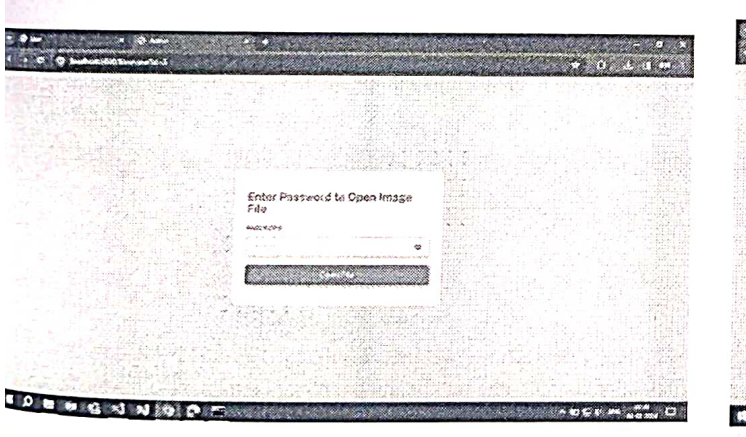
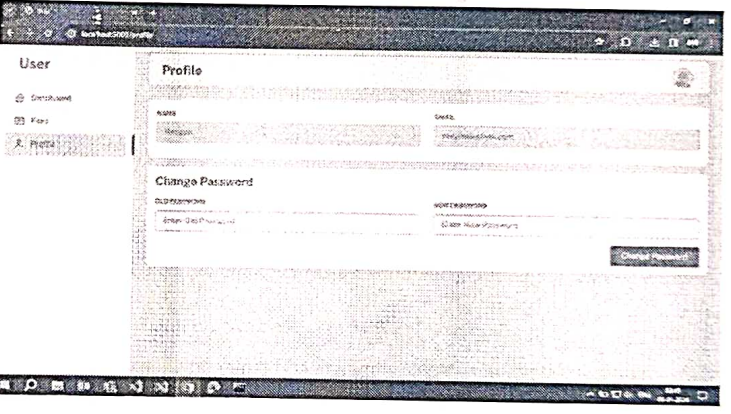
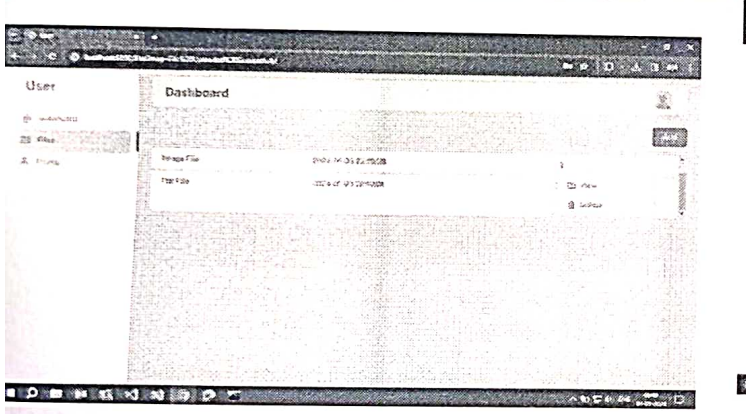
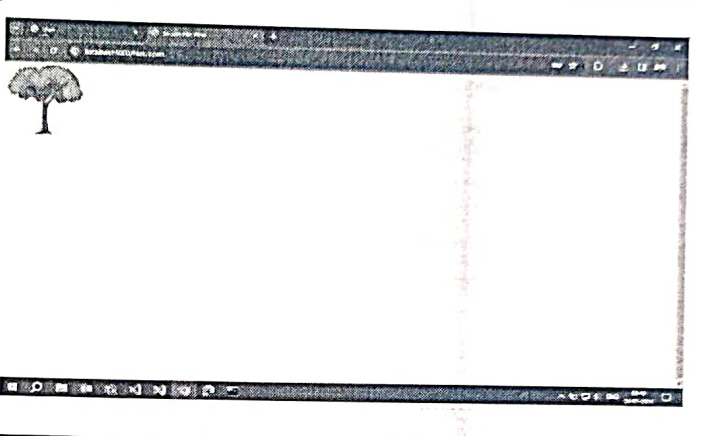
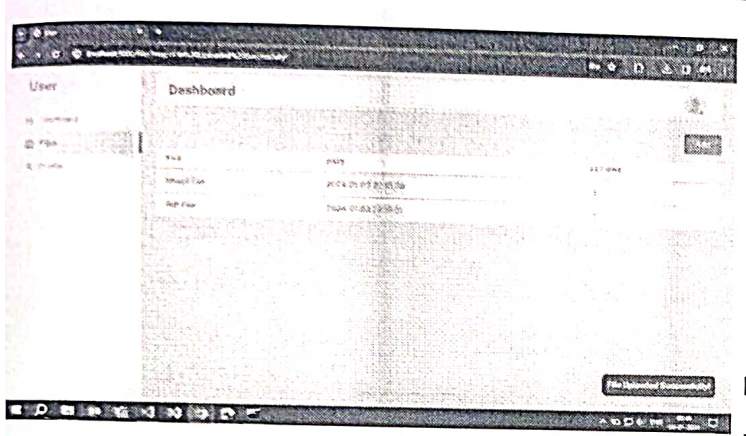
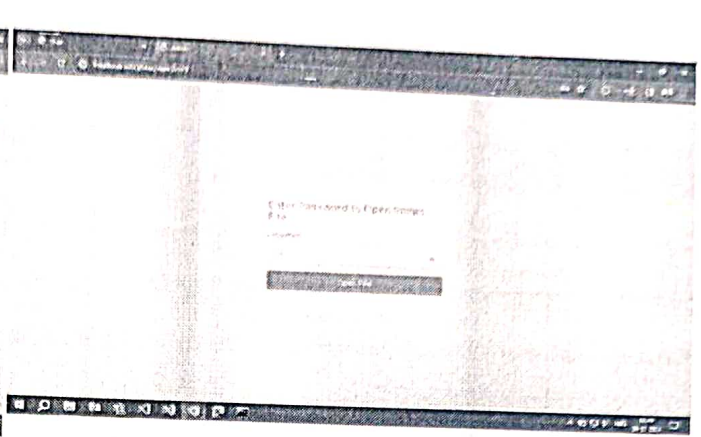
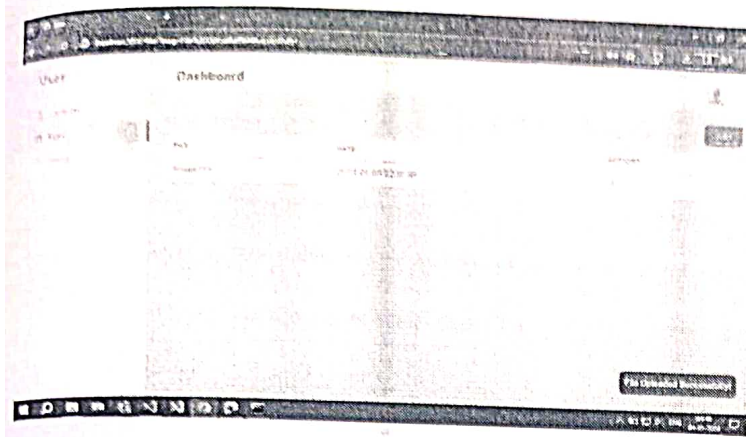
Data User (DU): File request from a Cloud User. Upon requesting the file, the user also receives a key via email that includes important details. The file is decoded using a reverse technique.

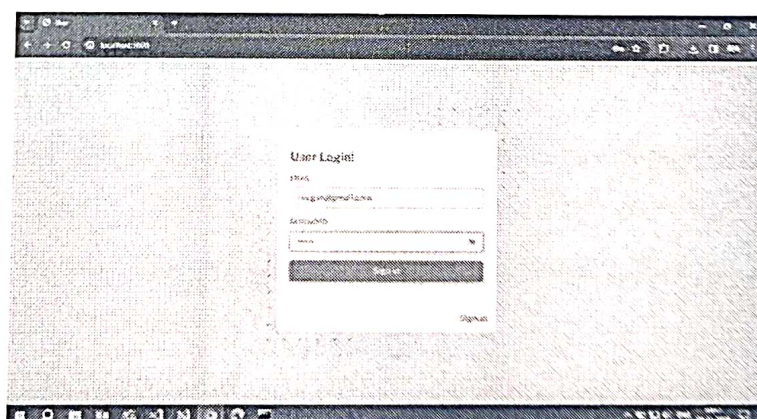
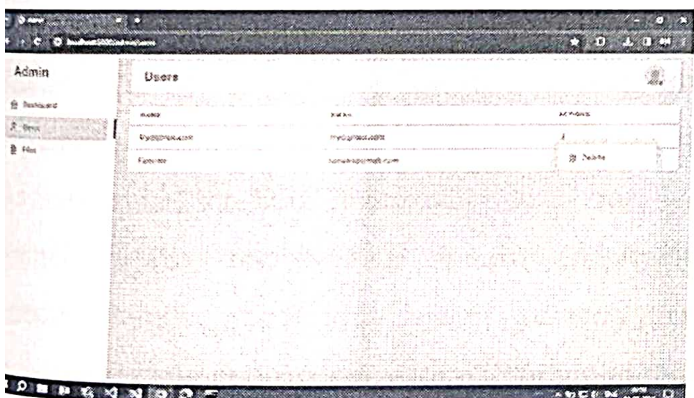
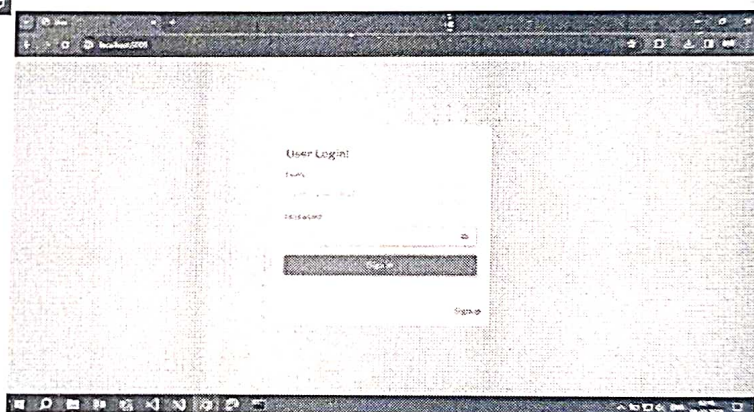
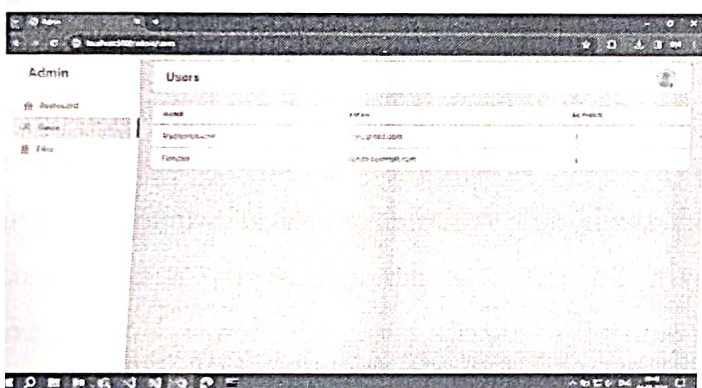
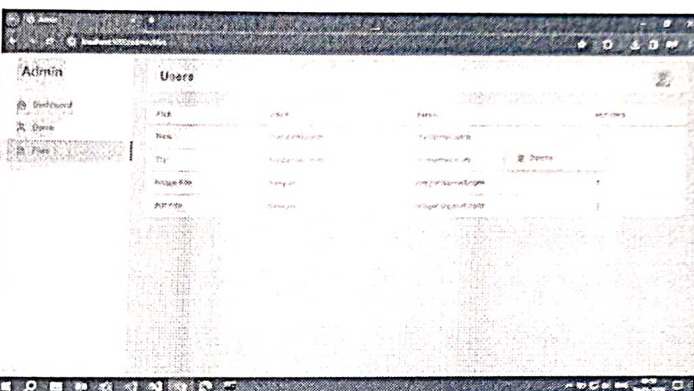
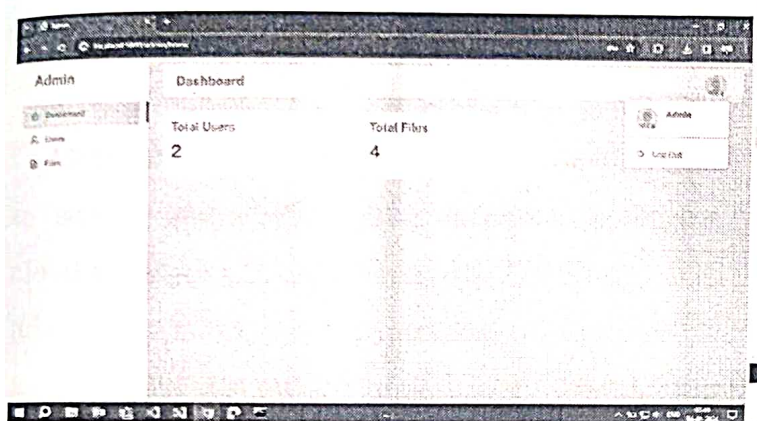
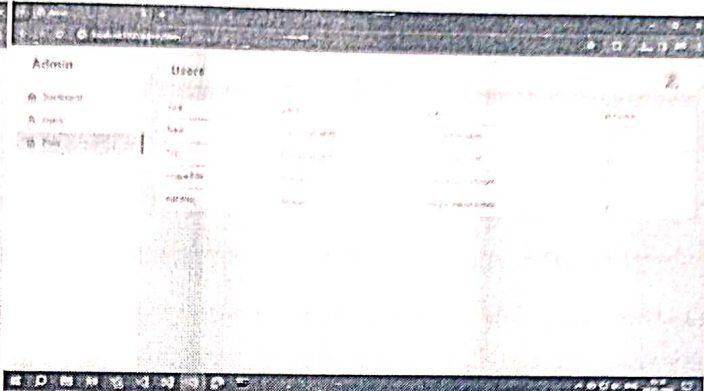
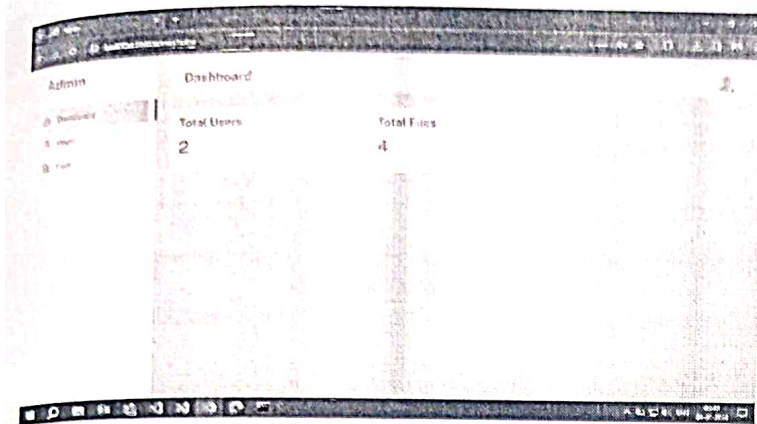
Admin: After entering the home page, admin acts as the owner of the application and can activate and deactivate users as well as view all uploaded file details and request details. If the username and password are entered correctly, only admin can access the home page; if the details entered are incorrect, admin cannot access the home page.

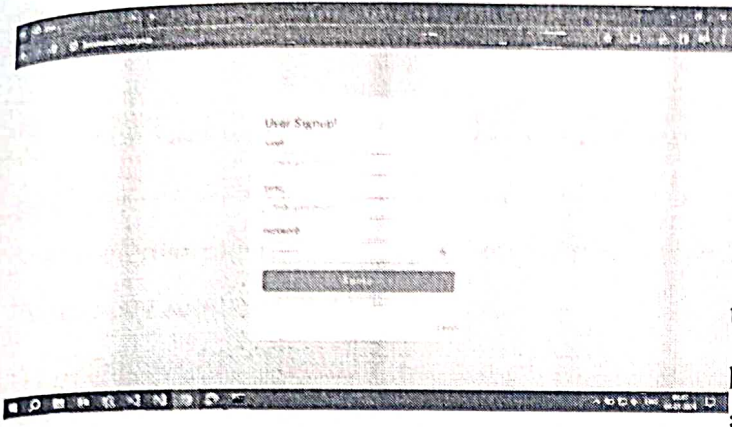
Cloud: With all registered users' information, owners' uploaded file information, and user-downloaded information, the admin of the cloud module may manage it.

Experimental Results









IV. CONCLUSIONS

Many large staggered organisations have moved to storing and sharing their information on the cloud because to its many benefits. The first step in this effort is to raise awareness of the serious security risks that owners of data face while sharing it on the cloud. The most popular information sharing schemes are then quickly reviewed to identify any shortcomings in each of them. This study suggests a privilege-based multilevel organisational data sharing strategy that allays the concerns and allows for the safe and effective exchange of data on the cloud. divides a document containing information into many sections based on the benefits to the client and the material's impact. Then, each information file portion is common, depending on the benefits of the information client. Assuming that the DBDH presumption is true, we formally show that is safe against adaptively selected plaintext attack. This system's primary goal is to safely store and retrieve data on the cloud that is under the exclusive control of the data owner. Data security concerns with cloud storage are

resolved through the use of steganography and cryptography methods. In order to secure data, the AES and RSA algorithms are used. Key data is securely preserved with the use of LSB technology (Steganography). Using the multithreading technology, the encryption and decryption procedure takes less time. We have achieved improved data integrity, high security, low latency, authentication, and secrecy with the aid of the suggested security mechanism.

REFERENCES

- [1] "Secure File Storage On Cloud Using Hybrid Cryptography Algorithm," Uttam Kumar, Mr. Jay Prakash, International Journal Of Creative Research Thoughts (IJCRT), 2021. 2320-2882 is the ISSN. [Prime Document].
- [2] Punam V. Maitri and Aruna Verma present "Secure File Storage In Cloud Computing Using Hybrid Cryptography Algorithm." In IEEE Wispnet 2020.
- [3] Rashidhagat and Purvi Joshi, "Hybrid Cryptography for Secure File Storage," International Journal of Innovative Research in Technology (IJIRT), 2020.
- [4] Bilal Habib, Bertrand Cambou, Christopher Philabaum," Using Public Key Exchange Scheme That Is Addressable (PKA) in the cloud computing, 2019 IEEE CNS.
- [5]. Tulip Dutta, Amarjyoti Pathak "Secure Data Sharing In Cloud Storage Using Key Aggregation

Cryptography," Tulip Dutta, Amarjyoti Pathak, 2019.

[6]. Bhale Pradeep Kumar Gajendra, Vinay Kumar Singh, and More Sujcet, "Achieving Cloud Security Through Identity-Based Encryption, MD5, and Third-Party Auditor," ICCCA2018.

[7] Shakeeba S. Khan, Professor R.R. Tuteja, "Cryptographic Algorithms and Security in Cloud Computing," 2017.

[8] J. Jenifer Sanjana, and T.M. Reuka.

"To Secure File Sharing Using RSATechniques In Cloud" was presented at the CNS 2021.

[9] Rajiv Varma and Vijay Verma present "Secure File Storage on Cloud Computing Utilizing AES Algorithm." IJIRT, in 2021.

[10] Kushdeep Kaur and Kashvi Jain, "Hybrid Cryptography to Secure File Storage," International Conference of Innovative Research in Technology (ICIRT), 2020.

[11] Jumana, Vinston Jacob, Sheena Paarek, Christopher, Public Key Exchange Scheme That Is Addressable (PKA), 2019 IEEE CNS.

[12] Gajendra Shenoji Singh, Kishore Agarwal, and Haseena, "On Achieving Cloud Security to Encryption, and Third Party Auditor," IEE 2020.

[13] Pranav S. Govindh, Kathrin Moses, Christopher Rancy "Cryptographic Algorithms and Security in Cloud Computing," 2018.

[14] Bello SA, Oyedele LO, Akinde OO, Bilal M, Delgado JM, LA Ajayi (2020) "Cloud Computing in Industry: To Store file Securely" 122.1.3441

[15] Putta Bharathi, Gayathri Annam, Jaya Bindu Kandi, Vamsi Krishna Duggana, Anjali T "Secure File Storage Using Hybrid Cryptography" IEEE 2021

[16] Jiang Lai, Swee-Huay Heng "Secure File Storage on Cloud Computing using Hybrid-Cryptography", International Journal of Informatics and Web Engineering 2022.

D. CERTIFICATE

CHRIST THE KING ENGINEERING COLLEGE

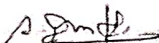
RUN BY FATHER'S OF MMI AND SISTER'S OF DMI
158/3, ONNIPALAYAM ROAD, KARAMADAI, COIMBATORE -641 104

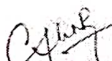


FIRST INTERNATIONAL CONFERENCE ON
INNOVATIVE RESEARCH IN ENGINEERING SCIENCES
(ICIRES-2024)

CERTIFICATE

M. AVANDHIKASRI, COMPUTER SCIENCE AND ENGINEERING, HAS
PRESENTED A PAPER ENTITLED 'TO USING HYBRID CRYPTOGRAPHY,
THE CLOUD STORE FILES SECURELY' FROM SATHYABAMA INSTITUTE
OF SCIENCE AND TECHNOLOGY AT THE FIRST INTERNATIONAL
CONFERENCE ON INNOVATIVE RESEARCH IN ENGINEERING SCIENCE
HELD AT CHRIST THE KING ENGINEERING COLLEGE ON 10.04.2024.


Dr. S. Dinesh Kirupha
Convenor


Dr. N. R. Gayathiri
Convenor


Dr. M. Jeyakumar
Principal

