

MULTI STAGE CLOUD SECURITY FOR CLOUD DATA USING AMALGAMATE DATA

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

By

TERRANCE BONIFACE J (Register No - 40111325)

THIRUPATHI PANDI M (Register No - 40111337)



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

SATHYABAMA

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

**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

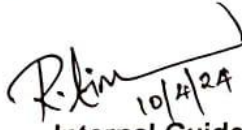
Accredited "A++" by NAAC | Approved by AICTE

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 **Terrance Boniface J (40111325)** and **Thirupathi Pandi.M (40111337)** who carried out the Project entitled **"MULTI STAGE CLOUD SECURITY FOR CLOUD DATA USING AMALGAMATE DATA"** under my supervision from November 2023 to April 2024.


10/4/24
Internal Guide

Mrs. R. Aishwarya, M.E.,(Ph.D.,)

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Jeppiaur, Chennai - 600 119.

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

Submitted for Viva voice Examination held on




Internal Examiner


External Examiner

DECLARATION

I, Thirupathi Pandi M(Register.No - 40111337), hereby declare that the Project Report entitled "MULTI STAGE CLOUD SECURITY FOR CLOUD DATA USING AMALGAMATE DATA" done by me under the guidance of Mrs. R. Aishwarya, M.E.,(Ph.D.,) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 12/04/2024

PLACE:Chennai

M. Thirupathi Pandi

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** 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 **Mrs. R. Aishwarya, M.E.,(Ph.D.,)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 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

The development of cloud computing technology with the explosive growth of unstructured data, cloud storage technology gets more attention and better development. The cloud provider does not have suggestions regarding the information and the cloud data stored and maintained globally anywhere in the cloud. The privacy protection schemes are usually based on encryption technology. There are many privacy preserving methods in the side to prevent data in cloud. We propose a three-layer storage framework based on fog computing. The proposed framework can both take full advantage of cloud storage and protect the privacy of data. Here we are using Hash-Solomon code algorithm is designed to divide data into different parts. If the one data part missing we lost the data information. In this framework we are using bucket concept based algorithms and secure the data information and then it can show the security and efficiency in our scheme. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively Software as a Service (SaaS): Client releases their application on a hosting environment which can be accessed through network from various clients by application users. The client does not manage or control the underlying cloud infrastructure with the possible exception of limited user-specific application configuration settings. Google Apps and Microsoft Office 365 are the examples for SaaS.

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	IV
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
	2.1 Open Problems in Existing System	6
	2.2 Disadvantage in Existing System	6
3	REQUIREMENTS ANALYSIS	7
	3.1 Hardware Requirement Specification Document	10
	3.2 Software Requirements Specification Document	10
4	DESCRIPTION OF PROPOSED SYSTEM	11
	4.1 Architecture Diagram	12
	4.2 Proposed Algorithm	12
	4.3 Selecting Methodology or Process Model	14
	4.4 Modules Description	14
	4.5 Requirement Analysis	15
	4.6 Functional Requirements	15
	4.7 Non-Functional Requirements	16
	4.8 UML Diagrams	16
	4.9 Use Case Diagram	18
	4.10 Sequence Diagram	19
	4.11 Activity Diagram	20
	4.12 Collaboration Diagram	21
	4.13 System Study	22
5	IMPLEMENTATION DETAILS	24
	5.1 Architecture	24
	5.2 System Testing	25
6	RESULT AND DISCUSSION	29
	6.1 File Upload	29
	6.2 Storage Scheme	30
	6.3 Recovery Module	30

7	CONCLUSION	31
	REFERENCES	32
	APPENDIX	33
	A.SOURCE CODE	33
	B.SCREENSHOTS	62
	C.RESEARCH PAPER	66

LIST OF FIGURES

CHAPTER NO.	FIGURE NAME	PAGE NO.
4.1	Architecture Diagram	12
4.2	Binary Tree	13
4.3	Use case Diagram	18
4.4	Sequence Diagram	19
4.5	Activity Diagram	20
4.6	Collaboration Diagram	21
B.1	Login Page	62
B.2	Register Page	62
B.3	Home Page	63
B.4	Upload Page	63
B.5	View Page	64
B.6	Download Page	64
B.7	Project Acceptance	65

CHAPTER 1

INTRODUCTION

Organizations will also need to train its employees to handle machines, software or any equipment for that matter. Now organizations are trying to implement recruitment techniques which lead to acquire talented employees. Artificial Intelligence term was coined by John McCarthy (1950) soon after when he published a paper entitled “Computing Machinery and intelligence” and this paper lead to open the doors to the filed termed as AI. Now, HR managers implement Artificial Intelligence technology to recruit, retain and inspire the proficient manpower which leads to success and growth for both the employer and employee. AI in recruiting has a major role in talent acquisition (Madeline Laurano Co-Founder & Chief Research Office). Nearly 30% of companies implementing AI for recruiting activity which can lesser down time, cost reduction and can place right talent in the right position. Artificial Intelligence is defined as “An area of study in the field of computer science. Artificial intelligence is concerned with the development of computers able to engage in human-like thought processes such as learning, reasoning, and self-correction”. Artificial intelligence is all about fast thinking with lot of knowledge, think as human, logical reasoning, etc., as it is a part of computer science leading to efficiency in problem solving by giving solutions.

In recent years, the proliferation of cloud computing has revolutionized the way businesses store, process, and access data. However, with this increased reliance on cloud services comes a heightened need for robust security measures to protect sensitive information from unauthorized access, data breaches, and other cyber threats. Traditional security methods often fall short in addressing the dynamic and distributed nature of cloud environments, necessitating innovative approaches to safeguard cloud data.

This paper introduces a multi-stage cloud security framework that leverages the concept of amalgamated data to enhance the protection of data stored in the cloud. Amalgamated data refers to the aggregation of various data sources, including user behavior patterns, system logs, network traffic, and application telemetry, to provide a comprehensive view of the cloud environment's security posture.

In subsequent sections of this paper, each layer of the proposed framework will be discussed in detail, including implementation strategies, best practices, and case studies demonstrating its effectiveness in enhancing cloud security. Additionally, practical considerations such as scalability, interoperability, and regulatory compliance will be addressed to ensure the feasibility and applicability of the proposed approach in diverse cloud environments.

Beyond the initial implementation of security measures, maintaining vigilance and adaptability is crucial in the ever-evolving landscape of cloud security. This layer emphasizes the importance of continuous monitoring and adaptation to detect new threats and vulnerabilities as they emerge. By employing real-time monitoring tools and automated security protocols, organizations can swiftly respond to changing threat landscapes and adapt their security measures accordingly. Moreover, regular audits and security assessments help ensure compliance with industry standards and regulatory requirements, fostering a culture of continuous improvement and resilience in the face of evolving cyber threats.

Incorporating this additional paragraph provides further emphasis on the importance of continuous monitoring and adaptation, highlighting the dynamic nature of cloud security and the need for organizations to remain vigilant in their efforts to safeguard their data assets.

In conclusion, the multi-stage cloud security framework presented in this paper offers a holistic approach to safeguarding cloud data, leveraging the power of amalgamated data and advanced security measures to mitigate evolving cyber threats effectively. By adopting this framework, organizations can enhance their resilience against security breaches and safeguard the integrity, confidentiality, and availability of their cloud data assets.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength.

Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength.

Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

1.A Secure Cloud-assisted Urban Data Sharing Framework for Ubiquitous-cities

From the experiment analysis, we can see that our dynamic operations have less computation 340 time. We hold that this framework can be applied into the real urban data sharing applications in real-world ubiquitous-cities. The features of this framework is suitable for citizens' sensitive data protection and utilization, such as the life electricity consumption and healthcare information, which can only be utilized and modified by some specific people and users besides the data owners.

2. Privacy-Preserving Smart Semantic Search Based on Conceptual Graphs Over Encrypted Outsourced Data

We introduce conceptual graphs (CGs) as a knowledge representation tool. Then, we present our two schemes (PRSCG and PRSCG-TF) based on CGs according to different scenarios. In order to conduct numerical calculation, we transfer original CGs into their linear form with some modification and map them to numerical vectors. Second, we employ the technology of multi-keyword ranked search over encrypted cloud data as the basis against two threat models and raise PRSCG and PRSCG-TF to resolve the problem of privacy-preserving smart semantic search based on CGs. Finally, we choose a real-world data set: CNN data set to test our scheme. We also analyze the privacy and efficiency of proposed schemes in detail.

3. A privacy preserving and copy-deterrence content-based image retrieval scheme in cloud computing

The feature vectors are protected by the secure KNN algorithm, and image pixels are encrypted by a standard stream cipher. In addition, considering the case that the authorized query users may illegally copy and distribute the retrieved images to someone unauthorized, we propose a watermark-based protocol to deter such illegal distributions. In our watermark-based protocol, a unique watermark is directly embedded into the encrypted images by the cloud server before images are sent to the query user. Hence, when an illegal image copy is found, the unlawful query user

who distributed the image can be traced by the watermark extraction. The security analysis and experiments show the security and efficiency of the proposed scheme.

4. Toward Efficient Multi-Keyword Fuzzy Search Over Encrypted Outsourced Data With Accuracy Improvement

Concretely, we develop a novel method of keyword transformation and introduce the stemming algorithm. With these two techniques, the proposed scheme is able to efficiently handle more misspelling mistake. Moreover, our proposed scheme takes the keyword weight into consideration during ranking. Like Wang et al.'s scheme, our proposed scheme does not require a predefined keyword set and hence enables efficient file update too. We also give thorough security analyses and conduct experiments on real world data set, which indicates the proposed scheme's potential of practical usage.

5. A Literature Survey on Secure Data Sharing in Cloud Storage with Key Aggregate Cryptosystem

In this survey we studied different cryptography techniques for data sharing security. One trivial solution to achieving secure data sharing in the cloud is for the data owner to encrypt his data before storing into the Cloud, and hence the data remain information-theoretically secure against the Cloud provider and other malicious users. With more mathematical tools, cryptography schemes are getting more versatile and often involve multiple keys for a single application. Thus it considers how to compress secret keys in public-key cryptograms which support delegation of secret keys for different cipher text classes in cloud storage. Our approach is more flexible than hierarchical key assignment.

2.1 OPEN PROBLEMS IN EXISTING SYSTEM

- Recent years witness the development of cloud computing technology. With the explosive growth of unstructured data, cloud storage technology gets more attention and better development.
- The computer technology has developed rapidly. Cloud computing has gradually matured through so many people effort's.
- In current storage schema, the user's data is totally stored in cloud servers. If the user lose their right of control on data and face privacy risk.
- The privacy protection schemes are usually based on encryption technology. These kinds of methods cannot effectively resist attack from the inside of cloud server.

2.2 DISADVANTAGE OF EXISTING SYSTEM

- Changes in the understanding of risk as a result of extending the data centre into the cloud.
- Low latency and location awareness

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 AIM

The aim of this paper is to propose a comprehensive multi-stage cloud security framework that leverages the concept of amalgamated data to enhance the protection of data stored in the cloud. This framework aims to address the inherent challenges of securing cloud environments by providing a multi-layered approach that encompasses preventive, detective, predictive, and responsive measures. By amalgamating diverse data sources such as user behavior patterns, system logs, network traffic, and application telemetry, the framework aims to provide a holistic view of the cloud environment's security posture.

3.2 SCOPE

Security Framework Design: This study will delve into the design and architecture of a multi-stage cloud security framework, emphasizing the integration of various security layers and the amalgamation of diverse data sources. The scope includes defining the components, interfaces, and interactions of the framework to establish a comprehensive security strategy.

Layered Security Measures: The scope encompasses the exploration of different layers of security measures within the proposed framework, including preventive, detective, predictive, and responsive controls. Each layer will be analyzed to understand its role, implementation strategies, and effectiveness in enhancing cloud security.

Amalgamated Data Sources and Analysis: This study will investigate the sources of data that can be amalgamated to provide comprehensive insights into cloud security. It will explore techniques for collecting, aggregating, and analyzing diverse data sources to detect anomalies, identify threats, and enhance situational awareness in cloud environments.

3.3 SYSTEM REQUIREMENT OUTCOME

3.3.1 Scalable Data Integration Platform:

Outcome: Develop a scalable data integration platform capable of aggregating and correlating diverse data sources from the cloud environment, including user activities, system logs, network traffic, and application telemetry. Requirement: Implement data integration tools and techniques that can efficiently process large volumes of data in real-time or near-real-time, ensuring timely analysis and detection of security threats.

3.3.2 Comprehensive Security Analytics Engine:

Outcome: Design and implement a comprehensive security analytics engine that can analyze amalgamated data to identify security threats, anomalies, and patterns within the cloud environment. Requirement: Utilize advanced analytics algorithms, machine learning models, and artificial intelligence techniques to analyze and correlate diverse data sources, enabling proactive threat detection and incident response.

3.3.3 Layered Security Controls:

Outcome: Establish multiple layers of security controls within the cloud environment, including preventive, detective, predictive, and responsive measures, to mitigate security risks and protect cloud data assets. Requirement: Implement a combination of access control policies, encryption mechanisms, intrusion detection systems, anomaly detection algorithms, and incident response protocols to address different types of security threats at various stages.

3.3.4 Real-Time Monitoring and Alerting:

Outcome: Implement real-time monitoring and alerting capabilities to detect and respond to security incidents and anomalies as they occur within the cloud environment. Requirement: Deploy monitoring tools and technologies that can continuously monitor cloud data activities, analyze security events in real-time, and generate alerts or notifications for security incidents requiring immediate attention.

3.3.5 Adaptive Security Measures:

Outcome: Develop adaptive security measures that can dynamically adjust security controls and responses based on changing threat landscapes and evolving patterns of cloud data activities. Requirement: Implement adaptive security algorithms and policies that can learn from historical data, adapt to emerging threats, and autonomously adjust security configurations and responses to mitigate risks effectively.

3.3.6 Compliance and Governance Framework:

Outcome: Establish a compliance and governance framework to ensure that the multi-stage cloud security solution aligns with regulatory requirements, industry standards, and organizational policies. Requirement: Define governance structures, policies, and processes for monitoring, auditing, and enforcing compliance with security regulations and standards applicable to the organization's industry and geography.

3.3.7 Continuous Improvement and Optimization:

Outcome: Foster a culture of continuous improvement and optimization by regularly evaluating and enhancing the effectiveness and efficiency of the multi-stage cloud security solution. Requirement: Implement mechanisms for collecting feedback, analyzing performance metrics, conducting security assessments, and identifying opportunities for optimization and enhancement within the cloud security framework.

3.3.8 Interoperability with Existing Systems:

Outcome: Ensure seamless interoperability with existing cloud infrastructure, applications, and security systems within the organization's IT ecosystem. Requirement: Implement standardized protocols and interfaces for data integration, security controls, and communication between the multi-stage cloud security solution and other systems, ensuring compatibility and ease of integration.

3.3.9 Resource Optimization and Efficiency:

Outcome: Optimize resource utilization and operational efficiency of the multi-stage cloud security solution to minimize costs and maximize performance. Requirement: Implement resource management techniques, such as workload balancing, resource allocation, and optimization algorithms, to ensure

efficient utilization of computing resources (e.g., CPU, memory, storage) within the cloud environment.

3.3.10 Data Privacy and Confidentiality:

Outcome: Safeguard the privacy and confidentiality of sensitive data stored and processed in the cloud environment, ensuring compliance with data protection regulations and privacy laws. Requirement: Implement data encryption, tokenization, and anonymization techniques to protect sensitive data at rest and in transit, mitigating the risk of unauthorized access or disclosure.

3.3.11 Resilience and Disaster Recovery:

Outcome: Improve resilience and disaster recovery capabilities to ensure business continuity and mitigate the impact of potential disruptions or data loss events within the cloud environment. Requirement: Implement data backup, replication, and fail over mechanisms to maintain data availability and recoverability in the event of hardware failures, natural disasters, or cyber attacks targeting cloud infrastructure.

3.4 REQUIREMENTS SPECIFICATION.

3.4.1 Hardware specifications:

System: Pentium Dual Core

Hard Disk: 120 GB

Monitor: 15" LED

Input Devices: Keyboard, Mouse

Ram: 4 GB

3.4.2 Software specification:

Operating system: Windows 7/10

Coding Language: Python

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

Artificial intelligence is a technology which can work smartly as equal to human brain in different situations. It gains attention and importance in automating recruiting system when compared to traditional recruitment methods. Recruitment is the central activity of all organization to work on. Now, recruitment industry is taking up growth by implementing smart way to recruit i.e., recruiting through artificial intelligence. And numerous industries are focusing on the changes taking place in recruitment process. AI technology has tremendous impact on recruitment activity as it enables the recruiter to align all some unstructured candidate bio-data, construct profile into uniformity, identify and match skill sets required for the industry. In today's era, recruiters think AI technology is competing with them for recruitment activity. But, it is human built software to make the work at ease while the process is carried on. To conclude the role of AI it is the combination of humans and AI that leads to data maintenance, save the cost and time to the organizations with more accuracy and access in total recruitment process.

ADVANTAGES OF PROPOSED SYSTEM

- Time saving
- Cost saving
- Hire with Quality

4.1 ARCHITECTURE DIAGRAM

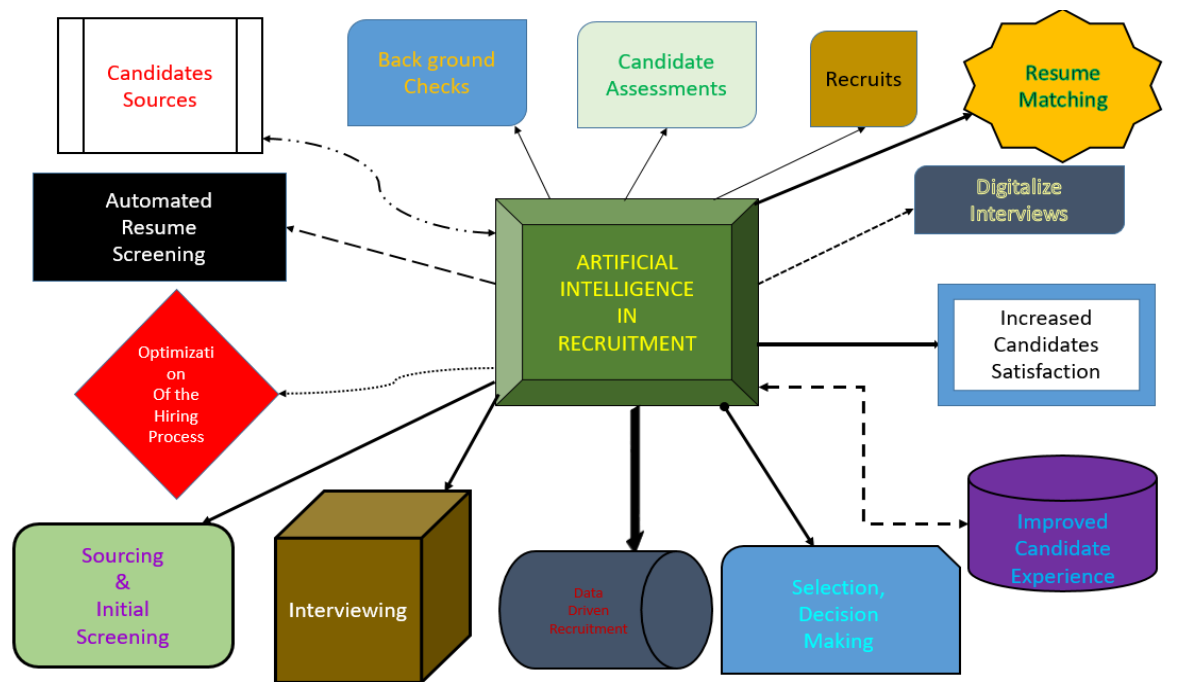


Fig. 4.1:Architecture Diagram

4.2 PROPOSED ALGORITHM

4.2.1 Decision Trees for Classification: A Machine Learning Algorithm

4.2.1.1 Introduction

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

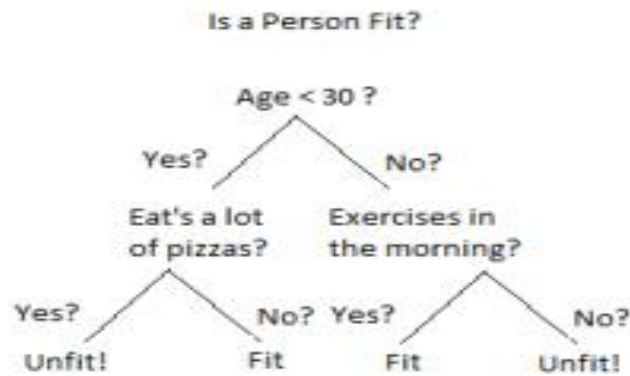


Fig. 4.2: Binary Tree

An example of a decision tree can be explained using above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes no type problem).

There are two main types of Decision Trees:

Classification trees (Yes/No types)

What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is Categorical.

Regression trees (Continuous data types)

Here the decision or the outcome variable is Continuous, e.g. a number like 123.

Working

Now that we know what a Decision Tree is, we'll see how it works internally. There are many algorithms out there which construct Decision Trees, but one of the best is called as ID3 Algorithm. ID3 Stands for Iterative Dichotomiser 3.

Before discussing the ID3 algorithm, we'll go through few definitions.

4.3 SELECTING METHODOLOGY OR PROCESS MODEL:

- Login
- Registration
- Storage Scheme
- Recovery Scheme

4.4 MODULES DESCRIPTION

4.4.1 Login Module

This is the first activity that opens when user open the website. User needs to provide a correct contact number and a password, which user enters while registering, in order to login into the website. If information provided by the user matches with the data in the database table then user successfully login into the website else message of login failed is displayed and user need to reenter correct information. A link to the register activity is also provided for registration of new users.

4.4.2 Registration Module

A new user who wants to access the web page needs to register first before login. By clicking on register button in login activity, the register activity gets open. A new user registers by entering full name, password and contact number. A user needs to enter password again in confirm password text box for confirmation. When user enters the information in all text boxes, on the click of register button, the data is transferred to database and user is directed to login activity again. Registered user then needs to login in order to access the web page. Validations are applied on all the text boxes for proper functioning of the web page. Like information in each text box is must that is each text box, either it is of name, contact, password or confirm password, will not be empty while registering. If any such text box is empty app will give message of information is must in each text box. Also data in password and confirm password fields must match for successful registration. Another validation is contact number must be valid one that is of 10 digits. If any such validation is violated then registration will be unsuccessful and then user needs to register again. Message that web page will display when one of the field is empty. If all such information is correct user will be

directed to login activity for login into the web page.

4.4.3 Storage Module

In this module, user can store their files into three different storage server. The data owner has not control over the data after it is uploaded on cloud. In this module, the original data get encrypted into three different layer's. The data in each layer can be encrypted by using different cryptography algorithm's and encryption key before storing them in the Cloud.

4.4.4 Recovery Module

In this Module, user can recover their files from three different storage server that is Cloud server, Fog server and Local machine. Here we are using Hash-Solomon code algorithm is designed to divide data into different parts. If the one data part missing we lost the data information.

4.5 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.6 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

4.6.1 Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

4.6.2 Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

4.7 NON- FUNCTIONAL REQUIREMENTS

4.7.1 Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

4.7.2 Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

4.7.3 Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

4.7.4 Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

4.8 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

4.9 USE CASE DIAGRAM:

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.

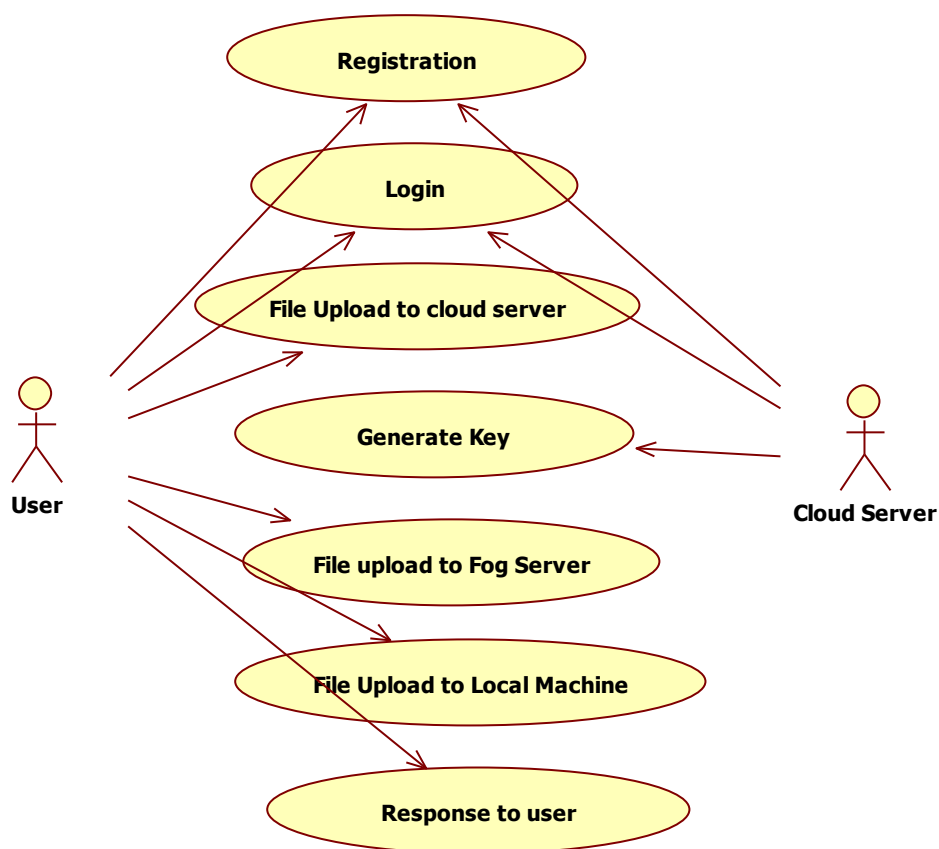


Fig. 4.3:Use case Diagram

4.10 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

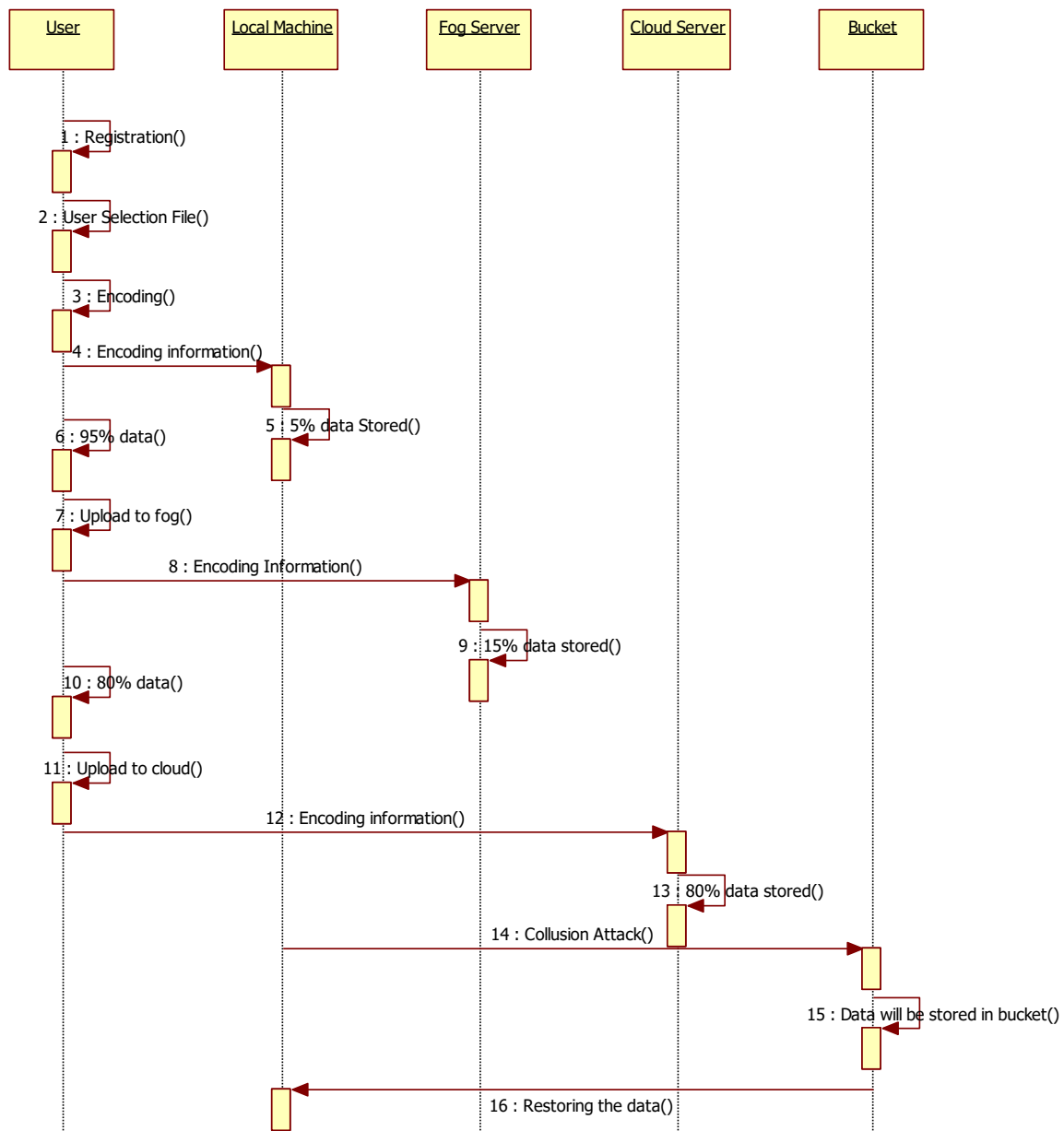


Fig.4.4:Sequence Diagram

4.11 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

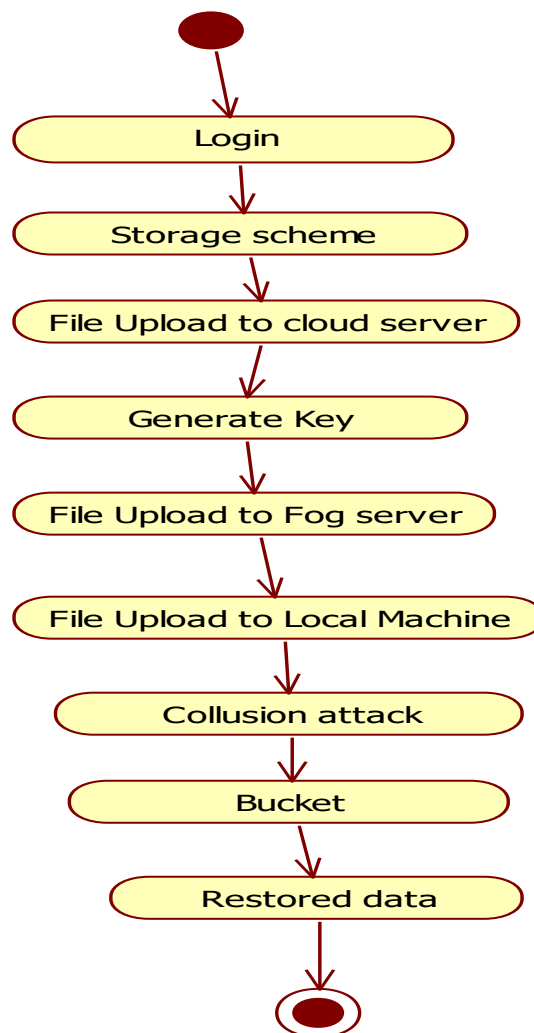


Fig. 4.5: Activity Diagram

4.12 COLLABORATION DIAGRAM:

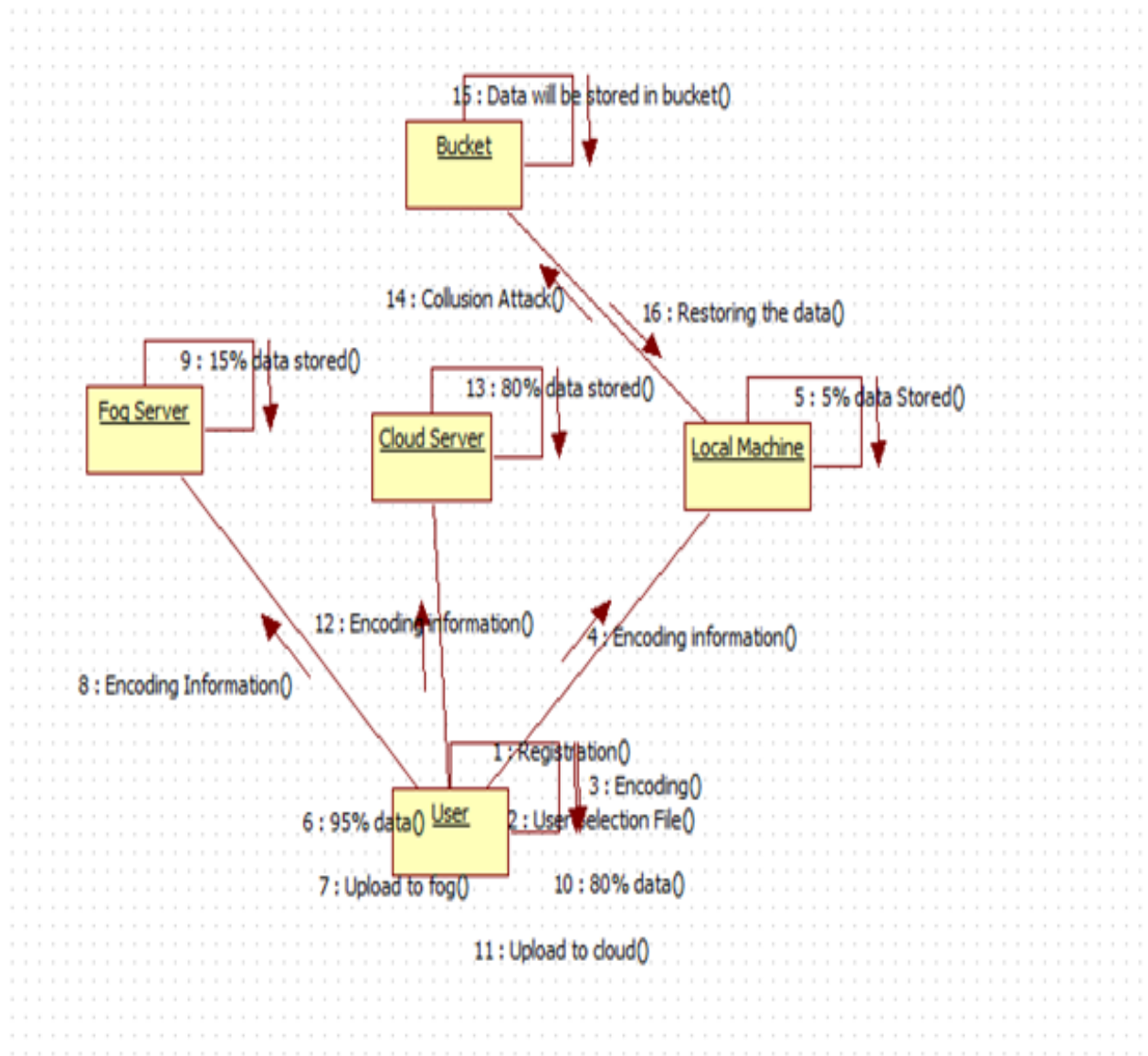


Fig. 4.6: Collaboration Diagram

4.13 SYSTEM STUDY :

4.13.1 FEASIBILITY STUDY :

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

4.13.1.1 ECONOMICAL FEASIBILITY :

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Economical feasibility of implementing secure storage solutions in the cloud involves analyzing the financial viability of the project. This includes assessing the costs associated with infrastructure, operating expenses, total cost of ownership, and potential return on investment. Organizations need to consider the balance between the initial investment and long-term benefits, such as improved data security, scalability, and cost savings. Evaluating factors like scalability, flexibility, and risk mitigation strategies is essential in determining whether the benefits outweigh the expenses and justify the investment in secure cloud storage solutions.

4.13.1.2 TECHNICAL FEASIBILITY :

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.13.1.3 SOCIAL FEASIBILITY :

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Social feasibility of implementing secure storage solutions in the cloud pertains to evaluating how the proposed project aligns with the values, norms, and expectations of the stakeholders and society at large. It involves assessing the potential impact of the project on various social factors, such as privacy, data security, accessibility, and user acceptance. Organizations need to consider factors like public perception, trust, and cultural considerations when introducing new technologies and storing sensitive data in the cloud. Addressing concerns related to data privacy, security breaches, and regulatory compliance is crucial for gaining stakeholders' trust and ensuring social acceptance of the secure storage solutions. Additionally, engaging with stakeholders through communication, education, and transparency initiatives can help mitigate resistance and foster a positive social environment for the implementation of cloud-based storage solutions.

CHAPTER 5

IMPLEMENTATION DETAILS

Designing and implementing a distributed learning management system (LMS) involves several key considerations, including scalability, fault tolerance, data consistency, and performance. Here's an overview of the implementation details for a distributed LMS.

5.1 ARCHITECTURE

Multi-stage cloud security for cloud data using amalgamate data refers to a comprehensive approach to securing data stored in cloud environments. This involves implementing security measures at various stages of data handling, such as creation, storage, access, and deletion. The term "amalgamate data" likely refers to integrating different types of data or combining data from various sources to strengthen security measures.

In this approach, encryption techniques are used to protect data both at rest and in transit, access controls and permissions are set up to ensure that only authorized users can access specific data, and identity verification mechanisms such as multi-factor authentication are implemented. Network security measures such as firewalls and intrusion detection/prevention systems are utilized to monitor and control traffic entering and leaving the cloud environment.

Additionally, monitoring and logging mechanisms are put in place to collect and analyze logs and monitoring data for suspicious activities or security incidents. Threat intelligence feeds are incorporated to identify and respond to emerging threats in real-time, while data loss prevention mechanisms prevent unauthorized access, sharing, or leakage of sensitive data. Security automation and orchestration are employed to automate security processes and orchestrate responses to security events, improving efficiency and reducing response times.

5.2 SYSTEM TESTING :

5.2.1 TESTING PROCESS :

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.

5.2.2 TYPES OF TESTS :

5.2.2.1 UNIT TESTING :

Unit testing involves the design of test cases that validate that the internal Program logic is functioning properly, and that program input produces 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 business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2.2.2 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 satisfaction, as shown by successfully 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.

5.2.2.3 FUNCTIONAL TESTING :

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 is used to identified classes of valid input must be accepted. Invalid Input is used to identified classes of invalid input must be rejected. Functions is used to identified functions must be exercised. Output is used to identify classes of application outputs.

Systems/Procedures is used to 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.

5.2.2.4 SYSTEM 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.

5.2.2.5 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 it's purpose. It is used to test areas that cannot be reached from a black box level.

In white-box testing, testers analyze the control flow, data flow, and algorithms within the software to design test cases that thoroughly exercise different paths and conditions in the code. This allows for a deeper understanding of how the software functions and enables testers to identify potential vulnerabilities, logical errors, and inefficiencies in the codebase.

5.2.2.6 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, such as specification or requirement 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.

5.2.2.7 TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

1. All field entries must work properly.
2. Pages must be activated from the identified link.
3. The entry screen, messages and responses must not be delayed. Features to be tested.
4. Verify that the entries are of the correct format No duplicate entries should be allowed.
5. All links should take the user to the correct page.

5.2.2.8 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or

software applications.

5.2.2.9 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.

5.2.2.10 ALPHA TESTING :

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are acceptable and preload work.

5.2.2.11 BETA TESTING :

In software development, a beta test is the second phase of software testing in which a sampling of the intended audience tries the product out. Beta testing can be considered "per-release testing." Beta test versions of software are now distributed to curriculum specialists and teachers to give the program a "real-world" test.

CHAPTER 6

RESULT AND DISCUSSION

This document delves into recent research concerning the protection of data within the increasingly prevalent domain of multi-cloud computing. It underscores the crucial role of encryption in securing sensitive information, examining both established solutions such as AES and emerging possibilities like homomorphic encryption. While acknowledging the efficiency of minimal encryption times associated with algorithms like SHA-1, the paper highlights the paramount importance of selecting algorithms tailored to the specific data being safeguarded and the evolving threat landscape. In addition to encryption, the document explores recommended practices for key management and recognizes potential performance and recovery considerations linked to robust data security measures. By delving into these aspects and citing specific references from the reviewed research, this paper aims to contribute significantly to the ongoing discourse on enhancing data security in multi-cloud environments

6.1 FILE UPLOAD

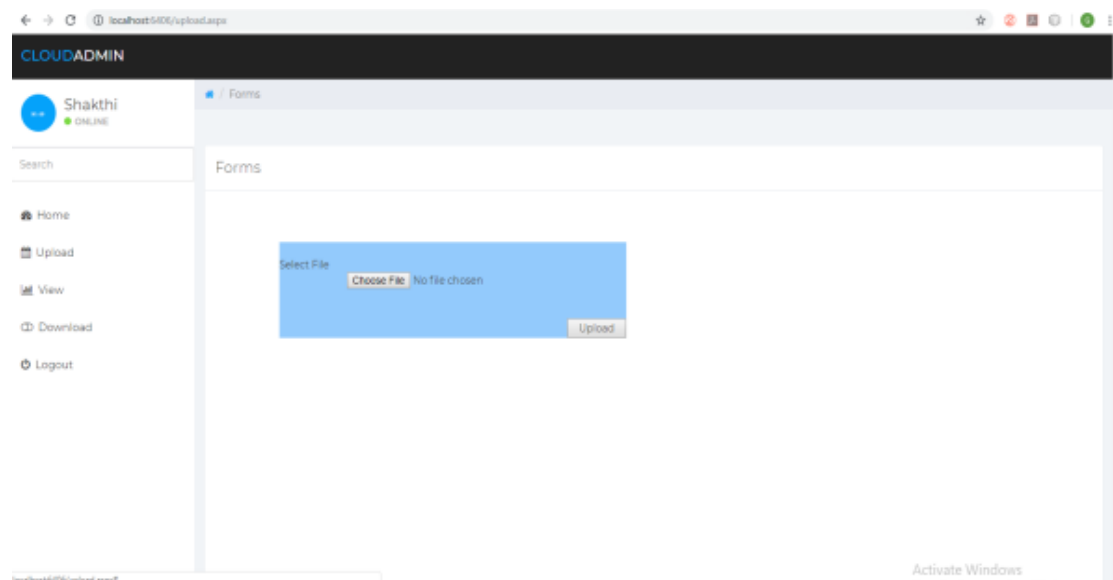


Fig. 6.1: File Upload

6.2 STORAGE SCHEME

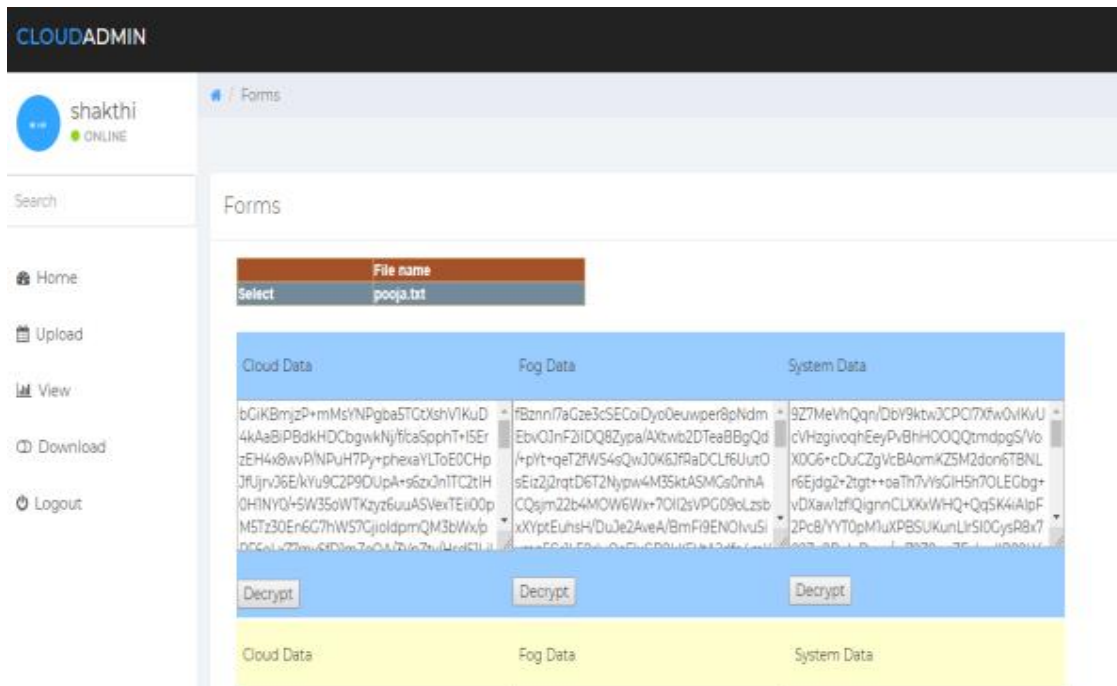


Fig. 6.2: Storage Scheme

6.3 RECOVERY MODULE

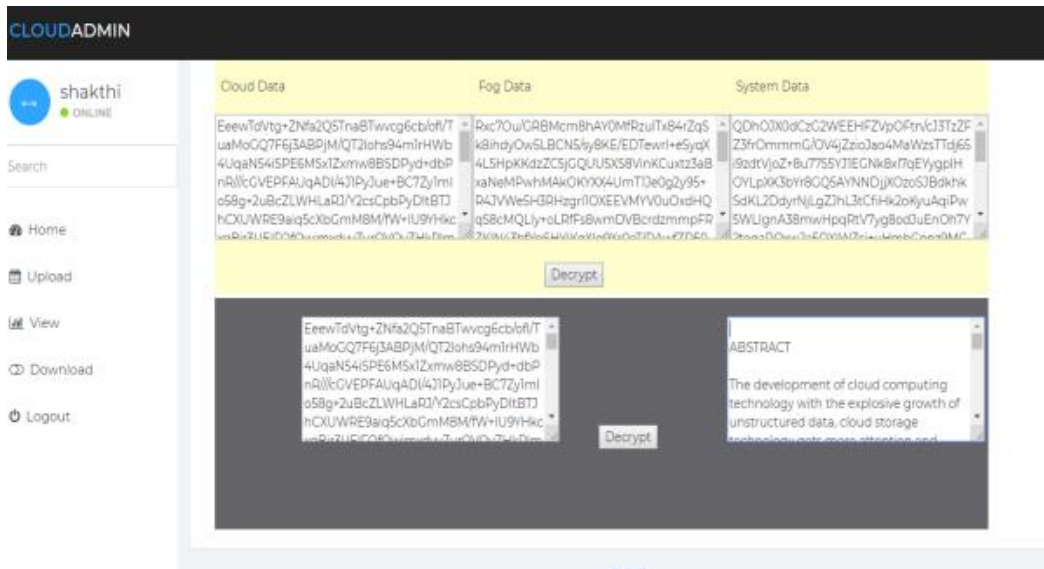


Fig. 6.3: Recover Module

CHAPTER 7

CONCLUSION

Numerous organizations opt for secure storage solutions to enhance the security of their stored data within the cloud. This strategic choice addresses the vulnerability associated with storing entire documents in a single cloud server, which poses a considerable risk. With the adoption of this approach, security is significantly improved. The method involves dispersing securely encrypted files across different servers within the cloud, utilizing various storage techniques. Rather than storing an entire document in a single cloud system, the document is divided into multiple segments, hashed, encrypted, and distributed across multiple cloud servers. This approach not only enhances security through advanced encryption measures but also ensures data integrity, offering organizations a valuable advantage in safeguarding their files. Improved data redundancy is achieved by dispersing encrypted files across multiple cloud servers, mitigating the risk of data loss in the event of server failures or outages. Enhanced scalability is facilitated by secure storage solutions in the cloud, allowing organizations to scale storage capacity up or down as needed without significant upfront investment. Cost-efficiency is another benefit, as organizations can leverage pay-as-you-go pricing models and avoid additional costs for hardware procurement and maintenance. Regulatory compliance is addressed through built-in compliance features, ensuring that sensitive data is handled in accordance with legal requirements such as GDPR, HIPAA, and PCI DSS. Cloud-based storage solutions also offer centralized management interfaces and robust monitoring tools, enabling efficient management of storage infrastructure and enforcement of security policies. However, organizations must be mindful of potential challenges such as data migration complexities, vendor lock-in, and the need for robust security controls to protect against data breaches and unauthorized access. Overall, secure storage solutions in the cloud provide a reliable, scalable, and cost-effective means of enhancing data security, integrity, and regulatory compliance while adapting to evolving business needs.

REFERENCES

- [1]. H.A. Bilal, M. Ilyas, Q. Tariq and M. Hummayun, "Requirements validation techniques: An empirical study", *International Journal of Computer Applications*, vol. 148, no. 14, 2016.
- [2]. M. Kamalrudin and S. Sidek, "A review on software requirements validation and consistency management", *International Journal of Software Engineering and Its Applications*, vol. 9, no. 10, pp. 39-58, 2015.
- [3]. K. Chandra, G. Kapoor, R. Kohli and A. Gupta, "Improving software quality using machine learning" in 2016 ICICCS-INBUSH, IEEE, pp. 115-118, February 2016.
- [4]. S.T. Demirel and R. Das, "Software requirement analysis: Research challenges and technical approaches", *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1-6, 2018, March.
- [5]. T. Kiran, S. Farhan, H. Tauseef and M. A. Fahiem, "Requirement validation model for virtual distributed system", *Int. J. Multidiscip. Sci. Eng.*, vol. 4, no. 8, pp. 34-42, 2013.
- [6]. Dharendra Pandey, Waris Khan and Vandana Pandey, *ROLE OF REQUIREMENT VALIDATION IN REQUIREMENT DEVELOPMENT*, 2012.
- [7]. U.A. Raja, "Empirical studies of requirements validation techniques", *2009 2nd International Conference on Computer Control and Communication*, pp. 1-9, 2009, February.
- [8]. X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning", *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1-130, 2009.
- [9]. Sun, Pan Jun. "Privacy protection and data security in cloud computing: a survey, challenges, and solutions." *IEEE Access* 7 (2019)
- [10]. Tutubala, Ntebaleng, and Topside E. Mathonsi. "A Hybrid Framework to Improve Data Security in Cloud Computing." *Big Data, Knowledge and Control Systems Engineering (BdKCSE)*. IEEE, 2021.

APPENDIX

A. SOURCE CODE

1.Login Page

```
using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;


public partial class Login : System.Web.UI.Page

{

    SqlConnection con;

    SqlDataReader rs;

    string conn = "Data Source=LAPTOP-APR8BG12;Initial Catalog=cloud1;Integrated Security=True";

    protected void Page_Load(object sender, EventArgs e)

    {


    }

    protected void Button2_Click(object sender, EventArgs e)

    {

        Response.Redirect("Register.aspx");

    }

}
```

```

protected void Button1_Click(object sender, EventArgs e)
{
    con = new SqlConnection(conn);
    con.Open();
    string s1;

    s1 = "select * from login1 where mailid='" + TextBox1.Text + "' and pwd='" +
    TextBox2.Text + "'";

    SqlCommand cmd = new SqlCommand(s1, con);
    rs = cmd.ExecuteReader();
    if (rs.Read())
    {

        string id = Convert.ToString(rs.GetValue(1));
        Session["uname"] = id.ToString();
        Session["mailid"] = TextBox1.Text;
        Response.Redirect("Default.aspx");
        rs.Close();
    }
    else
    {
        MsgBox.Show("Invalid username or password");
        TextBox1.Text = "";
        TextBox2.Text = "";
        rs.Close();
    }
}

```

```
}
```

```
}
```

2.Register Page

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Data.SqlClient;
```

```
using System.Linq;
```

```
using System.Web;
```

```
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
```

```
public partial class Register : System.Web.UI.Page
```

```
{
```

```
    SqlConnection con;
```

```
    SqlDataReader rs;
```

```
    string conn = "Data Source=LAPTOP-APR8BG12;Initial Catalog=cloud1;Integrated  
Security=True";
```

```
    protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
}
```

```
    protected void Button2_Click(object sender, EventArgs e)
```

```
{
```

```
    if (TextBox1.Text == "" || TextBox2.Text == "" || TextBox1.Text == "" || TextBox2.Text  
    == "" || TextBox2.Text == "")
```

```
{
```

```

MsgBox.Show("Fill All Datas");
}

else if (TextBox2.Text != TextBox5.Text)
{
MsgBox.Show("Password not matching");
}

else
{
con = new SqlConnection(conn);

string s2 = "select mailid from login1 where mailid = ('" + TextBox1.Text + "')";

con.Open();

SqlCommand cmd1 = new SqlCommand(s2, con);

SqlDataReader reader = cmd1.ExecuteReader();

if (reader.HasRows)
{
MsgBox.Show("Already Exist");
}

else
{
con = new SqlConnection(conn);

string s1 = "insert into login1 values('" + TextBox1.Text + "','" + TextBox3.Text + "','" +
+ TextBox4.Text + "','" + TextBox2.Text + "')";

con.Open();

SqlCommand cmd = new SqlCommand(s1, con);

cmd.ExecuteNonQuery();

TextBox1.Text = "";

```

```

TextBox2.Text = "";
TextBox3.Text = "";
TextBox4.Text = "";
TextBox5.Text = "";
Response.Redirect("Login.aspx");
}
}
}

protected void Button1_Click(object sender, EventArgs e)
{
Response.Redirect("Login.aspx");
}
}

```

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Register.aspx.cs"
Inherits="Register" %>

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head runat="server">

```

```

<meta charset="utf-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1">

```

```

<title>Lumino - Login</title>

```

```

<link href="css/bootstrap.min.css" rel="stylesheet">

```

```

<link href="css/datepicker3.css" rel="stylesheet">

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

<!--[if lt IE 9]>

<script src="js/html5shiv.js"></script>

<script src="js/respond.min.js"></script>

<![endif]-->

</head>

<body>

<div class="row">

<div class="col-xs-10 col-xs-offset-1 col-sm-8 col-sm-offset-2 col-md-4 col-md-offset-4">

<div class="login-panel panel panel-default">

<div class="panel-heading">Register</div>

<div class="panel-body">

<form id="form1" runat="server" role="form">

<fieldset>

<div class="form-group">

<asp:TextBox ID="TextBox1" runat="server" class="form-control" placeholder="E-mail" name="email" ></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox ID="TextBox3" runat="server" class="form-control" placeholder="Name" name="name" ></asp:TextBox>

</div>

<div class="form-group">

<asp:TextBox ID="TextBox4" runat="server" class="form-control" placeholder="Mobile Number" name="mob" ></asp:TextBox>

```

<div class="form-group">

```
<asp:TextBox ID="TextBox2" runat="server" class="form-control"
placeholder="Password" name="password" type="password" ></asp:TextBox>
```

<div class="form-group">

```
<asp:TextBox ID="TextBox5" runat="server" class="form-control" placeholder="Re
EnterPassword" name="password1" type="password" ></asp:TextBox>
```

</div>

```
<div class="form-group">
```

```
<asp:Button ID="Button2" runat="server" class="btn btn-primary" Text="Register"
OnClick="Button2_Click" />
```

[illegible]

```
<asp:Button ID="Button1" runat="server" class="btn btn-primary" Text="Login"
Width="101px" OnClick="Button1_Click" />
```

<div class="form-group">

</div>

</form>

<!-- /.col-->-->

</div><!-- /.row -->

```
<script src="js/jquery-1.11.1.min.js"></script>
```

```
<script src="js/bootstrap.min.js"></script>
```

```
</body>
```

```
</html>
```

3.HomePage

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Web;
```

```
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
```

```
public partial class Home : System.Web.UI.MasterPage
```

```
{
```

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
Label1.Text = Session["uname"].ToString();
```

```
}
```

```
}
```

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Home.master.cs"  
Inherits="Home" %>
```

```
<!DOCTYPE html>
```

https://fonts.googleapis.com/css?family=Montserrat:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

<!--[if lt IE 9]>

<script src="js/html5shiv.js"></script>

<script src="js/respond.min.js"></script>

<![endif]-->

</head>

<body>

<nav class="navbar navbar-custom navbar-fixed-top" role="navigation">

<div class="container-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#sidebar-collapse">Toggle navigation

</button>

CloudAdmin

</div>

</div><!-- /.container-fluid -->

</nav>

<div id="sidebar-collapse" class="col-sm-3 col-lg-2 sidebar">

<div class="profile-sidebar">

<div class="profile-userpic">

</div>

<div class="profile-usertitle">

```

<div class="profile-usertitle-name">

<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label></div>

<div class="profile-usertitle-status"><span class="indicator label-
success"></span>Online</div>

</div>

<div class="clear"></div>

</div>

<div class="divider"></div>

<div class="form-group">

<input type="text" class="form-control" placeholder="Search">

</div>

<ul class="nav menu">

<li><a href="Default.aspx"><em class="fa fa-dashboard">&nbsp;</em>
Home</a></li>

<li><a href="upload.aspx"><em class="fa fa-calendar">&nbsp;</em>
Upload</a></li>

<li><a href="view.aspx"><em class="fa fa-bar-chart">&nbsp;</em> View</a></li>

<li><a href="download.aspx"><em class="fa fa-toggle-off">&nbsp;</em>
Download</a></li>

<li><a href="Login.aspx"><em class="fa fa-power-off">&nbsp;</em> Logout</a></li>

</ul>

</div><!--/.sidebar-->

<div class="col-sm-9 col-sm-offset-3 col-lg-10 col-lg-offset-2 main">

<div class="row">

<ol class="breadcrumb">

<li><a href="#">

<em class="fa fa-home"></em>

```

```

</a></li>

<li class="active">Forms</li>

</ol>

</div><!--/.row-->

<div class="row">

<div class="col-lg-12">

<h1 class="page-header"></h1>

</div>

</div><!--/.row-->

<div class="row">

<div class="col-lg-12">

<div class="panel panel-default">

<div class="panel-heading">Forms</div>

<div class="panel-body">

<div class="col-md-6">

<form id="form1" runat="server">

<div>

<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>

</div>

</form>

</div>

</div>

</div>

```

```

</div><!-- /.panel-->

</div><!-- /.col-->

<div class="col-sm-12">

<p class="back-link"> <a href="#">Cloud</a></p>

</div>

</div><!-- /.row -->

</div><!-- /.main-->

<script src="js/jquery-1.11.1.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/chart.min.js"></script>

<script src="js/chart-data.js"></script>

<script src="js/easypiechart.js"></script>

<script src="js/easypiechart-data.js"></script>

<script src="js/bootstrap-datepicker.js"></script>

<script src="js/custom.js"></script>

</body>

</html>

```

4.Upload Page

```

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

```

```

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;


public partial class upload : System.Web.UI.Page
{
    public static string dt;

    public static string conn = "Data Source=LAPTOP-APR8BG12;Initial
    Catalog=cloud1;Integrated Security=True";

    SqlConnection con;

    SqlCommand cmd;

    SqlDataReader rs;

    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        if (FileUpload1.HasFile)
        {
            try
            {
                if (FileUpload1.PostedFile.ContentType == "text/plain")
                {
                    if (FileUpload1.PostedFile.ContentLength < 102400)

```

```

{
    string filename = Path.GetFileName(FileUpload1.FileName);

    FileUpload1.SaveAs(Server.MapPath("~/temp1/") + filename);

    var fileStream = new FileStream(Server.MapPath("~/temp1/") + filename,
        FileMode.Open, FileAccess.Read);

    using (var streamReader = new StreamReader(fileStream, Encoding.UTF8))
    {
        dt = streamReader.ReadToEnd();
    }

    string path = Server.MapPath("~/temp1/") + filename;

    FileInfo file = new FileInfo(path);

    if (file.Exists)//check file exsist or not
    {
        file.Delete();

    }

    Cipher c = new Cipher();

    string c_text= Convert.ToString( c.Encrypt(dt, "p@SSword"));

    int a = c_text.Length;

    int x =(int) a / 3;

    string sub = c_text.Substring(0, x);

    string sub1 = c_text.Substring(x, x);

    int x1 = a-(x + x);

    string sub2 = c_text.Substring(x+x, x1);

    string cipherText1 = CryptorEngine.Encrypt(sub, true);

    string cipherText2 = CryptorEngine.Encrypt(sub1, true);

```

```

string cipherText3 = CryptorEngine.Encrypt(sub2, true);

File.WriteAllText(Server.MapPath("~/cloud/") + filename, String.Empty);

using (StreamWriter writer = new StreamWriter(Server.MapPath("~/cloud/") +
filename))

{

writer.Write(cipherText1);


}

File.WriteAllText(Server.MapPath("~/cloud1/") + filename, String.Empty);

using (StreamWriter writer = new StreamWriter(Server.MapPath("~/cloud1/") +
filename))

{

writer.Write(cipherText2);


}

File.WriteAllText(Server.MapPath("~/temp/") + filename, String.Empty);

using (StreamWriter writer = new StreamWriter(Server.MapPath("~/temp/") +
filename))

{

writer.Write(cipherText3);

}

File.WriteAllText(Server.MapPath("~/bin/") + filename, String.Empty);

using (StreamWriter writer = new StreamWriter(Server.MapPath("~/bin/") +
filename))

{

writer.Write(cipherText3);

```

```

}

con = new SqlConnection(conn);

con.Open();

cmd = new SqlCommand("insert into file_m values('" + Session["mailid"].ToString() +
"', '" + filename + "')", con);

cmd.ExecuteNonQuery();

// FileUpload1.SaveAs(Server.MapPath("~/") + filename);

StatusLabel.Text = "Upload status: File uploaded!";

}

else

StatusLabel.Text = "Upload status: The file has to be less than 100 kb!";

}

else

StatusLabel.Text = "Upload status: Only Text files are accepted!";

}

catch (Exception ex)

{

StatusLabel.Text = "Upload status: The file could not be uploaded. The following
error occurred: " + ex.Message;

}

}

}

}

```

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Home.master"
AutoEventWireup="true" CodeFile="upload.aspx.cs" Inherits="upload" %>

```

```

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">

<div>

<table style="width: 100%; height: 774px;">

<tr>

<td style="width: 77px; height: 54px"></td>

<td style="width: 498px; height: 54px"></td>

<td style="height: 54px"></td>

</tr>

<tr>

<td style="width: 77px; height: 557px"></td>

<td style="width: 498px; height: 557px">

<asp:Panel ID="Panel1" runat="server" Height="737px" BorderColor="#CCFFFF">

<table style="width:100%; background-color: #99CCFF;">

<tr>

<td style="width: 184px; height: 20px;"></td>

<td style="width: 227px; height: 20px;">

<asp:Label ID="StatusLabel" runat="server" ForeColor="Red"></asp:Label>

</td>

<td style="height: 20px"></td>

</tr>

<tr>

<td style="width: 184px">

<asp:Label ID="Label1" runat="server" Text="Select File"></asp:Label>

</td>

<td style="width: 227px">&nbsp;</td>

```

```

<td>&nbsp;</td>

</tr>

<tr>

<td style="width: 184px">&nbsp;</td>

<td style="width: 227px">

<asp:FileUpload ID="FileUpload1" runat="server" />

</td>

<td>&nbsp;</td>

</tr>

<tr>

<td style="width: 184px">&nbsp;</td>

<td style="width: 227px">&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td style="width: 184px">&nbsp;</td>

<td style="width: 227px">&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<td style="width: 184px">&nbsp;</td>

<td style="width: 227px">&nbsp;</td>

<td>

<asp:Button ID="Button1" runat="server" Text="Upload" Width="85px"
OnClick="Button1_Click" />

</td>

```

```

</tr>

</table>

</asp:Panel>

</td>

<td style="height: 557px"></td>

</tr>

<tr>

<td style="width: 77px">&nbsp;</td>

<td style="width: 498px">&nbsp;</td>

<td>&nbsp;</td>

</tr>

</table>

</div>

</asp:Content>

```

5.View Page

```

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.IO;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.UI;

```

```
using System.Web.UI.WebControls;
```

```
public partial class view : System.Web.UI.Page
```

```
{
```

```
public static int xa = 0;
```

```
protected void Page_Load(object sender, EventArgs e)
```

```
{
```

```
if (!IsPostBack)
```

```
{
```

```
if (string.IsNullOrEmpty(Session["mailid"] as string))
```

```
{
```

```
Response.Redirect("Login.aspx");
```

```
}
```

```
else
```

```
{
```

```
SqlConnection con = new SqlConnection("Data Source=LAPTOP-APR8BG12;Initial  
Catalog=cloud1;Integrated Security=True");
```

```
string strSQL = "Select * from file_m where uname='" + Session["mailid"].ToString()  
+ "'";
```

```
SqlDataAdapter dt = new SqlDataAdapter(strSQL, con);
```

```
DataSet ds = new DataSet();
```

```
dt.Fill(ds, "uname");
```

```
con.Close();
```

```
GridView1.DataSource = ds;
```

```
GridView1.DataBind();
```

```
}
```

```

}

}

protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    TextBox1.Text = "";

    TextBox4.Text = "";
    TextBox5.Text = "";
    TextBox6.Text = "";
    TextBox7.Text = "";
    TextBox8.Text = "";
    TextBox9.Text = "";
    TextBox10.Text = "";

    string f_name = Convert.ToString(GridView1.SelectedRow.Cells[1].Text);

    string dt, dt1, dt2;

    var fileStream = new FileStream(Server.MapPath("~/cloud/" + f_name),
    FileMode.Open, FileAccess.Read);

    using (var streamReader = new StreamReader(fileStream, Encoding.UTF8))
    {
        dt = streamReader.ReadToEnd();
    }

    var fileStream1 = new FileStream(Server.MapPath("~/cloud1/" + f_name),
    FileMode.Open, FileAccess.Read);

    using (var streamReader = new StreamReader(fileStream1, Encoding.UTF8))
    {

```

```

dt1 = streamReader.ReadToEnd();

}

string path1 = Server.MapPath("~/temp/" + f_name);

FileInfo file1 = new FileInfo(path1);

if (file1.Exists)//check file exists or not

{

var fileStream2 = new FileStream(Server.MapPath("~/temp/" + f_name),
    FileMode.Open, FileAccess.Read);

using (var streamReader = new StreamReader(fileStream2, Encoding.UTF8))

{

dt2 = streamReader.ReadToEnd();

}

}

else

{

var fileStream2 = new FileStream(Server.MapPath("~/bin/" + f_name),
    FileMode.Open, FileAccess.Read);

using (var streamReader = new StreamReader(fileStream2, Encoding.UTF8))

{

dt2 = streamReader.ReadToEnd();

}

}

TextBox1.Text = dt;

TextBox4.Text = dt1;

TextBox5.Text = dt2;

```

```

}

protected void Button4_Click(object sender, EventArgs e)

{
    string plain1 = CryptorEngine.Decrypt(TextBox1.Text, true);
    TextBox6.Text = plain1;

    xa = xa + 1;

    if(xa==3)
    {
        xa = 0;

        Button6.Visible = true;
    }
}

protected void Button2_Click(object sender, EventArgs e)

{
    string plain1 = CryptorEngine.Decrypt(TextBox4.Text, true);
    TextBox7.Text = plain1;

    xa = xa + 1;

    if (xa == 3)
    {
        xa = 0;

        Button6.Visible = true;
    }
}

```

```

protected void Button3_Click(object sender, EventArgs e)
{
    string plain1 = CryptorEngine.Decrypt(TextBox5.Text, true);
    TextBox8.Text = plain1;

    xa = xa + 1;

    if (xa == 3)
    {
        xa = 0;

        Button6.Visible = true;
    }
}

protected void Button6_Click(object sender, EventArgs e)
{
    string txt = TextBox6.Text + TextBox7.Text + TextBox8.Text;

    TextBox9.Text = txt;
}

protected void Button5_Click(object sender, EventArgs e)
{
    Cipher c = new Cipher();

    string p_text = Convert.ToString(c.Decrypt(TextBox9.Text, "p@SSword"));

    TextBox10.Text = p_text;
}
}

```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Home.master"
AutoEventWireup="true" CodeFile="view.aspx.cs" Inherits="view" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
```

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
AutoGenerateSelectButton="True" BackColor="#DEBA84" BorderColor="#DEBA84"
BorderStyle="None" BorderWidth="1px" CellPadding="3" CellSpacing="2"
OnSelectedIndexChanged="GridView1_SelectedIndexChanged" Width="398px">
```

<FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />

```
<HeaderStyle BackColor="#A55129" Font-Bold="True" ForeColor="White" />
```

<PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />

```
<RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
```

```
<SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
```

```
<SortedAscendingCellStyle BackColor="#FFF1D4" />
```

```
<SortedAscendingHeaderStyle BackColor="#B95C30" />
```

```
<SortedDescendingCellStyle BackColor="#F1E5CE" />
```

```
<SortedDescendingHeaderStyle BackColor="#93451F" />
```

<Columns>

```
<asp:BoundField DataField="f_name" HeaderText="File name" />
```

</Columns>

</asp:GridView>

[illegible]

|
 |

```
<asp:Panel ID="Panel10" runat="server" BackColor="#99CCFF" Height="253px">
```

```

<br />

&nbsp;

<asp:Label ID="Label4" runat="server" Text="Cloud Data"></asp:Label>

<br />

<br />

<asp:TextBox ID="TextBox1" runat="server" Height="134px" TextMode="MultiLine"
Width="315px"></asp:TextBox>

<br />

<br />

<asp:Button ID="Button4" runat="server" OnClick="Button4_Click" Text="Decrypt" />

</asp:Panel>

</td>

<td style="height: 259px">

<asp:Panel ID="Panel2" runat="server" BackColor="#99CCFF" Height="253px">

<br />

&nbsp;

<asp:Label ID="Label2" runat="server" Text="Fog Data"></asp:Label>

<br />

<br />

<asp:TextBox ID="TextBox4" runat="server" Height="132px" TextMode="MultiLine"
Width="315px"></asp:TextBox>

<br />

<br />

<asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Decrypt" />

</asp:Panel>

</td>

```

```

<td style="height: 259px">

<asp:Panel ID="Panel3" runat="server" BackColor="#99CCFF" Height="253px">

&nbsp;<br />

<asp:Label ID="Label3" runat="server" Text="System Data"></asp:Label>

<br />

<br />

<asp:TextBox ID="TextBox5" runat="server" Height="131px" TextMode="MultiLine"
Width="315px"></asp:TextBox>

<br />

<br />

<asp:Button ID="Button3" runat="server" OnClick="Button3_Click" Text="Decrypt" />

</asp:Panel>

</td>

</tr>

<tr>

<td style="height: 199px">

<asp:Panel ID="Panel11" runat="server" BackColor="#FFFFCC" Height="253px">

<br />

&nbsp;<br />

<asp:Label ID="Label5" runat="server" Text="Cloud Data"></asp:Label>

<br />

<br />

<asp:TextBox ID="TextBox6" runat="server" Height="134px" TextMode="MultiLine"
Width="315px"></asp:TextBox>

<br />

<br />

```

[illegible]

```
<asp:TextBox ID="TextBox8" runat="server" Height="134px" TextMode="MultiLine" Width="315px"></asp:TextBox>
```


</asp:Panel>

</td>

|

```
<td colspan="3">
```

```
<asp:Panel ID="Panel14" runat="server" BackColor="#656468" Height="248px">
```


[illegible]

```
<asp:TextBox ID="TextBox9" runat="server" Height="134px" TextMode="MultiLine" Width="315px"></asp:TextBox>
```

[illegible]

```
<asp:Button ID="Button5" runat="server" OnClick="Button5_Click" Text="Decrypt" />
```

[illegible]

```
<asp:TextBox ID="TextBox10" runat="server" Height="134px" TextMode="MultiLine" Width="315px"></asp:TextBox>
```

</asp:Panel>

</table>

</asp:Content>

B. SCREENSHOTS

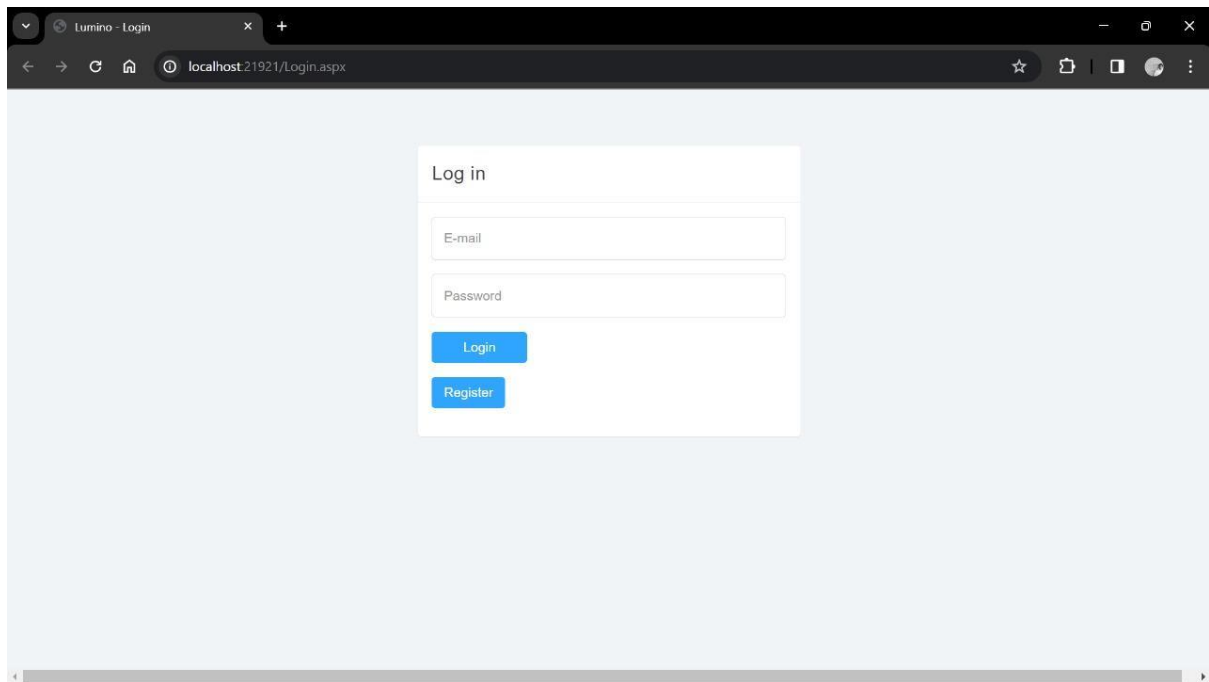


Fig. B.1:Login Page

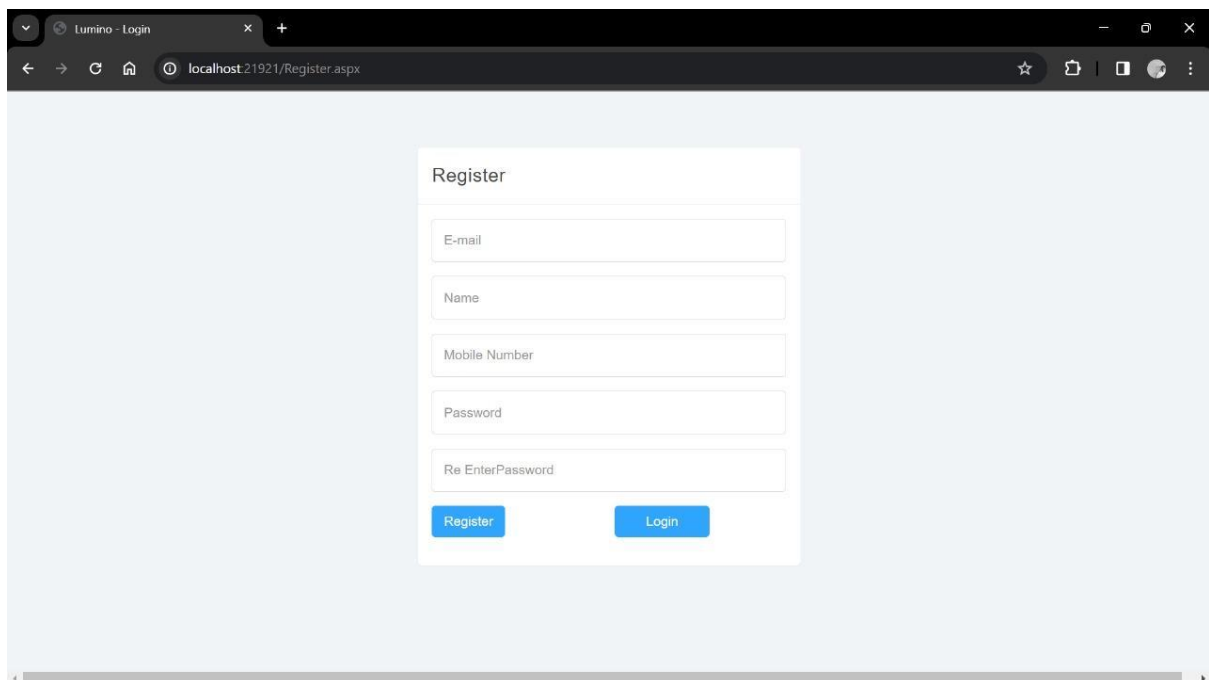


Fig. B.2:Register Page

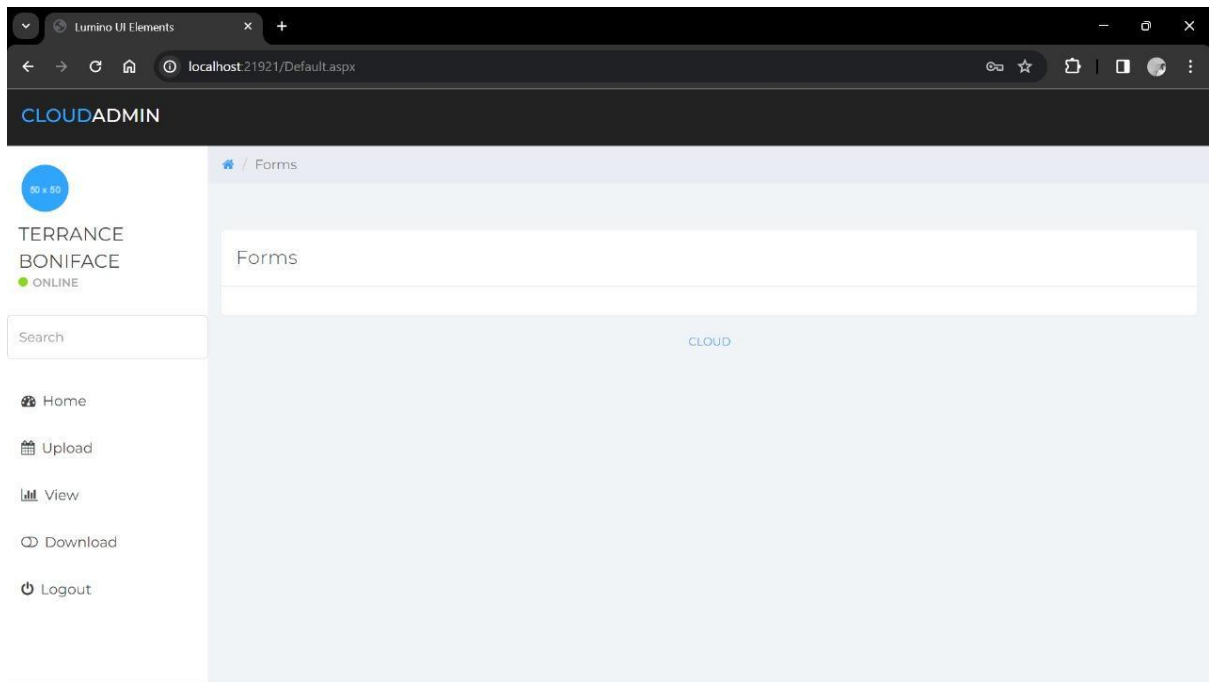


Fig. B.3:Home Page

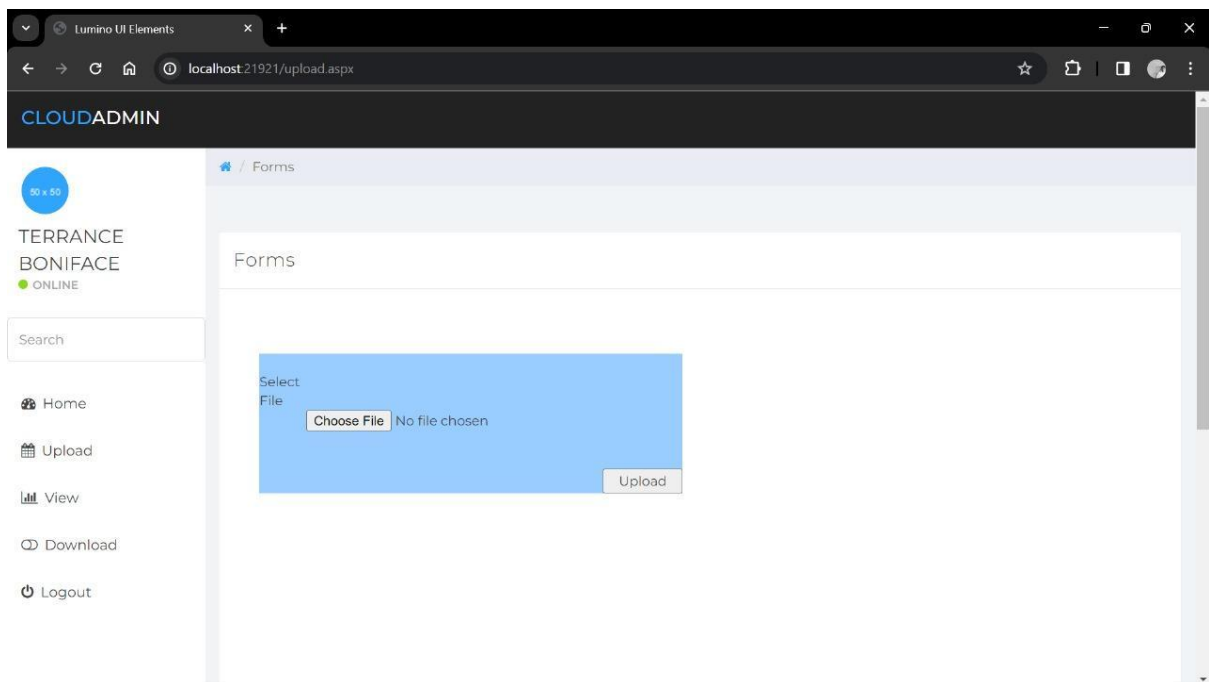


Fig. B.4:Upload Page

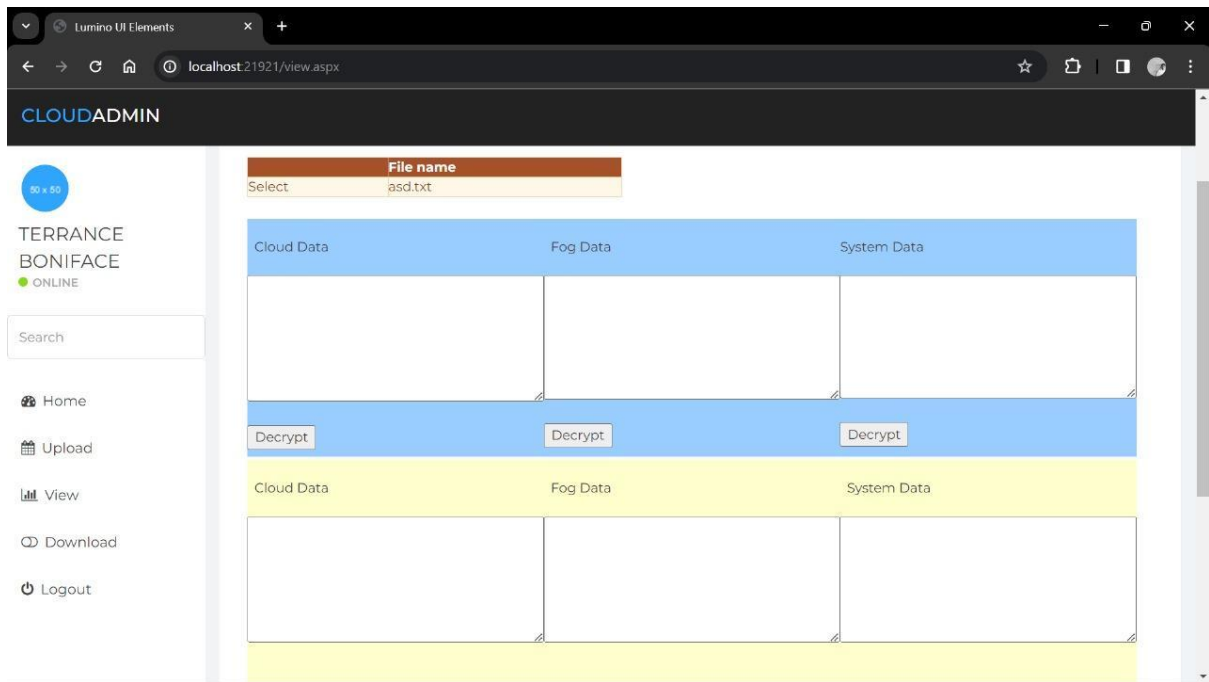


Fig. B.5:View Page

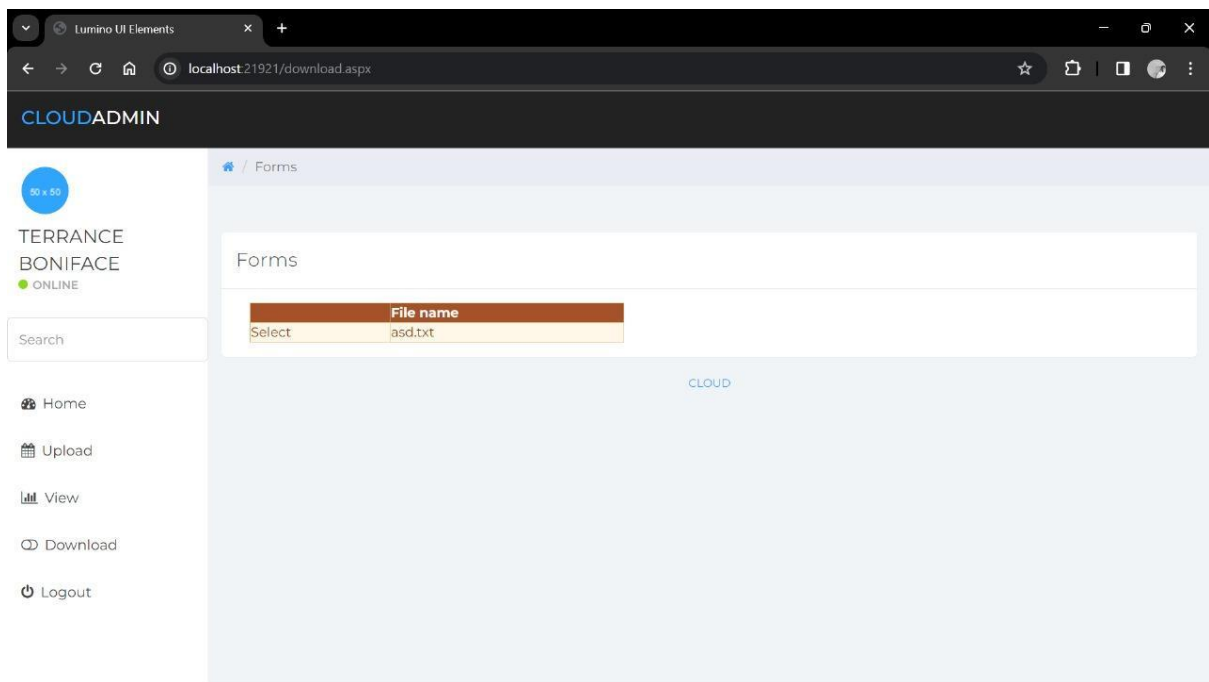


Fig. B.6:Download Page

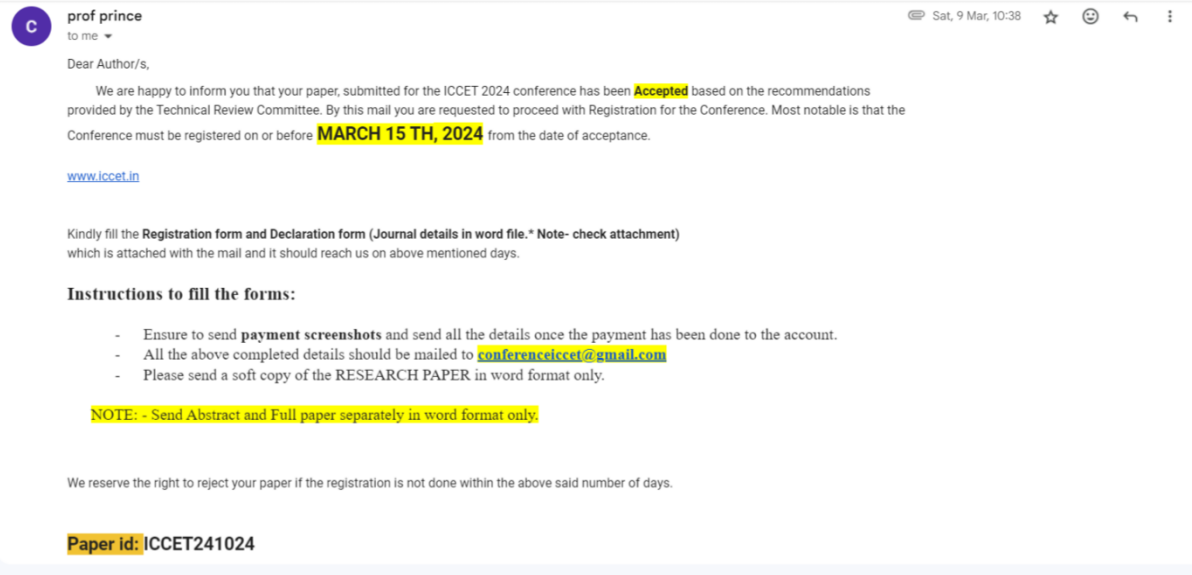


Fig. B.7:Project Acceptance

Using amalgamated data, a multi-stage cloud security system for cloud data

R. Aishwarya
Assistant Professor
Department of CSE
Sathyabama Institute of Science
and Technology
Chennai, India
aishwaryarajkumar7@gmail.com

Thirupathi Pandi
Student
Department of CSE
Sathyabama Institute of Science
and Technology
Chennai, India
pandigeetha1994@gmail.com

Terrance Boniface
Student
Department of CSE
Sathyabama Institute of Science
and Technology
Chennai, India
terranceboniface007@gmail.com

Abstract: The advancement of cloud computing technology has led to an increasing focus on and improvement of cloud storage technology due to the rapid growth of unstructured data. A cloud based company faces challenges when it comes to expand its services. Cloud privacy protection strategies typically rely heavily on Encryption. In a current storage system all of the user's data is stored exclusively on cloud servers, which could pose a risk to privacy if the user loses control over the data. To address the challenge of losing the data three-tiered storage architecture using fog computing has been implemented. The proposed framework has the ability to maximize the benefits of cloud storage while ensuring the confidentiality of data is maintained. In our approach, we implemented the Hash-Solomon code algorithm to divide the data into multiple segments. The absence of even a single data segment can result in the loss of valuable information, Our framework incorporates bucket-based algorithms to enhance the security of the data and integrity of the data. Based on computational intelligence, this algorithm poses the capability to calculate the distribution ratio that is stored across cloud, fog and local machine, respectively.

Keywords: *Multi Stage, Cloud Security, Amalgamate Data, Cloud Computing*

I INTRODUCTION

Cloud computing in computer science pertains to a particular type of cloud computing where computer services are provided similarly to how electricity is supplied. This enables outsourced clients to utilize these services without any concerns or responsibilities. They need to pay each month what he spent changed into like cloud computing; The user can simply use garage, computing power, or mainly designed improvement surroundings no matter how they feature. Internal cloud computing is typically internet based computing. Cloud computing may be observed in both massive scale cloud structure and huge scale structure using statistics structure, get entry to difficulties data obj. This leads to first class output content impact of cloud computing and fog computing on huge statistics settings may additionally vary. Countering an insider attack poses a significant challenge, as traditional methods designed to handle external attacks are ineffective due to the limitations of the current cybersecurity practices. Hence, alternative approaches are required to address this unique type of threat. To overcome this throughout the encryption process, the design incorporates three different components of user statics, each with varying sizes. However, it is worth nothing that these elements are missing some crucial

information. The implementation of a cloud computing model is utilized in these three areas discreetly. The data is safely stored on the cloud server and the individual's local computer.

II LITERATURE REVIEW

Cloud computing and its associated security challenges, along with potential counter measures, are subjects of extensive debate in contemporary research. While numerous surveys on cloud security exist, there is a noticeable gap in establishing a cohesive connection between identified issues and their solutions. Existing surveys often address virtualization issues lacking a unified framework that can Both broaden the definition of cloud security and carefully examine its unique needs.[1] Cloud computing (CC) involves the direct use of network resources like Processing power and data storage without direct user management. It has recently manifested as a combination of public and private data centers providing a unified Internet-based platform. Meanwhile, edge computing represents a growing paradigm bringing computation closer to end-users to enhance response times and optimize bandwidth.[2] This paper explores current trends in cloud security, with the objective of identifying evolving patterns and guiding future research to benefit both potential cloud users and providers.[3]

Cloud computing, functioning as an archetype, provides users with flexible and resource-efficient computing through a shared pool of resources emphasizing a pay-per-use model. The advantages of capital expenditure reduction and operational savings are notable, but certain obstacles, particularly security concerns, limit the widespread adoption of cloud computing. In particular, this article explores security issues that impact different cloud

ecosystem participants, including cloud service providers, data owners, and cloud users. It focuses on the crypto-cloud, which consists of components for computation, communication, and service level agreements. Its goal is to examine the reasons behind and consequences of cyberattacks and suggest improvements that will improve security.[4] Cloud computing has gained widespread attention in both academic and industrial research. The security issues surrounding this technology, which benefits both users and Cloud Service Providers (CSPs), have been thoroughly researched in the literature. In this systematic literature review (SLR) covering research studies from 2010 to 2020, 80 papers were selected to address specific research questions, revealing seven major security threats to cloud computing services. Notably, data tampering, leakage, data intrusion, and storage-related risks emerged as key concerns, while the SLR also suggests blockchain as a potential technology to mitigate these security issues and highlights areas for future research to enhance data confidentiality, integrity, and availability.[5] Cloud computing stands out as one of the rapidly advancing technologies in the computing realm, presenting various advantages alongside notable security concerns. This study delves into the data security challenges within the multi-tenant environment of cloud computing and proposes strategies to address these issues. Additionally, An overview of cloud computing models, including deployment and service delivery models, is given in the article. Data security is crucial for businesses, since any breach can seriously undermine trust and possibly cause the business to fail..[6]

Examining the various elements of cloud computing while emphasizing the inherent security and privacy issues with these systems is the aim of this study. It

also introduces a novel classification for recent security solutions within this domain. The study also analyzes various security risks that endanger cloud computing services, discusses unresolved problems, and suggests possible future paths. The focus of the study is to get a thorough awareness of the security issues that different cloud entities—such as cloud service providers, data owners, and users—face.[7]

In our technologically driven lives, all facets of daily activities are intertwined with technology, creating a vulnerable environment for attackers to exploit targeted organizations with minimal costs. Detecting such attacks poses a significant challenge for organizations. With the advent of cloud computing, users can store data online, allowing global accessibility; however, concerns about security, particularly for users less informed about the evolving threat landscape, persist. This study proposes cost-effective methods and techniques to safeguard data in an online environment, providing an accessible defense mechanism against potential attackers for all users.[8]

Privacy and security stand as pivotal concerns influencing the acceptance of cloud computing services. Existing research efforts on privacy protection in cloud computing, utilizing access control, attribute-based encryption (ABE), trust, and reputation, have been fragmented and lack cohesive logic. This paper undertakes a systematic review and analysis of these research accomplishments, addressing cloud computing architecture, concepts, and shortcomings, and introducing a privacy protection framework. Additionally, it explores various aspects of privacy protection, including ABE variants, access structures, revocation mechanisms, and proposes future directions and challenges for research in cloud computing privacy. Finally, the paper emphasizes the need for

corresponding privacy protection laws to address technical deficiencies.[9]

Cloud computing has evolved into an essential component of contemporary and prospective information technologies, offering features like information storage and remote access over the internet. While it has proven effective for various services, it is also susceptible to multiple threats. Incidents of fire attacks and data theft have emerged as significant concerns, especially given the confidential and sensitive nature of data stored in the cloud. This paper focuses on identifying and addressing such security challenges, aiming to offer suggestions for mitigating data breach issues.[10]

III EXISTING SYSTEM

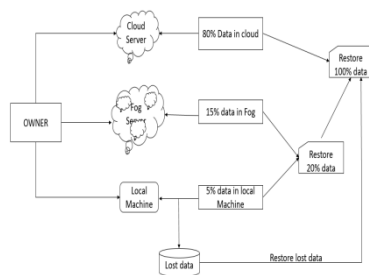
Cloud computing technology has made significant strides in recent years. With the exponential increase in unstructured data, there has been a rising focus and significant advancements in cloud storage technology. It is clear from the existing storage schema that cloud servers are where user data is kept. In the event that the customer is at risk of privacy and loses their ability to manage their data, alterations in risk perception brought on by the data center's cloud extension.

Location awareness and low latency.

IV PROPOSED SYSTEM

The three-layer storage framework capitalizes the cloud storage to protect the privacy of the data. Cloud computing is used by large number of people from different sectors. The three-layer framework divides the data into three distinct parts and stores the data into data parts. Missing of one data part leads to missing data information. In this proposed framework we used bucket concept based algorithms.

- CART idea in our device which minimizes records losses and reduce the procedure time.
- BCH code set of rules. It is highly elastic.
- The BCH code is used in lots of verbal exchange programs and has a small range termination of employment.



4.1 ARCHITECTURE

It is the primary motion that opens with the internet site consumer. Person User need to offer correct touch variety and password" Entered in registration to enter the website. If records User-supplied records matches the database desk and user successful login to

the website online, otherwise a message approximately failed login will seem is displayed and the person must re-input the perfect records. Connection Registration approaches for registering new users also are supplied.

To gain access to the website, a new user must first sign up. To register, click the "Register" button during the login process before proceeding. Complete the necessary actions by entering your full name, password, and contact number. Password confirmation is required by re-entering the password in the designated text field. Upon submitting user information, pressing the "Submit" button triggers changes in both the database and user status, prompting a login. Registered users can then access the web page. Implementing validations in all text fields is crucial for proper page functionality. Any discrepancies during registration, such as an invalid password confirmation, will be flagged, and the

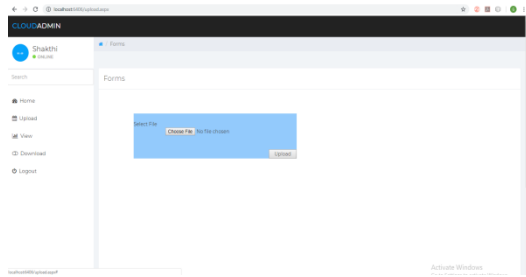
system will display informative messages. Additionally, it is essential to provide information in the password and confirm password fields during registration. A confirmation contact number should be strong and 10 characters long. Failure to meet these criteria will result in a registration failure, requiring the user to register again. If any fields are left empty, a message will be displayed on the web page. If all entered information is correct, the user will be redirected to the account login page.

Within this module, the user has the capability to store documents on three distinct servers. Once the data is uploaded to the cloud, the owner loses control over it. In this module, the initial data undergoes a process of division into three tiers, and encryption can be applied at any level utilizing various cryptographic methods.

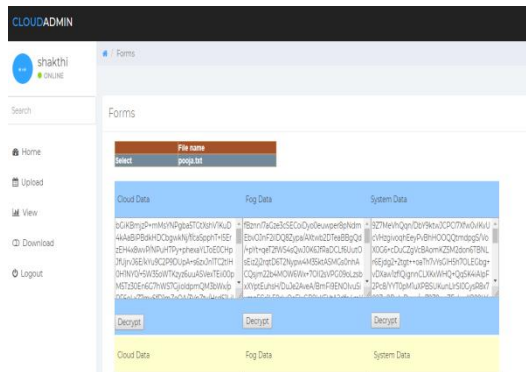
Within this module, users have the ability to access their files from three distinct storage locations, comprising a cloud server and a local computer. A technique for data division involves using Solomon's subtraction notation to partition the information into various elements. However, in the event that a portion of the data is missing, it results in a loss of information about that specific data.

V RESULTS

1. FILE UPLOAD



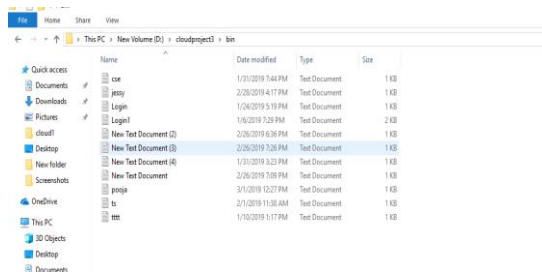
2.STORAGE SCHEME



3.RECOVERY MODULE



4. BUCKET MODULE



This document delves into recent research concerning the protection of data within the increasingly prevalent domain of multi-cloud computing. It

underscores the crucial role of encryption in securing sensitive information, examining both established solutions such as AES and emerging possibilities like homomorphic encryption. While acknowledging the efficiency of minimal encryption times associated with algorithms like SHA-1, the paper highlights the paramount importance of selecting algorithms tailored to the specific data being safeguarded and the evolving threat landscape. In addition to encryption, the document explores recommended practices for key management and recognizes potential performance and recovery considerations linked to robust data security measures. By delving into these aspects and citing specific references from the reviewed research, this paper aims to contribute significantly to the ongoing discourse on enhancing data security in multi-cloud environments.

VI CONCLUSION

Numerous organizations opt for secure storage solutions to enhance the security of their stored data within the cloud. This strategic choice addresses the vulnerability associated with storing entire documents in a single cloud server, which poses a considerable risk. With the adoption of this approach, security is significantly improved. The method involves dispersing securely encrypted files across different servers within the cloud, utilizing various storage techniques. Rather than storing an entire document in a single cloud system, the document is divided into multiple segments, hashed, encrypted, and distributed across multiple cloud servers. This approach not only enhances security through advanced encryption measures but also ensures data integrity, offering organizations a valuable advantage in safeguarding their files.

REFERENCES

- [1] Basu, Srijita, et al. "Cloud computing security challenges & solutions-A survey." *8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018.
- [2] Butt, Umer Ahmed, et al. "A review of machine learning algorithms for cloud computing security." *Electronics* 9.9 (2020): 1379.
- [3] Odun-Ayo, Isaac, Olasupo Ajayi, and Sanjay Misra. "Cloud computing security: Issues and developments." *Lecture notes in engineering and computer science* 2235 (2018).
- [4] Subramanian, Nalini, and Andrews Jeyaraj. "Recent security challenges in cloud computing." *Computers & Electrical Engineering* 71 (2018): 28-42.
- [5] Alouffi, Bader, et al. "A systematic literature review on cloud computing security: threats and mitigation strategies." *IEEE Access* 9 (2021): 57792-57807.
- [6] Kumar, P. Ravi, P. Herbert Raj, and P. Jelciana. "Exploring data security issues and solutions in cloud computing." *Procedia Computer Science* 125 (2018): 691-697.
- [7] Tabrizchi, Hamed, and Marjan Kuchaki Rafsanjani. "A survey on security challenges in cloud computing: issues, threats, and solutions." *The journal of supercomputing* 76.12 (2020): 9493-9532.
- [8] Tahirkheli, Abeer Iftikhar, et al. "A survey on modern cloud computing security over smart city networks: Threats, vulnerabilities, consequences, countermeasures, and challenges." *Electronics* 10.15 (2021)
- [9] Sun, Pan Jun. "Privacy protection and data security in cloud computing: a survey, challenges, and solutions." *IEEE Access* 7 (2019)
- [10] Tutubala, Ntebaleng, and Topside E. Mathonsi. "A Hybrid Framework to Improve Data Security in Cloud Computing." *Big Data, Knowledge and Control Systems Engineering (BdKCSE)*. IEEE, 2021.