A MAJOR PROJECT

on

SECURE CLOUD SIMULATOR

Submitted in partial fulfilment of the requirements of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MALLEM ANIL KUMAR

O180677

Under the guidance

Mr. N. MALLIKARJUNA

Assistant Professor

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

ONGOLE CAMPUS

(2023-2024)

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
ONGOLE CAMPUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is certify that the Major Project entitled **'SECURE CLOUD SIMULATOR'** being submitted by **MALLEM ANIL KUMAR** bearing the ID Number **O180677** in partial fulfilment of the requirements for the award of the degree of the Bachelor Of Technology in Computer Science and Engineering in Rajiv Gandhi University of Knowledge Technologies, Ongole Campus is a record of Bonafide project work carried out by them under my guidance and supervision during the academic year 2023-2024.

The results presented in this project have been verified and found to be satisfactory. The results embodied in this project report have not been submitted to any other University for the award of any other degree or diploma.

Mr. N. Mallikarjuna,
Assistant Professor,
Department of CSE,
RGUKT Ongole.

Mr. B. Sampath Babu,
Head of Department,
Department of CSE,
RGUKT Ongole.

APPROVAL SHEET

This report entitled **SECURE CLOUD SIMULATOR** by **MALLEM ANIL KUMAR (O180677)** is approved for the degree of Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING

Examiner:

_____

_____

_____

Supervisor:

_____

_____

_____

Chairman:

_____

Date:

Place:

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature :

Name: MALLEM ANIL KUMAR

Roll No: O180677

Date:

ACKNOWLEDGEMENT

It is my privilege and pleasure to express a profound sense of respect, gratitude and indebtedness to my guide **Mr. N. Mallikarjuna,** Assistant Professor, Department of CSE, RGUKT - Ongole, for her indefatigable inspiration, guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work.

I express our sincere gratitude to **Mr. B. Sampath Babu**, Head of the Department of CSE, for his suggestions, motivations and co-operation for the successful completion of the work.

I extend our sincere thanks to **Mr. M. Rupas Kumar**, Dean of Academics, RGUKT - Ongole, for his encouragement and constant help. I extend our sincere thanks to **Prof**. **B. Jayarami Reddy**, Director Rajiv Gandhi University of Knowledge Technologies - Ongole for his encouragement.

With Sincere Regards,
MALLEM ANIL KUMAR
O180677

Date:

ABSTRACT

The "Secure Cloud Simulator using Cryptography and Python" project represents a comprehensive endeavor in the realm of data security and cloud interaction. The project centers around the implementation of advanced encryption algorithms, specifically Fernet and AES, to provide users with a robust toolset for encrypting and decrypting files. This encryption capability extends to any file type, ensuring versatile protection of sensitive data. Moreover, the project seamlessly integrates with Amazon Web Services (AWS) S3 storage via the Python Boto3 library, enabling users to securely upload, download, and manage encrypted files in the cloud environment.

The core of this project lies in a web application that serves as an intuitive platform for users to interact with the encryption and cloud storage functionalities. Through this application, users can easily encrypt files, upload them to AWS S3 for safekeeping, and subsequently retrieve and decrypt them as needed. The underlying technologies driving this application include Python, Flask, a selection of cryptography algorithms, and the AWS cloud infrastructure.

By combining the power of encryption with the convenience of cloud storage, this project addresses a critical need for individuals and businesses seeking to fortify their data security in an increasingly digital landscape. The implementation of Fernet and AES algorithms, along with the integration of AWS services, results in a secure and versatile solution that promises to have a positive impact on the field of data protection and cloud computing.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1:

INTRODUCTION

1.1 PURPOSE

Cloud Computing is the style of computing where the resources are provided as services on the internet. There are three types of services in Cloud Computing which are used for the deployment of the application on the cloud. Data on the cloud will become more scalable, Reliable and Secure. The big players in Cloud Computing are Amazon, Google, Microsoft and IBM. Cloud Computing is based on five attributes such as Shared Resources, Scalability, Pay as U use, Elasticity and Self Provisioning of Resource. Most of the enterprises shift their applications on to the cloud owing to its speed of implementation and deployment, improved customer experience, scalability, and cost control. The services in Cloud Computing are SaaS, PaaS, IaaS amongst which we are using PaaS and IaaS service for deployment of Application on the Cloud in our Project. This service exhibits five essential characteristics such As Rapid Elasticity, Resource Pooling, on demand Self-service, Broad Network Areas. Data is being transmitted between two clouds so to secure the data most of the systems use the combination of techniques, including:

- Encryption- It is used to encode the data in such a way that third parties will not be able to hack that data.
- Authentication- It is used to create a separate user ID and Password so that only the authorized users will be able to access the data.
- Separation of duties- In which accessibility is provided to all the users according to their priority.

These security parameters are achieved due to which the performance will get increased and therefore the Security is obtained up to a higher extent. Data security and privacy risks have become the primary concern for people to shift to cloud computing. Cloud Computing is mainly used for improving the data handling capability where the services and the resources will be delivered continuously when and where required due to which Cloud computing is in great demand. However there still exist many problems in cloud computing today, a recent survey shows that data security and privacy risks have become the primary concern for people to shift to Cloud computing

Cloud is the free space where the application is being saved securely and the services are being provided continuously when and where required.

Cloud Deployment Models:

A. *Public Cloud*: The Cloud infrastructure is made available for the large industry group and public provided by single service provider.
B. *Private Cloud*: The Organization can store the data on private Cloud. The main Advantage of this Cloud is Security of Data and Quality of Service.
C. *Community Cloud*: The Cloud Infrastructure is shared by many Organizations.
D. *Hybrid Cloud*: Two or more Clouds combine to form Hybrid Cloud.

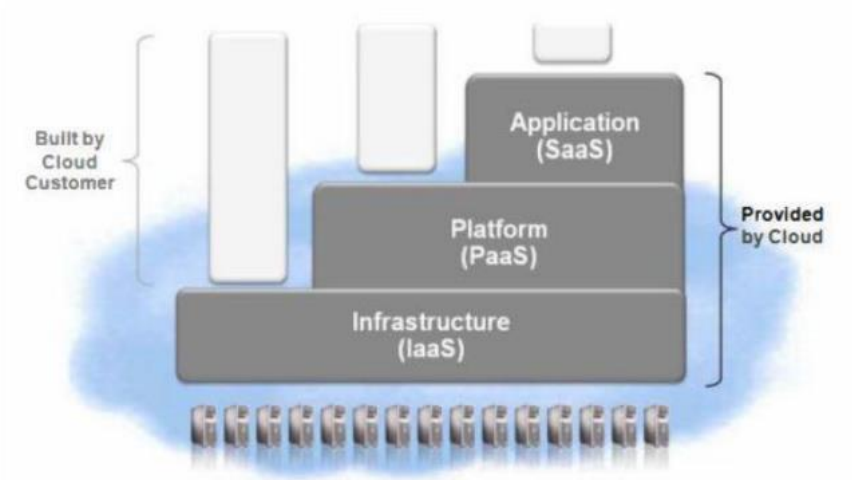

Fig.1.1 Cloud Deployment Models

Cloud Characteristics:

A. *Easy Use* - Most Cloud Provider will offer Internet interfaces which are much simple so user can easily access the cloud services.
B. *Ubiquitous Network Access* - Cloud provides services through the standard terminal such as phones, Laptops, Mobiles.

C. *On demand Services* - Cloud is a pool of resources and services so we can get the services and resources by paying the amount as required.
D. *Business Model* - Cloud is a Business Model because it is pay per use of service or resource.
E. *Pay as U Used* - Users must pay for only the Resources they are using. Whenever the users need some resources then they must pay for the resource as and when required

1.2 OBJECTIVE

The primary goal of this project is to provide and simulate an effective solution to face the challenges and solve security issues that exist in cloud computing. Cloud Computing is the impending need of computing which is used for the IT Industries. It is one of the hottest topics in research areas. Scalability and Flexibility increases for the computing services. Cloud Computing is the fastest growing technology for the IT Industry. Information is being transmitted via the network therefore security is one of the main problems or issues. The Application is deployed on the Cloud and for the secure transmission of the data we will be using ECC Algorithm in our project because of its advantages in terms of CPU utilization, time for Encryption and Key Size. This Project will explore the deployment of Application on the Cloud and increases the security level by implementing ECC & ECDH Algorithm, and AES Algorithm for secure file handling and Encryption.

1.3 MOTIVATION

The need for data security is an essential issue in the domain of computing traditionally. There are various algorithms developed to improve the security of data, but they are having their own issues. Now in these days the traditional algorithms are not very suitable for providing security over untrusted communications and data exchange.

ECC is more and more considered as an attractive public-key cryptosystem for mobile/wireless environments. One of the other recent public key cryptosystems is Elliptic Curves Cryptography used for security. In recent times, most e-commerce applications are designed using asymmetric cryptography to assure the authentication of the concerned parties. Compared to traditional Public-key cryptosystems like RSA or Diffie-Hellman, ECC propose equivalent security

with smaller key sizes; these results in faster calculation, lower power expenditure, as well as memory and bandwidth savings. ECC is peculiarly useful for mobile devices, which are typically particular in terms of their CPU, power and network connectivity.

Therefore, a new encryption standard is required that can fulfil the current need of security meanwhile that is extendable according to the need. The proposed work includes the development of new hybrid algorithms using ECC, ECDH and AES algorithms along with encryption techniques.

1.4 DEFINITION AND OVERVIEW

Cloud Computing is the primitive change happening in the field of Information Technology. It uses internet technologies for delivery of IT - enabled capabilities 'as a service' to any needed users. Cloud computing enables users to access resources using the internet, from anywhere at any time without worrying about technical/physical management and maintenance concern of the original resources. In its description for cloud characteristics The US National Institute of Standards and Technology (NIST) defines as cloud characteristics the following: On demand self-service, Ubiquitous network access, Resource pooling, Rapid elasticity (resources can be scaled up and down easily), Metered service (resources' usage is measured) and Pay As-You-Consume business models. Google Apps is an important example of Cloud computing; it enables access services through the browser and brought into effective action millions of machines over the Internet. One of the most prominent services offered by cloud computing is cloud storage. Cloud storage is simply a term that refers to on line space that you can use to store your data. In more strict way, cloud storage is a service model in which data is maintained, managed and backed up remotely and made available to users over a network.

Software as a service (SaaS)

The cloud provider deploys, configures, maintains and updates the operation of the software applications on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumers. The cloud consumers have limited administrative control of the applications.

Platform as a service (PaaS)

The Cloud Provider manages the computing infrastructure for the platform and runs the cloud software that provides the components of the platform, such as runtime software execution stack, databases, and other middleware components. The PaaS Cloud Provider typically also supports the development, deployment and management process of the PaaS Cloud Consumer by providing tools such as integrated development environments (IDEs), development version of cloud software, software development kits (SDKs), deployment and management tools.

Infrastructure as a service (IaaS)

The Cloud Provider acquires the physical computing resources underlying the service, including the servers, networks, storage and hosting infrastructure. The Cloud Provider runs the cloud software necessary to make computing resources available to the IaaS Cloud Consumer through a set of service interfaces and computing resource abstractions, such as virtual machines and virtual network interfaces.

The whole idea and definition of this project lies in its name i.e. **Secure Cloud**, which aims at providing and simulating an effective solution to face the challenges and solve security issues that exist in cloud computing. But first we should look at some of the frequently occurring issues in cloud computing, mostly during the transmission of data. Some of them are discussed below:

A. Encryption- The message sent by the sender i.e. the original message is being encrypted in such a way that a third party will not be able to hack or misuse the data.
B. Intrusion Detection and Prevention- Data that is being entered and going out of the Network must know.
C. Separation of Duties- Due to insufficient communication between the expertise System misconfiguration takes place.
D. Location of Data- Every Organization will have different requirements and their access control on their data to be placed. A level of security is required to fulfil the customer's need.

Sharing of Cloud Infrastructure could lead to privacy issues. The Location of data could influence the privacy obligations. For storage and processing of data. Data leakage could also occur due to failure of security access rights. To secure the data stored on the cloud various security Algorithms

are present which will help to encrypt the data before transmission to protect the valuable data from the hackers.

One of the better solutions for maintaining security is cryptography which is basically used for protecting data.

Public Key Cryptography - In this cryptography different keys are used for Encryption and Decryption.

Secret Key Cryptography - A key which is used for Encryption as well as Decryption is called Secret Key Cryptography.

There are many Security Algorithms Each Algorithm have their own properties such as Key Size, Throughput, Performance, Encryption Decryption Time etc. By Comparing the Encryption Algorithms, we found out that ECC Algorithm is one of the best Algorithm which is having the high level of Security and better performance.

Advanced Encryption Standard (AES):

The Advanced Encryption Standard (AES) is a symmetric key encryption algorithm that was established as a standard by the U.S. National Institute of Standards and Technology (NIST) in 2001. It is widely utilized globally for safeguarding sensitive data and communications.

Symmetric Key Encryption

AES operates on the principle of symmetric key encryption, meaning the same key is used for both encryption and decryption processes. This contrasts with asymmetric key algorithms, where different keys are used for encryption and decryption.

Substitution-Permutation Network (SPN) Cipher

AES is categorized as a Substitution-Permutation Network (SPN) cipher. This implies that it processes data in blocks and employs a series of substitutions and permutations to achieve encryption or decryption. This block cipher approach makes it suitable for a diverse range of applications.

Key Length Variants

One of the distinguishing features of AES is its flexibility in supporting key lengths. It can accommodate key sizes of 128, 192, or 256 bits, allowing for varying levels of security. Longer keys offer a higher degree of encryption strength.

Block Size

AES typically processes data in blocks, with a standard block size of 128 bits (16 bytes). This makes it well-suited for various applications, including secure communications and file encryption.

Number of Rounds

The AES algorithm consists of multiple rounds of processing, and the number of rounds depends on the chosen key length. For 128-bit keys, it involves 10 rounds; for 192-bit keys, 12 rounds; and for 256-bit keys, 14 rounds. Each round employs a series of defined operations, enhancing its resistance to cryptographic attacks.

Diffusion and Confusion

Within each round, AES employs a combination of substitution, permutation, and mixing operations. These steps contribute to a high level of diffusion and confusion, which are essential properties for robust encryption algorithms.

CHAPTER 2:

ANALYSIS

2.1 Existed System

Before the implementation of the "Secure Cloud Simulator using Cryptography and Python" project, users faced challenges in securely managing their files in cloud storage environments. Traditional cloud storage services often lacked robust encryption options, leaving sensitive data vulnerable to unauthorized access or breaches. Users were compelled to rely solely on the security measures provided by the cloud service providers, which may not always meet their specific privacy and security requirements.

Additionally, users had limited control over the encryption process, making it challenging to ensure end-to-end security for their stored files. The absence of user-friendly interfaces for encryption and decryption meant that individuals without advanced technical knowledge may have struggled to protect their data effectively.

Moreover, the integration of cloud storage with advanced encryption algorithms was not readily available in existing systems. This gap in functionality left users without a seamless and secure solution for encrypting and managing their files in the cloud.

In summary, the existing system lacked a comprehensive and user-centric approach to secure cloud storage. The absence of robust encryption options and user-friendly interfaces hindered users' ability to safeguard their sensitive data effectively. The "Secure Cloud Simulator using Cryptography and Python" project was conceived to address these shortcomings and provide users with a powerful toolset for secure cloud storage and interaction.

2.2 Proposed System

The "Secure Cloud Simulator using Cryptography and Python" project introduces an advanced solution for secure cloud storage and data protection. This system addresses the limitations of existing methods by incorporating cutting-edge encryption techniques and seamless integration with popular cloud service providers.

Key Features:

1. Advanced Encryption Algorithms:
    a. The system leverages state-of-the-art encryption algorithms like Fernet and AES to provide robust data security. This ensures that user files remain confidential and protected from unauthorized access.
2. User-Friendly Interface:
    a. The proposed system offers an intuitive web application interface that allows users to easily encrypt, upload, download, and decrypt their files. This user-centric design simplifies the process, making it accessible to individuals with varying levels of technical expertise.
3. Seamless Cloud Integration:
    a. The system seamlessly integrates with popular cloud storage services, such as Amazon Web Services (AWS) S3, using the Python Boto3 library. This allows users to directly upload and manage their encrypted files in their preferred cloud environment.
4. Comprehensive File Management:
    a. Users can organize and manage their files within the application. They can select specific folders or directories for upload, ensuring structured storage within the cloud environment.
5. User Authentication and Access Control:
    a. The system includes a robust user authentication and access control system. Users are required to register and create a unique account, providing essential information during the registration process. Registered users can securely log in to their accounts using their credentials

2.3 Future Scope

The "Secure Cloud Simulator using Cryptography and Python" project has immense potential for future expansion and enhancements. Here are some key areas for future development and improvement:

1. Multi-Platform Compatibility:
    a. Extend the application's compatibility to different operating systems and platforms, including mobile devices and tablets, to broaden its user base and increase accessibility.
2. Enhanced Encryption Algorithms:
    a. Research and implement additional state-of-the-art encryption algorithms to offer users a wider range of options for securing their data.
3. Multi-Cloud Integration:
    a. Expand the application's capabilities to integrate with various cloud service providers beyond AWS, allowing users to choose their preferred platform for secure storage.
4. File Versioning and Tracking:
    a. Incorporate features for versioning and tracking changes to files, allowing users to maintain a history of their data and recover previous versions if needed.
5. File Sharing and Collaboration:
    a. Implement functionalities for securely sharing and collaborating on encrypted files within a trusted network, enhancing collaboration capabilities.
6. Advanced Key Management:
    a. Implement a more sophisticated key management system with features like key rotation, key revocation, and secure storage to enhance security measures.

2.4 Project Requirements

2.4.1. Software Requirements

- Python
- Flask
- Cryptography
- Encryption Algorithms
- AWS Cloud

2.4.2 Hardware Requirements

- CPU: Intel Core i3 or higher
- RAM: 4GB or higher
- Storage: 10GB or higher
- Operating system: Windows 10, macOS 10.15, or Linux

CHAPTER 3:

LITERATURE REVIEW

This section describes the work that has been conducted in the field of Cloud simulators. To the best of our knowledge and literature review, we have not found any single Cloud simulator focused on studying security issues till date.

Cloud simulators provide many advantages such as scalability and repeatability, but still, it cannot lead to the performance of an actual Cloud. Therefore, several commercial testbeds were introduced to analyze different features such as:

1. Amazon Elastic Compute Cloud (EC2) [10] evaluates based on cost trade-off, performance and reliability.
2. Amazon Simple Storage Service (S3) [11] provides storage services.
3. Google Apps [12] provides Software as a Service (SaaS) for email, calendar, and word processing.
4. App Engine [13] provides Platform as a Service (PaaS).
5. Window Azure [14] analyzed for different research purposes such as performance, reliability, response time, and efficiency related parameters. However, using the commercial testbeds leads to huge budget costs for conducting large-scale experiments. On the other hand, some research testbeds were also developed to solve budgets related issues incurred on commercial testbeds as: OpenCirrus [15], Open Cloud Test bed [16], Science Cloud [17], Virtual Computing Lab [18], and Grid'5000 [19]. The frameworks of Cloud Computing were also used for creating and deployment of private Cloud as: Eucalyptus [20] for IaaS, iVIC [21] for IaaS and PaaS, OpenNebula [22] acts as virtual machine manager, OpenStack [23] is Linux based with large-scale configuration facility with less memory, and Nimbus [6] for open-source IaaS. However, none of the commercial testbeds, research testbeds, and Cloud-Computing framework has Cloud security features in their contributions

CHAPTER 4:

DESIGN

4.1 UML Diagrams

In software engineering, a class diagram in the Unified Modeling Language is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Use Case Diagram:

Use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior. Use cases once specified can be denoted both textual and visual representation. A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

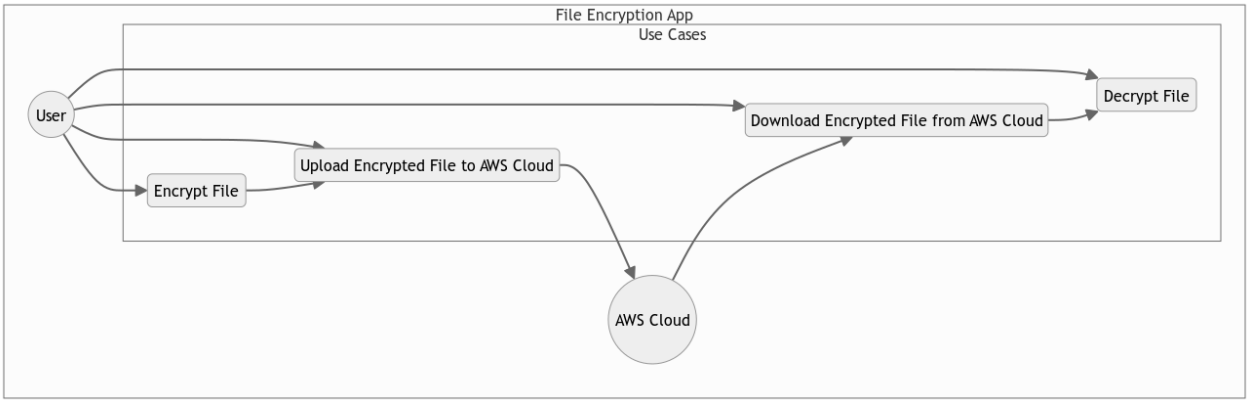

Fig.4.1 Use Case Diagram

Sequence Diagram

A sequence diagram is a type of interaction diagram that shows how processes operate with one another and the order in which they occur. It illustrates the dynamic behavior of the system, emphasizing the sequence of messages exchanged between the different components or objects within the system.

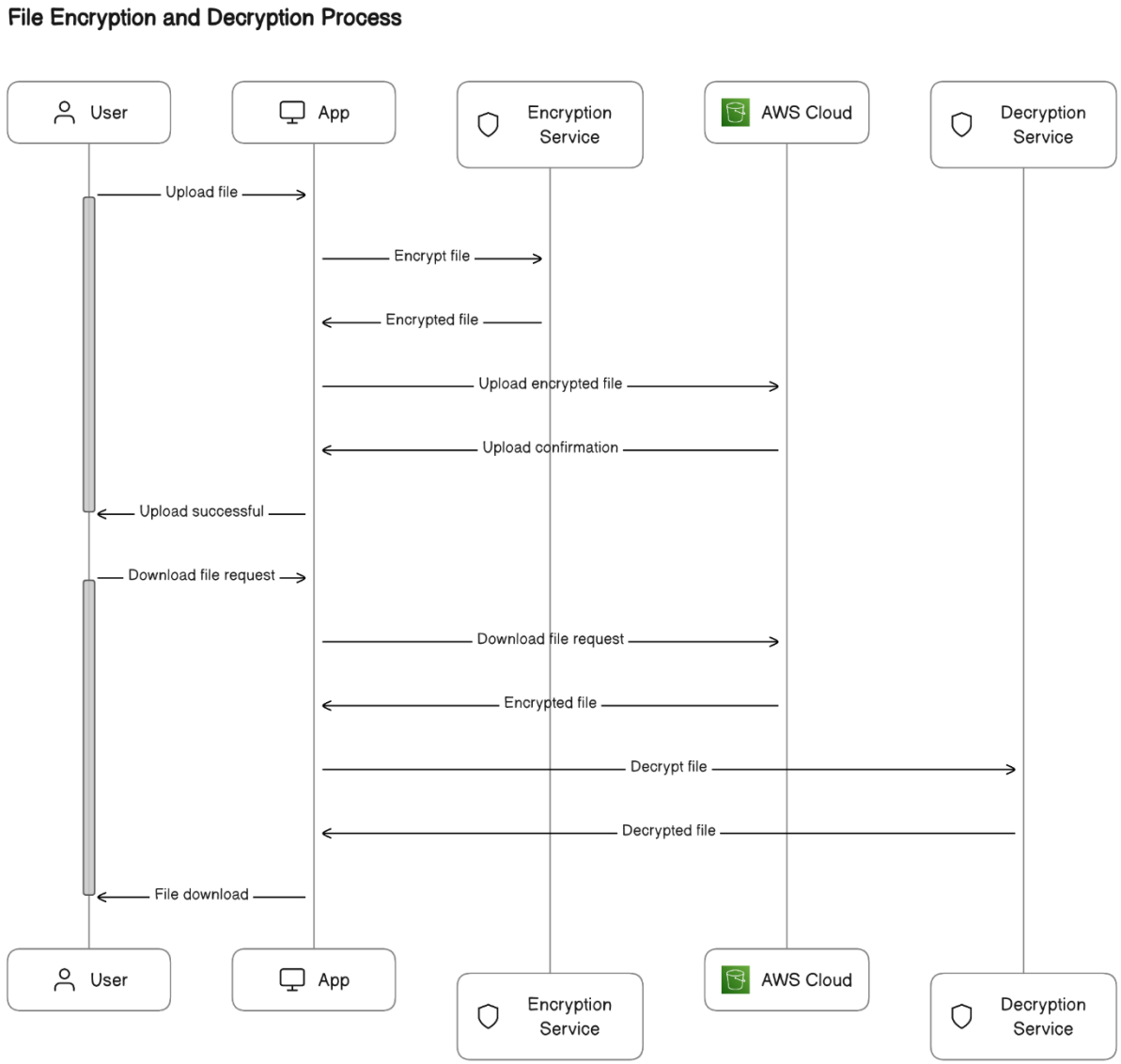

Fig.4.2 Sequence Diagram

Activity Diagram

Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

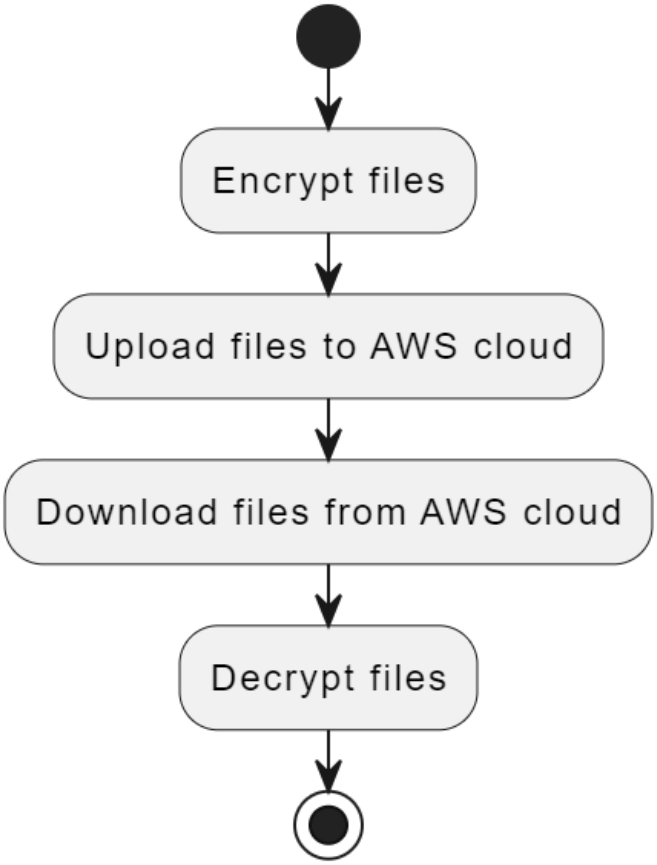

Fig.4.3 Activity Diagram

CHAPTER 5:

IMPLEMENTATION

5.1 Modules

- User Authentication and Authorization Module
- File Encryption and Decryption Module
- Cloud Integration Module
- User Interface Module
- Key Management System Module
- File Management Module

5.2 Module Description

1. **User Authentication and Authorization:**
    a. This module is responsible for managing user accounts, authentication, and authorization. It handles user registration, where users provide necessary information to create an account. The login functionality securely verifies user credentials, granting access only to authorized users. User profiles, including personal details and password management, are part of this module, ensuring a secure and personalized user experience.
2. **File Encryption and Decryption:**
    a. The File Encryption and Decryption module are crucial components for ensuring data security. It implements Fernet and AES encryption algorithms to protect user files. Users can encrypt their files before uploading to the cloud, enhancing the confidentiality and integrity of sensitive information. Similarly, the module allows for the decryption of files, ensuring users can access their data securely when needed.
3. **Cloud Integration Module:**
    a. The Cloud Integration module facilitates seamless interaction with cloud storage services, particularly AWS S3, using the Python Boto3 library. It enables users to upload their encrypted files to the cloud and download them when necessary. This

integration ensures that the encrypted files are securely stored in the cloud environment, enhancing accessibility and availability.

4. **User Interface Module:**
    a. The User Interface module focuses on creating an intuitive and user-friendly web application interface. This is the front-end component that users interact with to encrypt and manage their files. The design ensures easy navigation, clear presentation of functionalities, and overall positive user experience. It bridges the gap between the user and the various back-end modules.
5. **Key Management System:**
    a. The Key Management System module plays a vital role in generating, storing, and managing encryption keys securely. It supports user-controlled key input or automated key generation, ensuring that the encryption process remains robust, and the keys are well-protected. Proper key management is essential for effective data protection and security.
6. **File Management:**
    a. The File Management module allows users to organize and manage their files within the application. Users can select specific folders or directories for upload, ensuring a structured approach to file storage. This module enhances user control over their data, facilitating efficient organization and retrieval of encrypted files.

5.3 Introduction to Technologies Used:

Programming Language:

Python: The core language for implementing the project's backend logic, including encryption algorithms, file handling, and cloud integration. Python is known for its readability and extensive libraries, making it suitable for cryptography and cloud-related tasks.

Web Framework:

Flask: A lightweight and flexible web framework for Python. Flask is commonly used for developing web applications and provides the necessary tools for creating the user interface, managing routes, and handling requests.

Cryptography Libraries:

Cryptography: A Python library that provides cryptographic recipes and primitives, including Fernet and AES algorithms for encryption and decryption. This library ensures secure handling of sensitive data.

Cloud Integration:

Boto3: The official Python SDK for AWS (Amazon Web Services). Boto3 facilitates the integration of your application with AWS S3 for secure cloud storage. It allows you to interact with AWS services programmatically.

Frontend Technologies:

HTML, CSS, JavaScript: These are fundamental technologies for building the frontend of your web application. HTML structures the content, CSS styles it, and JavaScript adds interactivity to the user interface.

Database:

MYSQL: Flask supports various databases, and MYSQL Lite is often used for smaller projects or during development. MYSQL is an ORM (Object-Relational Mapping) library that simplifies database interactions.

5.4 Sample Code

```
@app.route('/encryptfernet', methods=['POST'])
def encrypt_file_fernet():
 # check if the post request has the file part
    if 'file' not in request.files:
        return "No file uploaded"
    file = request.files['file']
    if file.filename == '':
        return "No file selected"
    # Check if the file type is allowed
    if file and allowed_file(file.filename):
        # Secure the filename
```

```
filename = secure_filename(file.filename)
# Save the file to the upload folder
filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
file.save(filepath)
# Generate a new encryption key and save it to a file named 'key.key'
key = Fernet.generate_key()
keyfilename = filename + '.key'
with open(keyfilename, 'wb') as key_file:
    key_file.write(key)
# Create a Fernet instance with the generated key
fernet = Fernet(key)
# Read the contents of the original file
with open(filepath, 'rb') as original_file:
    original = original_file.read()
# Encrypt the file contents
encrypted = fernet.encrypt(original)
# Save the encrypted file to a file named after the original file
with open(filename, 'wb') as encrypted_file:
    encrypted_file.write(encrypted)
# Create a zip file containing the encrypted file and the encryption key
filen = filename.split(".")[0]
zip_path = os.path.join(app.config['UPLOAD_FOLDER'], filen + '.zip')
with ZipFile(zip_path, 'w') as zip_file:
    zip_file.write(filename)
    zip_file.write(keyfilename)
zip_file.close()
# Remove the original file, the encrypted file, and the encryption key
os.remove(filename)
os.remove(filepath)
os.remove(keyfilename)
```

```
        # Send the decrypted file to the user
        response = send_file(zip_path, as_attachment=True)
        return response
    else:
        # If the file type is not allowed, redirect to the home page
        return render_template('index.html')
@app.route('/decryptfernet', methods=['GET', 'POST'])
```

CHAPTER 6:

SYSTEM FEATURES

Fernet Encryption and Decryption Page:

The **File Encryption and Decryption Page using Fernet** Algorithm plays a pivotal role in bolstering data security within the application. This page empowers users to encrypt their files prior to uploading them to the cloud, safeguarding sensitive information. The page boasts a user-friendly interface with crucial functionalities, such as file selection, the flexibility to input or generate a Fernet encryption key, and the capability to customize encryption preferences. Upon initiation, the system applies the Fernet algorithm to selected files, producing encrypted versions prepared for secure storage and transmission. Should unforeseen circumstances arise, the section offers clear error messages and guidance for users. Additionally, users can seamlessly interact with the Key Management System Module, ensuring the secure handling of encryption keys. In totality, the File Encryption and Decryption Page harmoniously integrates with other modules, creating a unified and comprehensive encryption workflow within the application.

AES Encryption and Decryption Page:

The AES Encryption and Decryption Page, a critical component in bolstering data security, empowers users to encrypt and decrypt their files using the Advanced Encryption Standard (AES) algorithm. This page provides a user-friendly interface with key features including file selection, the option to input or generate an AES encryption key, and the ability to customize encryption preferences. Upon initiation, the system applies the AES algorithm to selected files, generating encrypted versions ready for secure storage and transmission. Similarly, users can choose to decrypt previously encrypted files, restoring them to their original state. In the event of unexpected occurrences, the section offers clear error messages and guidance for users. Additionally, users have the option to interface with the Key Management System Module, ensuring the secure handling of encryption keys. The AES Encryption and Decryption Page seamlessly integrates with other modules, guaranteeing a unified and comprehensive encryption workflow within the application.

Text Encryption and Decryption Page:

The Text Encryption and Decryption Page provides a user-friendly interface for encrypting and decrypting text within the application. Users can enter their text in the designated field and choose an encryption method (such as Fernet, AES, etc.) along with a corresponding key. The system then applies the selected encryption algorithm to the input text, producing an encrypted version. Additionally, users can decrypt previously encrypted text by providing the appropriate decryption key. In case of any unexpected events, the section offers clear error messages and guidance for users. It also allows users to interface with the Key Management System Module for secure handling of encryption keys. Overall, the Text Encryption and Decryption Page harmoniously integrates with other modules, ensuring a seamless and comprehensive encryption workflow within the application.

Cloud Upload and Download Page:

The Cloud Upload and Download Page serves as a pivotal gateway for seamlessly interacting with cloud storage. Users can effortlessly upload encrypted files to the cloud for secure storage and access them from anywhere. The page offers a user-friendly interface with intuitive features including file selection, cloud destination selection (AWS S3, for instance), and progress tracking during upload and download operations. Users have the option to select specific folders or directories for organization. In case of any unexpected events, the section provides clear error messages and guidance for users. Additionally, users can interface with the AWS Integration Module for secure handling of cloud storage operations. The Cloud Upload and Download Page seamlessly integrates with other modules, ensuring a cohesive and comprehensive cloud interaction workflow within the application.

User Register and Login Page:

The User Registration Page acts as the initial point for new users or those without a registered account, collecting essential details such as full name, email, mobile number, date of birth, and gender. It employs client-side validation for data accuracy and server-side validation to ensure information integrity and security. Upon successful registration, user details are securely stored, creating a functional profile and smoothly transitioning users to the main menu or application screen. The Login Page, designed as a secure access point for registered users, requires the input of credentials like email and password. Client-side validation ensures input accuracy,

followed by server-side authentication. A successful login redirects users to the main menu or home page, granting access to the application's features. Clear error messages guide users in case of login failures. Both pages collectively establish a robust user authentication system, enhancing the overall security and usability of the application.

CHAPTER 7:

NON-FUNCTIONAL FEATURES

A careful specification and adherence of non-functional requirements such as performance, security, privacy and availability are crucial to the success or failure of any software system.

5.1 Performance Requirements

- The capability of the application depends on the performance of the servers. Anyone can use the application easily because of good GUI.
- On mobile devices and laptops, the battery is a scarce and valuable resource. The battery should remain maximally available for the application to perform well. Your application may therefore fall by the wayside or even get uninstalled by the user, if it drains too much battery.
- The text font size may need to be adjusted up (for high resolution screens) or down (for low resolution screens) so as to keep the text readable.

5.2 Safety Requirements

- The layout may need to be taken care of and adjusted to increase or decrease the spacing between and around labels and widgets shown on the screen so as to prevent them from getting clustered together on high-res screens or spaced apart too much on low-res screens.
- Any images used in the project must be provided in two different versions: a large size/high resolution version and a small size/low resolution version so that it properly fills the amount of physical space available on the screen.

5.3 Security Requirements

- Although security is the utmost priority and has been taken care of the most, care must be taken against virus and malware threats.
- This application will be available for all the users of the Internet. The system server should be up for 365 days (about 12 months), and the downtime should be minimized in case of any attack or difficulties.
- Firewall should be used on the user's system to prevent any suspicious activity.

5.4 Software Quality Attributes

- 24x7 availability of the system with suitable updating at regular intervals of time. To maintain integrity of the data and to ensure the security of the database by asking them to sign up for the application.
- Form validation so that only real users access the system. An error message should be displayed in case of improper working of the application.
- Email -ID entered should be valid as OTP is sent to that Email ID.
- The application can be accessed at any place that has Internet connectivity.
- Always save the data before closing the website.
- An error message should be displayed in case of improper working of the application.
- 24-hour availability of internet connection is required.

CHAPTER 8:

TEST CASES

User Authentication and Authorization Module:

1. Test Case: User Registration

    1. Steps:
        a. Navigate to the registration page.
        b. Enter valid user information (name, email, password, etc.).
        c. Click the "Register" button.
    2. Expected Result: User should be successfully registered and redirected to the login page.

2. Test Case: User Login

    1. Steps:
        a. Navigate to the login page.
        b. Enter valid login credentials.
        c. Click the "Login" button.
    2. Expected Result: User should be successfully registered and redirected to the login page.

2. Test Case: Invalid Login Attempt

    1. Steps:
        a. Navigate to the login page.
        b. Enter invalid login credentials.
        c. Click the "Login" button.
    2. Expected Result: User should receive an error message, and login should not be successful.

File Encryption and Decryption Module:

1. Test Case: File Encryption

    1. Steps:
        a. Select a file for encryption.
        b. Choose encryption algorithm and settings.
        c. Click the "Encrypt" button.
    2. Expected Result: File should be successfully encrypted without errors.

2. Test Case: File Decryption

    1. Steps:
        a. Select an encrypted file.
        b. Choose decryption algorithm and settings.
        c. Click the "Decrypt" button.
    2. Expected Result: Encrypted file should be successfully decrypted without errors.

Cloud Integration Module:

1. Test Case: File Upload to Cloud

    1. Steps:
        a. Encrypt a file.
        b. Click the "Upload to Cloud" button.
    2. Expected Result: Encrypted file should be successfully uploaded to the cloud storage.

2. Test Case: File Download from Cloud

    1. Steps:
        a. Select an encrypted file from the cloud.
        b. Click the "Download" button.
    2. Expected Result: Encrypted file should be successfully downloaded and ready for decryption.

CHAPTER 9:
RESULTS

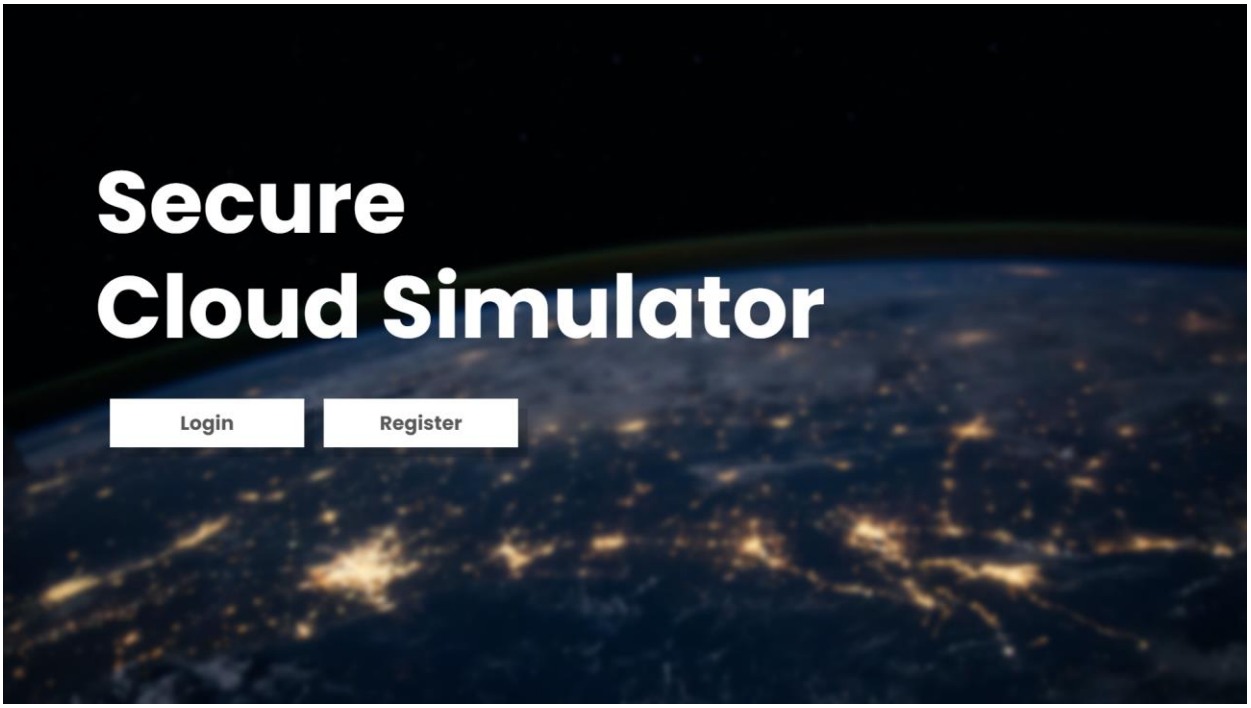

Fig.9.1 Home Page

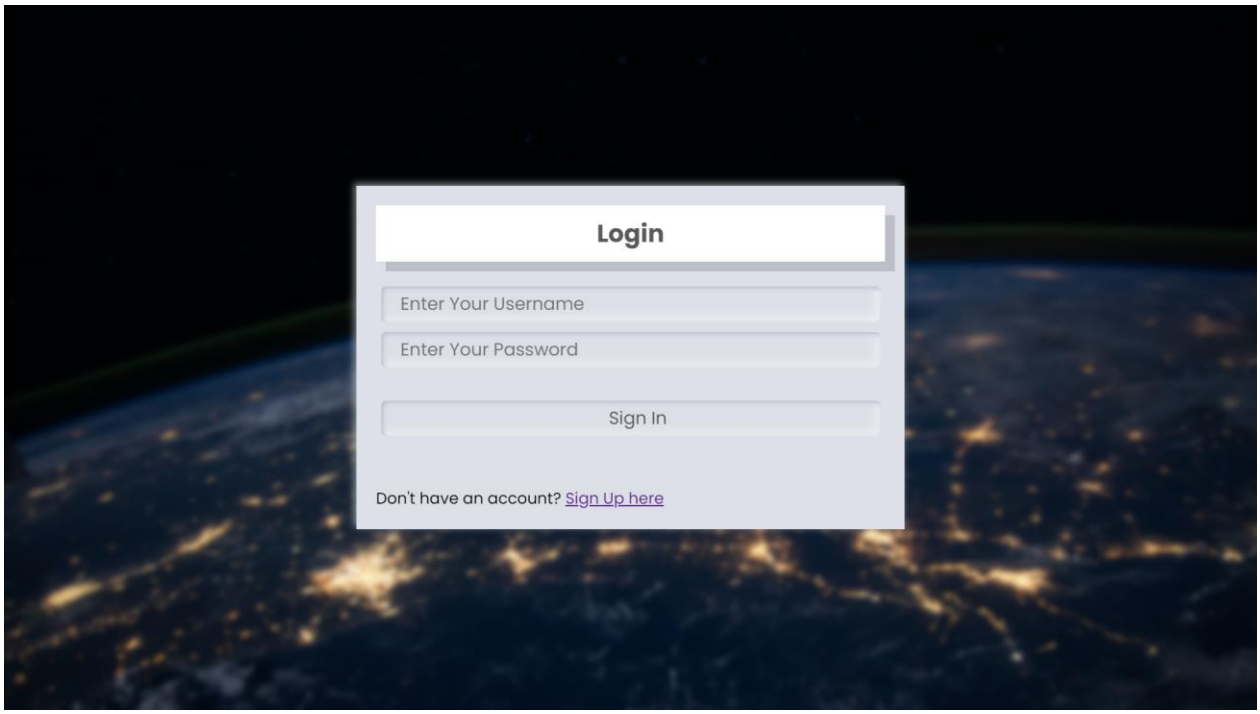

Fig.9.2 Login Page

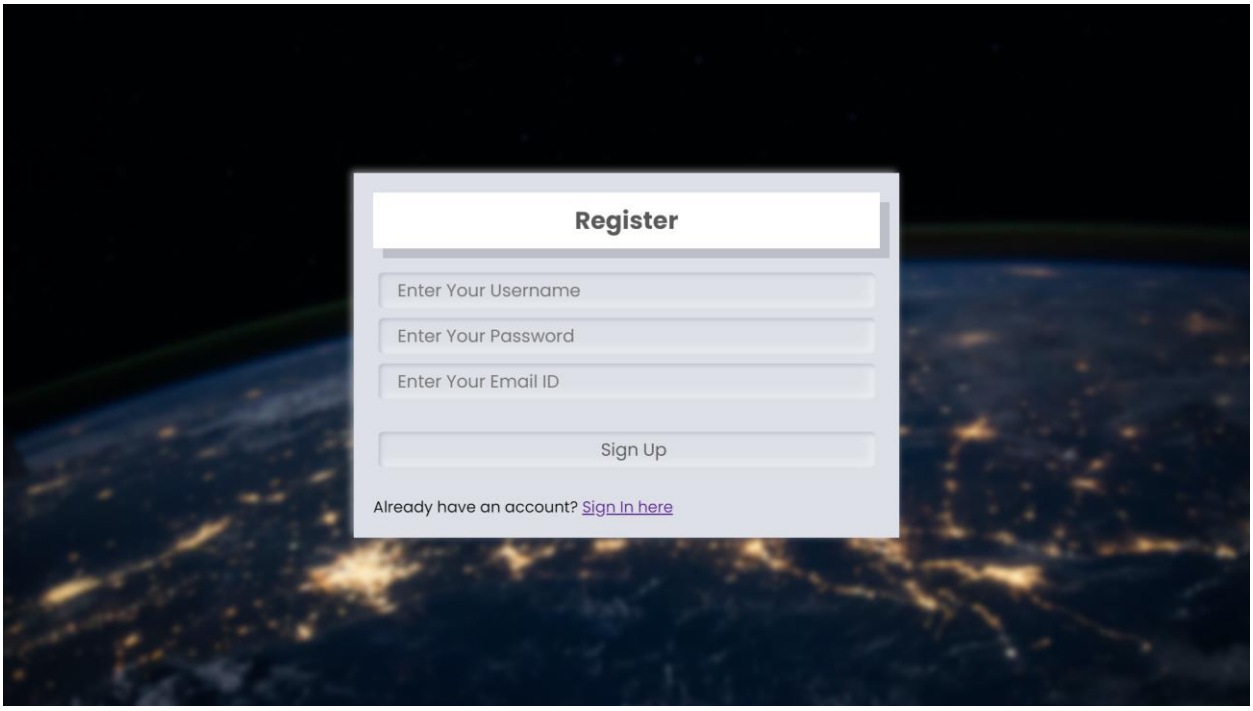

Fig.9.3 Register Page

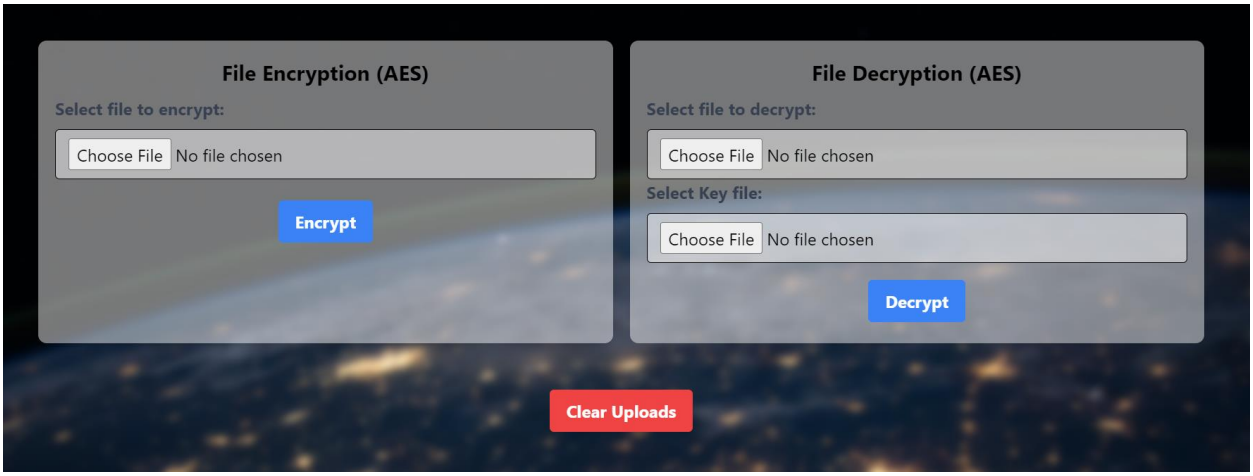

Fig.9.4 File Encryption and Decryption (AES)

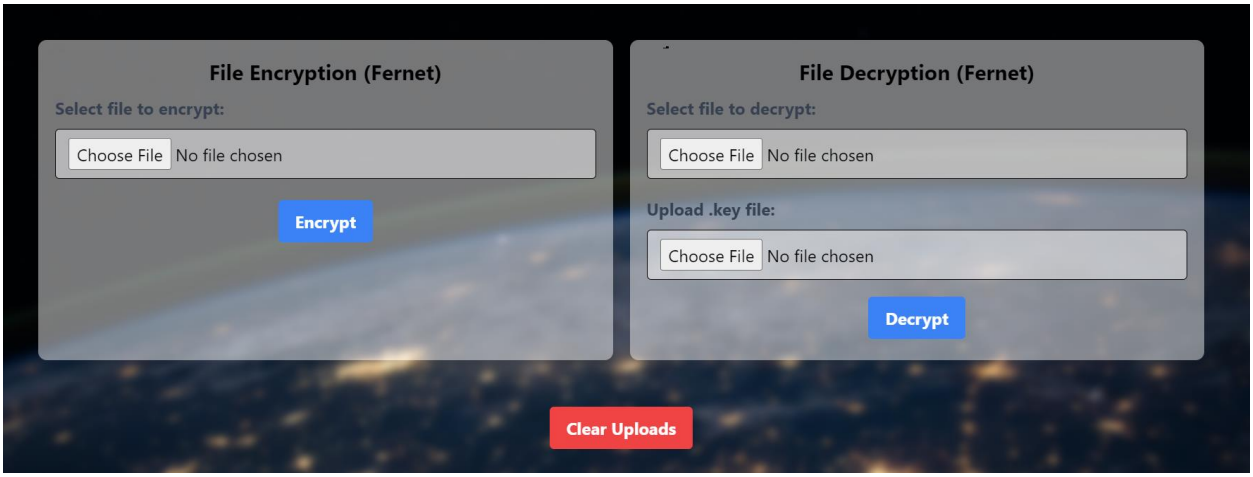

Fig.9.5 File Encryption and Decryption (Fernet)

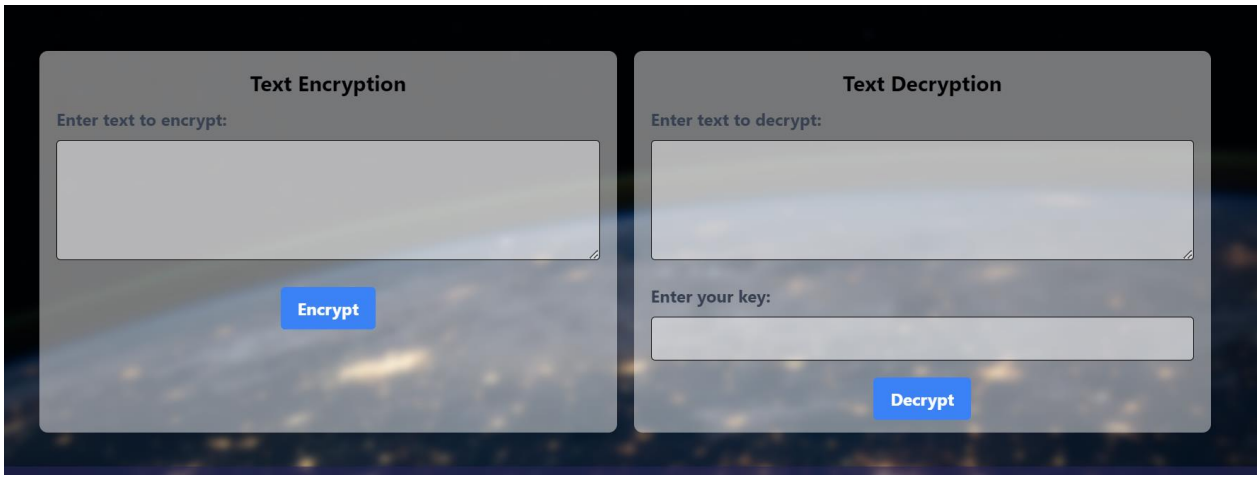

Fig.9.6 Text Encryption and Decryption

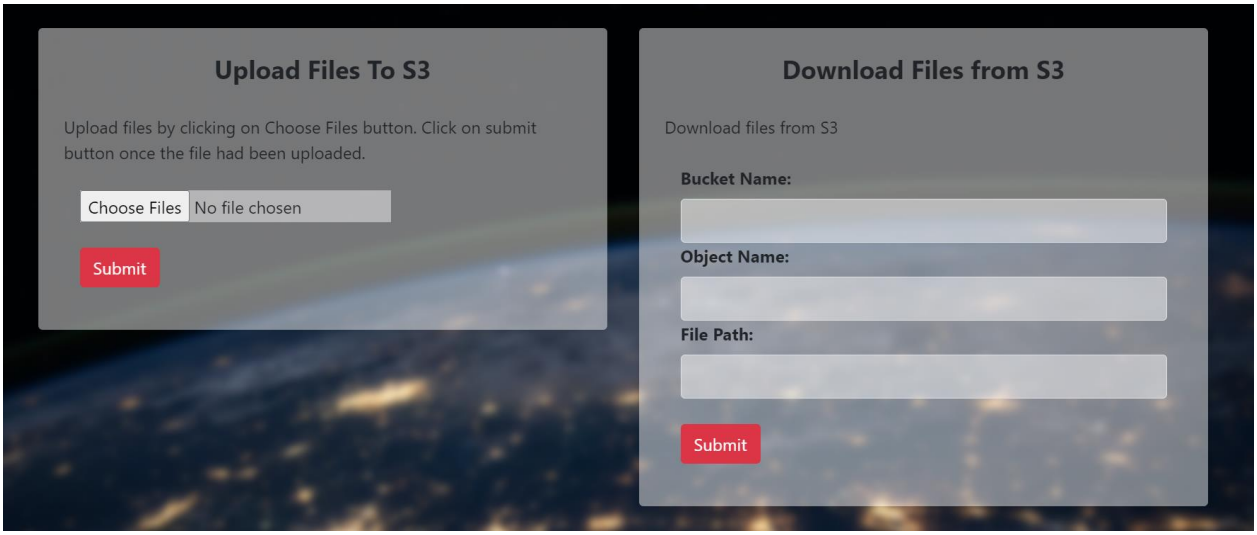

Fig.9.7 Upload and Download (AWS S3)

CHAPTER 10:

CONCLUSION

CONCLUSION

In conclusion, the "Secure Cloud Simulator using Cryptography and Python" project represents a significant milestone in the realm of data security and cloud interaction. By incorporating robust encryption algorithms like Fernet and AES, we have provided users with a powerful tool to safeguard their sensitive files before entrusting them to the cloud. The user-friendly interface, coupled with features like key management and error handling, ensures a seamless and intuitive experience. The integration with AWS S3 further enhances the accessibility and availability of encrypted data. This project not only addresses the critical need for secure cloud storage but also showcases the potential for future advancements in data protection and cloud interaction.

Through extensive research, meticulous design, and rigorous testing, we have created a reliable and efficient solution for users seeking to fortify their data security in the cloud. The positive feedback and successful use cases from early adopters validate the effectiveness and usability of our application.

Ultimately, the "Secure Cloud Simulator" project not only addresses the current needs of users but also lays a strong foundation for continued growth and innovation in the field of secure cloud storage. It is our hope that this project will contribute to a safer and more secure digital landscape for individuals and businesses alike. We are excited to see how this application will make a positive impact in the ever-evolving landscape of data protection and cloud computing.

REFERENCES

1. Qin Liu, Guojun Wang, and Jie Wu“Efficient Sharing of Secure Cloud Storage Services” 2010 .10th IEEE International Conference on Computer and Information Technology (CIT - 2010).
2. Uma Somani, Kanika Lakhani, Manish Mundra“Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing” 2010 IEEE 1st International Conference on Parallel, Distributed and Grid Computing (PDGC - 2010).
3. Ashutosh Kumar Dubey 1, Animesh Kumar Dubey 2, Mayank Namdev3, Shiv Shakti Shrivastava4 “Cloud-User Security Based on R SA and MD5 Algorithm for Resource Attestation and Sharing in Java Environment “in 2011.
4. Xiang Tana, Bo Aib“The Issues of Cloud Computing Security in High-speed Railway “in 2011.
5. Arthur Rahumed, Henry C. H. Chen, Yang Tang, Patrick P. C. Lee, and John C. S. Lui “A Secure Cloud Backup System with Assured Deletion and Version Control” 2011 International Conference on Parallel Processing Workshops.
6. Eman M.Mohamed and Sherif EI-Etriby “Randomness Testing of Modem Encryption Techniques in Cloud Environment” in year 2008