

ACKNOWLEDGEMENT

We would like to express our sincere thanks to all those individuals who assisted us in completing this final year project. The project would not be successful enough without the help of **Dr. D.V Patil**, Head of Department of Computer Engineering. We are thankful for his efforts to help us provide the useful resources throughout the project.

We would also like to thank **Mr. N. V. Alone** Sir, our project guide who guided us thoroughly from time to time. We would like to express our appreciation to our guide who has given us this opportunity to present our project **DECENTRALIZATION OF USER'S DATA ON CLOUD STORAGE USING BLOCKCHAIN**. Where we provide our users with security using our blockchain model. We faced many issues but are thankful to our guide who gave us support and helped us throughout this journey. This project is successful truly due to his co-operation.

We would also like to appreciate our subject teachers and professors who helped us theoretically and cleared our doubts whenever needed.

Finally, we must say thanks to our families and friends for their love, comfort, support, and encouragement to keep moving forward in the phase of obstacles in our lives.

TANAYA SHASHIKAT SHISODE

GAURAV KIRAN PATKARI

RENU ANIL PATIL

SAURABH GAJANAN KAYANDE

ABSTRACT

The purpose of our project is to identify the current need to safeguard information and allow our users to enjoy hassle free storage of data by using an emerging field of “Blockchain Technology” The decentralization of user data and its secure storage are critical concerns in the era of digital information. This research project focuses on the encryption of files using the RSA (Rivest-Shamir-Adleman) algorithm to ensure their confidentiality and integrity. The RSA algorithm, based on the mathematical complexity of prime factorization, provides a robust framework for secure communication and file transfer.

With the help of blockchain we have created a decentralized network that helps us to maintain scalability. Ethereum runs on smart contract functionality where users can sign up and create an account that ensures security to a high level. No chances of data fraud and information leaks.

The Cloud storage service is being used in our project that has many benefits over the common traditional physical storage methods, including more accessible data storage. The significance of our project is that we are able to provide encryption to the user’s data on cloud. We are demonstrating our project’s conclusions using various test cases related to our project.

TABLE OF CONTENTS

Sr. No.	Title of Chapter	Page No.
01	Introduction	07
1.1	Overview	07
1.2	Motivation	08
1.3	Problem Definition and Objectives	09
1.4	Project Scope & Limitations	10
1.5	Methodologies of Problem solving	12
02	Literature Survey	14
03	Software Requirements Specification	15
3.1	Assumptions and Dependencies	15
3.2	Functional Requirements	16
3.2.1	System Feature 1(Functional Requirement)	16
3.2.2	System Feature2 (Functional Requirement)	17
3.3	External Interface Requirements	18
3.3.1	User Interfaces	18
3.3.2	Hardware Interfaces	19
3.3.3	Software Interfaces	19
3.3.4	Communication Interfaces	20
3.4	Nonfunctional Requirements...	21
3.4.1	Performance Requirements	21
3.4.2	Security Requirements	22
3.4.3	Software Quality Attributes	22
3.5	System Requirements	24
3.5.1	Database Requirements	24
3.5.2	Software Requirements (Platform Choice)	25
3.5.3	Hardware Requirements	26
3.6	Analysis Models: SDLC Model to be applied	27
04	System Design	29
4.1	System Architecture	29
4.2	Data Flow Diagrams	31
4.3	UML Diagrams (Use case diagram, class diagram,object diagram, deployment diagram, component diagram, etc)	33
05	Project Plan	35
5.1	Project Estimate	35
5.2	Risk Management	37
5.3	Project Schedule	37
06	Project Implementation	40
6.1	Overview of Project Modules	40
6.2	Tools and Technologies Used	45
07	Software Testing	49

	7.1	Type of Testing	49
	7.2	Test cases & Test Results	50
08		Results	51
	8.1	Outcomes	54
	8.2	Applications	58
09		Conclusion and Future Work	59
10		References	60

LIST OF FIGURES

90

ILLUSTRATION		PAGE NO.
3.1	Home Page	18
3.2	File Upload	19
3.3	Database (User Details)	24
3.4	Database Ethereum Addresses	25
3.6	Waterfall SDLC Model	27
4.1	System Architecture	29
4.2	Data Flow Diagram (Level 0)	31
4.2	Data Flow Diagram (Level 1)	31
4.3	UML Diagram (Class 1)	32
4.5	UML diagram (Class 2)	34
5.1	Risk Management Process	36
6.1	Registration Model	40
6.2	Ethereum Block Module	41
6.3	Ethereum Block Verification Module	42
6.4	File Upload Module	43
6.5	Encryption and Decryption Model (RSA)	44
8.1	Registration Module	54
8.2	Sign Up Page	54
8.3	Login Page	55
8.4	Account Info Page	55
8.5	File Upload Page	56
8.6	File Download List Page	56
8.7	Ethereum Ganache Server	57

LIST OF TABLES

ILLUSTRATION		PAGE NO.
2.1	Literature Survey	14

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Most organizations around the world create huge amounts of data through their day-to-daywork. This data needs to be stored somewhere and should be easily accessible. This storage of easy access to huge data itself becomes a big problem for such organizations. Cloud is an emerging technology provides solutions to this problem by allowing such organizations to store data on cloud storage and it can be easily accessed through the internet. Large cloud providers don't guarantee the security of data being secure. Whereas there are continuous data breaches, and data leakages happening with such cloud-based storages. There are very few options available that could be used to ensure the security of data stored on the cloud.

With the system proposed, we leverage the security of cloud-based data in the blockchain. The results of the system promise to provide maximum security for the data stored on the cloud. As we are making use of the Blockchain's Decentralized functionality we enable multiple legit users to upload their data and guarantee the security by our encryption model The permission-based access control makes the system more reliable trustable, in turn making the data more secure.

Cryptography is a method of securing data from unauthorized access. In the blockchain, cryptography is used to secure transactions taking place between two nodes in a blockchain network. It has a peer-to-peer network that lessens the probability of breaching of user's data. We make use of the RSA cryptographic algorithm in our proposed system. Our project basically runs on 3 most important pillars i.e., Blockchain, Ethereum and XAMPP server. Used for their respective advantages of security, Ease and Maintainability.

1.2 MOTIVATION

We were motivated to build a decentralized secure storage system using blockchain is achieved by distributing the computation tasks to all the nodes of the blockchain network. One of the reasons we chose this subject was our keen interest in decentralization system which solves several problems of traditional systems, the single point of failure is one such problem and it highly depends on the network connectivity. The issues faced in the traditional cloud storage system and the losses due less security drove us forward to go with this project. In regular security systems algorithms and cryptography is used for securing transactions, data, or any kind of information. As these systems are old now, hackers have also learnt them to hack easily and as a result security of data is compromised. So there arises a need of system with commendable security which the users can rely on. Our proposed system uses blockchain which is decentralized in nature i.e it has multiple owners. It becomes impossible for hackers to hack and also data sharing becomes easy due to blockchain Future proofing data storage of Building decentralized user data storage using blockchain technology positions us for the future. It embraces the potential of emerging technologies, such as blockchain, and prepares us for the evolving data landscape. By adopting decentralized storage now, we can stay ahead of the curve and adapt to the changing needs and expectations of users. In Banking companies this system is most useful because the data stored in their system is confidential such as transaction id, available balance, Account Number, Transaction details, and all Money related important things etc are to be taken care by them. Hence the aim of this project is to implement high level security systems over traditional system

1.2 PROBLEM DEFINITION AND OBJECTIVE

Preserving user privacy is a crucial aspect of securing data in a blockchain - based cloud storage system. The problem involves developing techniques and protocols to address privacy challenges, such as hiding sensitive information from unauthorized access, providing selective data sharing options, and implementing privacy-enhancing technologies like zero-knowledge and various proofs or confidential transactions. From a user's perspective the model can bring a wide range of security benefits and opportunities. Traditional storage systems are not that secure and are prone to breaching and sharing of data without permission and are complex sometimes. Enhancing the user experience is an important aspect of securing user data. The problem involves developing user-friendly interfaces, simplifying key management processes, and ensuring seamless integration of blockchain-based cloud storage solutions with existing applications and systems to facilitate easy adoption and usage. The project is aimed at developing a Web Application for securing the user's data efficiently by which the user will experience hassle free secure storage promising complete confidentiality. Our project is mainly focused on online processing of 6 modules. To run the project, we can connect our system to the Ethereum blockchain and upload files on XAMPP server which is used as a cloud storage. Our aim is to build a secured network which will be robust to all kinds of hackings, security threats.

1.3 PROJECT SCOPE AND LIMITATIONS

Project scope –

- 1 The goal of the project is to provide a platform to help service providers improve their behaviors. Current trust models and cloud systems are not that effective.
- 2 The blockchain-based approach for trust-enabled service/resource management in cloud storage systems.
- 3 The Project delivers sustainable security system in an average cost which is affordable.
- 4 The Best practices of our proposed system would be seen in the fields of Software Security, Real Estates, Banking and finance, Shipping and Logistics Etc.

Limitations –

- 1 **Vulnerabilities in Smart Contracts:** Smart contracts, which are an integral part of blockchain systems, can introduce security vulnerabilities. Smart contracts are software programs stored on the blockchain, and if they contain bugs or vulnerabilities, they can be exploited to compromise the security of the stored data. It is crucial to thoroughly audit and test smart contracts to mitigate these risks.
- 2 **Immutable Data:** Blockchain's immutability, which prevents data modification, can be both an advantage and a limitation. If any erroneous or incorrect data is stored on the blockchain, it becomes challenging to rectify or remove that data. This limitation can be problematic if sensitive or incorrect information needs to be updated or deleted due to regulatory requirements or other legitimate reasons.
- 3 **Privacy Challenges:** While blockchain offers transparency and tamper resistance, it can present challenges regarding data privacy. Public blockchains make all transactions and data visible to all participants, compromising the privacy of user data. While private or permissioned blockchains can address this limitation to some extent, ensuring robust

privacy protections in a decentralized blockchain network requires additional mechanisms such as encryption or privacy-enhancing technologies.

- 4 Single Point of Failure:** Although blockchain is decentralized, cloud storage infrastructure can introduce a single point of failure. If the cloud storage provider experiences an outage, suffers a security breach, or fails to maintain data integrity, it can impact the security of the user's data stored on the blockchain. Relying on a single cloud storage provider for data storage introduces potential vulnerabilities and risks.
- 5 Key Management:** Blockchain-based security relies on cryptographic keys to secure data. Proper key management is critical for maintaining the confidentiality and integrity of user data. However, managing cryptographic keys securely can be challenging, especially for non-technical users. If keys are lost, stolen, or mishandled, it can lead to data loss or unauthorized access to sensitive information

1.4 METHODOLOGIES OF PROBLEM SOLVING

1.5.1 Define the problem

The first step to solving a problem is defining what the problem actually is. An effective problem solver will take the solutions of everyone involved in process, but different people might have different ideas on what the root cause of the issue really is

In traditional systems problem were

- Slow and ineffective system
- Breaching of data
- Complicated functionalities
- No guaranteed security.

1.5.2 List all the possible solutions

- Immutable security with SHA Algorithm
- RSA Algorithm for Security
- Google Cloud storage for data upload.
- Centralized system for Transfer of Data
- RSA algorithm and Decentralized system using Blockchain

1.5.3 Evaluate the options

Each option will have pros and cons, and it is important you list all of these, as well as how each solution could impact our app users. It is important to evaluate each of the option by taking into consideration its both sides.

1.5.4 Select the best solution

Above all the possible solutions that we have thought for our problem, We have chosen the best option after evaluating each one of them thoroughly. Among all the best option is Option no 5 i.e Using RSA Algorithm and decentralized system we will build our model on cloud storage. As Blockchain will allow multiple users to handle and RSA will give high encryption that perfectly is best solution for our problem.

1.5.5. Create an implementation plan

We will put our best solution into practice by physically practicing the RSA algorithm and testing its results. Also, we will check the working of our blockchain system to make sure it is scalable and easy to handle. The potential of the XAMPP server will be tested by performing use cases to make sure it works well.

1.5.6. Communicate your solution

There is one last step to consider as part of the problem-solving methodology, and that's communicating your solution. Without this crucial part of the process, we won't be able to know that the model we have made is perfectly solving our issues or not. Not everyone will be satisfied with our work, but we have to make sure to keep in mind our aim.

1.5.7 Evaluate the outcome

This will be the last step of problem solving. We will test our solution by applying various use cases and monitoring the results for each and every test case. Evaluation will be done on the basis of the results obtained and accordingly changes will be made if any.

CHAPTER 2

LITERATURE SURVEY

SR No	Name of the author/Publish Date	Name of Paper	Approach
1	Aishwarya Patil 06 June 2021	Securing cloud-based data storage using blockchain	This paper sharing is backed by various cloud providers that allows customers to store and share data on internet.
2	Manikandan D, Valliyanmai C, Karthika RN 04 Nov 2020	Blockchain based secure big data storage on cloud	The main goal of this paper is to introduce a system that leverages the security of cloud-based data on the blockchain.
3	Dhananjay Yadav, Aditi Shinde, Akash Nair 18 Oct 2019	Enhancing data security in cloud using blockchain	Aim of this paper is to make existing cloud storage more secure and decrease the data breaches and attacks.
4	Sarmah, S.S.	Application Of Blockchain On cloud computing	Goal of this paper is to introduce blockchain's applications in field of Cloud computing.

Table: 2.1: Literature Survey

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 ASSUMPTIONS AND DEPENDENCIES

Assumptions:

- Users must have basic knowledge of login and sign up
- The project assumes the availability and functionality of a suitable blockchain infrastructure, whether it is a public blockchain or a private /permissioned blockchain network.
- Users must be having prior basic knowledge of the Blockchain and web systems
- User must know how to run the XAMPP server and basic operations on Ethereum must be known.
- Users and stake holders should be willing to embrace decentralized storages solutions and understand the benefits and implications of utilizing blockchain for data management.

Dependency:

- The users possess a computer.
- The users have the internet facility available.
- The users have the web browser installed in their system.

3.2 FUNCTIONAL REQUIREMENTS

System Feature 1

- If/Then behaviors: We have implemented If/then loops for verification purpose. If OTP is correct then user can log in condition, they are priorly signed in
- Data handling system: Xampp is used for data handling.
- System workflows: System works in manner web pages are organized.
- Administrative functions: These includes administrative rights given to CEO, staff and office section
- Performance requirements: These are requirements to enhance System Performance. Ex GPUs
- Details of operations conducted for every screen.
- The functional requirements specify relationship between the inputs and outputs
- All the operations to be performed on the input data to obtain output are to be specified. This includes specifying the validity checks on the input and output data, parameters affected by the operations and the other operations, which must be used to transform the inputs into output.
- Functional requirements specify the behavior of the system for valid input and outputs.

System Feature 2

User Authentication: Implement a secure user authentication mechanism to ensure that only authorized users can access the system. This feature can include login credentials, multi-factor authentication, and password hashing to protect user accounts.

Blockchain Integration: Integrate a blockchain network, such as Ethereum, to store and manage encrypted user data. This feature involves interacting with the blockchain through smart contracts to ensure the immutability, transparency, and integrity of the stored data.

Distributed Storage: Implement a distributed storage system that utilizes blockchain technology to store and manage user data across multiple nodes in a decentralized manner. This feature ensures that user data is not stored in a single central location, enhancing data availability and resilience

Data Ownership and Control: Empower users with ownership and control over their data stored in the decentralized storage system. This feature includes providing users with the ability to grant or revoke access to their data, track data usage, and have transparency over how their data is being handled.

Secure Key Management: Develop a robust key management system to generate, store, and manage cryptographic keys used for data encryption and decryption. This feature includes secure key storage, key rotation, and protection against key loss or theft.

Access Control and Permissions: Implement access control mechanisms to regulate user access and permissions within the system. This feature ensures that only authorized individuals can view, modify, or delete specific data based on their roles and permissions.

EXTERNAL INTERFACE REQUIREMENTS

3.3.1 USER INTERFACES

Admin Dashboard UI:

An admin dashboard UI is designed specifically for administrators to manage and monitor the system. It provides features for user management, access control settings, generating reports, and monitoring system health and security

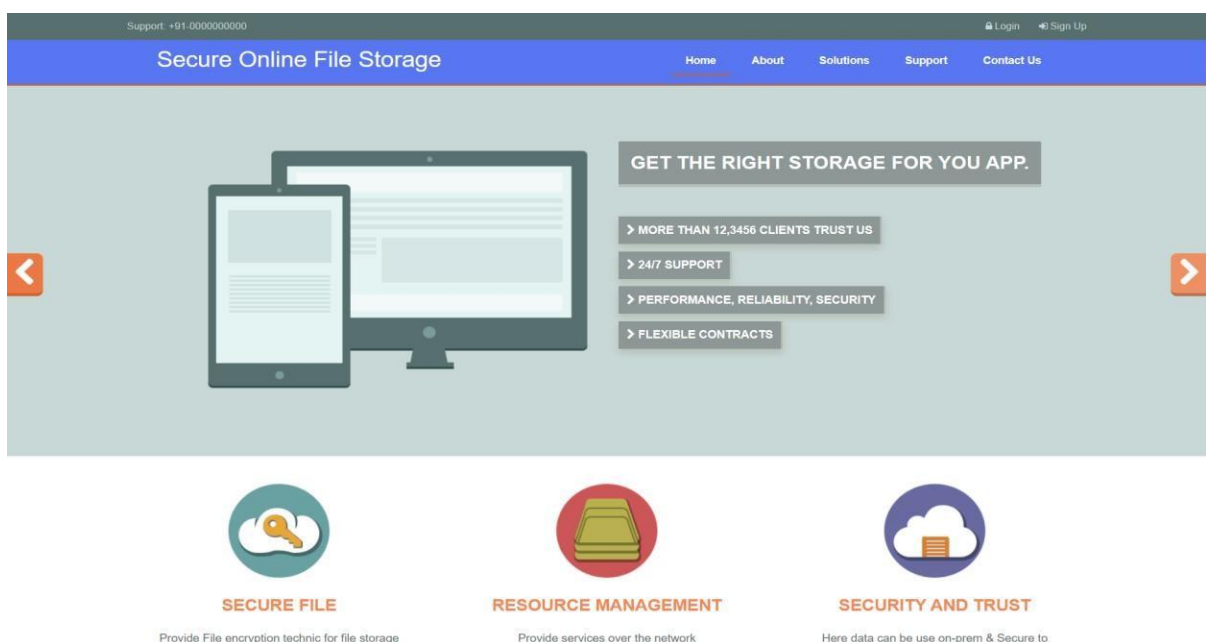


Figure 3.3.1: Home page

- **Command-Line Interface (CLI):** A command-line interface allows users to interact with the system through text-based commands. It can provide a more technical and efficient way for advanced users to perform operations such as uploading files, accessing metadata, and managing files securely.

Figure 3.3.2 : File Upload Page

3.3.2 HARDWARE INTERFACE

- Memory 2GB minimum, 4GB recommended
- Processor – intel i5 or i7
- Operating system – windows
- 256 SSD
- Intel HD 610(GTI)
- Screen resolution of 1366*768 pixels

3.3.3 SOFTWARE INTERFACE

The various types of user interfaces include:

- Graphical user interface (GUI): This includes Graphics performance driven by GraphicsCard
- Database Server: The system uses phpMyAdmin as the database server.

- **Web Server:** The system functions on XAMPP, using Apache server.
- **Form-based user interface:** This is interface generated by forms using HTML and using POST and GET methods forms connect to Database and PHP files.

3.3.4 COMMUNICATION INTERFACES

1. Web browser:

Opera web browsers have different versions. Chrome, Opera V1, Opera V2, Opera 3, etc, we need to decide whether our website must work ok with all versions or just on a new one.

2. Communication standards and Network server communications protocols:

HTTP, HTTPS, or FTP

HTTPS is the secure version of HTTP where communication(s) between the browser and the website are encrypted by TLS or SSL, its predecessor. Ultimately, FTP is more efficient at transferring large files, whereas HTTP is better for transferring smaller files such as web pages.

3. Electronic forms:

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application..

4. Blockchain:

Blockchain is used to establish peer to peer communication and perform tasks from any corner of the world to another with full security

3.4 NONFUNCTIONAL REQUIREMENTS

3.4.1 PERFORMANCE REQUIREMENTS

- **Throughput:** The system should have sufficient throughput to handle the expected volume of data transactions and storage requests. It should be able to process and store user data efficiently without significant delays or bottlenecks, ensuring a smooth user experience.
- **Latency:** The system should provide low latency for data retrieval and storage operations. Users should be able to access their data quickly and experience minimal delays when uploading, downloading, or modifying data stored in the cloud.
- **Scalability:** The system should be designed to scale effectively to accommodate increasing amounts of user data and growing user demands. It should handle concurrent transactions and storage requests efficiently, ensuring that performance remains consistent even as the system grows and usage.
- **Storage Capacity:** The system should provide sufficient storage capacity to accommodate the anticipated amount of user data. It should be able to scale the storage infrastructure, whether through cloud storage providers or decentralized storage networks.

3.4.2. SECURITY REQUIREMENTS

- **Data privacy:** The proposed system must ensure that the users must not face any kind of breaching and frauds. Public Blockchains are difficult to maintain Data Privacy, especially when sensitive information is stored on the blockchain.
- **Data Encryption:** User data should be encrypted using robust encryption algorithms before storing it in the cloud or on the blockchain. Encryption should be applied both during data transmission and at rest to protect data confidentiality and prevent unauthorized access.
- **Smart Contract:** Smart contract are self-executing contracts with predefined rules and conditions encoded within the blockchain. They are an integral part of blockchain technology and enable automated, secure, and transparent transactions. Here are the key functionalities of smart contracts
- **Sign-Up 2 Step Authentication:** 2-Step Verification is enabled, whenever you log in to your Google account from a new device or browser, you will be prompted to enter the verification code or complete the verification process using your chosen method. This adds an extra layer of security by requiring something you know (password) and something you have (verification code or device) to access your Google account.

3.4.3 SOFTWARE QUALITY ATTRIBUTES

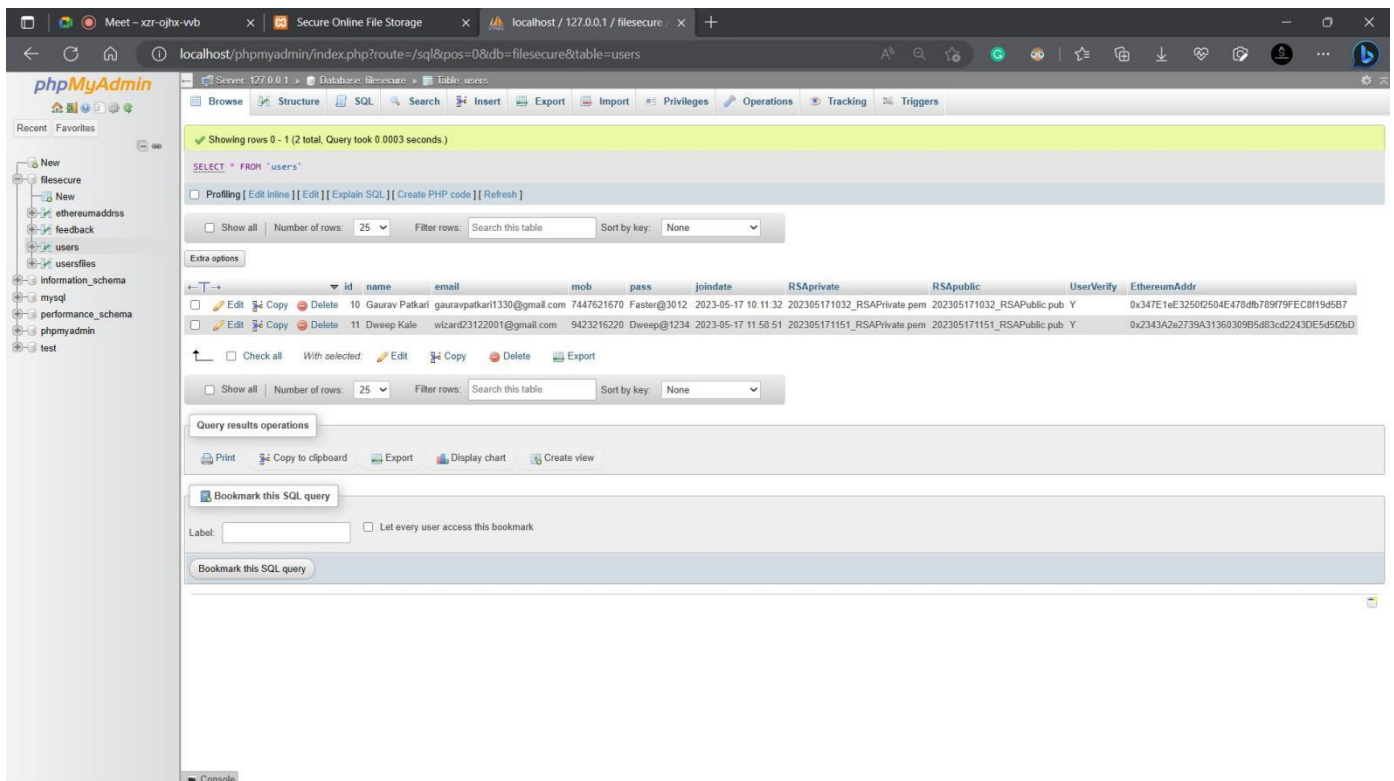
- **Performance:** Ensure that the system performs efficiently and responds quickly to user requests. Consider factors such as throughput, latency, response time, and scalability to handle increasing data volumes and user demands.
- **Reliability:** The system should be reliable, with high availability and minimal downtime. It should be designed to handle failures gracefully, provide fault tolerance, and recover from errors or disruptions without data loss or service interruption.
- **Security:** Implement robust security measures to protect user data, ensure data confidentiality and integrity, and prevent unauthorized access or tampering. This includes encryption, access control, secure communication protocols, and adherence to industry best practices for blockchain security.

- **Usability:** Design the system with a user-friendly interface and intuitive interaction flows. Users should be able to easily navigate, access, and manage their data, regardless of their technical expertise.
- **Scalability:** Ensure that the system can scale effectively to accommodate growing data volumes and user base. The architecture should support horizontal scalability, allowing for the addition of more nodes or storage capacity as needed, without significant performance

SYSTEM REQUIREMENTS

3.5.1 Database Requirements

- User Data: The user's data and credentials are stored in the php database. All the data is stored in an organized way. The data consists of attributes of Name, email id, Mobile number, Password, RSA Private key, RSA Public key and Ethereum Address respectively.
- Ethereum Transaction: Creates Ethereum address for every file upload process and individually for every user is displayed separately.
- User's File list: List of files will be displayed in the database.



The screenshot shows the phpMyAdmin interface for a database named 'filesecure'. The 'users' table is selected, and the query results are displayed. The table has 11 columns: id, name, email, mob, pass, joindate, RSAPrivate, RSAPublic, UserVerify, and EthereumAddr. Two rows of user data are visible.

	id	name	email	mob	pass	joindate	RSAPrivate	RSAPublic	UserVerify	EthereumAddr
<input type="checkbox"/>	10	Gaurav Patkari	gauravpatkari1330@gmail.com	7447621670	Faster@3012	2023-05-17 10:11:32	202305171032_RSAPrivate.pem	202305171032_RSAPublic.pub	Y	0x347E1eE32502504E478db78979FEC819d5B7
<input type="checkbox"/>	11	Dweep Kale	wizard23122001@gmail.com	9423216220	Dweep@1234	2023-05-17 11:58:51	202305171151_RSAPrivate.pem	202305171151_RSAPublic.pub	Y	0x2343A2e2739A31360309B5d83cd2243DE5d52bD

Figure 3.5.1: Database User Details (PHP)

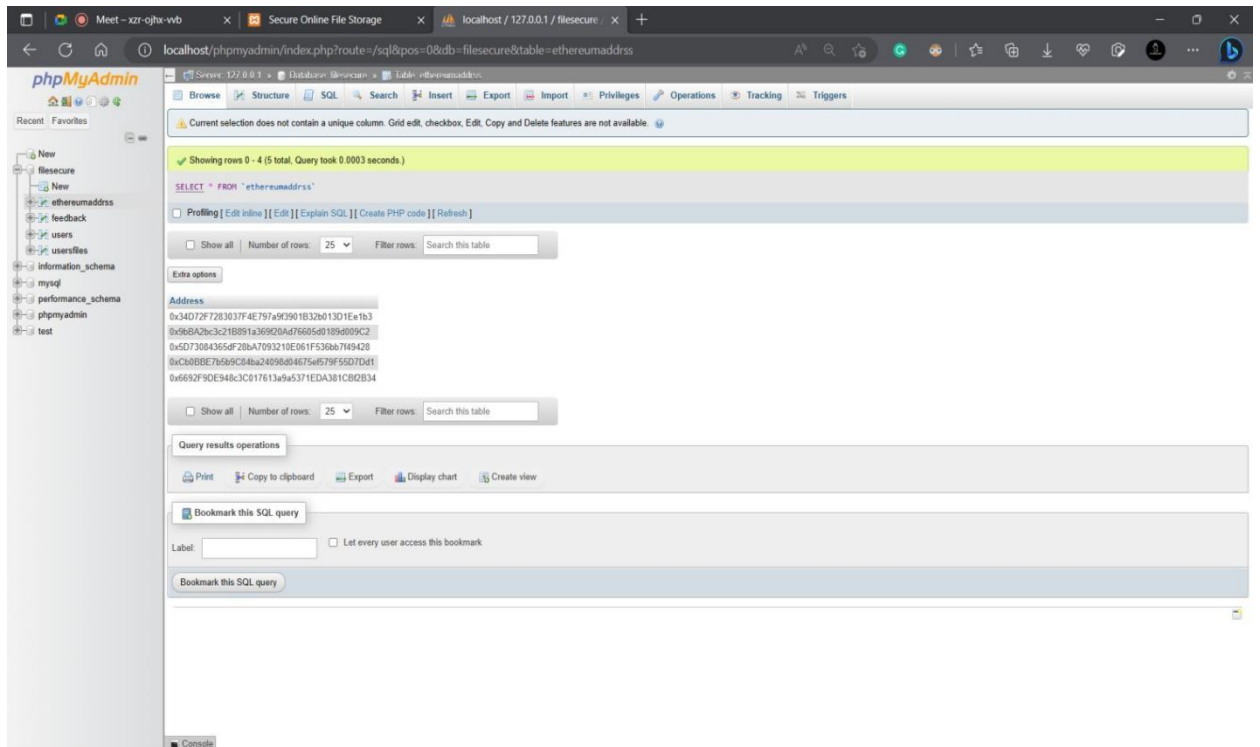


Figure 3.5.2: Database Ethereum Addresses (PHP)

3.5.2 SOFTWARE REQUIREMENTS (PLATFORM CHOICE)

- **Visual Code Studio:** Visual Studio (VS) is a powerful integrated development environment (IDE) developed by Microsoft. It provides a comprehensive set of tools and features for software development across various platforms, programming languages, and frameworks. Visual Studio supports a wide range of programming languages, including C#, C++, Visual Basic, F#, Python, JavaScript, and many more.
- **PHP:** PHP is a popular server-side scripting language used for web development. When it comes to choosing a database for project deployment with PHP, there are several options available. Some of the commonly used databases with PHP are:

XAMPP: It is a popular open-source software package that provides a complete development environment for web developers. The acronym XAMPP stands for cross-platform (X), Apache (A), MariaDB/MySQL (M), PHP (P), and Perl (P). It is designed to simplify the setup and configuration of a local web server environment on your computer. XAMPP includes the Apache web server, which is

one of the most widely used web servers in the world. Apache provides the infrastructure for serving web pages and handling HTTP

- **Ethereum Ganache:** Ethereum Ganache, formerly known as "TestRPC," is a development and testing tool for Ethereum blockchain applications. It is part of the Truffle Suite, a suite of tools that facilitates the development, testing, and deployment of Ethereum smart contracts and decentralized applications (dApps). It provides a local, in-memory Ethereum blockchain environment that allows developers to create a private test network for their applications. It is designed to simulate the behavior of a real Ethereum network but in a controlled and deterministic manner, making it suitable.
- **Webserver:** Flask can be run using a built-in development server or a production-ready web server, such as Apache or Nginx. The choice of web server depends on The specific requirements and needs of the application

3.5.3 HARDWARE REQUIREMENTS

- **A System:** A computer or laptop with good processor and SSD is needed to carry out all the functions in our proposed system.
- **Network:** A high-speed internet connection with sufficient band width is required to handle the expected user traffic and deliver news articles in a timely manner.
- **RAM:** The amount of RAM required will depend on the size of the database and the number of concurrent users. A minimum of 4GB RAM is recommended, but higher amounts may be needed for larger systems.

3.6 ANALYSIS MODELS

We decided to adopt an iterative software development life cycle (SDLC) model for the project. We chose the Waterfall model, which allows the linear sequential flow, which indicates that any development step process can start only after the previous one has finished technology stack. We followed best practices in coding, testing and deployment,

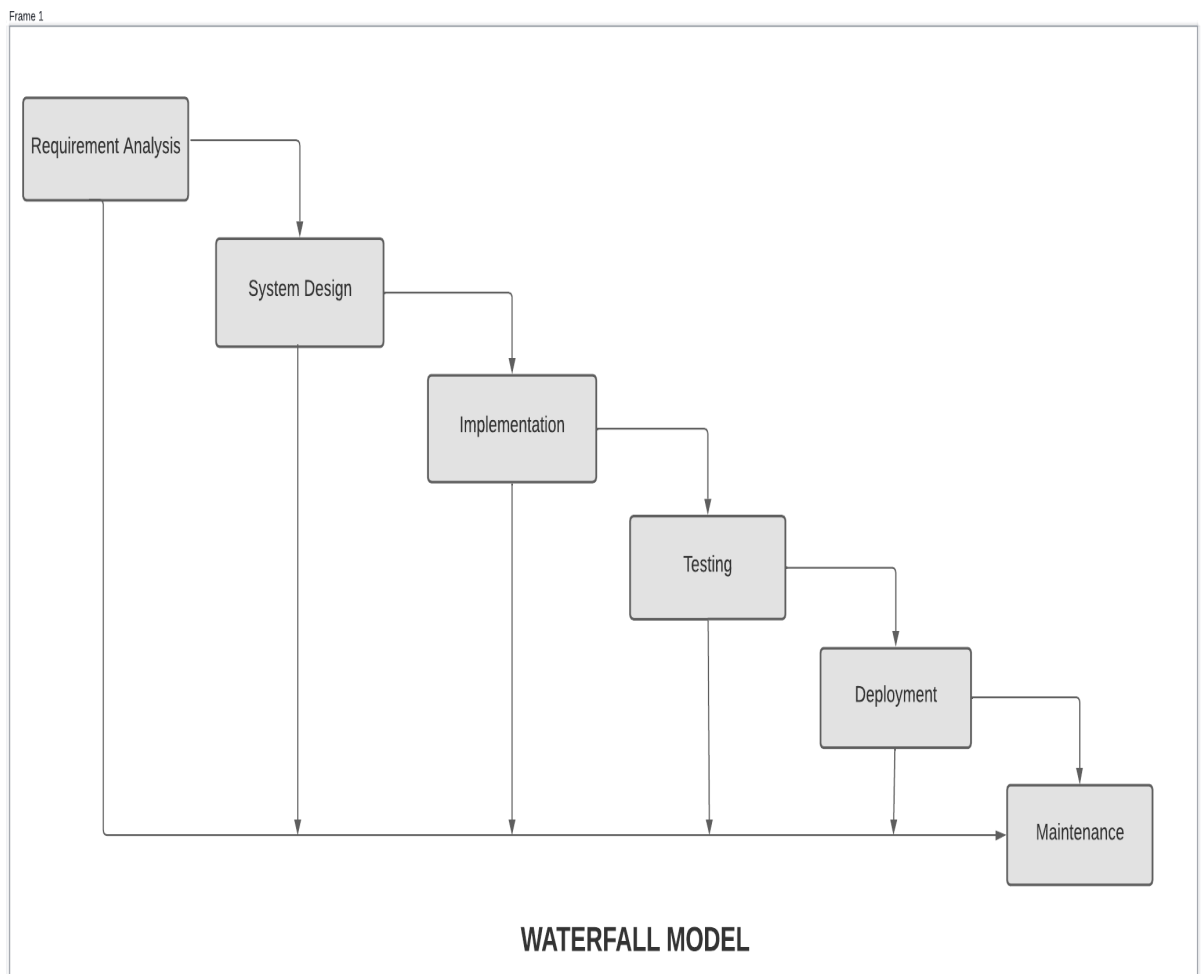


Figure 3.6.1: Waterfall SDLC Model

The waterfall methodology is composed of seven non-overlapping stages:

- 1. Requirements:** Potential requirements, deadlines guidelines for the project are analyzed and placed into a functional specification. This stage handles the defining and planning of the project without mentioning specific processes.
- 2. Analysis:** The system specifications are analyzed to generate product models and business will guide production. This is also when financial and technical resources are audited for feasibility.
- 3. Design:** A design specification document is created to outline technical design requirements such as programming language, hardware, data sources, architecture, and services.
- 4. Coding/Implementation:** The source developed using the models, logic requirements designated in the prior stages. Typically, the system is designed in smaller components, or units, before being implemented together.
- 5. Testing:** This is when quality assurance, unit, beta tests take place to report issues that may need to be resolved. This may cause a forced repeat of the coding stage for debugging.
- 6. Operation/Deployment:** The product or application is deemed fully functional and is deployed to a live environment.
- 7. Maintenance:** Corrective, adaptive and perfective maintenance is carried out.

CHAPTER 04

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

System Architecture consists of the interconnection of the system. The system is then connected to the database. The association, dependency, generalization among the classes are represented.

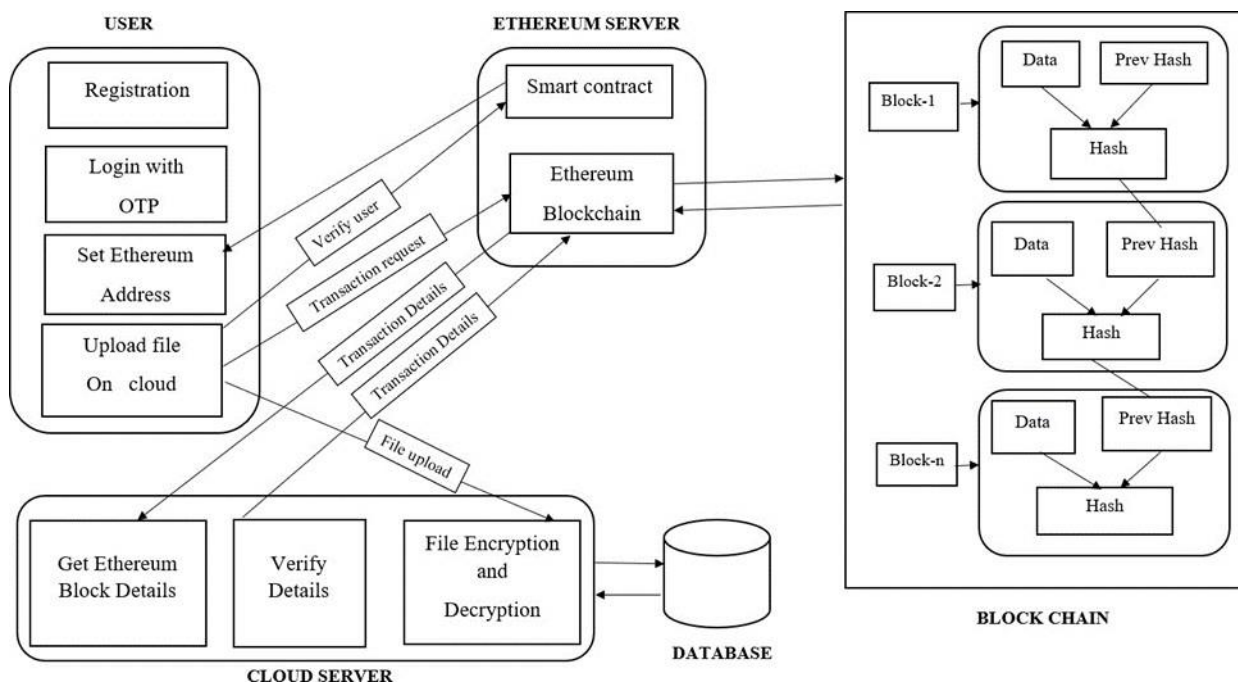


Figure 4.1: System Architecture

- To provide login facility to user, first user can register with us and then login, every time login user gets OTP on email for user security. After that user sets the Ethereum address as we provide facility to user to set Ethereum address instead of allotting randomly. After login user can upload file on cloud, at that time the file encrypted using RSA algorithm, then verify user throw smart contract of Ethereum. Then generate Ethereum block for that transaction related to upload file. Ethereum block details contain block id, block hash and transaction hash value. This value is stored on system server. This hash value is used just for verification because blockchain provide trust solutions.

- To provide login facility to user, first user can register with us and then login, every time login user gets OTP on email for user security. After that user set Ethereum address or also we can provide facility to user to set Ethereum address. After login user can upload file on cloud, at that time file encrypted using RSA algorithm, then verify user throw smart contract of Ethereum. Then generate Ethereum block for that transaction related to upload file. Ethereum block details contain block id, block hash and transaction hash value. This value is stored on system server. This hash value is used just for verification because block chain provide trust solutions.

1. User Layer

For all users we provide login facility, first user can register with us and then login.

User Get OTP on her mail id then, after OTP verification then user login successfully with our system

2. Ethereum Server layer

Each user own Ethereum address which is must to verify through Ethereum server using Ethereum smart contract at time of file upload. Generate Ethereum block for every file or data upload on server, each block contains a block id, block hash and transaction hash value.

3. Blockchain Layer

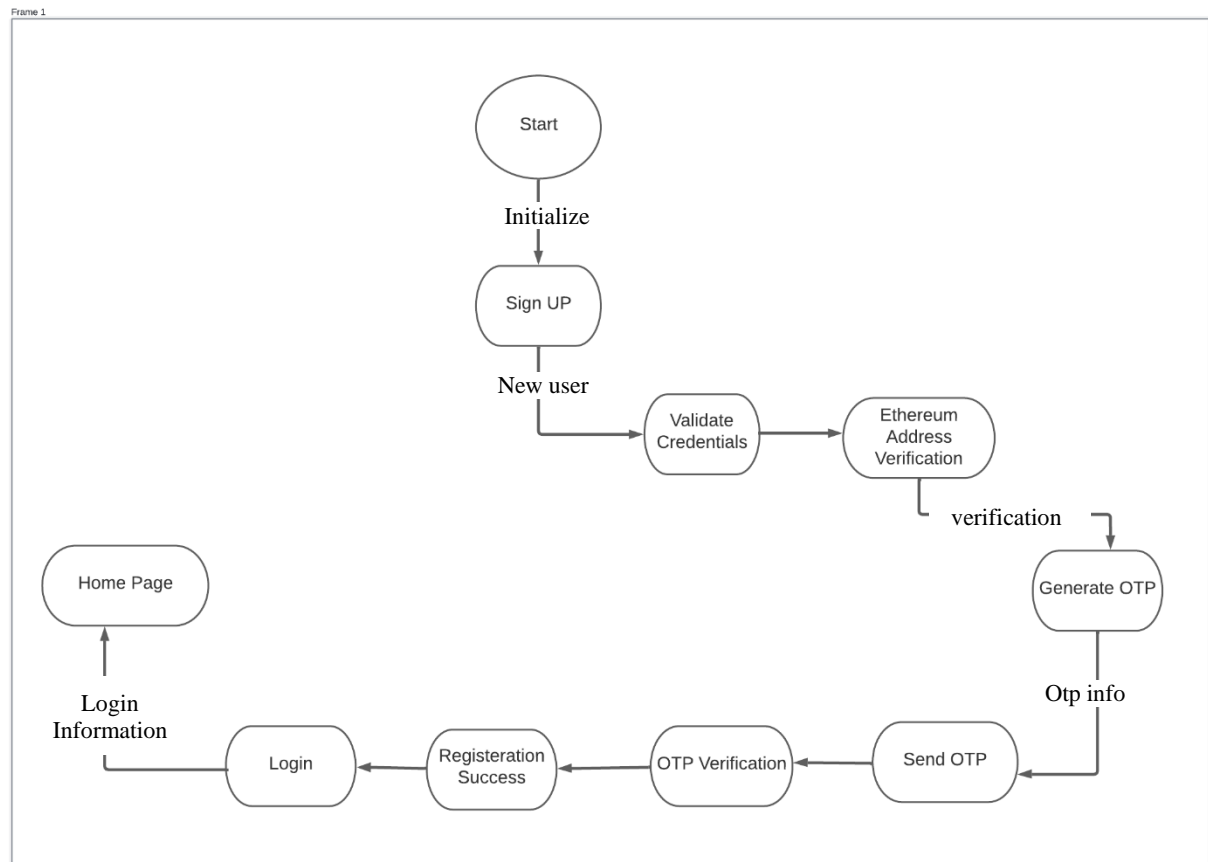
After Block generate, we store hash value on system server, this hash value is used just for verification because blockchain provide trust solutions.

4. Cloud Server Layer

After login user can upload file on cloud, whatever data file user wants to upload.

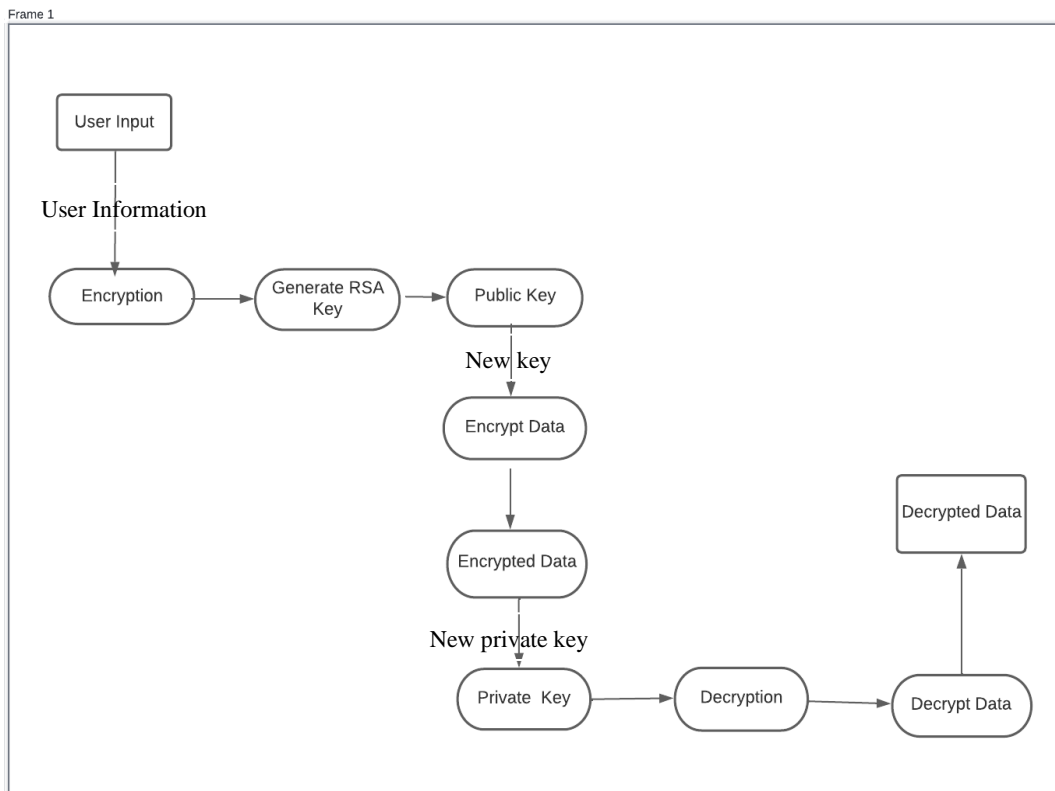
The uploaded file display in the cloud storage used in the proposed system.

DATA FLOW DIAGRAMS

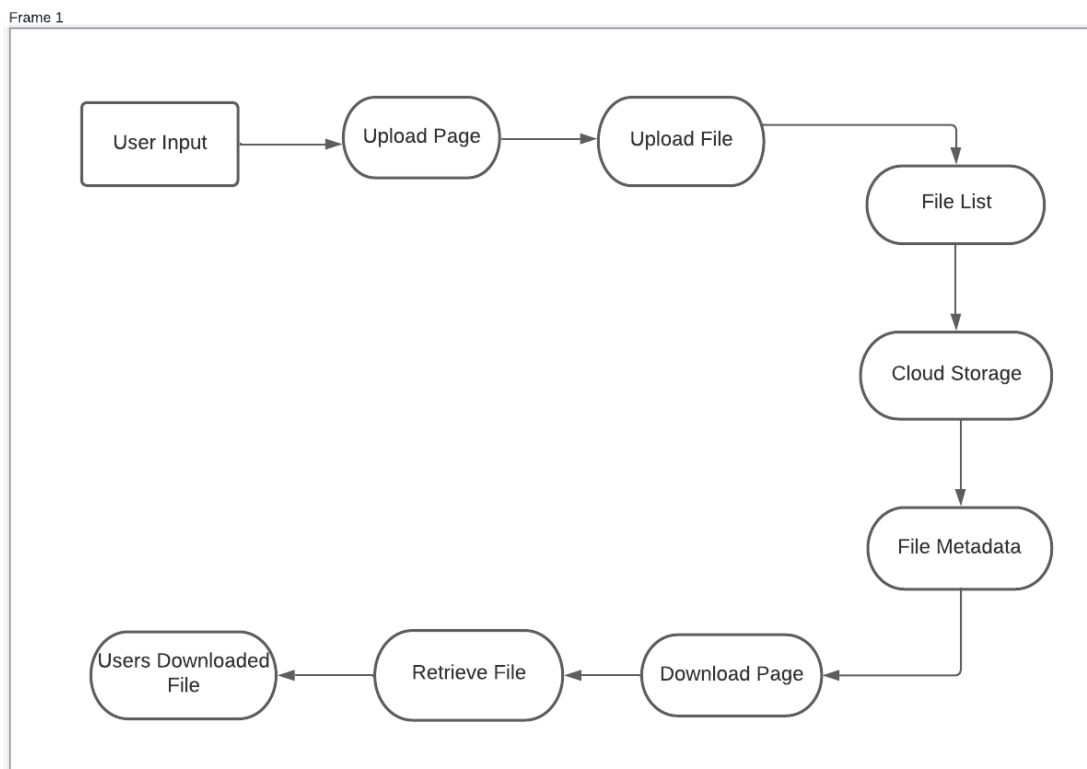


Data Flow Diagram Level 0

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. The level 0 DFD defines the basic first level of the project i.e the sign up step.



Data Flow Diagram Level 1



Data Flow Diagram Level 2

UML DIAGRAMS

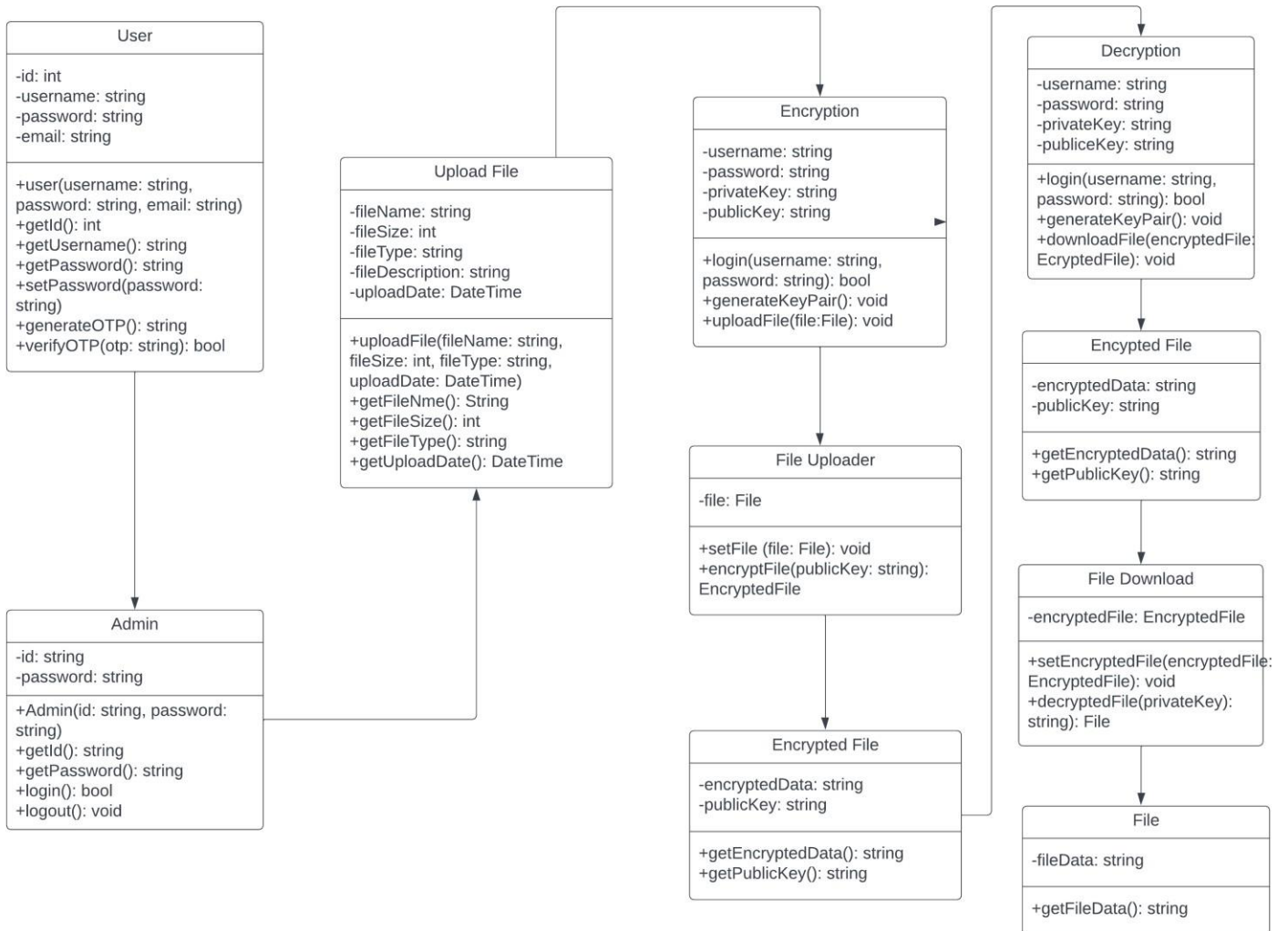


Figure 4.3: UML diagram (CLASS) 1

Here, every section is represented in the form of class and its attributes and interconnections between the modules are represented. It shows association, dependency, generalization among the classes. The actor firstly visits the home page of the website then next is the sign-up step and the registration is done after verifying. Ethereum address is verified in the next step, to make sure no other person except the authorized one can access any other person's Ethereum and after OTP verification the login is successful and file is uploaded.

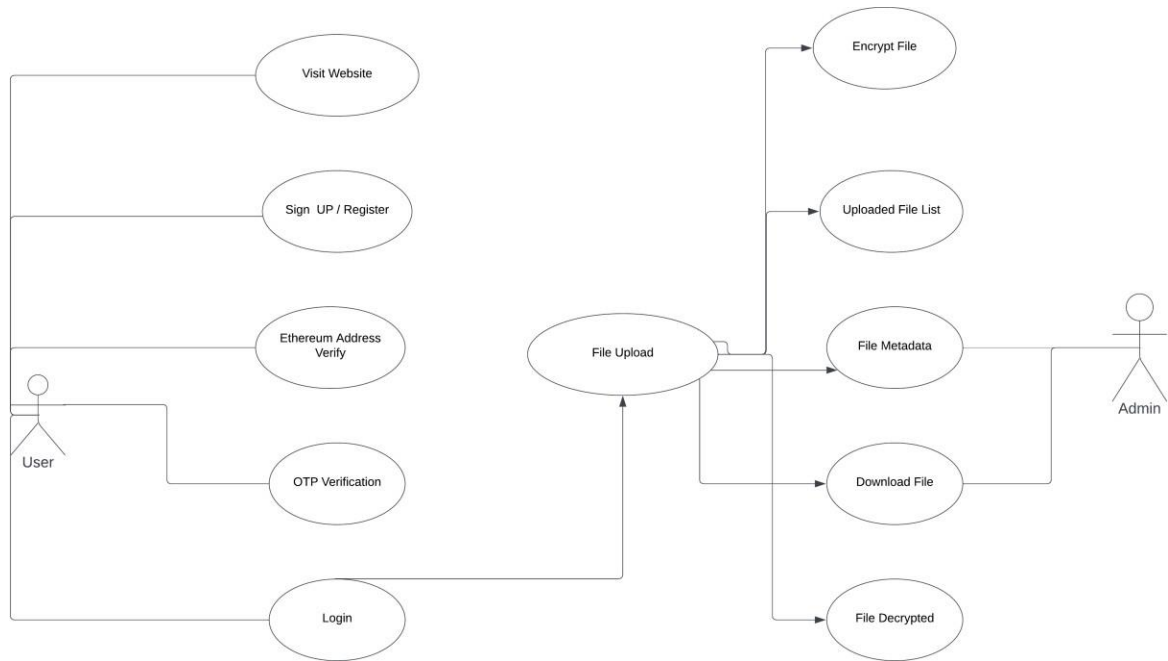


Figure 4.3: UML diagram (class) 2

CHAPTER 05

PROJECT PLAN

5.1 PROJECT ESTIMATION:

Project estimation is a complex process that revolved around predicting the time, cost, and scope that a project requires to be deemed finished. But in terms of software development or software engineering, it also takes the experience of the software development company, the technique they must utilize, the process they need to follow in order to finish the project (Software Development Life Cycle). Project Estimation requires the use of complex tools & good mathematical as well as knowledge about planning.

In most cases, the whole estimation process would cost the company rather considerable cost & time at the very first stage of developing a brand-new website, app, or software. However, this will act as the stepping stone to make the result more credible, realistic, and customer-satisfying.

Estimates:

1. Cost: Development Cost 15000 INR
2. Time: 7 Months
3. Size or scope: Institutional Level
4. Risk: Budget Risks
5. Resources: Google, chatGPT, Research papers
6. Quality – Bug Free and Robust

5.2.1. RISK MANAGEMENT

- Identify potential risks that could impact the development and operation of the blockchain Ethereum. These risks could include technical, operational, legal, or financial aspects.

Example risks: Data breaches, API limitations, scalability issues, content legality, user privacy concern

Risk Management Process:

Risk Management process can be easily understood with use of the following workflow:



Figure 5.1: Risk Management Process

Risk Management Practices:

- Software Risk Evaluation (SRE)
- Continuous Risk Management (CRM)
- Team Risk Management (TRM)

What is Risk Analysis?

Risk Analysis in project management is a sequence of processes to identify the factors that may affect a project's success. These processes include risk identification, analysis of risks, risk management and control, etc. Proper risk analysis helps to control possible future events that may harm the overall project. It is more of a pro-active than a reactive process.

5.2.3 Overview of Risk Mitigation, Monitoring, Management

- Implement security measures, such as encryption, secure authentication, and access controls, to protect user data.
- Continuously monitor the system for potential risks and vulnerabilities.
- Utilize logging and monitoring tools to detect and respond to security incidents and system anomalies.
- Regularly review and update risk assessment based on new information, changes in technology, or emerging threats.
- Establish a risk management plan that outlines roles, responsibilities, and procedures for risk mitigation and response.
- Regularly communicate and train team members on risk management strategies and protocols.

Periodically review and update the risk management plan to reflect changes in the system, technology, or regulatory environment

Cost management plan

Project Schedule :

1. Define project requirements and specifications.
2. Design the user interface for the system
3. Develop the back-end functionality using PHP and Ethereum Ganache.
4. Verify the Ethereum address to the user account for uploading process.
5. Implement user registration, login, and OTP authentication with the help of verified Ethereum address.
6. Set up database schema and data models for users account information.
7. Process and store the users account information data in the database.

8. Implement the uploading process in the system.
9. Use the metadata for details of file upload transaction in the system.
10. The uploaded file encrypted with the RSA algorithm in the system with key generation process in the blockchain.
11. Uploaded file data shown in the file list in the application.
12. The uploaded encrypted file save in the cloud storage i.e local storage
13. Address any bugs or issues identified during testing.
14. Conduct user decryption process of the file data using public key stored in the system.
15. Download the decrypted file from the file in the system.
15. Prepare for production deployment and launch.

5.1 TEAMORGANIZATION

5.4.1 Team Structure

1. Development Team Members:

- Full Stack Developers: Responsible for developing the back-end functionality using Flask, integrating with databases, and implementing blockchain process and processing logic.
- Front-End Developers: Responsible for designing and implementing the user interface using HTML, CSS, and JavaScript frameworks.
- Database Specialist: Responsible for database design, setup, and ensuring efficient storage and retrieval of data.
- Testing/QA Specialist: Responsible for testing the system, identifying, and reporting bugs, and ensuring proper functionality.

2. Project Guide:

- The project guide provides guidance, support, and expertise throughout the project.

- Helps in the initial project planning, requirements gathering, and technical advice.
- Reviews project progress, offers suggestions, and provides feedback for improvement.

5.4.2 Management Reporting and Communication

1. Regular Meetings:

- Conduct regular team meetings to discuss project progress, challenges, and updates.
- Weekly or bi-weekly meetings can help ensure everyone is aligned and aware of individual and team tasks.

2. Task Assignments:

- Clearly define and assign tasks to each team member based on their roles and responsibilities.
- Ensure everyone understands their assigned tasks and the expected deadlines.

3. Collaborative Documentation:

- Utilize collaborative platforms like Google Docs or Microsoft Teams to share project-related documents, including project plans, design specifications, and meeting minutes.
- Encourage team members to contribute and provide input to enhance documentation and maintain a centralized knowledge base.

4. Reporting Structure:

- Prepare regular project reports that highlight the overall project status, achieved milestones, and upcoming tasks.
- Share project reports with the project guide.

CHAPTER 06

PROJECT IMPLEMENTATION

OVERVIEW OF MODULES

1. Registration Module

This module helps the user to register on the system. Verification is done by sending an OTP to the user's email id and after successful verification the user, the user can log in to their created account and that completes the initial process of the system successfully.



Figure 6.1: Registration module

2. Set Ethereum address and Transaction (Ethereum block module)

In this module we set an Ethereum address individually for every user that has successfully completed the login. Whenever a file is uploaded it consists of a unique block id or address that differentiates one account with another. Each block contains a block id, block hash and transaction hash value.

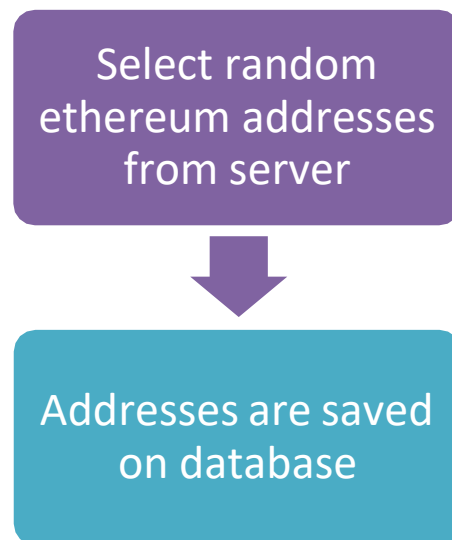


Figure 6.2: Ethereum Block module

3. Ethereum Block verification module

This module makes sure that the ethereum address is verified. The need of this module is that, none of the users must face hacking or breach and loss of account or data. The hash value is stored in the server and then verified.

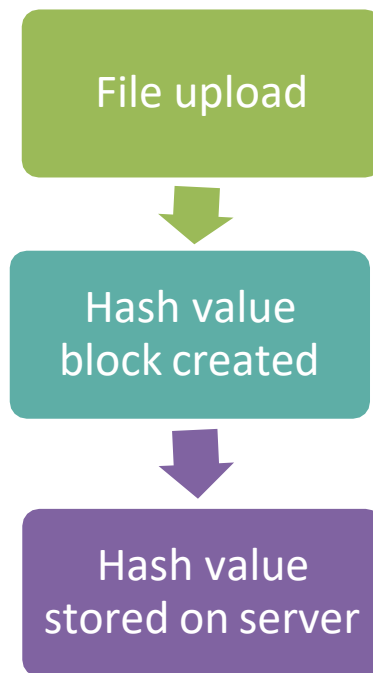


Figure 6.3 Ethereum Block verification module

4. File upload on cloud

This is an important module. In this section user uploads the files or data he wants to store in the system. The uploading is done on the cloud storage. In this case we are using local system as a cloud as it is similar and no other person can use the system except the owner.

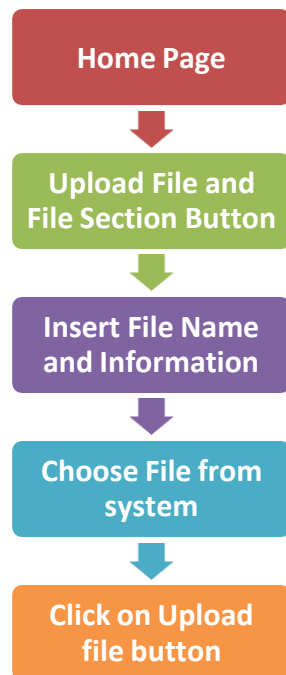


Figure 6.4: File Upload Module

5. File Encryption and Decryption Module

This module encrypts and decrypts the data that we have to provide security. We are using the RSA Algorithm for encryption purpose and the end of the module while downloading the file we again decrypt the data.

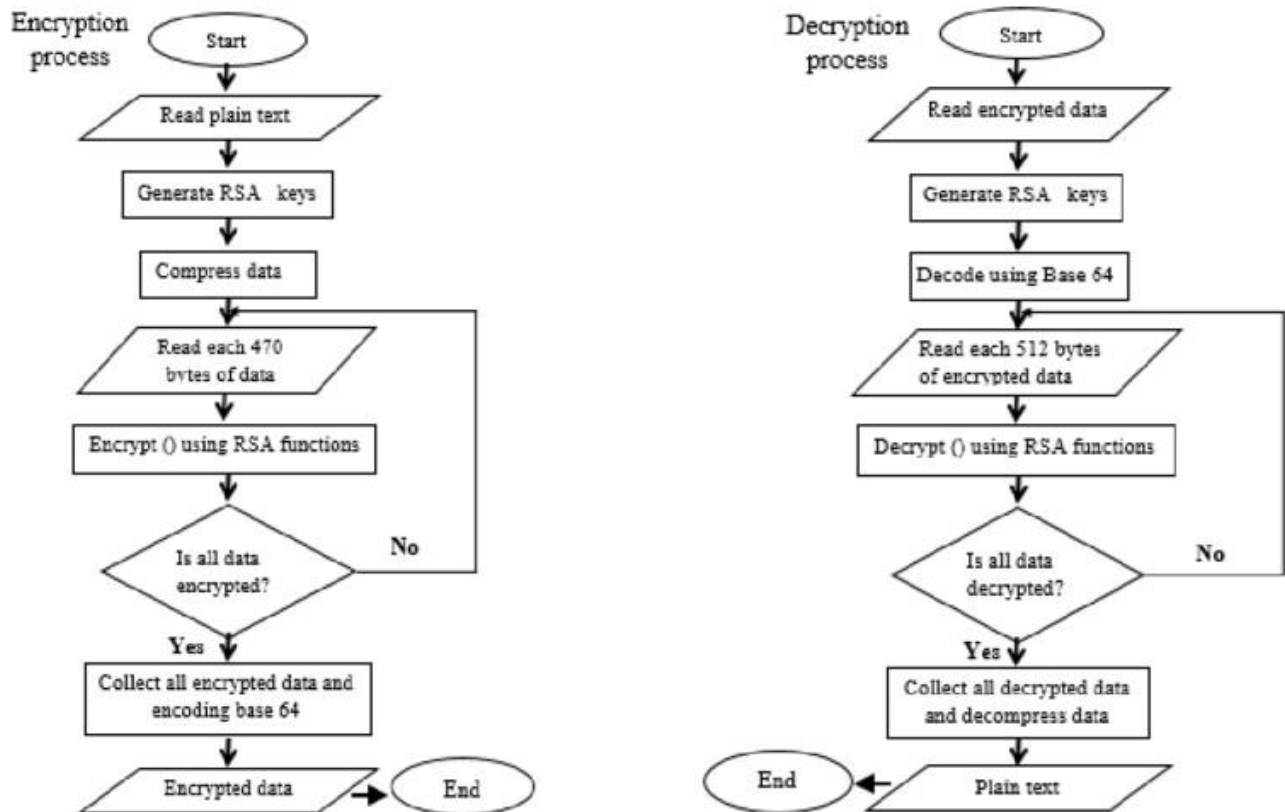


Figure 6.5: Encrypted an Decrypted Module (RSA)

6. File Cloud Storage module

After the files are encrypted using the RSA algorithm, the files are automatically uploaded to the cloud storage. The uploaded file will not be seen directly on to cloud. To get the file user has to download it from the website and then the user will decrypt the file according to his/her use and need.

TOOLS AND TECHNOLOGY USED

1 Front End Development:

Front-end development refers to the process of creating and implementing the user interface and user experience of a website or application. It involves utilizing various technologies, including HTML, CSS, and JavaScript, to build visually appealing and interactive web pages. Here's a brief explanation of each technology:

- **HTML (Hypertext Markup Language):**

HTML is the standard markup language used to structure the content of a web page. It provides a set of predefined tags and elements that define the structure and semantics of the content. HTML tags are used to represent headings, paragraphs, images, links, forms, and other elements on a web page. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by *tags*, written using angle brackets.

VERSION USED – HTML 5

- **CSS (Cascading Style Sheets):** CSS is a styling language that describes the presentation and layout of a web page. It is used to control the visual appearance of HTML elements, including colors, fonts, layouts, and animations. CSS allows developers to separate the structure (HTML) from the design (CSS), making it easier to maintain and update the styling of a website. In simpler words CSS helps to beautify our website by changing backgrounds, colors adding images etc.

VERSION USED – CSS 3

- **JavaScript:** JavaScript is a programming language that adds interactivity and dynamic behavior to web pages. It enables developers to manipulate HTML elements, handle events, perform calculations, make AJAX requests, and create interactive features such as sliders, dropdown menus, and form validation. JavaScript is essential for creating dynamic and responsive web applications. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is paradigm supporting event-driven, functional, and imperative programming styles.

- It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

2 Backend Development:

Back-end development involves creating and maintaining the server-side components of a web application. It focuses on handling data storage, processing, and business logic. Two commonly used technologies for back-end development are PHP and MySQL. Here is an overview of each:

- **PHP:** PHP (Hypertext Preprocessor) is a popular server-side scripting language specifically designed for web development. It is widely used for creating dynamic web pages and building web applications. PHP can interact with databases, handle form data, generate dynamic content, and perform various server-side operations. It has a large community and extensive documentation, making it relatively easy to learn and work with.
- **MySQL:** MySQL is a widely used open-source relational database management system (RDBMS). It is used to store, organize, and manage data in a structured manner. MySQL is compatible with various programming languages, including PHP. It provides robust features for managing databases, such as creating tables, querying data, defining relationships between tables, and ensuring data integrity. MySQL is known for its performance, scalability, and reliability.

3 Tools used:

- **VS Code:**

VS Code (Visual Studio Code) is a popular source code editor that is widely used by developers for various programming languages and frameworks. It is a lightweight and customizable editor that provides a range of features and extensions to enhance the development experience.

- **Ethereum Ganache Server:**

Ganache is a development tool in the Truffle Suite and is used for setting up a personal Ethereum Blockchain to deploy contracts, develop your applications, and run tests.

One of the popular cross-platform web servers is XAMPP, which enables programmers to construct and test their applications on a local web server. It was

created by the Apache Friends, and users can edit or change the native source code. It consists of the MariaDB database, the Apache HTTP Server, and interpreters for various programming languages like PHP and Perl. It is supported by various platforms, including the x64 package for macOS and Linux and the IA-32 package for Windows. It is available in 11 different languages.

- **XAMPP:**

The acronym XAMPP is made up of the letters X for Cross-Platform, A for Apache, M for MySQL, and Ps for PHP and Perl, respectively. It is an open-source collection of online solutions that contains the Apache server, MariaDB, PHP, and Perl modules along with command-line executables for a variety of servers.

Before releasing a website to the primary server, XAMPP enables a local host or server to test its website and clients via desktop and laptop PCs. It is a platform that offers an appropriate setting for testing and confirming the operation of projects based on Apache, Perl, MySQL, and PHP through the host's system. Among these technologies, the programming language Perl is utilized.

- **RSA Algorithm:**

Asymmetric cryptography, also known as public-key cryptography, uses a pair of keys: a public key and a private key. The public key is shared openly, while the private key is kept secret. The RSA algorithm utilizes the mathematical properties of large prime numbers to provide encryption and decryption capabilities.

Key Generation : Select two large prime numbers, p and q .

- Compute the modulus, $n = p * q$.
- Calculate Euler's totient function, $\phi(n) = (p - 1) * (q - 1)$.
- Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$. This value is the public key exponent.
- Compute the private key exponent d , which satisfies the equation $d * e \equiv 1 \pmod{\phi(n)}$.

Encryption : To encrypt a message M , the recipient uses the sender's public key (e, n) . The recipient converts the message into a numerical representation, typically using a predetermined mapping. The recipient applies the encryption formula: $C = M^e \pmod{n}$, where C is the ciphertext.

Decryption : To decrypt the ciphertext C , the recipient uses their private key d .

The recipient applies the decryption formula: $M = C^d \pmod{n}$, where M is the original message.

CHAPTER 07

SOFTWARE TESTING

TOOLS AND TECHNOLOGY USED

- **Unit Testing** - Unit testing involves testing individual units of code, such as functions or methods, in isolation. Unit testing would involve testing the different functions, smart contracts, or modules that constitute the decentralized storage system. The goal is to ensure that each unit of code performs as expected and produces the desired results.
- **Integration Testing:** Integration testing is the various components of the system work together harmoniously. In the case of decentralization using blockchain on cloud storage, integration testing would involve testing the interactions between the blockchain network, cloud storage providers, and any other third-party components or services involved.
- **Functional Testing:** Functional testing aims to test the overall functionality of the system from user's perspective. project functions as intended and meets the specified requirements. It focuses on verifying that the application performs as intended and meets the specified functional requirement. In this Testing user registration and login functionality: Ensure that users can successfully register, log in, and uploading, downloading securely through the blockchain.
- **UI testing** is conducted to ensure that the user interface of the proposed system functions correctly and provides a good user experience. This involves testing the visual appearance, layout, and interaction of the user interface elements. Testing the navigation and usability of the website: Ensure that users can easily navigate through different sections of the system website, access the account information, data related to that account and interact with the user interface components without any issues.
- **Testing responsiveness:** Verify that the user interface adapts appropriately to different screen sizes, such as desktop, mobile, and tablet devices, providing a seamless user experience.
- **Security Testing:** Security Testing involves the sensitivity of user data. It includes testing passwords, multi-factor authentication for successfully registration of user's account

TEST CASES AND RESULTS

Unit Testing Test Cases and Results:

Test Case:

Verify that the file uploaded by the user successfully secure with the help blockchain encryption.

Result: Passed

• Integration Testing Test Cases and Results:

Test Case: Verify that the Ethereum address verification with users account to execute the file upload process.

Result: Passed

• Functional Testing Test Cases and Results:

Test Case: Verify that users can register an account and login to access the account and the registration information.

Result: Passed

Test Case: Test the file upload functionality to ensure it works properly.

Result: Passed

• User Interface (UI) Testing Test Cases and Results:

Test Case: Verify that the homepage displays the file upload metadata.

Result: Passed

Test Case: Test the responsiveness of the UI on different screen sizes.

Result: Passed

• Security Testing Test Cases and Results:

Test Case: Test the encryption of file with blockchain RSA algorithm with RSA key generation process.

Result: Passed

Test Case: Verify the Two step authentication signup process with login credentials.

Results: Passed

OUTCOMES:

CHAPTER 8 RESULTS

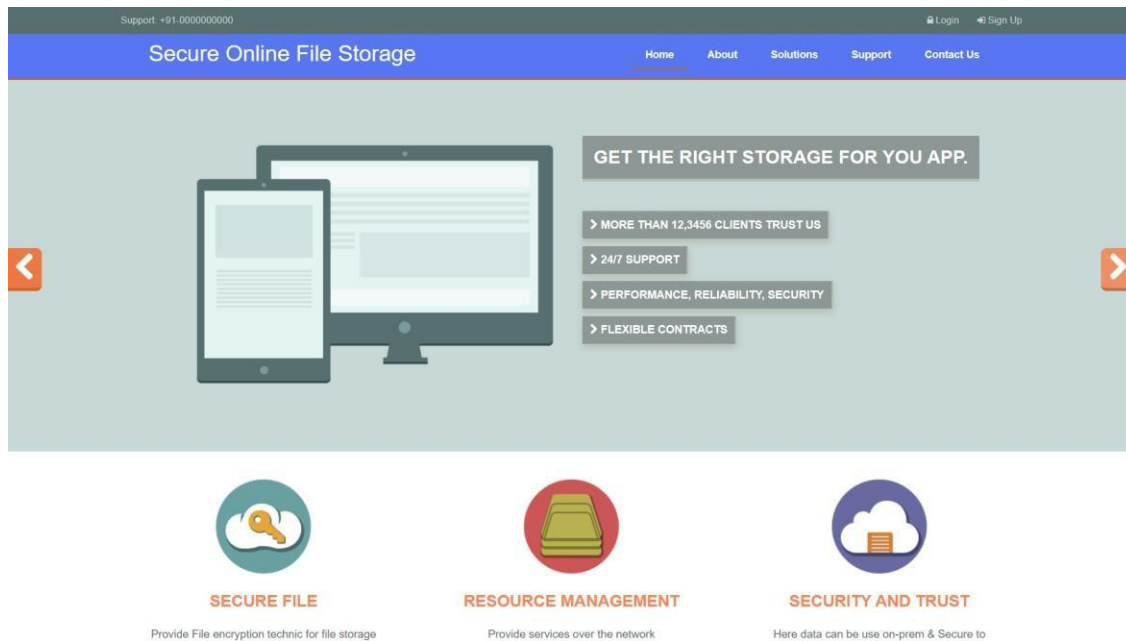


Figure 8.1 Homepage of the system

The screenshot shows the 'SIGN UP' form in the 'Secure Online File Storage' system. The form includes fields for Name (Gaurav Patkari), Email (gauravpatkari1330@gmail.com), Phone Number (7447621670), Password (0x2343A2e2739A31360309B5d83cd2243DE5d5f2bD), and a confirmation password field. A 'Sign Up' button and a '[Login]' link are provided. The footer contains the text 'ABOUT SECURE ONLINE FILE STORAGE' and 'ABOUT /'.

Figure 8.2 Sign Up Module

Support: +91-0000000000 [Login](#) [Sign Up](#)

Secure Online File Storage

[Home](#) [About](#) [Solutions](#) [Support](#) [Contact Us](#)

LOGIN

Login

[\[Join With Us\]](#)

ABOUT SECURE ONLINE FILE STORAGE

ABOUT /

Figure 8.3 Login Module

Support: +91-0000000000 [Login](#) [Sign Up](#)

Secure Online File Storage

[Home](#) [About](#) [Solutions](#) [Support](#) [Contact Us](#)

My Account

File List

Upload File

Change Password

Sign Out

My Account

My Profile

Name : adhish hulkeri

E-mail : adhishh99@gmail.com

Mobile No. : 8329306726

Join Date. : 2022-12-22 22:51:27

ABOUT SECURE ONLINE FILE STORAGE

ABOUT /

Figure 8.4 File Storage in System

Secure Online File Storage

[Home](#)
[About](#)
[Solutions](#)
[Support](#)
[Contact Us](#)

My Account

File List

Upload File

Change Password

Sign Out

Upload File

Title

Title

Info

Choose File

No file chosen

Upload File

ABOUT SECURE ONLINE FILE STORAGE

ABOUT /

Figure 8.5 File upload

Secure Online File Storage

[Home](#)
[About](#)
[Solutions](#)
[Support](#)
[Contact Us](#)

My Account

File List

Upload File

Change Password

Sign Out

Files Details

All Files

Title	File Info	Date Time	Show	
dfsfsd	sdldsf	2022-12-22 22:54:59	[Download]	
img	img	2022-12-23 00:12:44	[Download]	

ABOUT SECURE ONLINE FILE STORAGE

ABOUT /

Figure 8.6 File download list page

Ganache

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK
3

GAS PRICE
20000000000

GAS LIMIT
6721975

HARDFORK
MUIRGLACIER

NETWORK ID
5777

RPC SERVER
HTTP://0.0.0.0:7545

MINING STATUS
AUTOMINING

WORKSPACE
FILESEC

SWITCH

MNEMONIC

boring copper enforce excite amazing window lady powder easy churn tape belt

HD PATH

m/44'/60'/0'/0/account_index

ADDRESS

0xdC6Ba79507Fe79c82725A166358F1fab3eDECEf6

BALANCE

100.00 ETH

TX COUNT

0

INDEX

0

ADDRESS

0x3BF119c4c9060E43e20E56412035455Af7B4ac67

BALANCE

100.00 ETH

TX COUNT

3

INDEX

1

ADDRESS

0xD911ce33c9F62B06C5388Cf19D932abd83396b8c

BALANCE

100.00 ETH

TX COUNT

0

INDEX

2

ADDRESS

0x199A2a7872A9A6c10834620f15A04f0f7b6Fa6C0

BALANCE

100.00 ETH

TX COUNT

0

INDEX

3

ADDRESS

0x32fea858F99A905E5AFa1efC269Cb9EA2588Cdbd

BALANCE

100.00 ETH

TX COUNT

0

INDEX

4

ADDRESS

0xa3D0F52Dd8571A56680BeB3eA093d512c32efC0a

BALANCE

100.00 ETH

TX COUNT

0

INDEX

5

ADDRESS

0x005Cc4f7D911faDc0d0F0Cc18171Eb02c280AF3d

BALANCE

100.00 ETH

TX COUNT

0

INDEX

6

ADDRESS

BALANCE

TX COUNT

INDEX

Ganache									
ACCOUNTS		BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES		
CURRENT BLOCK 3	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK MUIRGLACIER	NETWORK ID 5777	RPC SERVER HTTP://0.0.0.0:7545	MINING STATUS AUTOMINING	WORKSPACE FILESEC	SWITCH	
BLOCK 3	MINED ON 2023-05-29 16:37:07		GAS USED 21000		1 TRANSACTION				
BLOCK 2	MINED ON 2023-05-20 18:17:44		GAS USED 21000		1 TRANSACTION				
BLOCK 1	MINED ON 2023-05-20 18:16:13		GAS USED 21000		1 TRANSACTION				
BLOCK 0	MINED ON 2023-05-20 17:59:58		GAS USED 0		NO TRANSACTIONS				

Figure 8.7 Database modules (Ethereum Ganache)

APPLICATIONS

- **Healthcare Data Security:** The healthcare industry deals with sensitive patient data that needs to be securely stored and shared. Implementing blockchain-based cloud storage can provide an added layer of security, ensuring the confidentiality and integrity of patients' medical records, facilitating secure data sharing between healthcare providers, and enabling patients to have control over their health data.
- **Financial Data Protection:** In the financial sector, blockchain-based cloud storage can enhance the security of sensitive financial data, such as banking records, transaction details, and personal financial information. By leveraging blockchain's immutability and encryption capabilities, financial institutions can reduce the risk of data breaches and unauthorized access, thereby protecting customer information and financial assets.
- **Legal Document Management:** Legal firms and organizations handle vast amounts of confidential and sensitive information. Blockchain-based cloud storage can be utilized to securely store legal documents, contracts, and case records, ensuring tamper-proof integrity and providing verifiable timestamps for document authenticity. This helps in maintaining the confidentiality and integrity of legal data.
- **Personal Data Protection:** With increasing concerns about data privacy, individuals are seeking more control over their personal information. Blockchain-based cloud storage can empower users to have ownership and control over their personal data, allowing them to share it securely with trusted parties and granting permissions based on their preferences. This is particularly relevant in contexts such as personal identity, social media data, and online profiles.
- **Government Data Security:** Governments deal with vast amounts of sensitive data related to citizens, public services, and national security. Blockchain-based cloud storage can enhance the security of government data, ensuring its integrity, enabling secure sharing between departments, and enhancing trust in government services.

CHAPTER 09

CONCLUSION

With increasing volumes of business data, demand for secure and inexpensive storage technology is on the up. Blockchain technology has given impetus to decentralized data storage systems that perfectly balance security, scalability, and affordability.

Since the technology is comparatively new, it may take a while before it is adopted widely and becomes mainstream. As we have seen, developers are still working towards resolving the evident shortcomings. However, the acknowledged benefits of decentralized blockchain-based storage already outweigh those of centralized storage systems – and this technology is only expected to become better in the future.

If the data-storage infrastructure of your own business needs an upgrade, it would be a good idea to get in touch with our blockchain experts so that they can work out a customized solution to fit your specific needs. Our team will look after all the technicalities, meaning that you do not need to worry any longer about data storage or security.

FUTUREWORK

1. Decentralization of the repositories and servers.
2. The IoT industry can gain largely from the integration of AI and distributed ledgers.
3. Edge Computing, it is somewhat a mixture of P to P (peer to peer), local networking, local cloud, grid, fog computing, distributed data storage and other enhanced solutions wherein some portion of the data is displaced from one or more central nodes to the end-users or the other edge local cloud computing, grid computing, fog computing, distributed data storage and other more sophisticated solutions.
4. With innovations in the IoT sector, M2M i.e., machine to machine transactions and transfers will also grow and provide a flow for the same, for example, environment or weather metrics. And important news to the public.

CHAPTER 10

REFERENCES:

1. Zyskind Guy and Oz Nathan, "privacy: Using blockchain to protect personal data" in IEEE Security and Privacy Workshops, IEEE, 2015.
1. Ruj Sushmita et al., "BlockStore: A Secure Decentralized Storage Framework on Blockchain", 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), 2018.
2. Li Dagang et al., "A Secure Data-Sharing Protocol Under Blockchain-Based Decentralized Storage Architecture", IEEE Networking Letters, vol. 1.1, pp. 30-33, 2019.
3. D. Sivaganesan, "BLOCK CHAIN ENABLED INTERNET OF THINGS", Journal of Information Technology, vol. 1.01, pp. 1-8, 2019.
4. Atieh, A.T. The Next Generation Cloud technologies: A Review On Distributed Cloud, Fog And Edge Computing and Their Opportunities and Challenges. Res. Rev. Sci. Technol. 2021, 1, 1–15. Available online: <https://researchberg.com/> (accessed on 5 March 2022).
5. Bacis, E.; Vimercati, S.D.C.D.; Foresti, S.; Paraboschi, S.; Rosa, M.; Samarati, P. Securing Resources in Decentralized Cloud Storage. IEEE Trans. Inf. Forensics Secur. 2020, 15, 286–298.
6. Baalamurugan, K.M.; Kumar, S.R.; Kumar, A.; Kumar, V.; Padmanaban, S. Padmanaban, Blockchain Security in Cloud Computing; Springer: Cham, Switzerland, 2022.
7. Kumar, A.; Abhishek, K.; Nerurkar, P.; Ghalib, M.R.; Shankar, A.; Cheng, X. Secure smart contracts for cloud-based manufacturing using Ethereum blockchain. Trans. Emerg. Telecommun. Technol. 2020, 33, e4129.
8. Taha, A.; Zakaria, A.; Kim, D.; Suri, N. Decentralized Runtime Monitoring Approach Relying on the Ethereum Blockchain Infrastructure. In Proceedings of the 2020 IEEE International Conference on Cloud Engineering (IC2E), Sydney, NSW, Australia, 21–24 April 2020; pp. 134–143.
9. Awadallah, R.; Samsudin, A.; Teh, J.S.; Almazrooie, M. An Integrated Architecture for Maintaining Security in Cloud Computing Based on Blockchain. IEEE Access 2021, 9, 69513–69526.
10. Dannen, C. Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners; Apress: Berkeley, CA, USA, 2017; Volume 318.
14. Goldin, P. State of the Cloud Report: DevOps Trends; Ferreira, D.M.P., Ed.; RightScale Inc.: Santa Barbara, CA, USA, 2021; pp. 1–19.

15. Monti, M.; Rasmussen, S. RAIN: A Bio-Inspired Communication and Data Storage Infrastructure. *Artif. Life* 2017, 23, 552–557.
16. Zhu, Y.; Lv, C.; Zeng, Z.; Wang, J.; Pei, B. Blockchain-based Decentralized Storage Scheme. *J. Phys. Conf. Ser.* 2019, 1237, 042008.
17. Sarmah, S.S. Application of Block chain in Cloud Computing. *Int. J. Innov. Technol. Explore Eng.* 2019, 8, 4698–4704.
18. Nedakovic, A. Analysis and Improvements of VerifyMed-The Blockchain Solution for Virtualized Healthcare Trust Relations.Security and Cloud Computing (SECCLO). Master's Thesis, Aalto University, Espoo, Finland, 2022.
19. Nizamuddin, N.; Salah, K.; Azad, M.A.; Arshad, J.; Rehman, M. Decentralized document version control using ethereum blockchain and IPFS. *Comput. Electr. Eng.* 2019, 76, 183–1