A

Project Report

on

CLOUD BROKERAGE ARCHITECTURE FOR EFFICIENT CLOUD SERVICE SELECTION

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTAPURAMU

in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

B BALAJI	(209E1A0515)
G UMESH CHANDRA	(209E1A0561)
S GNANADEEP	(209E1A0565)
G KRISHNA SAI	(219E5A0504)

Under the Guidance of

Dr K SANTHI

Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESWARA ENGINEERING COLLEGE

(Approved by AICTE, New Delhi, NBA & NAAC Accredited Institution with UGC section 2(f) & 12(b) &Affiliated to JNTUA, Anantapuramu)

Karakambadi Road, TIRUPATI – 517507

2020-2024

SRI VENKATESWARA ENGINEERING COLLEGE

(Approved by AICTE, New Delhi, NBA & NAAC Accredited Institution with UGC section 2(f) & 12(b) &Affiliated to JNTUA, Anantapuramu)

Karakambadi Road, TIRUPATI – 517507

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "A CLOUD BROKERAGE ARCHITECTURE FOR EFFICIENT CLOUD SERVICE SELECTION" a bonafide record of the project work done and submitted by

B BALAJI	**(209E1A0515)**
G UMESH CHANDRA	**(209E1A0561)**
S GNANADEEP	**(209E1A0565)**
G KRISHNA SAI	**(219E5A0504)**

for the partial fulfillment of the requirements for the award of B.Tech Degree in **COMPUTER SCIENCE AND ENGINEERING**, *JNT University Anantapur, Ananthapuramu.*

GUIDE **Head of the Department**

External Viva-Voce Exam Held on_____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We hereby declare that the project report entitled **"A CLOUD BROKERAGE ARCHITECTURE FOR EFFICIENT CLOUD SERVICE SELECTION"** done by us under the guidance of **Dr K SANTHI,** and is submitted in partial fulfillment of the requirements for the award of the Bachelor's degree in **Computer Science and Engineering.** This project is the result of our own effort and it has not been submitted to any other University or Institution for the award of any degree or diploma other than specified above.

B BALAJI	**(209E1A0515)**
G UMESH CHANDRA	**(209E1A0561)**
S GNANADEEP	**(209E1A0565)**
G KRISHNA SAI	**(219E5A0504)**

ACKNOWLEDGEMENT

We are immensely grateful to our guide and Head of the Department, **Dr. K Santhi**, for her invaluable guidance, encouragement, and kind support throughout our project work. Her helping attitude, suggestions, and encouragement have been instrumental in our successful completion of the project. We extend our heartfelt thanks to **Dr. K Santhi** for her unwavering assistance and motivation.

We have great pleasure in expressing our hearty thanks to our beloved Principal **Dr. C Chandrasekhar,** for spending his valuable time with us to complete this project. Successful completion of any project cannot be done without proper support and encouragement. We sincerely thank to the **Management** for providing all the necessary facilities during the course of study.

We would like to thank our parents and friends, who have the greatest contributions in all our achievements, for the great care and blessings in making us successful in all our endeavors.

B BALAJI **(209E1A0515)**
G UMESH CHANDRA **(209E1A0561)**
S GNANADEEP **(209E1A0565)**
G KRISHNA SAI **(219E5A0504)**

TABLE OF CONTENTS

ABSTRACT

The rapid expansion of cloud computing services presents consumers with abundant options but also introduces the challenge of selecting the optimal service amidst the vast array of choices. To address this, our paper proposes a groundbreaking brokerage-based architecture in the cloud. In this model, cloud brokers take on the responsibility of service selection. We introduce an efficient information management structure, the Bcloud-tree, designed for handling data from numerous cloud service providers. Our developed service selection algorithm recommends the most suitable cloud services to consumers. Through extensive real and synthetic cloud data experiments, we showcase a notable performance improvement compared to existing methods. Key terms include cloud computing, cloud brokerage, service selection, Bcloud-tree, indexing, and querying.

B BALAJI (209E1A0515)
G UMESH CHANDRA (209E1A0561)
S GNANADEEP (209E1A0565)
G KRISHNA SAI (219E5A0504)

Table of Figures

1 INTRODUCTION

Objective of the Project:

The project's objective is to streamline cloud service selection by introducing a novel brokerage- based architecture. This involves assigning service selection responsibilities to cloud brokers. Additionally, the project aims to design and implement the efficient Bcloud-tree indexing structure. The overarching goal is to provide consumers with a more straightforward and effective means of navigating the extensive range of cloud services by developing a service selection algorithm. The project's success is measured through extensive experiments demonstrating significant performance improvements over existing approaches.

Problem Statement:

The burgeoning cloud computing landscape presents a challenge for consumers in selecting optimal services due to the overwhelming variety. Collecting and analyzing information on numerous providers is time-consuming. This project addresses this issue by proposing a novel brokerage-based cloud architecture, utilizing the Bcloud-tree indexing structure. The objective is to streamline service selection, empowering cloud brokers to make informed choices on behalf of consumers. The problem lies in the current inefficiencies of navigating the vast array of cloud services, prompting the need for a more efficient and automated selection process.

Scope:

This project aims to enhance the efficiency of cloud service selection through a novel brokerage- based architecture and the implementation of the Bcloud-tree indexing structure. By delegating service selection responsibilities to cloud brokers, we address the challenge of consumers navigating a vast pool of options. The project's scope includes designing and implementing the Bcloud-tree, developing a service selection algorithm, and conducting comprehensive experiments to demonstrate significant performance improvements in recommending suitable cloud services to consumers.

Introduction:

Cloud services have revolutionized the landscape of modern businesses, offering unparalleled

flexibility and scalability in terms of storage space and computing capabilities. This paradigm shift is particularly significant for small and medium-sized enterprises, where efficient utilization of resources can directly impact competitiveness and growth. However, navigating the vast array of cloud service options poses a considerable challenge for businesses. To address this challenge, cloud brokerage mechanisms have emerged as essential facilitators in the cloud ecosystem, aiding users in service discovery, mediation, and monitoring.

Despite the proliferation of cloud brokerage mechanisms, the process of selecting the most suitable cloud services remains complex, especially given the ever-expanding pool of providers and service types. Traditional approaches have focused on defining selection criteria such as Quality of Service (QoS) and pricing, as well as determining their relative importance to users. However, with the rapid growth of cloud adoption and the multitude of emerging providers and services, a critical need arises for more efficient selection mechanisms.

In response to this need, we propose a novel cloud brokerage architecture aimed at streamlining the service selection process. Central to our approach is the development of a robust indexing structure, the Bcloud-tree, coupled with a powerful query engine. This architecture enables the efficient classification, storage, and retrieval of diverse cloud services, significantly reducing the time and effort required for selection.

2 LITERATURE SURVEY

[1] M. B. and T. F. Decision-making in cloud computing environments: A cost and risk based approach. Information Systems Frontiers, 14(4):87193, 2012.

The introduction sets the stage for the research by providing background information on cloud computing environments. It likely outlines the significance of decision-making in these environments, highlighting the complexities and challenges associated with managing costs and risks. The introduction may also introduce the main objectives or research questions of the study.

The existing section typically reviews related literature and previous research relevant to decision-making in cloud computing environments. It may discuss existing methodologies, models, or frameworks used for decision-making in this context. This section often identifies gaps or limitations in the existing literature that the current study aims to address.

The proposed section outlines the methodology, framework, or approach proposed by the authors to address the identified gaps or limitations. It may detail the specific factors considered in the cost and risk-based approach proposed by the authors. This section often describes how the proposed approach differs from or improves upon existing methods.

The summary section provides a concise overview of the key findings and contributions of the study. It may recap the main points discussed in the introduction, existing, and proposed sections. The summary typically concludes with implications for practice or future research directions.

[2] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6):599–616, 2009.

The introduction of the paper "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility" by R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, published in Future Generation Computer Systems in 2009, sets the stage by introducing the concept of cloud computing as a transformative paradigm in the field of information technology. The authors highlight the potential of cloud computing to revolutionize the way computing resources are accessed, delivered, and managed, drawing parallels to utilities like electricity and water. They acknowledge the growing interest and excitement surrounding cloud

computing but also emphasize the need to distinguish between hype and reality in order to fully understand its implications and challenges.

The existing section of the paper reviews the evolution of computing paradigms leading up to the emergence of cloud computing. It discusses the limitations of traditional IT models, such as the client-server architecture, and highlights the driving forces behind the shift towards cloud computing, including advances in virtualization, networking technologies, and the increasing demand for scalable and flexible computing resources. The authors also examine the key characteristics and components of cloud computing, such as on-demand self-service, resource pooling, and pay-as-you-go pricing models, while discussing the various types of cloud deployments (e.g., public, private, hybrid) and service models (e.g., SaaS, PaaS, IaaS).

In the proposed section, the authors outline their vision for the future of cloud computing and emerging IT platforms. They discuss potential applications and use cases spanning various domains, including business, academia, government, and healthcare, envisioning a future where computing resources are seamlessly provisioned and accessed over the internet on an as-needed basis. The authors also identify several challenges and research directions that need to be addressed to realize this vision, such as scalability, security, interoperability, and energy efficiency.

Finally, the summary section provides a recap of the key points discussed in the paper and reiterates the authors' call for a balanced perspective on cloud computing, acknowledging both its transformative potential and the practical challenges that must be overcome to fully realize its benefits. The authors emphasize the importance of ongoing research and collaboration across academia, industry, and government to address these challenges and drive the adoption of cloud computing as the fifth utility for delivering computing resources.

[3] C. W. Chang, P. Liu, and J. J. Wu. Probability-based cloud storage providers selection algorithms with maximum availability. In 41st International Conference on

The introduction of the paper likely sets the stage by discussing the growing importance of cloud storage and the challenges associated with selecting reliable cloud storage providers. It may touch upon the increasing demand for cloud services, the diversity of cloud providers available, and the need for effective algorithms to ensure maximum availability of data stored in the cloud. The introduction would likely provide context for the problem the authors aim to address and may briefly mention existing approaches and their limitations.

In this section, the paper likely reviews existing methodologies or algorithms used for selecting cloud storage providers. It may discuss traditional methods, such as deterministic selection based on factors like cost or geographical location, as well as more advanced techniques like load balancing or fault tolerance strategies. The authors would likely highlight the shortcomings of these existing approaches, such as their inability to adapt to dynamic environments or ensure optimal availability of data.

This section presents the novel probability-based algorithms proposed by C. W. Chang, P. Liu, and J. J. Wu. The authors likely detail the underlying principles of their approach, which may involve probabilistic modeling to assess the reliability and availability of different cloud storage providers. They would likely explain how their algorithms aim to maximize data availability by dynamically selecting the most suitable provider based on probabilistic considerations. The proposed algorithms may incorporate factors such as provider reliability, network latency, and data replication strategies to ensure high availability of data stored in the cloud.

The summary section would likely provide a recap of the key contributions of the paper. It may highlight the main findings and insights gained from the proposed probability-based selection algorithms. The authors may discuss the potential benefits of their approach, such as improved data availability, reduced downtime, and enhanced resilience to failures in cloud storage environments. Additionally, they may outline directions for future research or practical implications for deploying their algorithms in real-world cloud computing scenarios.

[4] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione. Smart cloud storage service selection based on fuzzy logic, theory of evidence and game theory. IEEE Transactions on Computers, PP(99):1–1, 2015.

The introduction of the paper likely sets the stage by discussing the increasing importance of cloud storage services and the challenges associated with selecting the most suitable service provider. It may also highlight the proliferation of cloud storage options, each with varying features, costs, and performance levels. The introduction would likely discuss the significance of efficient selection methods to help users make informed decisions when choosing cloud storage services. Additionally, it may briefly introduce the three main methodologies utilized in the proposed approach: fuzzy logic, theory of evidence, and game theory.

The section discussing existing methods would likely review previous research and approaches related to cloud service selection. This might include traditional decision-making techniques, heuristic methods, and possibly early attempts to incorporate fuzzy logic or game theory into the selection

process. The authors may discuss limitations or gaps in existing methodologies that their proposed approach seeks to address.

The proposed methodology section would delve into the details of how fuzzy logic, theory of evidence, and game theory are integrated to create a novel approach for smart cloud storage service selection. It would explain the rationale behind using each of these methodologies and how they complement each other to enhance decision-making in cloud service selection. The section would likely include mathematical models, algorithms, and explanations of how these models are applied in practice to evaluate and rank cloud storage service providers.

Summary: In the summary section, the authors would recap the main contributions of their work, emphasizing the novel aspects of their approach and how it advances the state-of-the-art in cloud service selection. They may highlight key findings, results, or insights gained from applying the proposed methodology. Additionally, the authors might discuss potential applications of their approach in real-world scenarios and suggest avenues for future research and development in this area. The summary would provide a concise overview of the paper's contributions and implications.

3 SYSTEM ANALYSIS

3.1 EXISTING METHOD:

The existing system for cloud service selection lacks efficiency and faces challenges in managing the vast array of dynamic cloud service information. The absence of a dedicated brokerage architecture results in time-consuming manual efforts from consumers to collect and analyse data from numerous providers. There is no specialized indexing structure, leading to difficulties in organizing and updating service information. Queries from consumers are not effectively grouped, impacting the overall responsiveness. Additionally, the current system lacks a comprehensive algorithm for handling diverse service selection queries. Performance evaluations reveal limitations when compared to the proposed Bcloud-tree-based approach, emphasizing the need for improvement in the existing system.

3.1.1 DISADVANTAGES:

The disadvantages of the existing system include:

1. Manual and Time-Consuming Selection: Consumers face the burden of manually collecting and analysing information from multiple cloud service providers, leading to a time-consuming selection process.

2. Lack of Specialized Architecture: The absence of a dedicated brokerage-based architecture results in inefficiencies in service selection and management.

3. Inefficient Data Organization: Without a specialized indexing structure, there are challenges in efficiently organizing and updating the large volume of dynamic cloud service information.

4. Poor Query Responsiveness: The existing system struggles with effectively grouping and responding to consumers' queries, impacting the overall responsiveness of the service selection process.

5. Limited Query Handling: The system lacks a comprehensive algorithm for handling diverse service selection queries, hindering its ability to accommodate various user requirements.

6. Performance Limitations: Comparative evaluations indicate performance limitations when compared to the proposed Bcloud-tree-based approach, highlighting the need for enhancements in the existing system.

3.2 PROPOSED METHOD:

We propose a streamlined cloud service selection process through generic brokerage architecture for this we are using the AES algorithm for the security and privacy to protect the data. The cloud broker receives consumers' requirements, groups similar queries, and utilizes a powerful query engine to search the Bcloud-tree for matching services. Our contributions include the Bcloud-tree, proficient in storing and indexing dynamic cloud service data, an effective query algorithm for diverse service selection queries, and demonstrated performance improvements over existing methods through real scenario-based evaluations of cloud providers.

3.2.1 ADVENTAGES:

The proposed cloud service selection approach offers several advantages over the existing system:

1. Efficiency: The generic cloud brokerage architecture streamlines the selection process, reducing the time and effort required by consumers to find suitable services.

2. Organization: The Bcloud-tree indexing structure efficiently stores and manages a large volume of dynamic cloud service information, providing a structured and organized system for data.

3. Query Flexibility: The query algorithm supports a wide range of service selection queries, allowing users to specify diverse requirements, such as intervals of desired values for multiple properties.

4. User-Friendly: By grouping similar queries and employing a powerful query engine, the system enhances user experience, making it more intuitive and responsive.

5. Real-World Validation: The approach is validated through studies involving real cloud providers, demonstrating its effectiveness and performance improvement over the existing method.

6. Automation: The cloud broker takes on the responsibility of service selection, automating the process for consumers and relieving them from the burden of manually analyzing and comparing services.

3.3 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.3.1 ECONOMICAL FEASIBILITY

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

3.3.2 TECHNICAL FEASIBILITY

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

3.3.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with

it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4 SYSTEM DESIGN

4.1 INTRODUCTION

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture, and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development, then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

4.2 UML DIAGRAMS

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

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

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

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

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

GOALS:

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

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

4.3 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modelling Language (UML) 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 the classes. It explains which class contains information.

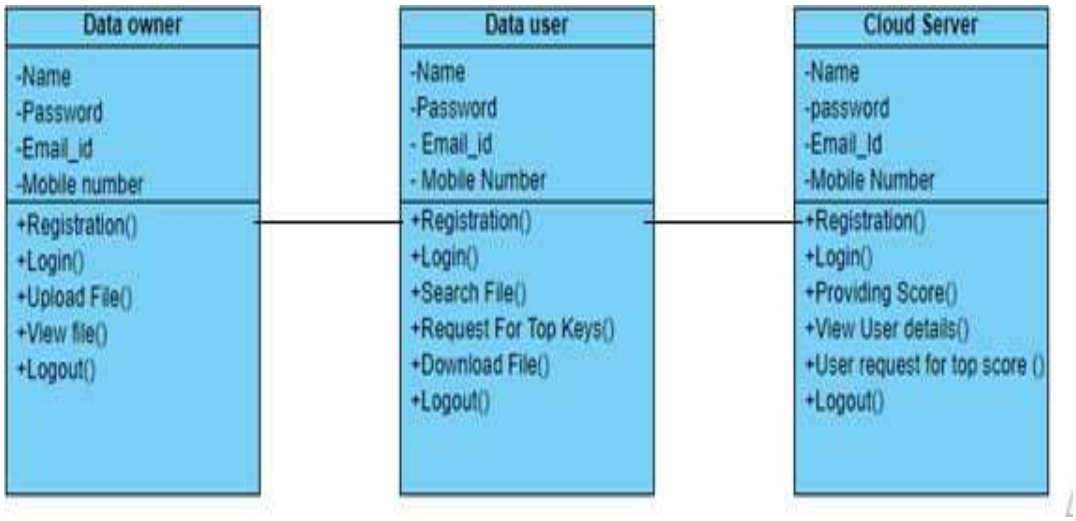


Fig 4-1 CLASS DIAGRAM

4.4 USE CASE DIAGRAM:

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

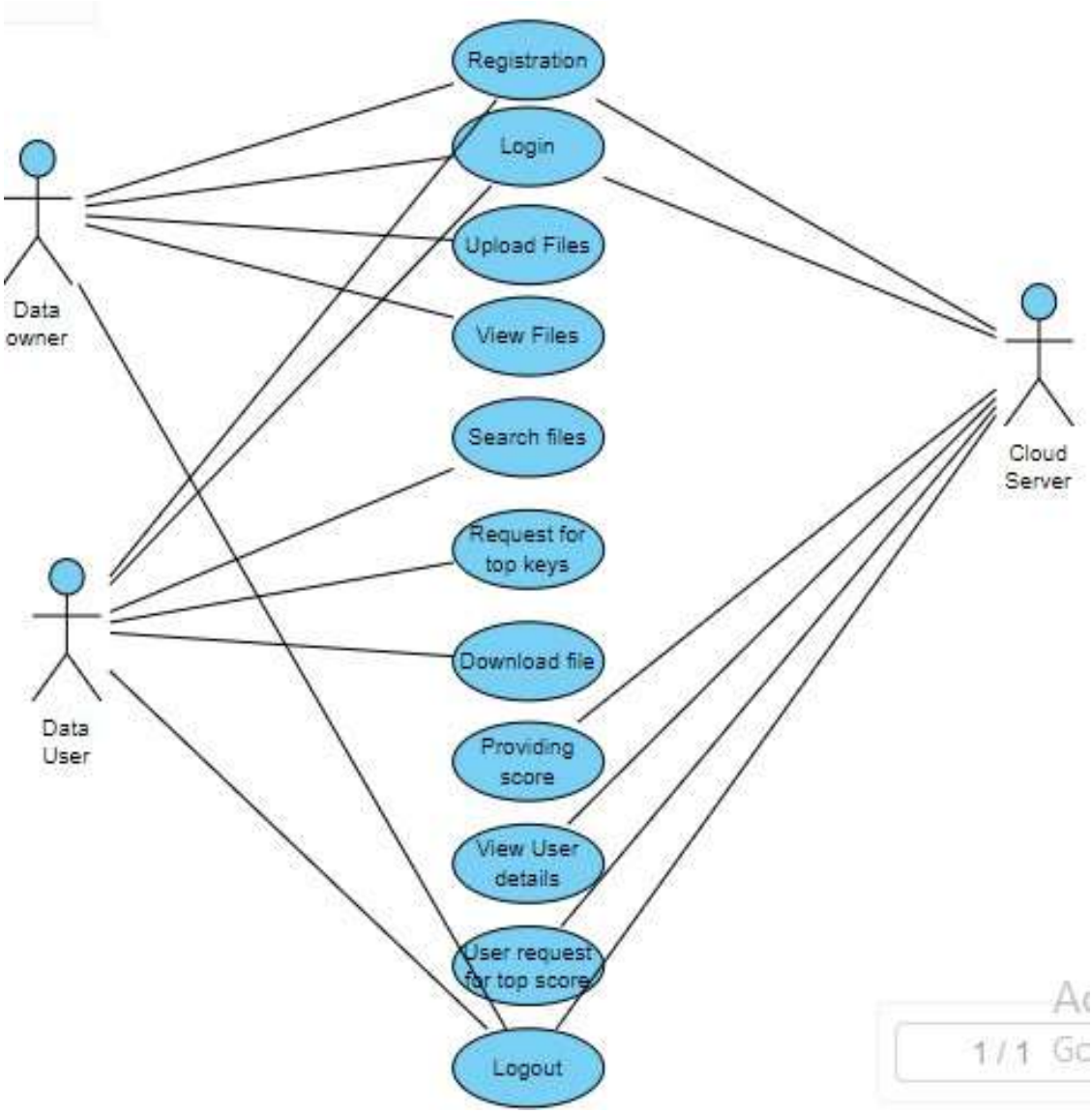


Fig 4-2USE CASE DIAGRAM

4.5 SEQUENCE DIAGRAM:

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

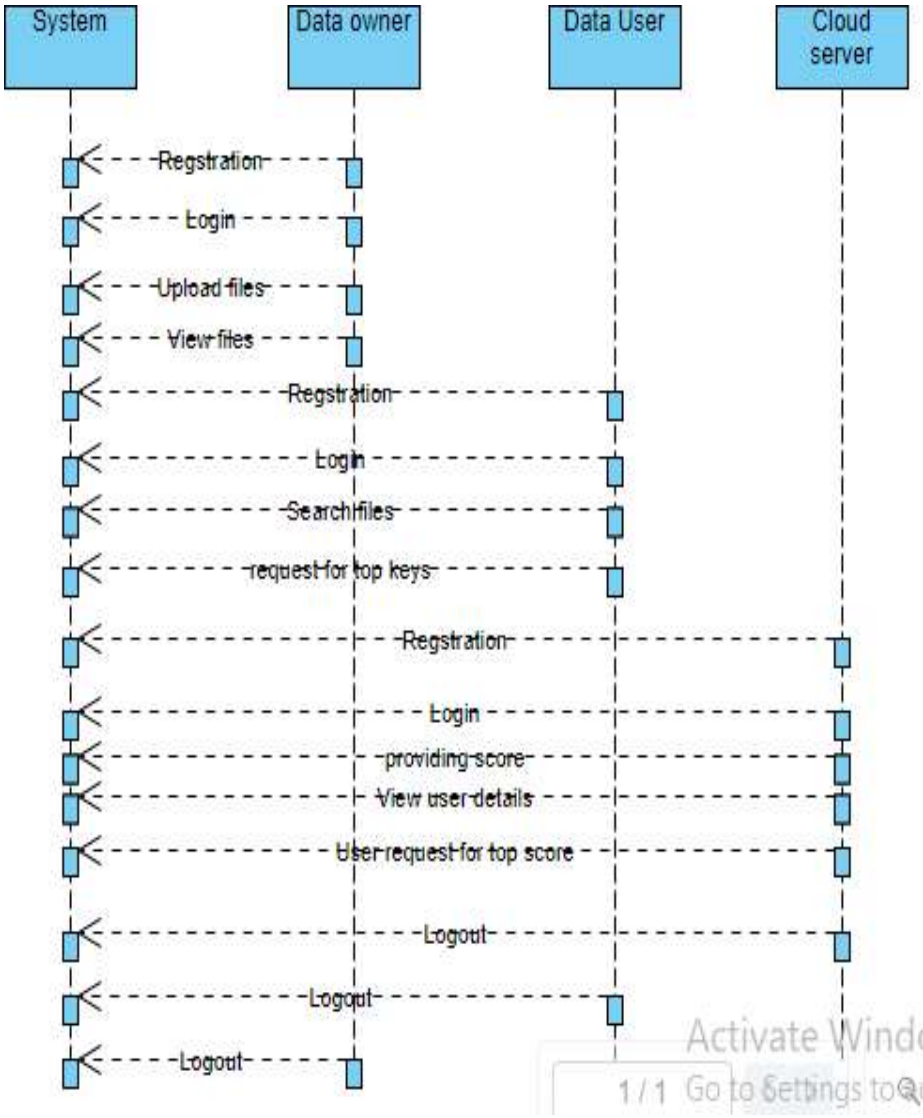


Fig 4-3SEQUENCE DIAGRAM

4.6 COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

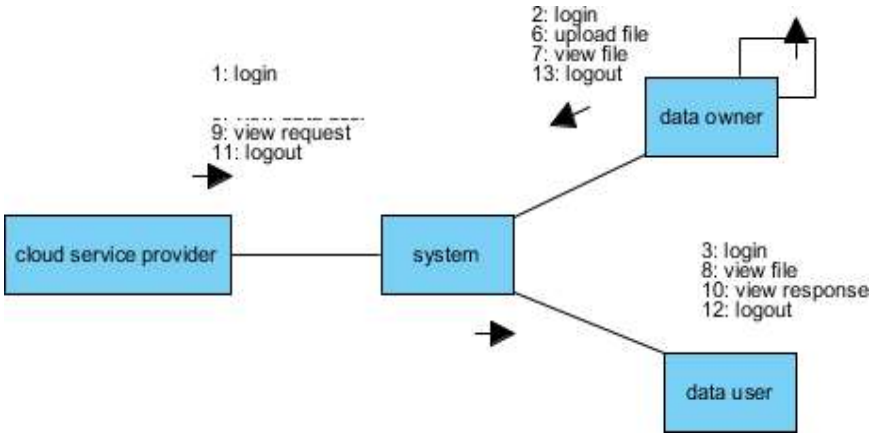


Fig 4-4COLLABORATION DIAGRAM

4.7 DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

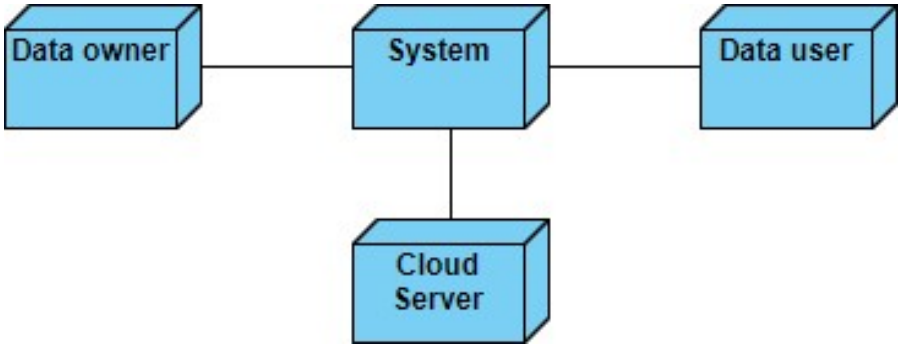


Fig 4-5DEPLOYMENT DIAGRAM

4.8 ACTIVITY DIAGRAM:

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

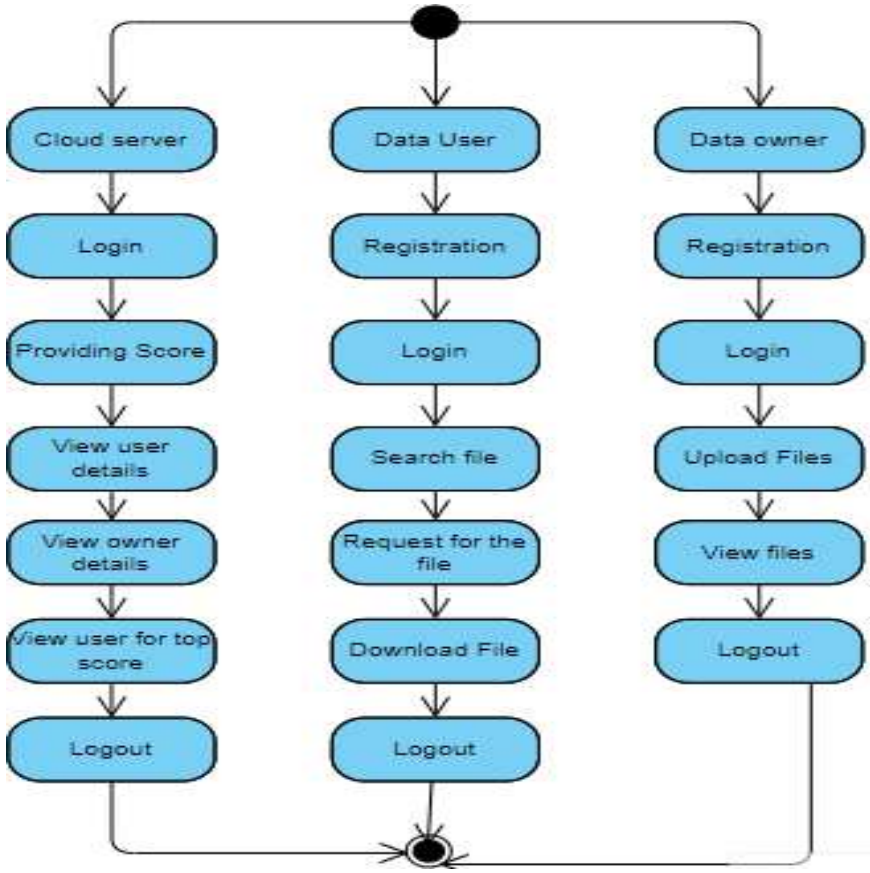


Fig 4-6ACTIVITY DIAGRAM

4.9 COMPONENT DIAGRAM:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

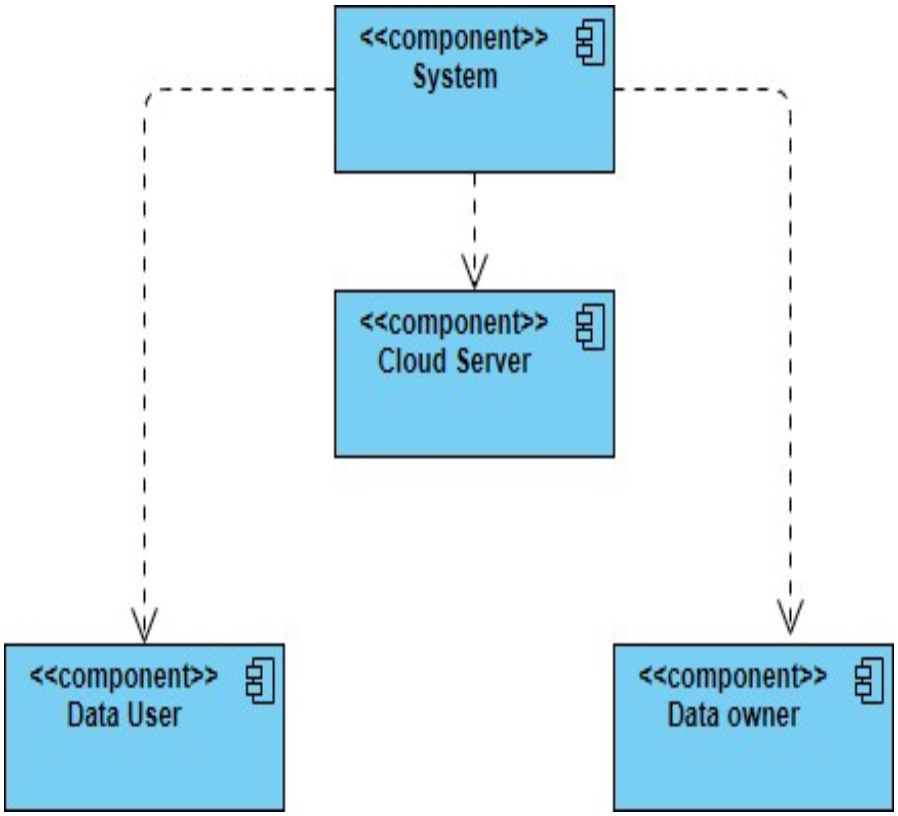


Fig 4-7COMPONENT DIAGRAM

4.10 ER DIAGRAM:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

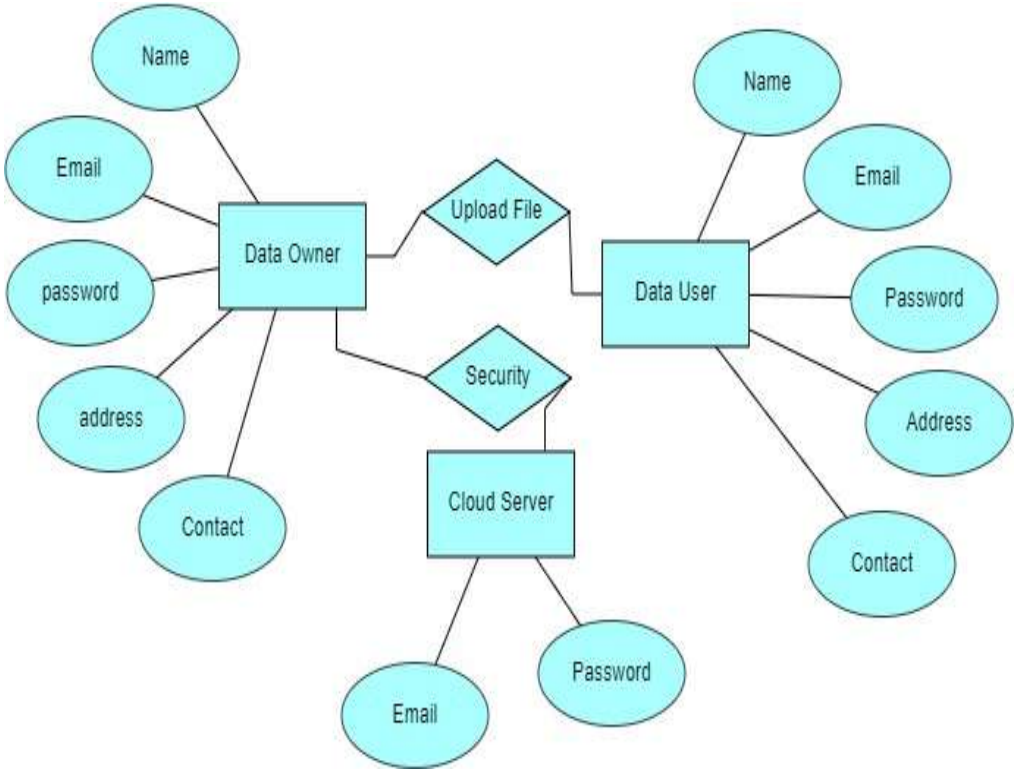


Fig 4-8ER DIAGRAM

4.11 DFD DIAGRAM:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

Level 1 Diagram:

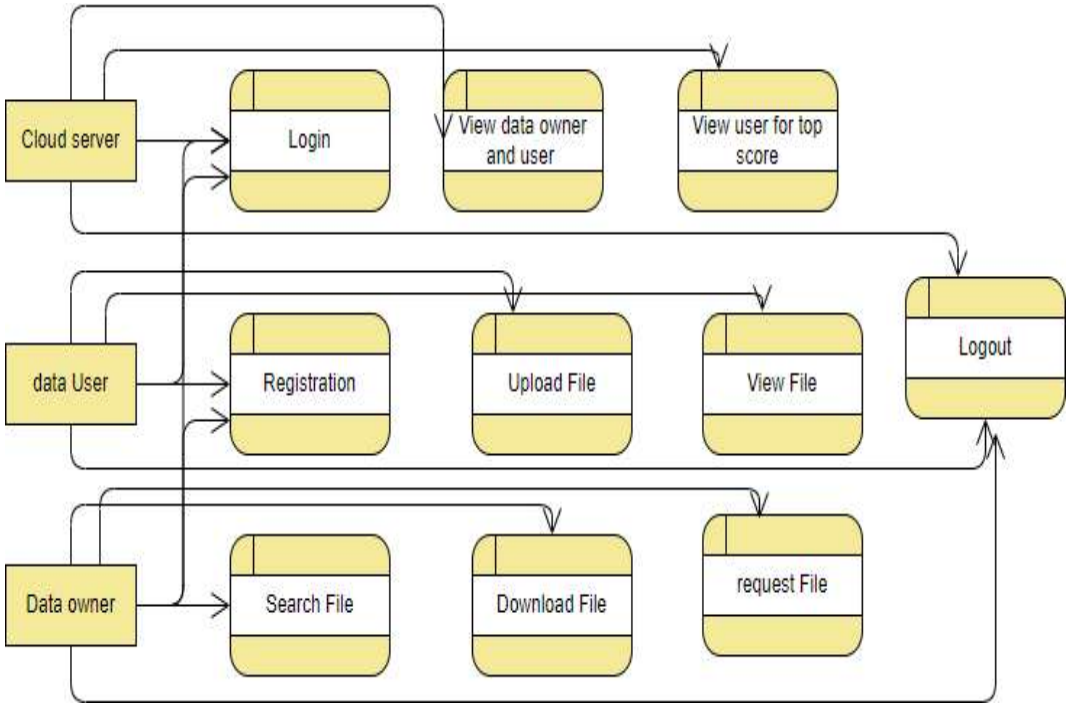


Fig 4-9DFD DIAGRAM

5 SYSTEM IMPLEMENTATION

5.1 HARDWARE & SOFTWARE REQUIREMENTS:

5.1.1 H/W CONFIGURATION:

- Hard Disk -160GB
- Processor - I3/Intel Processor
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA
- RAM - 4Gb

5.1.2 S/W CONFIGURATION:

- Operating System : Windows 7/8/10
- Server side Script : Python, HTML, MYSQL, CSS, Bootstrap.
- Libraries : PANDAS, Django, Smtlib
- IDE : PyCharm
- Technology : Python 3.6+

5.2 ARCHITECTURE

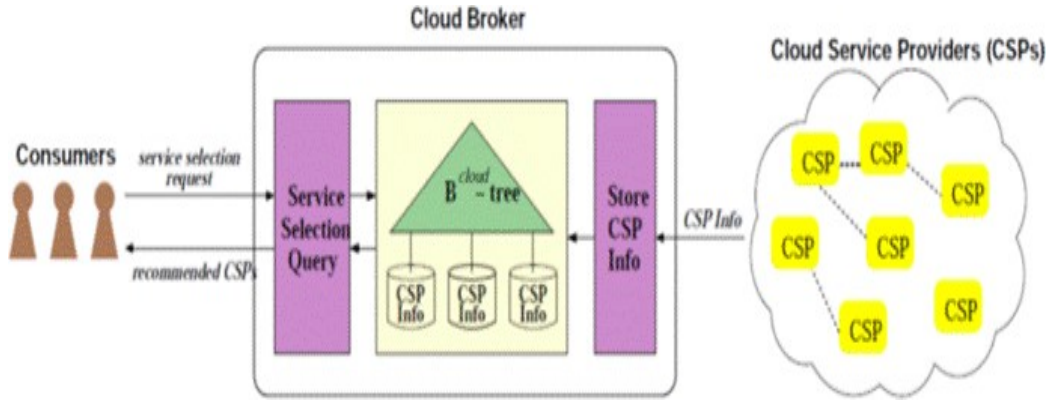


Fig 5-1 AES Architecture

ADVANCED ENCRYPTION STANDARD

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Working of the cipher :

AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

The number of rounds depends on the key length as follows :

128 bit key – 10 rounds

192 bit key – 12 rounds

256 bit key – 14 rounds

Creation of Round keys :

A Key Schedule algorithm is used to calculate all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.

Encryption :

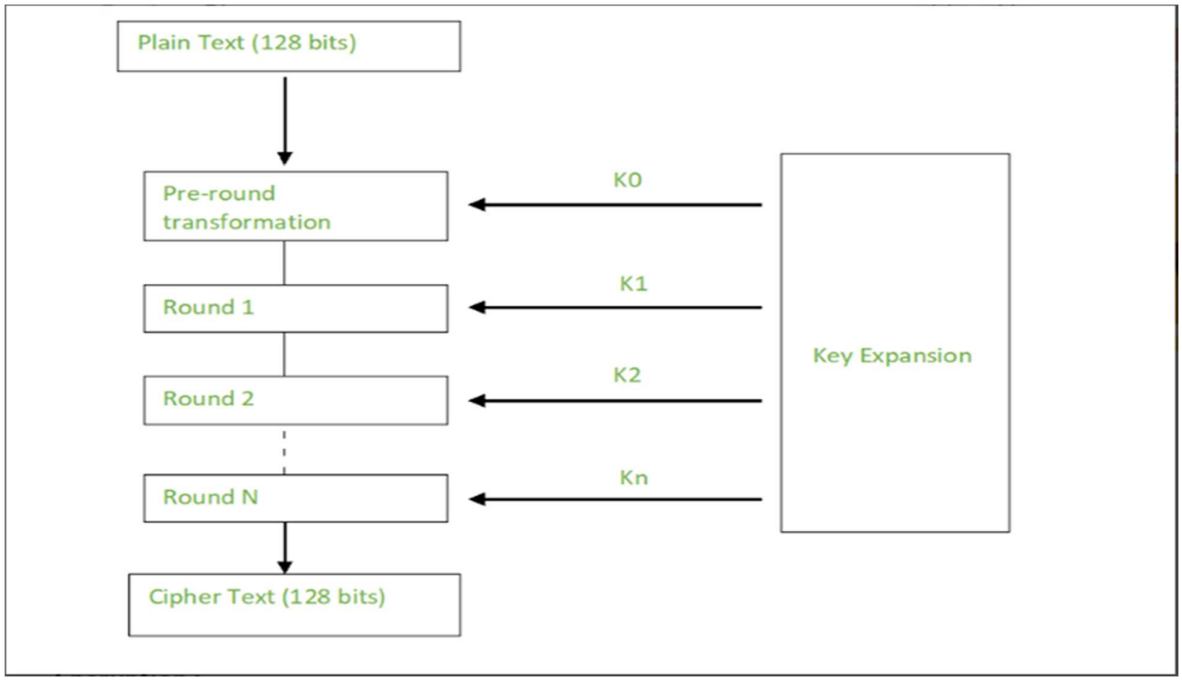
AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

[b0 | b4 | b8 | b12 |

| b1 | b5 | b9 | b13 |

| b2 | b6 | b10| b14 |

| b3 | b7 | b11| b15]



Each round comprises of 4 steps :

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes :

This step implements the substitution.

In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

ShiftRows :

This step is just as it sounds. Each row is shifted a particular number of times.

The first row is not shifted

The second row is shifted once to the left.

The third row is shifted twice to the left.

The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

[b0 | b1 | b2 | b3] [b0 | b1 | b2 | b3]

| b4 | b5 | b6 | b7 | -> | b5 | b6 | b7 | b4 |

| b8 | b9 | b10 | b11 | | b10 | b11 | b8 | b9 |

[b12 | b13 | b14 | b15] [b15 | b12 | b13 | b14]

MixColumns :

This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

[c0] [2 3 1 1] [b0]

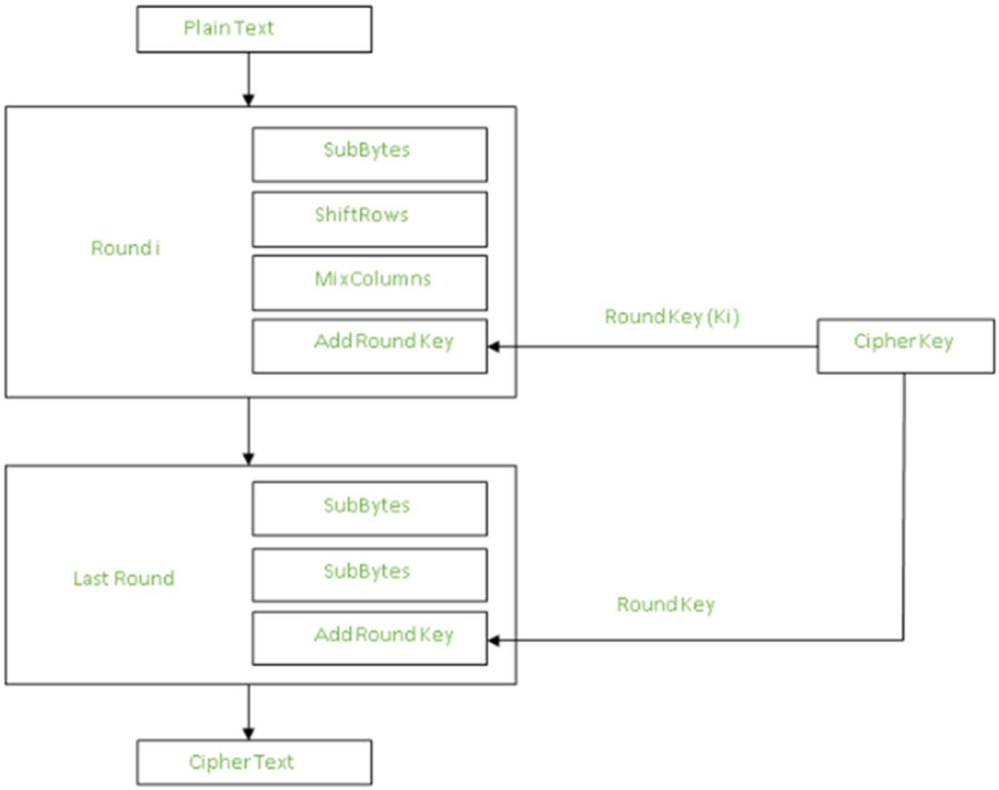
| c1 | | 1 2 3 1 | | b1 |

| c2 | | 1 1 2 3 | | b2 |

[c3] [3 1 1 2] [b3]

Add Round Keys :

Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.



After all these rounds 128 bits of encrypted data is given back as output. This process is repeated until all the data to be encrypted undergoes this process.

Decryption :

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes.Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

Inverse MixColumns :

This step is similar to the MixColumns step in encryption, but differs in the matrix used to carry out the operation.

$$\begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \end{bmatrix}$$

Inverse SubBytes :

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

5.3 BLOCK DIAGRAM:

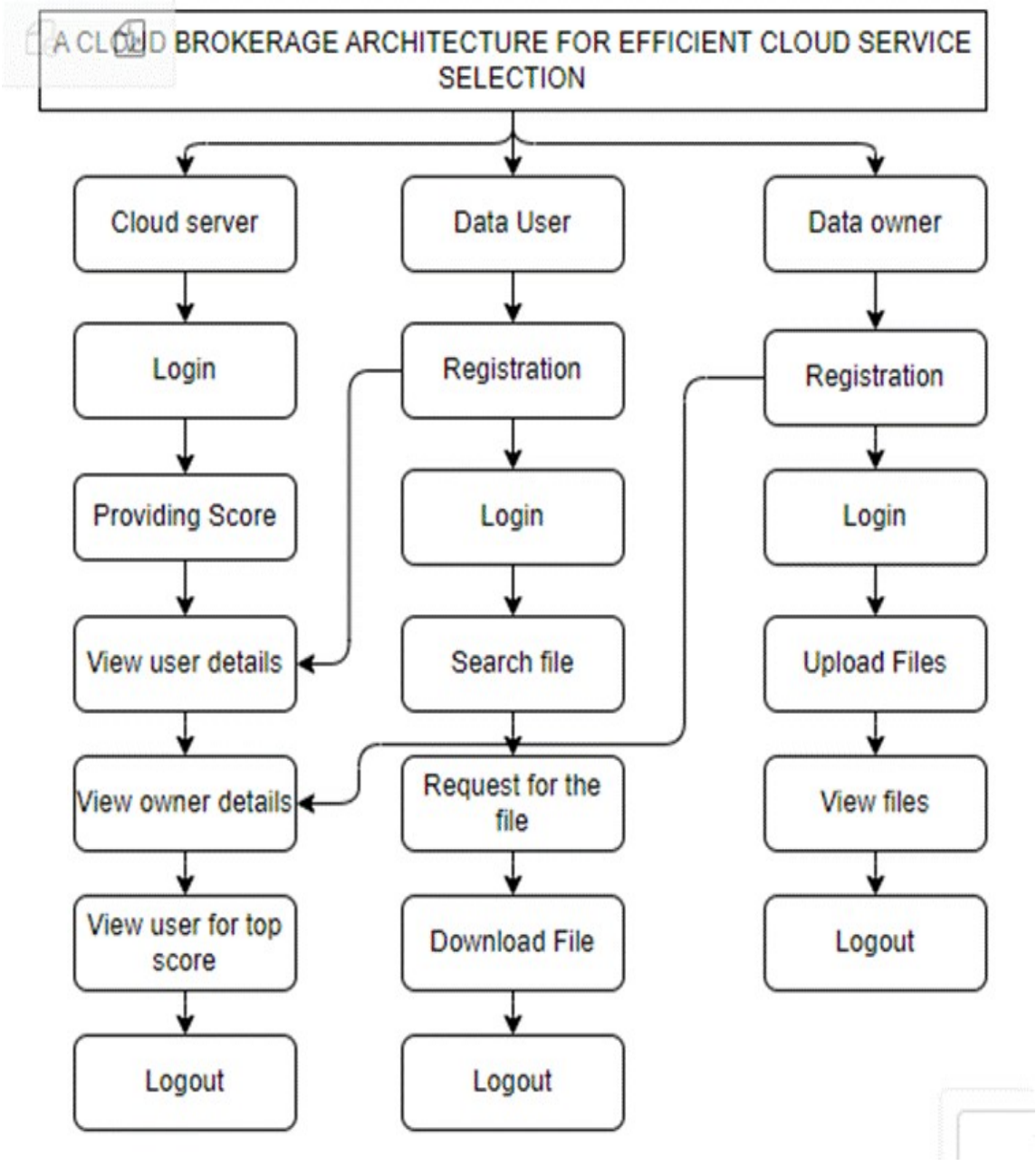


Fig 5-2Block diagram

5.4 INTRODUCTION TO PYTHON

Introduction:

Python is a versatile programming language known for its simplicity and readability. In this document, we'll explore the fundamentals of Python programming and its relevance in web development.

Python Basics:

Python allows for both interactive and script-based programming. Scripts, which are reusable text files containing Python statements, provide a convenient way to execute programs repeatedly without retyping. Python emphasizes ease of use with fewer syntax rules compared to other languages.

Python Features:

- Python is an open-source, high-level, and interpreted language.
- It supports multiple programming paradigms, including object-oriented, procedural, and functional programming.
- Python's standard library offers a broad range of functionalities, making it portable and compatible across different platforms.
- Its interactive mode enables direct interaction with the interpreter for testing and debugging.
- Python is easily extendable, allowing integration with other languages like C/C++, Java, and more.

Python for Web Development:

- Python is widely used in web development due to its simplicity and powerful libraries.
- Frameworks like Django, Flask, and Pyramid provide robust tools for building web applications.
- Python's web frameworks offer features like URL routing, HTML templating, database management, and security enhancements against common attacks.

Python Libraries for Web Development:

Various Python libraries enhance web development capabilities, including Requests for HTTP requests, BeautifulSoup for web scraping, SQLAlchemy for database management, and Matplotlib for data visualization.

Django:

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It follows the Model-View-Controller (MVC) architectural pattern, emphasizing the reusability of components and the "don't repeat yourself" (DRY) principle. Django provides built-in features for URL routing, database management (via Django ORM), authentication, and templating, making it ideal for building scalable and maintainable web applications.

5.5 FRONTEND TECHNOLOGIES (HTML, CSS, JAVASCRIPT)

HTML, CSS, and JavaScript are the core technologies used in frontend web development. Here's a closer look at each:

HTML (Hyper Text Markup Language): HTML is the standard markup language used to create the structure and content of web pages. It consists of a series of elements, each defining different parts of a webpage, such as headings, paragraphs, links, images, and forms.

CSS (Cascading Style Sheets): CSS is a stylesheet language used to control the presentation and layout of HTML documents. It allows developers to style elements, control typography, adjust layout, and create visual effects to enhance the appearance of web pages.

JavaScript: JavaScript is a dynamic programming language that runs in the browser, allowing developers to add interactivity and dynamic behavior to web pages. It enables features such as form validation, animations, dynamic content updates, and interactive user interfaces.

5.6 USER MANUAL FOR EXECUTING DJANGO PROJECT

Setting Up Environment:

Ensure Python and Django are installed on your system. You can install Django using pip:

pip install django

Navigate to your project directory.

Running the Development Server:

Open the terminal or command prompt.

Navigate to your project directory if not already there.

Run the following command to start the development server:

python manage.py runserver

The development server will start, and you'll see output indicating the server address (usually http://127.0.0.1:8000/).

Accessing the Application:

Open a web browser.

Enter the URL provided by the development server (e.g., http://127.0.0.1:8000/) in the address bar.

Press Enter to access your Django application.

Interacting with the Application:

Once the application loads, you can navigate through different pages and functionalities as designed in your Django project.

Submit forms, interact with dynamic elements, and explore the features implemented in your application.

Stopping the Development Server:

To stop the development server, go back to the terminal or command prompt where the server is running.

Press Ctrl + C to stop the server. Confirm the action if prompted.

6 SYSTEM TESTING

6.1 INTRODUCTION

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

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application
.it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Test strategy and approach

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

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format

- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2.2 INTEGRATION TESTING

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

6.2.3 FUNCTIONAL TESTING

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

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

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

6.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the

configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.5 WHITE BOX TESTING

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

6.2.6 BLACK BOX TESTING

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

6.2.7 ACCEPTANCE TESTING

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

6.3 VERIFICATION & VALIDATION TESTING

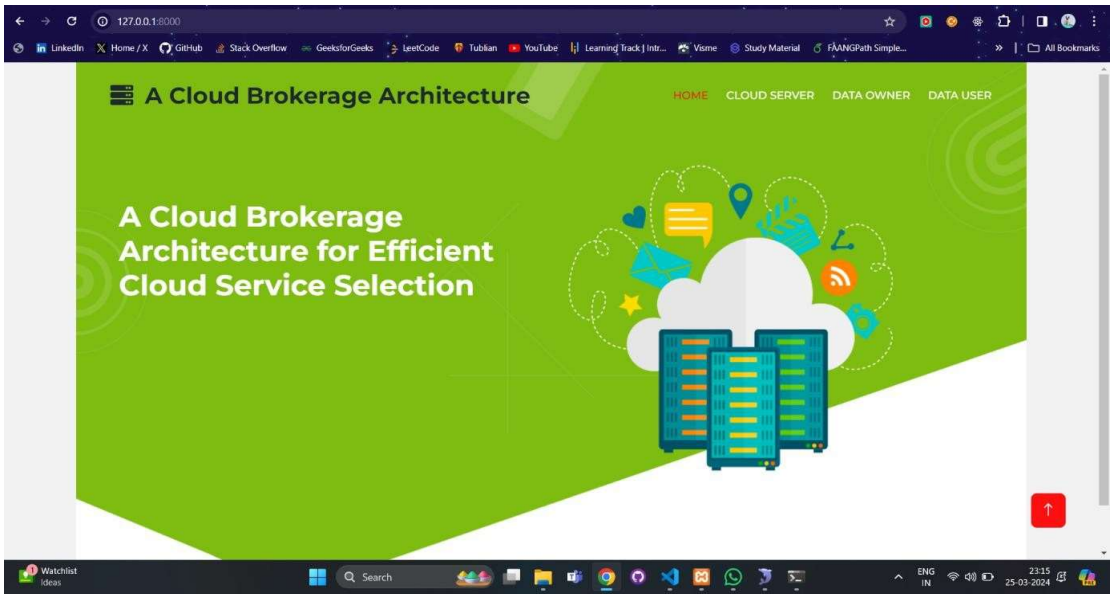
Confirmation Testing guarantees that the framework being created actualizes an exact capacity. This testing is utilized to check that the product that is to be created executes all the functionalities that are specified in the design document. Validation testing ensures that the system going to be produced is in accordance with buyer requirements.

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

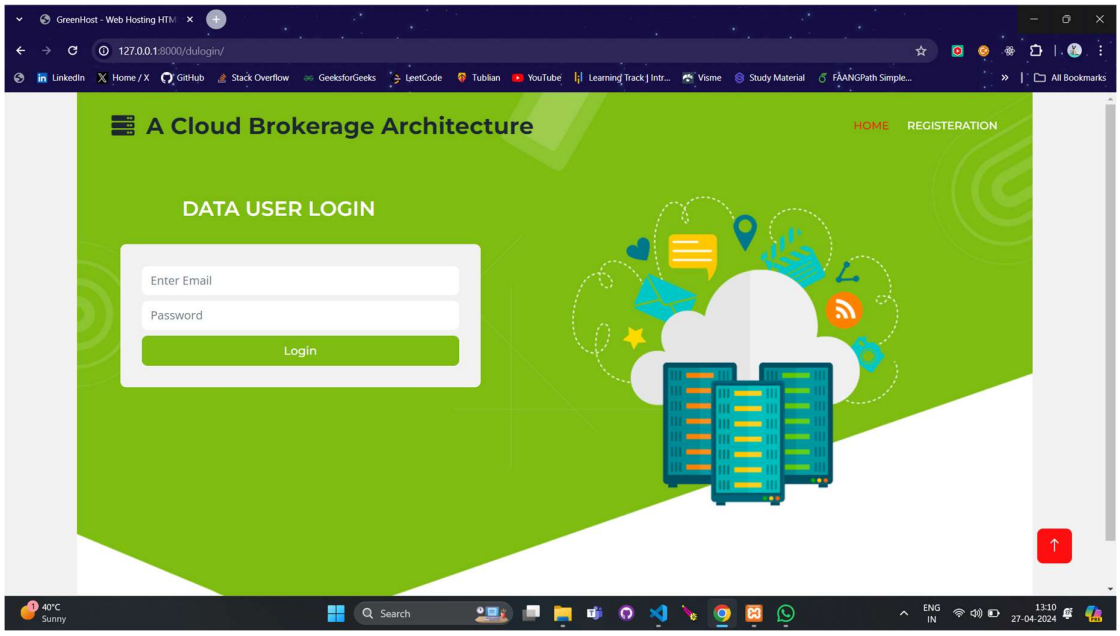
7 RESULTS

OUTPUT SCREENSHOTS:

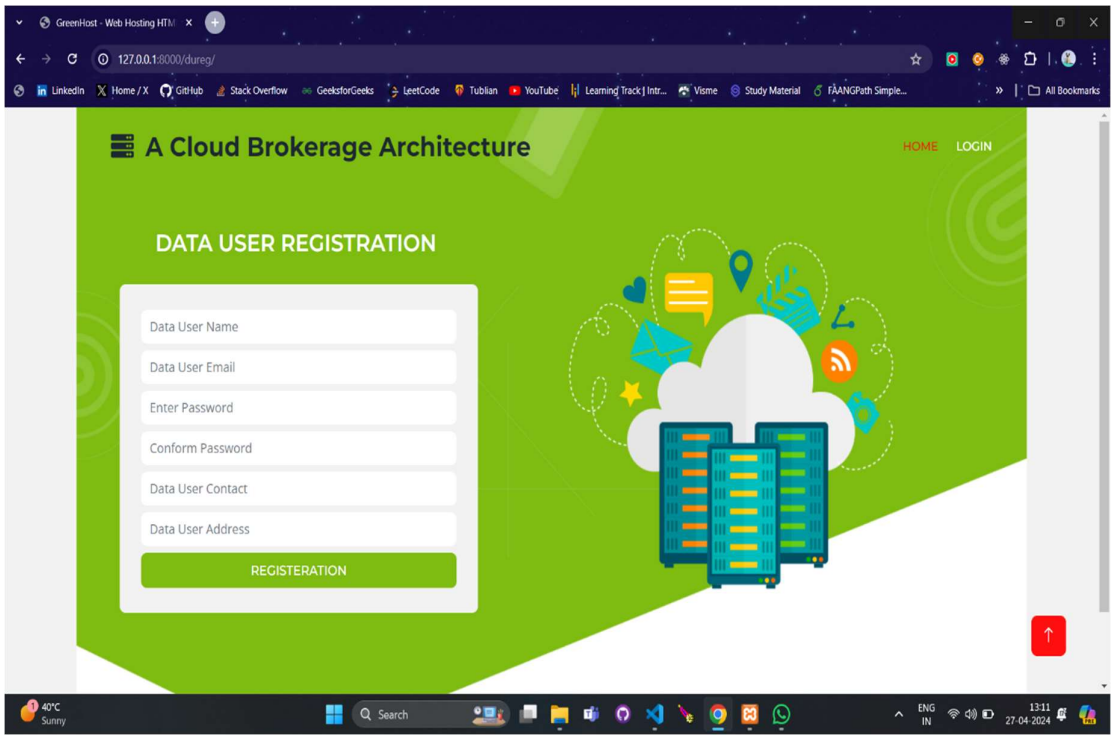
1. **Home Page:** The below figure, shows the home page and it is the first page of CDA. It contains login options for end user and Admin.



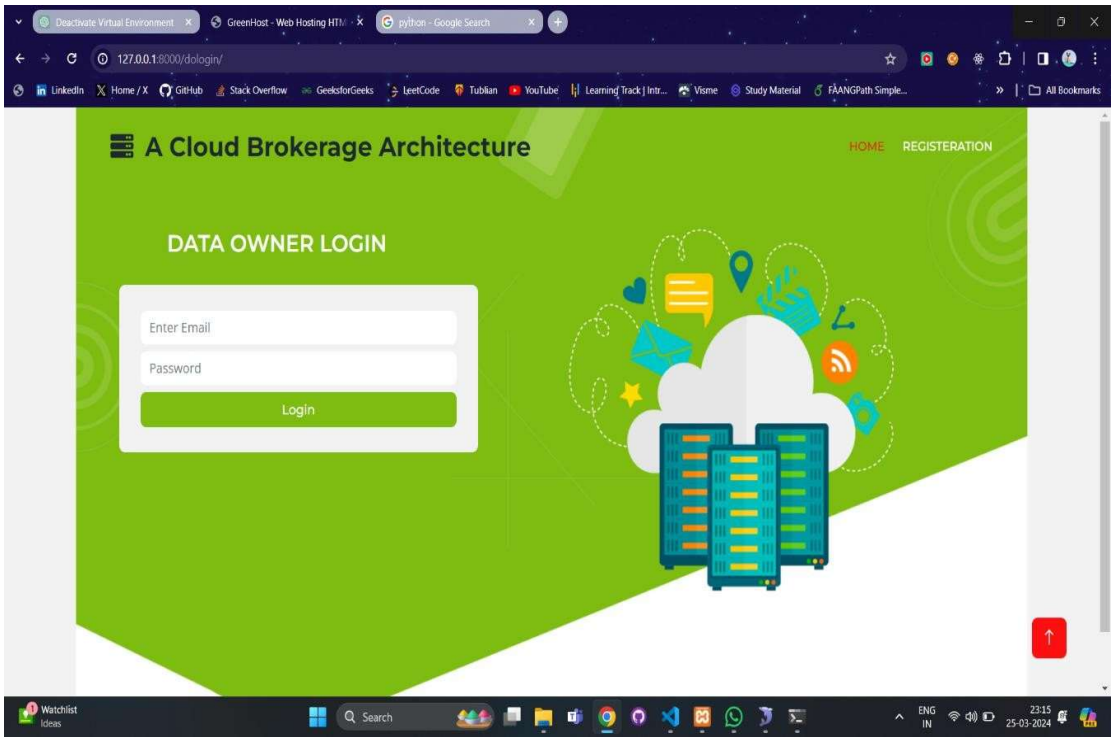
2. **User Login:**



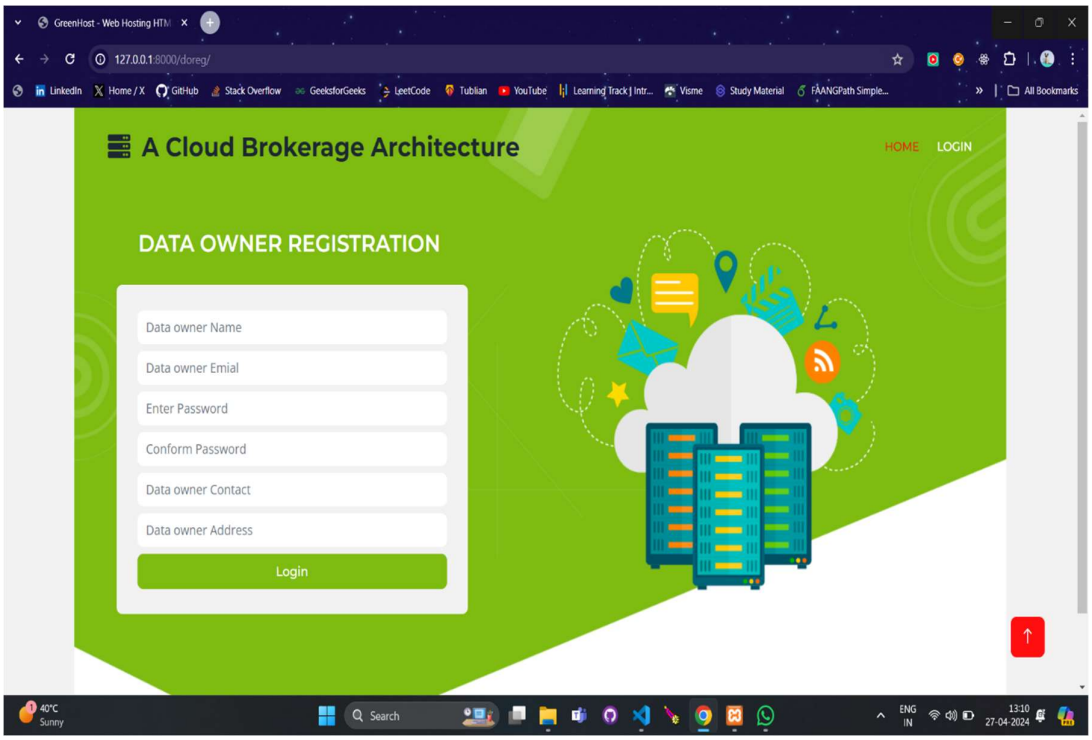
3. Data User Registration:



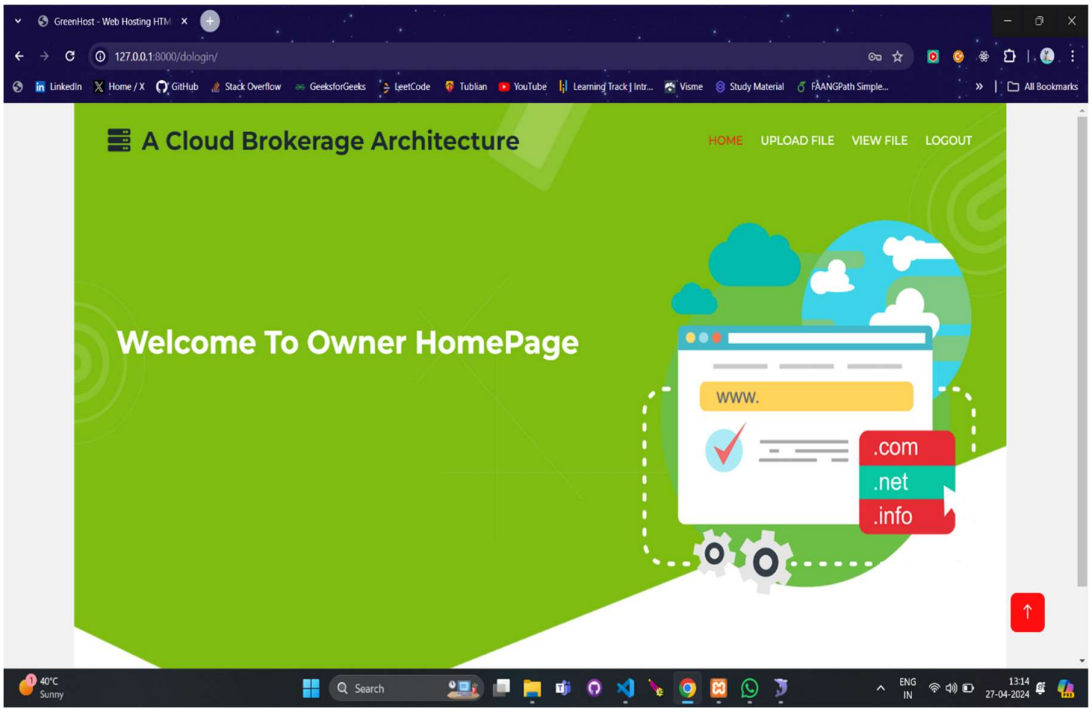
4. Data Owner Login:



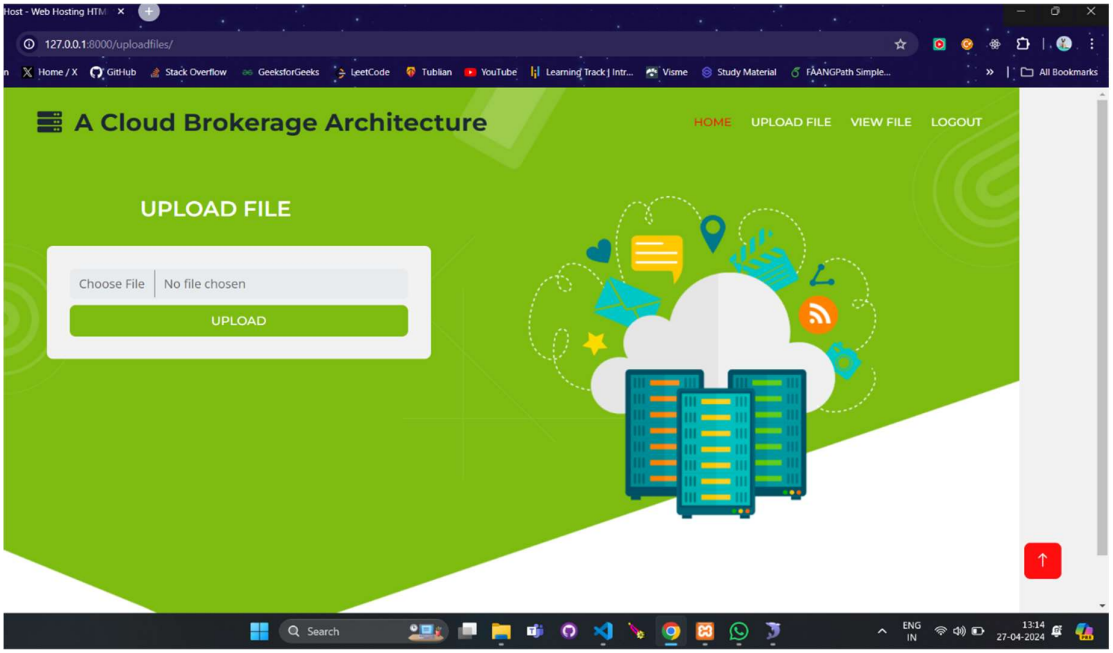
5. Data Owner Registration:



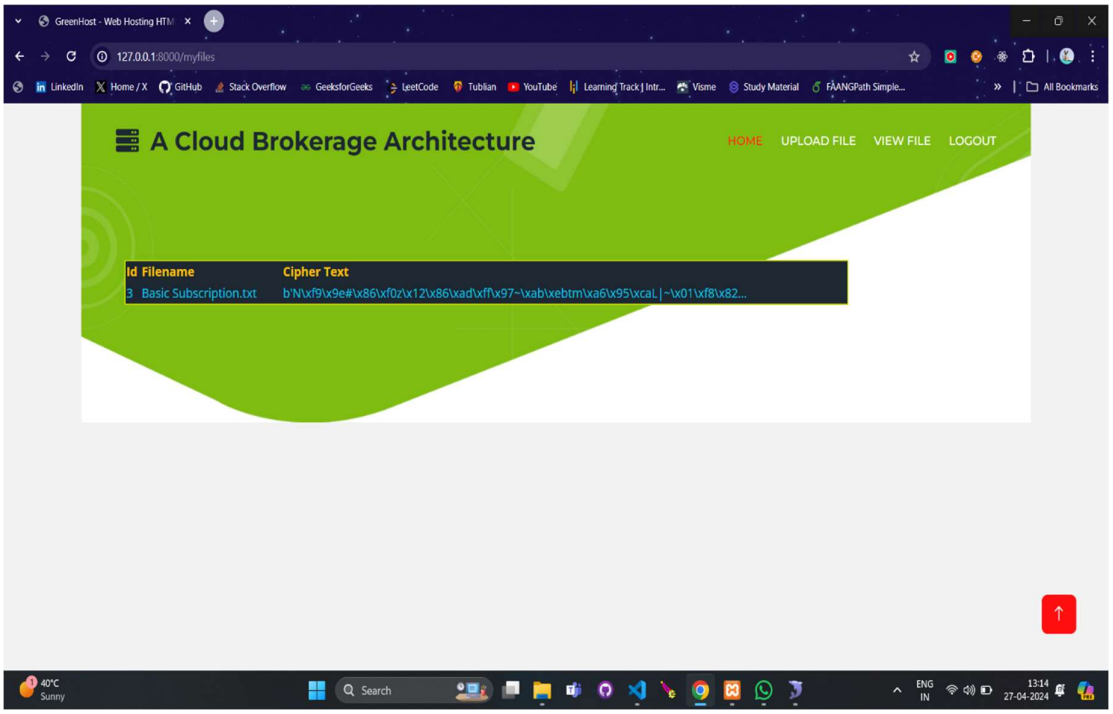
6. Data Owner Home Page:



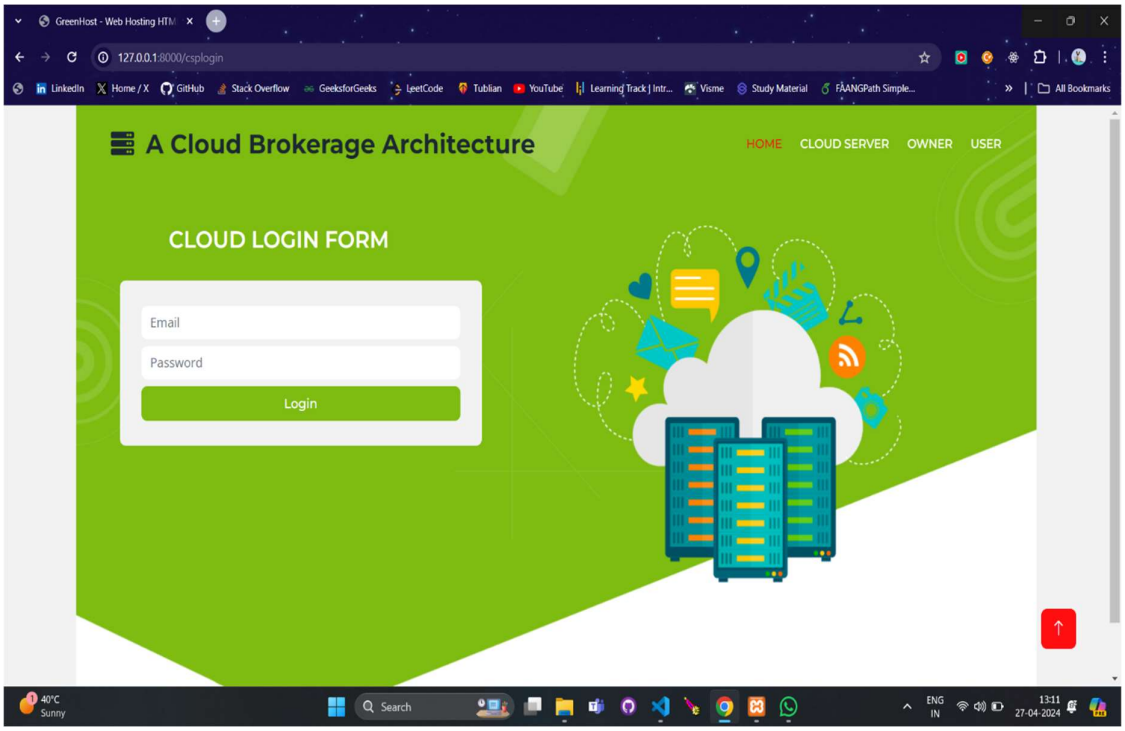
7. **Uploading Files:** Upon logging in as a data owner, navigate to the 'Upload files' option and proceed to upload your files.



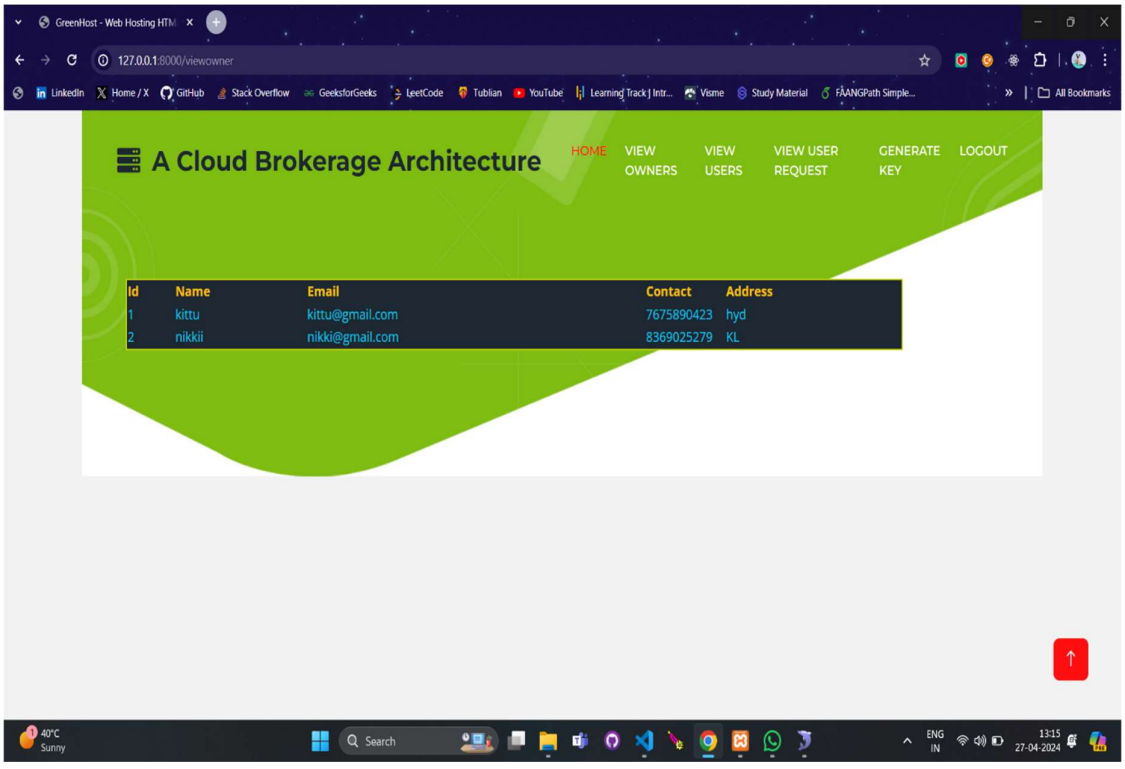
8. **Viewing Uploaded Files:** As a data owner, access the 'View files' option to review the files you have uploaded.



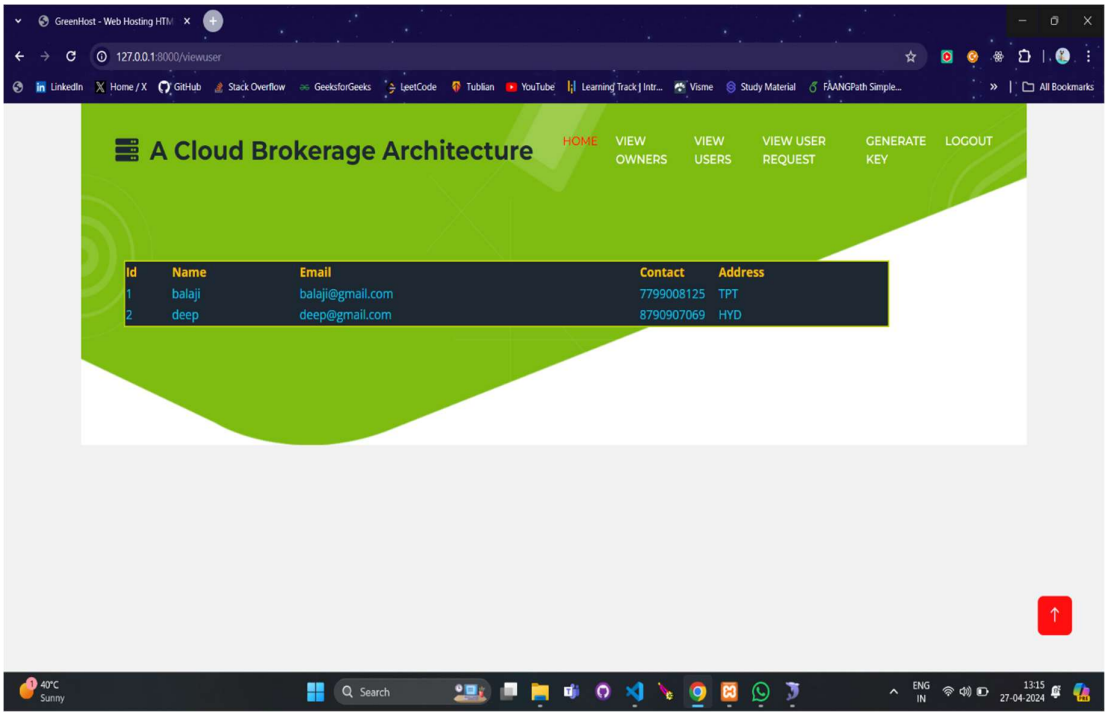
9. CSP Login:



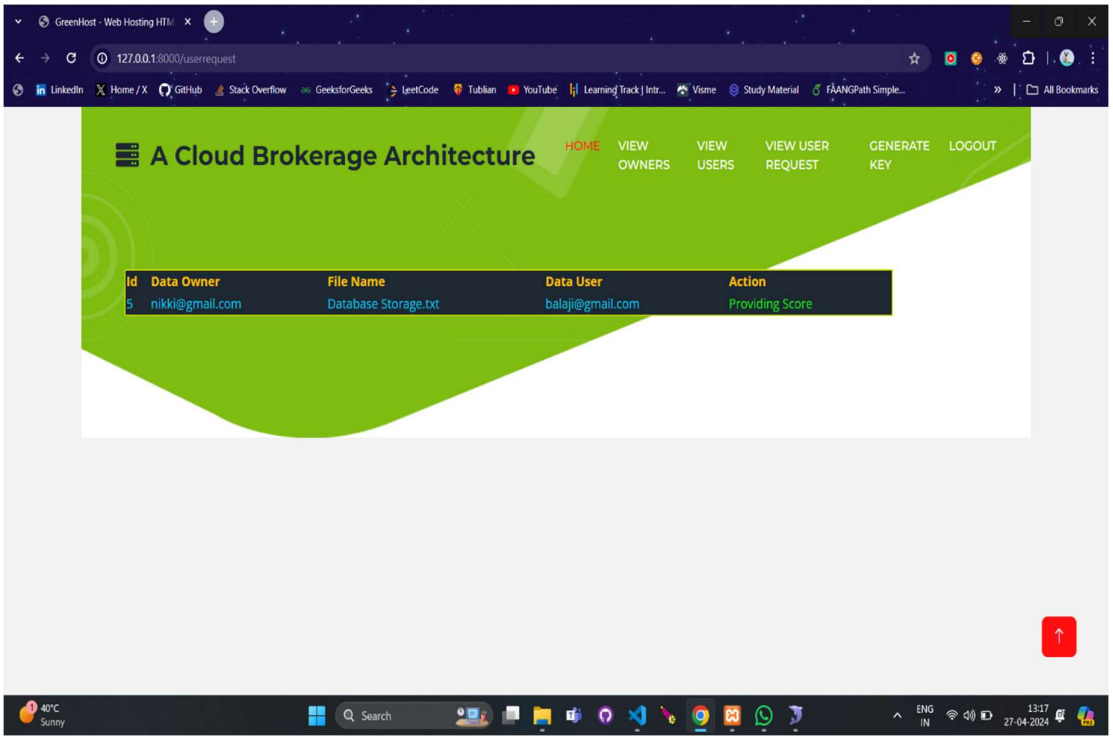
10. Viewing Data Owners: Access the 'View owners' option to see a list of data owners registered within the cloud service platform.



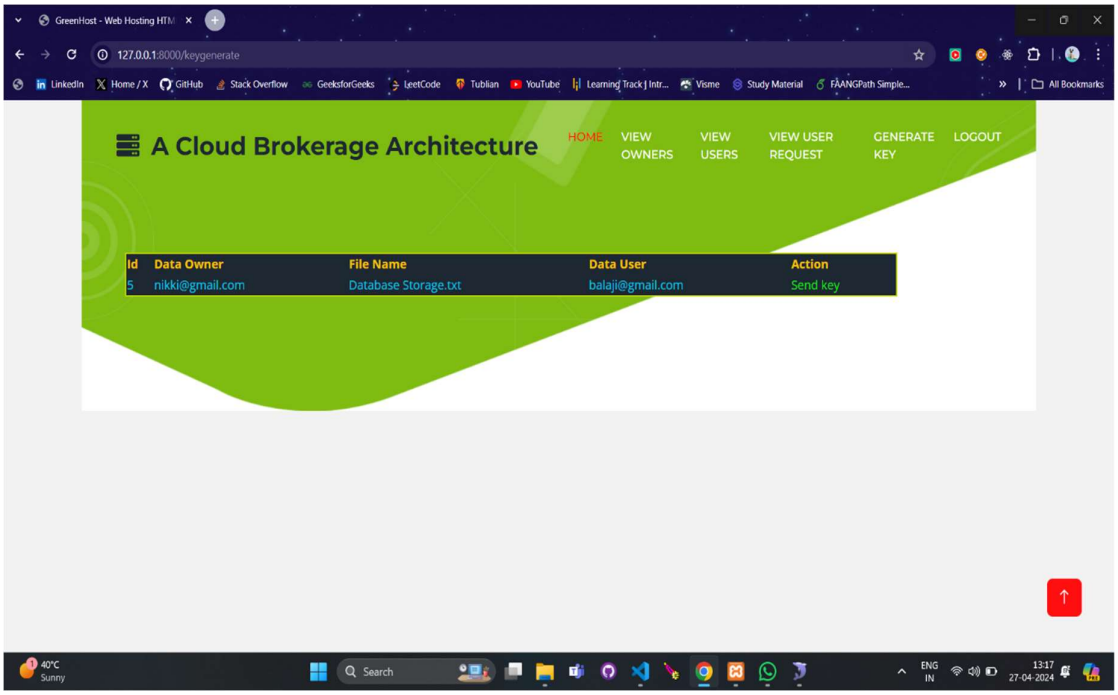
11. **Viewing Data Users:** Navigate to 'View Users' to observe a list of data users who are utilizing services on the cloud platform.



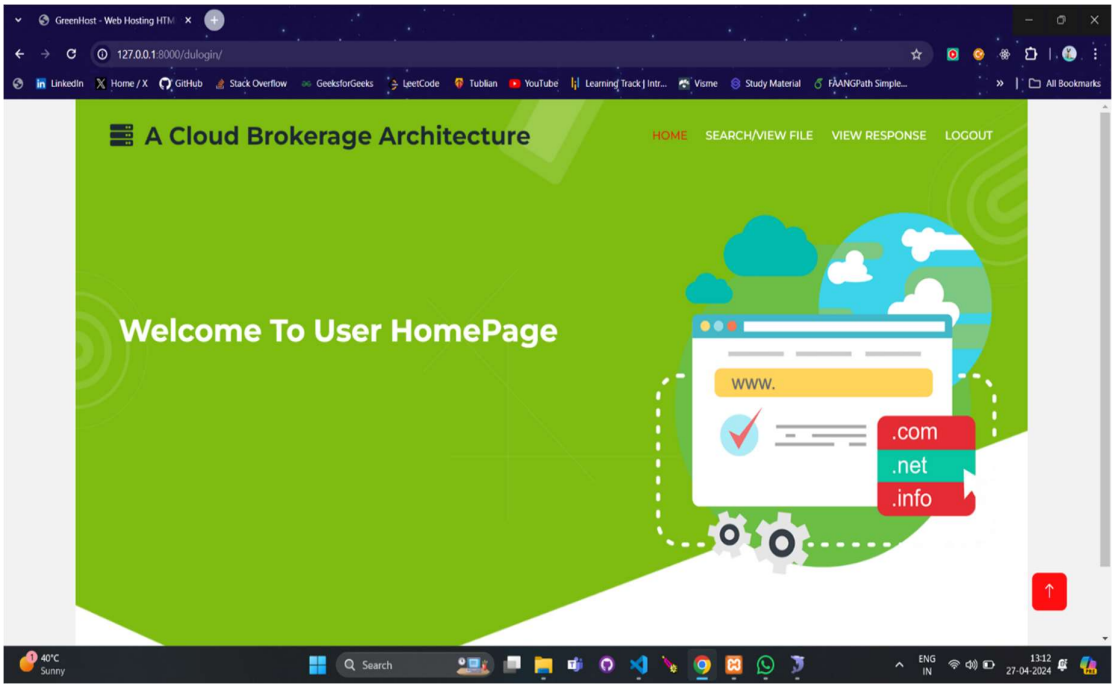
12. **Scoring User Requests:** Review and assign scores to user requests by selecting the 'View user request' feature.



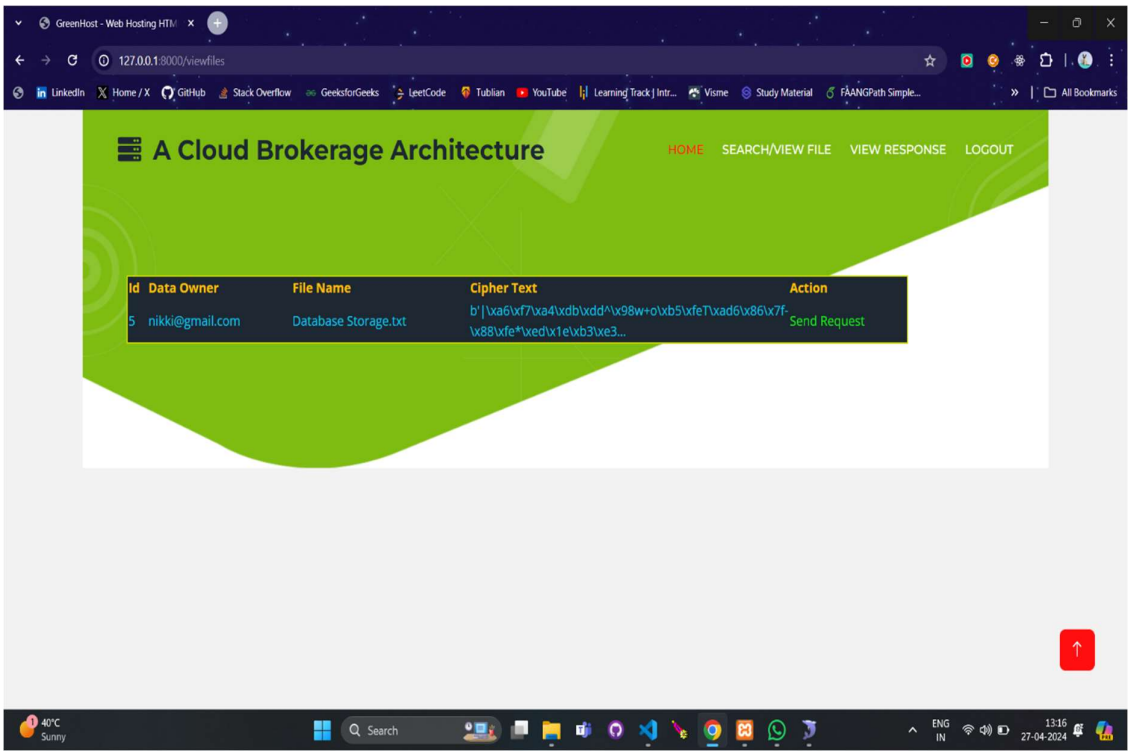
13. **Generating and Sending Keys:** Click on 'Generate key' to create an encrypted key securely. Then, utilize the 'Send key' functionality to provide the generated key to data users for the purpose of secure file downloads.



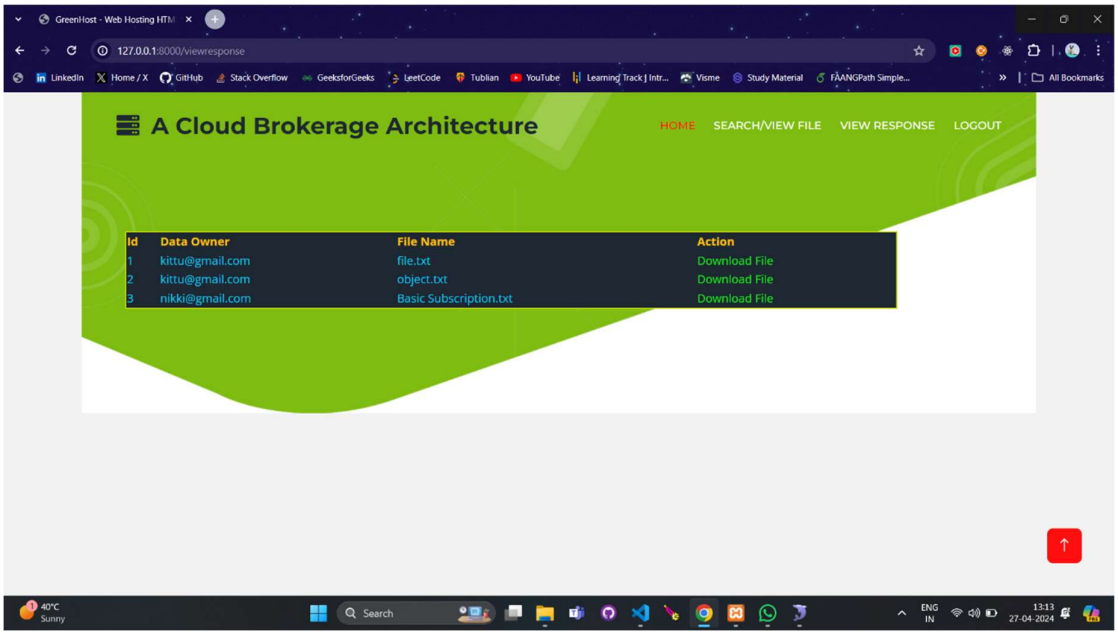
14. **User Home Page:**



15. **Viewing Available Files and Sending Access Requests:** Use the 'Search/View file' option to see the list of available files. To request access or download, click on 'Send request'.



16. **Viewing Granted Requests and Downloading Files:** Check the status of your requests by clicking on 'View response'. Once your request is granted, you can download the file by clicking on the 'Download file' button.



8 SAMPLE CODE

Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <title>A Cloud Brokerage Architecture</title>

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Favicon -->

    <link rel="icon" href="/static/img/favicon.ico">

    <!-- Google Fonts -->

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600;700&f
amily=Open+Sans:wght@400;500;600;700&display=swap" rel="stylesheet">

    <!-- Icon Font Stylesheets -->

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-
icons.css" rel="stylesheet">

    <!-- Libraries Stylesheets -->

    <link href="/static/lib/animate/animate.min.css" rel="stylesheet">

    <link href="/static/lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">

    <!-- Customized Bootstrap Stylesheet -->

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

    <!-- Template Stylesheet -->

    <link href="/static/css/style.css" rel="stylesheet">

</head>
```

```
<body>

   <div class="container-xxl bg-white p-0">

      <!-- Spinner Start -->

      <div id="spinner" class="show bg-white position-fixed translate-middle w-100
vh-100 top-50 start-50 d-flex align-items-center justify-content-center">

         <div class="spinner-border text-primary" style="width: 3rem; height: 3rem;"
role="status">

            <span class="sr-only">Loading...</span>

         </div>

      </div>

      <!-- Spinner End -->

      <!-- Navbar & Hero Start -->

      <div class="container-xxl position-relative p-0">

         <nav class="navbar navbar-expand-lg navbar-light px-4 px-lg-5 py-3 py-lg-0">

            <a href="#" class="navbar-brand p-0">

               <h2 class="m-0"><i class="fa fa-server me-3"></i>A Cloud Brokerage
Architecture</h2>

            </a>

            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarCollapse">

               <span class="fa fa-bars"></span>

            </button>

            <div class="collapse navbar-collapse" id="navbarCollapse">

               <div class="navbar-nav ms-auto py-0">

                  <a href="{% url 'index' %}" class="nav-item nav-link
active">HOME</a>

                  <a href="{% url 'csplogin' %}" class="nav-item nav-link">CLOUD
SERVER</a>

                  <a href="{% url 'dologin' %}" class="nav-item nav-link">DATA
OWNER</a>
```

```
                    <a href="{% url 'dulogin' %}" class="nav-item nav-link">DATA
USER</a>
                </div>
            </div>
        </nav>
        <div class="container-xxl py-5 bg-primary hero-header mb-5">
            <!-- Your main content goes here -->
        </div>
    </div>
    <!-- Navbar & Hero End -->
    <!-- Full Screen Search Start -->
    <div class="modal fade" id="searchModal" tabindex="-1">
        <div class="modal-dialog modal-fullscreen">
            <div class="modal-content" style="background: rgba(29, 40, 51, 0.8);">
                <div class="modal-header border-0">
                    <button type="button" class="btn bg-white btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body d-flex align-items-center justify-content-center">
                    <div class="input-group" style="max-width: 600px;">
                        <input type="text" class="form-control bg-transparent border-light p-
3" placeholder="Type search keyword">
                        <button class="btn btn-light px-4"><i class="bi bi-
search"></i></button>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```
        <!-- Full Screen Search End -->

        <!-- Back to Top -->

        <a href="#" class="btn btn-lg btn-secondary btn-lg-square back-to-top"><i
class="bi bi-arrow-up"></i></a>

    </div>

    <!-- JavaScript Libraries -->

    <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"></sc
ript>

    <script src="/static/lib/wow/wow.min.js"></script>

    <script src="/static/lib/easing/easing.min.js"></script>

    <script src="/static/lib/waypoints/waypoints.min.js"></script>

    <script src="/static/lib/counterup/counterup.min.js"></script>

    <script src="/static/lib/owlcarousel/owl.carousel.min.js"></script>

    <!-- Template Javascript -->

    <script src="/static/js/main.js"></script>

</body>

</html>
```

manage.py

```
#!/usr/bin/env python

"""Django's command-line utility for administrative tasks."""

import os

import sys

def main():

    """Run administrative tasks."""

    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'cloudbrokrage.settings')

    try:
```

```
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
if __name__ == '__main__':
    main()
```

settings.py

```
"""
Django settings for cloudbrokrage project.
Generated by 'django-admin startproject' using Django 4.1.4.
For more information on this file, see
https://docs.djangoproject.com/en/4.1/topics/settings/
For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.1/ref/settings/
"""
from pathlib import Path
import os
# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.1/howto/deployment/checklist/
# SECURITY WARNING: keep the secret key used in production secret!
```

```
SECRET_KEY = 'django-insecure-dkya=@45e^j0+ho0i1q5pmgoj@(6xzw4*npvt3-
xsf)toctv75'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'cloudapp',

    'userapp',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT_URLCONF = 'cloudbrokrage.urls'

TEMPLATES = [

    {
```

```
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
WSGI_APPLICATION = 'cloudbrokrage.wsgi.application'
# Database
# https://docs.djangoproject.com/en/4.1/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME':'cloudbase',
        'USER':'root',
        'PASSWORD' : '',
        'PORT' : '3306',
    }
}
# Password validation
# https://docs.djangoproject.com/en/4.1/ref/settings/#auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [
    {  'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {  'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {  'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {  'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
# mail code
EMAIL_BACKEND="django.core.mail.backends.smtp.EmailBackend"
EMAIL_HOST="smtp.gmail.com"
EMAIL_USE_TLS=True
EMAIL_USE_SSL=False
EMAIL_HOST_USER="user@gmail.com"
EMAIL_HOST_PASSWORD="userpassword"
EMAIL_PORT='587'
# Internationalization
# https://docs.djangoproject.com/en/4.1/topics/i18n/
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/
STATIC_URL = 'static/'
```

```
STATICFILES_DIRS=[os.path.join(BASE_DIR,'static')]
# Default primary key field type
# https://docs.djangoproject.com/en/4.1/ref/settings/#default-auto-field
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

urls.py

```
from django.urls import path
from . import views
urlpatterns=[
    path('',views.index,name='index'),
    path('dologin/',views.dologin,name='dologin'),
    path('doreg/',views.doreg,name='doreg'),
    path('uploadfiles/',views.uploadfiles,name='uploadfiles'),
    path('myfiles',views.myfiles,name='myfiles'),
    path('dulogin/',views.dulogin,name='dulogin'),
    path('dureg/',views.dureg,name='dureg'),
    path('viewdatausers/',views.viewdatausers,name='viewdatausers'),
    path('acceptdatauser/<id>',views.acceptdatauser,name='acceptdatauser'),
    path('viewfiles',views.viewfiles,name='viewfiles'),
    path('sendrequest/<int:id>',views.sendrequest,name='sendrequest'),
    path('viewresponse',views.viewresponse,name='viewresponse'),
    path("viewdatafile/<int:id>",views.viewdatafile,name='viewdatafile')
   path('showfile',views.showfile,name='showfile'),
    path('viewdatafile',views.viewdatafile,name='viewdatafile'),
    path('mydatafile',views.mydatafile,name='mydatafile'),
]
```

9 CONCLUSION

In this project, a brokerage-based architecture for cloud computing systems is presented, alongside an efficient cloud service selection algorithm. The algorithm recommends cloud service providers to consumers based on their requirements. A novel indexing structure, namely Bcloud-tree, is designed to facilitate the organization and retrieval of information about service providers. Additionally, an efficient service selection query algorithm is developed on top of the Bcloud-tree to quickly retrieve desired service providers based on users' service requests. Experimental results demonstrate significant improvements in efficiency and accuracy compared to prior work. Future plans include building an automated parser for extracting manifest variables of cloud service providers and designing strategies to allow cloud users to negotiate some terms of the service level agreements with potential service providers.

10 REFERENCES

[1] Cloud-fuser: Fuzzy ontology and {MCDM} based cloud service selection. Future Generation Computer Systems, 57:42–55, 2016.

[2] S. at TechTarget. Newservers: 2011 top cloud computing provider, 2011.

[3] M. B. and T. F. Decision-making in cloud computing environments: A cost and risk based approach. information Systems Frontiers, 14(4):87193, 2012.

[4] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6):599–616, 2009.

[5] C. W. Chang, P. Liu, and J. J. Wu. Probability-based cloud storage providers selection algorithms with maximum availability. In 41st International Conference on Parallel Processing, pages 199–208, 2012.

[6] CloudSwitch. http://www.cloudswitch.com/.

[7] M. Eggebrecht. Is cloud brokerage the next big thing? In http://www.ciozone.com/index.php/Cloud-Computing/Is-Cloud- Brokerage-the-Next-Big-Thingu.html.

[8] EQUINIX. http://www.equinix.com/industries/cloud-providers/.

[9] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione. Smart cloud storage service selection based on fuzzy logic, theory of evidence and game theory. IEEE Transactions on Computers, PP(99):1–1, 2015

[10] W.-J. Fan, S.-L. Yang, H. Perros, and J. Pei. A multi-dimensional trust- aware cloud service selection mechanism based on evidential reasoning approach. International Journal of Automation and Computing, 12(2):208–219, 2014.

[11] N. Ghosh, S. K. Ghosh, and S. K. Das. Selcsp: A framework to facilitate selection of cloud service providers. IEEE Transactions on Cloud Computing, 3(1):66–79, 2015.

[12] S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh. Efficient service recommendation system for cloud computing market. In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pages 839–845, 2009.

[13] J. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28:100–108, 1979.

[14] Q. He, J. Han, Y. Yang, J. Grundy, and H. Jin. Qos-driven service selection for multi-tenant saas. In IEEE 5th International Conference on Cloud Computing (CLOUD), pages 566–573, 2012.

[15] Ivan. Cloud business trend: Cloud brokerage. Inm http://www.cloudbusinessreview.com/2011/04/26/cloud-business-trend- cloud-brokerage.html.