**ENABLE SECURE AND EFFICIENT MULTIMEDIA CLOUD SERVER**

**PROJECT REPORT**

Submitted by

**K. GOKILAVANISRI**

**(Reg. No: 22-IT-35)**

Under the Guidance of

**Mrs. V. Prabavathi M.Sc., B.Ed., M. Phil., NET., (Ph. D)**

Nallamuthu Gounder Mahalingam College (Autonomous), Pollachi

Affiliated to Bharathiar University, Coimbatore

in partial fulfilment of the requirements for the award of

**BACHELOR OF SCIENCE [INFORMATION TECHNOLOGY]**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**NALLAMUTHU GOUNDER MAHALINGAM COLLEGE**

**(AUTONOMOUS)**

**(An ISO 9001:2015 Certified Institution)**

**(Re-Accredited by NAAC with A++ grade)**

**Pollachi-642 001**

**APRIL-2025**

## 

DECLARATION

## **DECLARATION**

## **K. GOKILAVANISRI (Reg. No: 22-IT-35)** do here by declare that this project entitled **“ENABLE SECURE AND EFFICIENT MULTIMEDIA CLOUD SERVE**R**”** submitted to the Department of Information Technology, NGM College, Affiliated to Bharathiar University, Coimbatore in partial fulfilment of the requirement for the award of **Bachelor Degree** **in Information Technology** is a record of original work done by me during the period of study at NGM College, Pollachi, Under the guidance of **Mrs. V. Prabavathi** M.Sc., B.Ed., M.Phil., NET., (Ph. D) **Assistant Professor, Department of Information Technology**.

## PLACE: Pollachi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## DATE: Signature of the Candidate

## (K. GOKILAVANISRI)

## 

## 

## CERTIFICATES

**CERTIFICATE – I**

I hereby declare that the original project work entitle **“ENABLE SECURE AND EFFICIANT MULTIMEDIA”** is a record work done by **K.GOKILAVANISRI** under the supervision and guidance of me.

PLACE: Pollachi

DATE:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature of the Guide**

**Mrs. V. Prabavathi M.Sc.,B.Ed.,M.Phil.,NET., (Ph.D)**

**CERTIFICATE -II**

This is to certify that the project entitled “**ENABLE SECURE AND EFFICIENT MULTIMEDIA CLOUD SERVER”** is a bonafied record of work done by **K.GOKILAVANISRI (Reg.No.:22-IT-35)** submitted in partial fulfilment of the Requirements for the award of the degree Bachelor of Science (Information Technology) under Bharathiar University, Coimbatore, during the academic year 2024-2025 under my supervision.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature of the Guide** **Signature of the HOD**

Counter Signed by

For external viva – voce examinations held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

I thank God for giving me good health and wisdom to do my project and complete it successfully. It is my immense pleasure to acknowledge the contributors towards the successful completion of this project.

It is a great privilege and pleasure of mine to express my deep sense of gratitude to **Dr. R. MANICKACHEZIAN M.Sc., M.S., Ph.D**., Principal, Nallamuthu Gounder Mahalingam College, Pollachi for allowing me to take on this project.

I express my heart full thanks to **Mr. K. VIJAYAKUMAR** **MCA., M.Phil. SET**  Head, Department of Information Technology for his kind help and guidance in completing this project.

I would like to express my deep sense of gratefulness and sincere thanks to my guide **Mrs. V. PRABAVATHI M.Sc., B.Ed., M.Phil., NET.,(Ph. D)** Assistant Professor, Department of Information Technology, NGM College, Pollachi, for her valuable suggestions and timely guidance, which paved way to streamline and who is behind the success of this project and my career.

I also extend my sincere thanks to all faculty members and staff of the **Department of Information Technology, NGM College, Pollachi,** for their valuable assistance and encouragement.

Finally, I am profoundly grateful to my family, friends, and well-wishers for their unwavering support, motivation, and encouragement throughout this project.

**CONTENTS**

**CONTENT**

**Page. No.**

**1. INTRODUCTION 1**

1.1. OVERVIEW OF THE PROJECT

1.2. SYSTEM SPECIFICATION

1.2.1. HARDWARE SPECIFICATION

1.2.2. SOFTWARE SPECIFICATION

1.3. FRONT END-VB.NET

1.4. BACK END-SQL SERVER

**2. SYSTEM ANALYSIS 7**

2.1. EXISTING SYSTEM

2.1.1. DISADVANTAGES OF EXISTING SYSTEM

2.2. PROPOSED SYSTEM

2.2.1. ADVANTAGES OF PROPOSED SYSTEM

**3. SYSTEM DESIGN 8**

3.1. DESIGN NOTATION 3.1.1. DATA FLOW DIAGRAM

3.1.2. E R DIAGRAM

3.2. DESIGN PROCESS

3.2.1. INPUT DESIGN

3.2.2. DATABASE DESIGN

3.2.3. TABLE DESIGN

3.2.4. OUTPUT DESIGN

**4 .SYSTEM TESTING AND IMPLEMENATION 19**

4.1. SYSTEM TESTING

4.2. IMPLEMENTATION

**5. CONCLUSION AND FUTURE ENHANCEMENT 22**

**6. BIBLIOGRAPHY 23**

**APPENDIX 24**

**SAMPLE SCREENS**

**INTRODUCTION**

**1.INTRODUCTION**

**1.1. OVERVIEW OF THE SYSTEM**

This project is entitled as **“ENABLE SECURE AND EFFICIENT MULTIMEDIA CLOUD SERVER”**. This project is developed in .Net platform. The front end of this project is ASP.Net and the back end is SQL Server. It is designed primarily to enable the storage and retrieval of data while the computation is carried out by the cloud server.

Cloud server is a computer attached to a network that has the primary purpose of providing a location for shared disk access, i.e. shared storage of computer files (such as images) that can be accessed by the workstations that are attached to the same computer network. The term server highlights the role of the machine in the client–server scheme, where the clients are the workstations using the storage. A file server is not intended to perform computational tasks, and does not run programs on behalf of its clients.

Cloud servers generally offer some form of system security to limit access to files to specific users or groups. These servers work within the hierarchical computing environment which treat users, computers, applications and files as distinct but related entities on the network and grant access based on user or group credentials.

**MODULES**

* Authentication Module
* User Registration Module
* Network Accessing Module
* Share Data Module
* Data Access Module

**MODULE DESCRIPTION**

* **Authentication Module**

This module is to register the new users and previously registered users can access in this application. The Users can enter and do the uploading /download files into the servers.

* **User Registration Module**

Through this module, the user can login their account and can modify their personal details such as changing the address, date of birth etc.

* **Network Accessing Module**

Through this module, the friend request can be given to other user and if the

request is accepted, the users are added in the list.

* **Share Data Module**

File sharing is the practice of distributing or providing access to digitally stored information, such as multimedia files.

* **Data Access Module**

This module allows us to download the file which is uploaded by the users.

According to the rights specified by the users. The users rights allow all the users to

download the files.

**1.2 SYSTEM SPECIFICATION**

**1.2.1 HARDWARE SPECIFICATION**

|  |  |
| --- | --- |
| Processor | : Intel Core i3 |
| RAM | : 4 GB DDR4 |
| Hard Disk | : 500 GB HD / SSD |
| Clock Speed | : 2.1 GHz |
|  |  |

**1.2.2 SOFTWARE SPECIFICATION**

|  |  |
| --- | --- |
| Technology | : Microsoft Visual Studio 2010 |
| Front End | : ASP.Net 2010 |
| Back End | : SQL Server 2008 |

**1.3 FRONT END- ASP.NET**

Programs written for the .NET Framework execute in a [e](http://en.wikipedia.org/wiki/Software)nvironment that manages the program's [runtime r](http://en.wikipedia.org/wiki/Runtime)equirements. Also part of the .NET Framework, this runtime environment is known as the Common Language Runtime (CLR). The CLR provides the appearance of an [application virtual machine so](http://en.wikipedia.org/wiki/Virtual_machine#Application_virtual_machine) that programmers need not consider the capabilities of the specific [CPU t](http://en.wikipedia.org/wiki/Central_Processing_Unit)hat will execute the program. The CLR also provides other important services such as security, [memory management,](http://en.wikipedia.org/wiki/Memory_management) and [exception handling.](http://en.wikipedia.org/wiki/Exception_handling) The class library and the CLR together compose the .NET Framework.

The Common Language Runtime (CLR) is the virtual machine component of the .NET framework. All .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling.

The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages using the .NET Framework. The BCL provides classes which encapsulate a number of common functions.

Installation of computer vehicle must be carefully managed to ensure that it does not interfere with previously installed vehicle, and that it conforms to security requirements. .NET framework includes design features and tools that help address these requirements. The design is meant to address some of the vulnerabilities, such as buffer overflows, that have been exploited by malicious vehicle.

The design of the .NET Framework allows it to theoretically be platform agnostic, and thus cross-platform compatible. That is, a program written to use the framework should run without change on any type of system for which the framework is implemented. The intermediate CIL code is housed in .NET assemblies. As mandated by specification, assemblies are stored in the Portable Executable (PE) format, common on the Windows platform for all DLL and EXE files.

The assembly consists of one or more files, one of which must contain the manifest, which has the metadata for the assembly. The complete name of an assembly (not to be confused with the filename on disk) contains its simple text name, version number, culture, and public key token.

The public key token is a unique hash generated when the assembly is compiled, thus two assemblies with the same public key token are guaranteed to be identical from the point of view of the framework. The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. Editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages.

ASP.NET makes it easy to perform common tasks, from simple form submission and costumer authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

**1.4 BACK END- SQL SERVER**

SQL Server adds certain features including a [master data management](http://en.wikipedia.org/wiki/Master_data_management) system branded as [Master Data Services](http://en.wikipedia.org/wiki/Microsoft_SQL_Server_Master_Data_Services), a central management of master data entities and hierarchies. Also Multi Server Management, a centralized console to manage multiple SQL Server 2008 instances and services including relational databases, Reporting Services, Analysis Services & Integration Services.

SQL Server new features and enhancements include Always On SQL Server Failover Cluster Instances and Availability Groups which provides a set of options to improve database availability Contained Databases which simplify the moving of databases between instances, new and modified Dynamic Management Views and Functions, programmability enhancements including new spatial features, metadata discovery, sequence objects and the THROW statement, performance enhancements such as Column Store Indexes as well as improvements to On Line and partition level operations and security enhancements including provisioning during setup, new permissions, improved role management, and default schema assignment for groups.

SQL Server provides a new in-memory capability for tables that can fit entirely in memory. Whilst small tables may be entirely resident in memory in all versions of SQL Server, they also may reside on disk, so work is involved in reserving RAM, writing evicted pages to disk, loading new pages from disk, locking the pages in RAM while they are being operated on, and many other tasks. By treating a table as guaranteed to be entirely resident in memory much of the 'plumbing' of disk-based databases can be avoided.

**2. SYSTEM ANLYSIS**

**2.1 EXISTING SYSTEM**

In modern businesses the design of file servers is complicated by competing demands for storage space, access speed, and recoverability, ease of administration, security, and budget. This is further complicated by a constantly changing environment, where new hardware and technology rapidly vanishes old equipment, and yet must seamlessly come online in a fashion compatible with the older machinery.

**2.1.1 DISADVANTAGES OF EXISTING SYSTEM**

* Low disk Space available on existing system
* Users account will Expire
* Limited size of data
* The information can be shared through the public domain
* No separate security for administrator

**2.2 PROPOSED SYSTEM**

These servers work within the hierarchical computing environment which treat users, computers, applications and files as distinct but related entities on the network and grant access based on user or group credentials. In many cases, the directory service spans many file servers, potentially hundreds for large organizations. In the past, and in smaller organizations, authentication can take place directly to the server itself.

**2.2.1 ADVANTAGE OF PROPOSED SYSTEM**

* Search engine visibility is more in proposed system
* Cost Efficient and Better security in this proposed system
* Effective application to be used within the organization
* Information are send securely in this proposed system

**3. SYSTEM DESIGN**

**3.1 DESIGN NOTATION**

**3.1.1 DATAFLOW DIAGRAM**

**Level 0**

User

Enable Secure And Efficient Multimedia Cloud Server

Access Server

**Interaction**

**Login**

**Level 1**

**New**

1.0

User Registration

User

**Create**

User Account

1.1

Network request

**Generate**

Interaction Report

**Level 2**

**ID, PWD**

2.1

Network Access

**New**

User Details

User

**Access**

2.2

Interaction

**View**

**Update**

Share Data

**3.1.2 ENTITY RELATIONSHIP DIAGRAM**

View

Interaction

Maintenance

Membership

Users

Network Request

File Sharing

**3.2 DESIGN PROCESS**

System design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture.

**3.2.1 INPUT DESIGN**

Input design is the process of converting user-originated inputs to a computer-based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system.

The input forms in these projects are,

* **User Registration Form** – Form to collect user information for account creation (e.g., username, password, email, phone number).
* **User Authentication Form** – Form for users to log in using credentials (e.g., username/email and password).
* **Network Request Form** – Form for submitting network-related requests, such as accessing a particular service or resource. (e.g., request for IP address, network access permissions).
* **Data Sharing Form** – Form to request or manage sharing of data between users or systems, defining permissions and access rights (e.g., sharing files, granting access to databases).

**3.2.2 DATABASE DESIGN**

The database design is a must for any application developed especially more for the data store projects. Since the chatting method involves storing the message in the table and produced to the admin, proper handling of the table is a must.

In the project, admin table is designed to be unique in accepting the username and the length of the username and password should be greater than zero.

**3.2.3 TABLE DESIGN**

1. **Table Name:** User\_Details

**Primary Key:** UserID

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| UserID | nvarchar | 20 | Registered user ID |
| Password | nvarchar | 20 | Registered Password |
| Email | nvarchar | 30 | Registered user Email |
| RegQuestion | nvarchar | MAX | Registered user security question |
| RegAnswer | nvarchar | MAX | Registered user security Answer |
| CreateDate | date | --- | Created date of registered user |

1. **Table Name:** Membership\_Details

**Foreign Key:** UserID

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **DataType** | **Size** | **Description** |
| UserID | nvarchar | 20 | Registered user ID |
| UserName | nvarchar | 50 | Registered user Name |
| FullName | nvarchar | 50 | Full name of registered user |
| Address | nvarchar | MAX | Address of the registered user |
| DOB | date | --- | Date of birth of the registered user |
| Gender | varchar | 6 | Gender of the registered user |
| UserImage | varbinary | MAX | User image of the registered user |

1. **Table Name:** NetworkRequest\_Details

**Primary Key:** DisplayID

**Foreign Key:** UserID

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| DisplayID | nvarchar | 20 | Display ID |
| UserID | nvarchar | 20 | Registered user ID |
| Message | nvarchar | MAX | Message sent by the user |
| UserStatus | nvarchar | 20 | Status of the User |
| ActivatedDate | date | --- | Date of the Activated |

1. **Table Name:** FileSharing\_Details

**Primary Key:** FSID

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| FSID | bigint | 5 | File Sharing ID |
| SData | varbinary | MAX | Shared data |
| SName | nvarchar | 50 | Shared file name |
| SSize | varbinary | MAX | Shared file size |
| SType | nvarchar | 20 | Shared file type |
| DateShared | date | --- | Date of file shared |
| UserName | nvarchar | 50 | User Name of file shared |

1. **Table Name:** Interaction\_Details

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| UserID | nvarchar | 20 | Registered user ID |
| Interaction | nvarchar | MAX | Interaction by user |
| InteractionDate | date | --- | Date of Interaction |

**Foreign Key:** UserID

**3.2.4 OUTPUT DESIGN**

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. In any system, the output design determines the input to be given to the application. The output is designed in such a way that it is attractive, convenient and informative. Forms are designed in ASP.NET with various features, which make the console output more pleasing. As the outputs are the most important sources of information to the users, better design should improve the system’s relationships with us and also will help in decision-making. Form design elaborates the way output is presented and the layout available for capturing information.

The output forms in these projects are,

* Data Sharing Form - Form is a document used to obtain consent and define terms when sharing data between parties. It ensures that the data provider and recipient understand the purpose, scope, and conditions of sharing.
* Data Accessing Form - Form is a request document used to gain access to specific data. It helps organizations control and monitor who can access data, ensuring security.

**4. SYSTEM TESTING AND IMPLEMENTATION**

**4.1 SYSTEM TESTING**

Testing is the process of confirming that a program or system does what it is proposed off, Testing is the only way to assure the quality of s/w and it is an umbrella activity rather that a separate phase. This is an activity to be performed in parallel with the s/w efforts and one that consists of its own phase of analysis, design, implementation, execution and maintenance.

During analysis and design, an s/w verification plan and acceptance test plan is prepared. The verification plan describes the methods to be used in verifying that the requirements are satisfied by the design documents and that the source is consistent with the requirements specification and design documents. The acceptance test plan includes test cases, outcomes and capabilities demonstrated by each test case. Following completion of the verification plan and Acceptance plan, an s/w verification review is held to evaluate the adequacy of the plans.

During product evolution, in-process audits are conducted to verify consistency and completeness of the work products. Items to be audited for consistency include interface specification for hardware and software: internal design verses functional requirements verses test descriptions.

Prior to product delivery, a functional audit and a physical audit performed. The functional audit reconfirms that all the requirements have been met. The physical audit verifies that the source code and all associated documents are complete, consistent with one another and ready to deliver. An s/w verification summary is prepared to describe the results of all reviews.

1. **Security Testing**: Ensure the cloud server is secure in handling multimedia content (e.g., video, images).
2. **Efficiency Testing**: Verify that the server performs optimally when processing and delivering multimedia content.

| **Test Case ID** | **Test Case Description** |  | **Objective** | **Test Steps** | **Expected Result** | **Pass Criteria** |
| --- | --- | --- | --- | --- | --- | --- |
| **SEC-01** | Secure Upload of Multimedia Files |  | Ensure secure upload of multimedia files to the server. | 1. Upload multimedia file over HTTPS. 2. Check upload directory permissions. 3. Scan file for malware. | Files should upload securely over HTTPS with proper validation and no malicious content. | Upload over HTTPS, file scanned and no malware detected. |
| **SEC-02** | Access Control (Authentication and Authorization) |  | Test unauthorized access to multimedia content. | 1. Attempt to access files without login. 2. Attempt access with unauthorized user. 3. Log in as authorized user. | Unauthorized access should be denied, only authorized users can access the content. | Unauthorized users denied access, authorized users allowed access. |
| **SEC-03** | Data Encryption at Rest and in Transit |  | Ensure encryption of multimedia data during upload/download and storage. | 1. Verify SSL/TLS for file transfer. 2. Check encryption of stored files (e.g., AES). | Data should be encrypted in transit (SSL/TLS) and at rest (AES or similar encryption). | Data encrypted in transit and storage. |
| **SEC-04** | Secure File Deletion |  | Verify that multimedia files are securely deleted from the server. | 1. Upload multimedia files. 2. Request file deletion. 3. Verify the files are completely deleted. | Files should be securely and fully deleted without residual data. | Files fully deleted and cannot be restored. |
| **EFF-01** | Server Response Time for Multimedia Content |  | Verify server response time when retrieving multimedia content. | 1. Request a multimedia file. 2. Measure the time it takes to retrieve and deliver the file. | The server should respond within an acceptable time frame (e.g., < 3 seconds). | Server response time under 3 seconds. |
| **EFF-02** | Scalability Under Load |  | Test the server's ability to handle multiple simultaneous requests for multimedia content. | 1. Simulate 100+ concurrent users requesting files. 2. Measure server response time and resource usage. | The server should scale without performance degradation, handling concurrent requests effectively. | Server handles load without crashes or delays. |
| **EFF-03** | Efficient Bandwidth Usage |  | Test the server's management of bandwidth when serving multimedia content. | 1. Request large files (videos/images). 2. Measure bandwidth usage. 3. Check if the server optimizes file size (e.g., compression, streaming). | The server should optimize bandwidth usage for multimedia delivery (e.g., compression, adaptive bitrate). | Bandwidth usage is optimized. |
| **EFF-04** | Cache Effectiveness |  | Verify that caching mechanisms are effectively used for multimedia content. | 1. Request a file multiple times. 2. Verify if caching is used (browser cache, CDN). 3. Measure time to load the file on subsequent requests. | The file should load faster on subsequent requests due to caching. | File load time is faster on subsequent requests, indicating caching is used. |
| **BKP-01** | Backup and Recovery of Multimedia Content |  | Verify multimedia content is backed up and can be recovered. | 1. Upload multimedia files. 2. Backup files. 3. Simulate server failure. 4. Attempt file recovery. | Multimedia files should be recoverable from backup without corruption. | Files successfully restored with no significant data loss. |

**4.2 SYSTEM IMPLEMENTATION**

Implementation refers to post-sales process of guiding a client from purchase to use of the software or hardware that was purchased. This includes requirements analysis, scope analysis, customizations, systems integrations, user policies, user training and delivery. These steps are often overseen by a project manager using [project management](http://en.wikipedia.org/wiki/Project_management) methodologies. Software Implementations involve several professionals that are relatively new to the knowledge based economy such as business analysts, technical analysts, solutions architects, and project managers.

To implement a system successfully, a large number of inter-related tasks need to be carried out in an appropriate sequence. Utilizing a well-proven implementation methodology and enlisting professional advice can help but often it is the number of tasks, poor planning and inadequate resourcing that causes problems with an implementation project, rather than any of the tasks being particularly difficult. Similarly with the cultural issues it is often the lack of adequate consultation and two-way communication that inhibits achievement of the desired results.

System implementation generally benefits from high levels of user involvement and management support. User participation in the design and operation of information systems has several positive results. The relationship between users and information systems specialists has traditionally been a problem area for information systems implementation efforts. Users and information systems specialists tend to have different backgrounds, interests, and priorities. This is referred to as the user-designer communications gap. These differences lead to divergent organizational loyalties, approaches to problem solving, and vocabularies.

5. CONCLUSION AND FUTURE ENHANCEMENTS

5.1 CONCLUSION

The goal of **ENABLE SECURE AND EFFICIENT MULTIMEDIA CLOUD SERVER** is to develop the system that was found to be accurate and user friendly. So the system was treated with all the sample data. The system also provides full security and high reliability. The security helps the user by not allowing any unauthorized person to enter the system and access the personal data.

As focused this systemhas agreat facility to all users. All users can benefit using this project, because users can share data in network with high security. We hope that our project may satisfy the user.The up gradation if any to the system can be done without affecting proper functioning of the system. The documentation provided helps a person with a minimum amount of knowledge to understand it well. It eased the user with the less typing of messages and the user can select everything in simply selecting the options available under menu. Since the System is highly modularized, each part of the system can be implemented whenever necessary.

**5.2 FUTURE ENHANCEMENT**

This **ENABLE SECURE AND EFFICIENT MULTIMEDIA CLOUD SERVER** can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when require in this application. There is flexibility in all the modules. Reusability is possible as and when require in this application. We can update it next version. Its cost is under the budget and make within given time period

**6. BIBLIOGRAPHY**

**BOOK REFERENCES**

1. Dino Esposito, “Programming Microsoft ASP.NET 3.5”, second edition, Feb 23, 2008**.**
2. Anne Boehm and Joel Murach, “Murach's ASP.NET 3.5 Web Programming with C# 2008”, Third ediction, Oct 10, 2008.
3. George Shepherd, “Microsoft ASP.NET 3.5: Step by Step”, Second edition, Feb 23, 2008.
4. “**Pro SQL Server 2005”,** Robin Dewson, Apress Publisher.
5. “Inside the SQL Server Query Optimize”, Benjamin Nevarez, Red gate books , 2011.

**WEB REFERENCES**

1. **www.net-tutorials.com** – (Secure cloud server, multimedia hosting, cloud security, web server setup)
2. **www.tutorialspoint.com/asp.net** – (ASP.NET cloud security, multimedia handling in ASP.NET, efficient media storage)
3. **www.asp.net-tutorials.com** – *(*ASP.NET Web API for multimedia, cloud security for ASP.NET, performance optimization)
4. **www.sqlmag.com** - (SQL Server cloud security, multimedia database management, performance optimization, cloud database security)

**APPENDIX**

**SAMPLE SCREENS**































