# SECURE SHARING OF MEDICAL HEALTH RECORD IN CLOUD

**By**

**M.AJITH KUMAR,**

**Register No: 820318621302**

of

**A.V.C COLLEGE OF ENGINEERING**

Mannampandal, Mayiladuthurai - 609305

**A PROJECT REPORT**

Submitted to the

**FACULTY OF**

**INFORMATION AND COMMUNICATION ENGINEERING**

*In the partial fulfilment of the requirements*

*for the award of the degree*

*of*

**MASTER OF COMPUTER APPLICATIONS**

MAY - 2021

# CERTIFICATE

# A.V.C COLLEGE OF ENGINEERING

**MAYILADUTHURAI, MANNAMPANDAL-609 305,**
**NAGAPATTINAM(DT), TAMIL NADU**
|Approved by AICTE & Affiliated by Anna University, Chennai|
|Re-Accredited by NAAC with 'B++' Grade (2nd Cycle)|
|An ISO 9001:2015 Certified Institution|

## Department of Computer Applications

(An Recognized Research Centre of Anna University, Chennai)

**BONAFIDE CERTIFICATE**

This is to Certified that this project report titled, **"SECURE SHARING OF MEDICAL HEALTH RECORD IN CLOUD"** is the bonafide work of **Mr.M.AJITH KUMAR (Reg.No.820318621302)** who carried out the project under my supervision.Certified further to the best of my knowledge the work reported here does not form part of any other report or dissertation on the basis of which a degree was conferred on an earier occasion this or any other candidate.

**SUPERVISOR** **DIRECTOR**

**Place:**

**Date:**

Submitted for the Anna University Examination held on:_____

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

# Fantasy Solution

*Find ur Passion...*

Date: 06-05-2021

To

The Director
Department of Computer Applications,
A.V.C.College of Engineering
Mannampandal, Mayiladuthurai.

Sir,

**Sub: MCA Final Year Project – Completion - Reg.**

This is to certify that **Ajithkumar. M** bearing **(Reg.No:820318621302)**, of **A.V.C.College of Engineering** has been completed his final year project work entitled as**" SECURE SHARING OF MEDICAL HEALTH RECORD IN CLOUD "** in our organization during the period of December 2020 to April 2021.

**Platform: PHP & MYSQL**


Authorized Signature
(With Seal)

#16, Samnath Plazza, 3rd Floor, Melapudur, Trichy - 620 001.
**Ph : 0431-4250535 | Mob : 90430 95535**
www.fantasysolution.in | info@fantasysolution.in

# DECLARATION

# DECLARATION

I, **M.AJITH KUMAR (Reg.No:820318621302),** student of **A.V.C COLLEGE OF ENGINEERING,** Department of Computer Applications, would like to declare that the project work entitled as **"SECURE SHARING OF MEDICAL HEALTH RECORD IN CLOUD"** is the result of the original work done by me during the course of studyand is sunmitted on the partial fulfillment for the award of the degree of **"MASTER OF COMPUTER APPLICATIONS"** of Anna University,Chennai.

**SIGNATURE**

**PLACE:Mannampandal**

**DATE:** **(M.AJITHKUMAR)**

# ABSTRACT

# SECURE SHARING OF MEDICAL HEALTH RECORD IN CLOUD


## ABSTRACT

In cloud secure personal data sharing is the important issues because it creates several securities and data confidentiality problem while accessing the cloud services. Many challenges present in personal data sharing such as data privacy protection, flexible data sharing, efficient authority delegation, computation efficiency optimization, are remaining toward achieving practical fine-grained access control in the Personal Information Sharing system. Personal records must be encrypted to protect privacy before outsourcing to the cloud. Aiming at solving the above challenges, here propose an efficient data sharing mechanism for Personal Data Sharing, which not only achieves data privacy, fine-grained access control and authority delegation simultaneously. Proposed methodology is presented to secure patients' MHR (Medical Health Record) in the healthcare cloud using the duplicate generation technique with a two server based computing facility. Duplicate server serves as a second gallery to contain duplicate MHR that appear to the attacker as if it is the original MHR. When user uploading a file on original server, corresponding duplicate file will be stored on another server. In this method, the decoy files are called when an attacker is detected as accessing the system, in our proposed methodology the duplicate files are retrieved from the beginning to ensure better security. In proposed approach RSA algorithm is implement to encrypt the medical records.

# ACKNOWLEDGEMENT

# ACKNOWLEDGEMENT

This project work itself is an acknowledgement to the intensity drive and technical competence of many individuals who have contributed to it. I express my thanks to everyone who contributed and guided me much for the successful completion of this simple effort.

I express my sincere thanks to **Dr. C. SUNDAR RAJ, M.E., Ph.D., Principal, and A.V.C College of Engineering** who helped me in providing required facility in completing the project**.**

I proudly thank our **Dr. *S.* SELVAMUTHUKUMARAN, MCA., Ph.*D.*, Vice principal and Director , Department of Computer Applications, and A.V.C College of Engineering** who allowed me to do my final project work independently and effectively

I am grateful to my internal guide **Dr.M.SENTHILURUGAN,MCA, PhD Professor, Department of Computer Applications, A.V.C College of Engineering ,** for his valuable guidance, idea, advice and encouragement for the successful completion of this project to identify our mistakes and also achieving our goal.

With heartly pleasure I thank **MS.N.VIJAYA BANU**,**MCA ,**Programmer ,Fantasy Soluctions (P) Ltd, My external project guide for guiding me throughout my entire progress.

Finally ,with a deep feeling of indepthness express my thanks to all faculty and staff members of our department for their guidance my beloved parents ,family members and friends who helped me throughout the project

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# LIST OF TABLES:

# LIST OF FIGURES:

# LIST OF ABBREVIATION:

| S.NO | ABBREVIATION | DESCRIPTION |
|---|---|---|
| 1 | **PHP** | Hypertext-Pre-processor |
| 2 | **MySQL** | My Structured Query Language |
| 3 | **GPL** | General Public License |
| 4 | **HTML** | Hyper Text Markup Language |
| 5 | **UML** | Unified Modeling Language |
| 6 | **MHR** | Medical Health Record |
| 7 | **RDBMS** | Relational Database Management System |
| 8 | **LAMP** | Linux, Apache,Mysql,Perl/PHP/Python |
| 9 | **HIE** | Health Information Exchange |
| 10 | **KAS** | Key Assignment Scheme |
| 11 | **ABE** | Attribute Based Encryption |

# CHAPTER 1

# INTRODUCTION

# 1. INTRODUCTION

The e-Healthcare information offers unique security, privacy and confidentiality challenges that require a fresh examination of the mainstream concepts and approaches to information security. The significance of security and privacy in eHealthcare information raised the issues of individual consent, confidentiality and privacy, which are the main determinants in adopting and successful utilising the e-Healthcare information. Current trends in the domain of e-Healthcare information management point to the need for comprehensive incorporation of security, privacy and confidentiality safeguards within the review of e-Healthcare information management frameworks and approaches. This raises major challenges that demands holistic approaches spanning a wide variety of legal, ethical, psychological, information and security engineering. The e-healthcare information is varied and complex in nature. It is collected, maintained and utilised by a variety of players within the healthcare profession as well as in other sectors, where it is required for purposes such as insurance, employment and research. The structure of healthcare is multi-dimensional as it can be viewed in time-oriented, source-oriented and clinical problem-oriented terms with further dimensions being possible. In practice, health information is scattered across and within organisations and countries. The period for utilising health information spans over a lifetime of an individual. There may be a statutory time period from the death of a person after whose expiry the deceased's healthcare information may be destroyed. The destruction of health information by a controller of such information is a legally regulated process. A key aspect of the nature of healthcare information is that it is personal It appears that this approach is increasingly being discarded in some places, where it seems legal ownership of health information is bestowed on the patient while the healthcare unit is designated as a controller with legal rights, interests and obligations over the information. Thus, use of health information always requires the consent of the individual owner. In practice, there is a separation between ownership and control of health information, the owner of healthcare information may not be the one who controls its collection, storage and processing. Therefore, this necessitates distinction between owners, the controllers, processors and users of healthcare information. The later are governed by the laws on the protection of information to ensure the consent and preserve the owners' privacy and confidentiality.

# CHAPTER 2

# ORGANIZATION PROFILE

# CHAPTER-2

# 2.ORGANIZATION PROFILE

**Company profile:**

Fantasy solution as a leading IT solution and service provider, provides innovative information technology - enabled solutions and services to meet the demands arising from social transformation, shaping new life styles for individuals and creating values for the society. Focusing on software technology, Fantasy solution provides industry solutions and product engineering solutions, related software products & platforms, and services, through seamless integration of software and services, software and manufacturing, as well as technology and industrial management capacity.

Fantasy solution helps industry customers establish best practices in business development and management. The fantasy solution serves include real time projects, web designing, web hosting, software development and training etc, in many of which, has a leading market share. Notably, Fantasy Solution has participated in the formulation of many national IT standards and specifications.

Fantasy solution has the world's leading product engineering capabilities, ranging from consultation, design, R&D, and integration to testing of embedded software, in the fields of automotive electronics, smart devices, digital home products, and IT products. The software provided by fantasy solution runs in a number of globally renowned brands. Particularly offering the services that include application development & maintenance, ERP implementation & consulting, testing, performance engineering, software localization & globalization, IT infrastructures, BPO, IT education & training, etc. Sticking to its business philosophy and brand commitment

of “Beyond Technology”, fantasy solution is dedicated to providing innovative information technologies to drive the sustainable development of society, as well as becoming a company that is well recognized and respected by employees, share holders, customers, and society.

**Our services:**

In this ever-changing environment, keeping a competitive edge means being able to anticipate and respond quickly to changing business conditions. Fantasy solution is a global software development company providing IT solutions to enterprises worldwide. Combining proven expertise in technology, and an understanding of emerging business trends, Fantasy delivers a range of software development solutions that includes e-business solutions, computer telephony, enterprise applications, professional web site design and development, product engineering, Electronic Health Records, CMS Software’s, Payment Gateway solutions, Time and attendance tracking software’s, Debt collection software’s, Appointment Reminder Solutions, Medical Transcription Services etc. We study, design, develop, enhance, customize, implement, maintain and support various aspects of information technology.

**Mission :**

Fantasy Solutions’ mission includes:

Providing high quality software development services, professional consulting and development outsourcing that would improve our customers’ operations;

Making access to information easier and securer (Enterprise Business);

Improving communication and data exchange (Business to Business);

Providing our customers with a Value for Money and

Providing our employees with meaningful work and advancement opportunities.

**Vision:**

Fantasy Solutions is a leading IT company for Consulting Services and Deployment of best of breed Business Solutions to top tier domestic and international customers.

**Address:**

Company Name: Fantasy solution,

Melapudur, Trichy.

website: www.fantasysolution.in

Gmail: fantazysolution@gmail.com

Contact No: 9043095535

# CHAPTER 3

# SYSTEM CONFIGURATION

# 3. SYSTEM CONFIGURATION

## 3.1 HARDWARE CONFIGURATION

- CPU type : Intel Pentium 4
- Clock speed : 3.0 GHz
- Ram size : 512 MB
- Hard disk capacity : 40 GB
- Monitor type : 15 Inch color monitor
- Keyboard type : internet keyboard

## 3.2 SOFTWARE CONFIGURATION

- Operating system : Windows OS
- Front End : PHP
- Back end : MySQL server
- Tool : Macromedia Dreamweaver 8
- Application : Web Application

# CHAPTER 4

# SOFTWARE FEATURES

# 4. SOFTWARE FEATURES

## 4.1 FRONT END: PHP

PHP: Hypertext Preprocessor (the name is a recursive acronym) is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as standalone interpreter on most operating systems and computing platforms.

PHP was originally created by RasmusLerdorf in 1995 and has been in continuous development ever since. The main implementation of PHP is now produced by The PHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) because restrictions exist regarding the use of the term PHP. Hypertext refers to files linked together using hyperlinks, such as HTML (HyperText Markup Language) files. Preprocessing is executing instructions that modify the output. Below is a demonstration of the difference between HTML and PHP files.

**Accessing an HTML Page**

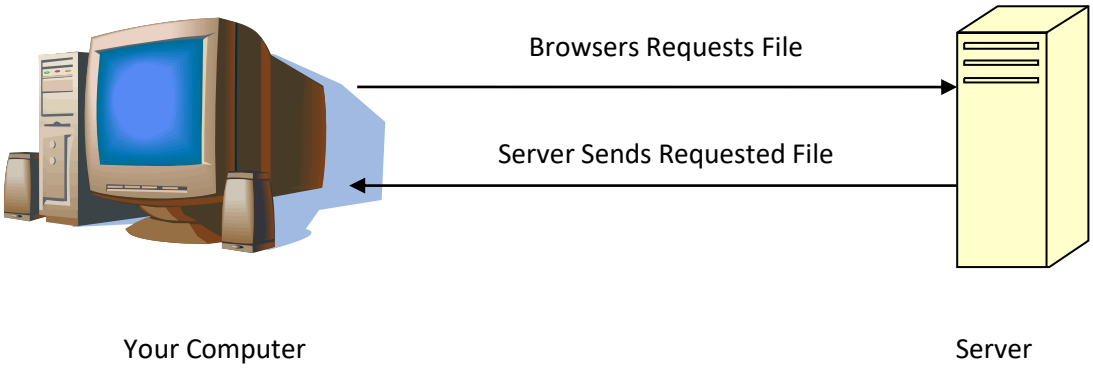



**Fig 4.1 Accessing an HTML page**

1. Your browser sends a request to that web page's server (computer) for the file (HTML or image) you wish to view.
2. The web server (computer) sends the file requested back to your computer.
3. Your browser displays the file appropriately.
4. If you request a PHP file (ends with ".php"), the server handles it differently.

**Accessing a PHP Page**

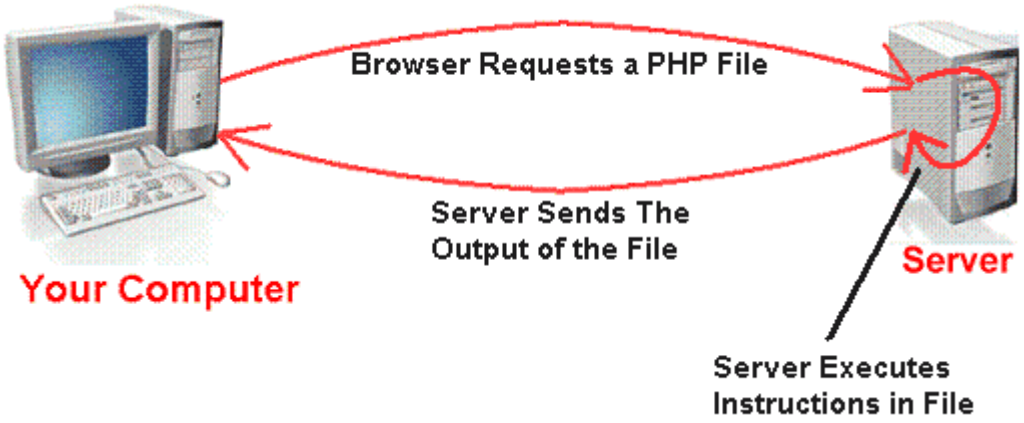



**Fig 4.2 Accessing a PHP Page**

1. Your browser sends a request to that web page's server for the PHP file you wish to view.
2. The web server calls PHP to interpret and perform the operations called for in the PHP script.
3. The web server sends the output of the PHP program back to your computer.
4. Your browser displays the output appropriately.

**Benefit of PHP**

Because the server does processing, the output of PHP files changes when its input changes. For example, most of the pages on the Horticulture site have only two (2) PHP commands:

1. Include the header file that defines the links on the left, the banner, and the quick links at the top.
2. Include the footer file that displays the mission statement and Horticulture contact information.

Because including the files is performed every time the PHP file is accessed, when the header/footer files change, the new content will be immediately updated. In other words, if you add a new link, every page that includes the header will immediately display the new link.

**Security**

About 30% of all vulnerabilities listed on the National Vulnerability Database are linked to PHP. These vulnerabilities are caused mostly by not following best practice programming rules: technical security flaws of the language itself or of its core libraries are not frequent (23 in 2008, about 1% of the total). Recognizing that programmers make mistakes, some languages include taint checking to detect automatically the lack of input validation which induces many issues. Such a feature is being developed for PHP, but its inclusion in a release has been rejected several times in the past. There are advanced protection patches such as Suhosin and Hardening-Patch, especially designed for Web hosting environments.

PHPIDS adds security to any PHP application to defend against intrusions. PHPIDS detects attacks based on cross-site scripting (XSS), SQL injection, header injection, directory traversal, remote file execution, remote file inclusion, and denial-of-service (DoS)

**Syntax**

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to control structures described in PHP code). The most common delimiters are `<?php` to open and `?>` to close PHP sections. `<script language="php">` and `</script>` delimiters are also available, as are the shortened forms `<?`or`<?=` (which is used to echo back a string or variable) and `?>` as well as ASP-style short forms `<%` or `<%=` and `%>`. While short delimiters are used, they make script files less portable as support for them can be disabled in the PHP configuration, and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

The first form of delimiters, `<?php` and `?>`, in XHTML and other XML documents, creates correctly formed XML 'processing instructions'. This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Variables are prefixed with a dollar symbol, and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted (`""`) and here-doc strings provide the ability to interpolate a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: `/* */` marks block and inline comments; `//` as well as `#` are used for one-line comments. The `echo` statement is one of several facilities PHP provides to output text, e.g., to a Web browser.

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. `if` conditions, `for` and `while` loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

**Data types**

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit signedinteger equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, and hexadecimal notations. Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is NULL. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources. Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with both keys and values,

and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, nowdoc or heredoc syntax.

**Functions**

PHP has hundreds of base functions and thousands more via extensions. These functions are well documented on the PHP site; however, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for thread programming, although it does support multi-process programming on POSIX systems.

## 4.2 BACK END: MySQL

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google(though not for searches), ImagebookTwitter, Flickr, Nokia.com, and YouTube.

## RDBMS Terminology

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

- **Database** − A database is a collection of tables, with related data.
- **Table** − A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column** − One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row** − A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy** − Storing data twice, redundantly to make the system faster.
- **Primary Key** − A primary key is unique. A key value cannot occur twice in one table. With a key, you can only find one row.
- **Foreign Key** − A foreign key is the linking pin between two tables.
- **Compound Key** − A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index** − An index in a database resembles an index at the back of a book.
- **Referential Integrity** − Referential Integrity makes sure that a foreign key value always points to an existing row.

## MySQL Database

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

- MySQL is released under an open-source license. So you have nothing to pay to use it.

- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**Features**

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licences. They have a common code base and include the following features:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- Information schema
- Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)
- X/Open XAdistributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine

- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Transactions with the InnoDB, and Cluster storage engines; savepoints with InnoDB
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
- Full-text indexing and searching using MyISAM engine
- Embedded database library
- Unicode support (however prior to 5.5.3 UTF-8 and UCS-2 encoded strings are limited to the BMP, in 5.5.3 and later use utf8mb4 for full unicode support)
- ACID compliance when using transaction capable storage engines (InnoDB and Cluster)
- Partititoned tables with pruning of partitions in optimiser
- Shared-nothing clustering through MySQL Cluster
- Hot backup (via mysqlhotcopy) under certain conditions
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application

**HTML**

HTML is a markup language for describing web documents (web pages).

- Hyper is the opposite of linear. It used to be that computer programs had to move in a linear fashion. This before this, this before this, and so on. HTML does not hold to that pattern and allows the person viewing the World Wide Web page to go anywhere, any time they want.
- Text is what you will use. Real, honest to goodness English letters.
- Mark up is what you will do. You will write in plain English and then mark up what you wrote. More to come on that in the next Primer.

- Language because they needed something that started with “ L ” to finish HTML and Hypertext Markup Louie didn’t flow correctly. Because it’s a language, really but the language is plain English.

HTML remains for Hyper Text Markup Language. It is a basic content designing dialect used to make hypertext records. It is a stage free dialect not at all like most other programming dialect. HTML is impartial and can be utilized on numerous stage or desktop. It is this component of HTML that makes it mainstream as standard on the WWW.

This adaptable dialect permits the making of hypertext connections, otherwise called hyperlinks. These hyperlinks can be utilized to unite reports on diverse machine, on the same system or on an alternate system, or can even indicate purpose of content in the same record.

HTML is utilized for making archives where the accentuation is on the presence of the record. It is likewise utilized for DTP. The records made utilizing HTML can have content with diverse sizes, weights and hues. It can also contain graphics to make the document more effective.

# CHAPTER 5

# PROJECT DESCRIPTION

# 5. PROJECT DESCRIPTION

## 5.1 OVERVIEW OF THE PROJECT

Many regions in the developed countries are establishing health information exchanges (HIEs), which are cloud-based information clearing houses where information can be more easily shared between hospitals, health systems, physicians, and clinics. There are many technology vendors and service providers, who are already building cloud-based HIEs, many of which are already functioning and providing tremendous value to patients, administrative authorities, and providers. Many pharmaceutical companies are starting to tap the cloud to improve research and drug development to discover newer, cheaper, and more effective treatment protocols and medicines. Hospitals and physicians are starting to see cloud-based medical records and medical image archiving services.

## 5.2 PROBLEM STATEMENT

The e-healthcare information is varied and complex in nature. It is collected, maintained and utilized by a variety of players within the healthcare profession as well as in other sectors, where it is required for purposes such as insurance, employment and research. The period for utilizing health information spans over a lifetime of an individual. There may be a statutory time period from the death of a person after whose expiry the deceased's healthcare information may be destroyed. The destruction of health information by a controller of such information is a legally regulated process. A key aspect of the nature of healthcare information is that it is personal. It is recognized that health information belongs to the individual who is the subject of such information. It appears that this approach is increasingly being discarded in some places, where it seems legal ownership of health information is bestowed on the patient while the healthcare unit is designated as a controller with legal rights, interests and obligations over the information. Thus, use of health information always requires the consent of the individual owner. In practice, there is a separation between ownership and control of health information, the owner of healthcare information may not be the one who controls its collection, storage and processing. Therefore, this necessitates distinction between owners, the controllers, processors and users of healthcare information. The later are governed by the laws on the protection of information to ensure the consent and preserve the owners' privacy and confidentiality.

## 5.3 MODULE DESCRIPTION

- Medical Cloud Framework
- Upload Medical Files
- Data Encryption
- Duplicate Storage
- File Access

**Medical Cloud Framework**

There is a significant volume of healthcare data generated daily. The data are important and vital for decision making and delivering the best care for patients. Cloud computing is a cost effective method that facilitates real-time data collection, data storage and exchange between healthcare organizations. Cloud infrastructure is characterized with a high throughput and a high volume storage; two important factors for efficient data analysis of large patients' population. Security and privacy are of the major concerns for using cloud-based healthcare services. Healthcare organization should have electronic medical records in order to use the cloud infrastructure. In order to cope with the rapid advancements in information technology and the utilization of cloud based services, efforts should be dedicated to move healthcare data form the traditional paper based to the electronic format. Then, regional legislation and policies should be enacted to regulate and control the usage of healthcare data.

**Upload Medical Files**

Cloud computing allows data collection and transfer to healthcare organizations. Data are collected from hospitals in the form of patient details, doctor details, medical reports and prescription details and then transmitted wirelessly to healthcare external processing units where patient's physician monitor and analyse those data. Administrator should maintain the doctor details and also enter the patient details. Then allocate the patients to the doctor based on their disease. Doctor should login and view the allocated patient details. Then doctor can add patient's medical reports and prescriptions for allocated patients. These medical details and files are securely stored in health care cloud.

**Data Encryption**

In this module, in order to make health data's more secure use multi party in cloud computing system. The data's encrypted with identity policy can be decrypted only if the identity policy is satisfied. Where the health data is encrypted using attributes and key policy. And the user with a particular attribute and key policy alone will be able to decrypt the health data after it is verified by "key distribution centre" and the "secure data distributor". This technique can be used in medical field for secure storage of patient details and limiting to particular doctor access. It's used to achieve fine–grained access control. A user can decrypt the ciphertext if only his attributes in the private key satisfy the access tree specified in the ciphertext. By doing so, the encryptor holds the ultimate authority about the encryption policy.

**Duplicate Storage**

This technique can be considered as an illusion technique, as it makes the attacker believe that he/she has accessed the user's original medical files while in fact it is just a duplicate file. Thus, both authorized and unauthorized users will be referred to the Duplicate Storage as the first step, while authorized legitimate users, as a second step, will be referred to the Original Cloud after being verified. We believe that by setting the default value of the as shown and the OMBD as hidden, we keep the original MBD more secure. Also, we believe that verifying that the user is legitimate is much easier than detecting the attacker, which is why we tried to deal with the attacker in the first place by offering the DMBD as the first step.

**File Access**

User profiling is a familiar technique that can be applied to model in what way, at what time, and how considerable users access their information in the healthcare cloud. This method of behavior-based security is commonly used in fraud detection application. In our proposed system, once the user accesses his/her account, by default the duplicate storage is shown. The legitimate user already knows that the gallery he/she accessed is not his/her original one, so would move on to the next step. Moving to the next step, the legitimate user can access his/her original medical files after being verified by passing the key verification. The key verification is the process of verifying the secret key by authorized user. Thus, if he/she passes the key

verification, that means he/she is the authorized user, so will be able to access the original medical files which are located on the cloud computing layer.

# CHAPTER 6

# SYSTEM ANALYSIS

# 6. SYSTEM ANALYSIS

## 6.1 EXISTING SYSTEM

The key assignment scheme (KAS) was first considered to achieve cryptographic access control. Designed a space-efficient key assignment scheme based on a binary tree and proposed a tree-based cryptographic access control mechanism. Existing system introduced the cryptographic hierarchical access control for dynamic structures and how to achieve access control in publicly verifiable outsourced computation based on KAS. This explored the relations between all security notions for hierarchical key assignment schemes. Proposed two hierarchical and shared key assignment schemes based on symmetric encryption and public key threshold broadcast encryption separately. Since the security classes in KAS should be set in advance, the access policy must follow the set security classes. Attribute based encryption (ABE) can achieve flexible access control over encrypted data. It was formalized two forms of ABE: key-policy ABE (KP-ABE) and ciphertext policy ABE (CP-ABE). In KP-ABE, the secret key is associated with an access policy, while an access policy is assigned to the ciphertext in CP-ABE. A user can decrypt a ciphertext if the set of attributes satisfies the access policy. Role Based Access Control (RBAC) provides efficient access control mechanism for each participant in EMR system. Here access controls are assigned and key sharing based on users role.

## 6.2 DRAWBACKS

- Outsourced computation brings extra communication cost.
- Authority delegation problem.
- Does not suitable for large healthcare corporations.

## 6.3 PROPOSED SYSTEM

Proposed system adopt two different public cloud servers to achieve secure outsourced computation, such as outsourced key generation/encryption/re-encryption key generation/ decryption. Actually, one public cloud server (e.g., public cloud 2) is sufficient for outsourced decryption, but not enough for other operations, because all the secret will be exposed to the unique cloud server. The access control model consists of five entities: private key generator (PKG), public cloud 1, public cloud 2, data owners and data consumers. Proxy Re-encryption is used to re-encrypt the data before sending it to the data consumer. Here propose an efficient data sharing mechanism for Personal Data Sharing, which not only achieves data privacy, fine-grained access control and authority delegation simultaneously, but also optimizes the computation efficiency and is suitable for resource constrained servers. Most of the data consumers are honest, while few of them are corrupt and will leakage their secret keys in the collusion. On the contrary, PKG and data owner are assumed to be fully trusted. Besides, public cloud 1 and public cloud 2 cannot collude with each other. The non-collusive assumption is reasonable, because the client can demand that two cloud servers cannot reveal users' information by contract. In proposed work, PR-ABE (Attribute Based Encryption with Proxy Re-encryption) technique implements to provide secure encryption of medical data. To improve the access control, here partial key sharing scheme will be implement. Using this, data owner can send partial secret key for the requested user. This approach overcomes the key guessing attack in data retrieval process. Proposed system will be implementing using PHP as front end and SQL is for back end process. This approach has modules like Framework Creation, Medical files uploading, Data Encryption, duplicate Storage, File access and alert system. Input process has file storage and output was provide secure to medical files using two cloud.

## 6.4 ADVANTAGES

- Communication cost is small and fixed.
- Prevent public cloud servers from learning secret information.
- Improved computation efficiency for PKG and USER.
- No collusion between two clouds.

# CHAPTER 7

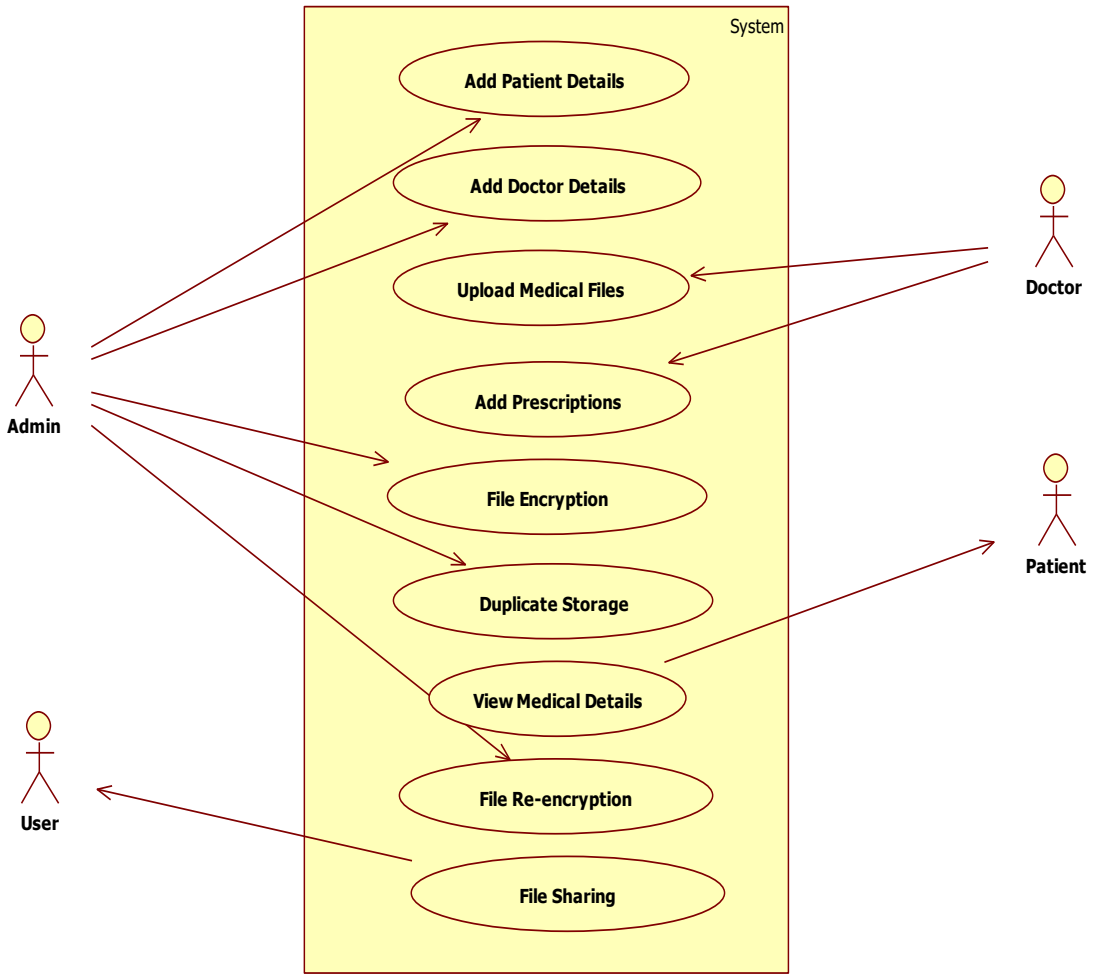
# SYSTEM DESIGN

# 7. SYSTEM DESIGN

## 7.1 UML DIAGRAM

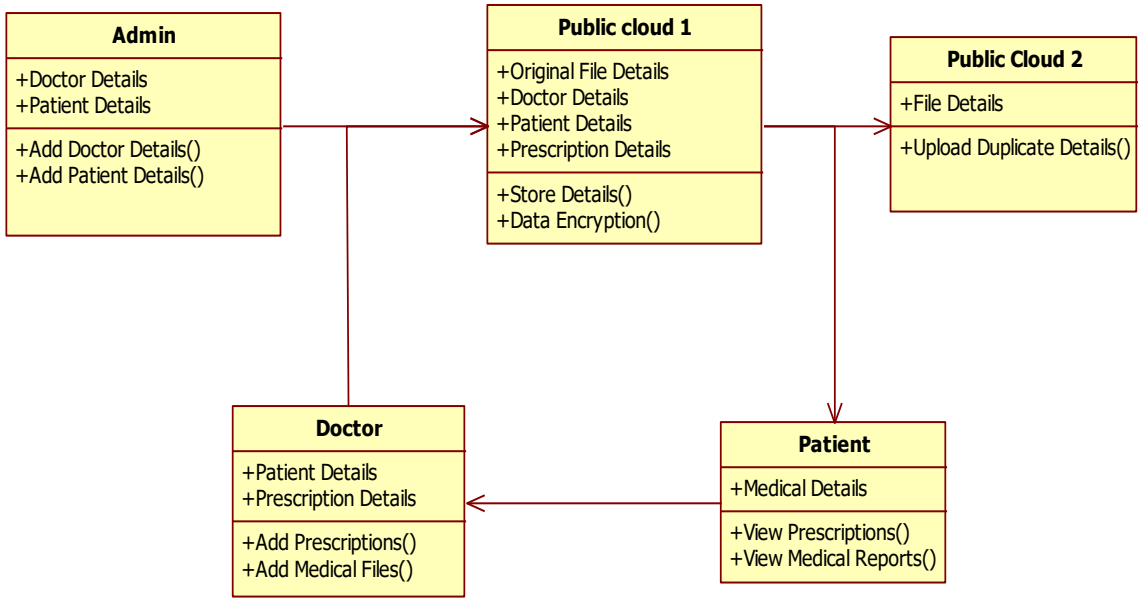
### 7.1.1 Use Case Diagram

A use case is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.



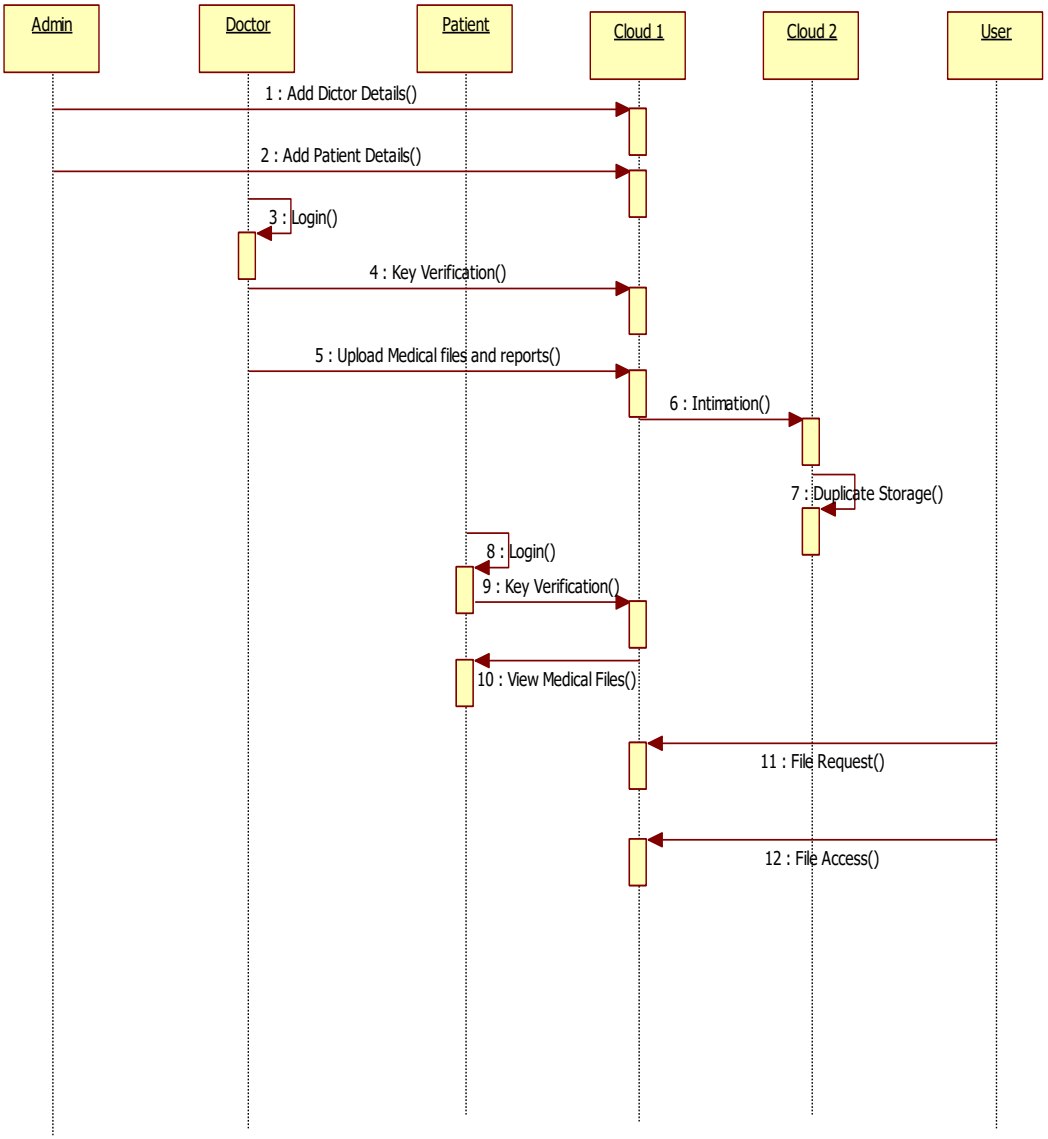
### 7.1.2 Class Diagram

In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.



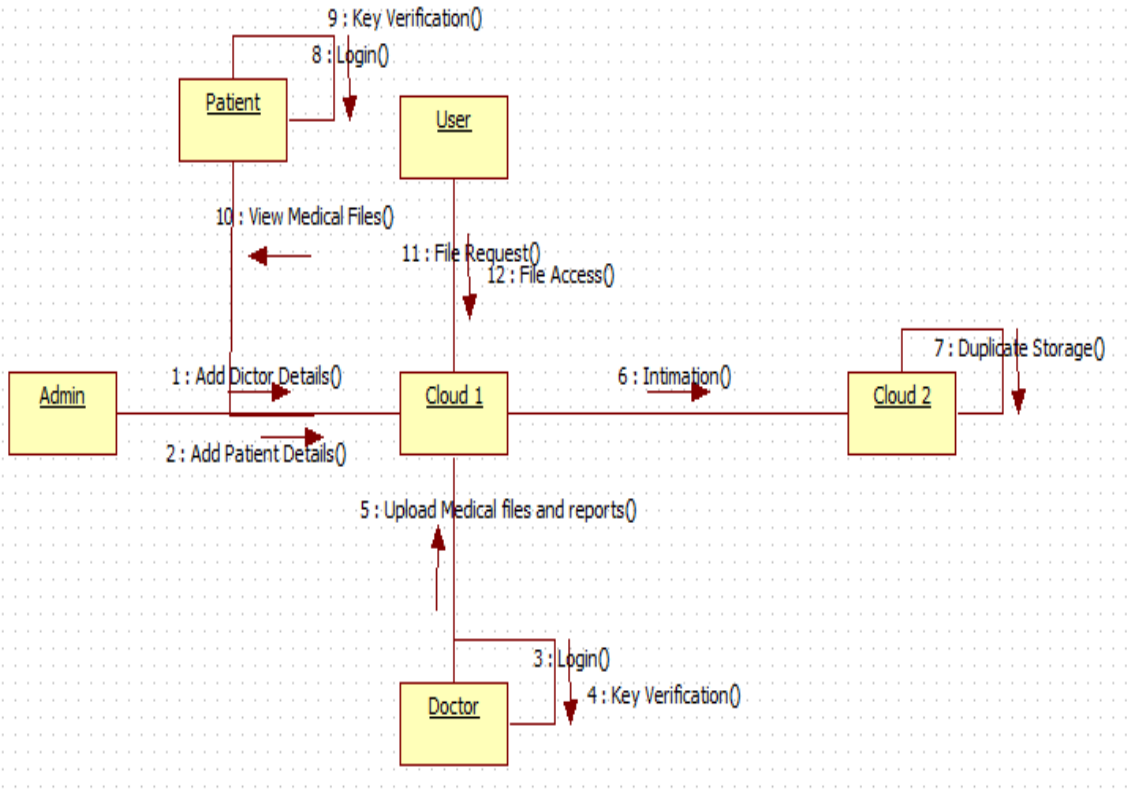
### 7.1.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. Sequence diagram is sometimes called event trace diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines, different processes that live simultaneously and horizontal arrows. The messages exchanged between them.



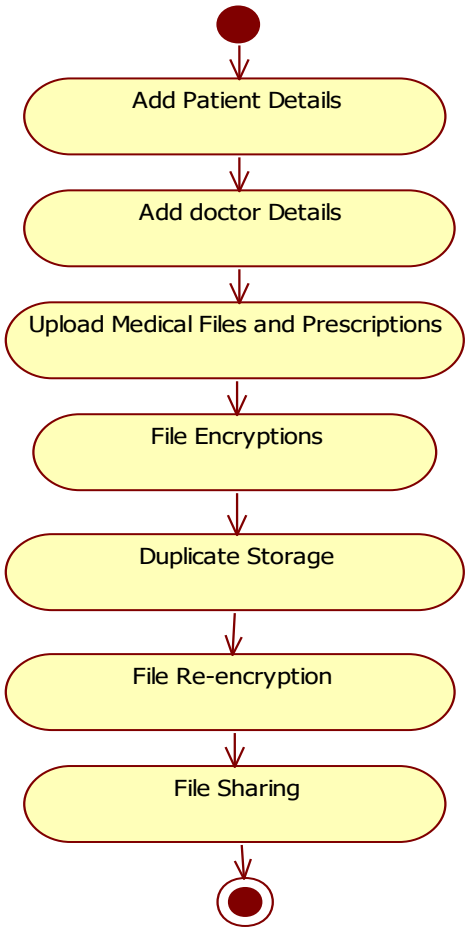
### 7.1.4 Collaboration Diagram

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.



### 7.1.5 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and action with support for choice, iteration and concurrency. Fig 6.6 shows the activities processed in proposed work. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes Activity diagrams show the overall flow of control.



## 7.2 DATA BASE DESIGN:

A table is a data structure that organizes information into rows and columns. It can be used to both store and display data in a structured format. For example, databases store data in tables so that information can be quickly accessed from specific rows. Websites often use tables to display multiple rows of data on page. Spreadsheets combine both purposes of a table by storing and displaying data in a structured format. Databases often contain multiple tables, with each one designed for a specific purpose. For example, a company database may contain separate tables for employees, clients, and suppliers. Each table may include its own set of fields, based on what data the table needs to store. In database tables, each field is considered a column, while each entry (or record), is considered a row. A specific value can be accessed from the table by requesting data from an individual column and row.

### 7.2.1 TABLE NAME: ADMIN

**Description:** This table contains login information of doctor and patient

| **Field** | **Type** | **Constraints** |
|---|---|---|
| username | varchar(20) | NOT NULL |
| password | varchar(20) | NOT NULL |

### 7.2.2 TABLE NAME: DOCTOR REGISTER

**Description:**This table contains information of doctor

| **Field** | **Type** | **Constraints** |
|---|---|---|
| id | int(20) | PRIMARY KEY |
| name | varchar(30) | NOT NULL |
| age | int(20) | NOT NULL |
| gender | varchar(30) | NOT NULL |
| qualification | varchar(30) | NOT NULL |
| specialization | varchar(20) | NOT NULL |
| experience | varchar(20) | NOT NULL |
| hosname | varchar(30) | NOT NULL |
| avtime | time(30) | NOT NULL |
| dname | varchar(20) | NOT NULL |
| password | varchar(20) | NOT NULL |
| address | varchar(50) | NOT NULL |
| city | varchar(20) | NOT NULL |
| state | varchar(20) | NOT NULL |
| pnumber | int(10) | NOT NULL |
| email | varchar(30) | NOT NULL |
| dkey | varchar(50) | NOT NULL |
| status | varchar(10) | NOT NULL |

### 7.2.3 TABLE NAME: USER REGISTER

**Description:**This table contains user information

| **Field** | **Type** | **Constraints** |
|---|---|---|
| id | int(20) | PRIMARY KEY |
| name | varchar(30) | NOT NULL |
| dob | varchar(20) | NOT NULL |
| age | int(20) | NOT NULL |
| address | varchar(50) | NOT NULL |
| pnumber | int(10) | NOT NULL |
| email | varchar(30) | NOT NULL |
| pname | varchar(20) | NOT NULL |
| password | varchar(20) | NOT NULL |
| acckey | int(50) | NOT NULL |
| hsname | varchar(50) | NOT NULL |
| status | varchar(10) | NOT NULL |

### 7.2.4 TABLE NAME:FILE

**Description:** This table contains information about user files

| Field | Type | Constraints |
|---|---|---|
| id | int(20) | PRIMARY KEY |
| docid | varchar(20) | NOT NULL |
| uid | varchar(20) | NOT NULL |
| file | varchar(50) | NOT NULL |
| details | varchar(100) | NOT NULL |
| tab1 | varchar(50) | NOT NULL |
| key1 | varchar(20) | NOT NULL |
| key2 | varchar(20) | NOT NULL |
| status | varchar(20) | NOT NULL |
| status1 | varchar(20) | NOT NULL |

### 7.2.5 TABLE NAME:ATTACK

**Description:** This table contains unauthorized user information

| Field | Type | Constraints |
|---|---|---|
| uname | varchar(20) | NOT NULL |
| password | int(20) | NOT NULL |
| date | date | NOT NULL |
| time | time | NOT NULL |

# CHAPTER 8

# SYSTEM TESTING

# 8. SYSTEM TESTING

## 8.1 TEST PLAN

Testing Level Specific Test Plans**:** Plans for each level of testing.

- Unit Test Plan
- Integration Test Plan
- System Test Plan

## 8.2 TEST CASE

A test case has components that describe input, action and an expected response, in order to determine if a feature of an application is working correctly. A test case is a set of instructions on "HOW" to validate a particular test objective/target, which when followed will tell us if the expected behavior of the system is satisfied or not.

Characteristics of a good test case:

- Accurate: Exacts the purpose.
- Economical: No unnecessary steps or words.
- Traceable: Capable of being traced to requirements.
- Repeatable: Can be used to perform the test over and over.
- Reusable: Can be reused if necessary.

| S.NO | SCENARIO | INPUT | EXPECTED OUTPUT | ACTUAL OUTPUT |
|---|---|---|---|---|
| 1 | Admin Add Details | Admin add Patient and Doctor Details | Register Successfully | Register Successfully |
| 2 | Doctor Login | Doctor's Username Password and Key Verification | If correct directed to doctor home page otherwise show "Invalid Login" | If correct directed to doctor home page otherwise show "Invalid Login" |

| | | | | |
|---|---|---|---|---|
| 3 | File Upload | Doctor upload patient File in cloud | File uploaded successfully | File uploaded successfully |
| 4 | Patient Register | Patient's Username Password and Key Verification | If correct directed to doctor home page otherwise show "Invalid Login" | If correct directed to doctor home page otherwise show "Invalid Login" |
| 5 | File Request | Send File Request | Request Send | Request Send |
| 6 | Accept Request | Owner Accept Request | Key Sharing | Key Sharing |

## 8.3 BUG REPORTS

Bug report providing any important information uncovered by the tests accomplished and including assessments of the quality to the testing effort, the quality of the software under test and statistics derived from incident reports

| **Bug id** | **Test module** | **Details** | **Bug status** |
|---|---|---|---|
| B01 | User module | User can entered invalid username and password. It shows incorrect user name and password | Success |

# CHAPTER 9

# CONCLUSION

# 9. CONCLUSION

In this project proposed a new mechanism is proposed to protect the healthcare data in the cloud. This system has a double layer protection in which the MHRs are stored in the cloud. Encryption/ Decryption will be done in one layer and in the other layer, duplicate files will be created and stored. To this end, two cloud storages are generated for different purpose. The original medical files are kept secretly in the cloud and the duplicate cloud is used as a duplicate file storage. Therefore, instead of retrieving the duplicate medical files only when any unauthorized access is discovered, the user, by default, accesses the duplicate files in cloud 2. The original server is only accessible by a user after verifying the authenticity of the user. Thus, the original multimedia data become more secure by setting the default value of the duplicate storage, while the original medical files are kept in a secure hidden cloud.

# CHAPTER 10

# FUTURE ENHANCEMENT

# 10. FUTURE ENHANCEMENT

In future work, we can extend the framework to implement the system with various encryption algorithms and also other cryptographic approaches in real time images and medical videos. Implement Stenography based approach to hide the Medical Data inside the Medical Image or other Images to provide secure sharing.

# APPENDICES

# APPENDICES

## APPENDICES 1: SOURCE CODE

**Dbconnect.php**

```
<?php

$connect=mysql_connect("localhost","root","");

mysql_select_db("record1",$connect);

?>

\----------------------------------
```

**Doctor details**

```
<?Php

include("dbconnect.php");

session_start();

extract($_POST);

if(isset($_POST['btn']))

{

//$qrr=mysql_query("select * from hospitel where hospitelname='$hosname'");

//$tr=mysql_fetch_array($qrr);

//$acckey=$tr['akey'];

$uk=$name;
```

```
 $sk=md5($uk);

 $s1=rand(200000,800000);

$q=mysql_query("select max(id) from docregister");

                        $r=mysql_fetch_array($q);

                        $id=$r['max(id)']+1;

$qry=mysql_query("insert into
docregister(id,name,age,gender,qualification,specialization,experience,hosname,avtime,dname,p
assword,address,city,state,pnumber,email,dkey,status)values('$id','$name','$age','$gender','$quali'
,'$special','$experience','$hosname','$avtime','$name','$s1','$address','$city','$state','$pnumber','$e
mail','$acckey','0')");

if($qry)

{

$msg='User Name:'.$name.' Password:'.$s1;

$to=$email;

include("email.php");

$objEmail      =      new CI_Email();

                                                                $objEmail-
>from('fantest.mail@gmail.com', "File");

                                                                //$objEmail-
>from('cloudservice@projectone.in', "Cloud Service");

                                                                $objEmail->to("$email");

                                                                //$objEmail->cc($txt_cc);

                                                                //$objEmail->bcc($txt_bcc);
```

```
                                        $objEmail->subject("Login
UserName And Password");

                                        $objEmail-
>message("$msg");

                                        if(file_exists($filename))
                                                {
                                                        $objEmail-
>attach($filename);
                                                }
                                                if ($objEmail-
>send())
                                                {

        //$qrt=mysql_query("update file set status='2' where id='$fid'");
$qrt;
                                                echo 'mail sent
successfully';

        //header("location:view.php");
                                                }
                                                else
```

```
{
echo 'failed';
}

header("location:doctor.php");
}
else
{
echo "enter valide data";
}
}?>
```

=----------

**Patient register**

```
<?php
include("dbconnect.php");
session_start();
extract($_POST);
if(isset($_POST['btn']))
{
//$qrr=mysql_query("select * from hospitel where hospitelname='$hsname'");
```

```
//$tr=mysql_fetch_array($qrr);

//$acckey=$tr['akey'];

$uk=$name;

 $sk=md5($uk);

 $s1=rand(10000,20000);

 $q=mysql_query("select max(id) from userregister");

                        $r=mysql_fetch_array($q);

                        $id=$r['max(id)']+1;

                        $acckey=rand(10000,20000);

$qry=mysql_query("insert into
userregister(id,name,dob,age,address,pnumber,email,pname,password,acckey,hsname,status)valu
es('$id','$name','$dob','$age','$address','$pnumber','$email','$name','$s1','$acckey','$hsname','0')")
;

if($qry)

{

$msg='User Name:'.$name.' Password:'.$s1;

$to=$email;

include("email.php");

$objEmail       =       new CI_Email();

                                                                $objEmail-
>from('fantest.mail@gmail.com', "File");

                                                                //$objEmail-
>from('cloudservice@projectone.in', "Cloud Service");
```

```
$objEmail->to("$email");
//$objEmail->cc($txt_cc);
//$objEmail->bcc($txt_bcc);
$objEmail->subject("Login UserName And Password");

$objEmail->message("$msg");

if(file_exists($filename))
    {
        $objEmail->attach($filename);
    }
    if ($objEmail->send())
    {

    //$qrt=mysql_query("update file set status='2' where id='$fid'");
$qrt;
        echo 'mail sent successfully';
```

```
        //header("location:view.php");
                                                                    }
                                                                    else
                                                                    {
                                                                    echo 'failed';
                                                                    }


header("location:patient.php");
}
else
{
echo "";
}
}
?>
```

**View.php**

```
<form action="" method="post">
                                    <div align="center" class="style2">
                                      <p>Patient  Details</p>
```

```
                    </div>
                     <table width="80%" border="1" align="center">
<tr>
  <td height="57"><div align="center" class="style1">Sl.No</div></td>
  <td><div align="center" class="style1">Patient Name</div></td>
  <td><div align="center" class="style1">Address</div></td>
  <td><div align="center" class="style1">Phone Number</div></td>
  <td><div align="center" class="style1">Email Id </div></td>
</tr>
                              <?php
                              $i=1;

                              $qry=mysql_query("select * from userregister");
                              while($row=mysql_fetch_array($qry))
                              {
                              ?>
<tr>
  <td height="64"><div align="center"><?php echo $i;?></div></td>
  <td><div align="center"><?php echo $row['name'];?></div></td>
  <td><div align="center"><?php echo $row['address'];?></div></td>
  <td><div align="center"><?php echo $row['pnumber'];?></div></td>
```

```
  <td><div align="center"><?php echo $row['email'];?></div></td>
 </tr>
                                <?php
                                $i++;
                                }
                                ?>
</table>
                        <div align="center" class="style2">
                          <p>Doctor  Details</p>
                    </div>
                        <table width="90%" border="1" align="center">
 <tr>
   <td height="57"><div align="center" class="style1">Sl.No</div></td>
   <td><div align="center" class="style1">Hospital Name</div></td>
   <td><div align="center" class="style1">Doctor Name </div></td>
   <td><div align="center" class="style1">Specialization</div></td>
   <td><div align="center" class="style1">Address</div></td>
   <td><div align="center" class="style1">Phone Number </div></td>
   <td><div align="center" class="style1">Email Id </div></td>
 </tr>
 <?php
```

```
                    $i=1;


                    $qry=mysql_query("select * from docregister");
                    while($row=mysql_fetch_array($qry))
                    {
                    ?>
  <tr>
    <td height="64"><div align="center"><?php echo $i;?></div></td>
    <td><div align="center"><?php echo $row['hosname'];?></div></td>
    <td><div align="center"><?php echo $row['name'];?></div></td>
    <td><div align="center"><?php echo $row['specialization'];?></div></td>
    <td><div align="center"><?php echo $row['address'];?></div></td>
    <td><div align="center"><?php echo $row['pnumber'];?></div></td>
    <td><div align="center"><?php echo $row['email'];?></div></td>
  </tr>
  <?php
                    $i++;
                    }
                    ?>
</table>
                <p> </p>
```

```
</form>
```

---

**KGC**

**Login.php**

```
<?php
include("dbconnect.php");
session_start();
extract($_POST);
if(isset($_POST['btn']))
{
$qry=mysql_query("select * from admin where username='$uname' &&
password='$password'");
$num=mysql_num_rows($qry);
//echo $num;
        if($num==1)
{
?>
<script language="javascript">
alert("Login successfully..");
```

```
window.location.href="admin.php";
</script>
<?php
//header("location:admin.php");
}
else
{
?>
<script language="javascript">
alert("UserName And Password Wrong..");
window.location.href="index.php";
</script>
<?php


}
}
?>
```

**Sent.php**

```
<?php
include("dbconnect.php");
```

```
session_start();
extract($_POST);
echo $pid=$_REQUEST['id'];
$qry=mysql_query("select * from userregister where id='$pid'");
$row=mysql_fetch_array($qry);
echo $email=$row['email'];
$acckey=rand(10000,56296);
$msg='Login Key:'.$acckey;
$to=$email;
include("email.php");
$objEmail = new CI_Email();
                                                    $objEmail-
>from('fantest.mail@gmail.com', "Key");
                                                    //$objEmail-
>from('cloudservice@projectone.in', "Cloud Service");
                                                    $objEmail->to("$email");
                                                    //$objEmail->cc($txt_cc);
                                                    //$objEmail->bcc($txt_bcc);
                                                    $objEmail->subject("Login
Key");
```

```
                                                    $objEmail-
>message("$msg");


                                                    if(file_exists($filename))
                                                        {
                                                            $objEmail-
>attach($filename);
                                                        }
                                                        if ($objEmail-
>send())
                                                        {

        $qrt=mysql_query("update userregister set acckey='$acckey',status='1' where id='$pid'");
$qrt;
?>
<script language="javascript">
alert("Login Successfully..");
window.location.href="admin.php";
</script>
<?php
                                                        echo 'mail sent
successfully';
```

```
        //header("location:view.php");
                                                                    }
                                                                    else
                                                                    {
                                                                    echo 'failed';
                                                                    ?>
<script language="javascript">
alert("Login Unsuccessfully..");
//window.location.href="index.php";
</script>
<?php
                                                                    }


?>
```

---

**File upload**

```
<?php
include("dbconnect.php");
session_start();
```

```
extract($_POST);

$did=$_SESSION['did'];

$id=$_REQUEST['id'];

$qry=mysql_query("select * from userregister where id='$id'");

$r=mysql_fetch_array($qry);

if(isset($_POST['btn']))

{

$q=mysql_query("select max(id) from file");

                        $r=mysql_fetch_array($q);

                        $id=$r['max(id)']+1;

$pb_key=$_POST['public_key'];

$uploadDirectory = "file/"; //folder to save the encrypted file

$fileName = $_FILES['file']['name'];

$fname=$_FILES['file']['name'];

$tempFileName = $_FILES['file']['tmp_name'];

$error = $_FILES['file']['error'];

$fileContentType = $_FILES['file']['type'];

$fileSize = $_FILES['file']['size'];

$uk=$fname;

 $sk=md5($uk);

 $key1=substr($sk,0,8);
```

```
 $uk1=$dname;

 $sk1=md5($uk1);

$key2=substr($sk1,0,8);

//move_uploaded_file( $_FILES['fileToUpload']['tmp_name'],"manager/".$$fileName);

if($error==UPLOAD_ERR_OK){

$file = fopen($tempFileName,"r");

$content = fread($file,filesize($tempFileName));

//fwrite($content,$tempFileName);

$encryptedContent = base64_encode($content);

$encryptedFileSaveName=$uploadDirectory.$fileName;

$encryptedFile = fopen($encryptedFileSaveName,'w');


fwrite($encryptedFile,$encryptedContent);

fclose($encryptedFile);

//print("File has been upload and encrypted successfully.");

//include('1.php');

}

else{

//print("Error while uploading.....");

}
```

move_uploaded_file($_FILES['file']['tmp_name'],"upload/".$fname);

```
$qr=mysql_query("insert into
file(id,docid,uid,file,details,tab1,key1,key2,status,status1)values('$id','$did','$id','$fname','$detail
s','$tab1','$key1','$key2','0',")");

if($qr)

{

move_uploaded_file($_FILES['file']['tmp_name'],"upload/".$fname);

?>

<script language="javascript">

alert("File Upload Successfully..");

window.location.href="view.php";

</script>

<?php

}

else

{

echo "Not Appointment";



}

}?>
```

---

**Login.php**

```
<?php
include("dbconnect.php");
session_start();
extract($_POST);
$did=$_SESSION['did'];
$qry=mysql_query("select * from docregister where id='$did'");
$row=mysql_fetch_array($qry);
$email=$row['email'];
$acckey=rand(10000,56296);
$msg='Login Key:'.$acckey;
$to=$email;
include("email.php");
$objEmail = new CI_Email();
                                                    $objEmail-
>from('fantest.mail@gmail.com', "Key");
                                                    //$objEmail-
>from('cloudservice@projectone.in', "Cloud Service");
                                                    $objEmail->to("$email");
                                                    //$objEmail->cc($txt_cc);
                                                    //$objEmail->bcc($txt_bcc);
```

```
                                                    $objEmail->subject("Login
Key");


                                                    $objEmail-
>message("$msg");


                                                    if(file_exists($filename))
                                                        {
                                                            $objEmail-
>attach($filename);
                                                        }
                                                        if ($objEmail-
>send())
                                                        {

        $qrt=mysql_query("update docregister set dkey='$acckey' where id='$did'");
$qrt;
?>
<script language="javascript">
alert("Login Successfully..");
window.location.href="login1.php";
</script>
<?php
```

```
                                                    echo 'mail sent
successfully';

        //header("location:view.php");

                                                    }
                                                    else
                                                    {
                                                    echo 'failed';
                                                    ?>
<script language="javascript">
alert("Login Unsuccessfully..");
window.location.href="index.php";
</script>
<?php
                                                    }


?>
```

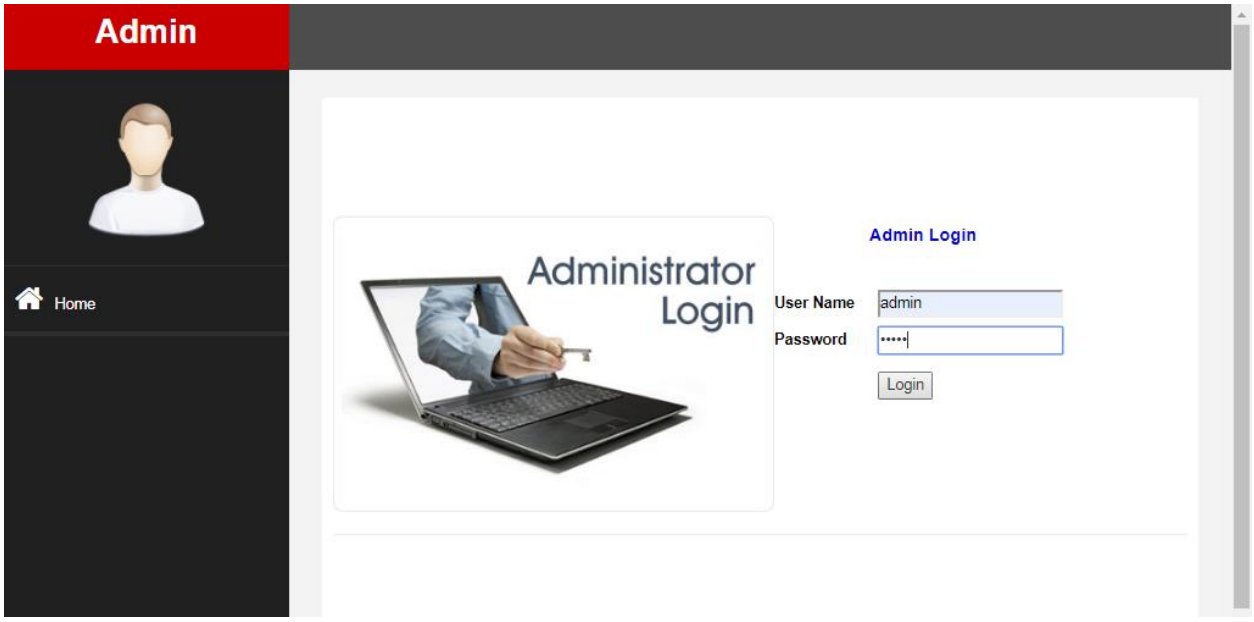
**Decrypt.php**

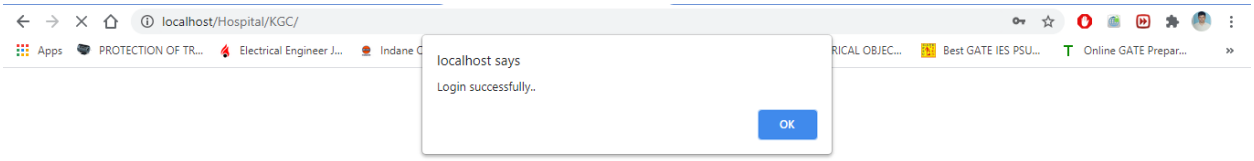
```
<?php
include('functions.php');
$src = 'file/simple.png';
```

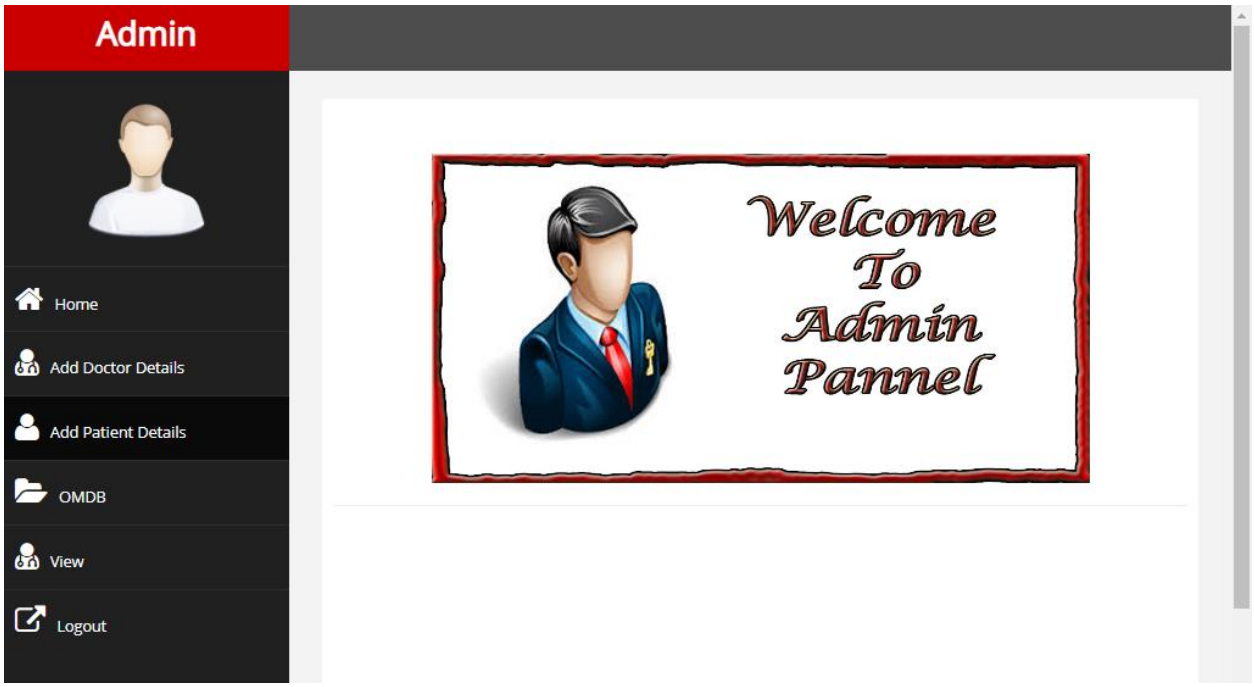
```
$im = imagecreatefrompng($src);
$real_message = '';
for($x=0;$x<100;$x++){
  $y = $x;
  $rgb = imagecolorat($im,$x,$y);
  $r = ($rgb >>16) & 0xFF;
  $g = ($rgb >>8) & 0xFF;
  $b = $rgb & 0xFF;

  $blue = toBin($b);
  $real_message .= $blue[strlen($blue)-1];
//echo $real_message = toString($real_message);
}
echo $real_message = toString($real_message);
 $real_message;
//die;
?>
```

# APPENDICES 2: SCREENSHOTS

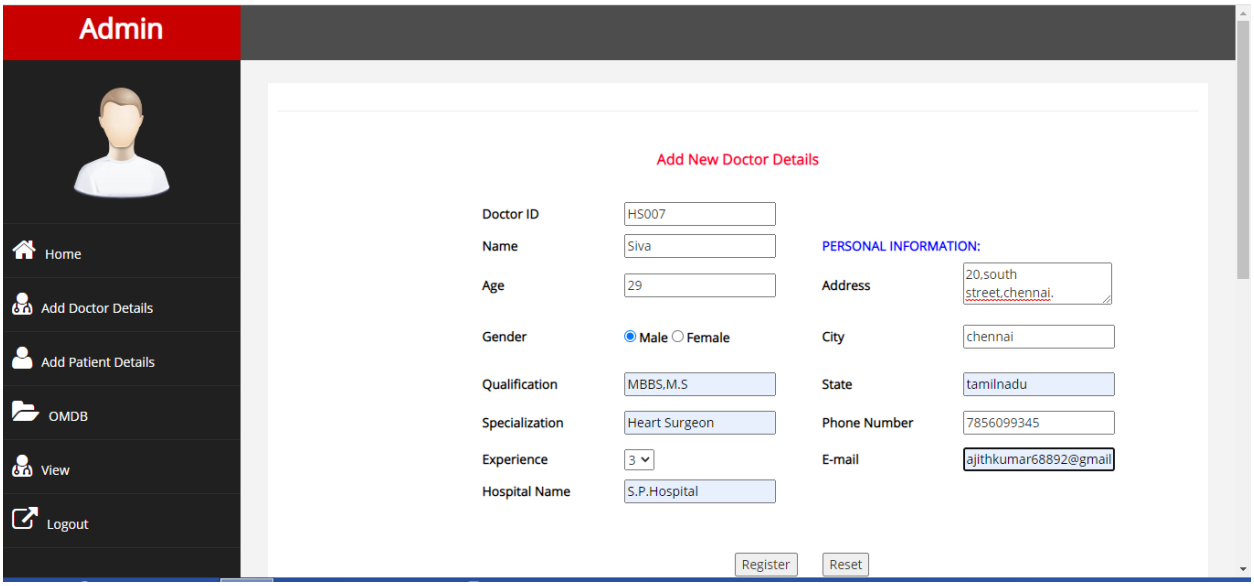
## ADMIN LOGIN:



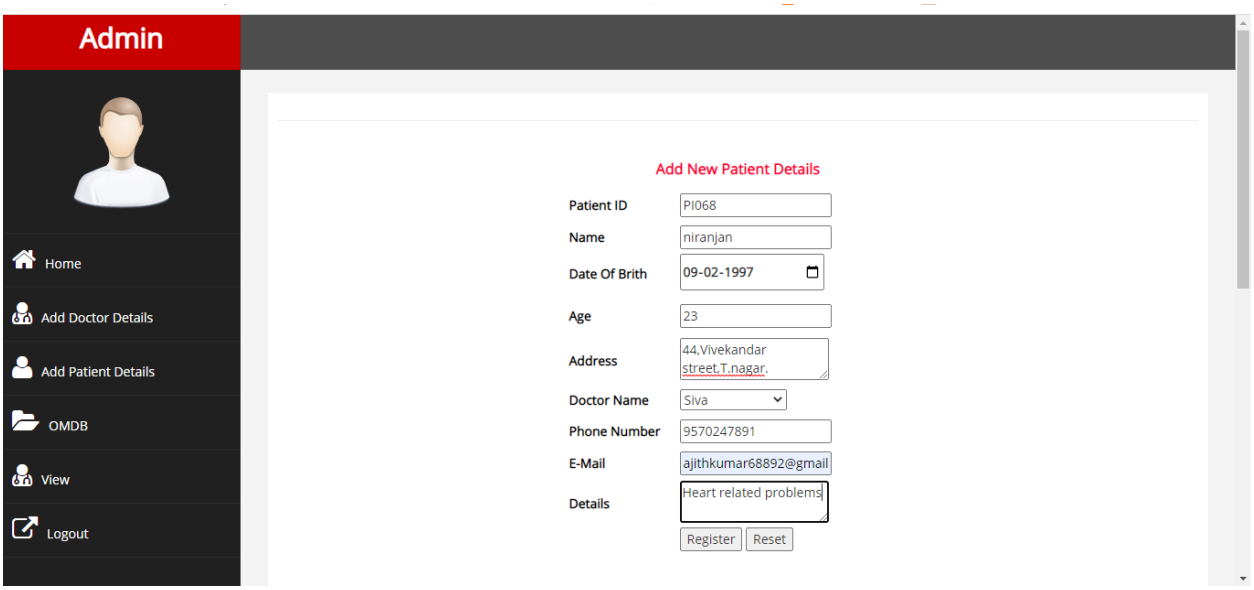





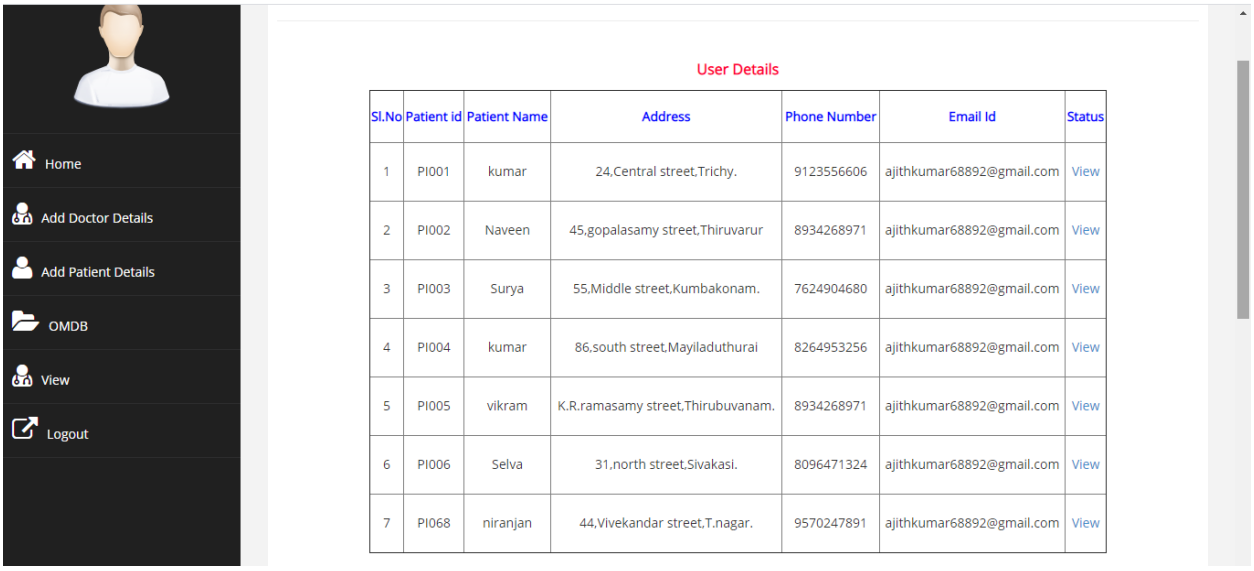

DOCTOR REGISTER:



PATIENT REGISTER:

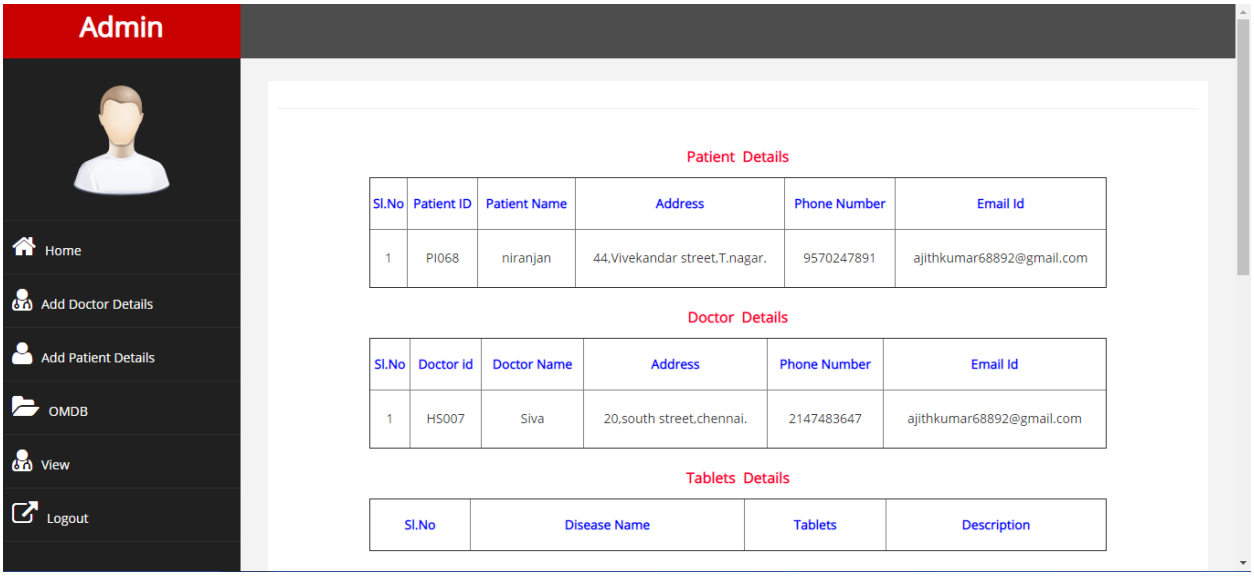



USER DETAILS:

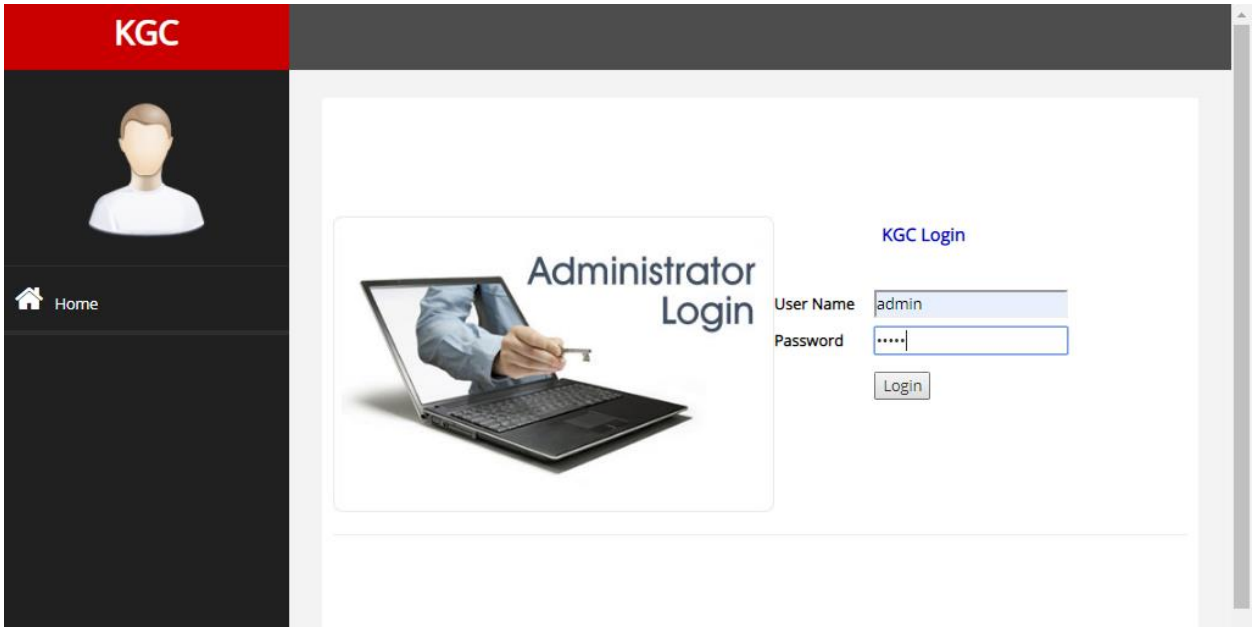



| Sl.No | Patient id | Patient Name | Address | Phone Number | Email Id | Status |
|---|---|---|---|---|---|---|
| 1 | PI001 | kumar | 24,Central street,Trichy. | 9123556606 | ajithkumar68892@gmail.com | View |
| 2 | PI002 | Naveen | 45,gopalasamy street,Thiruvarur | 8934268971 | ajithkumar68892@gmail.com | View |
| 3 | PI003 | Surya | 55,Middle street,Kumbakonam. | 7624904680 | ajithkumar68892@gmail.com | View |
| 4 | PI004 | kumar | 86,south street,Mayiladuthurai | 8264953256 | ajithkumar68892@gmail.com | View |
| 5 | PI005 | vikram | K.R.ramasamy street,Thirubuvanam. | 8934268971 | ajithkumar68892@gmail.com | View |
| 6 | PI006 | Selva | 31,north street,Sivakasi. | 8096471324 | ajithkumar68892@gmail.com | View |
| 7 | PI068 | niranjan | 44,Vivekandar street,T.nagar. | 9570247891 | ajithkumar68892@gmail.com | View |

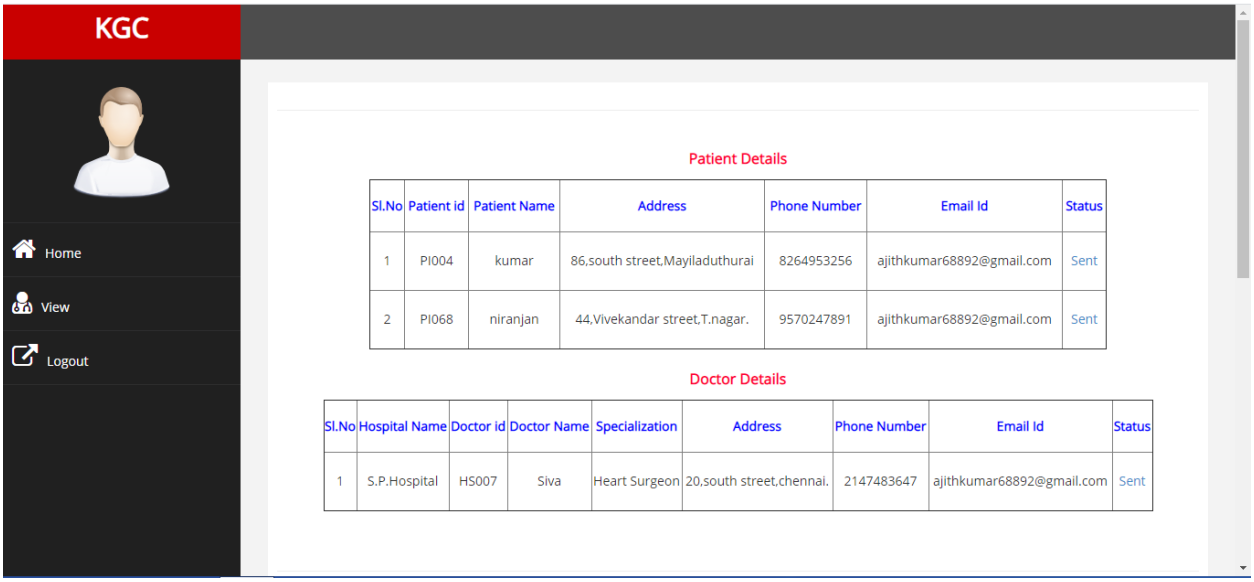
VIEW DETAILS:



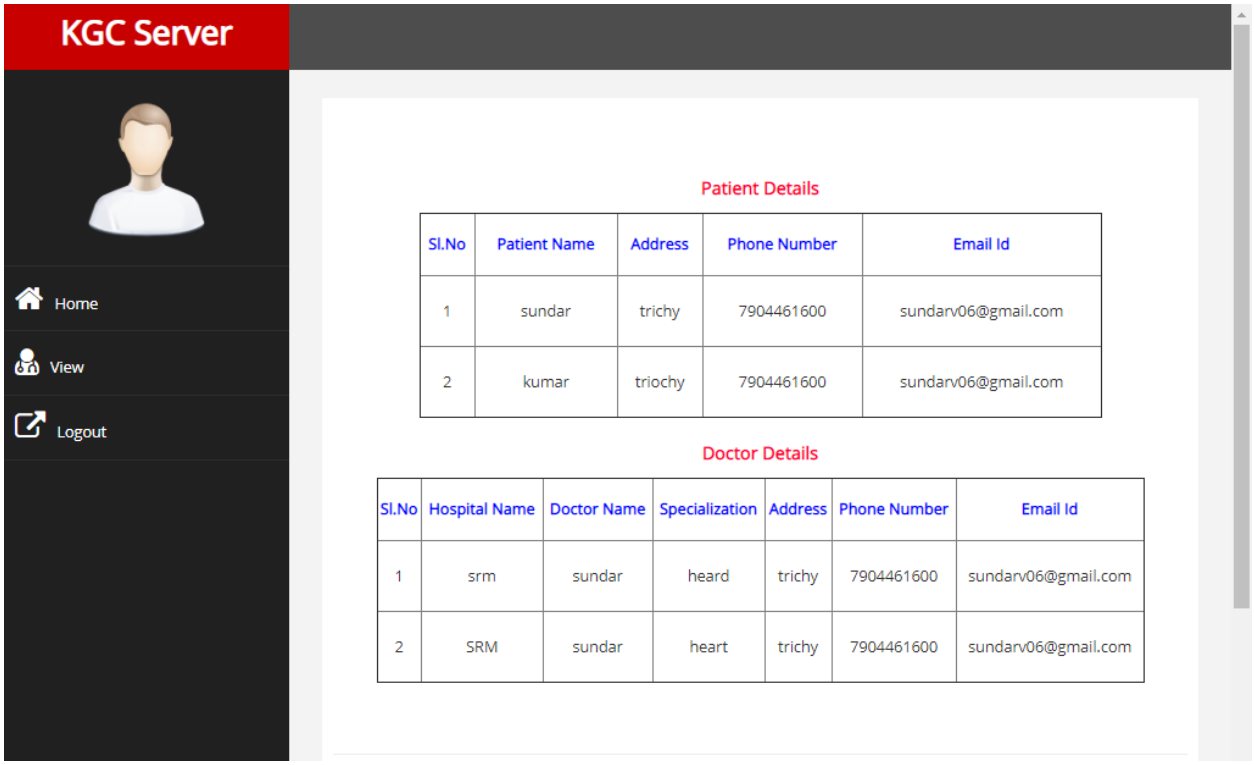

KGC LOGIN:



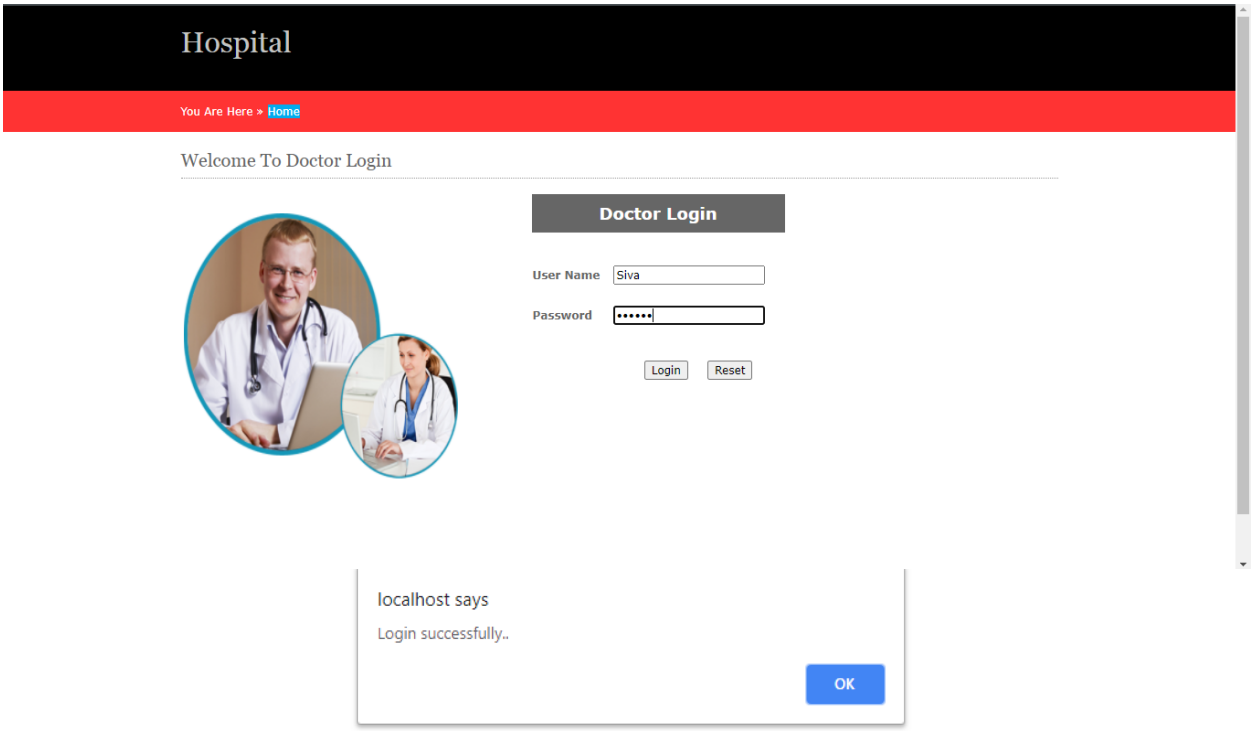

KGC VIEW:



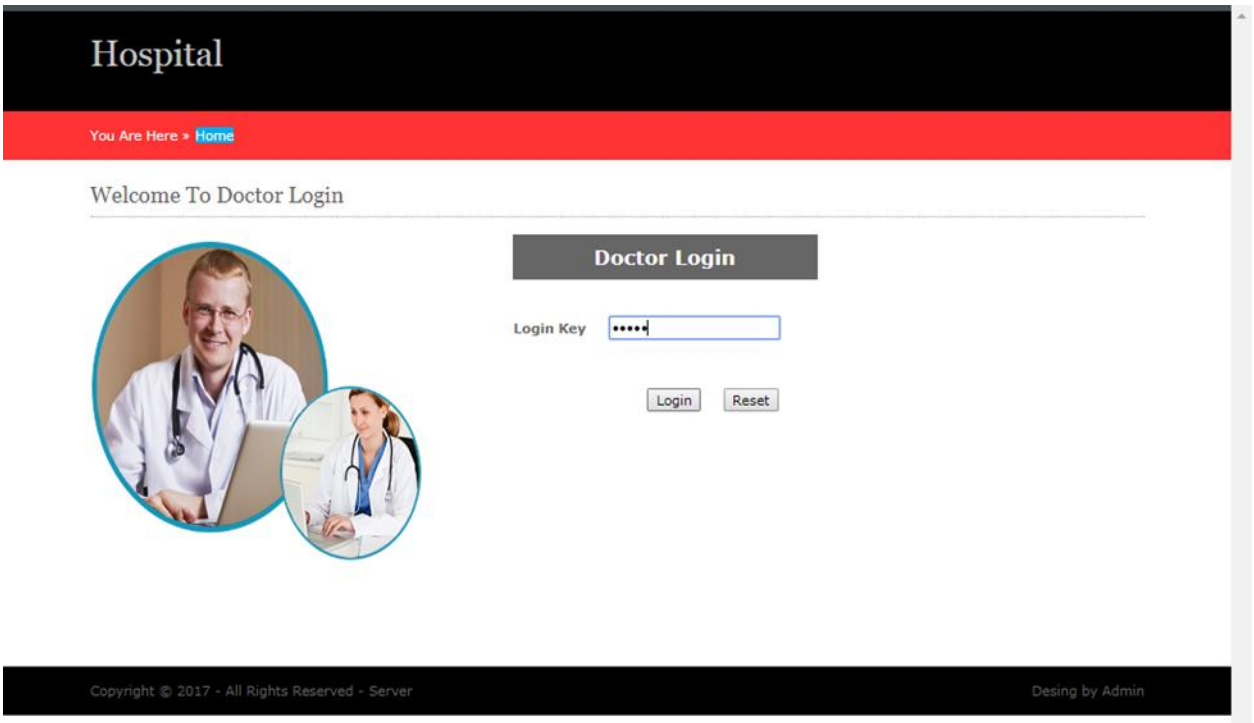

KGC SERVER:



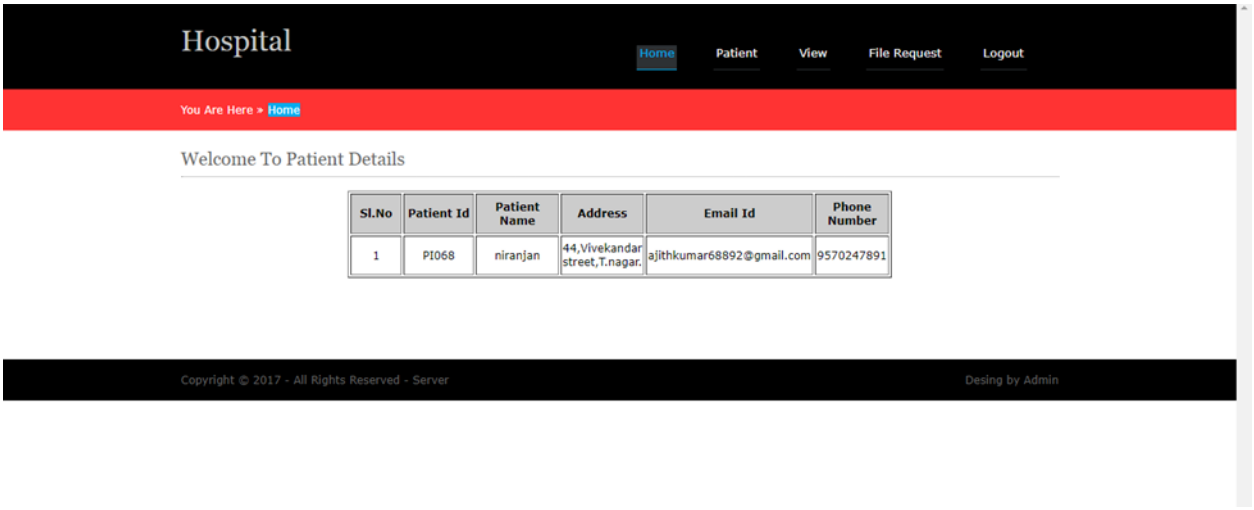
DOCTOR LOGIN:



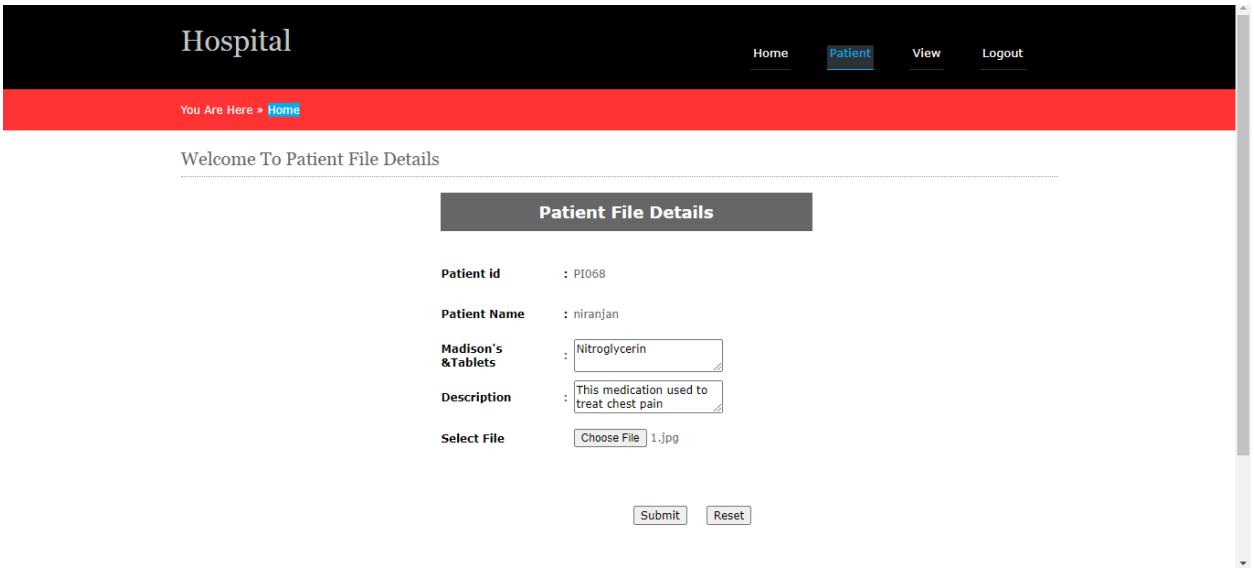

ACCESS KEY LOGIN:



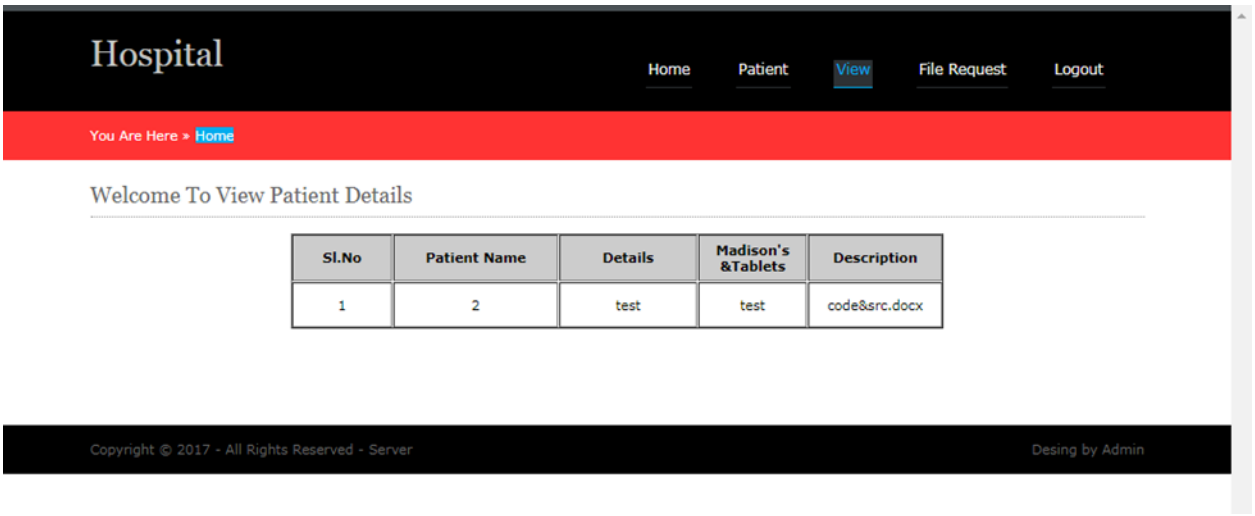

HOME PAGE:



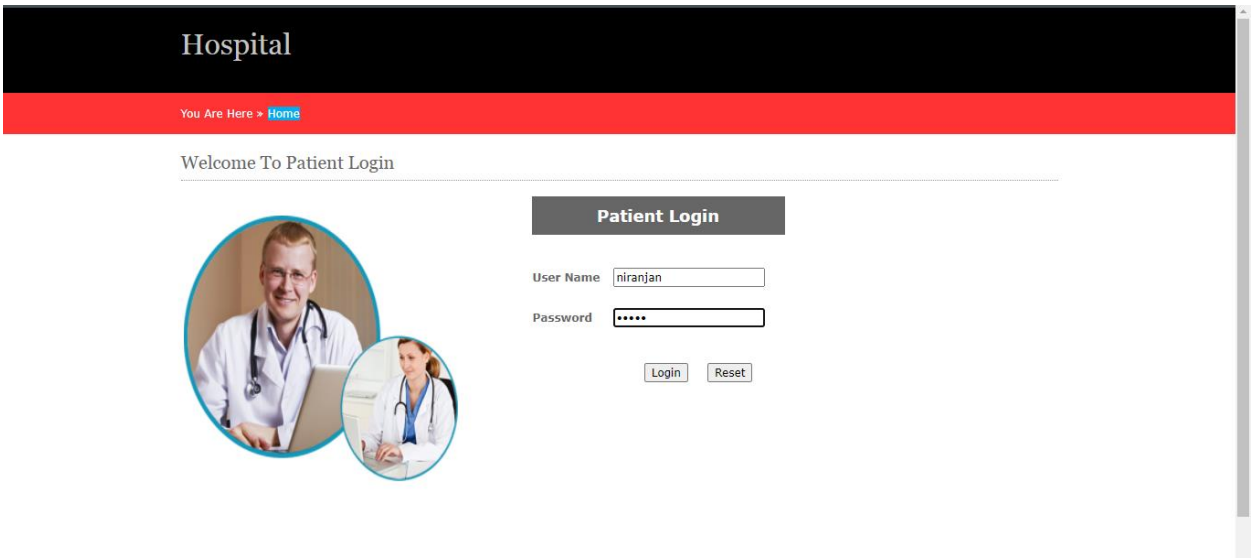

PATIENT FILE DETAILS:



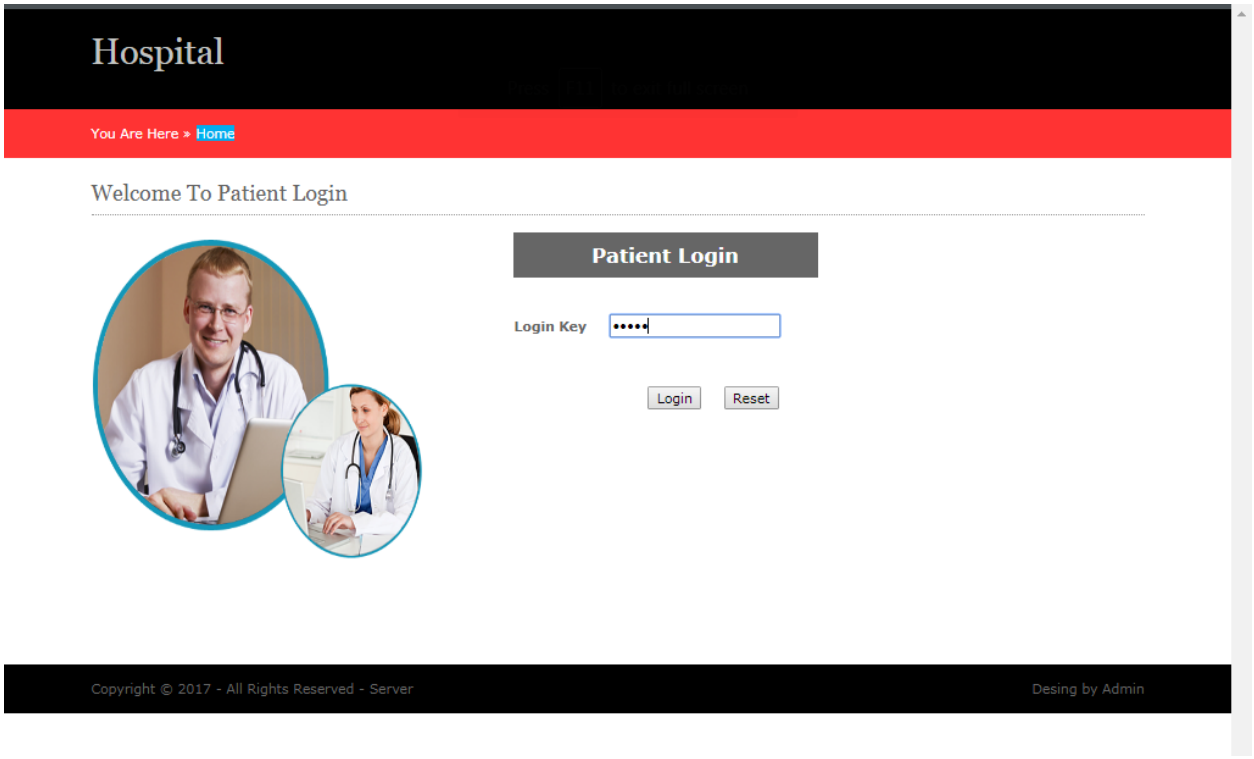

VIEW PATIENT DETAILS:




PATIENT LOGIN:

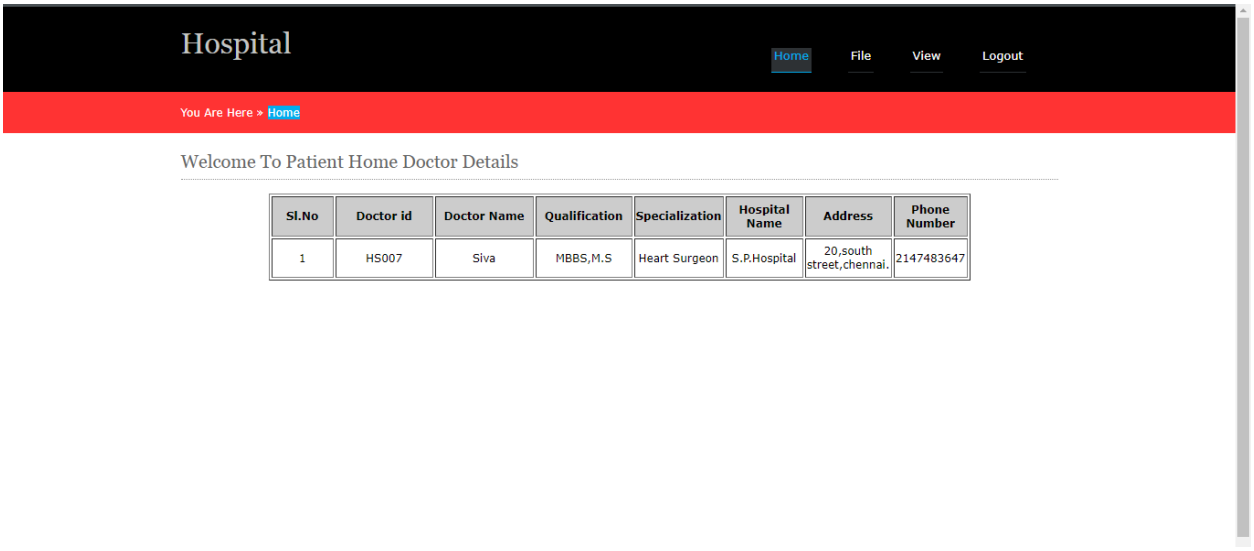


LOGIN KEY:



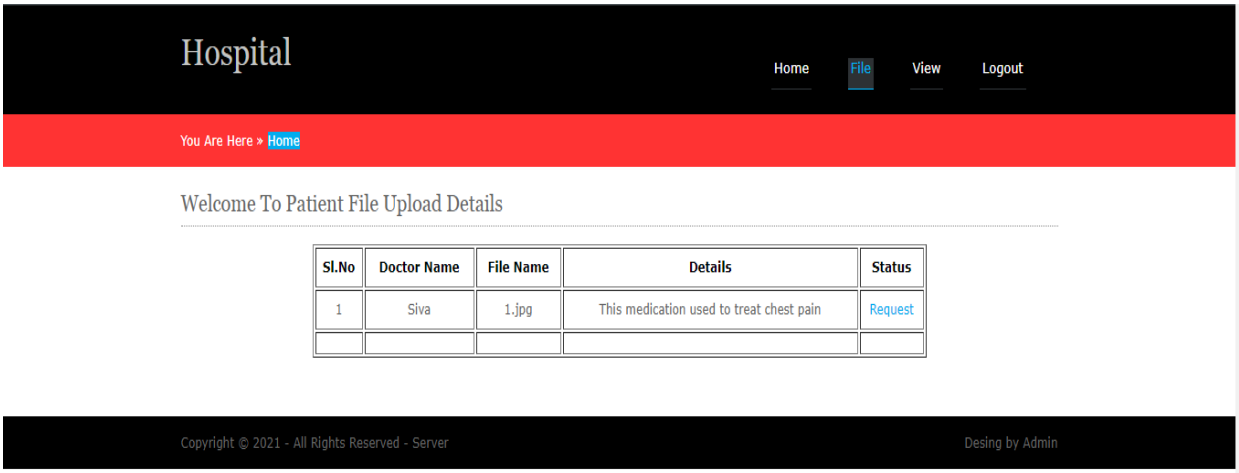


PATIENT HOME

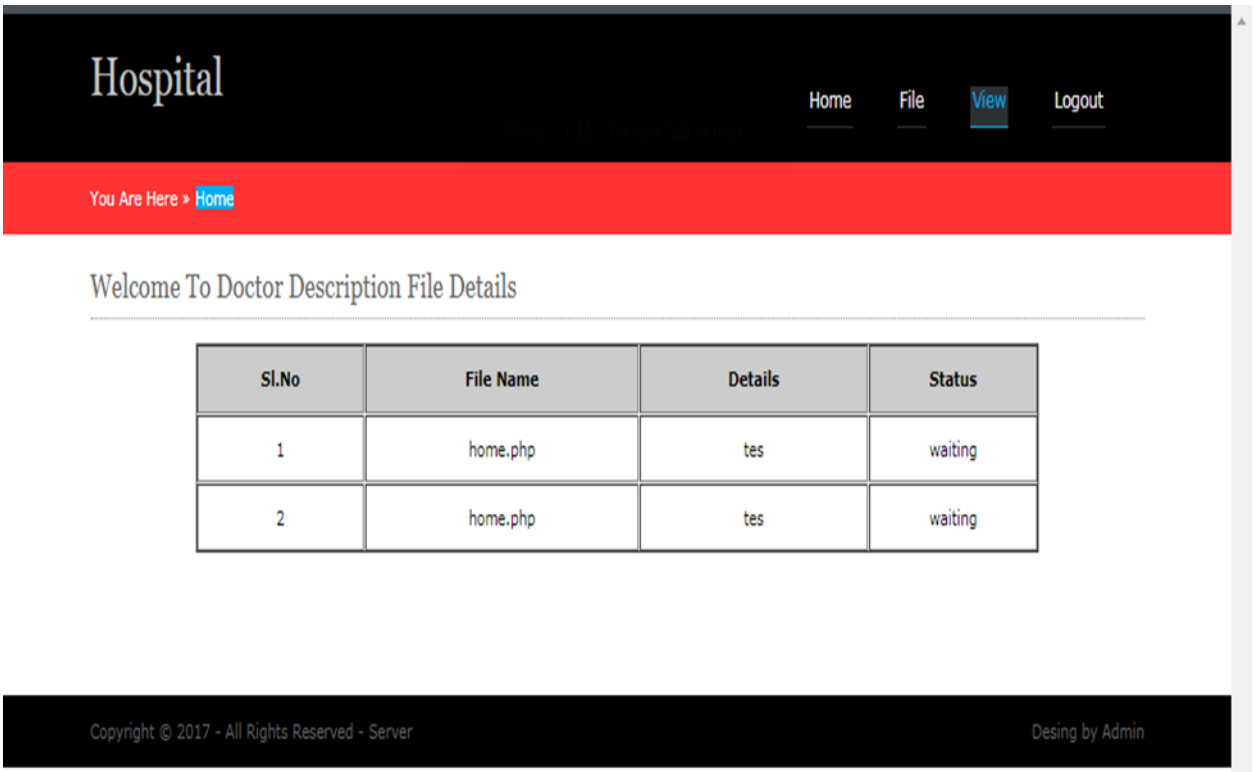



| Sl.No | Doctor id | Doctor Name | Qualification | Specialization | Hospital Name | Address | Phone Number |
|---|---|---|---|---|---|---|---|
| 1 | HS007 | Siva | MBBS,M.S | Heart Surgeon | S.P.Hospital | 20,south street,chennai. | 2147483647 |

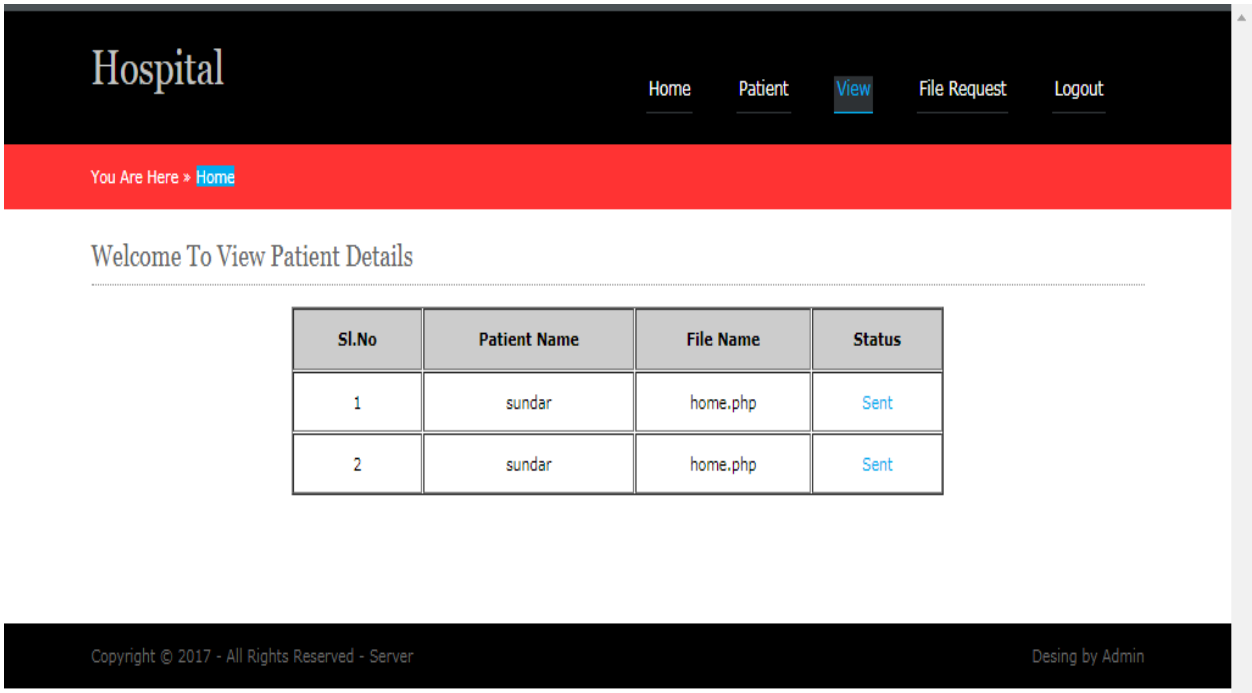
PATIENT FILE DETAILS:



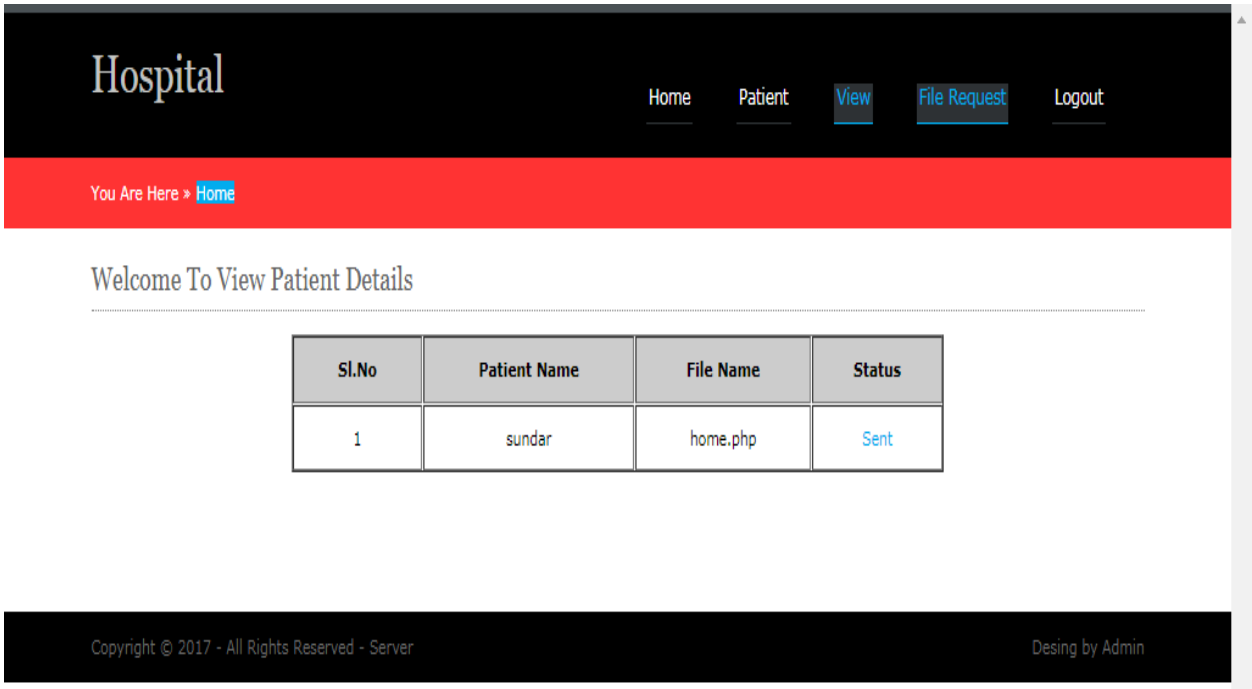

DOCTOR PRESCRIPTION DETAILS:



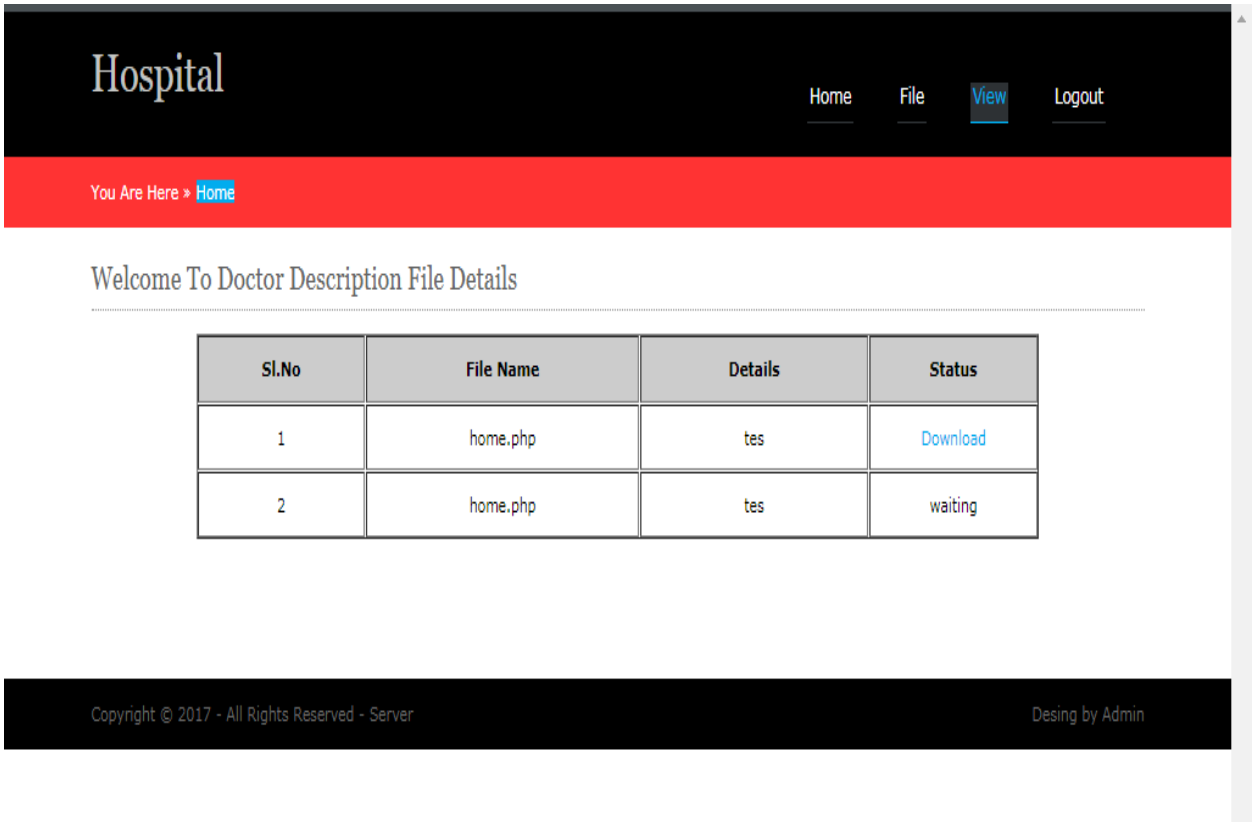
**VIEW PATIENT DETAILS:**



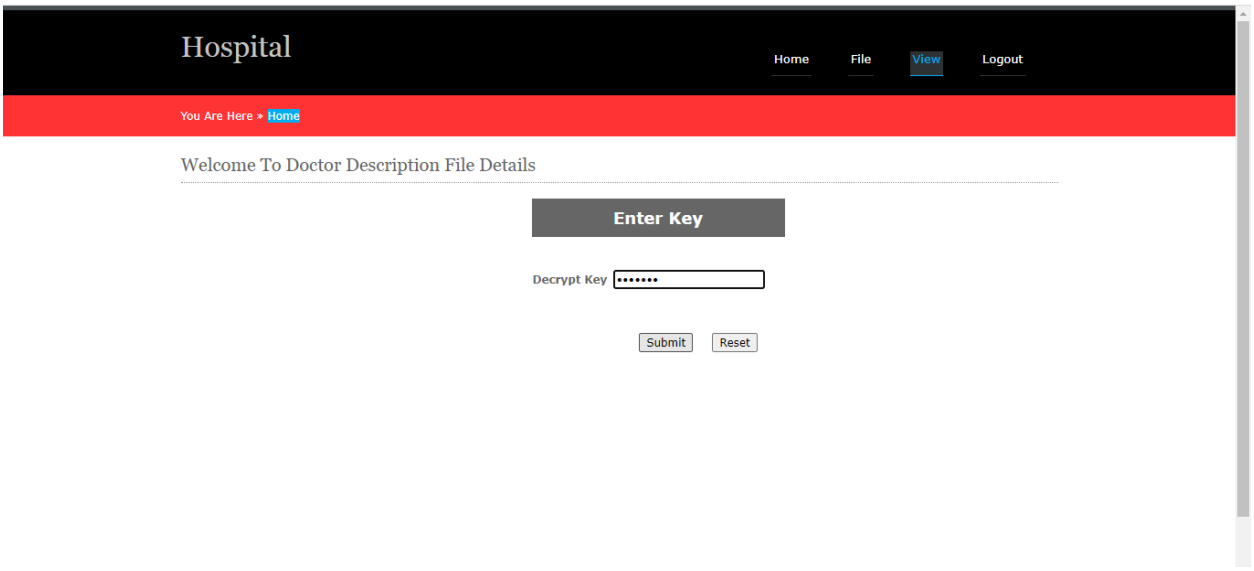

FILE REQUEST DETAILS:



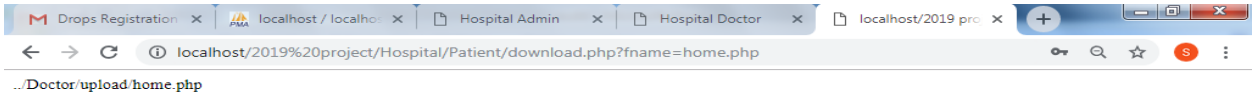
**DOWNLOAD FILE:**



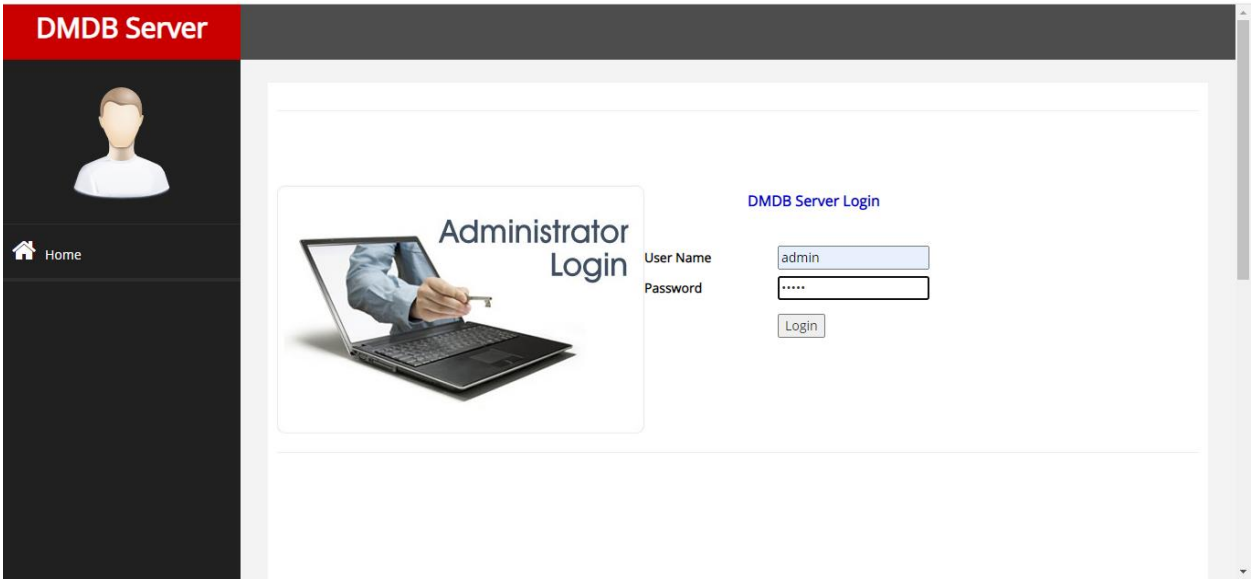

DECRYPT KEY:



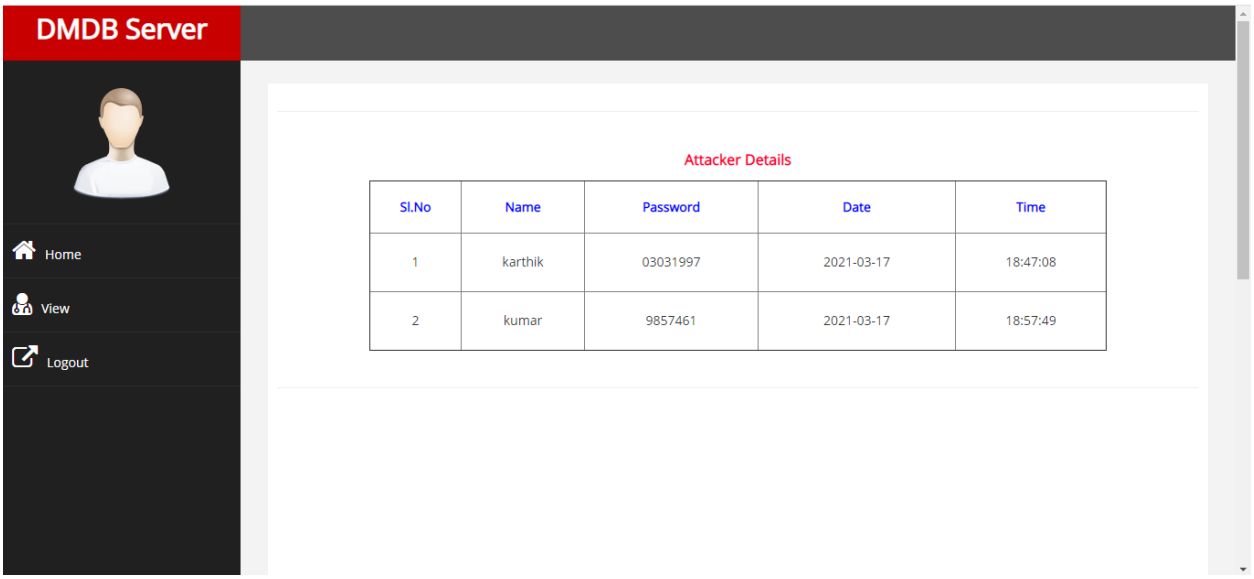
FILE STORED:



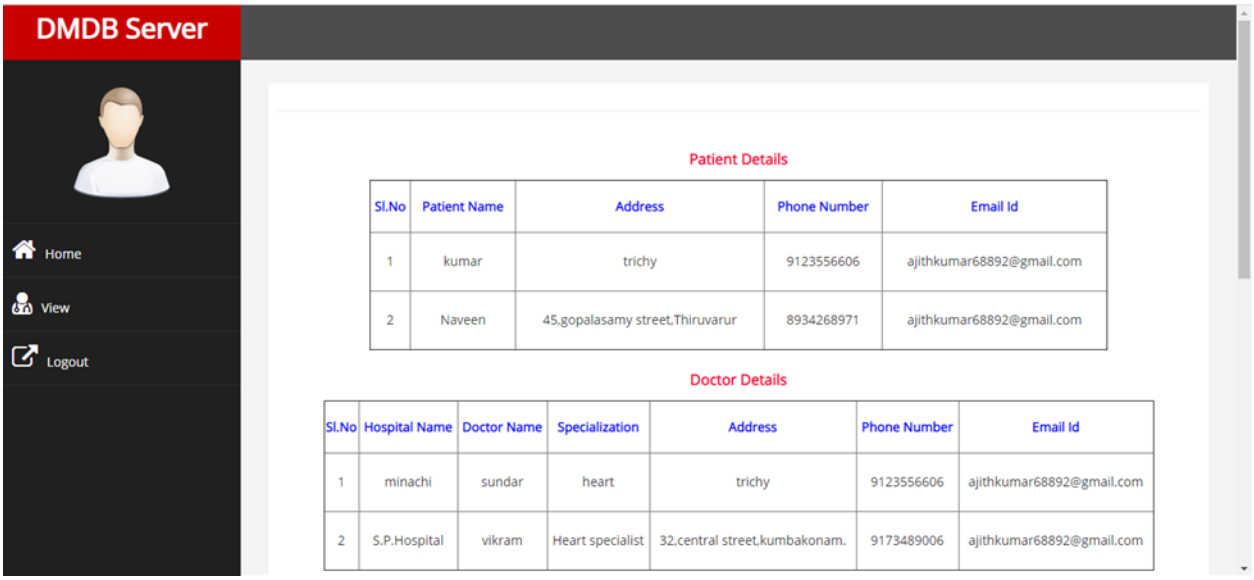

## DMDB SERVER:



# ATTACKER DETAILS:




# VIEW USER DETAILS:



# REFERENCES

# REFERENCES

## BOOK REFERENCE:

[1] Tiwari, Deepnarayan, and G. R. Gangadharan. "SecCloudSharing: Secure data sharing in public cloud using ciphertext-policy attribute-based proxy re-encryption with revocation." International Journal of Communication Systems 31, no. 5 (2018): e3494.

[2] Zhang, Y., Zheng, D., Li, Q., Li, J., & Li, H. (2016). Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing. Security and Communication Networks, 9(16), 3688-3702.

[3] Alderman, James, Jason Crampton, and Naomi Farley. "A framework for the cryptographic enforcement of information flow policies." In Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, pp. 143-154. 2017.

[4] Li, Jin, Yinghui Zhang, Xiaofeng Chen, and Yang Xiang. "Secure attribute-based data sharing for resource-limited users in cloud computing." Computers & Security 72 (2018): 1-12.

[5] Castiglione, Arcangelo, Alfredo De Santis, Barbara Masucci, Francesco Palmieri, Aniello Castiglione, Jin Li, and Xinyi Huang. "Hierarchical and shared access control." IEEE Transactions on Information Forensics and Security 11, no. 4 (2015): 850-865.

[6] Alderman, James, Christian Janson, Carlos Cid, and Jason Crampton. "Access control in publicly verifiable outsourced computation." In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, pp. 657-662. 2015.

[7] G. Rathi, Abinaya. M, Deepika. M¸ Kavyasri. T," Healthcare Data Security in Cloud Computing" International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 3, March 2015 Copyright to IJIRCCE 10.15680/ijircce.2015.0303089 1807

**WEBSITE REFERENCE:**

1. www.ntu.edu.sg
2. www.tutorialspoint.com
3. www.freeprojectz.com
4. www.edureka.co
5. www.code-projects.org