A Project Report

on

Secure File Storage On Cloud Using Hybrid Cryptography

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

S. PRIYANKA	**(174G1A0562)**
M. SREEVALLI	**(174G1A0592)**
K. SIVA VISHNU SAI	**(174G1A0583)**
J. PREM	**(174G1A0561)**

Under the Guidance of

Mrs. P. Rohini, M.Tech

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram village, B K Samudram Mandal, Ananthapuramu- 515701.

2020-2021

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY: ANANTAPURAMU

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))

Rotarypuram village, B K Samudram Mandal, Ananthapuramu- 515701.

Certificate

This is to certify that the project report entitled Secure File Storage On Cloud Using Hybrid Cryptography is the bonafide work carried out by **S. Priyanka** bearing Roll Number 174G1A0562**, M. Sreevalli** bearing Roll Number 174G1A0592**, K. Siva Vishnu Sai** bearing Roll Number 174G1A0583 and **J. Prem** bearing Roll Number 174G1A0561 in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2020-2021.

Guide

Mrs. P. Rohini, M.Tech
Assistant Professor

Head of the Department

Dr. G.K.V. NarasimhaReddy, Ph.D
Professor & HOD

Date:

EXTERNAL EXAMINER

Place: Ananthapuramu

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express our indebted gratitude to our Guide **Mrs.P.Rohini, Assistant Professor, Computer Science and Engineering Department**, who has guideded us a lot and encouraged us in every step of the project work. We thank her for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep felt gratitude to **Mr.P.Chitralingappa , Associate Professor and Mrs.M.Soumya, Assistant Professor** project coordinator valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We are very much thankful to **Dr. G.K.V.Narasimha Reddy, Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to **Dr.G.Balakrishna,, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

DECLARATION

We, Ms. S. Priyanka bearing reg no: 174G1A0562, Ms. M. Sreevalli bearing reg no: 174G1A0592, Mr. K. Siva Vishnu Sai bearing reg no: 174G1A0583, Mr. J. Prem bearing reg no:174G1A0561, students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, Rotarypuram, hereby declare that the dissertation entitled "SECURE FILE STORAGE ON CLOUD USING HYBRID CRYPTOGRAPHY" embodies the report of our project work carried out by us during IV Year Bachelor of Technology under the guidance of Mrs. P. Rohini, M.Tech, Department of CSE and this work has been submitted for the partial fulfillment of the requirements for the award of Bachelor of Technology degree.

The results embodied in this project report have not been submitted to any other Universities of Institute for the award of Degree.

S.PRIYANKA	Reg no: 174G1A0562
M.SREEVALLI	Reg no: 174G1A0592
K.SIVA VISHNU SAI	Reg no: 174G1A0583
J.PREM	Reg no: 174G1A0561

Contents

List of Figures

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
DES	Data Encryption Standard
RSA	Rivest-Shamir-Adleman

ABSTRACT

The proposed model is liable to meet the required security needs of data center of cloud. AES, DES and RSA are used for the encryption of file slices takes minimum time and has maximum throughput for encryption and decryption from other symmetric algorithms. The idea of splitting and merging adds on to meet the principle of data security. The hybrid approach when deployed in cloud environment makes the remote server more secure and thus, helps the cloud providers to fetch more trust of their users. For data security and privacy protection issues, the fundamental challenge of separation of sensitive data and access control is fulfilled. Cryptography technique translates original data into unreadable form. Cryptography technique is divided into symmetric key cryptography and public key cryptography. This technique uses keys for translate data into unreadable form. So only authorized person can access data from cloud server. Cipher text data is visible for all people.

CHAPTER-1

INTRODUCTION

1.1 Introduction

Today's technologies are growing at very fast speed and deliver the user with many attractive services to reduce the burden of large volumetric data storage and maintenance. Nowadays, many online services are applicable which provides all services and data online such as e-messaging, e-billing, e-transaction, e-mail etc. All these services required user's data online for processing. This data may be any confidential information, which is required by user to be safe from any malicious activity like-healthcare information, bank transaction, credit card details, etc. A high requirement arises for security and protection of data from any unauthorized user as leakage of confidential information may result in serious impairment to user. This increases the security requirement of confidential data before actually migrating it over online internet access. We need to develop a sound, safe and secure framework to protect our confidential data from any such malicious attack. There is a need to convert confidential data into some another form, which becomes inexplicable for any attacker and only authorized users are able to understand that exactly what data is communicated. One of the major techniques to achieve this requirement is cryptography, also known as "code making" or "code generation". It allows us to translate a message into unreadable form for malicious attacker. Another part of this approach is known as cryptanalysis and "code breaking.

CHAPTER-2

LITERATURE SURVEY

2.1 Introduction

Existing key relation technique while performing encryption/decryption of file and address the security challenges needs to be resolve in cloud computing. By experiment analysis they had also proved that CA inverter and shifter during encryption and decryption respectively helps to reduce the time complexity as well as deal with various security attacks more efficiently.

Author Fadhil proposed a hybrid cryptographic technique by a combination of public RSA cryptosystem and knapsack. This proposed technique is less complex and more secure than individual algorithm. It works in two stages- first perform the RSA encryption and forward its output to knapsack approach. Reverse process needs to be applied while performing the decryption at receiver end.

Author Zissis proposed various security issues need to consider while adopting cloud computing such as data integrity, confidentiality, availability, threats, identification and authentication, etc. A new actor namely third-party auditor has also been introducing who will perform auditing on the user request. This auditing feature helps the user to get information regarding its data integrity.

Author Amit introduce randomized cryptographic technique. They have introduced the different variations for Ceaser cipher using public key cryptography and randomized technique. Cryptographic techniques utilize the same data.

2.2 Existing System

In existing system cloud used to use any one of the encryption technique and keys verification are done using identity of user. Based on application requirement different encryption techniques are used.

2.3 Disadvantages

Only single encryption techniques are used and keys are not managed effectively there are changes of leakage of keys**.**

2.4 Proposed System

In order to improve security for cloud data compare to existing techniques where keys are shard security between users new hybrid cryptography technique is proposed where three types of encryption are used AES, DES and RSA and LSB steganography technique is used for secure key sharing.

2.5 Advantages

- The system is very secure and robust in nature.
- Data is kept secured on cloud server which avoids unauthorized access.
- The key is also safe as it embeds the key using their methods.

CHAPTER-3

ANALYSIS

3.1 Requirement Specification

Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

3.2 Hardware Requirements

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

- System : Pentium IV 2.4 GHz.
- Hard Disk. : 100 GB.
- Monitor : 15 VGA Color.
- Mouse : Logitech.
- RAM : 1 GB.

3.3 Software Requirements

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

- Operating system : Windows 10
- Coding Language : Python
- Tool : Anaconda, Visual Studio IDE
- Database : SQL lite

3.4 Installation of Visual Studio IDE

The Visual Studio integrated development environment is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. An integrated development environment (IDE) is a feature-rich program that can be used for many aspects of software development. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to ease the software development process.

Step 1) Download Visual Studio

First, visit the following Visual Studio free download link https://visualstudio.microsoft.com/downloads/

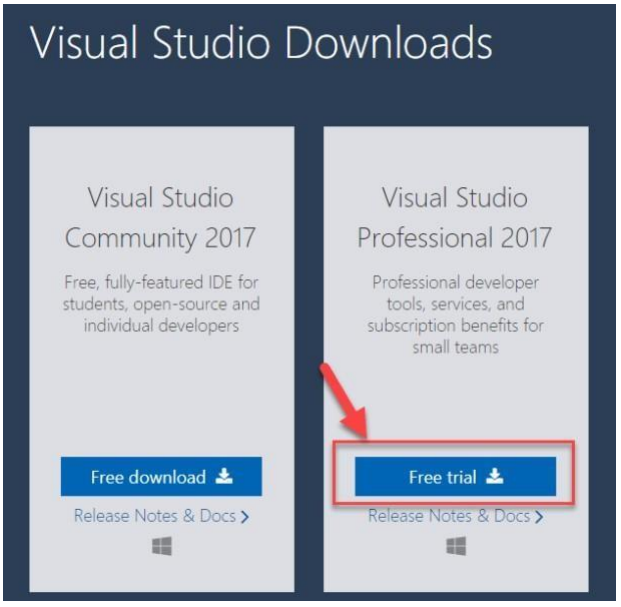


Fig.3.1. Visual Studio Downloads

You can select

- Visual Studio 2019 Community Edition
- Visual Studio 2019 Professional Edition (30 Day Free Trial)

Step 2) Open the .exe file

Click on the downloaded exe file

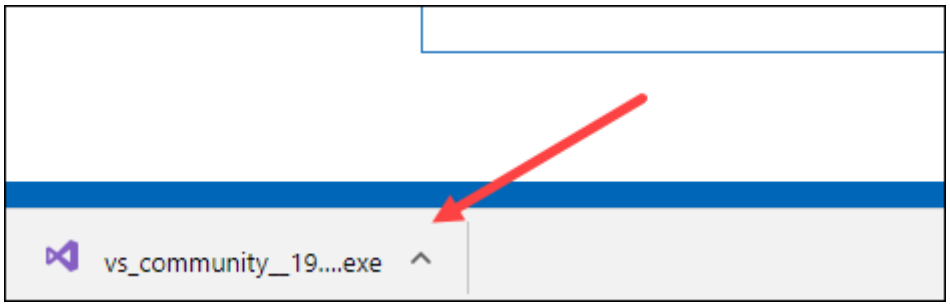


Fig.3.2. Executable File

Step 3) Start the installation

In the next screen, click continue to start Visual Studio installation

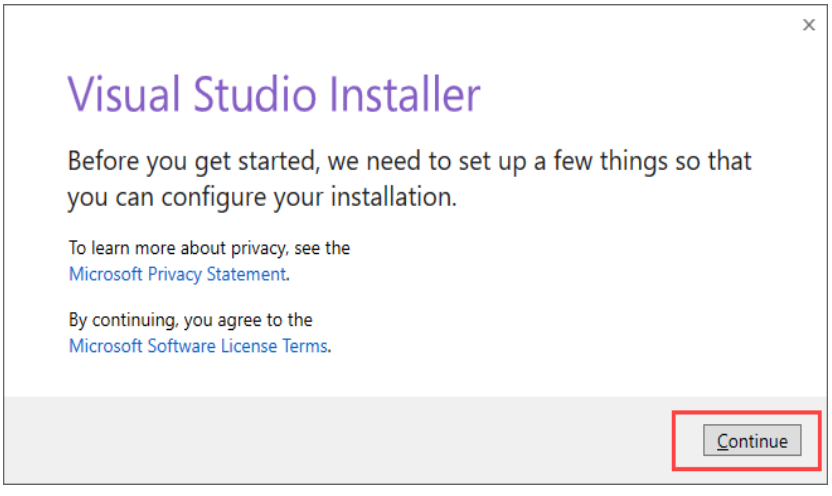


Fig.3.3. Visual Studio Installer

Step 4) Let the installation complete

Visual Studio will start downloading the initial files. Download speed will vary as per your internet connection.

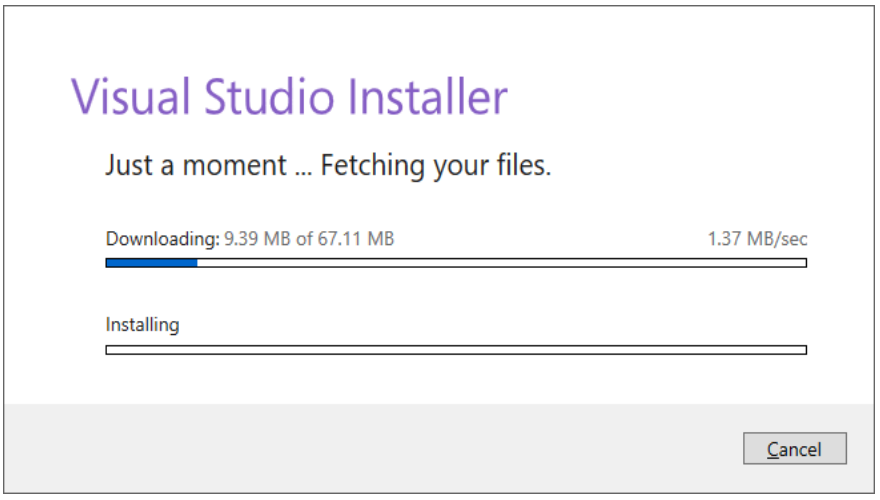


Fig.3.4. Downloading Visual studio Installer

Step 5) Choose the software version

In next screen, click install

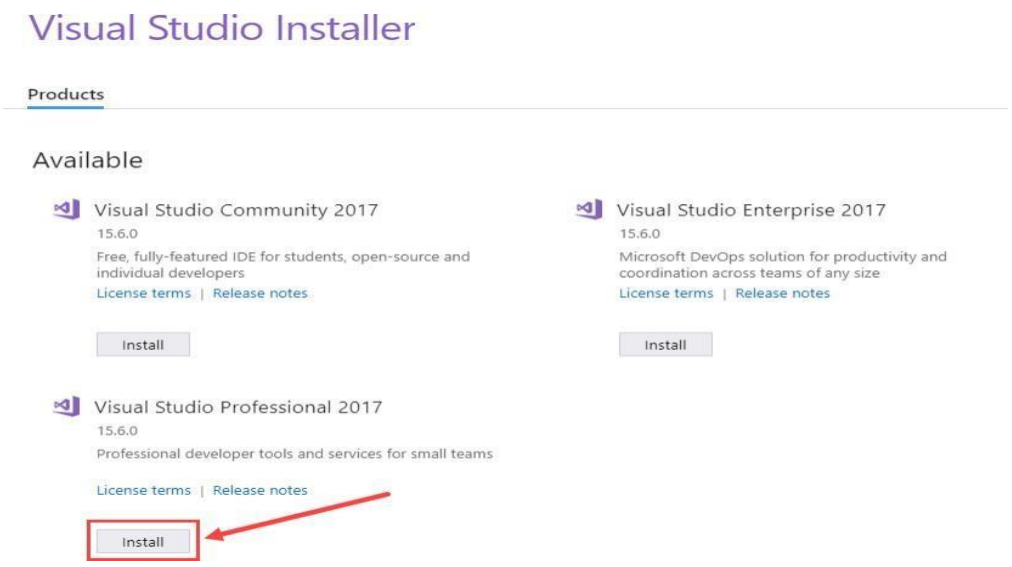


Fig.3.5. Choosing Software Version

Step 6) Select the desktop version

In next screen,

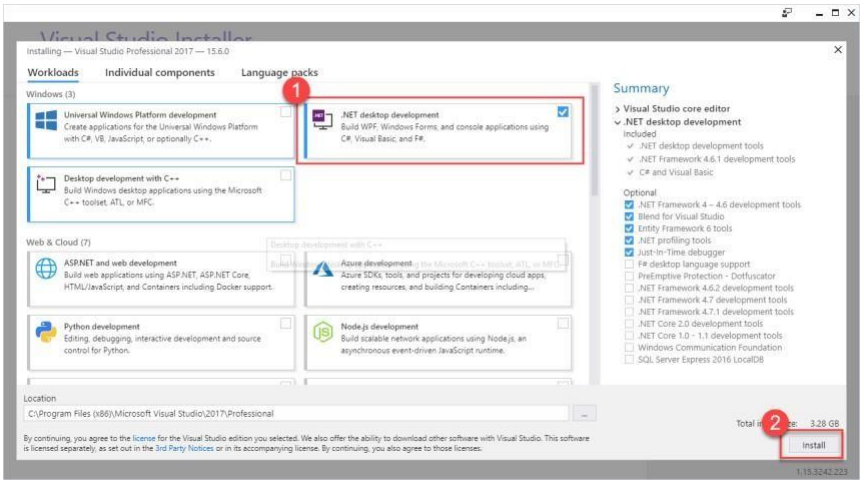


Fig.3.6. Selecting Desktop Version

1. Select ".Net desktop development"
2. Click install

Step 7) Wait for the files to be downloaded

Visual Studio will download the relevant files based on the selection in step 6

Fig.3.7. Files Download

Step 8) Reboot your PC

Once the download is done, you will be asked to reboot the PC to complete Visual Studio setup

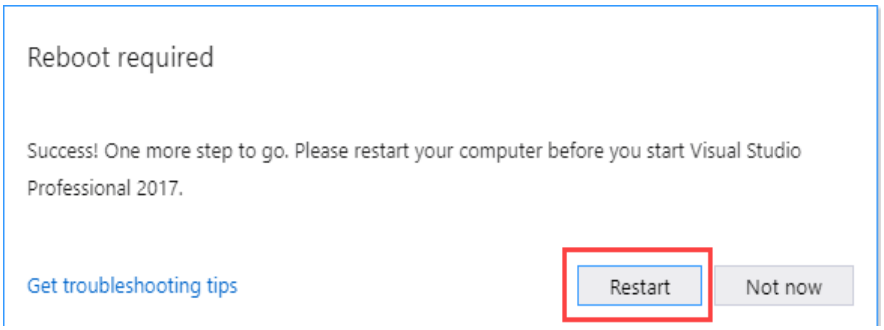


Fig.3.8. Restart the Pc

Step 9) Open Visual Studio

Post reboot, open the Visual Studio IDE.

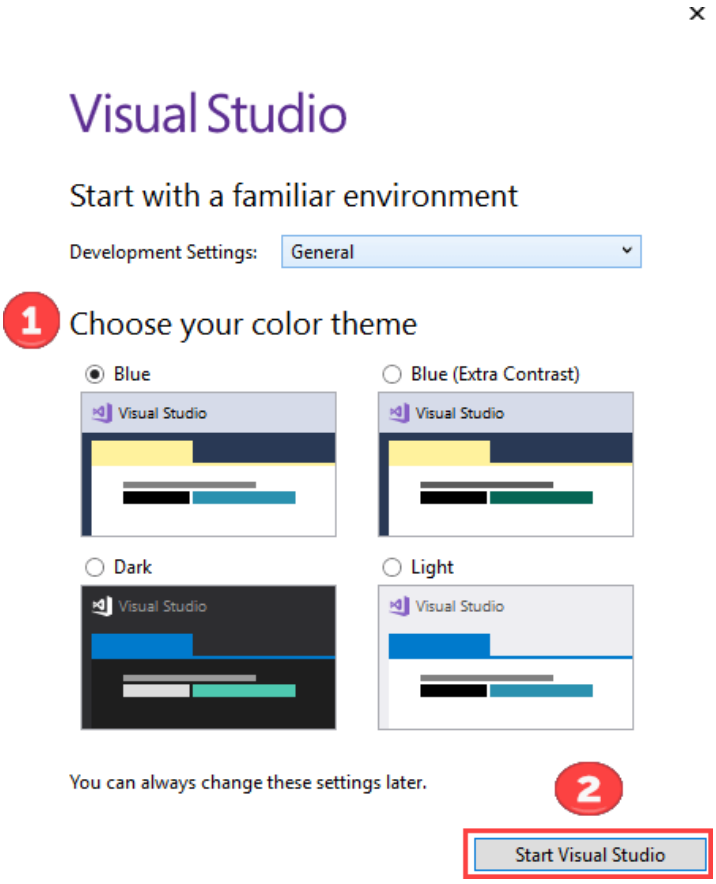


Fig.3.9. Setting Theme

1. Select a theme of your choice

2. Click Start Visual Studio

Step 10) Start using Visual Studio

In Visual Studio IDE, you can navigate to File menu to create new applications.

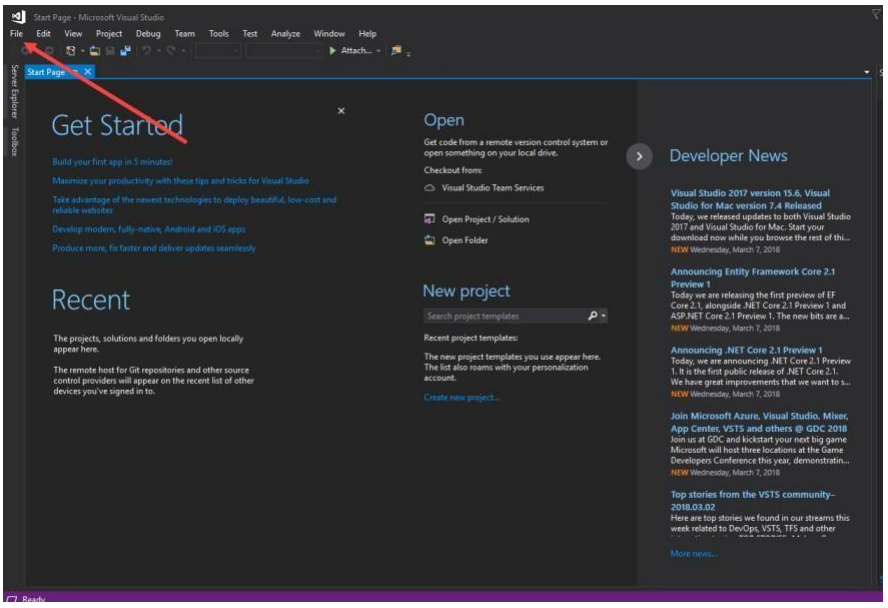


Fig.3.10. Starting Visual Studio IDE

3.5 Installation of Anaconda

3.5.1 Overview

Anaconda is a package manager, an environment manager, and Python distribution that contains a collection of many open source packages (numpy, scikit-learn, scipy, pandas to name a few). If you need additional packages after installing Anaconda, you can use Anaconda's package manager, conda or pip to install those packages. This is highly advantageous as you don't have to ma manage dependencies between multiple packages.

1. Go to the Anaconda Website and choose either a Python 3.x graphical installer (A) or a Python 2.x graphical installer (B).
2. If you aren't sure which Python version you want to install, choose Python 3. Do not choose both.

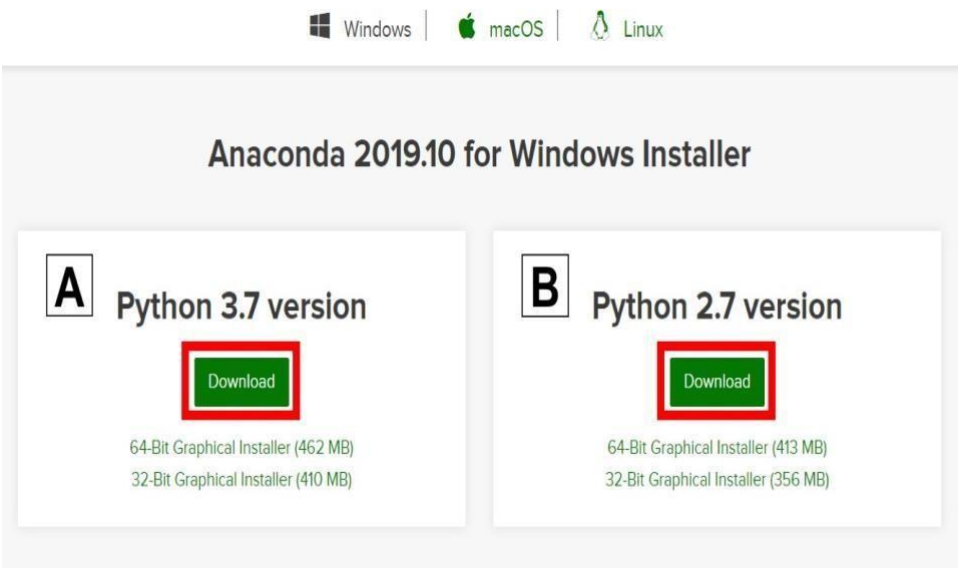


Fig.3.11. Choosing Python Version

2. Locate your download.

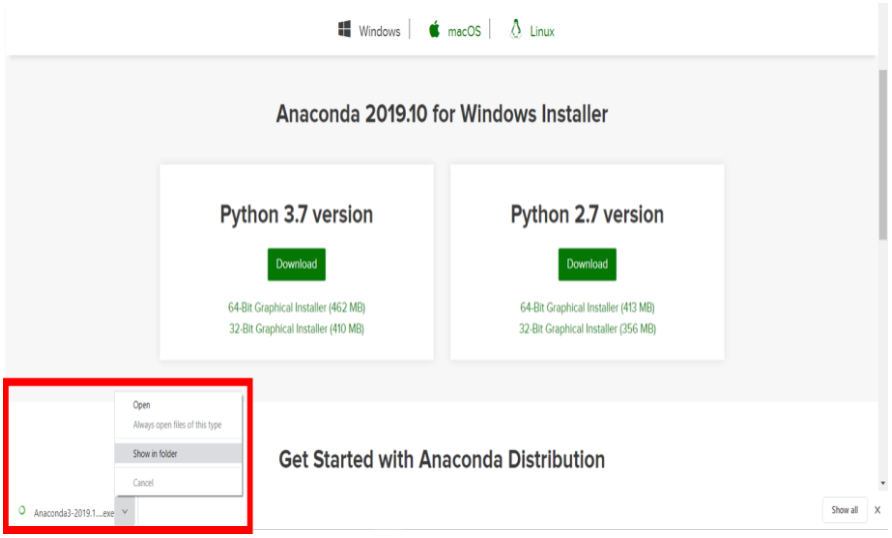


Fig.3.12. Location

Installing as administrator is for the case you don't have permission install anaconda in the location you want or to add anaconda to your path.

When the screen below appears, click on Next.

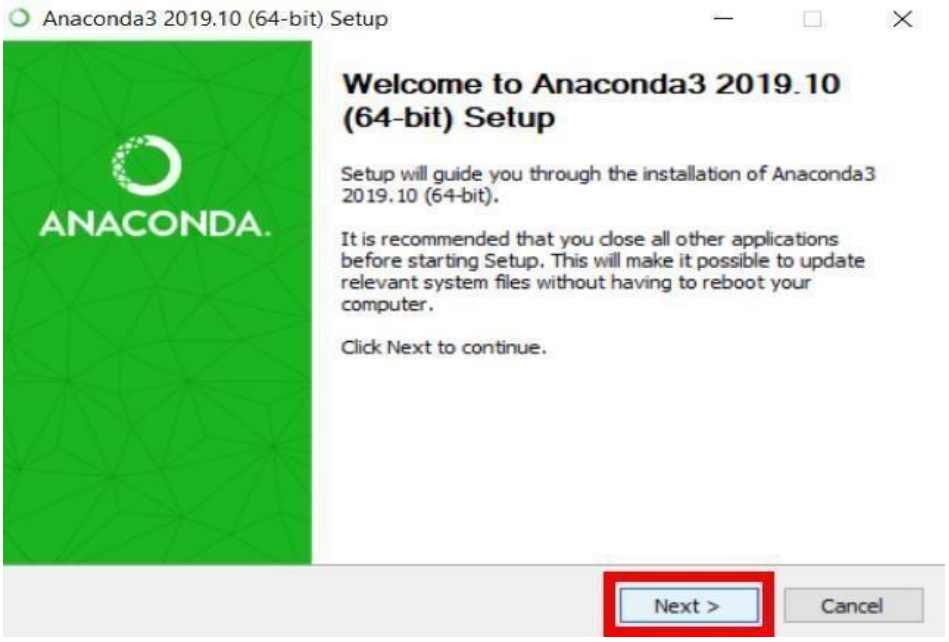


Fig.3.13. Setup

3. Read the License Agreement and click on I Agree.

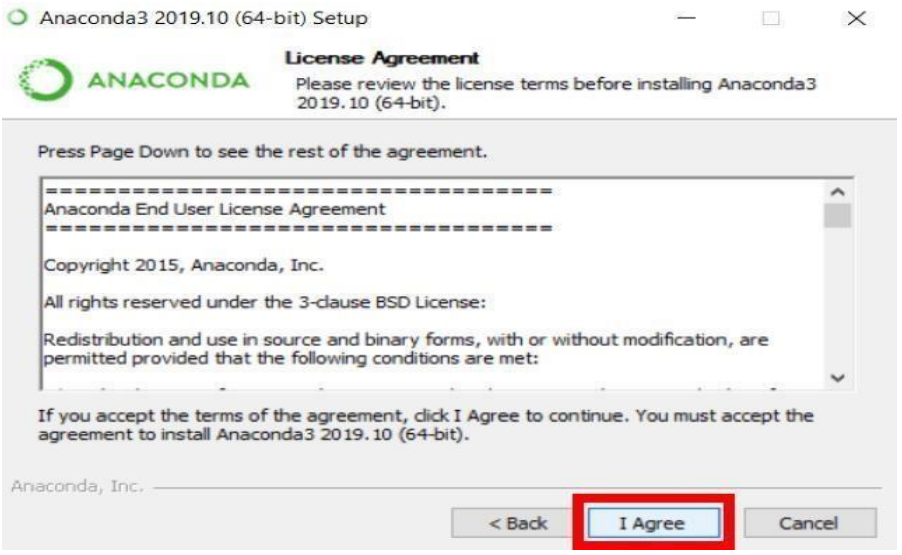


Fig.3.14. Lincense Agreement

4. Choose either Just Me (recommended) or All Users.

If you aren't sure which to select, choose Just Me as this can mitigate potential issues if you don't have administrator privileges.

5. Please make a note of your installation location (1) and then click Next (2).

Your installation location can vary so keep note of where you installed anaconda. In the example image on the left, the path is similar to if you selected “Just Me” for step 4. In the example image on the right, the path is similar to if you selected “All Users” for step 4.

6. This is an important part of the installation process. The recommended approach is to not check the box (1) to add Anaconda to your path. This means you will have to use Anaconda Navigator or the Anaconda Command Prompt (located in the Start Menu under “Anaconda”) when you wish to use Anaconda (you can always add Anaconda to your PATH later if you don’t check the box). If you want to be able to use Anaconda in your command prompt, please use the alternative approach and check the box. Click on Install (2). This is important. Consider what you are doing in this step.

7. Click on Next.

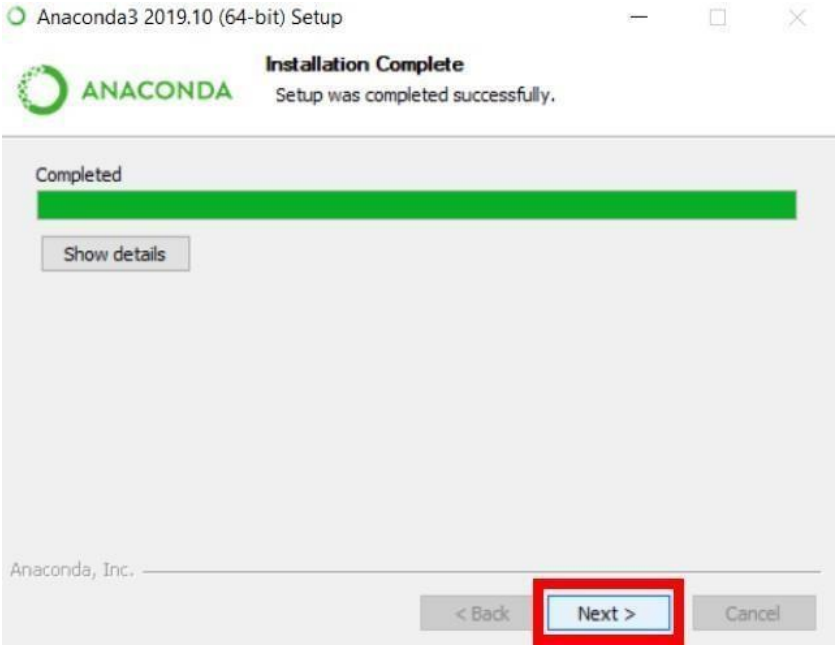


Fig.3.15. Installation Complete

8. Click on Finish.

Fig.3.16. Installation Completed

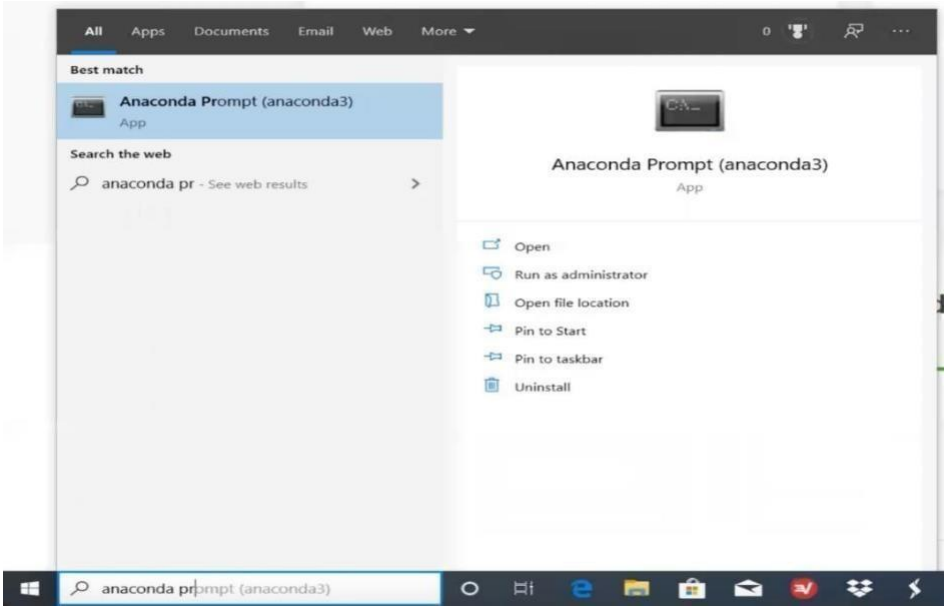


Fig.3.17. Anaconda Prompt

3.6 DB Browser Installation

Whether we want to practice SQL programming or just want a simple database to store info, the starting point is the same. We are going to need to create a database. When we first run the Slate Database Browser, we will see a straightforward main window with a menu bar, a toolbar and three tabs. When we first get started,

obviously there won't be any database structure available, so the main display area will be blank.

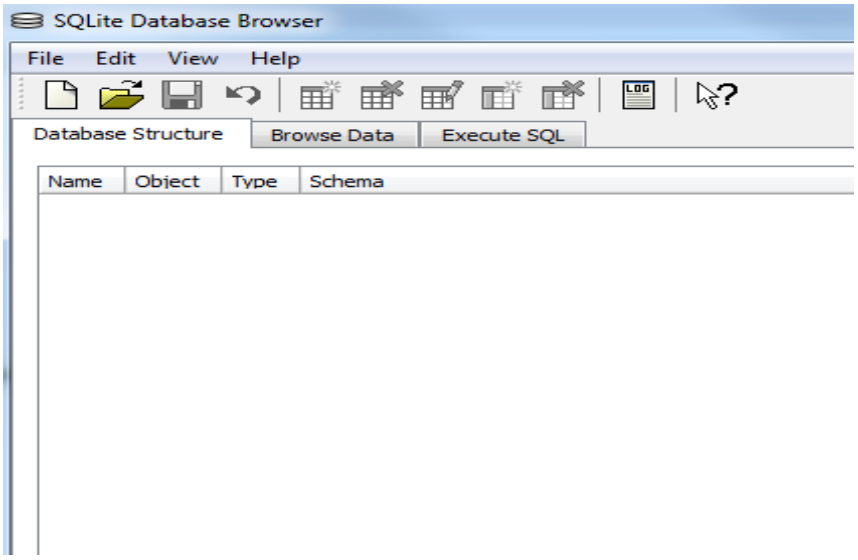


Fig.3.18. SQLite Database Browser

Click on "File" and we can either click on "New Database" or click on "Import" to import data that we might already have in some other format, like an existing database from an SQL file, or an Excel table that we have exported to a CSV file. Either format can be imported into our new Slate database.

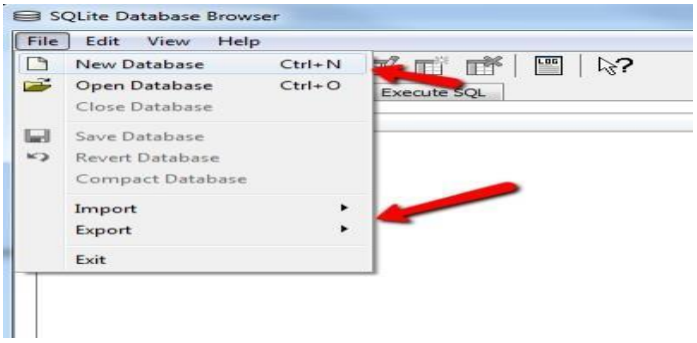


Fig.3.19. Creating Database

If we want to start from scratch, then click "New Database", and we will need to create the structure of our database. Create our first table, add database fields to that table, and define the format for each field (text, number, etc…).

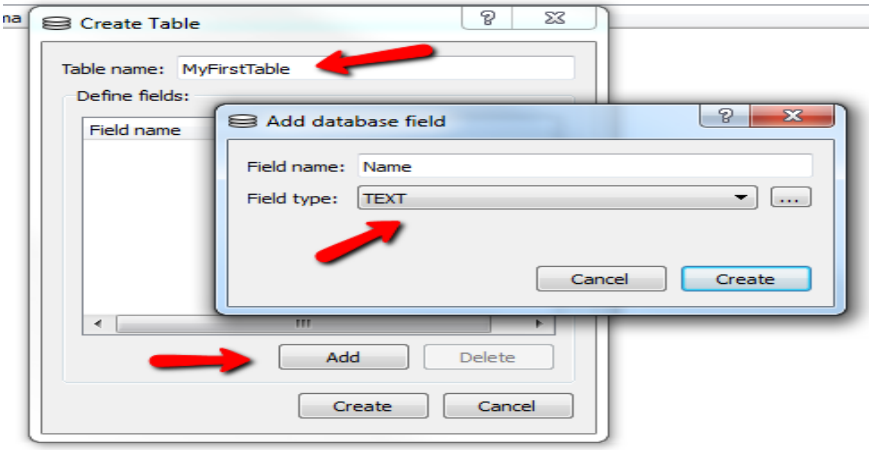


Fig.3.20. Adding Database Field

Each database field can be a string (text), a number (numeric), a blob (binary data) or an integer key.

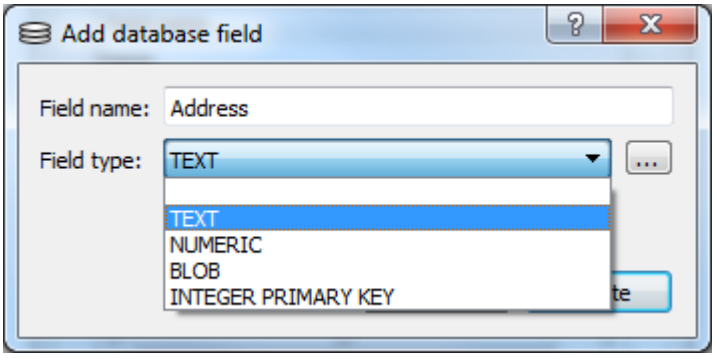


Fig.3.21. Selecting Field Type

When we are done creating our first table in the database, we will see the structure under the Database Structure tab on the main window. As we create each table in the database, we will see the tree start forming that will contain all tables and the fields within them. This is a fast, quick overview of what our entire database looks like, and an easy way to navigate it once it's starts growing.

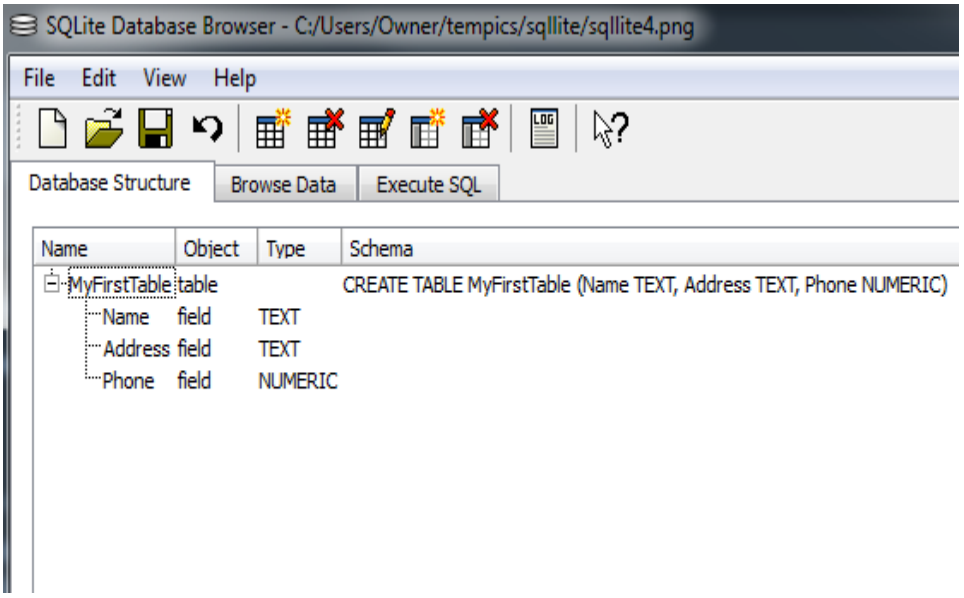


Fig.3.22. Database Structure

Viewing and manipulating our database data is as simple as clicking the "Browse Data" tab and editing the records directly. This is also where we can create new data records, delete records, or search for data within very large tables.

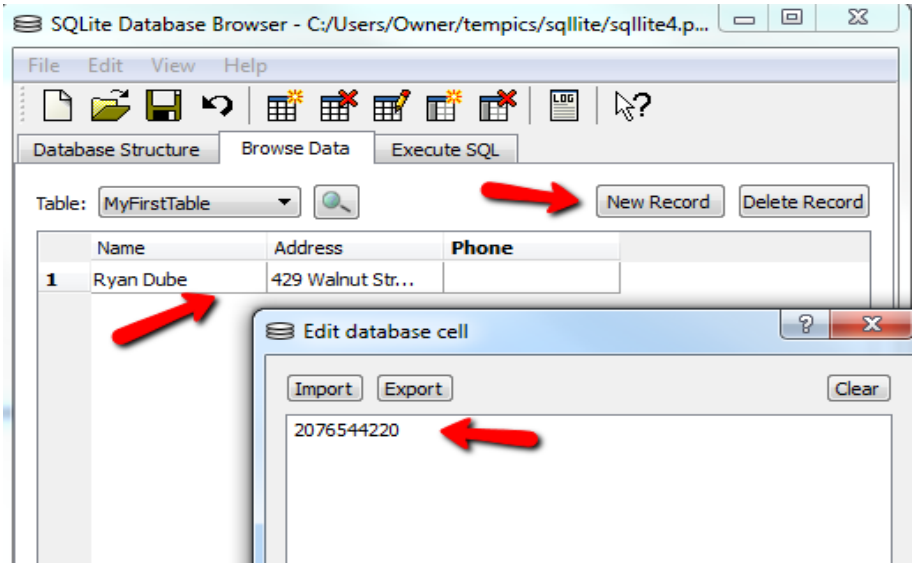


Fig. 3.23. Browsing Data

Of course, the most important feature – at least the main reason that we installed the software is the "Execute SQL" tab, where we can enter your SQL command strings that we want to run on our database. When we click on "Execute query", we will see the results of the query in the "Data returned" field. Or we will see the errormessage.

However, as a tool to learn SQL, the error message field is kind of nice because it tells us what we are doing wrong. We can use that as a clue to rework our SQL statement and try again.

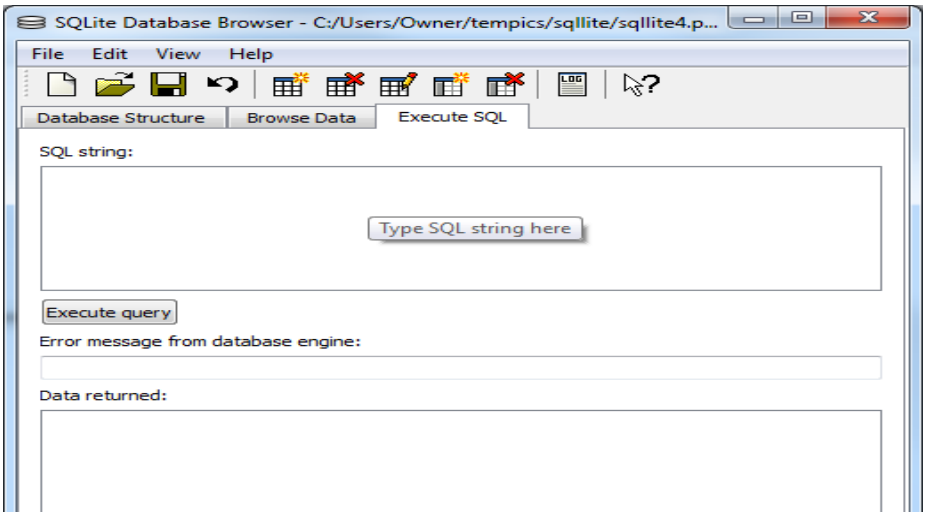


Fig.3.24. Executing SQL

Another feature, is the SQL Log window that we can open up by clicking on the "Log" icon in the toolbar. This shows us a complete log of all SQL statements that have been executed. This is nice when we have just gotten lost and our query is completely messed up from all of the tweaks we have tried. We can go back into the log and find the original version of our query before it got all twisted up.

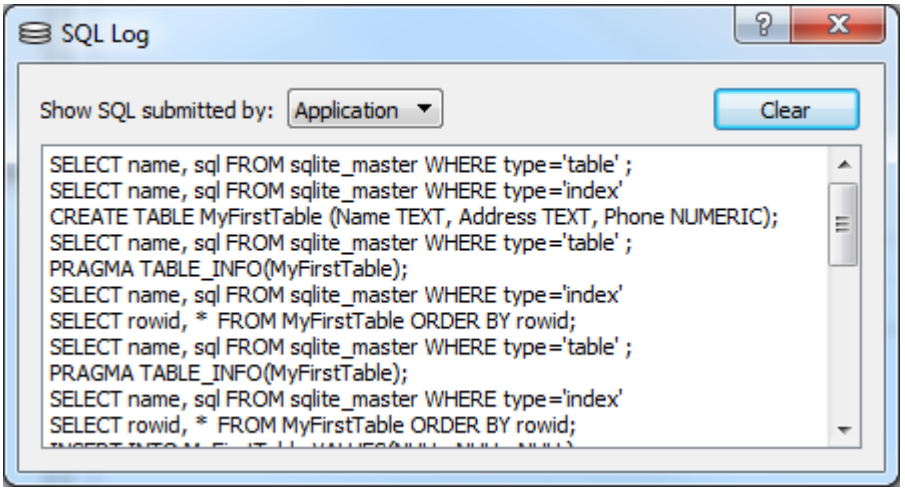


Fig.3.24. SQL Log Window

Slate Browser is a sweet application to get started with database programming and if we want to quickly create a personal database to store some data that we have

got kicking around. Having it in such a database gives us the added benefit of conducting SQL queries on it, which we could not really do with the data if it's just in some spreadsheet.

So, give Slate Browser a try and see if it gives our SQL programming skills a bit of a boost.

3.7 Languages Used

The programming language that was used in our Breast Cancer Detection project is Python. The implementation of source code was done through python. Python is an interpreted, interactive, object-oriented programming language which is suitable for implementing machine learning algorithms in easier way.

Features of Python

Python provides lots of features that are listed below:

Easy to Learn and Use:

Python is easy to learn and use. It is developer-friendly and high-level programming language.

Expressive Language:

Python language is more expressive means that it is more understandable and readable.

Interpreted Language:

Python is an interpreted language i.e. interpreter executes the code line by line at a line. This makes debugging easy and thus suitable for beginners.

Cross-platform Language:

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

Free and Open Source:

Free and open source are easily available. The Flexibility makes it different.

Python language is freely available at official web address. The source-code is also available. Python is opted by most of the people.

Object-Oriented Language:

Python supports object-oriented language and concepts of classes and objects come into existence.

Extensible:

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

Large Standard Library:

Python has a large and broad library and provides rich set of module and functions for rapid application development.

GUI Programming Support:

Graphical user interfaces can be developed using Python.

Integrated:

It can be easily integrated with languages like C, C++ and JAVA etc.

3.7 Flask Framework

Flask is the micro web framework written in python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Templates are files that contain static data as well as placeholders for dynamic data. A template is rendered with specific data to produce a final document. Flask uses the Jinja template library to render templates.

This way, whenever you want to install or deploy your project, you'll have all the necessary packages in the requires list. You'll also have everything you need to set up and install the package in site-packages. For more information on how to write an installable Python distribution, check out the docs on setup.py.

Within the todo directory containing your source code, create an app.py file and a blank_init__.py file. The__init__.py file allows you to import from todo as if it were an installed package. The app.py file will be the application's root. This is where all the Flask application goodness will go, and you'll create an environment variable that points to that file. If you're using pipenv (like I am), you can locate your virtual environment with pipenv --venv and set up that environment variable in your environment's activate script.

CHAPTER-4

DESIGN

4.1 UML Introduction

The unified modelling language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. □

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

4.2 Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

4.3 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact

with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

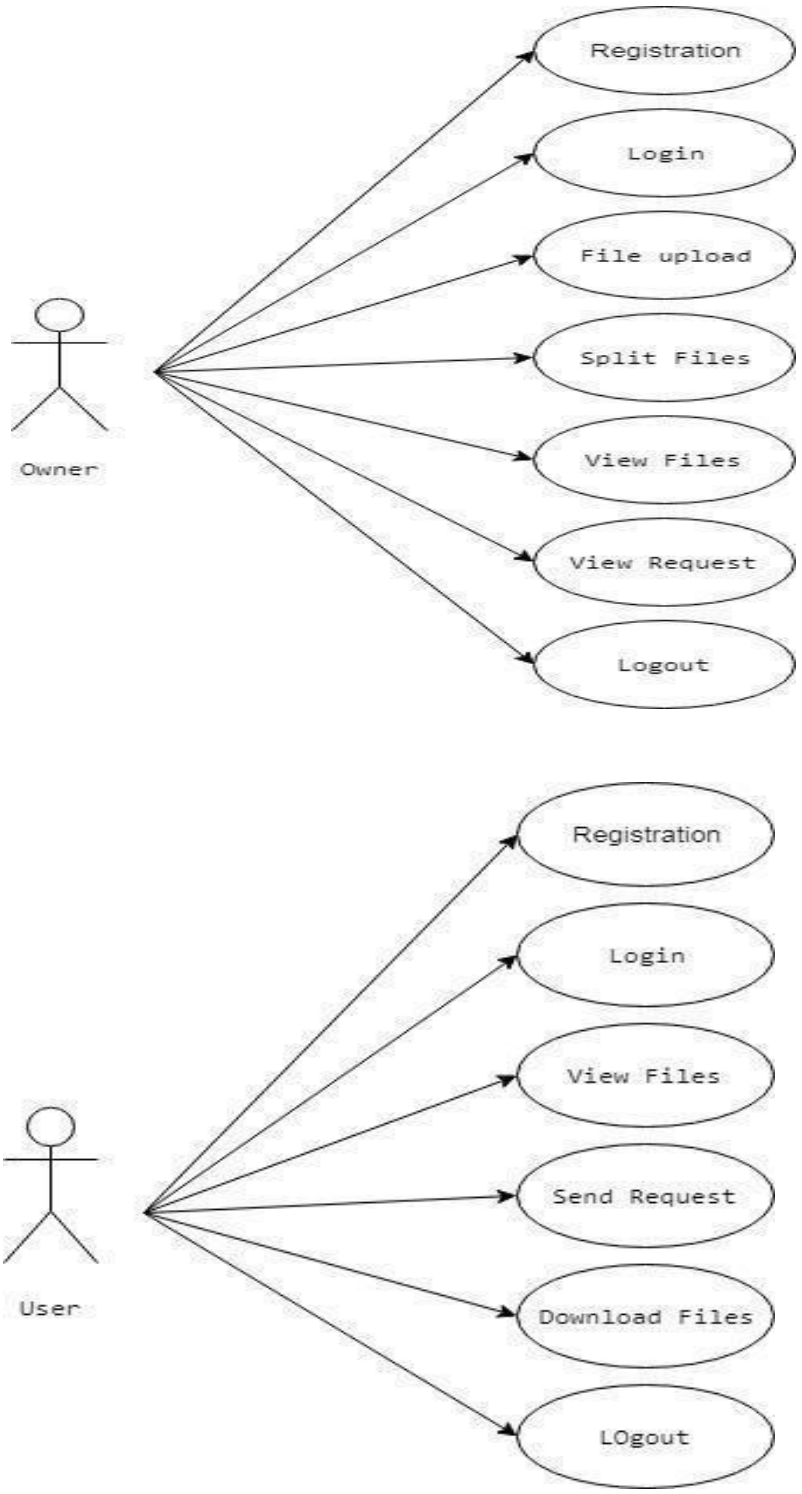


Fig.4.3. Usecase Diagram

4.4 CLASS DIAGRAM

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. A class with three sections, in the diagram, classes is represented with boxes which contain three parts

- The upper part holds the name of the class.
- The middle part contains the attributes of the class.
- The bottom part gives the methods or operations the class can take or undertake.

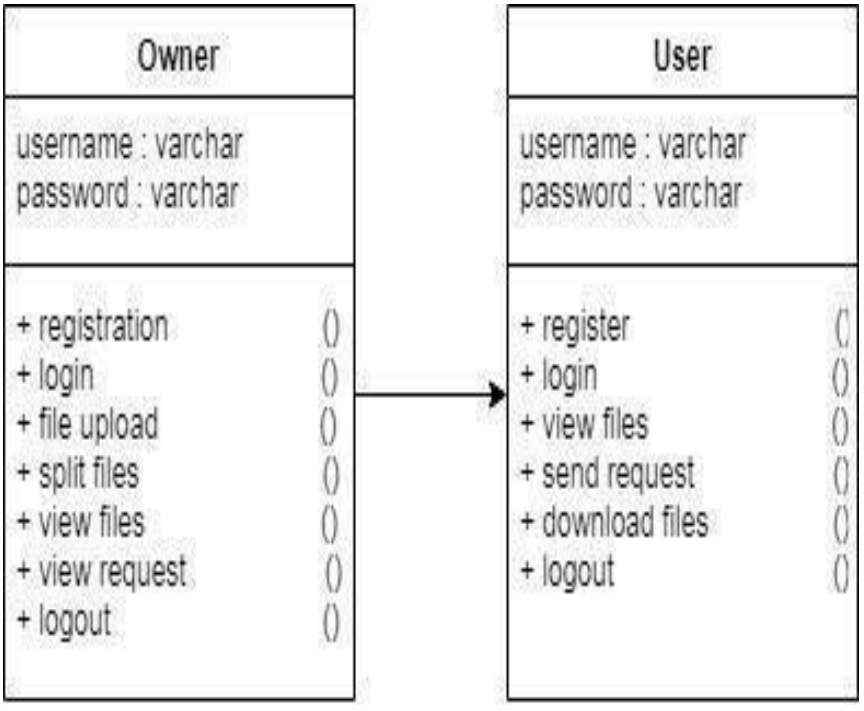
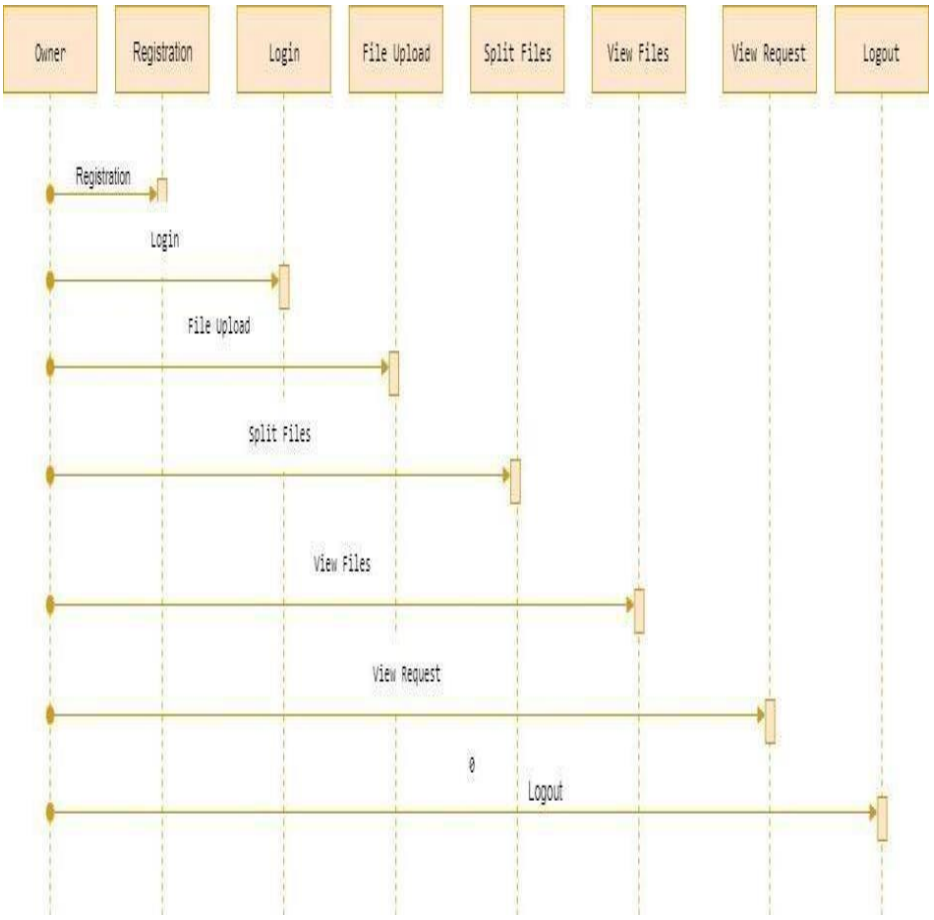


Fig.4.4. Class Diagram

4.5 SEQUENCEDIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



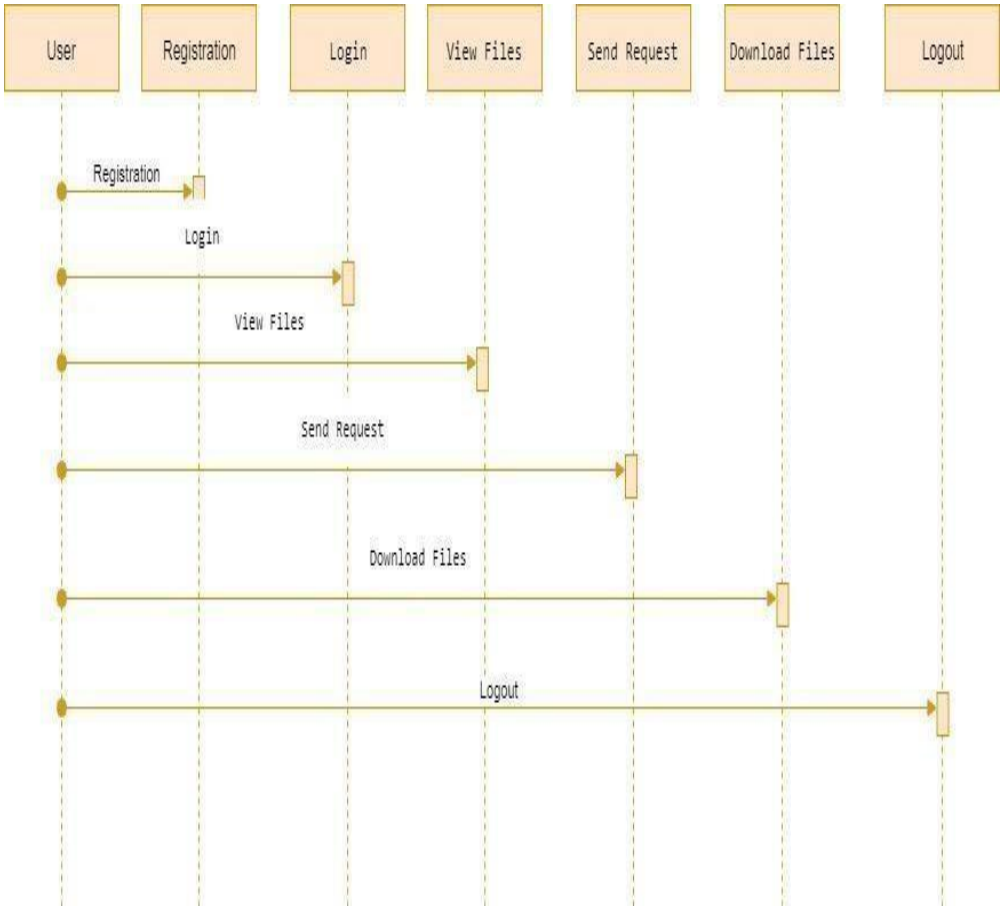


Fig .4.5. Sequence Diagram

4.6 Activity Diagram

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

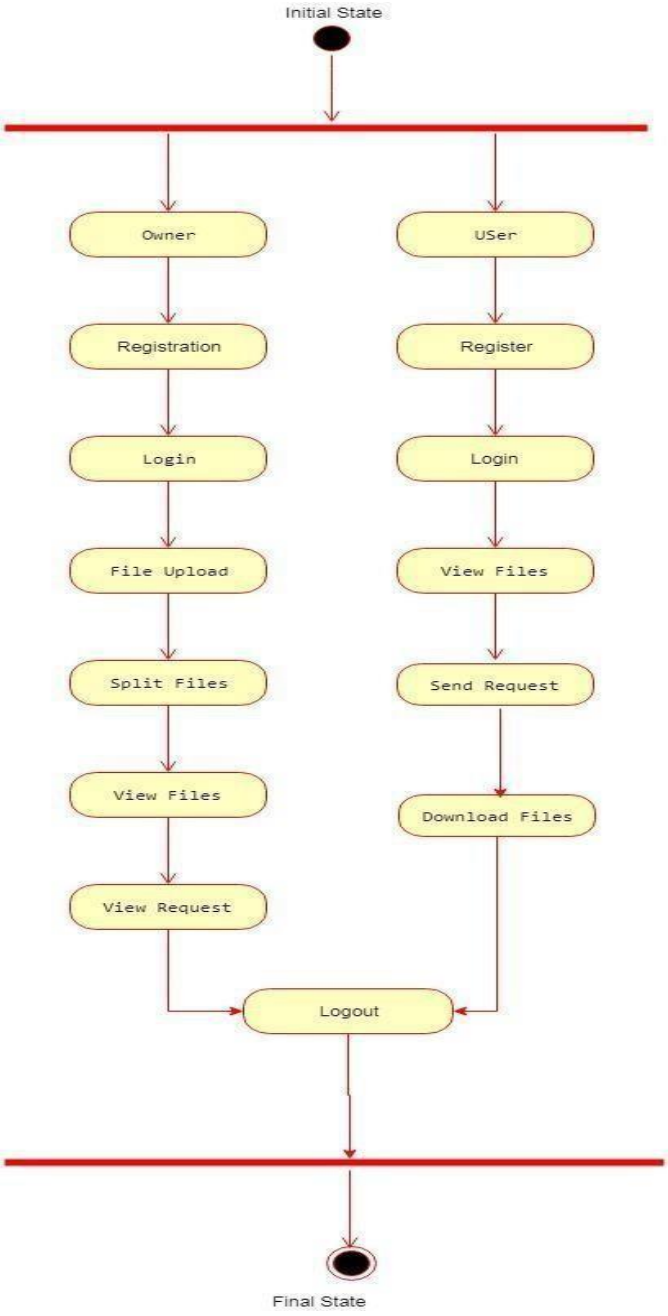


Fig.4.6. Activity Diagram

4.7. Component Diagram

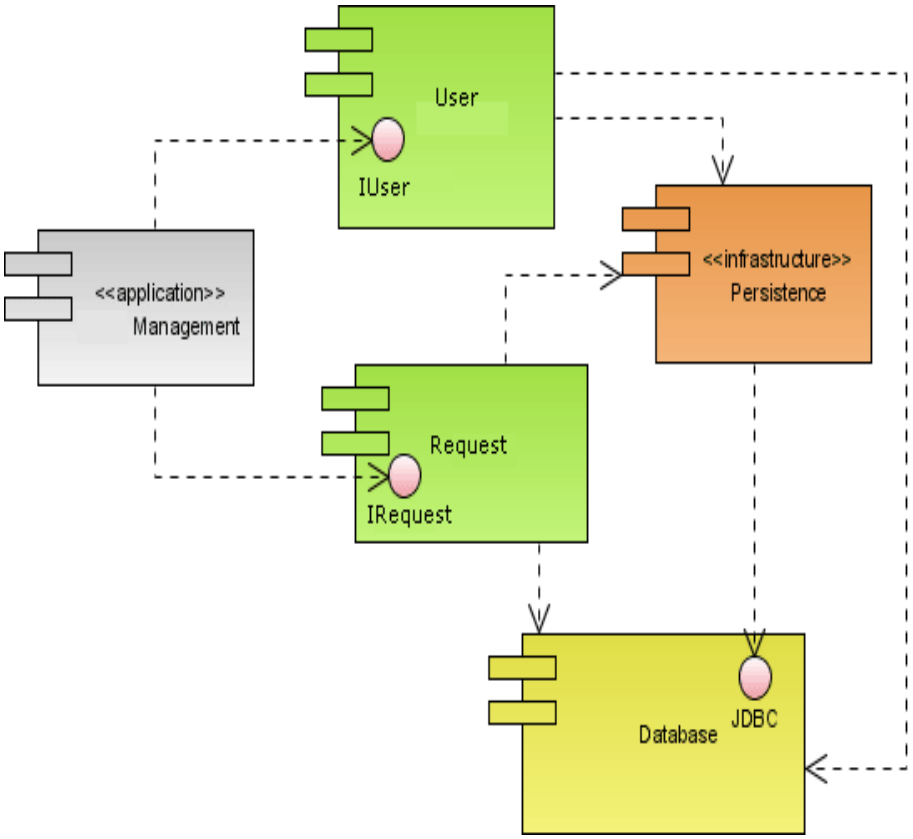


Fig.4.7. Component Diagram

CHAPTER-5

IMPLEMENTATION

5.1 AES Encryption

All of the cryptographic algorithms we have looked at so far have some problem. The earlier ciphers can be broken with ease on modern computation systems. The DES algorithm was broken in 1998 using a system that cost about $250,000. It was also far too slow in software as it was developed for mid-1970's hardware and does not produce efficient software code. Triple DES on the other hand, has three times as many rounds as DES and is correspondingly slower. As well as this, the 64 bit block size of triple DES and DES is not very efficient and is questionable when it comes to security. What was required was a brand new encryption algorithm.

One that would be resistant to all known attacks. The National Institute of Standards and Technology (NIST) wanted to help in the creation of a new standard. However, because of the controversy that went with the DES algorithm, and the years of some branches of the U.S. government trying everything they could to hinder deployment of secure cryptography this was likely to raise strong skepticism. The problem was that NIST did actually want to help create a new excellent encryption standard but they couldn't get involved directly.

Unfortunately they were really the only ones with the technical reputation and resources to the lead the effort. Instead of designing or helping to design a cipher, what they did instead was to set up a contest in which anyone in the world could take part. The contest was announced on the 2nd of January 1997 and the idea was to develop a new encryption algorithm that would be used for protecting sensitive, non- classified, U.S. government information. The ciphers had to meet a lot of requirements and the whole design had to be fully documented (unlike the DES cipher). Once the candidate algorithms had been submitted, several years of scrutinisation in the form of cryptographic conferences took place. In the first round of the competition 15 algorithms were accepted and this was narrowed to 5 in the second round.

The fifteen algorithms are shown in table 7 of which the 5 that were selected are shown in bold. The algorithms were tested for efficiency and security both by some of the worlds best publicly renowned cryptographers and NIST itself. After all this investigation NIST finally chose an algorithm known as Rijndael. Rijndael was named after the two Belgian cryptographers who developed and submitted it - Dr. Joan Daemen of Proton World International and Dr. Vincent Rijmen, a postdoctoral researcher in the Electrical Engineering Department of Katholieke Universisteit Leuven. On the 26 November 2001, AES (which is a standarised version of Rijndael).

AES stands for Advanced Encryption System and its a symmetric encryption algorithm.It is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.Here is the wiki link for AES.The AES engine requires a plain-text and a secret key for encryption and same secret key is required to again decrypt it.

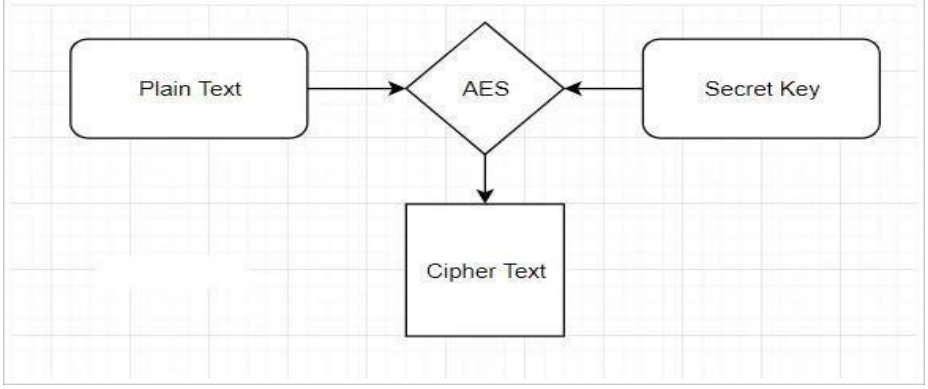


Fig.5.1. AES Encryption

5.1.1 AES Encryption in Python

Following is the sample program in python that performs AES encryption. Here, we are using AES with CBC mode to encrypt a message as ECB mode is not semantically secure. The IV mode should also be randomized for CBC mode.

If the same key is used to encrypt all the plain text and if an attacker finds this key then all the cipher can be decrypted in the similar way. We can use salt and iterations to improve the encryption process further. In the following example we are using 128 bit encryption key.

5.1.2 Encryption Code:

```
import sys
import base64
from Crypto.Cipher import AES
class AESCipher(object):
        def init (self, key):
            self.bs = 16
            self.cipher = AES.new(key, AES.MODE_ECB)

        def encrypt(self, raw):
            raw = self._pad(raw)
            encrypted = self.cipher.encrypt(raw)
            encoded = base64.b64encode(encrypted)
            return str(encoded, 'utf-8')

        def decrypt(self, raw):
            decoded = base64.b64decode(raw)
            decrypted = self.cipher.decrypt(decoded)
            return str(self._unpad(decrypted), 'utf-8')
```

```
def _pad(self, s):
    return s + (self.bs - len(s) % self.bs) * chr(self.bs - len(s) % self.bs)

def _unpad(self, s):
    return s[:-ord(s[len(s)-1:])]
```

5.2 Implementing RSA Using Python

The first task is to generate the prime numbers p and q.

We must use Big Integer instead of the standard int because an integer variable cannot exceed 231 − 1 while a Big Integer can simulate arbitrary-precision integers. We can apply all the usual mathematical operations to Big Integer as well as others like mod ular arithmetic, gcd, primarily testing etc.

When constructing a Big Integer we specify a bit-length and the amount of times t, that we want the Miller-Rabin probabilistic test (Below) to run on the Big Integer, as well as supplying a random set of bits for these tests. This will generate a random integer which is probably prime with the specified bit-length.

The probability that the new Big Integer represents a prime number will exceed (1 − 1/4 t). From this we can easily generate n and m. The next step is to calculate e which must be co-prime to m, i.e. gcd(e, m) = 1. We begin by letting e = 3, if gcd(e, m) 6= 1 we let e be the next odd number.

We continue in this fashion until the gcd(e, m) = 1. The reason we only use odd numbers is because m will always be even so therefore no even number will be co-

prime to m. The Big Integer class utilizes Euclid's algorithm (Below) to calculate gcd's. We now have all the components of the public key.

We must now calculate d such that de mod m = 1. BigInteger uses the method mod Inverse to find d. de mod m = 1 ⇒ 1 − de = mk..... where k is an integer ⇒ 1 = mK + dem & e are known. This has a unique solution because m and e are co-prime - We made it so in the last paragraph. This solution is got using the Extended Euclid Algorithm (Below). We now have all the information we require to encrypt integers. We use the encryption function f(x) = y = x e mod n. BigInteger uses the method mod Pow to calculate y.

5.3 DES Algorithm

DES (and most of the other major symmetric ciphers) is based on a cipher known as the Feistel block cipher. This was a block cipher developed by the IBM cryptography researcher Horst Feistel in the early 70's. It consists of a number of rounds where each round contains bit-shuffling, non-linear substitutions (S-boxes) and exclusive OR operations. Most symmetric encryption schemes today are based on this structure (known as a feistel network). As with most encryption schemes, DES expects two inputs - the plaintext to be encrypted and the secret key.

The manner in which the plaintext is accepted, and the key arrangement used for encryption and decryption, both determine the type of cipher it is. DES is therefore a symmetric, 64 bit block cipher as it uses the same key for both encryption and decryption and only operates on 64 bit blocks of data at a time5 (be they plaintext or ciphertext). The key size used is 56 bits, however a 64 bit (or eight-byte) key is actually input. The least significant bit of each byte is either used for parity (odd for DES) or set arbitrarily and does not increase the security in any way.

All blocks are numbered from left to right which makes the eight bit of each byte the parity bit. Once a plain-text message is received to be encrypted, it is arranged into 64 bit blocks required for input. If the number of bits in the message is not evenly divisible by 64, then the last block will be padded. Multiple permutations and substitutions are incorporated throughout in order to increase the difficulty of performing a cryptanalysis on the cipher.

However, it is generally accepted that the initial and final permutations offer little or no contribution to the security of DES and in fact some software implementations omit them (although strictly speaking these are not DES as they do not adhere to)

The DES algorithm is the most popular security algorithm. It's a symmetric algorithm, which means that the same keys are used to encrypt/decrypt sensitive data.

Key length is 8 byte (64 bit). So, to encrypt/decrypt data, the DES algorithm uses an 8-byte key, but 1 byte (8 bit) for parity checking. It's a block cipher algorithm that's why the data block size of DES algorithm is 64 bit. To encrypt/decrypt data, the DES algorithm uses the Feistel structure. So, it uses some round to encrypt/decrypt data. Though data block size is 64 bit, the number of rounds will be 16 rounds.

So, it will use different subkeys for each round. so the number of subkeys will be 16 subkeys. For more info on the process of finding subkeys, you can learn more here. However, for this tutorial, we will be skipping this part.

5.4 Modes of Operation

There are different modes of operation when using the DES algorithm. If each 64 bit is encrypted or decrypted independently, then this mode is ECB.

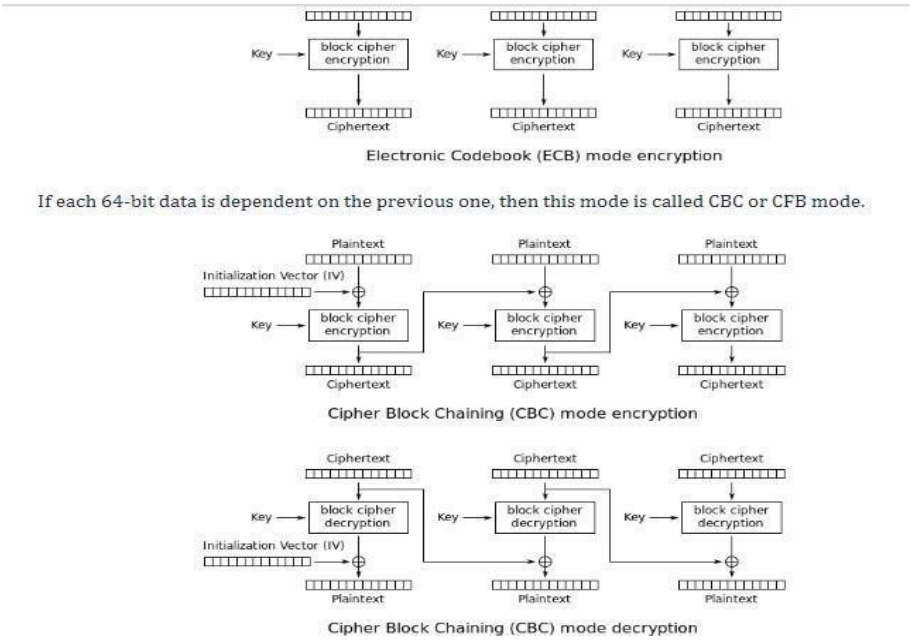


Fig.5.2. CBC mode decryption

5.5 Implementation

For implementation, we have to use a security provider. In this case, we will use the boun cycastle provider. We can use a secret key as a plaintext or byte array that will be defined by us, or we can generate a random secret key using the KeyGenerator from javax.crypto package. We will see both methods. So now, we have to add a provider

5.5.1 Sample code

```
import os
import datetime
import hashlib
```

```
import Crypto.Cipher

from flask import Flask, session, url_for, redirect, render_template, request,

abort, flash

from database import
db_connect,user_reg,owner_reg,owner_login,upload_file,owner_viewfiles,upl
oad_clo uddata,user_request,owner_request,user_lastdownload

from database import
onwer_viewdata,user_loginact,user_viewfile,user_viewfiledata,user_down,veri
fy_user,verify_user2,user_finaldown,owner_update,user_down1fromwerkzeug
.utils import secure_filename

import cv2

from AES import AESCipher

from des import des

import random

import  base64

import cv2

from stegano import lsb

from RSA import encrypt,decrypt,generate

from cloud import uploadFile,downloadFile,close

app = Flask(_name_)

app.secret_key = os.urandom(24)

@app.route("/")

def FUN_root():

    return render_template("index.html")

@app.route("/owner")

def FUN_admin():

    return render_template("owner.html")
```

```
@app.route("/ownerlogact",methods = ['GET','POST'])
def owner_logact():
  if request.method == 'POST':
    status=owner_login(request.form['username'],request.form['password'])
    if status == True:
      session['username'] = request.form['username']
      return render_template("ownerhome.html",m1="sucess")
    else:
      return render_template("owner.html",m1="Login Failed")
@app.route("/user/")
def FUN_student():
   return render_template("user.html")
@app.route("/userreg/")
def FUN_userreg():
   return render_template("userreg.html")
@app.route("/userregact", methods = ['GET','POST'])
def user_regact():
  if request.method == 'POST':
    status =user_reg
(request.form['username'],request.form['password'],request.form['dob'],request.form['e
mail'],request.form['city'],request.form['contactno'])
    if status == True:
     return render_template("user.html",m1="Success")
    else:
     return render_template("user.html",m1="Login failed")
@app.route("/userlogact",methods = ['GET','POST'])
```

```
def user_logact():
  if request.method == 'POST':
    status=user_loginact(request.form['email'],request.form['password'])
    if status == True:
      session['email'] = request.form['email']
      return render_template("userhome.html",m1="Success")
    else:
      return render_template("user.html",m1="Login Failed")
@app.route("/userhome")
def user_home():
    return render_template("userhome.html")
@app.route("/vf/")
def user_vf():
     viewfile = user_viewfile(session['email'])
return render_template("vf.html", viewfiledata = viewfile)
```

Here, we can see that we are not using IV for ECB. We created a cipher instance and init this with DES parameters. For encryption, we are using the Cipher mode ENCRYPT_MODE, and for decrypt, we are using DECRYPT_MODE. Other parameters remain the same for both encryption and decryption. The key should be the same for encryption and decryption

5.6 Input and Output Screen Design (Snapshots)

The snapshots of our project are in this format. They are:

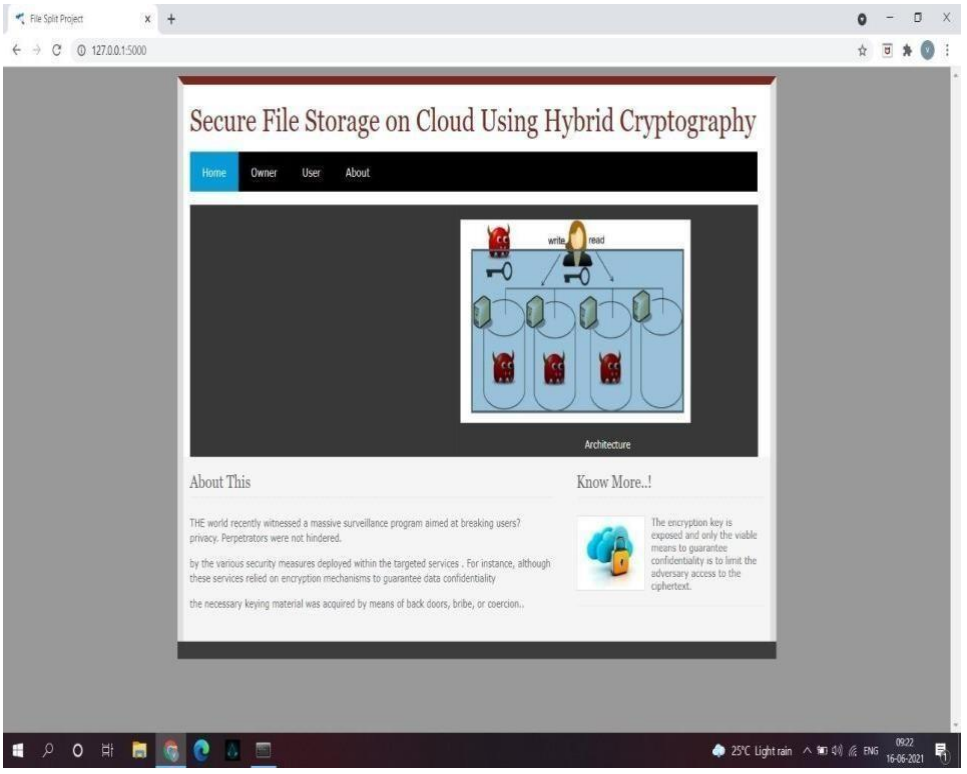


Fig.5.3. Home Page

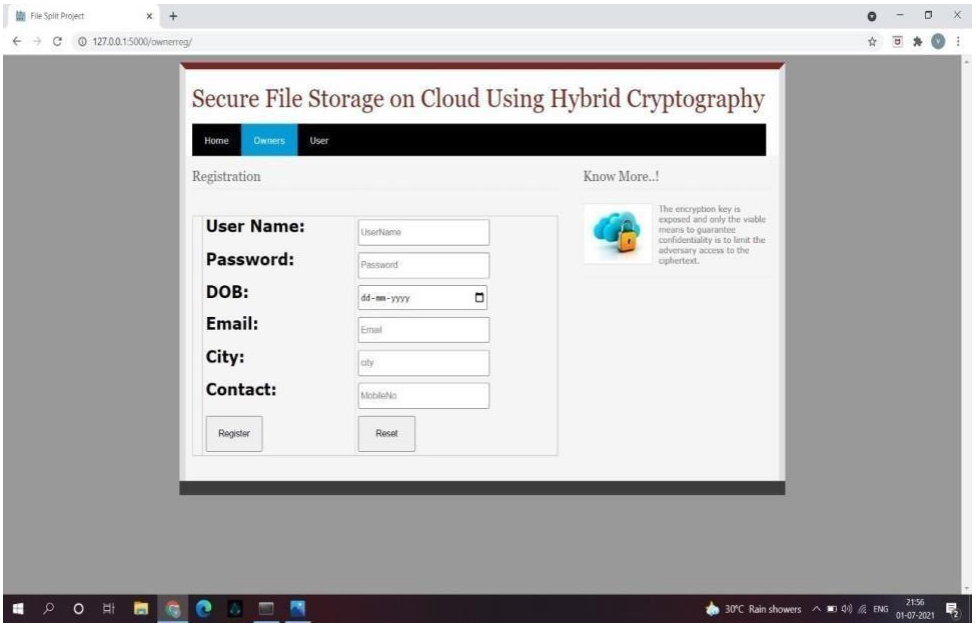


Fig.5.4. Owner Registration Page

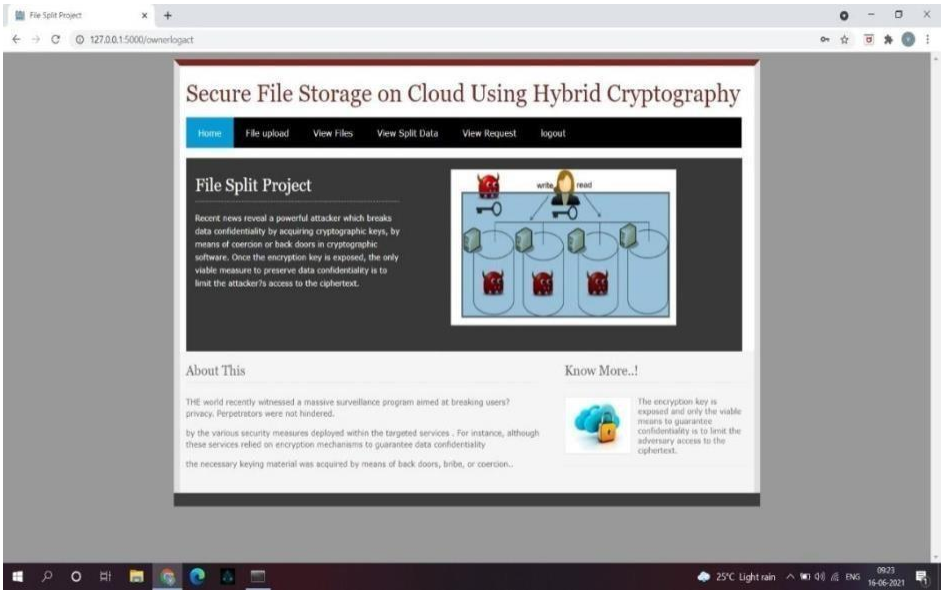


Fig.5.5. Owner Login Page

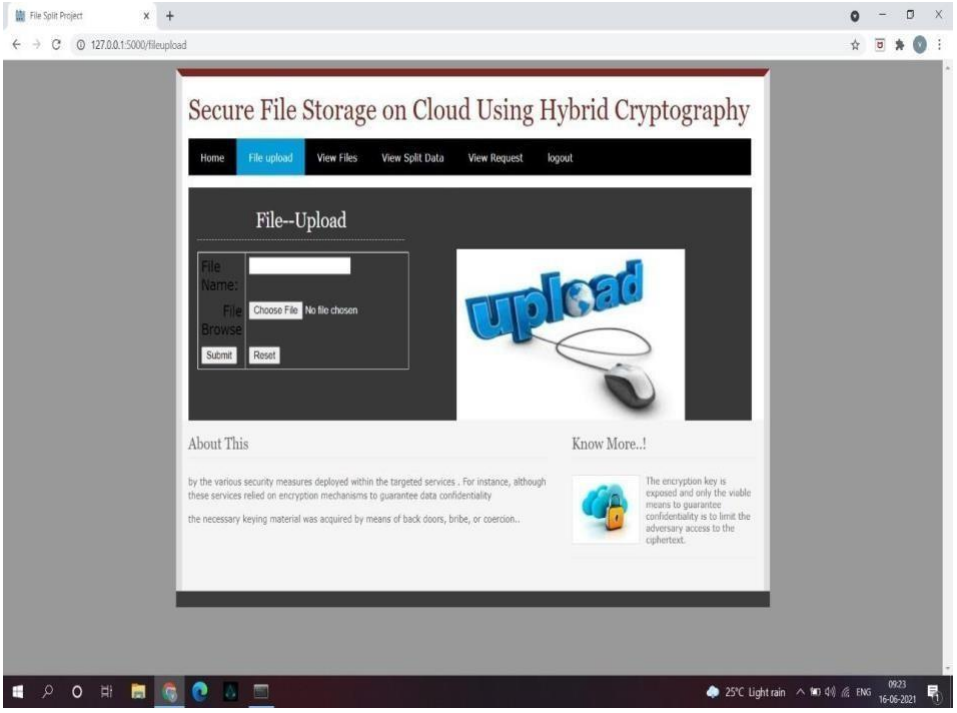


Fig.5.6. File Split

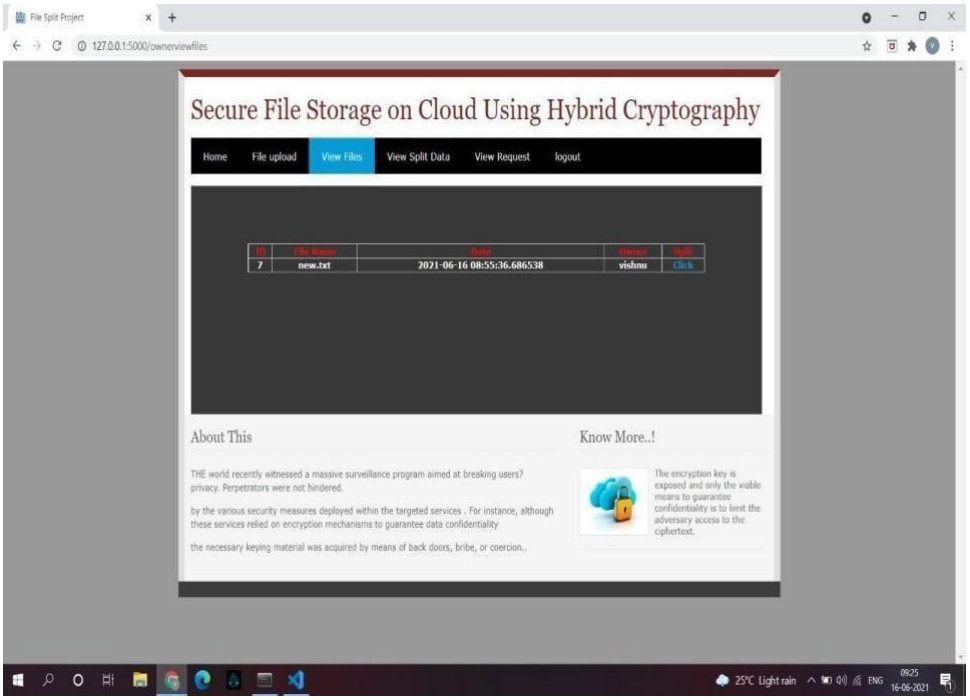


Fig.5.7. View Files

.

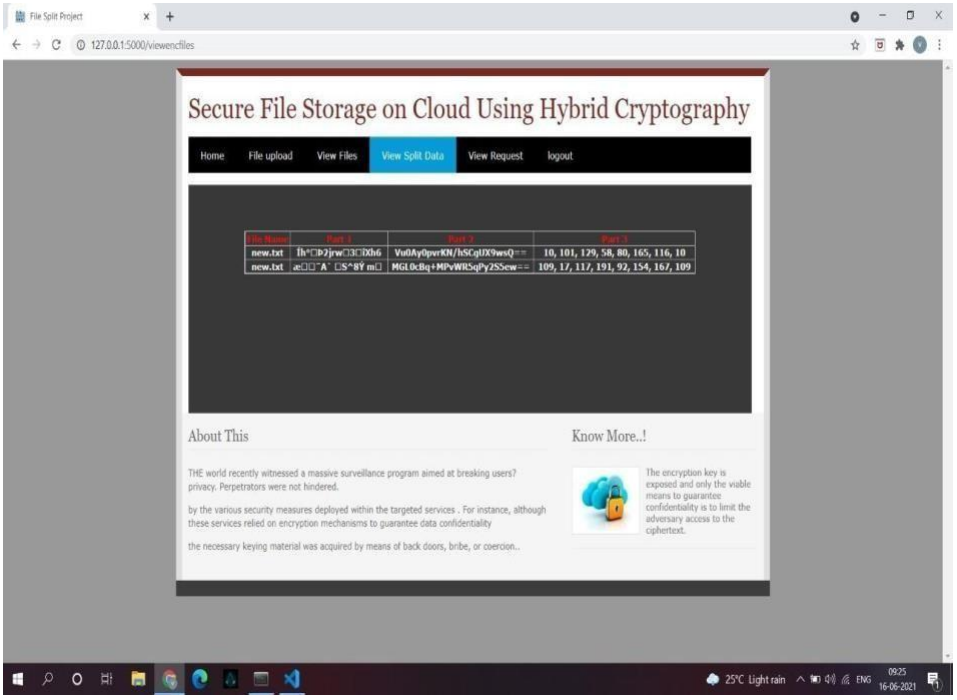


Fig.5.8. View Split Data

.

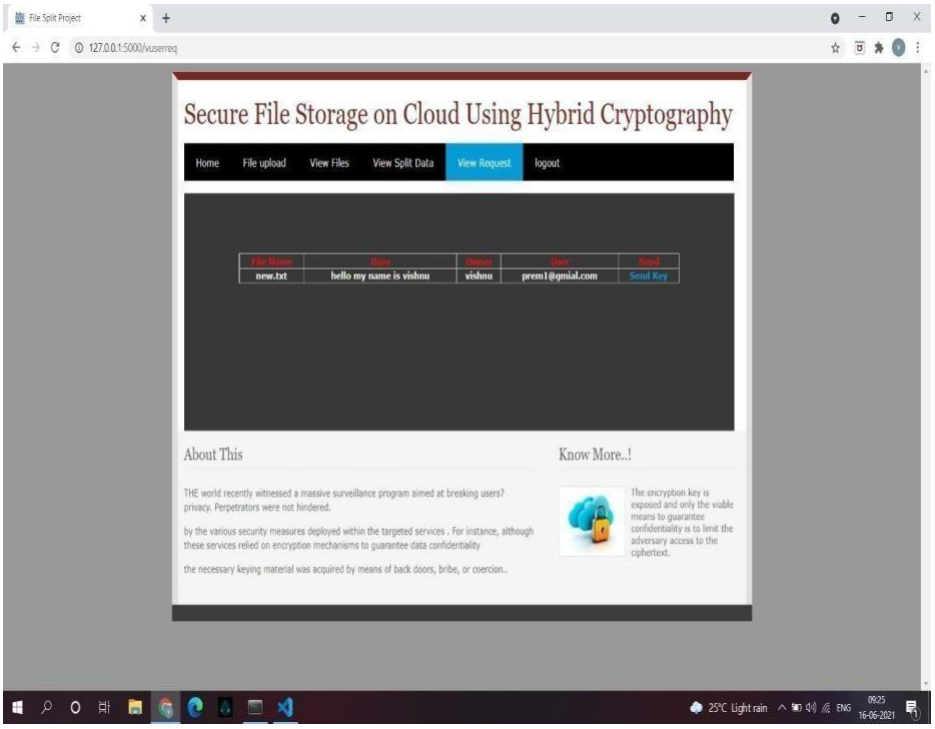


Fig.5.9. View Request

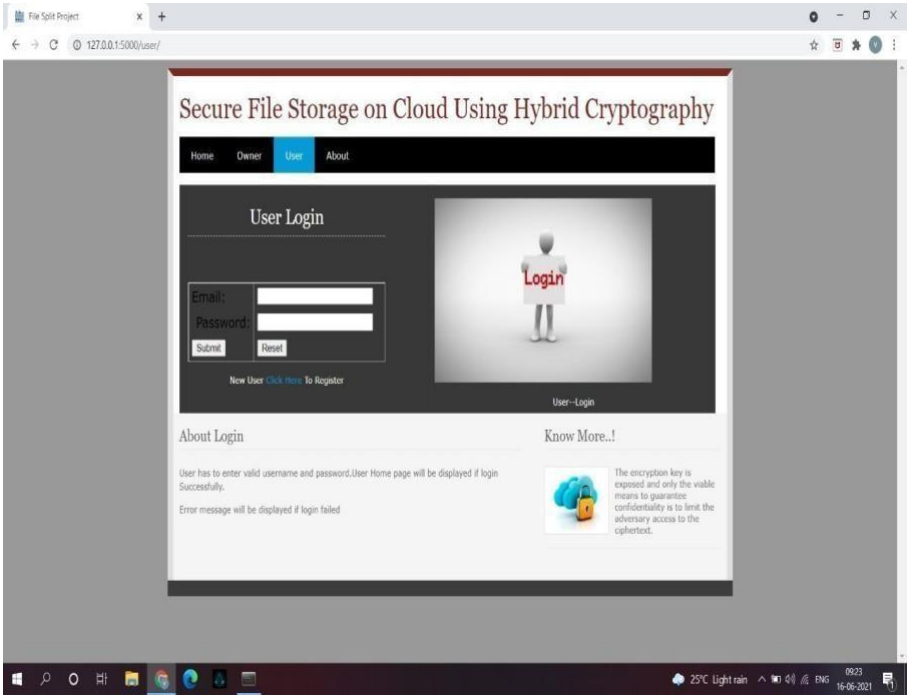


Fig.5.10. User Login

.

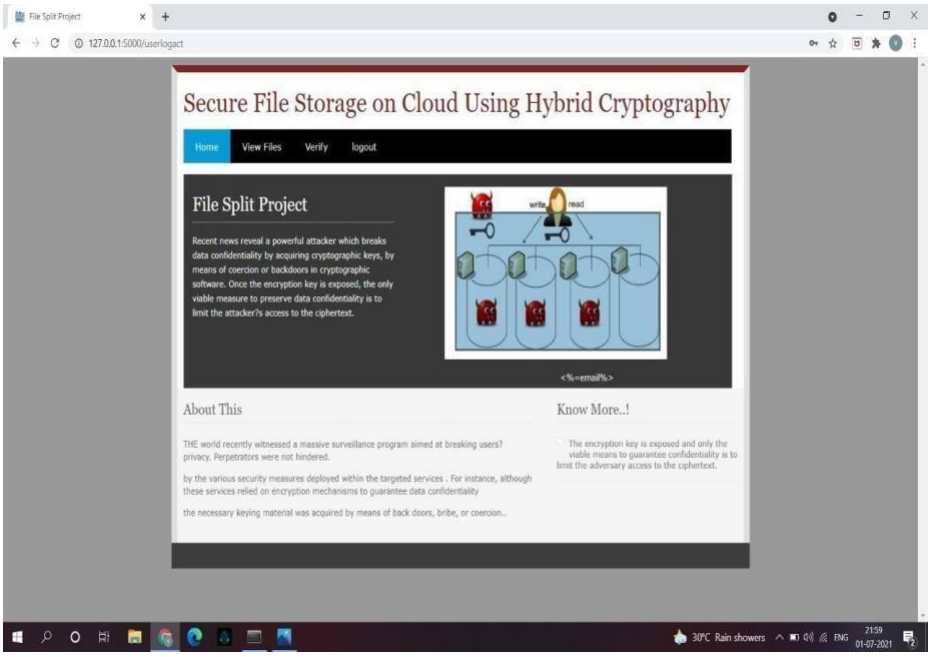


Fig.5.11. User Home Page

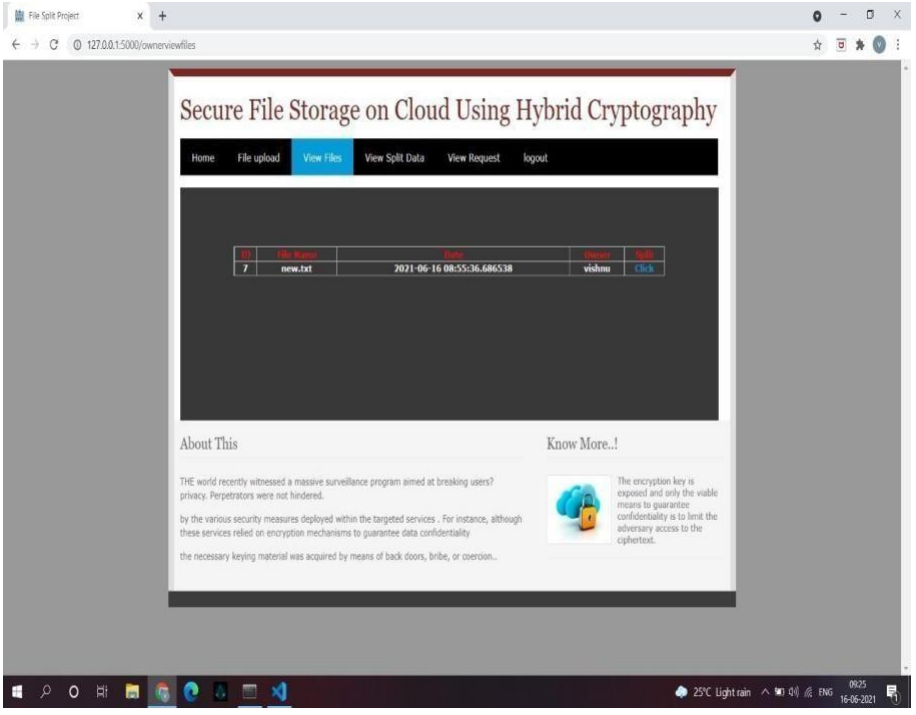


Fig.5.12. View Files

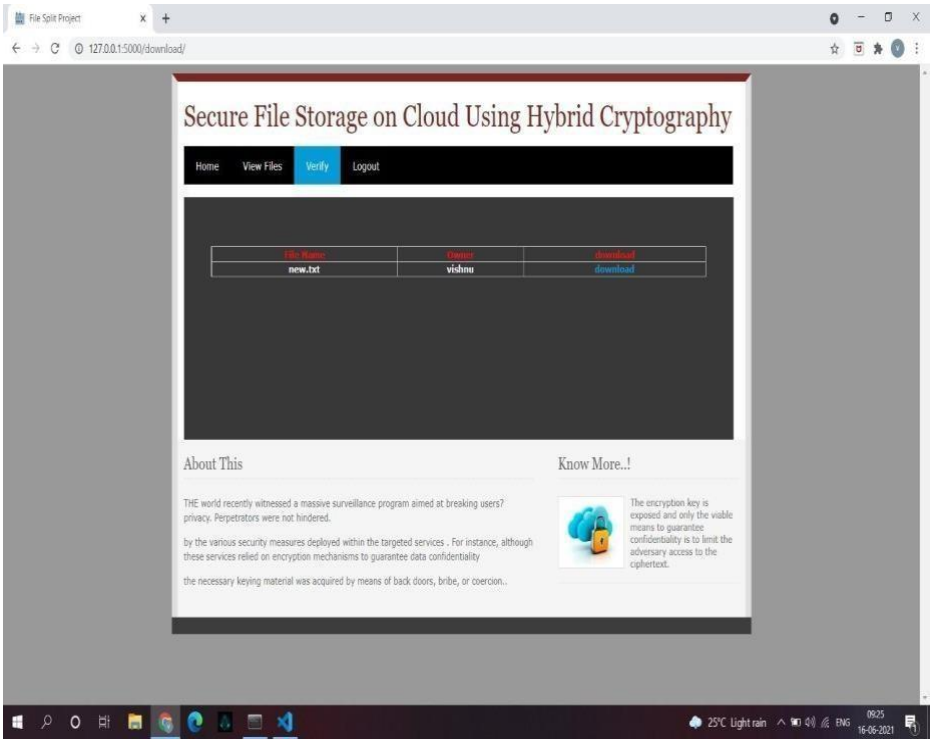


Fig.5.13. User Verify

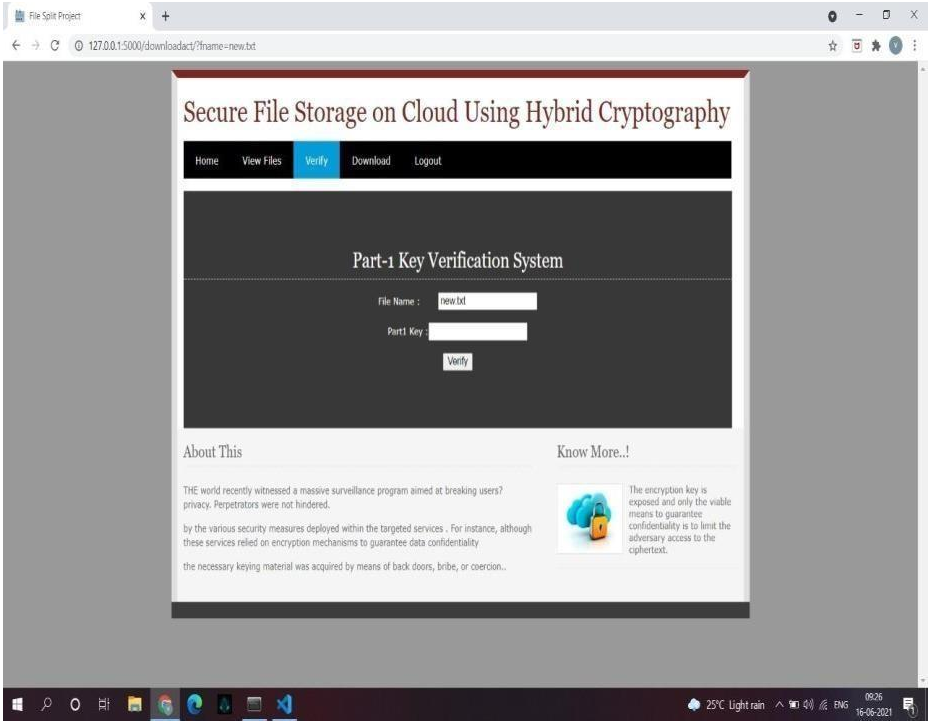


Fig.5.14. Decrypt File Part1

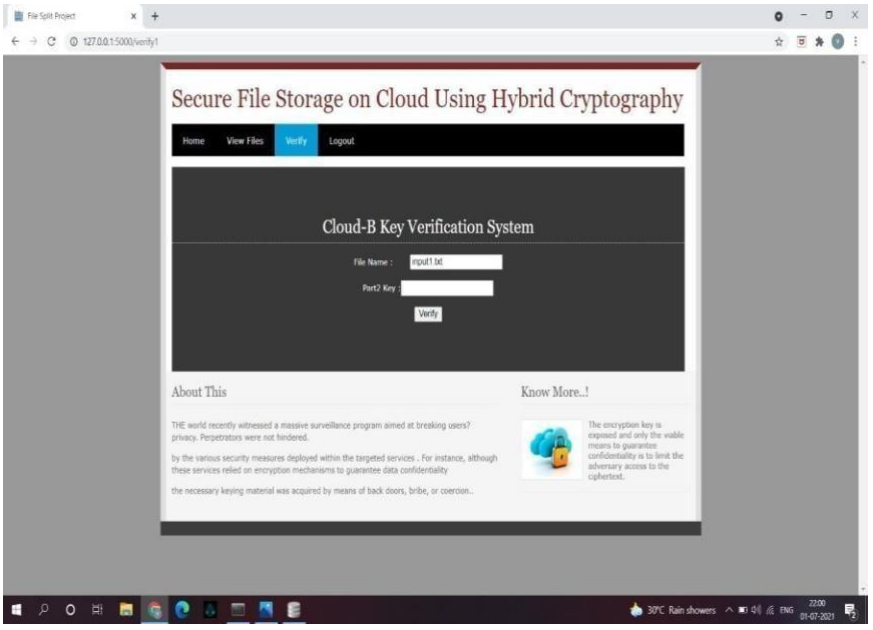


Fig.5.15. Decrypt File Part

CHAPTER-6

TESTING & VALIDATION

6.1 Introduction

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

6.2 Design of Test Cases and Scenarios

The completion of a system will be achieved only after it has been thoroughly tested. Though this gives a feel the project is completed, there cannot be any project without going through this stage. Hence in this stage it is decided whether the project can undergo the real time environment execution without any break downs. Therefore a package can be rejected even at this stage.

6.2.1 Testing Methods

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

Black box testing - Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification- based testing.

White box testing - White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithm (and the code that implement these). White box testing methods can also be used to evaluate the

completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

compliant strategy is a smart option when lacking skills and time in the team to create an approach.

1. **Product.** Some products such as contract development software and weapons systems tend to have requirements that arc well –specified. This could lead to synergy with an analytical strategy that is requirements based.

2. **Business.** Business considerations and strategy are often important. If using legacy system as a model for a new one, one could use a model -based strategy.

3. **Regulations.** At some instances, one may not only have to satisfy stakeholders, hut regulators as well. In this case, one may require a methodical strategy which satisfies these regulators.

You must choose testing strategies with an eye towards the factors mentioned earlier, the schedule, budget, and feature constraints of the project and the realities of the organization and its politics.

Module Testing - To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

Integration Testing - fter the module testing, the integration testing is applied. When linking the there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

Acceptance Testing- When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

6.4 Validation

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected, and a final series of software tests validation testing may begin.

Reasonable expectation is defined in the software requirement specification a document that describes all user-visible attributes of the software. The specification contains a section titled "Validation Criteria". Information contained in that section forms the basis for a validation testing approach.

CONCLUSION

The main goal is to securely store and access data in cloud that is not controlled by the owner of the data. We exploit the technique of AES, DES , RSA cryptography encryption to protect data files in the cloud. Two part of the cloud server improved the performance during storage and accessing of data. These Encryption algorithm used for encryption is another advantage to improve the performance during encryption and decryption process. We assume that this way of storing and accessing data is much secure and have high performance. Our efforts are going on to solve the problem of security issues by using single encryption algorithm of data in cloud computing environment.

BIBLIOGRAPHY

[1] Peter Mel and Tim Grace, "The NIST Definition of Cloud Computing", NIST, 2010.

[2] Achill Buhl, "Rising Security Challenges in Cloud Computing", in Proc. of World Congress on Information and correspondence Technologies ,pp. 217-222, Dec. 2011.

[3] Srinivasarao D et al., "Breaking down the Superlative symmetric Cryptosystem Encryption Algorithm", Journal of Global Research in Computer Science, vol. 7, Jul. 2011

[4] Tingyuan Nye and Tang Zhang "An investigation of DES and Blowfish encryption algorithm", in Proc. IEEE Region 10 Conference, pp. 1-4 ,Jan. 2009.

[5] Jitendra Singh Adam et al.," Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering ,vol. 2,Aug. 2012.