**Image Steganography**

**A PROJECT REPORT**

**Submitted By**

**RITIKA KHANDELWAL**

**University Roll No 1900290149083**

**Submitted in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Ms. NEELAM RAWAT**

### ASSOCIATE PROFESSOR



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(June 2021)**

**DECLARATION**

I hereby declare that the work presented in this report entitled “IMAGE STEGANOGRAPHY", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

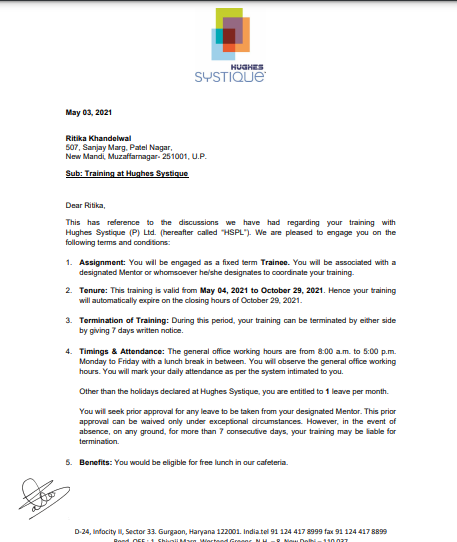
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

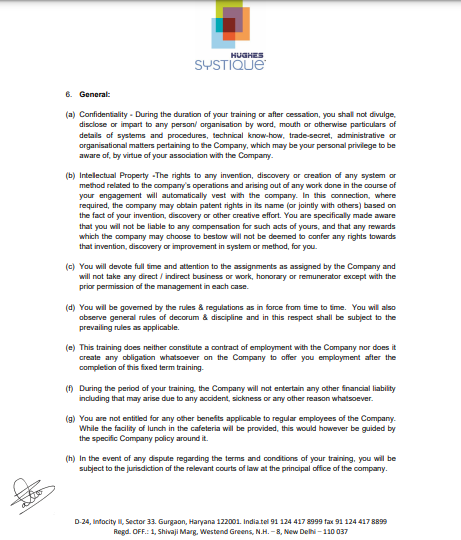
Name : Ritika Khandelwal

Roll. No. : 1900290149083

Branch : Master of Computer Applications

**COMPANY CERTIFICATE**

****

****

**CERTIFICATE**

Certified that **Ritika Khandelwal (Roll No. 1900290149083)** have carried out the project work having “**Image Steganography**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**RITIKA KHANDELWAL**

**University Roll No 1900290149083**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

**Ms. Neelam Rawat**

Associate Professor

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Ajay kr. Shrivastava**

**Professor & Head, Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**ABSTRACT**

Steganography is a form of security technique through obscurity, the science and art of hiding the existence of a message between sender and intended recipient. Data hiding is the art of hiding data for various purposes such as; to maintain private data, secure confidential data and so on. Securely exchange the data over the internet network is very important issue. Steganography has been used to hide secret messages in various types of files, including digital images, audio and video. Different applications have different requirements of the steganography technique used. The main motive of this technique is to hide the important information byte by byte in an image pixles. The input image can be of any size but must be greater than the size of a input message. This process is usefull when we to send some important information to somebody without anybody else noticing the secret information. After the transmission the reciever should also use the same technique that was used for encoding to decode the message. This process hides the data in such a manner that there will be no noticible changes in an image. It basically hides the existence of the data. Each and every byte of data is converted into its binary equivalent and then are processed to get stored in the iamge pixel. The language used for this process is python with some of its in-built libraries. For the image we use PIL (i.e. PYTHON IMAGING LIBRARY). Steganography usually deal with the way of hide the existence of communicated data in such a way that it remains confidential. It maintain secrecy between the two communicating bodies. Secrecy are achieved In the image steganography, by embedded data into the cover images and generating a stego-images.

**ACKNOWLEDGEMENTS**

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Ms. Neelam Rawat**  for his guidance, help and encouragement throughout my research work .Their enlightening, ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Ritika Khandelwal**

**TABLE OF CONTENTS**

Page no.

Declaration ii

Company Certificate iii

Certificate vi

Abstract vii

Acknowledgement viii

**CHAPTER 1: INTRODUCTION 1-6**

* 1. Project Description 1
  2. Project Purpose 2
  3. Problem Statement 2
  4. Project Scope 2
  5. What is Steganography? 3
  6. Types of Steganography 4
  7. Steganography Ways 5

**CHAPTER 2: LITERATURE REVIEW 7-10**

2.1 AES Algorithm 7

2.2 Least Significant Bit 7

2.3 Steganography 9

2.4 Python 10

**CHAPTER 3: REQUIREMENT ANALYSIS 11-14**

3.1 Functional Requirement 11

3.2 Non-Functional Requirement 12

3.3 System Requirement 12

**CHAPTER 4: FEASIBILITY STUDY 15-17**

4.1 Introduction 15

4.2 Technical Feasibility 16

4.3 Economical Feasibility 16

4.4 Operational Feasibility 17

4.5 Social and Behavioral Feasibility 17

4.6 Legal Feasibility 17

**CHAPTER 5: SYSTEM DESIGN 18-20**

5.1 Class Diagram 18

5.2 Use Case Diagram 19

5.3 Data Flow Diagram 20

**CHAPTER 6: BRIEF ALGORITHM AND IMPLEMENTATION 21-46**

6.1 LSB 21

6.2 How LSB Technique Works? 21

6.3 Hiding text inside an image using Python 22

**CHAPTER 7: TESTING 47-53**

7.1 Introduction 47

7.2 Unit Testing 49

7.3 Validation Testing 50

7.4 Output Testing 50

7.5 Integration Testing 51

7.6 User Acceptance Testing 53 7.7 Black Box Testing 53

7.8 White Box Testing 53

**CHAPTER 8: SNAPSHOTS 55-63**

**CHAPTER 9: CONCLUSION AND FUTURE SCOPRE 64-65**

9.1 CONCLUSION 64

9.2 FUTURE SCOPE 65

**LIST OF FIGURES**

1.1 BASIC MODEL OF STEGANOGRAPHY 3

5.1 CLASS DIAGRAM 18

5.2 USE CASE DIAGRAM 19

5.3 CONTEXT DIAGRAM 20

8.1 HOME PAGE 55

8.2 GUIDELINES FOR THE PROJECT 56

8.3 CHOOSE THE FILE 57

8.4 FIND THE IMAGE 58

8.5 OUTPUT OF SEARCHED IMAGES 59

8.6 ENTER THE ENCODING TEXT 60

8.7 SELECT THE IMAGE 61

8.8 DECODE THE FILE 62

8.9 RESULT 63

**Chapter – 1**

**INTRODUCTION**

**1.1 PROJECT DESCRIPTION**

In this project, we propose to develop a system to hiding data by using "STEGANOGRAPHY" technique as I used many methods stands on some techniques to have at the back-end a software for hiding data based on hiding algorithms. After studying the data hiding algorithms we found many ways to hiding data by using the multimedia files and the main question for me was "Where hidden data hides?" as we found by our search to know where the data hides it's important to know what is the file type of the data that it shall be hidden and the cover file type so it is possible to alter graphic or sound files slightly without losing their overall viability for the viewer and listener.

With audio, you can use bits of file that contain sound not audible to the human ear. With graphic images, you can remove redundant bits of color from the image and still produce a picture that looks intact to human eye and is difficult to discern from its original. It is in those bits that stego hides its data.

By the final of our research we developed a software uses an algorithm, to embed data in an image; The purposed system is called "Steganography", the aim of this project is to encrypt the data; the meaning of encrypt is to hide the data over an image using different steganographic algorithms, in this system LSB is the algorithms that we use to hiding the data.

**1.2 PROJECT PURPOSE**

In this project we primarily concentrated on the data security issues when sending the data over the network using steganographic techniques.

The main objectives of our project are to product security tool based on steganography techniques to hider message carried by stego-media which should not be sensible to human beings and avoid drawing suspicion to the existence of hidden message.

**1.3 PROBLEM STATEMENT**

This project addresses the security problem of transmitting the data over internet network, the main idea coming when we start asking that how can we send a message secretly to the destination? The science of steganography answers this question. Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions.

In this document, we proposed some methods and algorithms of an image steganography system to hide a digital text of a secret message.

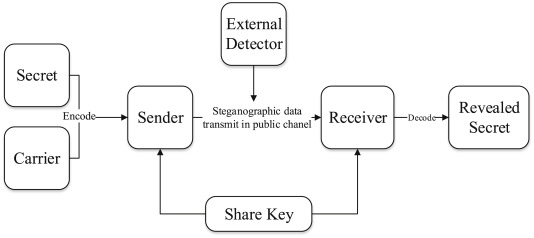
**1.4 PROJECT SCOPE**

Our project scope is developed for hiding information in any image file to ensure the safety of exchange the data between different parties and provide better security during message transmission.

The scope of the project is implementation of steganography tools for hiding information includes any type of information file and image files and the path where the user wants to save image and extruded file. We will use LSB technique; the proposed approach is to use the suitable algorithm for embedding the data in an image files; we will show a brief of this algorithm that we used to hiding data.

**1.5 WHAT IS STEGANOGRAPHY?**

Steganography is the secret process of which nobody can known of except the one who is encoding the secret message inside the image(i.e. Sender) and the other for whom the message is being encoded(i.e. reciever). In other words it is also known as the study of unperceivable communication. Steganography is the process in which the image is input by the user and after encoding it with the secret data a STEGO-image is generated. Which is slighlty change from the original image but the difference in unnoticible.In the current time, Steganography is used in many places and one of the important example is Army duty stations. This is the place where steganography is the only safe medium to use, because they don‟t want there secret information to be shared with the ones across the border.

****

1Fig. 1.1 Basic Model Of Steganography

**ADVANTAGES OF STEGANOGRAPHY**

* Difficult to detect by someone who doesn‟t know about this communication, whether the transaction is taking place or not.
* In order to view the hidden message user should have to enter the secret key which was generated using the steganographic algorithm.
* Easy to implement with the help of LSB algorithm.
* Can be used in arm forces and intelligence agencies.

**DISADVANTAGES OF STEGANOGRAPHY**

* Sender and receiver should have the same software to encrypt or decrypt message.
* If encryption key is lost, then important information will be lost too.
* The cover image used should have size greater than the message bytes.
* Hidden information can be viewed by attackers if proper encryption algorithm used.

**1.6 TYPES OF STEGANOGRAPHY**

**Text steganography**

This is a method in which we hide the secret information inside the other text file, the main motive in this process is to share our secret information with the help of another normal message which doesn‟t get suspicious.

**Audio steganography**

In sound steganography, we insert mystery messageiinto digitized sound sign which output slight modifying of double grouping of the relating sound

document. There are a few strategies are accessible for sound steganography. We will have a short report presentation on some of sound.

**Video steganography**

It is a method of steganography in which we hide the secret infromation inside the video file. In this discrete co-sine transforms is used to alter the digit which are used to hide the info in each of the image in the videos, which can only be detected by some software but is unnoticeable by the humans eye.

**1.7 STEGANOGRAPHY WAYS**

As it is known there is much communication between people and organizations through the use of the phone, the fax, computer communications, radio, and of course all of these communications should be secure.

There are basically three Steganography types: -

* Pure Steganography.
* Secret key Steganography.
* Public key Steganography.

**Pure Steganography**

Pure Steganography is a Steganography system that doesn't require prior exchange of some secret information before sending message; therefore, no information is required to start the communication process: the security of the system thus depends entirely on it secrecy.

In most applications, pure Steganography is preferred, since no stego-key must be shared between the communication partners, although a pure Steganography protocols don't provide any security if an attacker knows the embedding method.

**Secret Key Steganography**

A secret key Steganography system is similar to a symmetric cipher, where the sender chooses a cover and embeds the secret message into the cover using a secret key. If the secret key used in the embedding process is known to the receiver, he can reverse the process and extract the secret message.

Anyone who doesn't know the secret key should not be able to obtain evidence of the encoded information.

**Public Key Steganography**

Public key Steganography does not depend on the exchange of a secret key. It requires two keys, one of them private (secret) and the other public: the public key is stored in a public database, whereas the public key is used in the embedding process. The secret key is used to reconstruct the secret message.

**Chapter – 2**

**Literature Review**

**2.1 AES ALGORITHM**

We propose a multilayered secure scheme to transfer sensitive text over an unreliable network. The secret text is first encrypted using the AES algorithm. The cipher text produced is hidden in an audio file. The audio file is in turn encrypted in parallel using the concept of Shamir's technique based on CRT to maximize resource utilization. [1]

 In this paper the embedded information is applied as text. Before embedding the text in image, text is encrypted using Advanced Encryption Standard (AES) algorithm. The text can be a sentence or a key with alphabetic words having the length of 8 characters. Using Least Significant Bit (LSB) method, the encrypted text is embedded into the “LL sub-band wavelet decomposed image”. [8]

**2.2 LEAST SIGNIFICANT BIT (LSB)**

Develop a novel method to embed important data in the host image so that the interceptors will not notice about the existence of the data. The basic concept of the proposed method is by simple LSB substitution. [2]

This paper proposes an information hiding technique using the concept of the modified least significant bit substitution (MLSBS). The proposed technique utilizes the XOR feature of the original image (OI) pixel for embedding the secret bits. At the embedding end, the last three LSBs of the OI image pixels are

considered for finding the featured bits. The obtained bits are replaced with the least significant bit (LSB) of each pixel. [4]

Owing to the inefficiency to hide large volume of secret data for the reversible data hiding (RDH) image steganography approaches, we propose two improved RDH based approaches, such as (1) improved dual image based least significant bit (LSB) matching with reversibility, and (2) n-rightmost bit replacement (n-RBR) and modified pixel value differencing (MPVD). [5]

In this paper, an efficient visually meaningful image compression and encryption (VMICE) scheme is proposed by combining compressive sensing (CS) and Least Significant Bit (LSB) embedding. First, the original image (Iorig) is compressed and encrypted into a secret image (Isec) by CS and Zigzag confusion. Next, dynamic LSB embedding is utilized to randomly embed Isec into a separate carrier image (Icar) to create the final visually meaningful (VM) cipher image (Iciph), which is the same size as Iorig. [6]

Universal quantitative steganalysis suffers from the curse of dimensionality and requires large number of instances (training samples) to produce qualitative results. In this paper, a universal quantitative steganalyser for spatial LSB based algorithms using reduced number of instances and features is proposed. A rich combination of global and local features models are employed as core features. [11]

In this paper, a new steganalysis technique is proposed on the basis of statistical observations on difference image histograms for the reliable detection of least significant bit (LSB) steganography. A physical quantity is derived from the transition coefficients between difference image histograms of an image and its processed version produced by setting all bits in the LSB plane to zero. [13]

**2.3 STEGANOGRAPHY**

Various image steganography methods have been developed to improve the stego image quality, i.e. imperceptibility, capacity, and security. This research proposes an adaptive method that can select the most optimal pattern to minimize the error ratio due to message embedding. [3]

Image Steganography is a methodology used to hide the messages inside of the cover image. Initially, the secret information is encrypted by using the RSA Algorithm. Then the encrypted secret information is hidden in the Least Significant Bit (LSB) of the different components of the color image in such a way that the original quality of the image to be minimized. [7]

The usage of embedded systems is omnipresent in our everyday life, e.g., in smartphones, tablets, or automotive devices. These devices are able to deal with challenging image processing tasks like real-time detection of faces or high dynamic range imaging. [10]

This paper presents a novel method for detection of LSB matching steganography in grayscale images. This method is based on the analysis of the differences between neighboring pixels before and after random data embedding. In natural images, there is a strong correlation between adjacent pixels. This correlation is disturbed by LSB matching generating new types of correlations. The presented method generates patterns from these correlations and analyzes their variation when random data are hidden. [12]

Steganography is the art of encoding/embedding secret information in cover media in such a way so as not to provoke an eavesdropper's suspicion. The primary purpose of this paper is to provide three levels of security, first is provided by

complementing the secret message, second by hiding complemented secret message in cover image pixels that are selected randomly by using pseudo random number generator, third by using inverted bit LSB method2 as stegnographic technique rather than simple LSB, thus, reduces the chance of the hidden message being detected. [14]

**2.4 PYTHON**

Python has become the predominant programming language for general data science and analytics. In this chapter, we will discuss the core concepts and design goals of the Python programming language. We will also cover the core functionality of the language, key data structures, and important packages. [9]

In this paper, we proposed a novel method to embed a series of ternary secret data into a cover image based on an improved Least-Significant-Bit (LSB) scheme using the modulo three strategy. Our new method can hide two ternary numbers into each grayscale pixel, normally only modify the two LSBs of the pixel, while it may cause overflow/underflow and a carry/borrow. [15]

**Chapter – 3**

**REQUIREMENTS ANALYSIS**

**3.1 FUNCTIONAL REQUIREMENTS**

Functional requirements are the requirements that that deﬁne speciﬁc behavior or function of the system.

* Login:

Login function will authenticate the sender if username and password ares correct Otherwise it will exit the system.

* Secret Text Message File:

In this ﬁle you will have to write secret message to hide or you can select any text ﬁle of secret message.

* Cover Image:

Cover Image is the image is to be selected in which secret text message can be hidden.

* Sender:

In this Sender send this stego image ﬁle to intended recipient to which he does want to communicate.

* Receiver:

In this receiver receives the stego image and opens in decryption option for getting hidden text message inside that image.

**3.2 NON- FUNCTIONAL REQUIREMENTS**

* Safety Requirements:

Sender and Receiver should make sure that only they are having the same software to encrypt and decrypt data inside image. Both should take care of eavesdropping.

* Security Requirements:

We are going to develop a software in which embedding secret text data in image.Only sender and receiver should be aware of encrypted ﬁle. User should not unfold the message regarding sent image as well as receiver information.

* Software Quality Attributes:

The Quality of the software is maintained in such a way that only sender and receiver can communicate through image. There is no probability of knowing secret image.

**3.3 SYSTEM REQUIREMENTS**

**Software Requirements**

* + Operating System: Windows 10.
  + Front End: Python.

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) as well as its [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.

Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing) and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management). It also features dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as [MIME](https://en.wikipedia.org/wiki/MIME) and [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) are supported. It includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary-precision decimals, manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing).

As of March 2021, the [Python Package Index](https://en.wikipedia.org/wiki/Python_Package_Index) (PyPI), the official repository for third-party Python software, contains over 290,000 packages with a wide range of functionality, including:

* [Automation](https://en.wikipedia.org/wiki/Automation)
* [Databases](https://en.wikipedia.org/wiki/Databases)
* [Graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interfaces)
* [Machine learning](https://en.wikipedia.org/wiki/Machine_learning)
* [Multimedia](https://en.wikipedia.org/wiki/Multimedia)
* [Computer Networking](https://en.wikipedia.org/wiki/Computer_networking)
* [Test frameworks](https://en.wikipedia.org/wiki/Test_framework)
* [Web frameworks](https://en.wikipedia.org/wiki/Web_framework)

Tool: Microsoft Visual Studio

**Microsoft Visual Studio** is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program), as well as [websites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Machine_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

Visual Studio includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (the [code completion](https://en.wikipedia.org/wiki/Code_completion) component) as well as [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring). The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a [code profiler](https://en.wikipedia.org/wiki/Profiling_(computer_programming)), designer for building [GUI](https://en.wikipedia.org/wiki/GUI) applications, [web designer](https://en.wikipedia.org/wiki/Web_designer), [class](https://en.wikipedia.org/wiki/Class_(computing)) designer, and [database schema](https://en.wikipedia.org/wiki/Database_schema) designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for [source control](https://en.wikipedia.org/wiki/Source_control) systems

**Hardware Requirements**

* Minimum hardware requirements: Pentium 3166 HZ or Higher 128 mb RAM
* Intell I7 4.80 GHZ 8GB Ram.

**Chapter – 4**

**FEASIBILITY STUDY**

**4.1 INTRODUCTION**

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it

.The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is there after designed and developed. The feasibility study takes in to consideration, the risks involved in the project development beforehand. Therefore in this phase we have to do feasibility study which is the test of the website according to its work ability, impact on the organization, ability to meet user need and effective use of resources.

We do the feasibility study for website to analyze the risks, costs and benefits relating to economics, technology and user organization. There are several types of feasibility depending on the aspect they cover. Import of these includes:

**4.2 TECHNICAL FEASIBILITY**

This is an important outcome of preliminary investigation. It comprise of following questions:-

Can the work of project be one with the current equipment, existing software and available man power resource?

If Technology is required what are the possibilities that it can be developed?

We can strongly say that it is technically feasible, since there will not be much difficulty in getting resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization.

* 1. **ECONOMICAL FEASIBILITY**

It deals with question related to the economy. It comprise of the following questions:-

Are there sufficient benefits in creating the system to make the cost acceptable?

Are the costs of not creating the system so great that the project must be undertaken?

Development of this application is highly economically feasible. The organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. Even after the development, the organization will not be in condition to invest more in the organization.

**4.4 OPERATIONAL FEASIBILITY**

The operational feasibility consists of the following activity:-

Will the system be useful if it is developed &implemented?

No doubt that this prediction system will be very useful as it predicts the diabetes (chronic disease) with some alert message, precautions and suggestions.

**4.5 SOCIAL AND BEHAVIORAL FEASIBILITY**

It deals with the various issues related to the human behavior like:-

Whether the user be able to adapt a new change or not?

Whether the ambiance we are providing suits the user or not?

**4.6 LEGAL FEASIBILITY**

It deals with the question related to the legal issues. It comprise of the following questions:-

* Contract Signing
* Software License agreement
* Issues related to cyber laws.
* Legal issues relating to the man power contract.

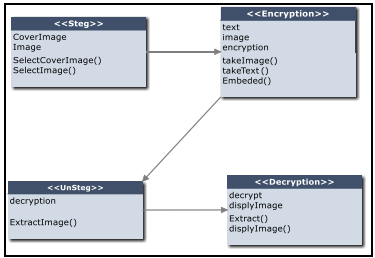
**Chapter – 5**

**SYSTEM DESIGN**

Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software is determined, software developers will design or employ designers to develop a plan for a solution. It includes construction component and algorithm implementation issues which shown in as the architectural view.

**5.1 CLASS DIAGRAM**

A class diagram is a picture for describing generic descriptions of possible systems. Class diagrams and collaboration diagrams are alternate representations of object models. Class diagrams contain classes and object diagrams contain objects, but it is possible to mix classes and objects when dealing with various kinds of metadata, so the separation is not rigid.

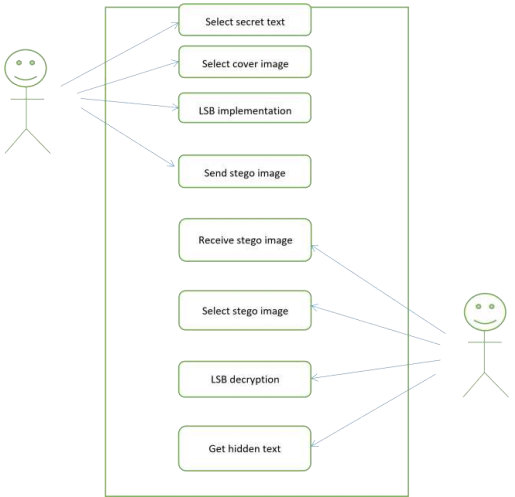
****

**Fig 5.1-Class Diagram**

## 5.2 USE CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

* Scenarios in which your system or application interacts with people, organizations, or external systems
* Goals that your system or application helps those entities (known as actors) achieve

****

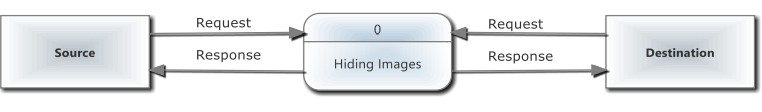
**Fig 5.2-Use Case Diagram**

**5.3 DATA FLOW DIAGRAM**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

****

**Fig 5.3-Context Diagram**

**Chapter – 6**

**BRIEF ALGORITHM IMPLEMENTATION**

**6.1 LSB (Lease significant bit)**

We can describe a digital image as a finite set of digital values, called pixels. Pixels are the smallest individual element of an image, holding values that represent the brightness of a given color at any specific point.

So we can think of an image as a matrix (or a two-dimensional array) of pixels which contains a fixed number of rows and columns.

Least Significant Bit (LSB) is a technique in which the last bit of each pixel is modified and replaced with the secret message’s data bit. If we change MSB it will have a larger impact on the final value but if we change the LSB the impact on the final value is minimal, thus we use least significant bit steganography.

## ****6.2 HOW LSB TECHNIQUE WORKS?****

Each pixel contains three values which are Red, Green, Blue, these values range from **0 to 255**, in other words, they are 8-bit values. [4] Let’s take an example of how this technique works, suppose you want to hide the message “**hi**” into a **4x4** image which has the following pixel values:

**[(225, 12, 99), (155, 2, 50), (99, 51, 15), (15, 55, 22),(155, 61, 87), (63, 30, 17), (1, 55, 19), (99, 81, 66),(219, 77, 91), (69, 39, 50), (18, 200, 33), (25, 54, 190)]**

Using [the ASCII Table](http://www.asciitable.com/), we can convert the secret message into decimal values and then into binary: **0110100 0110101.**Now, we iterate over the pixel values one by one, after converting them to binary, we replace each least significant bit with that message bits sequentially (e.g 225 is 11100001, we replace the last bit, the bit in the right (1) with the first data bit (0) and so on).This will only modify the pixel values by +1 or -1 which is not noticeable at all.

The resulting pixel values after performing LSBS is as shown below:

**[(224, 13, 99),(154, 3, 50),(98, 50, 15),(15, 54, 23),(154, 61, 87),(63, 30, 17),(1, 55, 19),(99, 81, 66),(219, 77, 91),(69, 39, 50),(18, 200, 33),(25, 54, 190)]**

# ****6.3 HIDING TEXTINSIDE AN IMAGE USING PYRTHON****

In this section, we can find a step-by-step of the hide and reveal process using Python code.

**Step 1:** Import all the required python libraries

from tkinter import \*

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows.

from PIL import ImageTk,Image

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

from tkinter import ttk

The tkinter.ttk module contains all the new ttk widgets. It’s a good practice to always use themed widgets whenever they’re available.

* Separating the widget’s behavior and appearance – the ttk widgets attempt to separate the code that implements the widgets’ behaviors from their appearance through the [styling system](https://www.pythontutorial.net/tkinter/ttk-style/).
* Native look & feel – the ttk widgets have the native look and feel of the platform on which the program runs.
* Simplify the state-specific widget behaviors – the ttk widgets simplify and generalize the [state-specific widget behavior](https://www.pythontutorial.net/tkinter/ttk-style-map/).

from tkinter import filedialog

By using these you don’t have to design standard dialogs your self. Example dialogs include an open file dialog, a save file dialog and many others. Besides file dialogs there are other standard dialogs, but in this article we will focus on file dialogs.

File dialogs help you open, save files or directories. This is the type of dialog you get when you click file,open. This dialog comes out of the module, there’s no need to write all the code manually.

from tkinter import messagebox

The messagebox module is used to display the message boxes in the python applications. There are the various functions which are used to display the relevant messages depending upon the application requirements.

import os

The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system dependent functionality. The \*os\* and \*os.path\* modules include many functions to interact with the file system.

import socket

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.  
They are the real backbones behind web browsing. In simpler terms there is a server and a client.

import encoding as ec

import decoding as dec

import findimage

**Step 2:** Write a function to encode secret message into the image by altering the LSB.

def binary\_to\_decimal(k):

temp = []

for n in k:

val = int(n)

fin,inc = 0,0

while val != 0:

di = val % 10

fin = fin + di \* 2 \*\* inc

val = val//10

inc += 1

temp.append(fin)

return temp

def encode\_into\_pixels(i, word, pixels):

inival, ind = i\*3, 0

for k in range(inival, inival + 3):

for j in range(0, 3):

if j == 2 and k == inival + 2:

continue

pixels[k][j] = pixels[k][j][0:7] + word[ind]

ind += 1

def encode(statement, image\_path):

statement = "<RNA? "+statement+" ?RNA>"

flag, new\_file\_path = False, ''

imp = image\_path.rfind('.')

extension = image\_path[imp + 1:]

if extension.lower() != 'jpg' and extension.lower() != 'png':

messagebox.showerror("ERROR","Invalid file type")

#exit()

elif not os.path.exists(image\_path):

messagebox.showerror("ERROR","File does not exists")

#exit()

else:

new\_file\_path = image\_path[0:imp]+'1.'+'png'

image = Image.open(image\_path,'r')

pix\_val = list(image.getdata())

pix\_val = [list(ele) for ele in pix\_val]

width, height = image.size

values = list(statement)

characters =list(map(lambda x: format(ord(x),'08b'),values))

try:

if (width \* height) // 9 < len(characters):

messagebox.showerror("ERROR","Text size increases Image file size")

exit()

except Exception as e:

messagebox.showerror("ERROR","\nIssue: ",e,"Image encoding not successful!!")

exit()

else:

newlist = []

for i in range(0, len(pix\_val)):

pix\_val[i] = list(map(lambda x: format(x,'08b'),pix\_val[i]))

newlist.append(pix\_val[i])

for i in range(0, len(characters)):

encode\_into\_pixels(i, characters[i], newlist)

newlist = [tuple(binary\_to\_decimal(e)) for e in newlist]

new\_image = Image.new(image.mode, image.size)

new\_image.putdata(newlist)

new\_image.save(new\_file\_path)

response=messagebox.showinfo("Info","Image encoded successfully, New image path: "+new\_file\_path)

**Step 3:**Create a function for finding the image.

def saveImage(image,dir):

with open(dir+ '/' + str(image['id'])+".jpg", "wb") as f:

r1 = requests.get(image["largeImageURL"])

f.write(r1.content)

def storeImages(imagelist, folder):

mode = 0o666

for i in imagelist:

val = i['largeImageURL'].rfind('.')

extension = i['largeImageURL'][val+1:]

if extension.lower() == 'jpg' or extension.lower() == 'png':

saveImage(i,folder)

else:

pass

def getImages(words,num,folder=os.getcwd()):

a1 = "https://pixabay.com/api/?key=16344627-110e29474f27c28a4a9923a6a&q="

a2 = '+'.join(words.split(' '))

a3 = "&safesearch=true&order=popular&per\_page="+str(num)

res = requests.get(a1+a2+a3)

data = res.json()

storeImages(data['hits'],folder)

showinfo("Success","Images stored in "+str(folder))

**Step 4:**Define a function to decode the hidden message from the stego image.

def BinaryToDecimal(binary):

decimal = 0

for digit in binary:

decimal = decimal\*2 + int(digit)

return decimal

def get\_ascii(i,newlist):

inival=i\*3

asc=''

for k in range(inival,inival+3):

for j in range(0,3):

if j==2 and k==inival+2:

continue

asc=asc+newlist[k][j][-1]

return asc

def decode(img\_path):

image = Image.open(img\_path,'r')

pix\_val = list(image.getdata())

pix\_val = [list(ele) for ele in pix\_val]

newlist = []

for i in range(0, len(pix\_val)):

pix\_val[i] = list(map(lambda x: format(x,'08b'),pix\_val[i]))

newlist.append(pix\_val[i])

strn=''

i=0

st\_end=' ?RNA>'

while True:

if strn[-6:]==st\_end:

break

if len(strn)==6:

if strn!='<RNA? ':

return [False]

ascstr=get\_ascii(i,newlist)

ascstr=BinaryToDecimal(ascstr)

ch=chr(ascstr)

strn=strn+ch

i+=1

strn=strn[6:]

strn=strn[0:-6]

l=[]

l.insert(0,'True')

l.insert(1,strn)

return l

**Step 5:**Main Function ()

flag=False

statement=''

path=0

store\_dir=''

img\_p=''

var=''

var1=''

def select\_dir():

global store\_dir

store\_dir=filedialog.askdirectory()

def fetchImages(words,num\_of\_images):

global store\_dir

try:

socket.create\_connection(("www.google.com",80))

except socket.gaierror as e:

messagebox.showerror("ERROR","Please check your internet connection")

except Exception as e:

messagebox.showerror("ERROR",e)

else:

num = 20

if words=='':

messagebox.showerror("ERROR","Please enter words to be searched")

return

if num\_of\_images == '':

num\_of\_images=20

elif num\_of\_images.isalpha():

messagebox.showerror("ERROR","Digits Only Please")

return

elif int(num\_of\_images) < 3 or int(num\_of\_images) > 200:

messagebox.showerror("ERROR","The number of images suould be between 3 - 200")

return

messagebox.showinfo("MESSAGE","Please wait! Pop up will appear once it is finished")

num = int(num\_of\_images)

if store\_dir == '':

findimage.getImages(words,num\_of\_images)

else:

findimage.getImages(words,num\_of\_images,store\_dir)

def test(cw):

def goback():

cw.deiconify()

newTop.withdraw()

cw.withdraw()

newTop=Toplevel(bg="#02001c")

newTop.geometry("450x250+500+300")

container = Frame(newTop, bg='#02001c')

container.grid(row=0, column=0,padx=30,pady=(30,5))

lab=Label(container,text="Enter the keywords separated by space",bg="#02001c",fg="gold")

getword=Entry(container,width=60,relief="sunken")

lab2=Label(container,text="Enter the nunber of images you want",bg="#02001c",fg="gold")

getnum=Entry(container,width=25,relief="sunken")

btnFolder=Button(container,text="Select folder to save images",bg="deepskyblue",fg="#02001c",command=select\_dir,width=20)

btn=Button(container,text="Search and save",bg="deepskyblue",fg="#02001c",command=lambda: fetchImages(getword.get(),getnum.get()),width=20)

btnback=Button(newTop,text="Continue",bg="salmon",fg="#02001c",command=goback)

lab.grid(pady=10,padx=10,row=0,column=0,columnspan=2)

getword.grid(pady=10,padx=10,row=1,column=0,columnspan=2)

lab2.grid(pady=5,row=2,column=0,padx=(10,0))

getnum.grid(pady=5,padx=(0,10),row=2,column=1)

btnFolder.grid(pady=20,padx=(10,2),row=3,column=0)

btn.grid(pady=20,padx=(2,10),row=3,column=1)

btnback.grid(pady=(0,10),padx=1,sticky=E,row=1,column=0)

getword.delete(0,END)

getword.focus()

newTop.protocol("WM\_DELETE\_WINDOW",exit)

def click():

global top3, statement,img\_P,path, top1\_50

if path==0:

statement=e.get()

if statement=='':

messagebox.showerror("ERROR","Text cannot be empty")

return

if path==1 and var1=='':

messagebox.showerror("ERROR","Select an file")

return

def goback():

top1\_50.deiconify()

top3.withdraw()

top1\_50.withdraw()

top3=Toplevel(bg="#02001c")

top3.geometry("380x245+500+300")

container = Frame(top3, bg='#02001c')

container.grid(row=0, column=0,padx=(30,0),pady=30)

l3=Label(container,text="Select the image in which you wish to encode your data",bg="#02001c",fg="gold")

b3=Button(container,text="Choose from files",bg="deepskyblue",fg="#02001c",borderwidth=4,relief=RAISED,command=cdi,width=15)

b4=Button(container,text="Advanced",bg="deepskyblue",fg="#02001c",borderwidth=4,relief=RAISED,command=lambda: test(top3),width=15)

b5=Button(container,text="Encode",bg="yellowgreen",fg="#02001c",borderwidth=4,relief=RAISED,command=enc,width=15)

b6=Button(top3,text="Back",bg="salmon",fg="#02001c",borderwidth=4,relief=RAISED,command=goback,width=6)

l3.grid(pady=10, padx=20,row=0,column=0,columnspan=2)

b3.grid(pady=10, padx=(20,5),row=1,column=0)

b4.grid(pady=10, padx=(5,20),row=1,column=1)

b5.grid(pady=(20,0), padx=20,row=2,column=0,columnspan=2)

b6.grid(pady=5,padx=5,row=1,column=0,sticky=E)

top3.protocol("WM\_DELETE\_WINDOW",exit)

def read(p):

global statement

try:

file\_path = p

imp = file\_path.rfind('.')

extension = file\_path[imp + 1:]

if extension.lower() != 'txt':

messagebox.showerror("ERROR","Invalid file type")

exit()

if not os.path.exists(file\_path):

messagebox.showerror("ERROR","File does not exists")

exit()

else:

with open(file\_path,'r') as f:

statement = f.read()

except Exception as e:

messagebox.showerror("ERROR","\nIssue: ",e)

exit()

def cdf():

top1\_50.filename=filedialog.askopenfilename(initialdir=os.getcwd()+"/sample documents",title="Select a file",filetypes=(("TXT","\*.txt"),("All files","\*.\*")))

global var1

var1=top1\_50.filename

read(var1)

def enc():

global statement,img\_p,var1,path

if img\_p=='':

messagebox.showerror("ERROR","Select an image")

return

ec.encode(statement,img\_p)

r=messagebox.askyesno("Quit","Do you wish to continue?")

if r==0:

root.quit()

else:

top3.withdraw()

guid()

def cdi():

global statement,img\_p,top3

top3.filename=filedialog.askopenfilename(initialdir=os.getcwd()+"/sample images",title="Select a file",filetypes=(("JPG","\*.jpg"),("All files","\*.\*")))

img\_p=top3.filename

def line():

def goback():

top1.deiconify()

top1\_50.withdraw()

global e,path, l, statement, top1, top1\_50

top1.withdraw()

path=0

top1\_50=Toplevel(bg="#02001c")

top1\_50.geometry("480x150+500+300")

container = Frame(top1\_50, bg='#02001c')

container.grid(row=0, column=0,padx=20,pady=(30,0))

l=Label(container)

l=Label(container,text="Enter the encoding text:",bg="#02001c",fg="gold")

l.grid(row=0,column=0)

l.config(font=("Arial", 10))

e=Entry(container,width=45,relief="sunken")

e.grid(row=0,column=1,columnspan=3,pady=3)

b=Button(container,text="Select image",bg="deepskyblue",fg="#02001c",command=click,borderwidth=4,relief=RAISED,width=15)

b.grid(row=1,column=1,padx=10,pady=(20,5))

b1=Button(top1\_50,text="Back",bg="salmon",fg="#02001c",borderwidth=4,relief=RAISED,command=goback,width=6)

b1.grid(row=1,column=0,pady=(0,10),sticky=E)

top1\_50.protocol("WM\_DELETE\_WINDOW",exit)

def file\_c():

def goback():

top1.deiconify()

top1\_50.withdraw()

top1.withdraw()

global top1\_50,l3,b3,b4,path

top1\_50=Toplevel(bg="#02001c")

top1\_50.geometry("490x160+500+300")

path=1

container = Frame(top1\_50, bg='#02001c')

container.grid(row=0, column=0,padx=30,pady=(30,0))

l3=Label(container,text="Select the file conatining data you want to encode",bg="#02001c",fg="gold")

b3=Button(container,text="Choose the files",bg="deepskyblue",fg="#02001c",borderwidth=4,relief=RAISED,command=cdf,width=15)

l3.grid(pady=10, padx=10,row=0,column=0)

b3.grid(pady=10, padx=10,row=0,column=1)

b=Button(container,text="Select image",bg="deepskyblue",fg="#02001c",borderwidth=4,relief=RAISED,command=click,width=15)

b1=Button(top1\_50,text="Back",bg="salmon",fg="#02001c",borderwidth=4,relief=RAISED,command=goback,width=6)

b.grid(pady=(10,0), padx=20,row=1,column=0)

b1.grid(pady=5,padx=10,row=1,column=0,sticky=E)

top1\_50.protocol("WM\_DELETE\_WINDOW",exit)

def encoding():

def goback():

top2.deiconify()

top1.withdraw()

global e, l,statement,top1

top2.withdraw()

top1=Toplevel(bg="#02001c")

top1.geometry("280x150+500+300")

container = Frame(top1, bg='#02001c')

container.grid(row=0, column=0,padx=30,pady=15)

b1=Button(container,text="Encode single line message",borderwidth=4,relief=RAISED,bg="yellowgreen",fg="black",command=line, width=25)

b2=Button(container,text="Encode message from file",borderwidth=4,relief=RAISED,bg="yellowgreen",fg="black",command=file\_c,width=25)

b3=Button(top1,text="Back",bg="salmon",fg="black",borderwidth=4,relief=RAISED,command=goback,width=6)

b1.grid(pady=5, padx=20)

b2.grid(pady=5, padx=20)

b3.grid(pady=5,padx=15,sticky=E)

top1.protocol("WM\_DELETE\_WINDOW",exit)

def deci():

global var

if var=='':

messagebox.showerror("ERROR","Select an Image")

else:

d(var)

def d(v):

try:

image\_path = v

imp = image\_path.rfind('.')

extension = image\_path[imp + 1:]

if extension.lower() != 'png':

messagebox.showerror("ERROR","Invalid file type")

if not os.path.exists(image\_path):

messagebox.showerror("ERROR","File does not exists")

else:

l1=dec.decode(image\_path)

if l1[0]=='True':

new\_file\_path = image\_path[0:imp]+'\_dec.txt'

f=open(new\_file\_path,'w')

f.write(l1[1])

f.close()

response=messagebox.showinfo("Info","Image decoded successfully, New text file path: "+new\_file\_path)

if response =="ok":

r=messagebox.askyesno("Quit","Do you wish to continue?")

if r==0:

top3.withdraw()

root.quit()

else:

top3.withdraw()

guid()

if l1[0]==False:

messagebox.showerror("ERROR","Image has no encoded data")

except Exception as e:

messagebox.showerror("ERROR","\nIssue: "+str(e))

def cdfd():

top3.filename=filedialog.askopenfilename(initialdir="/sample images",title="Select a file",filetypes=(("PNG","\*.png"),("All files","\*.\*")))

global var

var=top3.filename

def decoding():

def goback():

top2.deiconify()

top3.withdraw()

top2.withdraw()

global top3,var

top3=Toplevel(bg="#02001c")

top3.geometry("440x170+500+300")

container = Frame(top3, bg='#02001c')

container.grid(row=0, column=0,padx=30,pady=(30,0))

l3=Label(container,text="Select the image you want to decode",bg="#02001c",fg="gold")

b3=Button(container,text="Choose the files",bg="deepskyblue",fg="#02001c",borderwidth=4,relief=RAISED,command=cdfd,width=12)

b4=Button(container,text="Decode",bg="yellowgreen",fg="#02001c",borderwidth=4,relief=RAISED,command=deci,width=12)

b5=Button(top3,text="Back",bg="salmon",fg="#02001c",borderwidth=4,relief=RAISED,command=goback,width=6)

l3.grid(pady=10, padx=20,row=0,column=0)

b3.grid(pady=10, padx=20,row=0,column=1)

b4.grid(pady=10, padx=20,row=1,columnspan=2,column=0)

b5.grid(pady=5, padx=5,row=1,column=0,sticky=E)

top3.protocol("WM\_DELETE\_WINDOW",exit)

def guid():

def goback():

root.deiconify()

top2.withdraw()

root.withdraw()

global top2

top2=Toplevel(bg="#02001c")

top2.geometry("+300+200")

B1=Button(top2)

B2=Button(top2)

L1=Label(top2,text="GUIDELINES FOR THE PROJECT",bg="#02001c",fg="deepskyblue",font=(("Arial",20,"bold italic"))).grid(row=0,column=0,columnspan=2)

L2=Label(top2,text="1.Select encoding button for encoding the data or decoding button for decoding.",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=1,column=0,columnspan=2)

L3=Label(top2,text="2.If you select encoding, select the button according to your choice.",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=2,column=0,columnspan=2)

L4=Label(top2,text="3.You can enter the text directly or select a text document containing data.",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=3,column=0,columnspan=2)

L5=Label(top2,text="4.Select image from the images folder to encode data into the image.",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=4,column=0,columnspan=2)

L6=Label(top2,text="5.If you select decoding,select the image to be decoded from the image

folder",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=5,column=0,columnspan=2)

L7=Label(top2,text="6.The decoded information is obtained in a text file present in the same folder.",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=6,column=0,columnspan=2)

L8=Label(top2,text="7.Use the advanced button to search images based on keywords",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=7,column=0,columnspan=2)

L9=Label(top2,text="8.Please wait for pop up after choosing image file since big image file might take time to process.",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=8,column=0,columnspan=2)

L10=Label(top2,text="9.Number of images should be between 3 and 200 and if no value is entered the default value is 20.",bg="#02001c",fg="whitesmoke",font=(("Arial",13))).grid(row=9,column=0,columnspan=2)

B1=Button(top2,text="Encoding",bg="gold",fg="#02001c",borderwidth=4,relief=RAISED,command=encoding).grid(row=10,column=0,pady=10, sticky=E)

B2=Button(top2,text="Decoding",bg="gold",fg="#02001c",borderwidth=4,relief=RAISED,command=decoding).grid(row=10,column=1,padx=5,pady=10,sticky=W)

B3=Button(top2,text="Back",bg="salmon",fg="#02001c",borderwidth=4,relief=RAISED,command=goback).grid(row=10,column=2,padx=5,pady=10,sticky=N)

top2.protocol("WM\_DELETE\_WINDOW",exit)

def home():

global img

img=ImageTk.PhotoImage(Image.open("sample images/saved1.png"))

img\_label=Label(root,image=img,width=500,height=324)

img\_label.grid(row=0,column=0)

s1=Label(root,text="PROJECT",bg="#02001c",fg="gold",font=(("Arial",16,"bold

italic")))

s2=Label(root,text="IMAGE

STEGANOGRAPHY",bg="#02001c",fg="deepskyblue",font=(("Arial",16,"bold italic")))

s3=Label(root,text="Prepared

By",bg="#02001c",fg="salmon",font=(("Arial",14,"underline")))

s4=Label(root,text="RITIKA

KHANDELWAL",bg="#02001c",fg="whitesmoke",font=(("Arial",14)))

B=Button(root,text="Guidelines for the project",bg="greenyellow",fg="#02001c",borderwidth=4,relief=RAISED,command=guid)

s1.grid(row=1,column=0,pady=10)

s2.grid(row=2,column=0,pady=10)

s3.grid(row=3,column=0)

s4.grid(row=4,column=0)

B.grid(row=7,column=0,pady=10)

root=Tk()

root.title("STEGANOGRAPHY")

root.config(bg="#02001c")

root.geometry('505x600+400+30')

home()

root.mainloop()

**Chapter – 7**

**TESTING**

**7.1 INTRODUCTION**

Process of creating a program consists of the following phases:

1. defining a problem;

2. designing a program;

3. building a program;

4. analyzing performances of a program, and

5. final arranging of a product.

According to this classification, software testing is a component of the third phase, and means checking if a program for specified inputs gives correctly and expected results. So the main aim of testing is to analyze the performance and to evaluate the errors that occur when the program is executed with different input sources and running in different operating environments.

Software testing is an important component of software quality assurance, andmany software organizations are spending up to 40% of their resources on testing. For lifecritical software (e.g., flight control) testing can be highly expensive. Because of that, many studies about risk analysis have been made. This term means the probability that a software project will experience undesirable events, such as schedule delays, cost overruns, or outright cancellation.

There are a many definitions of software testing, but one can shortly define that as: A process of executing a program with the goal of finding errors. So, testing means that one inspects behavior of a program on a finite set of test cases (a set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement) for which valued inputs always exist.

In practice, the whole set of test t cases is considered as infinite, therefore theoretically there are too many test cases even for the simplest programs. In this case, testing could require months and months to execute.

So, how to select the most proper set of test cases? In practice, various techniques are used for that, and some of them are correlated with risk analysis, while others with test engineering expertise.

Testing is an activity performed for evaluating software quality and for improving it. Hence, the goal of testing is systematical detection of different classes of errors (error can be defined as a human action that produces an incorrect result) in a minimum amount of time and with a minimum amount of effort.

The types of testing are:

* Unit testing
* Validation testing
* Integration testing
* User acceptance testing
* Output testing
* Black box and white box testing

**7.2 UNIT TESTING**

Unit testing is the approach of taking a small part of testable application and executing it according to the requirements and testing the application behavior. Unit testing is used for detecting the defects that occur during execution .

When an algorithm is executed, the integrity should be maintained by the data structures. Unit testing is made use for testing the functionality of each algorithm during execution.

Unit testing can be used in the bottom up test approach which makes the integration test much easier. Unit testing reduces the ambiguity in the units.

Unit testing uses regression testing, which makes the execution simpler. Using regression testing, the fault can be easily identified and fixed.

In this project, the purposed system of hiding the data using different phases likes encryption, decryption, etc.

So, for getting the correct output all the functions that are used are executed and tested at least once making sure that all the control paths, error handling and control structures are in proper manner.

Unit testing has its applications for extreme programming, testing unit frame works and good support for language level unit testing.

**7.3 VALIDATION TESTING**

Validation is the process of finding whether the product is built correct or not. The software application or product that is designed should fulfill the requirements and reach the expectations set by the user.

Validation is done while developing or at the final stage of development process to determine whether it is satisfies the specified requirements of users.

Using validation test the developer can qualify the design, performance and its operations. Also the accuracy, repeatability, selectivity, Limit of detection and quantification can be specified using Validation testing.

**7.4 OUTPUT TESTING**

After completion of validation testing the next process is output testing. Output testing is the process of testing the output generated by the application for the specified inputs.

This process checks weather the application is producing the required output as per the user's specification or not.

The output testing can be done by considering mainly by updating the test plans, the behavior of application with different type of inputs and with produced outputs, making the best use of the operating capacity and considering the recommendations for fixing the issues.

**7.5 INTEGRATION TESTING**

Integration testing is an extension to unit testing, after unit testing the units are integrated with the logical program.

The integration testing is the process of examining the working behavior of the particular unit after embedding with program. This procedure identifies the problems that occur during the combination of units.

The integration testing can be commonly done in three approaches:

* Top-down approach
* Bottom-up approach
* Umbrella approach

**Top-down approach:**

In the top-down approach the highest level module should be considered first and integrated. This approach makes the high level logic and data flow to test first and reduce the necessity of drivers.

One disadvantage with top-down approach is its poor support and functionality is limited.

**Bottom-up approach:**

Bottom-up approach is opposite to top-down approach. In this approach, the lowest level units are considered and integrated first. Those units are known as utility units. The utility units are tested first so that the usage of stubs is reduced.

The disadvantage in this method is that it needs the respective drivers which make the test complicated, the support is poor and the functionality is limited.

**Umbrella approach:**

The third approach is umbrella approach, which makes use of both the top - bottom and bottom - top approaches.

This method tests the integration of units along with its functional data and control paths. After using the top - bottom and bottom-top approaches, the outputs are integrated in top - bottom manner.

The advantage of this approach is that it provides good support for the release of limited functionality as well as minimizing the needs of drivers and hubs.

The main disadvantage is that it is less systematic than the other two approaches.

**7.6 USER ACCEPTANCE TESTING**

User acceptance testing is the process of obtaining the confirmation from the user that the system meets the set of specified requirements. It is the final stage of project; the user performs various tests during the design of the applications and makes further modifications according to the requirements to achieve the final result. The user acceptance testing gives the confidence to the clients about the performance of system.

**7.7 BLACK BOX TESTING**

Black box testing is the testing approach which tells us about the possible combinations for the end-user action.

Black box testing doesn't need the knowledge about the interior connections or programming code.

In the black box testing, the user tests the application by giving different sources and checks whether the output for the specified input is appropriate or not.

**7.8 WHITE BOX TESTING**

White box testing is also known as "glass box" or "clear box" or "open box" testing. It is opposite to the black box testing. In the white box testing, we can create test cases by checking the code and executing in certain intervals and know the potential errors.

The analysis of the code can be done by giving suitable inputs for the specified applications and using the source code for the application blocks.

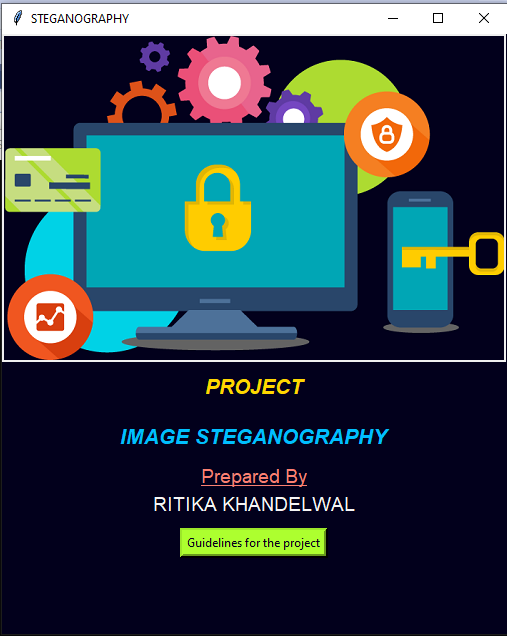
The limitation with the white box testing is that the testing only applies to unit testing, system testing and integration testing.

These are the different testing approaches that can be used for testing the application which is developed using Microsoft Visual studio.

**Chapter - 8**

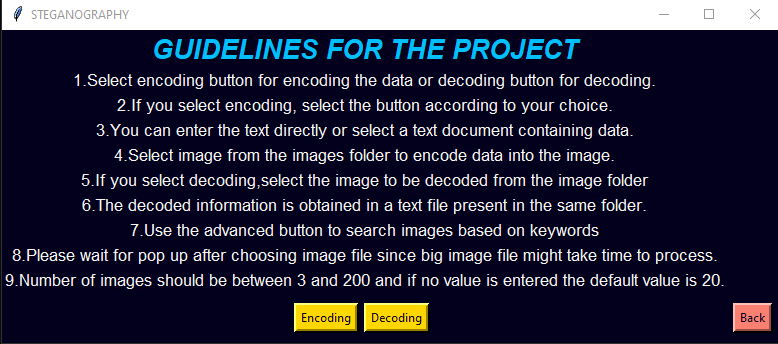
**SNAPSHOTS**

**HOME PAGE**

****

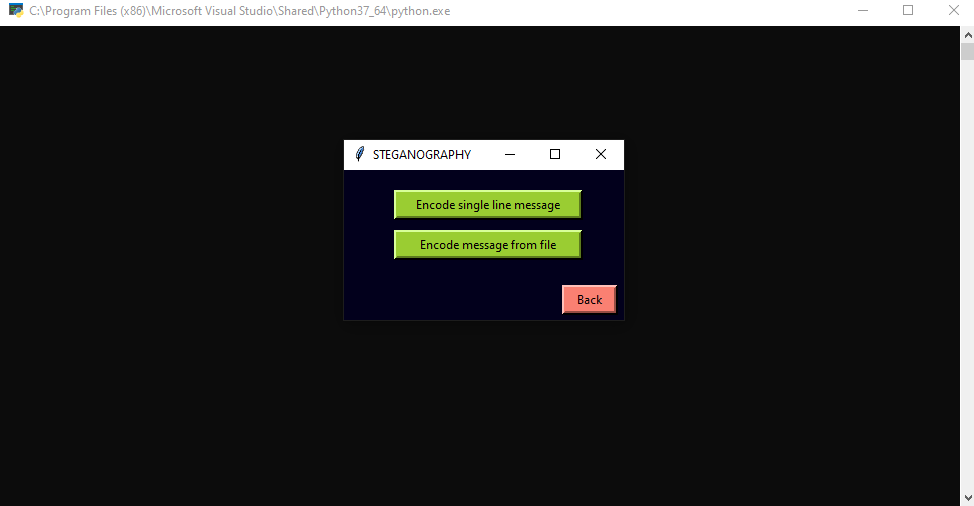
**Figure 8.1- Home Page**

**GUIDELINES FOR THE PROJECT**

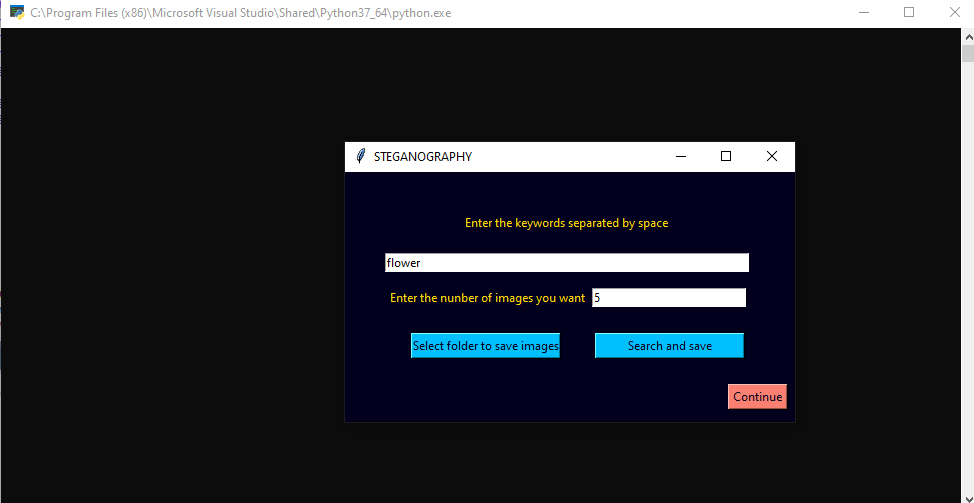
****

**Figure 8.2-Guidelines for the project**

**CHOOSE THE FILE FOR ENCODING**

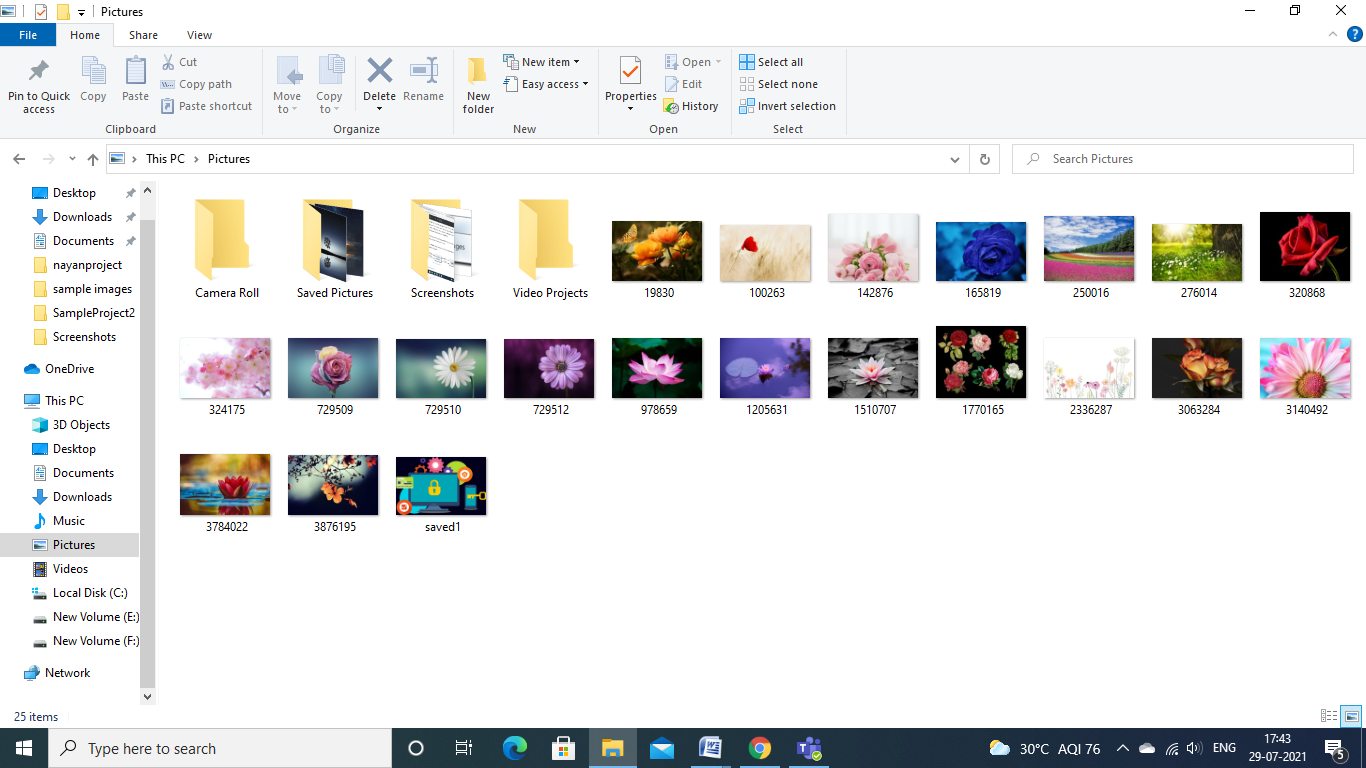
**** **Figure 8*.***3**-Choose the File**

**FIND THE IMAGE**

****

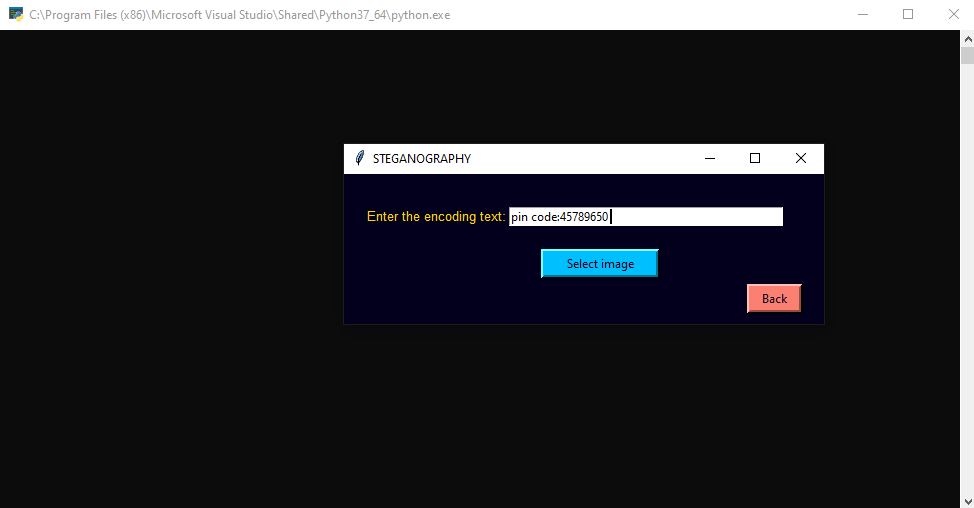
**Figure 8*.***4**-Find the image**

**OUTPUT OF SEARCHED IMAGES**

****

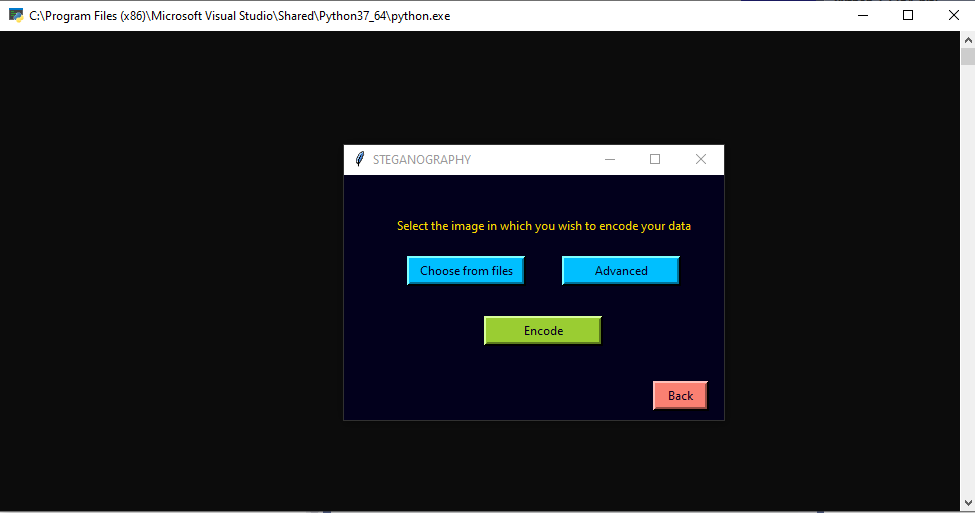
**Figure 8*.***5**-Output of Searched images**

**ENTER THE ENCODING TEXT**

****

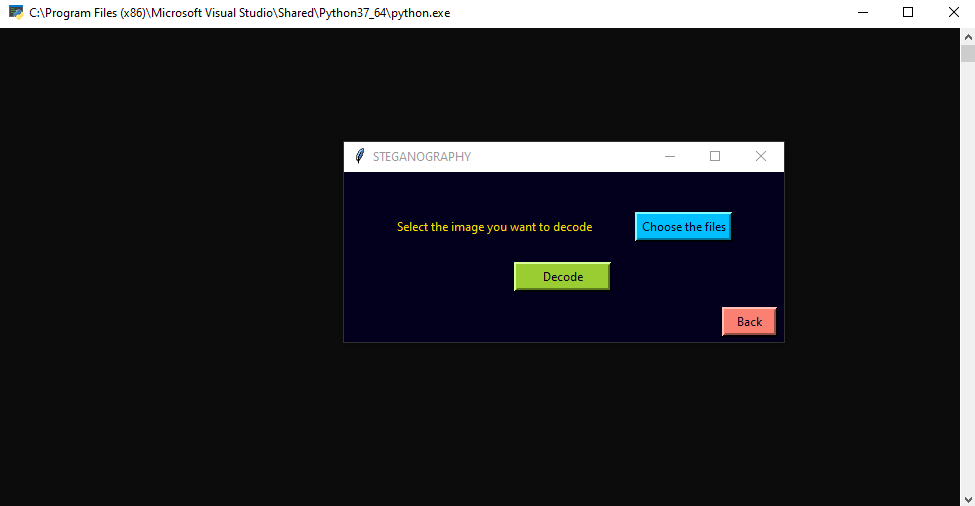
**Figure** 8***.***6**- Enter the Encoding Text**

**SELECT THE IMAGE**

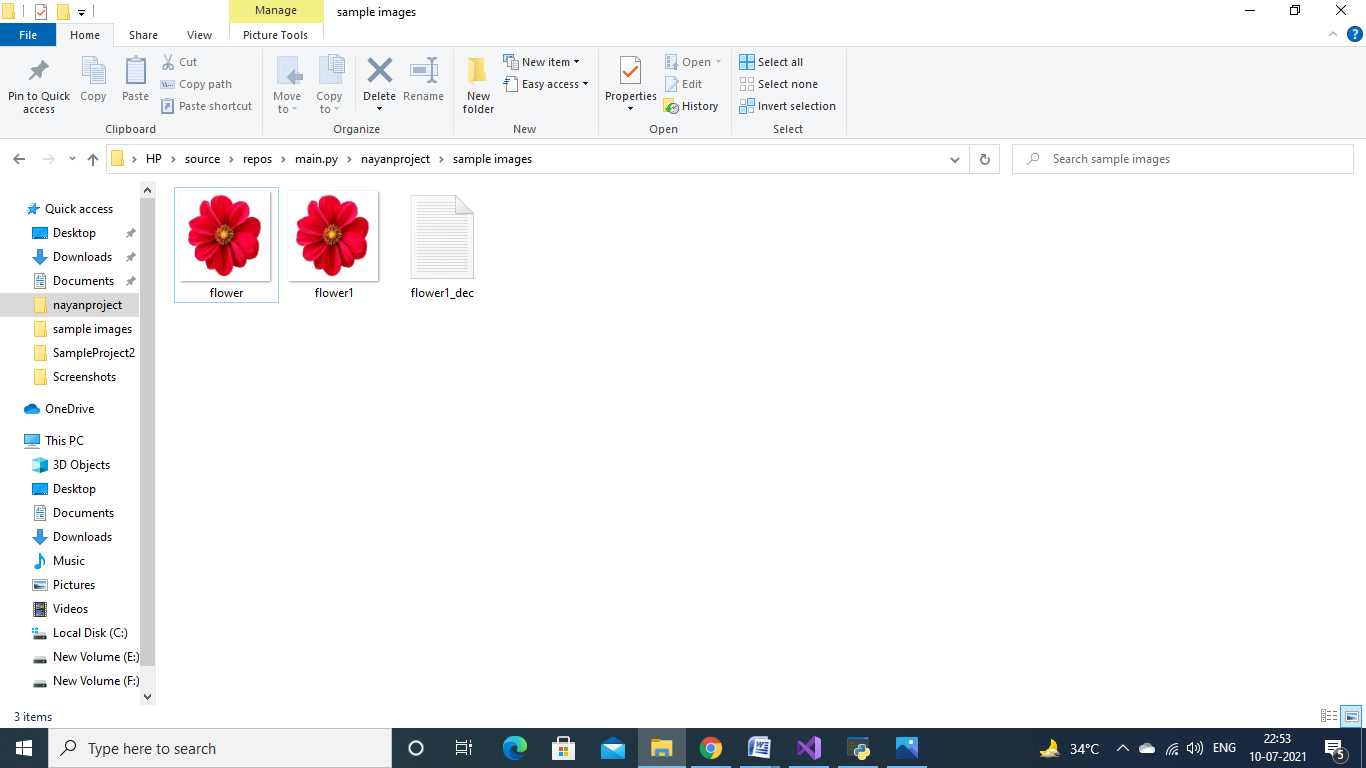
****

**Figure** 8***.***7**- Select the image**

**DECODE THE FILE**

**Figure** 8***.***8**- Decode the file**

**RESULT**

**Figure** 8***.***9**- Result**

**Chapter - 9**

**CONCLUSION & FUTURE RECOMMENDATIONS**

**9.1 CONCLUSIONS**

Although only some of the main image steganographic techniques were discussed in this document, one can see that there exists a large selection of approaches to hiding information in images. All the major image file formats have different methods of hiding messages, with different strong and weak points respectively. Where one technique lacks in payload capacity, the other lacks in robustness. For example, the patchwork approach has a very high level of robustness against most type of attacks, but can hide only a very small amount of information.

Least significant bit (LSB) in both BMP and GIF makes up for this, but both approaches result in suspicious files that increase the probability of detection when in the presence of a warden.

The proposed approach in this project uses a new steganographic approach called image steganography. The application creates a stego image in which the personal data is embedded inside the cover file image.

Used the Least Significant Bit algorithm in this project for developing the application which is faster and reliable and compression ratio is moderate compared to other algorithms.

**9.2 FUTURE RECOMMENDATIONS**

The major limitation of the application is designed for images cover files. It accepts only images as a carrier file. The future work on this project is to improve the compression ratio of the image to the text.

This project can be extended to a level such that it can be used for the different types of multimedia files.

**References**

[1]. D'Souza, Divya Jennifer, and Minu P. Abraham. "A Multilayered Secure Scheme for Transmission of Sensitive Information based on Steganalysis." *Procedia Computer Science* 78 (2016): 375-381.

[2]. Wang, Ran-Zan, Chi-Fang Lin, and Ja-Chen Lin. "Image hiding by optimal LSB substitution and genetic algorithm." *Pattern recognition* 34, no. 3 (2001): 671-683.

[3]. Rustad, Supriadi, Abdul Syukur, and Pulung Nurtantio Andono. "Inverted LSB image steganography using adaptive pattern to improve imperceptibility." *Journal of King Saud University-Computer and Information Sciences* (2021).

[4]. Praghash, K., Ch Vidyadhari, G. NirmalaPriya, and Rajan Cristin. "Secure information hiding using LSB features in an image." *Materials Today: Proceedings* (2021).

[5]. Sahu, Aditya Kumar, and Gandharba Swain. "High fidelity based reversible data hiding using modified LSB matching and pixel difference." *Journal of King Saud University-Computer and Information Sciences* (2019).

[6].Chai, Xiuli, Haiyang Wu, Zhihua Gan, Yushu Zhang, Yiran Chen, and Kent W. Nixon. "An efficient visually meaningful image compression and encryption scheme based on compressive sensing and dynamic LSB embedding." *Optics and Lasers in Engineering* 124 (2020): 105837.

[7].Karthikeyan, N., K. Kousalya, N. Jayapandian, and G. Mahalakshmi. "Assessment of composite materials on encrypted secret message in image steganography using RSA algorithm." *Materials Today: Proceedings* (2021).

[8]. Reddy, M. Indra Sena, and AP Siva Kumar. "Secured data transmission using wavelet based steganography and cryptography by using AES algorithm." *Procedia Computer Science* 85 (2016): 62-69.

[9]. Berger, Arthur Asa. *Cultural criticism: A primer of key concepts*. Vol. 4. Sage Publications, 1995.

[10]. Jaskolka, Karina, Jürgen Seiler, Frank Beyer, and André Kaup. "A Python-based laboratory course for image and video signal processing on embedded systems." *Heliyon* 5, no. 10 (2019): e02560.

[11]. Veena, S. T., and S. Arivazhagan. "Quantitative steganalysis of spatial LSB based stego images using reduced instances and features." *Pattern Recognition Letters* 105 (2018): 39-49.

[12]. Lerch-Hostalot, Daniel, and David Megias. "LSB matching steganalysis based on patterns of pixel differences and random embedding." *Computers & security* 32 (2013): 192-206.

[13]. Zhang, Tao, and Xijian Ping. "A new approach to reliable detection of LSB steganography in natural images." *Signal processing* 83, no. 10 (2003): 2085-2093.

[14]. Bhardwaj, Rupali, and Vaishali Sharma. "Image steganography based on complemented message and inverted bit LSB substitution." *Procedia Computer Science* 93 (2016): 832-838

[15]. Xu, Wen-Long, Chin-Chen Chang, Tung-Shou Chen, and Liang-Min Wang. "An improved least-significant-bit substitution method using the modulo three strategy." *Displays* 42 (2016): 36-42.