Dharmsinh Desai University, Nadiad
Faculty of Technology
Department of Computer Engineering

B. Tech. CE Semester – VI

Subject: (CE – 619) Service Oriented Computing

Project Title:

Image Steganography

Submitted By:

Harshit Tarsariya	CE136	18CEUBG080
Janak Vaghasiya	CE142	18CEUOS058
Jayesh Zinzuvadia	CE149	18CEUBG064

Guided by:

Prof. Prashant M. Jadhav

and

Prof. Apurva A. Mehta

Dharmsinh Desai University, Nadiad
Faculty of Technology
Department of Computer Engineering

CERTIFICATE

This is to certify that the project work carried out in

the subject of ***Service Oriented Computing*** *is the bonafide work of*

1)	***Harshit Tarsariya***	***CE136***	***18CEUBG080***
2)	***Janak Vaghasiya***	***CE142***	***18CEUOS058***
3)	***Jayesh Zinzuvadia***	***CE149***	***18CEUBG064***

of Bachelor of Technology, Semester ***6*** *in the branch of*

Computer Engineering *during the academic year* ***2020-2021***

Guide

Prof. Apurva A. Mehta
Assistant Professor,
Department of
Computer Engineering,
Dharmsinh Desai
University,
Nadiad

Guide

Prof. Prashant M. Jadhav
Associate Professor,
Department of
Computer Engineering,
Dharmsinh Desai
University,
Nadiad

HOD

Dr. C. K. Bhensdadia
Head of Department,
Department of
Computer Engineering,
Dharmsinh Desai
University,
Nadiad

Table of Contents

1. Abstract

“Everything in this world has a hidden meaning”

-Nikos Kazantzakis

Image Steganography project is about hiding the data within an image file. The main purpose of this project is to covert communication i.e. to hide the existence of a message inside an image and to further secure it from a third party using the data encryption standards.

For example – let’s say that Alice wants to send a secret message to Bob. For that, Alice will first encrypt the message using the secret key, known to only Alice and Bob. Then, Alice will select a cover image and will use the application to covert the message in it. In the end, Alice will send the image to the Bob.

Now, after receiving the image, Bob will use the application to extract the message from the image and then decrypt it using the same secret key used by Alice during the encryption.

Thus, in this way a message can be shared secretly and securely between two people. Others will know nothing about the hidden message. Even if they knew it, then also they will not be able to interpret the message correctly because the message is encrypted.

2. Introduction

2.1 Brief Introduction:-

Steganography is an art of hiding messages over some medium like Text, Image, Video, etc. The medium is also known as “cover”. Based on the “cover”, we have different categories like Text Steganography, Image Steganography and Video Steganography. But here main focus of the study is Image Steganography.

A digital image is composed of a finite set of digital values, called pixels. One can think of an image as a matrix of pixels which contains a fixed number of rows and columns. As digital image has three channels, so bits can be changed as per the data we want to hide. But changing any bits will impact the image in larger way and both original and encoded image can be differentiated. So the solution is to change only the last bit, LSB of each channel which contributes less in the image visualization. This is known as the **LSB** technique.

Also, if someone knows that the current image is encoded then they can reverse the process and can know the data which is hidden. So to avoid such situation we have used data encryption standards like **AES** and **DES** which are much secure. So the image which is encoded has the data hidden in it in the encrypted form and can only be decrypted if the attacker knows the private key.

2.2 Tools, Technology and Platform used:-

1) Programming Language: C#

2) Technology used: ASP.NET Framework

3) IDE: Visual Studio 2019

4) Packages/Namespace Used:

System	System.Web.Http	System.Security.Cryptography
System.Drawing	System.IO	System.IO.Compression
System.Drawing.Imaging	System.Text	Owin

3. Software Requirements Specifications

3.1 Product Scope

The system is designed to perform image steganography operations to hide the data secretly and securely inside an image. Scope of the system is global and open for all users. System provides various functionalities to the users like encoding (embedding the message inside the image), decoding (extracting the message from the image) and using either AES or DES methods for encryption and decryption of the message.

3.2 Types of User

Here, there is only one end-user who is going to use this application.

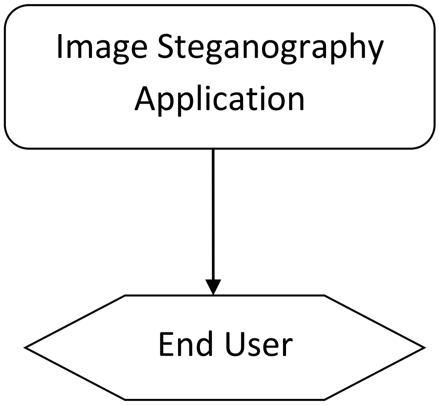


Fig. 3.1: Product Perspective Diagram

3.3 System Functional Requirements

NOTE: Here, the functional requirements are listed as module wise

R1. Encode Message

Description: This requirement is for embedding the data inside the image using the LSB (Least Significant Bit) technique. Return the image having hidden data.

Input: Message, Cover Image

Output: Stego Image

R2. Decode Message

Description: This requirement is for extracting the hidden data from the stego image using the LSB (Least Significant Bit) technique.

Input: Stego Image

Output: Display the hidden data from the image

R3. AES (Advanced Encryption Standard) method

Description: This requirement is for using the AES encryption and decryption methods for the security of the embedded data inside the image.

R3.1. AES Encryption

Input: Plain text, Key

Output: Cipher text

R3.2. AES Decryption

Input: Cipher text, Key

Output: Plain text

R4. DES (Data Encryption Standard) method

Description: This requirement is for using the DES encryption and decryption methods for the security of the embedded data inside the image.

R4.1. DES Encryption

Input: Plain text, Key

Output: Cipher text

R4.2. DES Decryption

Input: Cipher text, Key

Output: Plain text

3.4 Other Non-functional Requirements

1) Performance

The application should run efficiently.

2) Security

This application is concerned with security aspects so security of data is must.

3) Reliability

The application must ensure that the system is reliable in its image steganography operations.

4) Responsiveness

User Interface must be responsive in nature i.e. depending on the screen size or device size, the UI should also adjust/change its size. UI must be interactive and user friendly in nature.

4. Design

4.1 Use Case Diagram

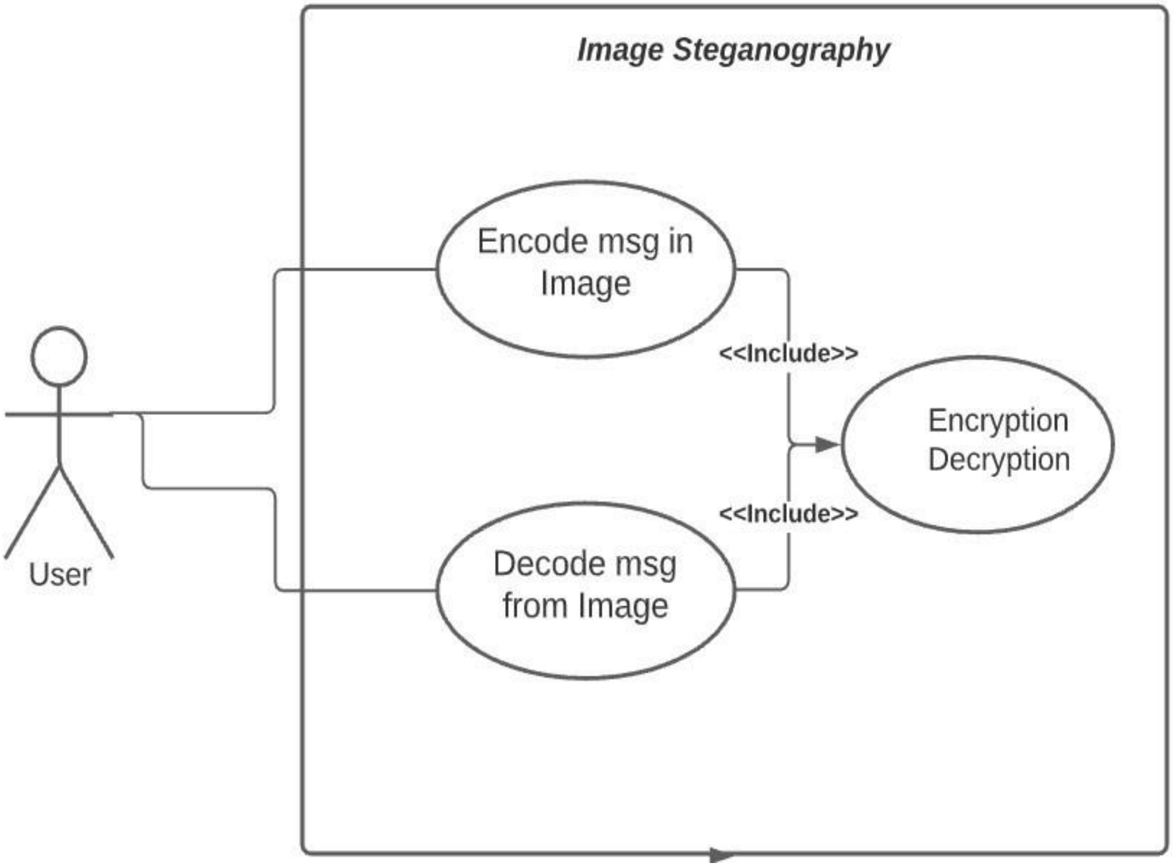


Fig. 4.1: Use-Case Diagram

4.2 Class Diagram

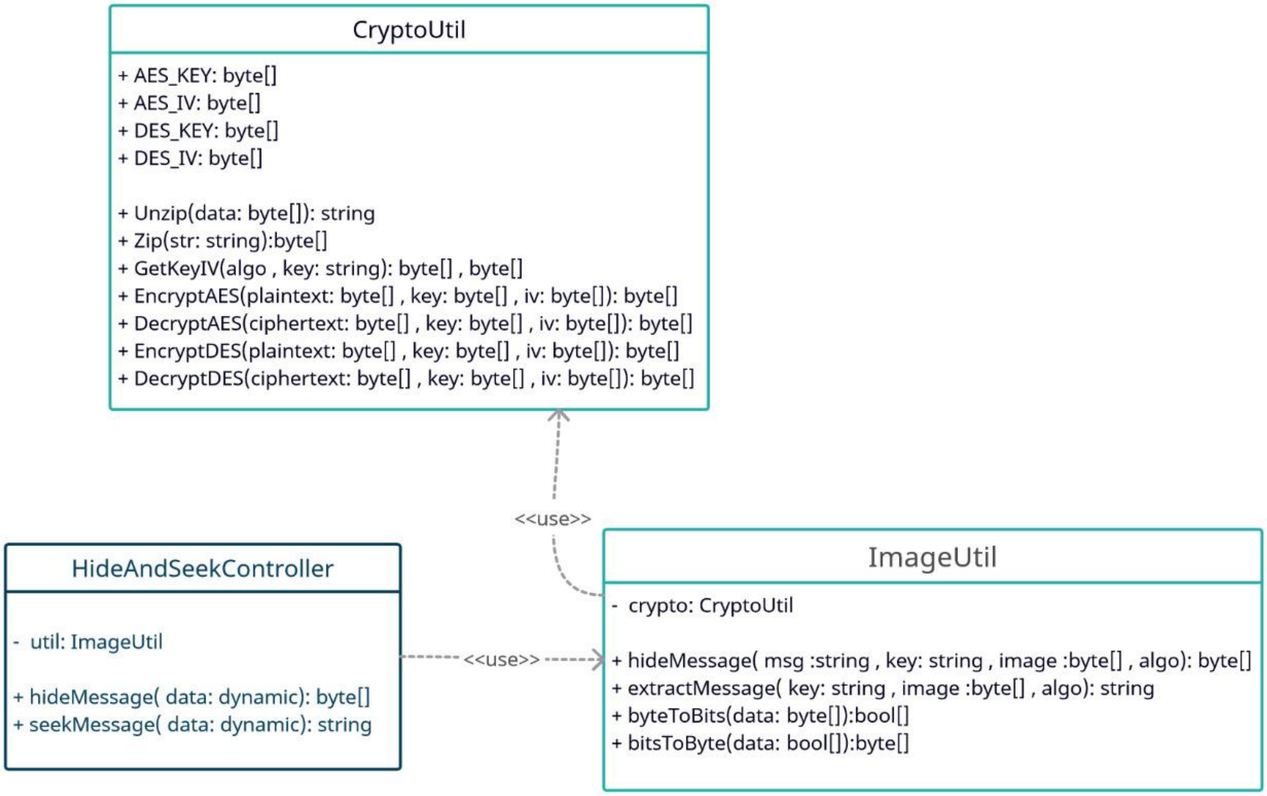

CryptoUtil
+ AES_KEY: byte[]
+ AES_IV: byte[]
+ DES_KEY: byte[]
+ DES_IV: byte[]
+ Unzip(data: byte[]): string
+ Zip(str: string):byte[]
+ GetKeyIV(algo , key: string): byte[] , byte[]
+ EncryptAES(plaintext: byte[] , key: byte[] , iv: byte[]): byte[]
+ DecryptAES(ciphertext: byte[] , key: byte[] , iv: byte[]): byte[]
+ EncryptDES(plaintext: byte[] , key: byte[] , iv: byte[]): byte[]
+ DecryptDES(ciphertext: byte[] , key: byte[] , iv: byte[]): byte[]
<<use>>
HideAndSeekController
- util: ImageUtil
+ hideMessage(data: dynamic): byte[]
+ seekMessage(data: dynamic): string
<<use>>
ImageUtil
- crypto: CryptoUtil
+ hideMessage(msg :string , key: string , image :byte[] , algo): byte[]
+ extractMessage(key: string , image :byte[] , algo): string
+ byteToBits(data: byte[]):bool[]
+ bitsToByte(data: bool[]):byte[]

Fig. 4.2: Class Diagram

4.3 State Diagram

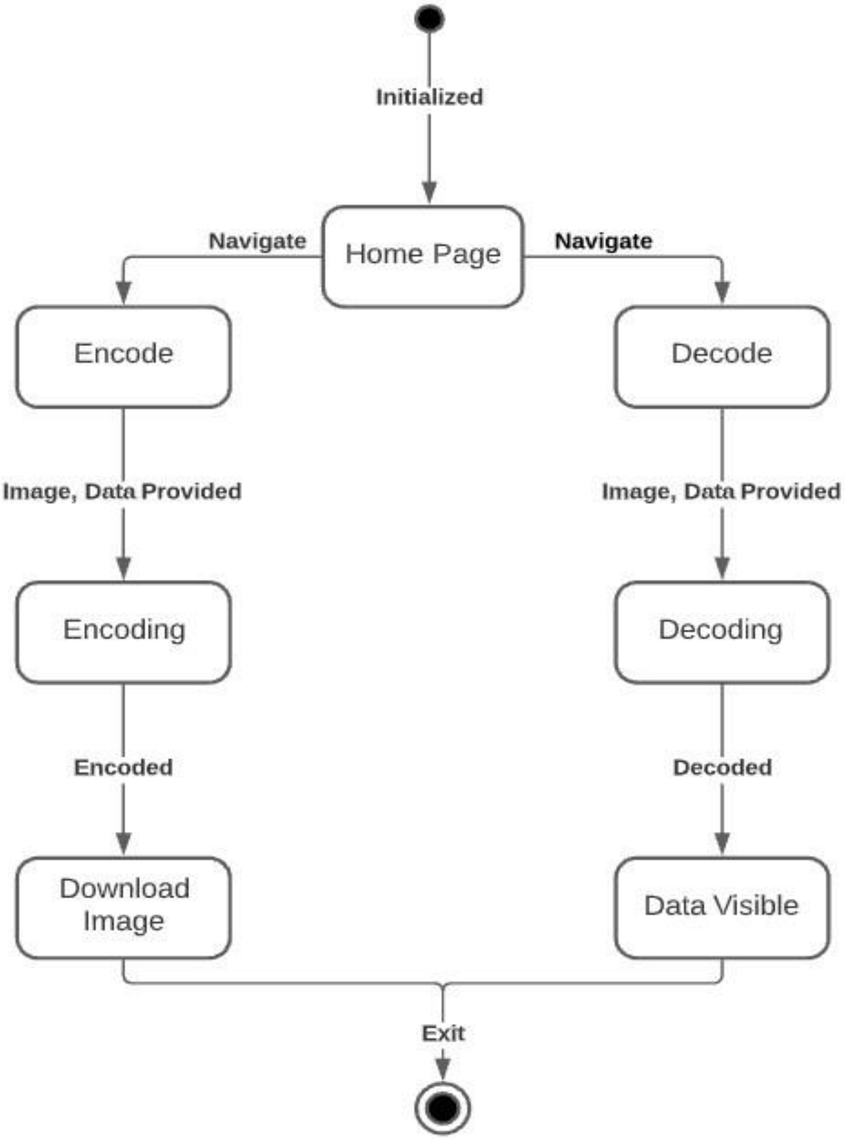


Fig. 4.3: State Diagram

5. Implementation Details

5.1 Web API Project Architecture

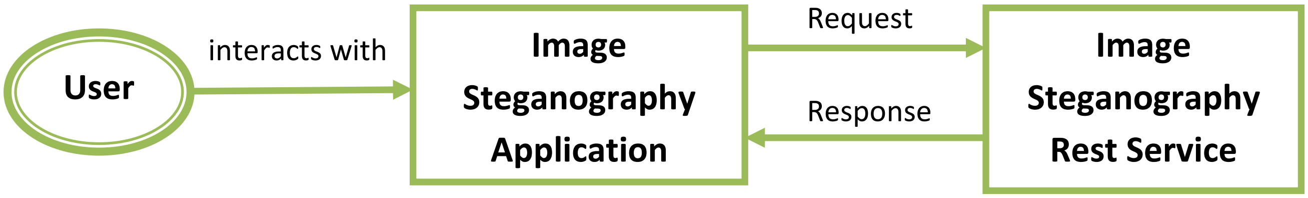
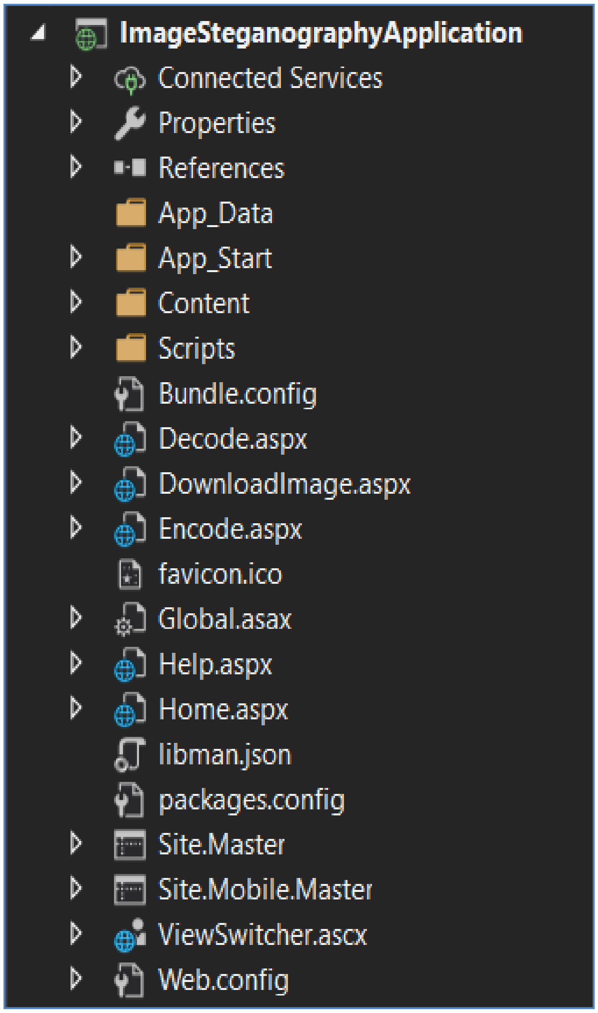

User
interacts with
Image Steganography Application
Request
Response
Image Steganography Rest Service

Fig. 5.1 Project Architecture

5.2 Image Steganography Front-End Application



```
ImageSteganographyApplication
    Connected Services
    Properties
    References
    App_Data
    App_Start
    Content
    Scripts
    Bundle.config
    Decode.aspx
    DownloadImage.aspx
    Encode.aspx
    favicon.ico
    Global.asax
    Help.aspx
    Home.aspx
    libman.json
    packages.config
    Site.Master
    Site.Mobile.Master
    ViewSwitcher.ascx
    Web.config
```

Fig. 5.2 Folder Structure of Image Steganography Application

Purpose of each file/folder

1) Content – Contains CSS and bootstrap files for UI design

2) Scripts – Contains JavaScript and jQuery code for dynamic behaviour

3) Home.aspx – Home page of the application describing Image Steganography in brief

4) Encode.aspx – For hiding the data inside a cover image

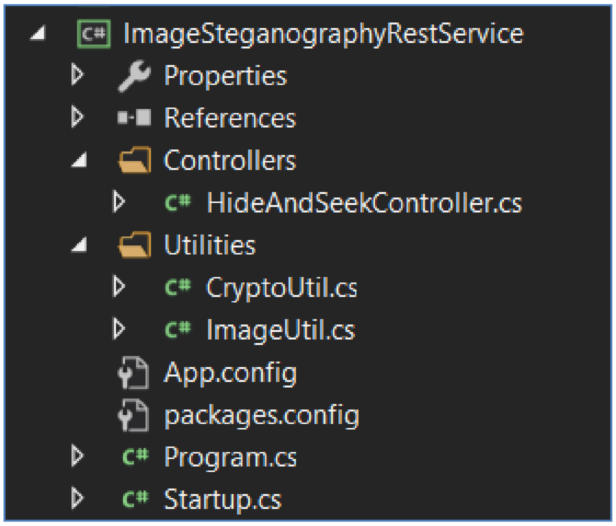
5) DownloadImage.aspx – To download the stego image

6) Decode.aspx – For extracting the data from the stego image

7) Help.aspx – To guide the user about how to use this application and Contact Developers for queries.

5.3 Image Steganography Rest Service

Project’s Folder Structure



```
ImageSteganographyRestService
    Properties
    References
    Controllers
        HideAndSeekController.cs
    Utilities
        CryptoUtil.cs
        ImageUtil.cs
    App.config
    packages.config
    Program.cs
    Startup.cs
```

Fig. 5.3 Folder Structure for Image Steganography Rest Service

Implementation Details:-

1) HideAndSeekController.cs

It is rest controller which has two action method

- hideMessage // hide message using ImageUtil class object

- seekMessage //extract message using ImageUtil class object

2) CryptoUtil.cs

It is a utility class which contains methods for encryption, decryption, zip and unzip.

Currently only AES & DES are supported

We have used system's built-in library to perform this task

Workflow:

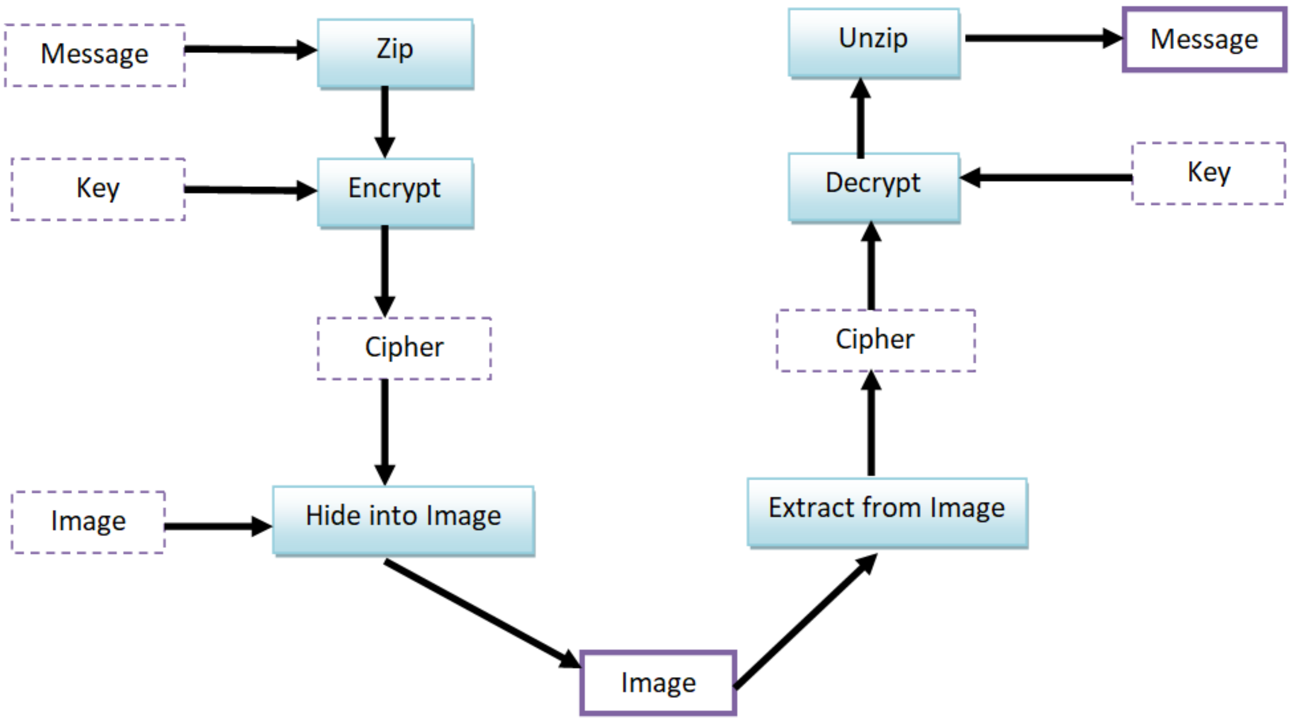


Fig. 5.4 Workflow chart

3) ImageUtil.cs

- hideMessage //here data of client will be embedded into cover image

- seekMessage //extract message from cover image(image and key must be valid)

- For data compression/expansion and encryption/decryption it uses CryptoUtil class object

6. Testing

6.1 Testing Method Used

We have performed unit testing during the development. But for testing purpose, we have used black box testing method.

For black box testing, we have designed the test cases for each sub project and have tested it in our application. Also, we have observed the output and note down the results in the next section.

6.2 Test Cases

6.2.1 Encode Message

Test Case ID	Parameters	Test Data (Cover Image)	Actual Output (Stego Image)
T1	***Message***: “The best feeling in the world is to know that your Parents are Smiling because of You” ***Enc. Type:*** AES ***Key:*** 123456		
T2	**Message:** “Alls Well That Ends Well” **Enc. Type:** DES **Key:** 72910		

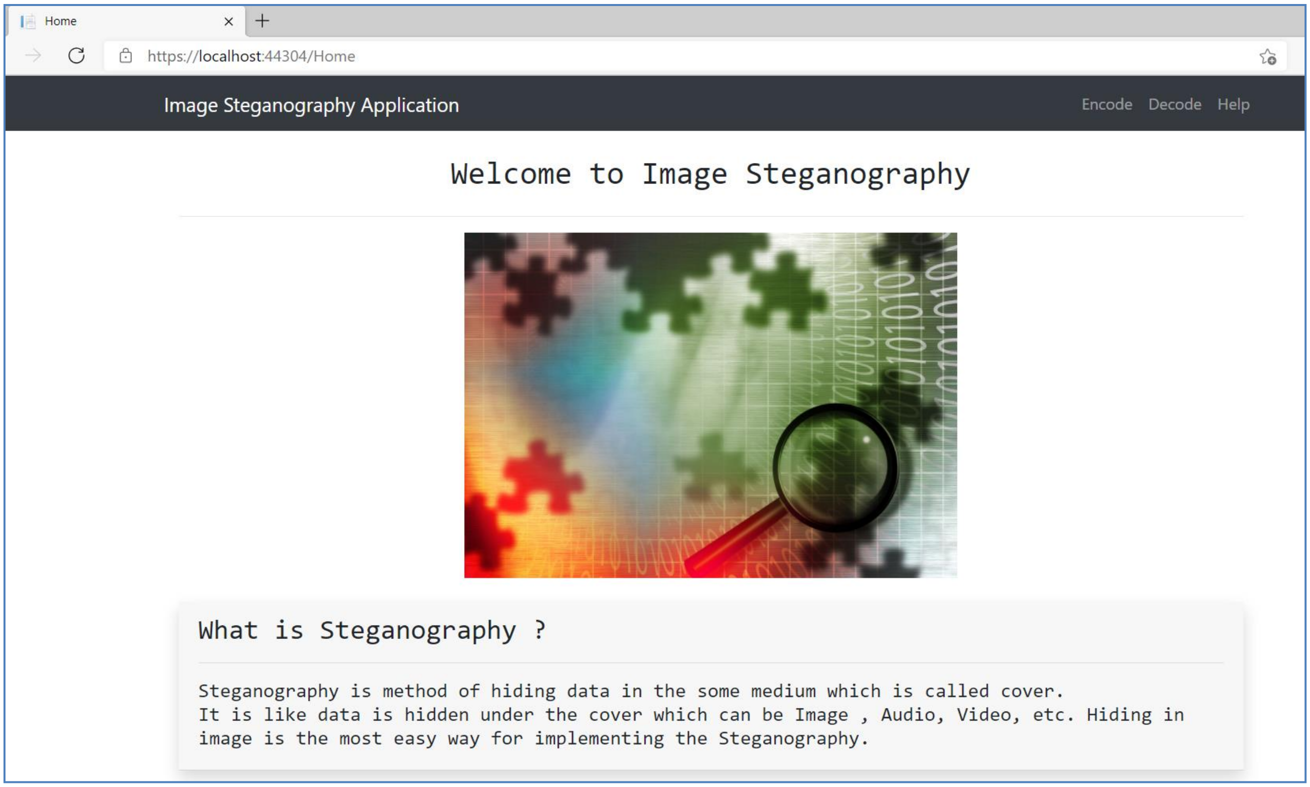
6.2.2 Decode Message

Test Case ID	Test Data (Stego Image)	Expected Output (Message)	Actual Output (Message)	Pass/Fail
T1	 For correct Key and Enc. Type	The best feeling in the world is to know that your Parents are Smiling because of You	The best feeling in the world is to know that your Parents are Smiling because of You	Pass
T2	 For correct Key and Enc. Type	Alls Well That Ends Well	Alls Well That Ends Well	Pass
T3	 For incorrect Key or Enc. Type	Empty message	“” Empty string indicates that there is no hidden text inside the image. It will not prompt the User that Key or Enc. Type is incorrect.	Pass

7. Screenshots

7.1 Output Screenshots:-

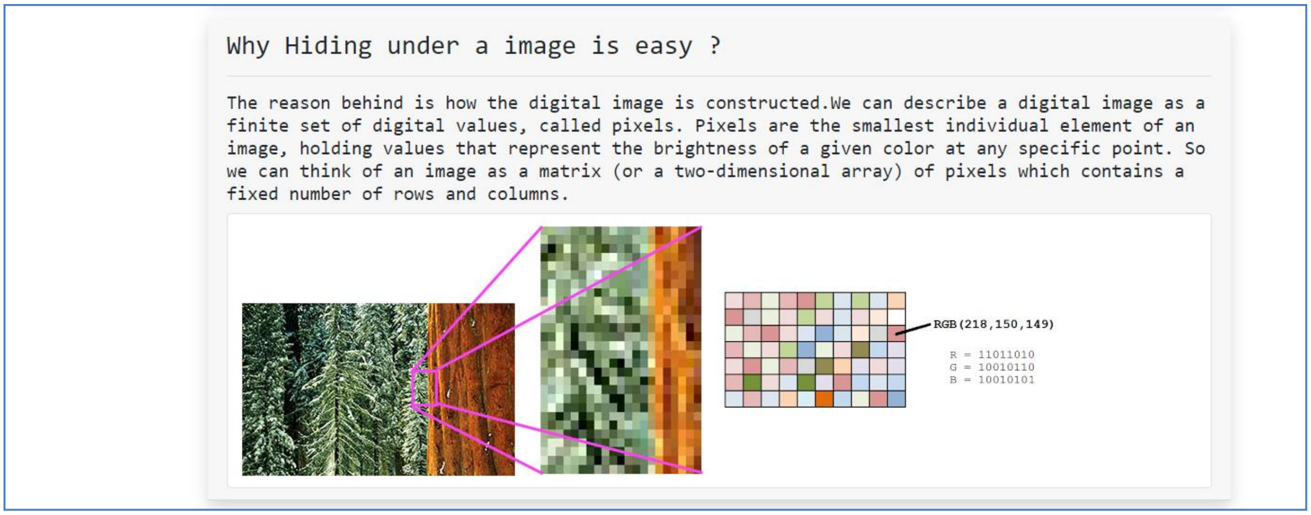
7.1.1 Home.aspx



Welcome to Image Steganography

What is Steganography ?

Steganography is method of hiding data in the some medium which is called cover. It is like data is hidden under the cover which can be Image , Audio, Video, etc. Hiding in image is the most easy way for implementing the Steganography.

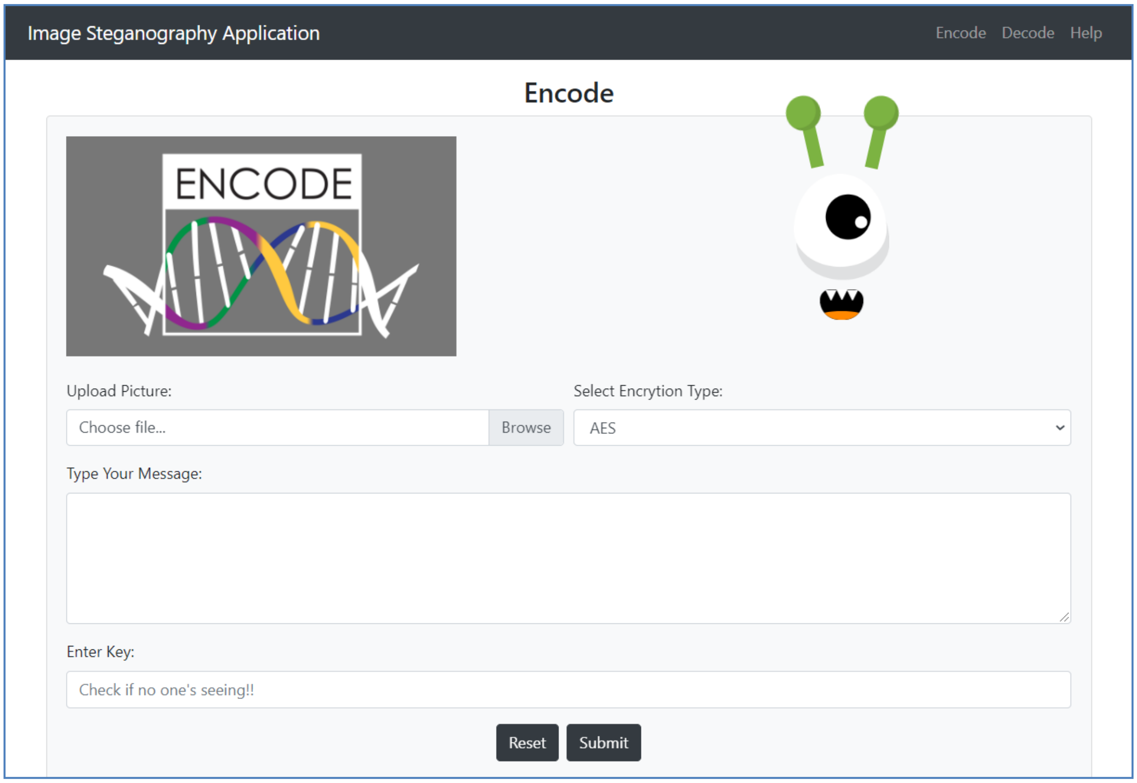


Why Hiding under a image is easy ?

The reason behind is how the digital image is constructed.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.

Fig. 7.1 Home page of the application

7.1.2 Encode.aspx

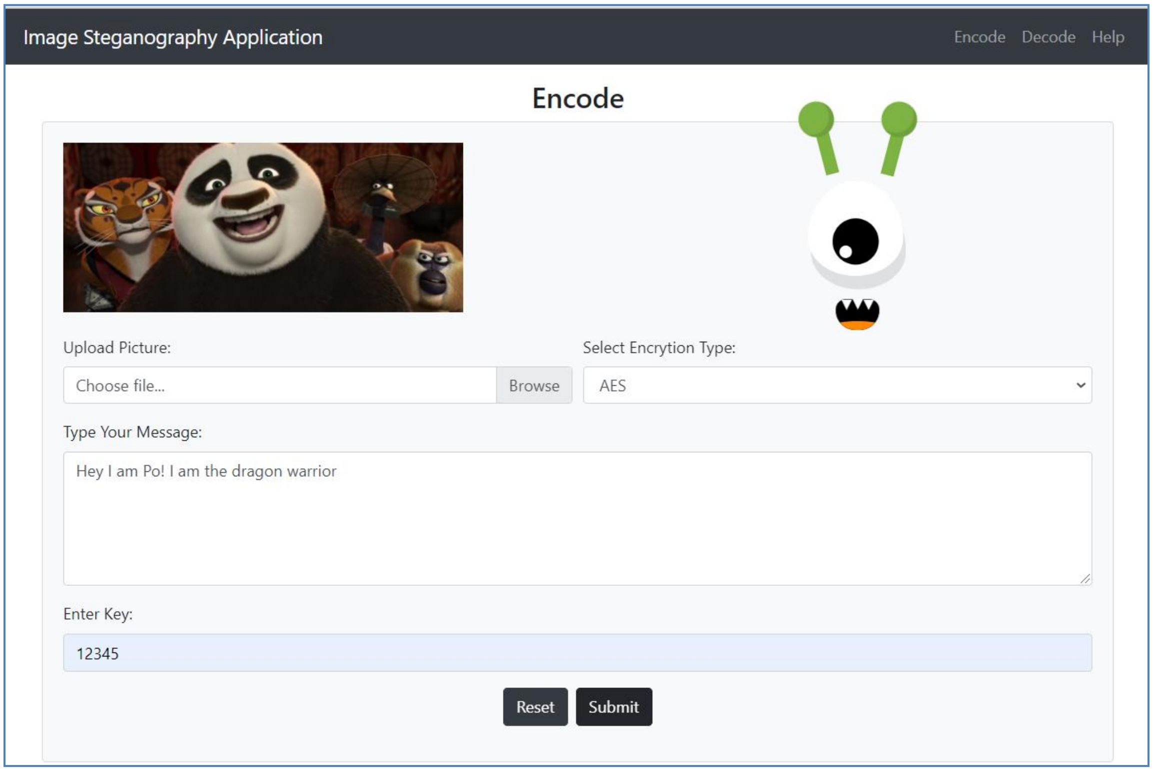


Upload Picture: Choose file...

Select Encryption Type: AES

Type Your Message:

Enter Key: Check if no one's seeing!!



Upload Picture: Choose file...

Select Encryption Type: AES

Type Your Message: Hey I am Po! I am the dragon warrior

Enter Key: 12345

Fig 7.2 Hiding message inside the image

7.1.3 DownloadImage.aspx

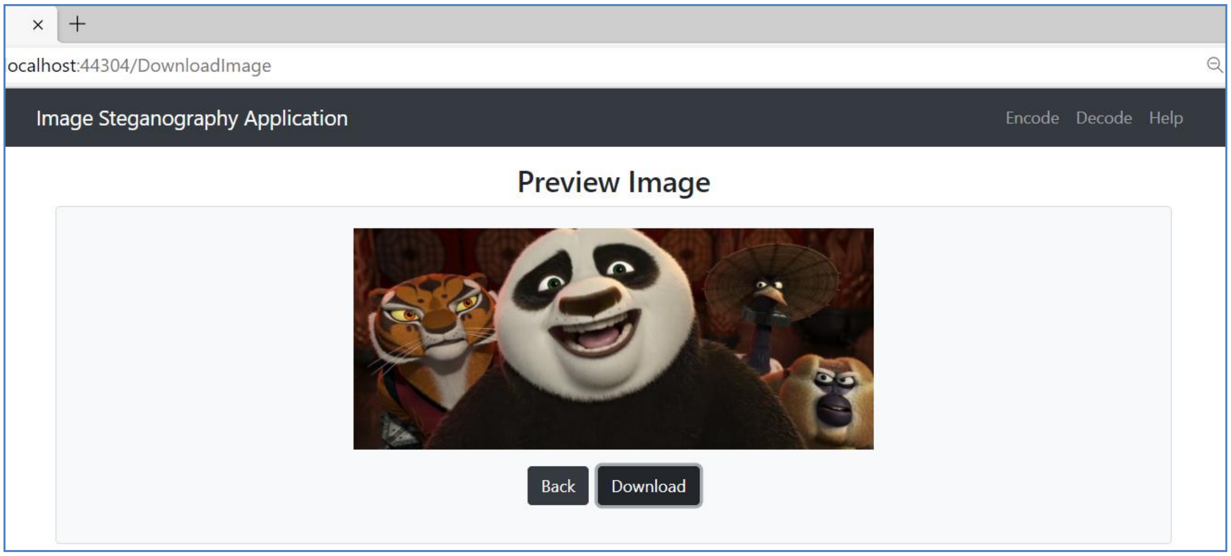


Fig. 7.3 Stego Image Preview

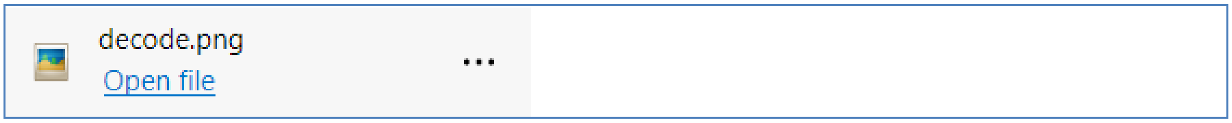


Fig. 7.4 Stego Image Downloaded

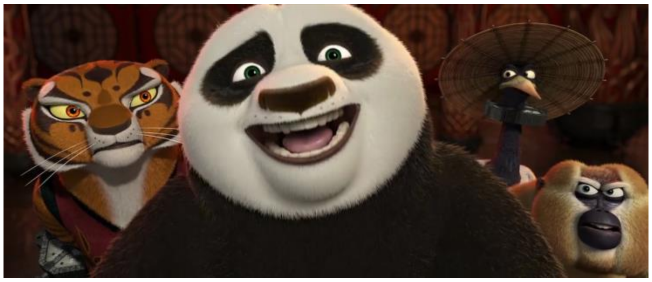
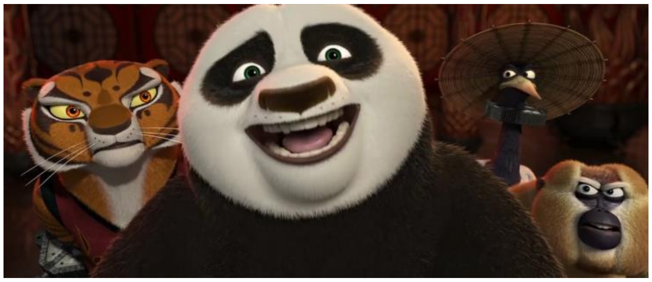
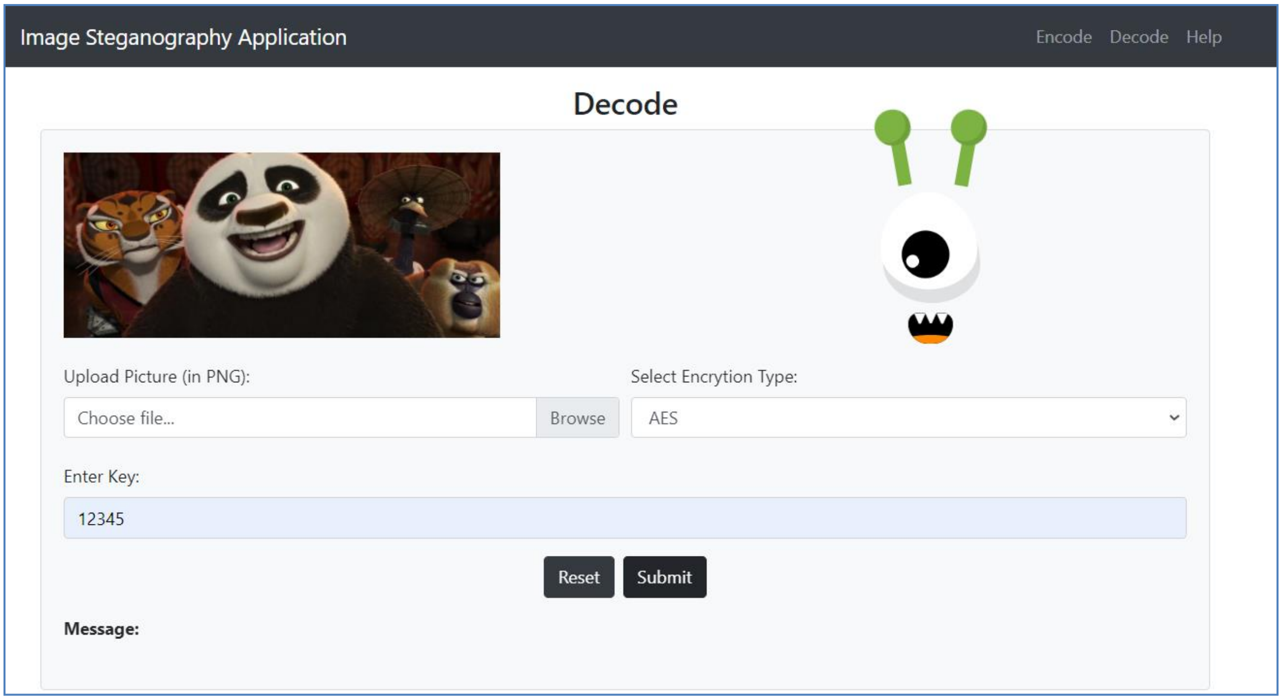
	

Image	Stego Image

Fig. 7.5 Difference between original image and stego image

7.1.4 Decode.aspx



Decode

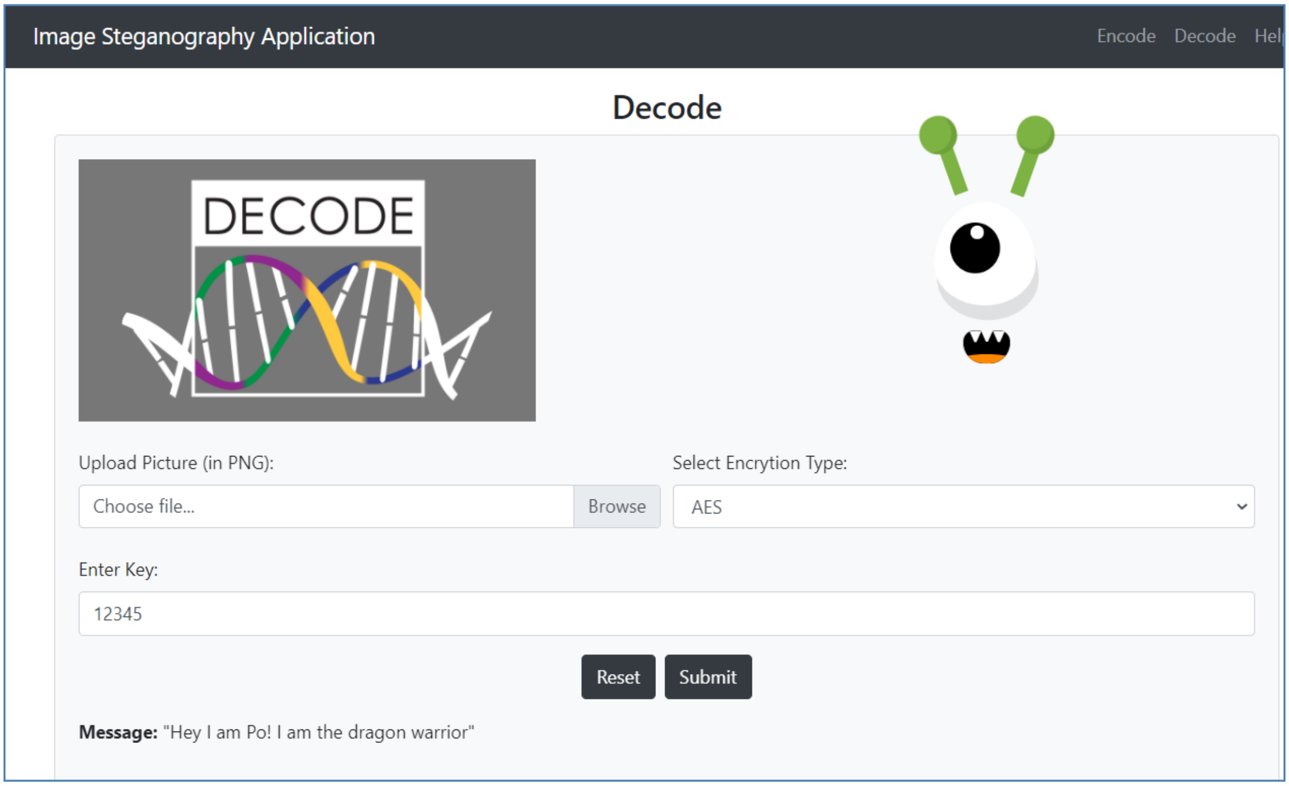
Upload Picture (in PNG): Choose file...

Select Encrytion Type: AES

Enter Key: 12345

Message:

Fig. 7.6 Extracting text from the image



Decode

Upload Picture (in PNG): Choose file...

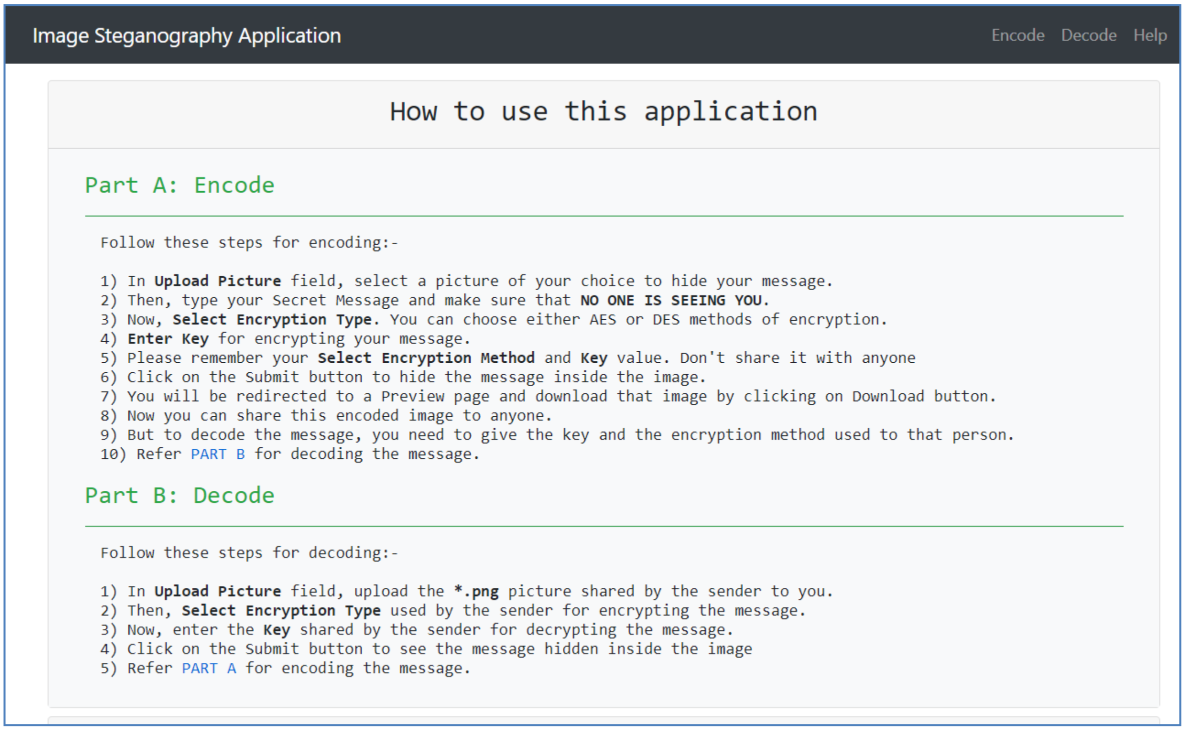
Select Encrytion Type: AES

Enter Key: 12345

Message: "Hey I am Po! I am the dragon warrior"

Fig. 7.7 Displaying Hidden Message

7.1.5 Help.aspx



How to use this application

Part A: Encode

```
Follow these steps for encoding:-

1) In Upload Picture field, select a picture of your choice to hide your message.
2) Then, type your Secret Message and make sure that NO ONE IS SEEING YOU.
3) Now, Select Encryption Type. You can choose either AES or DES methods of encryption.
4) Enter Key for encrypting your message.
5) Please remember your Select Encryption Method and Key value. Don't share it with anyone
6) Click on the Submit button to hide the message inside the image.
7) You will be redirected to a Preview page and download that image by clicking on Download button.
8) Now you can share this encoded image to anyone.
9) But to decode the message, you need to give the key and the encryption method used to that person.
10) Refer PART B for decoding the message.
```

Part B: Decode

```
Follow these steps for decoding:-

1) In Upload Picture field, upload the *.png picture shared by the sender to you.
2) Then, Select Encryption Type used by the sender for encrypting the message.
3) Now, enter the Key shared by the sender for decrypting the message.
4) Click on the Submit button to see the message hidden inside the image
5) Refer PART A for encoding the message.
```

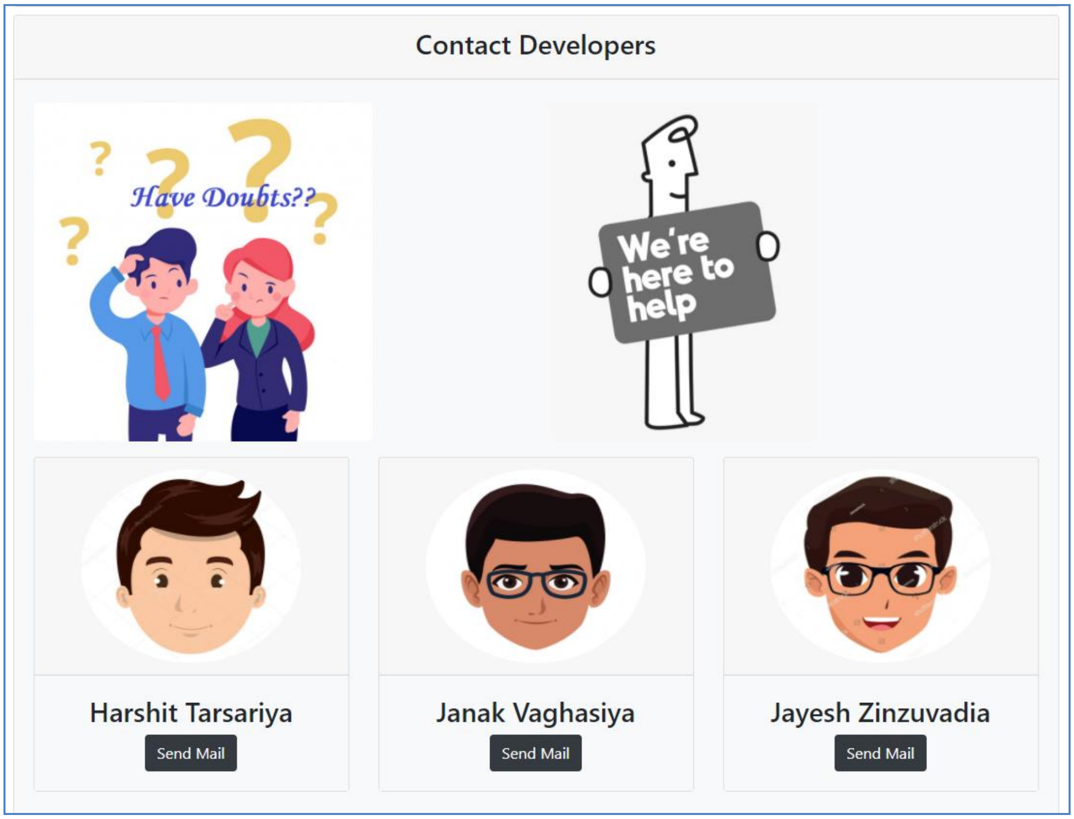


Fig. 7.8 User Manual (First) and Fig. 7.9 Contact Developers page (Second)

8. Conclusion

This project is aimed at developing *Image Steganography* project using the concept of LSB Technique, AES/DES methods and Web API Service.

The idea was to covert the message by first encrypting the message and then hiding it inside an image. The key for the encryption is known only to the sender and receiver. So that others can’t be able to interpret the message.

To convert the idea into implementation, we have used the ASP.NET Framework and have used C# language for coding the backend rest service and frontend as well.

So, after performing various tests, we conclude that our project is working successfully. We have implemented all the stated requirements. But of course, there is always a scope for improvement and learning.

In the end, we learn about service oriented architecture and project implementation in WCF and Web API.

We are now looking forward to overcome the existing limitations and to add some new features which are discussed in the next section.

9. Limitation and Future Extension

9.1 Limitation

- Data that can be embedded into the image is highly dependent on size of Image, so we can only embedded 3*(Image Width * Image Height) and additionally somewhat more by compression techniques.
- Message can be lost if image is in jpeg as it uses lossy compression.

9.2 Future Extension

- Currently there are two encryption standard DES and AES used, but can be extended by various other like RSA, ECC, Elgamal, etc.
- Digital Signature can be done for checking the Integrity or the check if data is corrupted in channel or not.

10. Bibliography

Following links and websites are referred during the project development:-

1) PMJ sir’s lecture Web API Slides/Screenshots
2) Microsoft Docs - https://docs.microsoft.com/en-in/
3) Google for any queries - https://www.google.com/
4) StackOverflow for solving the errors - https://stackoverflow.com/
5) Wikipedia article on Steganography - https://en.wikipedia.org/wiki/Steganography