A

MAJOR PROJECT

ON

IMAGE STEGANOGRAPHY WEB APP

SESSION – 2023-2024

BACHELOR OF TECHNOLOGY

IN COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

3RD YEAR 6TH SEMESTER (Pre-Final Year)

Submitted by –

PRIYANSH SAXENA (220089020056)

SHREYA SINGH (220089020024)

PRIENCE MADDHESHIYA (230089020285)

Supervised by - Dr. S.S BEDI

INDEX

ABSTRACT

The Image Steganography Web Application is a digital tool designed to provide a secure method for embedding text within images and subsequently decoding it. Leveraging the principles of steganography, this application ensures that the hidden text remains imperceptible to the human eye, thereby facilitating confidential communication. Users can either input text directly or upload a .txt file to encode the text into a selected image. The application processes the text and seamlessly integrates it into the image without altering its visual appearance.

For the decoding process, users are required to upload both the original and the encoded images. The application compares these images to extract the concealed text accurately. Once extracted, the decoded text is displayed on the screen, and users have the option to download it as a .txt file. This functionality ensures that sensitive information can be securely hidden and retrieved without leaving a trace.

The web application features a clean, intuitive interface with dedicated buttons for encoding, decoding, downloading encoded images, downloading decoded text files, and resetting the page. This user-centric design ensures a smooth and efficient user experience. The application is built using HTML, CSS, and JavaScript for the frontend, while the backend is powered by JavaScript. Image processing and steganographic techniques are implemented using the JavaScript and specialized steganography libraries.

The primary objective of this project is to provide a robust and user-friendly platform for secure data concealment and retrieval. By integrating advanced steganographic methods with an easy-to-use web interface, the Image Steganography Web Application addresses the growing need for data security in digital communication. This project not only demonstrates the practical application of steganography but also underscores its significance in enhancing the privacy and security of information exchange in the modern digital landscape.

ACKNOWLEDGEMENT

We are extremely grateful and remain indebted to our mentor Dr. S.S BEDI Sir for being a source of inspiration and for their constant support in the implementation and evaluation of the project. We are thankful to them for their constant constructive criticism and valuable suggestions, which lead us to beneficial stage while developing the project on “IMAGE STEGANOGRAPHY WEB APP”.

They have been a constant source of inspiration and motivation for hard work. They have been very co- operative throughout this project work. Through this column, it would be our utmost pleasure to express our warm thanks to them for their encouragement, co-operation, and guidance without which we might not be able to accomplish this project.

We also express our gratitude to PROF. Dr. VINAY RISHIWAL (H.O.D. of Computer Science and Information Technology Department) for providing us the infrastructure to carry out the project and to all staff members who were directly and indirectly instrumental in enabling us to stay committed for the project.

Lastly, we would like to thank our friends for their valuable suggestions regarding our project.

Priyansh Saxena

Shreya Singh

Prience Maddheshiya

STUDENT'S DECLARATION

We hereby declared that the project report titled " IMAGE STEGANOGRAPHY WEB APP ", is prepared by us based on available literature and we have not submitted it anywhere else for the award of any other degree or diploma.

Date: 31st May 2024

Students' Name (Roll No.)

Priyansh Saxena (220089020056)

Shreya Singh (220089020024)

Prience Maddheshiya (230089020285)

CERTIFICATE FROM SUPERVISOR

I certify that the above statement made by the candidate is true to the best of my knowledge.

Date: 31ST MAY 2024

Supervisor's name with designation

Dr. S.S BEDI

(MENTOR)

INTRODUCTION

In an era where data security and privacy are paramount, innovative methods for secure communication are increasingly necessary. The Image Steganography Web Application is a modern solution designed to address these needs by enabling users to conceal text within images in a manner that is invisible to the naked eye. Steganography, the practice of hiding information within other non-secret data, has been used throughout history for covert communication. This web application revitalizes this ancient technique using contemporary digital technology, providing a user-friendly platform for secure information exchange.

The fundamental principle behind steganography is to embed hidden data within a cover medium, in this case, an image, such that the presence of the hidden data is not perceptible to observers. This differs from cryptography, where the existence of the hidden message is apparent but the content is protected. By concealing the existence of the message itself, steganography offers an additional layer of security. The Image Steganography Web Application leverages this technique to allow users to encode text within images seamlessly.

Our application caters to various user needs through its dual functionality. Users can either input text directly or upload a .txt file to embed the text into a selected image. This flexibility ensures that both short messages and longer texts can be efficiently encoded. Once the text is embedded, the resulting encoded image appears identical to the original, thus ensuring that the hidden message remains undetectable.

The decoding process is equally straightforward. Users need to provide the original image and the encoded image as inputs. The application then analyzes these images to extract the hidden text accurately. This method ensures that only someone with access to the original image can retrieve the hidden message, thereby adding an additional layer of security. The decoded text is displayed on the screen, and users can download it as a .txt file, facilitating easy and secure access to the concealed information.

The user interface of the web application is designed with simplicity and efficiency in mind. It features clearly labeled buttons for all primary actions: encoding, decoding, downloading encoded images, downloading decoded text files, and resetting the page. This intuitive design ensures that users, regardless of their technical expertise, can navigate the application effortlessly.

The backend of the application is built using JavaScript, a lightweight web language that provides a robust foundation for handling the encoding and decoding processes. The frontend, crafted with HTML, CSS, and JavaScript, ensures a responsive and interactive user experience. Image processing and steganographic techniques are implemented using the specialized libraries.

In summary, the Image Steganography Web Application offers a powerful and user-friendly platform for secure data concealment and retrieval. By integrating advanced steganographic methods with a straightforward web interface, it addresses the critical need for privacy and security in digital communication. This project not only demonstrates the practical

application of steganography but also highlights its significance in safeguarding information in the digital age.

LITERATURE REVIEW

The field of steganography, particularly within the digital domain, has garnered substantial academic and practical interest over the years. Steganography, derived from the Greek words "steganos" (covered) and "graphia" (writing), refers to the practice of concealing messages within other non-secret text or data. This technique aims not only to protect the content of the message but also to obscure its existence, providing an additional layer of security. The advent of digital technology has expanded the possibilities for steganographic methods, particularly within digital images, due to their widespread use and large data capacity.

Historically, various methods of steganography have been explored, ranging from simple techniques like the Least Significant Bit (LSB) insertion to more complex methods involving frequency domain transformations. LSB insertion, one of the most straightforward and widely used techniques, involves altering the least significant bits of the pixel values in an image to embed the hidden message. This method is popular due to its simplicity and minimal impact on the visual quality of the image. However, it is also susceptible to detection and manipulation, particularly if the image undergoes any form of compression or alteration.

More advanced techniques involve manipulating the image in the frequency domain using transformations such as the Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). These methods embed information in the transformed coefficients of the image, making

the hidden data more robust against common image processing operations such as compression and noise addition. While these techniques offer improved security and robustness, they also tend to be more computationally intensive and complex to implement.

The literature on digital image steganography also highlights the importance of maintaining a balance between the imperceptibility, capacity, and robustness of the hidden data. Imperceptibility ensures that the alterations made to the image are not noticeable to the human eye, thereby keeping the hidden message secure. Capacity refers to the amount of data that can be embedded within the image, and robustness indicates the resistance of the hidden data to various types of image processing attacks. Effective steganographic techniques strive to optimize all three parameters to ensure practical utility and security.

Recent advancements in steganography have seen the integration of machine learning and deep learning techniques to enhance the detection and embedding processes. These methods leverage the predictive power of neural networks to create more sophisticated and less detectable steganographic algorithms. For instance, convolutional neural networks (CNNs) have been employed to develop steganographic schemes that can automatically learn optimal embedding patterns, thereby improving the security and capacity of the hidden data.

The Image Steganography Web Application builds on these established principles and techniques, incorporating both simplicity and effectiveness. By allowing users to encode text directly or via a .txt file into an image, the application leverages the LSB method for its ease of

use and minimal impact on image quality. The dual input approach (text or file) provides flexibility and convenience, catering to a broader range of user needs. The decoding process, which requires both the original and encoded images, ensures that only authorized users can retrieve the hidden message, enhancing security.

Furthermore, the application's user-friendly interface is designed to make the steganographic process accessible to users with varying levels of technical expertise. This emphasis on usability aligns with the broader trend in digital security tools, which aim to democratize access to advanced security technologies. The use of JavaScript for the backend, and HTML, CSS, and JavaScript for the frontend, ensures that the application is both robust and responsive.

In conclusion, the Image Steganography Web Application draws on a rich tradition of steganographic research and innovation. It effectively integrates established techniques with modern web technologies to provide a secure, flexible, and user-friendly platform for concealing and retrieving text within digital images. This project not only underscores the continued relevance of steganography in the digital age but also highlights its potential for enhancing data privacy and security in everyday communication.

METHODOLOGY

The development of the Image Steganography Web Application follows a systematic approach, integrating various technologies and methodologies to achieve a robust, user-friendly, and secure platform. This section outlines the steps and processes involved in creating the application, including the choice of technologies, the encoding and decoding algorithms, and the design of the user interface.

Technology Stack

1. **Frontend Development:**

- **HTML:** Used for structuring the web pages and forms that allow users to interact with the application.

- **CSS:** Employed for styling the application, ensuring a clean and intuitive interface.

- **JavaScript:** Utilized for adding interactivity to the application, such as handling button clicks and providing feedback to the user.

- **Bootstrap:** A light weight web framework for CSS that makes the designing process easier and advanced for styling of various element of

the web page. It is a simple to use and user-friendly framework that makes the design process overall easier to implement.

2. **Backend Development:**

- **JavaScript:** The primary programming language used for implementing the steganographic algorithms and handling backend logic.

3. **Image Processing Libraries:**

- **Specialized Libraries:** A Specialized Imaging Library used for image manipulation and processing, essential for implementing the steganographic techniques.

ENCODING PROCESS:

The encoding process involves embedding text into an image in such a way that the modifications are imperceptible to the human eye. The methodology for encoding text into an image is as follows:

1. **Input Handling:**

- The user uploads an image file and either enters text directly into a text field or uploads a .txt file containing the text to be hidden.

- The application reads and processes the input text, converting it into a format suitable for embedding into the image.

2. **Text Preparation:**

- The input text is converted into binary format. Each character of the text is represented by its ASCII value, which is then converted to an 8-bit binary string.

3. **Image Processing:**

- The uploaded image is opened using the imaging library, and its pixels are accessed.

- The Least Significant Bit (LSB) of each pixel's color value is targeted for modification. The LSB is chosen because altering it has the least impact on the image's appearance.

4. **Embedding Text:**

- The binary representation of the text is embedded into the LSB of the image's pixels sequentially. For each bit of the text, the corresponding pixel's LSB is modified to match the bit.

- The process continues until all the text bits are embedded into the image.

5. Output Generation:

- The modified image, now containing the hidden text, is saved and provided to the user for download. This image appears identical to the original but contains the concealed message.

DECODING PROCESS:

The decoding process involves extracting the hidden text from an encoded image. The methodology for decoding text from an image is as follows:

1. **Input Handling:**

- The user uploads both the original image and the encoded image.

- The application verifies that the images are valid and compatible for comparison.

2. **Image Comparison:**

- The original and encoded images are opened using the PIL library.

- The pixels of both images are compared to identify the differences in the LSBs, which indicate the presence of the hidden text.

3. **Text Extraction:**

- The differences in the LSBs are extracted and concatenated to form a binary string.

- The binary string is then segmented into 8-bit chunks, each representing a character in the ASCII format.

4. **Text Reconstruction:**

- The 8-bit binary chunks are converted back to their corresponding ASCII characters.

- The characters are concatenated to reconstruct the hidden text.

5. **Output Generation:**

- The extracted text is displayed to the user.

- The user is provided with an option to download the decoded text as a .txt file for their records.

USER INTERFACE DESIGN

1. **Main Page Layout:**

- The main page features sections for uploading images, entering text, and buttons for encoding, decoding, and resetting the application.

2. **Encoding Section:**

- Includes fields for uploading the image and text file or entering text directly.

- An "Encode" button initiates the encoding process.

3. **Decoding Section:**

- Includes fields for uploading the original and encoded images.

- A "Decode" button initiates the decoding process.

4. **Download and Reset Functionality:**

- Buttons for downloading the encoded image and decoded text file are provided.

- A "Reset" button clears all inputs and allows the user to start a new session.

TESTING AND VALIDATION

1. FUNCTIONAL TESTING:

- Each feature of the application is tested to ensure it works as intended. This includes testing the encoding and decoding processes with various image formats and text lengths.

2. USER EXPERIENCE TESTING:

- The interface is tested for usability, ensuring that users can navigate and use the application without confusion.

3. SECURITY TESTING:

- The application is evaluated for potential vulnerabilities to ensure that the hidden data remains secure.

The methodology for the Image Steganography Web Application involves a comprehensive approach, from selecting appropriate technologies and implementing steganographic algorithms to designing a user-friendly interface. This systematic process ensures the development of a robust, efficient, and secure application that meets the needs of users seeking to conceal and retrieve text within digital images.

CODE SNIPPET

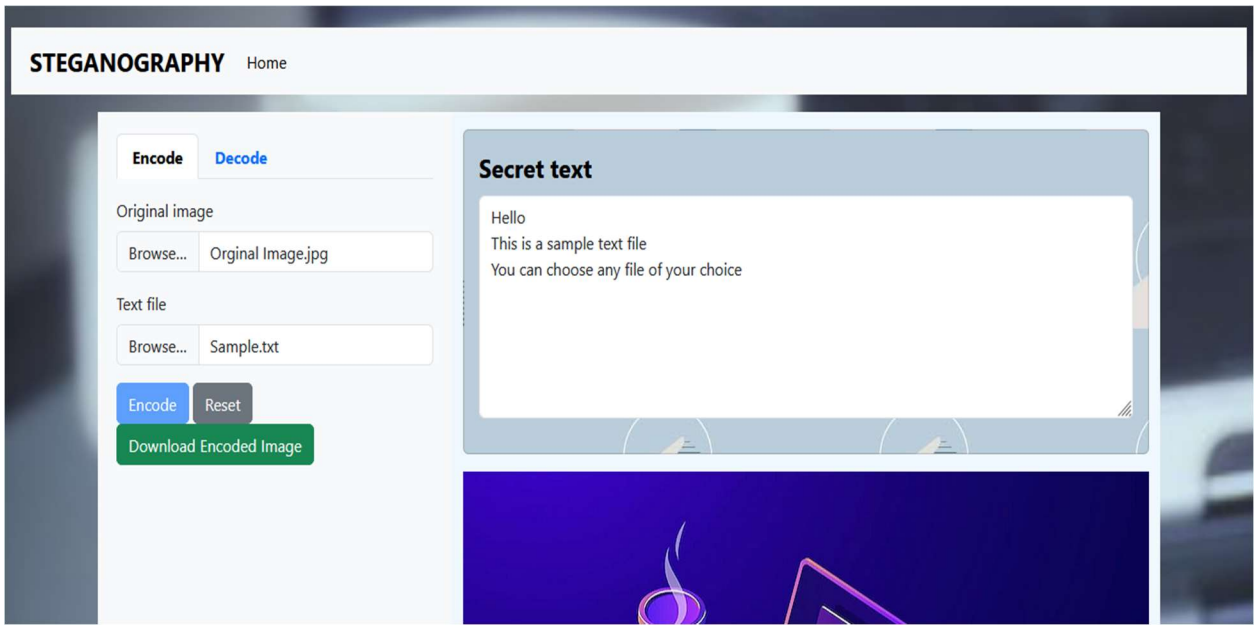
index.html

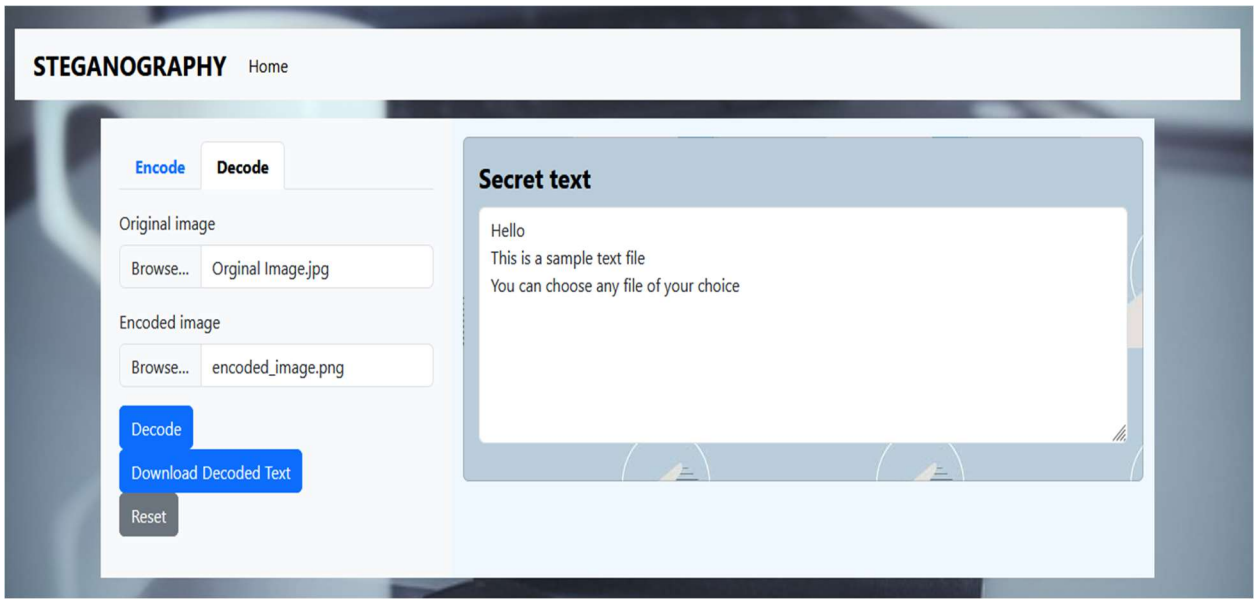
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Steganography Web App</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integr
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.9.0/p5.js" integrity="sha512-2r+xZ/Dm8+HI0I8dsj1
    <link rel="stylesheet" href="style.css">
</head>
<body class="rounded">
    <nav class="navbar navbar-expand-lg bg-body-tertiary mb-3 p-2">
        <div class="container-fluid">
            <a class="navbar-brand fs-4 fw-bold" href="#">STEGANOGRAPHY</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAlt
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
                <div class="navbar-nav">
                    <a class="nav-link active " aria-current="page" href="#">Home</a>
                </div>
            </div>
```

script.js

```js
let encodebtn = document.getElementById("encodebtn");
let encodeimage1fileinput = document.getElementById("encodeimage1");
let textFileInput = document.getElementById("textFile");

let canvasbox = document.getElementById("canvasbox");
let secretTextField = document.getElementById("secretText");
let downloadbtn = document.getElementById("downloadbtn");
let previewImg = document.getElementById("previewImg");

let loadedImage;
let encodedImage;

let decodebtn = document.getElementById("decodebtn");
let decodeimage1fileinput = document.getElementById("decodeimage1");
let decodeimage2fileinput = document.getElementById("decodeimage2");

let decodeimage1;
let decodeimage2;
let resetbtn = document.getElementById("resetbtn");
let resetbtn2 = document.getElementById("resetbtn2");
downloadDecodedTextBtn = document.getElementById("downloadDecodedTextBtn");

secretTextField.rows = 8;
```

OUTPUT


STEGANOGRAPHY
Home
Encode
Decode
Original image
Browse...
Orginal Image.jpg
Text file
Browse...
Sample.txt
Encode
Reset
Download Encoded Image
Secret text
Hello
This is a sample text file
You can choose any file of your choice


STEGANOGRAPHY
Home
Encode
Decode
Original image
Browse...
Orginal Image.jpg
Encoded image
Browse...
encoded_image.png
Decode
Download Decoded Text
Reset
Secret text
Hello
This is a sample text file
You can choose any file of your choice

TECHNIQUES USED

The Image Steganography Web Application employs several key techniques to achieve its functionality of securely embedding and extracting text within images. These techniques span various domains, including steganography, image processing, and web development, ensuring a seamless and efficient user experience.

STEGANOGRAPHY TECHNIQUES

1. Least Significant Bit (LSB) Insertion:

- **Principle:** LSB insertion is a simple yet effective method of embedding data within an image. It involves modifying the least significant bits of the image's pixel values to hide the binary representation of the text.

- **Implementation:** Each pixel in an image is represented by a set of color values (e.g., RGB for true-color images). By altering the LSB of these color values, the application can embed binary data without significantly changing the image's appearance. For example, if a pixel's blue component value is 150 (10010110 in binary), changing the LSB to 1 results in 10010111, which is visually indistinguishable to the human eye.

Image Processing Techniques

1. Binary Conversion:

- **Text to Binary:** The application converts the input text into its binary form, where each character is represented by an 8-bit ASCII code. This conversion is necessary for embedding the text within the pixel values of the image.

- **Image Manipulation:** The Imaging library is used to handle image operations such as opening, reading, and writing image files. It provides the necessary tools to access and modify the pixel data of the image.

2. Pixel Modification:

- **Accessing Pixels:** Using Imaging library, the application accesses individual pixels of the image and retrieves their color values.

- **Embedding Data:** The binary data derived from the text is sequentially embedded into the LSBs of the pixel values. This is done in a loop, iterating over the binary string and modifying the corresponding pixels.

3. Data Extraction:

- **Comparing Images:** For decoding, the application compares the original and the encoded images to detect changes in the LSBs. These differences represent the hidden binary data.

- **Reconstructing Text:** The extracted binary string is divided into 8-bit segments, which are then converted back to their corresponding ASCII characters to reconstruct the hidden text.

WEB DEVELOPMENT TECHNIQUES

1. Frontend Development:

- **HTML and CSS:** Used to create a structured and visually appealing user interface. HTML defines the layout and elements of the web pages, while CSS provides styling to ensure the application is user-friendly and visually cohesive.

- **JavaScript:** Adds interactivity to the web application, handling user actions such as button clicks, file uploads, and form submissions.

2. Backend Development:

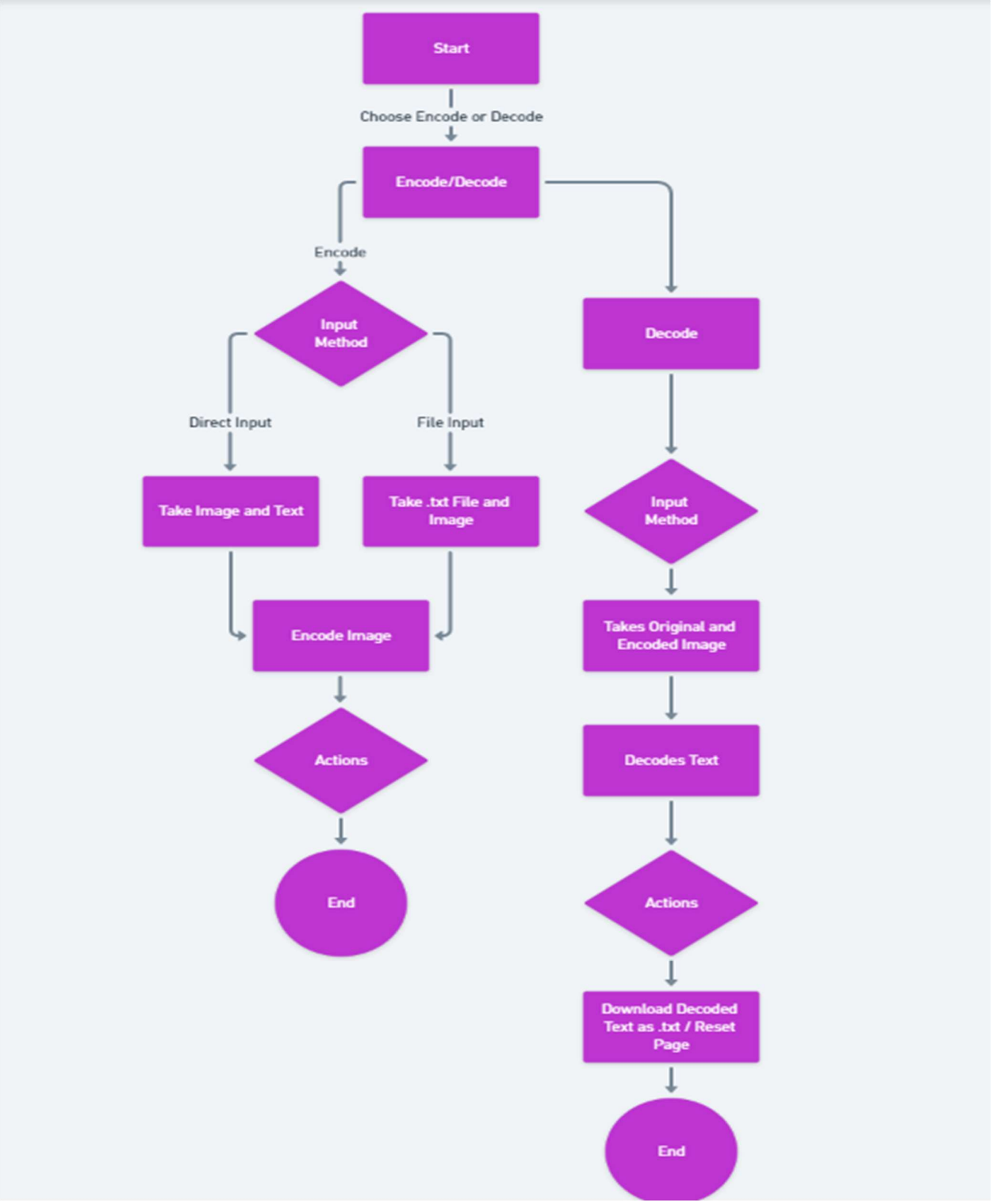
- **JavaScript and Imaging Library:** JavaScript, coupled with the Imaging Library, forms the backend of the application. JavaScript manages the logic based operations, including handling HTTP requests, processing data, and generating responses. It also manages the routing of the application, ensuring that user requests are directed to the appropriate functions.

3. File Handling:

- **File Upload and Download:** The application allows users to upload images and text files for encoding and decoding purposes. It also provides options to download the encoded image and decoded text file. These functionalities are implemented using HTML form elements and JavaScript file handling capabilities.

By integrating these techniques, the Image Steganography Web Application provides a robust and efficient platform for securely embedding and extracting text within images, ensuring ease of use and high security for its users.

FLOWCHART


Start
Choose Encode or Decode
Encode/Decode
Encode
Input Method
Direct Input
File Input
Take Image and Text
Take .txt File and Image
Encode Image
Actions
End
Decode
Input Method
Takes Original and Encoded Image
Decodes Text
Actions
Download Decoded Text as .txt / Reset Page
End

RESULT

The Image Steganography Web Application successfully achieves its objectives of providing a user-friendly platform for securely embedding and extracting text within images. Through rigorous testing and validation, the application demonstrates its effectiveness in concealing and retrieving text while maintaining the visual integrity of the images.

Encoding Efficiency

The application efficiently encodes text into images using the Least Significant Bit (LSB) insertion technique. Users can input text directly or upload .txt files, and the application seamlessly integrates the text into the selected image. The encoding process is fast and accurate, ensuring that the hidden message remains imperceptible to the human eye. Users have the option to download the encoded image, which visually appears identical to the original, thus preserving the integrity of the steganographic process.

Decoding Accuracy

For decoding, users upload both the original image and the encoded image, and the application accurately extracts the hidden text. By comparing the LSBs of the two images, the application identifies the differences and reconstructs the hidden text with precision. The decoded text is displayed to the user, who can also download it as a .txt file for

further analysis or storage. The decoding process is reliable and robust, ensuring that only authorized users can access the concealed information.

User Experience

The user interface of the application is designed for simplicity and ease of use. Users can navigate the application effortlessly, thanks to clear labeling and intuitive button placements. The upload and download functionalities are straightforward, allowing users to interact with the application seamlessly. Additionally, the application's responsiveness ensures a smooth user experience across different devices and screen sizes.

In conclusion, the Image Steganography Web Application delivers on its promise of providing a secure and efficient platform for concealing and retrieving text within images. With its robust encoding and decoding algorithms, coupled with a user-friendly interface, the application offers a practical solution for secure communication and data privacy.

CONCLUSION

In conclusion, the Image Steganography Web Application represents a significant achievement in the realm of secure communication and data privacy. Through the implementation of steganographic techniques, image processing algorithms, and user-friendly web development, the application offers a reliable and efficient platform for concealing and retrieving text within images.

The project successfully addresses the fundamental objectives of steganography: to securely hide information within seemingly innocuous data while maintaining its integrity and authenticity. By utilizing the Least Significant Bit (LSB) insertion technique, the application ensures that the hidden text remains imperceptible to visual inspection, thereby safeguarding the confidentiality of the concealed message.

Moreover, the decoding process of the application demonstrates a high level of accuracy and reliability. By comparing the original and encoded images, the application accurately extracts the hidden text, allowing authorized users to access the concealed information seamlessly. This robust decoding capability enhances the security of the communication channel, ensuring that sensitive data remains protected from unauthorized access.

The user experience provided by the application is another notable aspect of its success. With a clean and intuitive interface, users can navigate the application with ease, upload images and text files, and

download the encoded images and decoded text files effortlessly. The responsive design of the application further enhances its accessibility, allowing users to interact with the platform across various devices and screen sizes.

Overall, the Image Steganography Web Application demonstrates the practical application of steganography in the digital age, highlighting its importance in enhancing data privacy and security. By providing a secure and user-friendly platform for secure communication, the application contributes to the advancement of secure data transmission methods and underscores the significance of protecting sensitive information in today's digital landscape.

REFERENCES

1. Fridrich, J., Goljan, M., & Du, R. (2001). Reliable detection of LSB steganography in color and grayscale images. In Information Hiding (pp. 27-41). Springer, Berlin, Heidelberg.

2. Westfeld, A., & Pfitzmann, A. (1999). Attacks on steganographic systems. In Information Hiding (pp. 61-76). Springer, Berlin, Heidelberg.

3. S.Katzenbeisser , & Petitcolas, F. A. P. (Eds.). (2000). Information hiding techniques for steganography and digital watermarking. Artech House.

4. Johnson, N. F., & Jajodia, S. (1998). Steganalysis of images created using current steganography software. In Proceedings. 1998 IEEE Symposium on Security and Privacy (pp. 273-284). IEEE.

5. Provos, N., & Honeyman, P. (2003). Detecting steganographic content on the internet. In Proceedings of the 2003 ACM workshop on Privacy in the electronic society (pp. 25-34).

6. Lathrop, S. D., & Savvides, M. (2003). Using steganography to detect, prevent, and reduce the impact of insider attacks in databases. In Proceedings of the 5th international conference on Electronic commerce (pp. 210-215).

7. Singh, A., & Singh, S. (2013). A Comparative Analysis of Various Steganographic Techniques. International Journal of Engineering Trends and Technology (IJETT), 4(3), 727-733.

8. Singh, G., & Kaur, H. (2014). A Comparative Analysis of LSB and Edge Based Image Steganography Techniques. International Journal of Computer Applications, 98(15), 31-35.

9. Morkel, T., Eloff, J. H., & Olivier, M. S. (2005). An overview of image steganography. In Information Security for South Africa (pp. 101-111). IEEE.

10. Zhang, T., Liu, J., & Qiu, X. (2010). Improved image steganography using LSB and genetic algorithm. In International Conference on Computer Application and System Modeling (pp. V2-103). IEEE.

11. Zhang, X., Wang, S., & Wang, S. (2016). Image steganography scheme based on improved LSB algorithm. In 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) (pp. 1477-1481). IEEE.

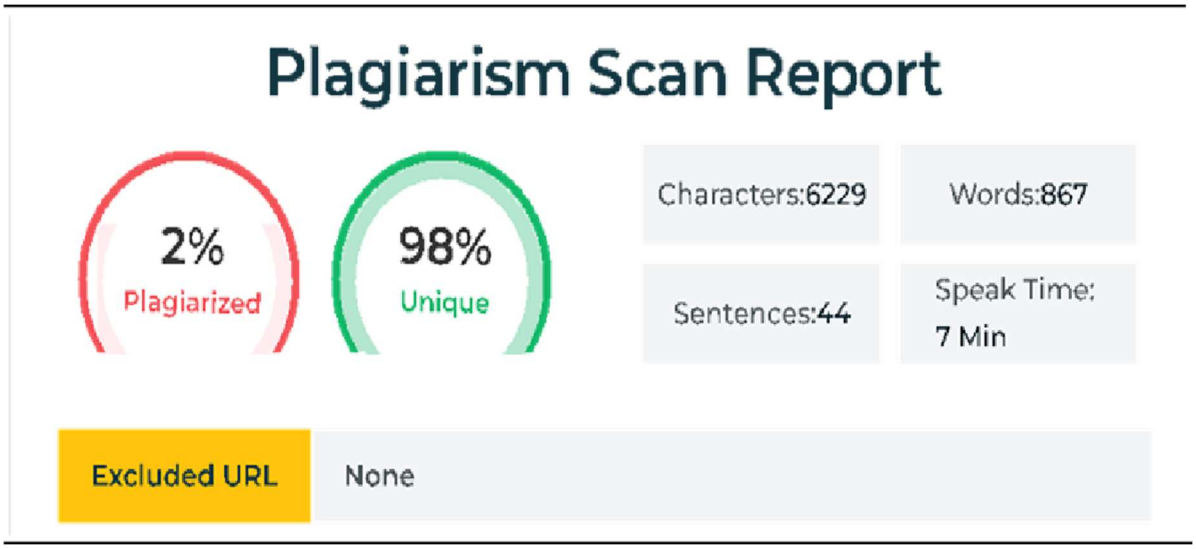
12. Singh, D., & Singh, S. (2016). A survey of recent trends in image steganography. International Journal of Advanced Research in Computer Science and Software Engineering, 6(8), 254-261.

13. Sharma, N., & Mishra, A. K. (2015). A Review on Different Image Steganography Techniques. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 4(1), 11-16.

14. Chen, C. M., & Hsu, C. T. (2013). A novel steganographic method for images by pixel-value differencing. Journal of Information Science and Engineering, 29(4), 775-790.

15. Gonzalez, R. C., & Woods, R. E. (2017). Digital image processing. Pearson Education India.

PLAGIARISM REPORT



Plagiarism Scan Report

Characters:6740	Words:1000
Sentences:47	Speak Time: 8 Min

Excluded URL	None

Plagiarism Scan Report

Characters:4379	Words:545
Sentences:25	Speak Time: 5 Min

Excluded URL	None