CS3430 Fundamentals of Cyber security

Mini Project Report

Secure Steganography Web Application with User Authentication

Submitted

By

Adhithya C (1RUA24CSE7000)

Sohan PM (1RUA24CSE7016)

Parinitha RK (1RVU23CSE329)

Under the guidance of:

Dr./Prof. Sunil Kumar J

School of Computer Science and Engineering

RV University, Bangalore

RV UNIVERSITY

Go, change the world

an initiative of RV EDUCATIONAL INSTITUTIONS

School of Computer Science and Engineering

CERTIFICATE

Certified that the CS3430 Fundamentals of Cyber Security Mini Project work titled Secure Steganography Web Application with User Authentication is carried out by , Adhithya C (1RUA24CSE7000),Sohan PM (1RUA24CSE7016) and Parinitha RK (1RVU23CSE329) who are bonafide students of the School of Computer Science and Engineering, RV University, Bengaluru, during the year 2025–26. It is certified that all corrections/ suggestions from all the continuous internal evaluations have been incorporated into the project and in this report.

Dr./ Prof. Sunil Kumar J

Faculty Guide

Program Director

1 Abstract

The biggest concerns in today's digital world are the security of data and privacy. This project, "Secure Steganography Web Application with User Authentication," proposes a system that will embed secret messages in images using steganography but permits access to the secret messages to only authorized people. The system is developed in Flask (Python) using SQLite, integrating MFA-password and verification via email (or OTP)-prior to allowing decoding. As compared to typical steganography, this project provides user authentication and access control, hence a more secure and surreptitious method of communication.

2 Problem statement

Traditional steganography allows any user to encode and decode hidden data within an image, making it vulnerable if the encoded file is accessed by unauthorized users. One of the key issues that this project addresses is the lack of secure access control in steganographic systems.

Our approach towards a solution introduces MFA to verify the credentials of the user and one more authentication factor, like an OTP or verification through an email, before decoding. This will ensure that even if the attacker gets hold of the encoded image, still without authentication, he or she cannot retrieve the hidden message, hence achieving data confidentiality and user-level access control.

3. Introduction

While digital networks are becoming conduits for information exchange, ensuring confidentiality of sensitive data has become one of the most critical challenges. Traditional methods of encryption are secure, yet all suspicious-looking data invites attacks. Steganography hides the very existence of the data.

Steganography is the process of hiding secret messages in a medium like image, audio, or video. This project uses image steganography with the method of Least Significant Bit (LSB), where message bits are hidden in the least significant bits of pixel values. This technique provides subtle yet effective data hiding without altering the visual appearance of the image.

However, the previous steganography systems did not include authorization and verification; thus, any person who possesses the decoding program can retrieve the message from the hidden messages. In this regard, our system implements MFA to verify that only the registered and verified users can decode messages. Adding an additional layer of security, MFA necessitates both a password and a secondary verification code (OTP/email verification) before access is granted.

This system is developed on Flask, a lightweight Python framework, and SQLite for secure user data storage. The image operations are done with the Pillow library, while base-64 and binary encoding techniques are in place for embedding and extracting secret messages. For MFA, email-based OTP verification or simulated 2-step verification is used to ensure legitimate users access decoding functionality only.

This integration of steganography with multi-factor authentication helps to enhance data security as well as prevent the misuse of sensitive information. The real-world relevance of this project involves secure communication, defense, digital watermarking, and preservation of privacy. The future work can be done on using biometric verification along with the use of end-to-end encryption for more secure protection.

Literature Survey Summary

1. SMFA: Strengthening Multi-Factor Authentication with Steganography for Enhanced Security (Sarower et al., IEEE, 2025)

   Here, a new authentication framework named SMFA (Steganography-based Multi-Factor Authentication) is proposed, which extends the traditional multi-factor systems to embed the authentication tokens (like OTPs) inside images by using steganography along with cryptographic protection. Fundamentally, the user conducts a basic login with credentials and then receives an image that contains a secondary factor encrypted within it. The user performs the extraction and decryption of the hidden factor to complete the authentication. Authors prove that SMFA effectively mitigates the common attack vectors such as interception, phishing, and replay attacks because of the hiding of the second authentication factor in an image. Performance evaluations show very minimal overhead for embedding/extracting data, making the approach feasible in practical deployments.

2. Combining Steganography and Biometric Cryptosystems for Secure Mutual Authentication and Key Exchange (R. Rashid et al., 2013)

   The paper "Combining Steganography and Biometric Cryptosystems for Secure Mutual Authentication and Key Exchange" by Rasber Rashid et al. (2013) presents a security model that incorporates biometric authentication with steganography for enhanced data security in communication systems. This approach either generates or binds cryptographic keys using the user's biometric data, including fingerprints or iris patterns. These are then hidden inside digital images using steganography techniques before their transmission. In this way, sensitive authentication information with keys will not be revealed to an attacker even when the communication channel is compromised. At the receiver's end, a procedure of extracting the hidden data is carried out, while mutual authentication of both the parties is assured. The hybrid technique proposed showed a better level of confidentiality and integrity of authentication while reducing all possible risks of key theft or exposure of the biometric template. To summarize, this paper takes a look at how merging biometric data with steganography can yield secure and covert key exchange and verification of users.

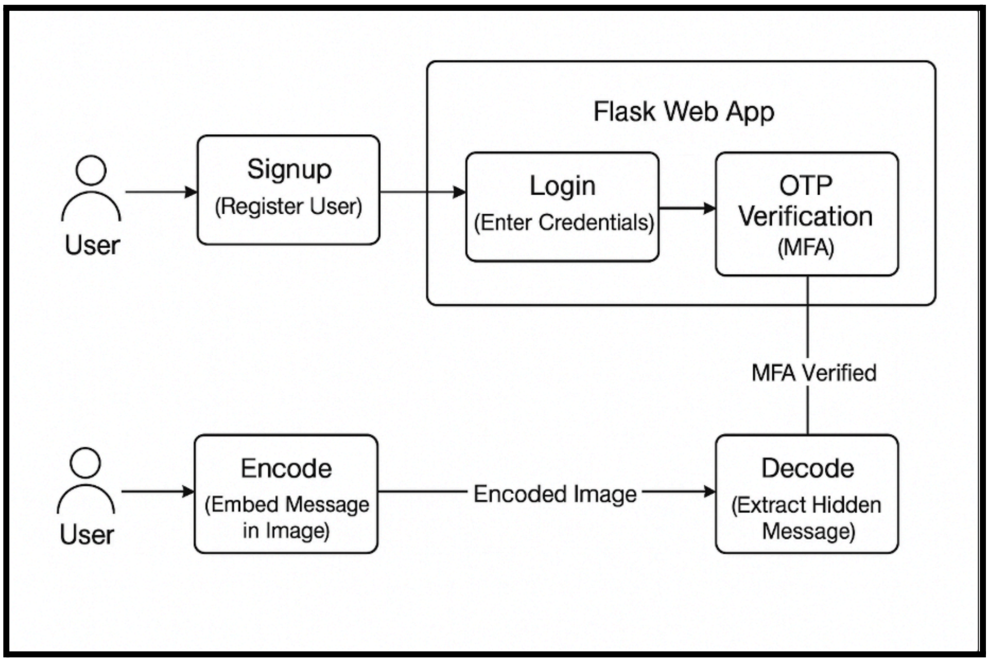
3. Kour, J., & Verma, D. (2014). Steganography techniques–A review paper. *International Journal of Emerging Research in Management &Technology ISSN*, 2278-9359.

   The paper reviews various steganography techniques for secure and hidden communications. To begin with, it defines that while cryptography encrypts the data and makes it unreadable, it is nonetheless a source of curiosity; however, steganography conceals the very existence of the data by embedding it into cover media, which can be in the form of images, audio, or video. Further, the authors go on to discuss various types of steganography, which includes text, image, audio, video, and network-based, each of which applies unique embedding strategies. Various techniques such as LSB, Pixel Value Differencing, DCT, and DWT are reviewed together with their different strengths and weaknesses. Other meaningful factors which ensure steganographic efficiency and are discussed here include robustness, imperceptibility, payload capacity, and PSNR. At the end, the authors have concluded that a combination of steganography and cryptography enhances data confidentiality and that further research should focus on enhancement of robustness against attacks with high image quality and capacity.

4. Privacy Preserving Multi-Factor Authentication with Biometrics by Bhargav-Spantzel, Squicciarini, and Bertino.

   It proposes a privacy-preserving multi-factor authentication system that will use biometrics-like fingerprints or face data in combination with other factors like passwords or smart cards for authentication. In fact, the prime objective is to make authentication safer and more private, especially in online or federated identity systems where users log in to numerous services through one identity.Instead of transferring biometric data or templates to any server, which can lead to privacy leaks, the transformed system makes use of special mathematical techniques to convert those biometric features to cryptographic keys and verifies them through zero-knowledge proofs, whereby users can prove their identity without revealing their passwords or biometric data.

4 Architecture Diagram



5. Implementation

1. User registration (signup)

- The user enters a username and a password.
- The system stores the password and stores the credentials in an SQLite database.
- No MFA is required at this stage.

2. Encoding (No Authentication Required)

- The user selects an image and types a secret message.
- The system uses Least Significant Bit (LSB) steganography to embed the message.
- The encoded image is displayed and is saved in the directory.
- No login or MFA is required here; this is an open access feature.

3. Login with MFA

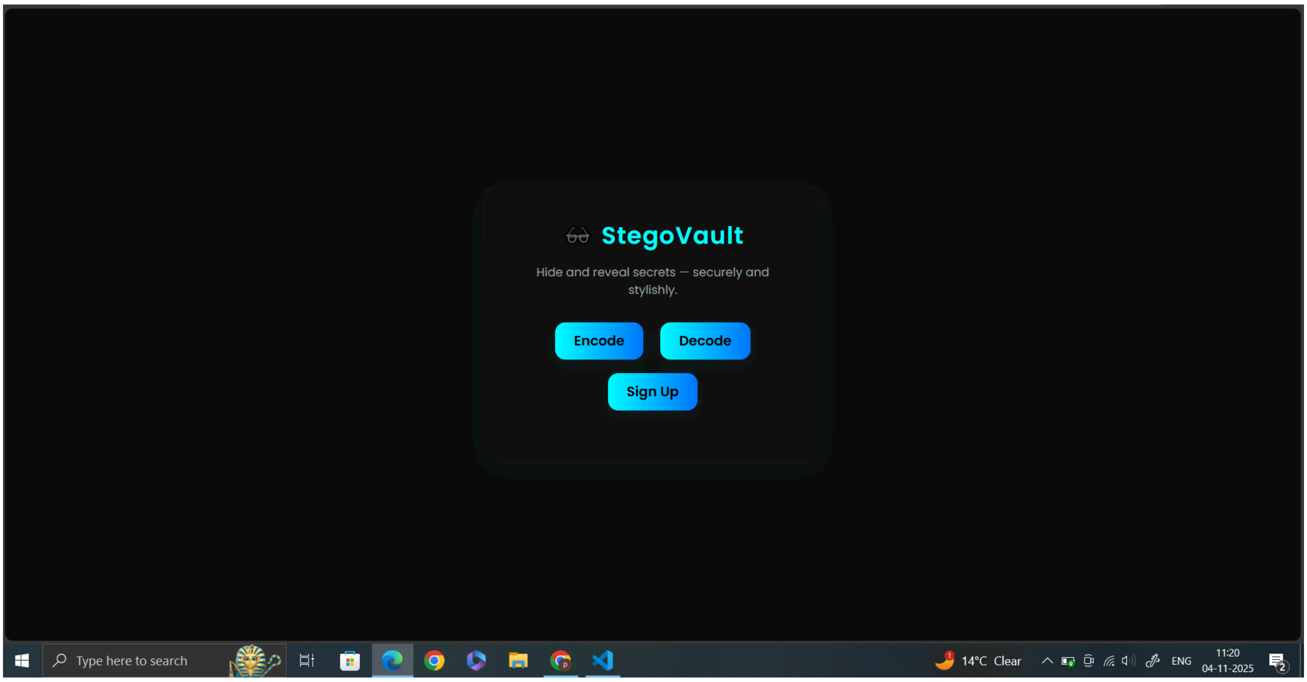
- Users must log in to decode.
- The system checks the credentials from the SQLite database.
- Once the login succeeds, a One-Time Password (OTP) is generated:
- - Sent to the registered email, or
- - Displayed on-screen for demonstration.
- The OTP is stored temporarily, for example, in a session or in the database with an expiry.
- The user must enter the OTP correctly to complete the MFA.

4. Decoding (MFA Protected)

- Before decoding, the backend checks:
- Is the user logged in?
- Has MFA (OTP verification) succeeded?
- If both are valid:
- The user uploads the encoded image.
- The LSB algorithm extracts the hidden message.
- The message will appear securely on-screen.
- If the MFA is invalid or expired, the user is redirected to the Login/OTP page.

6. Results

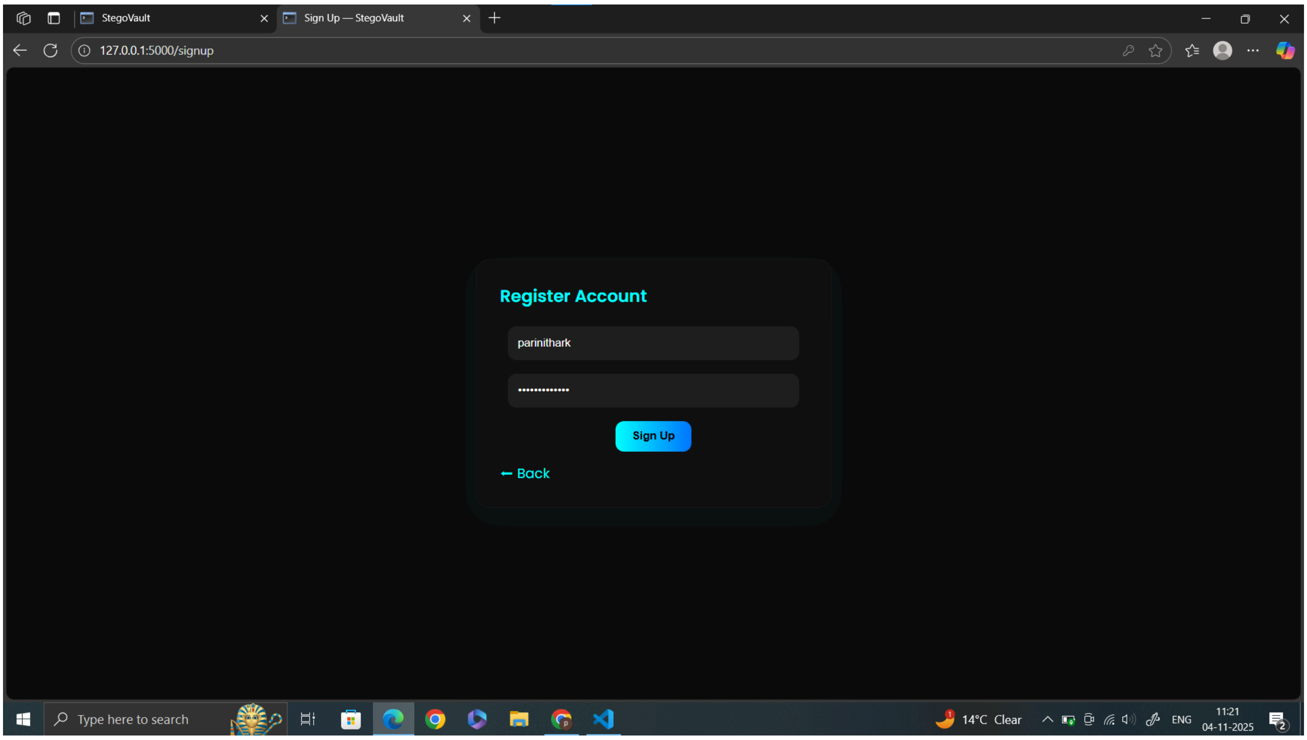
1. Home page :



StegoVault

Hide and reveal secrets — securely and stylishly.

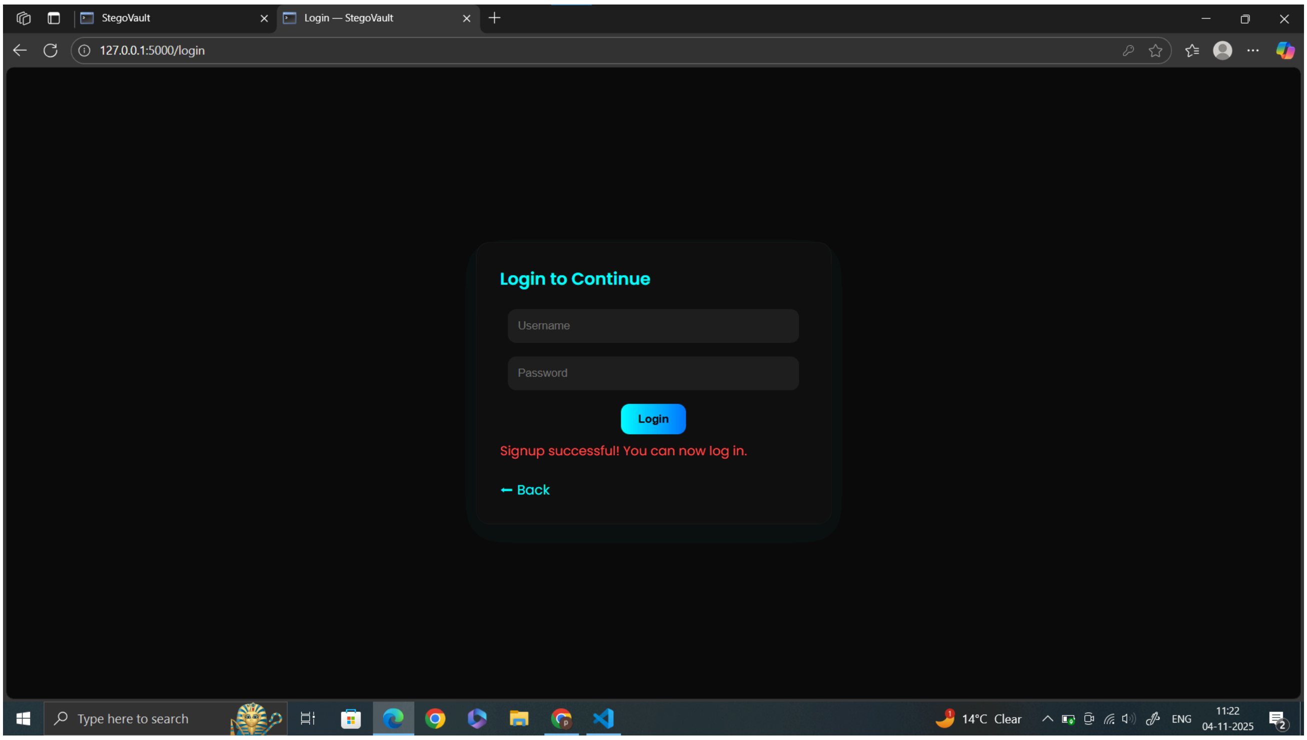
2. Signup page:



Register Account

parinithark

•••••••••••••



Login to Continue

Username

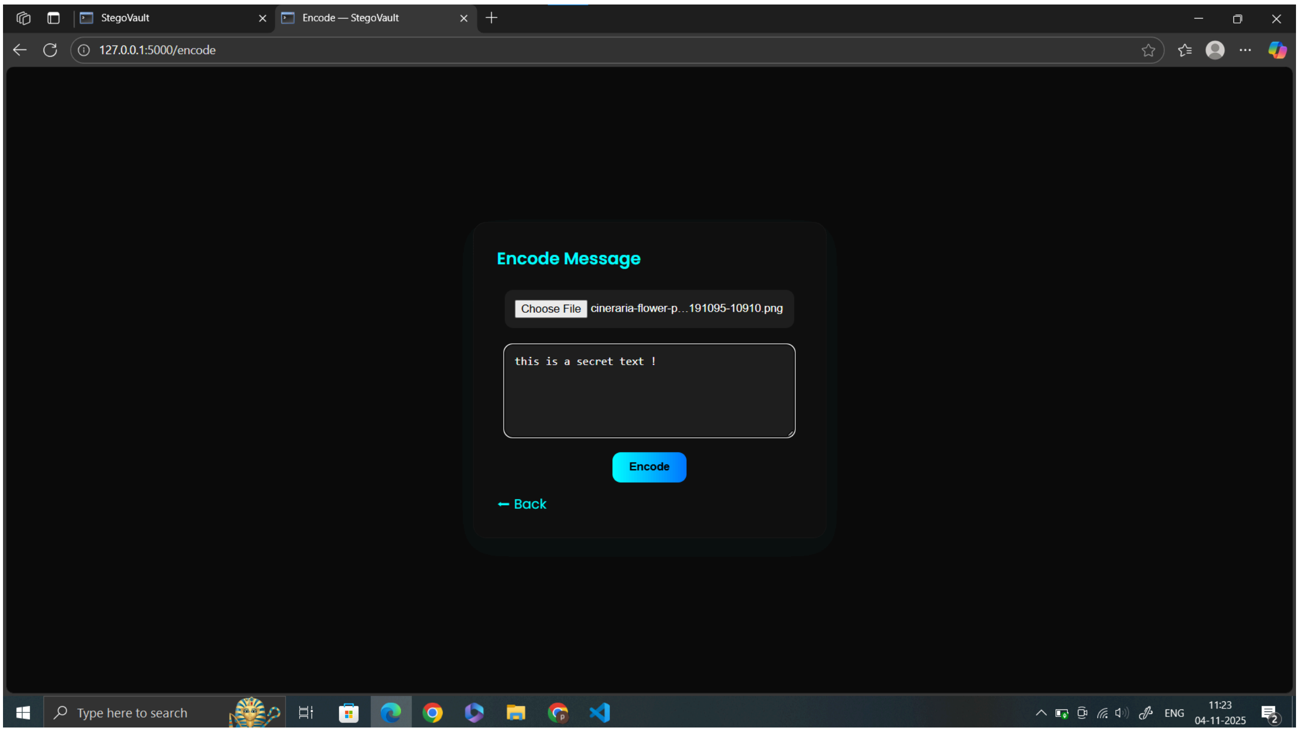
Password

Login

Signup successful! You can now log in.

← Back

3. hide text in an image:



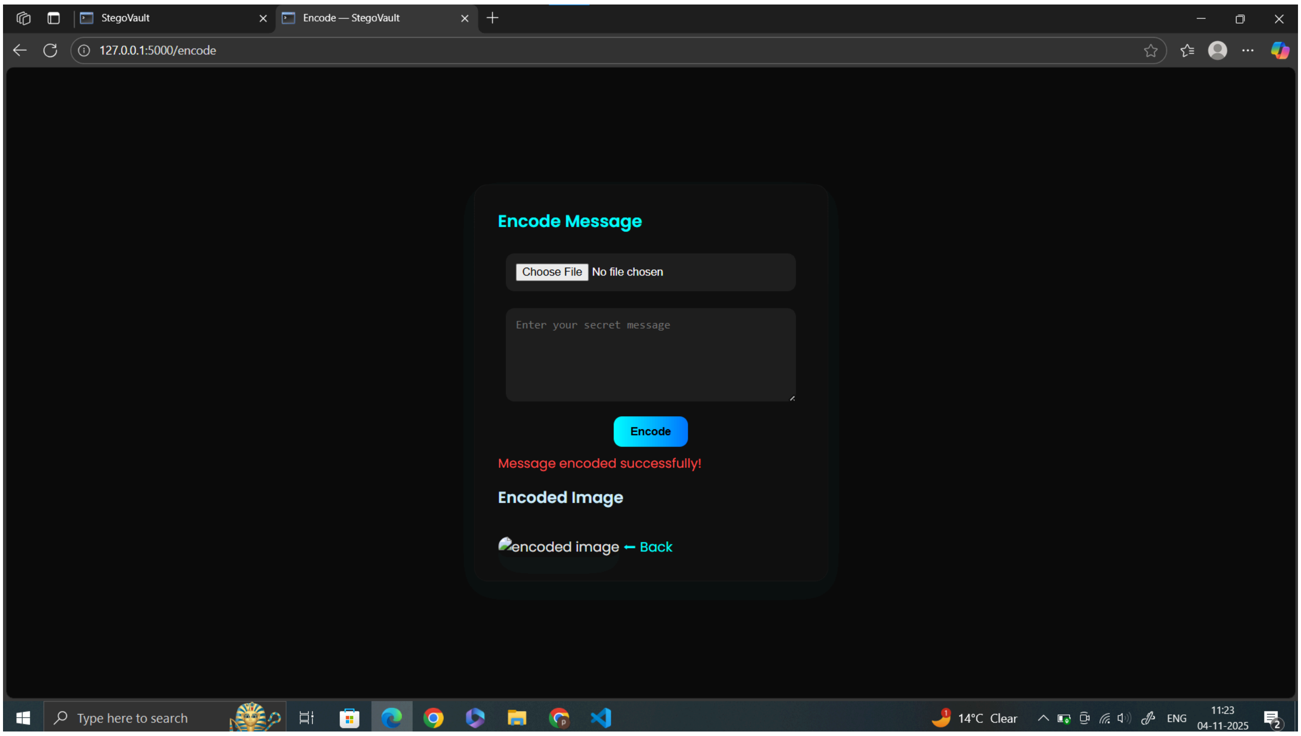
Encode Message

Choose File cineraria-flower-p...191095-10910.png

```
this is a secret text !
```

Encode

← Back

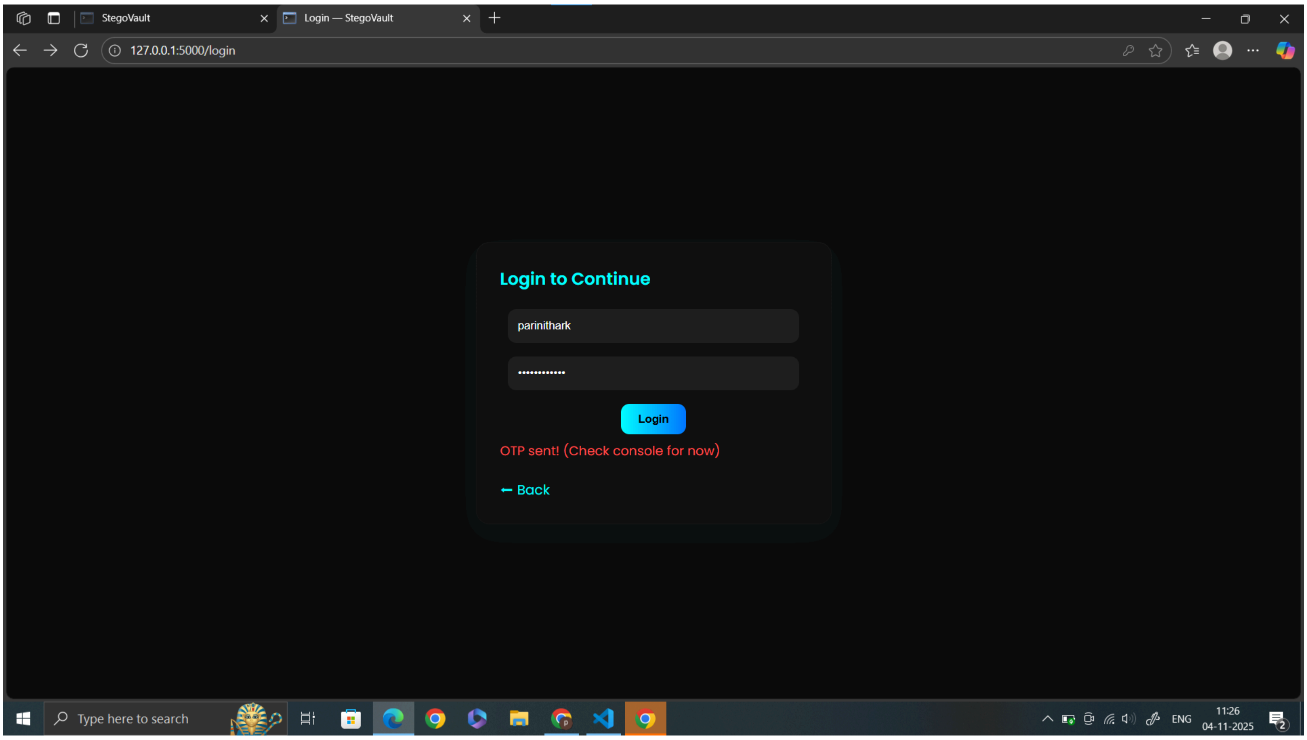


Encode Message

Message encoded successfully!

Encoded Image

4. authenticate for decoding:

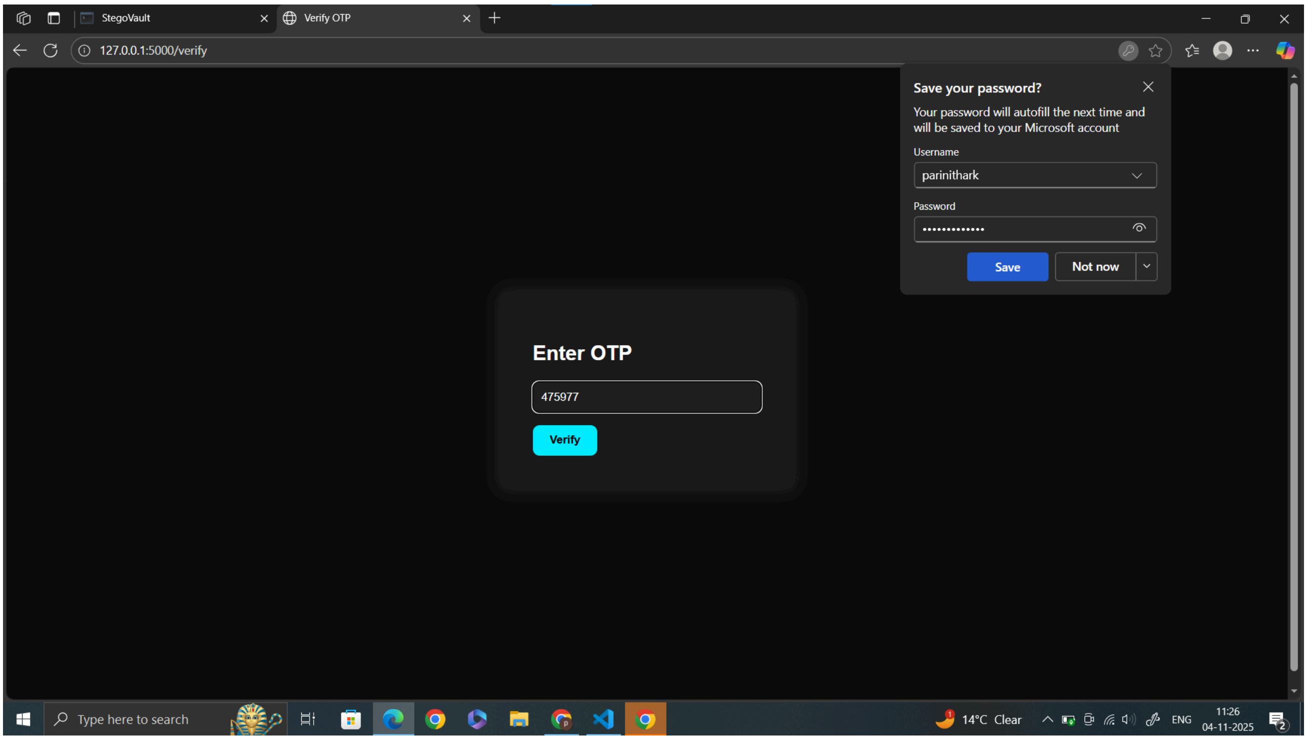


Login to Continue

parinithark

••••••••••••

OTP sent! (Check console for now)



Save your password?

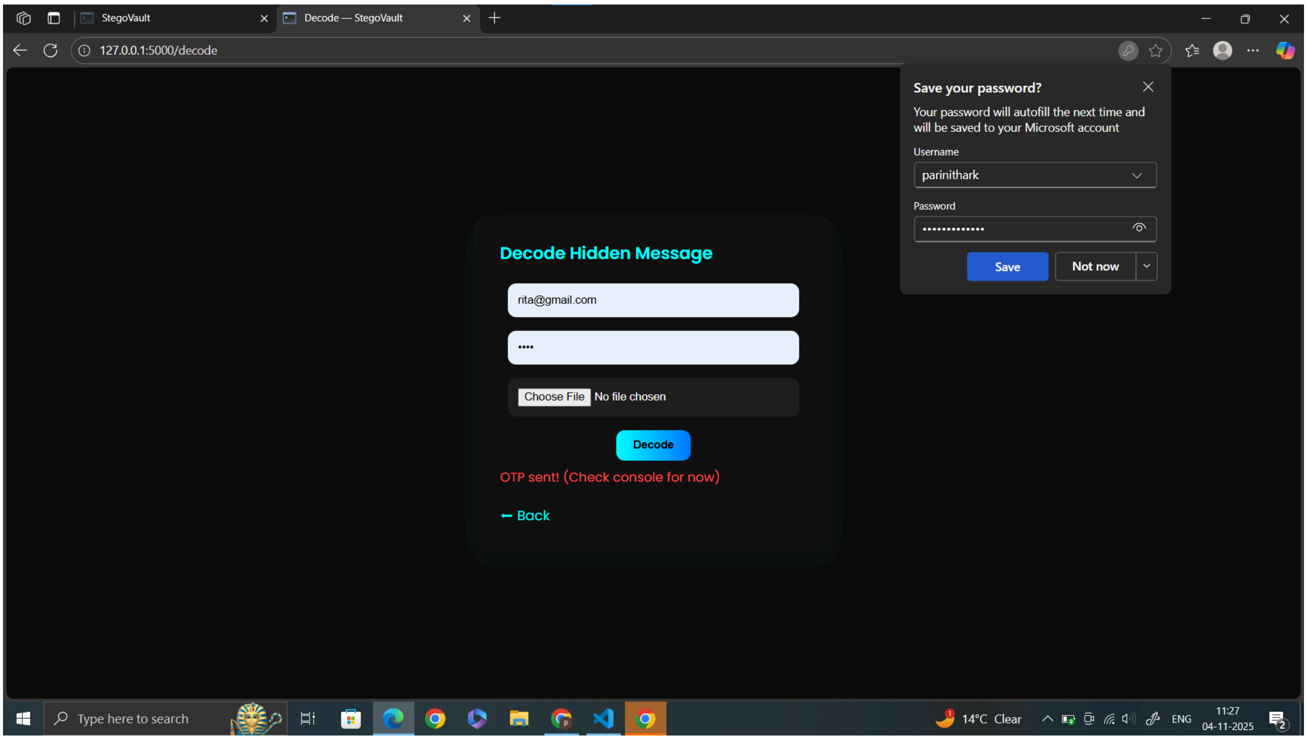
Your password will autofill the next time and will be saved to your Microsoft account

Username: parinithark

Password: ••••••••••••••

Enter OTP

475977



Save your password?

Your password will autofill the next time and will be saved to your Microsoft account

Username: parinithark

Password: ••••••••••••••

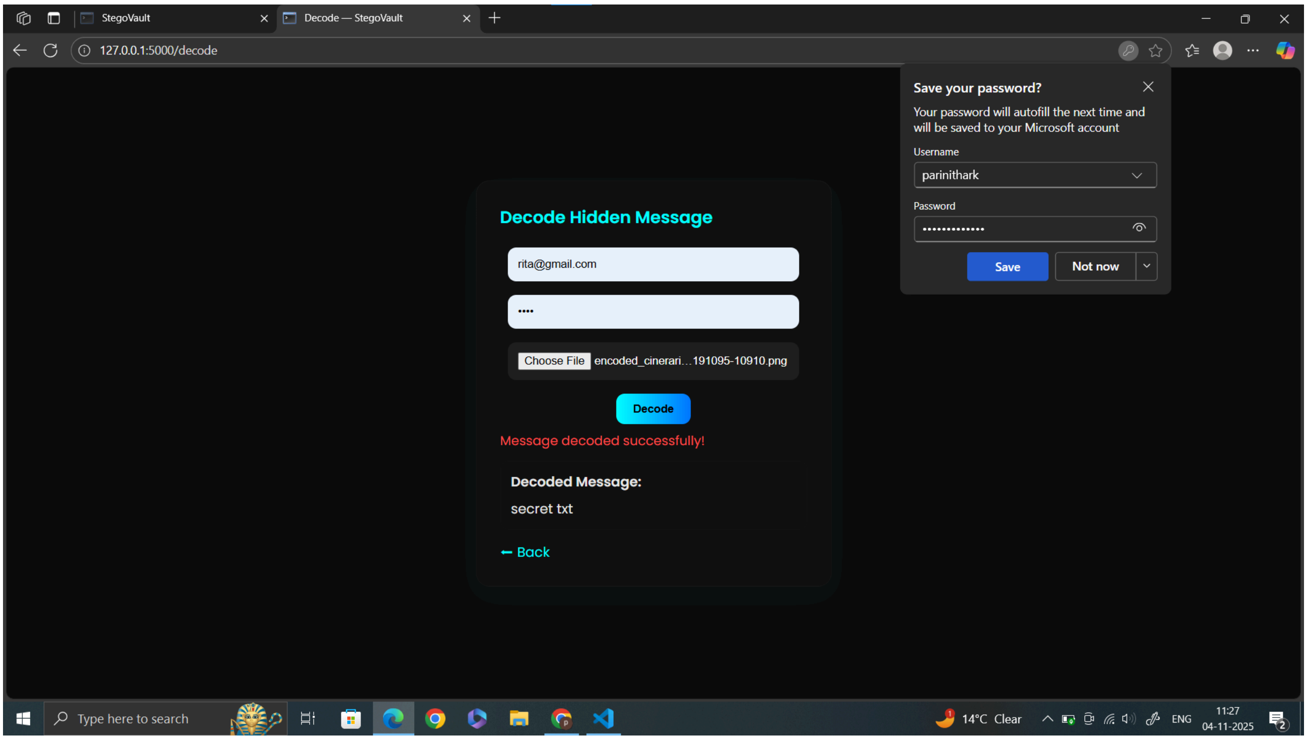
Decode Hidden Message

rita@gmail.com

••••

Choose File No file chosen

OTP sent! (Check console for now)



Decode Hidden Message

rita@gmail.com

••••

Choose File encoded_cinerari…191095-10910.png

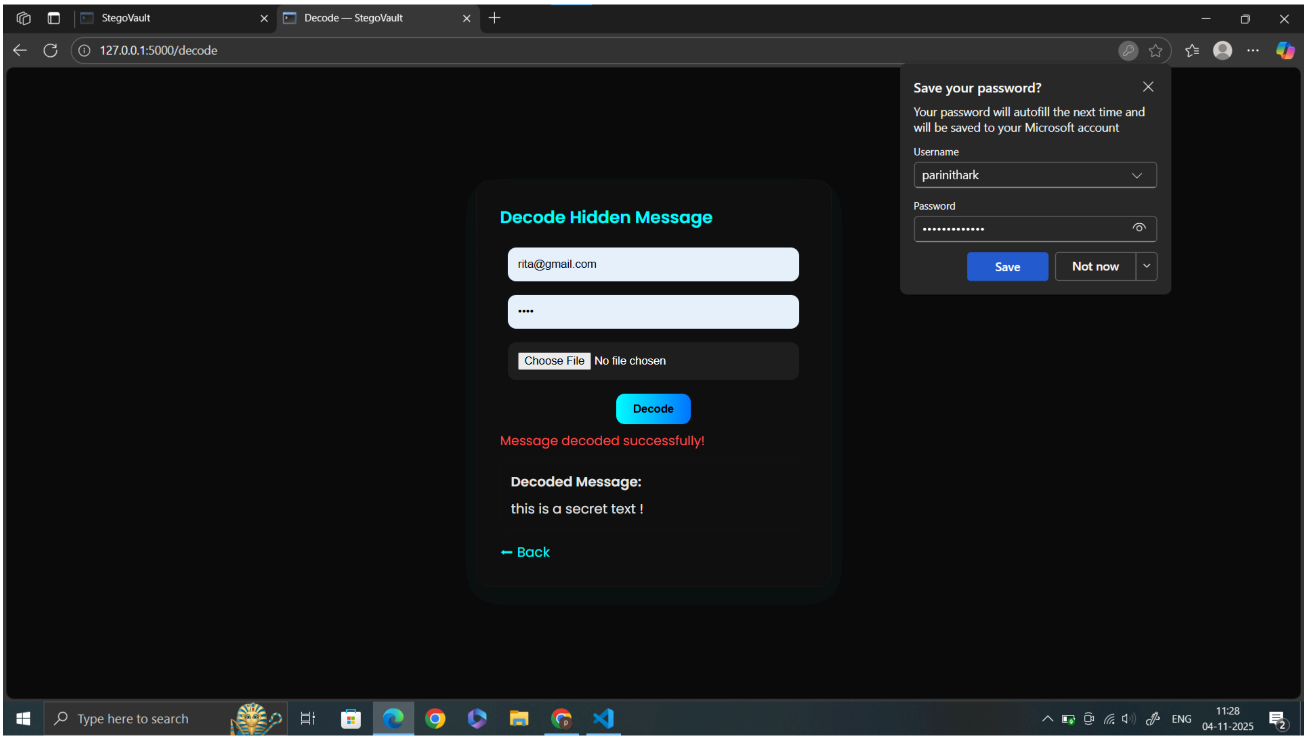
Decode

Message decoded successfully!

Decoded Message:

secret txt

← Back



Decode Hidden Message

rita@gmail.com

••••

Choose File No file chosen

Decode

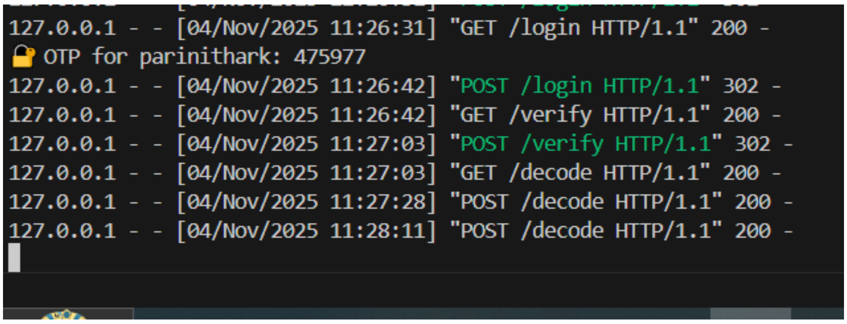
Message decoded successfully!

Decoded Message:

this is a secret text !

← Back

5. OTP is received in the terminal



```
127.0.0.1 - - [04/Nov/2025 11:26:31] "GET /login HTTP/1.1" 200 -
🔐 OTP for parinithark: 475977
127.0.0.1 - - [04/Nov/2025 11:26:42] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [04/Nov/2025 11:26:42] "GET /verify HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2025 11:27:03] "POST /verify HTTP/1.1" 302 -
127.0.0.1 - - [04/Nov/2025 11:27:03] "GET /decode HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2025 11:27:28] "POST /decode HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2025 11:28:11] "POST /decode HTTP/1.1" 200 -
```

Security Features

- Authentication + MFA: Allow decoding only after verification.
- Session-based access: only valid sessions can reach the decoding route.
- OTP Expiration: OTP shall be valid for one-time use only.
- Secure Database: Credentials are kept safe in SQLite.

7. Conclusion

This project focuses on the development of a secure image steganography web application, where users can encode a secret message in an image with the LSB technique. Ensuring privacy and security, it authenticates users with MFA at the time of decoding, allowing only authorized users to retrieve the hidden data. The web application has been developed using Flask, SQLite, and PIL and provides an easy interface for the users to encode and decode images intuitively while enforcing powerful security controls. This is the best description of how cryptography and steganography can be combined toward the protection of data confidentiality. The current design will be enhanced in the near future by adding end-to-end encryption, cloud storage, and advanced biometric authentication in order to enhance the usability and security of this system further.

8. References :

[1]. Sarower, A. H., Bhuiyan, T., Hasan, M. M., Arefin, M. S., & Hossain, G. (2025). SMFA: Strengthening Multi-Factor Authentication With Steganography for Enhanced Security. *IEEE Access*.

[2]. Al-Assam, H., Rashid, R., & Jassim, S. (2013, December). Combining steganography and biometric cryptosystems for secure mutual authentication and key exchange. In the 8th *International Conference for Internet Technology and Secured Transactions (ICITST-2013)* (pp. 369-374). IEEE.

[3] Kour, J., & Verma, D. (2014). Steganography techniques–A review paper. *International Journal of Emerging Research in Management &Technology ISSN*, 2278-9359.

[4] Al-Assam, H., Rashid, R., & Jassim, S. (2013, December). Combining steganography and biometric cryptosystems for secure mutual authentication and key exchange. In *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)* (pp. 369-374). IEEE.