بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম

الجامعة الإسلامية العالمية شيتغونغ

International Islamic University Chittagong

ARTIFICIAL INTELLIGENT LAB PROJECT REPORT

TOPIC: Hide & Seek (An Image Based Steganography Website)

Course Title: Artificial Intelligence Lab **Course Code:** CSE-3636

Submission Date: 09.07.2025

Submitted by:

Akhi Moon Jahan

C223202

6AF

Dept. of CSE, IIUC

Fayeza Afrah Hissan

C223206

6AF

Dept. of CSE, IIUC

Submitted to:

Sara Karim,

Adjunct Lecturer,

Dept of CSE, IIUC.

Signature:

Remarks:

Table of Contents

Introduction

In the digital age, securing information transmission has become increasingly important, not just in government or corporate domains, but also for everyday privacy, communication, and hobbyist data protection. One such method is **steganography** — the art of hiding information within other non-suspicious data like images, audio, or videos. Unlike cryptography, which disguises the meaning of a message, steganography disguises its very existence.

There are several types of steganography:

- **Text Steganography** – hiding data within documents or formatted text
- **Image Steganography** – embedding messages into image pixels
- **Audio Steganography** – hiding messages in sound signals
- **Video Steganography** – encoding secrets in video frames

Our project, **"Hide & Seek"**, is a web-based **image steganography** system built with Python (Django)**,** focused on ensuring not just secure data hiding but also ethical message filtering through an integrated **AI-powered message validation engine**. The application uses the **Least Significant Bit (LSB)** technique for encoding and decoding messages within images and incorporates a **Constraint Satisfaction Problem (CSP)**-based mechanism to validate each message before it can be embedded.

Software Requirements

Code Editor:

- Visual Studio Code: It is an integrated development environment (IDE) commonly used for web development. We used it to write and edit our HTML, CSS, and Python code.

Complier/Interpreter:

- PowerShell: It is a command-line shell and scripting language provided by Microsoft. We used it to run our website and execute commands.

Languages:

- HTML: It is a markup language used for creating the structure and content of web pages.
- CSS: It is a stylesheet language used for styling the appearance of web pages.
- Python: It is a versatile programming language used for various purposes, including web development. We used the Python Django framework, which is a high-level Python web framework, to build our steganography website.
- JavaScript: It is a scripting language used to add interactivity to the website, such as form validation, dynamic alerts, and responsive UI behaviors.

Hardware Requirements

➢ Information of Processing System Used for This Application:

- 12th Gen Intel(R) Core (TM) i3-1215U 1.20 GHz 8GB RAM
- Windows 11 Home Single Language

➢ Minimum Hardware Requirement:

- Pentium 3 166 MHZ Or Higher 128 mb RAM

---or---

- A modern processor from a recent generation, such as the 10th,11th, or 12th generation Intel Core processors.

Methodology

The "Hide & Seek" system is built on two main mechanisms:

1. **Image Steganography using Least Significant Bit (LSB) substitution**, and
2. **AI-integrated Message Validation using Constraint Satisfaction Problem (CSP) logic**.

These components work together to ensure that secret messages are not only securely embedded within images but also ethically filtered and accessible only to intended recipients.

1. LSB (Least Significant Bit) Steganography

The **LSB substitution method** is one of the simplest and most widely used techniques in image steganography. In a **24-bit RGB image**, each pixel consists of three color channels — Red, Green, and Blue — each stored in 8 bits, totaling **24 bits per pixel**.

For example:

- A red pixel value may be `10001011` (139 in decimal).
- The **rightmost bit** is the **Least Significant Bit (LSB)**.
- Changing `10001011` to `10001010` has negligible visual impact but encodes a bit of hidden data.

How It Works:

1.During Encoding:

- The **secret message and password** are first converted to **binary**.
- A **fixed-size header** is created at the start of the image data, which includes:
  - **Message Length (in bits)** – so the decoder knows how many bits to extract.
  - **Password** – to validate authorized access before revealing the message.
- The message and password bits are then embedded into the **LSBs of image pixels**, one bit per channel.
- Because the **Human Visual System (HVS)** is not sensitive to such minor changes, the modified image appears the same.

2.During Decoding:

- The system first reads the **message length** and the **embedded password** from the LSBs.

- If the **user-entered password** matches the embedded one, the system proceeds to extract the message bits using the length header.

Advantages:

- Simple, fast, and easy to implement.
- High data embedding capacity.
- Little to no visible distortion in the image.

Limitations:

- Vulnerable to image compression, cropping, or noise.
- Message size must be smaller than the image's pixel capacity.
- Password protection is lightweight (not encrypted), so it should be used for basic validation only.

2. CSP-Based Message Validation (AI Integration)

Before any message is embedded, *Hide & Seek* performs **message validation** using a **Constraint Satisfaction Problem (CSP)** approach. This ensures that messages are appropriate, clean, and formatted for ethical communication.

CSP Basics:

A **Constraint Satisfaction Problem** involves:

- **Variables**: Each part of the message (e.g., word, character, length).
- **Domains**: Possible valid values for each variable (e.g., dictionary words, ASCII symbols).
- **Constraints**: Rules that restrict the variable combinations.

The system checks each message against a predefined set of constraints. If any constraint fails, encoding is blocked, and the user receives an error message.

Implemented Constraints:

Constraint Type	**Purpose**
Length Check	Message must fit within image capacity.
Profanity/Hate Speech/Banned Words	Uses `better_profanity` and some custom banned words and hate phrases to block inappropriate content.
Dictionary Check	Validates words using `nltk.corpus.words` to ensure meaningful messages.
Emoji/Unicode Filter	Blocks emojis and non-ASCII symbols via the `emoji` library.
Capitalization Rule	Rejects ALL CAPS messages to discourage spam/shouting.
Binary Pattern Block	Prevents messages resembling raw binary data.
Punctuation Check	Flags missing punctuation that may indicate suspicious content.

This **CSP validator acts like a smart filter**, improving both security and usability by allowing only **well-formed**, **ethical**, and **human-readable** messages to be hidden.

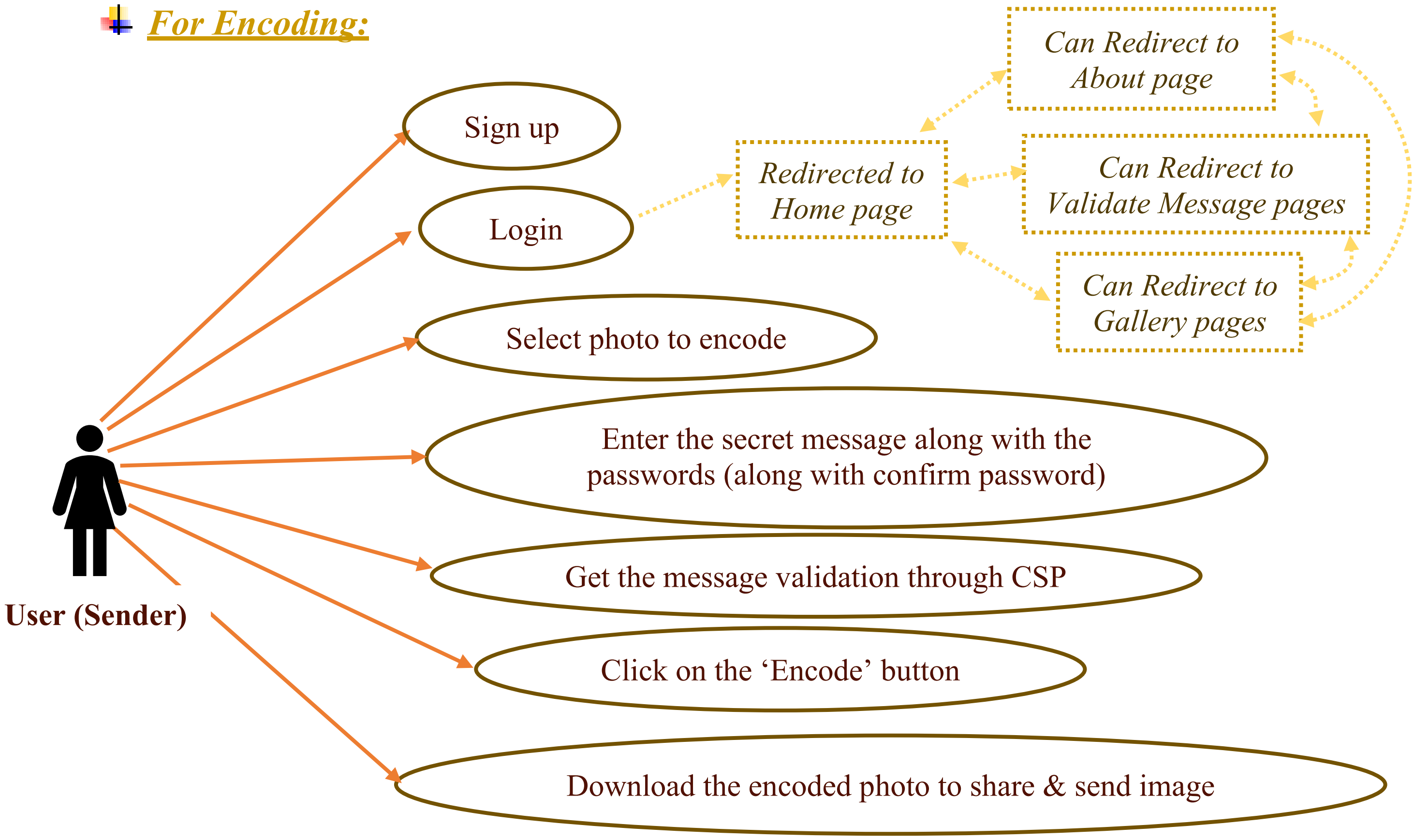
Integrated Workflow:

1. **User uploads an image**, writes a secret message, and provides a password.
2. The message is passed to the **CSP-based validator**.
3. If all constraints are satisfied:
   - The system calculates **message length**.
   - Converts the **message and password to binary**.
   - **Embeds** them using the **LSB substitution method**.
4. The result is an encoded **stego-image**.
5. During decoding:
   - The system extracts the **message length and password**.
   - If the password matches, it retrieves and reconstructs the **hidden message**.

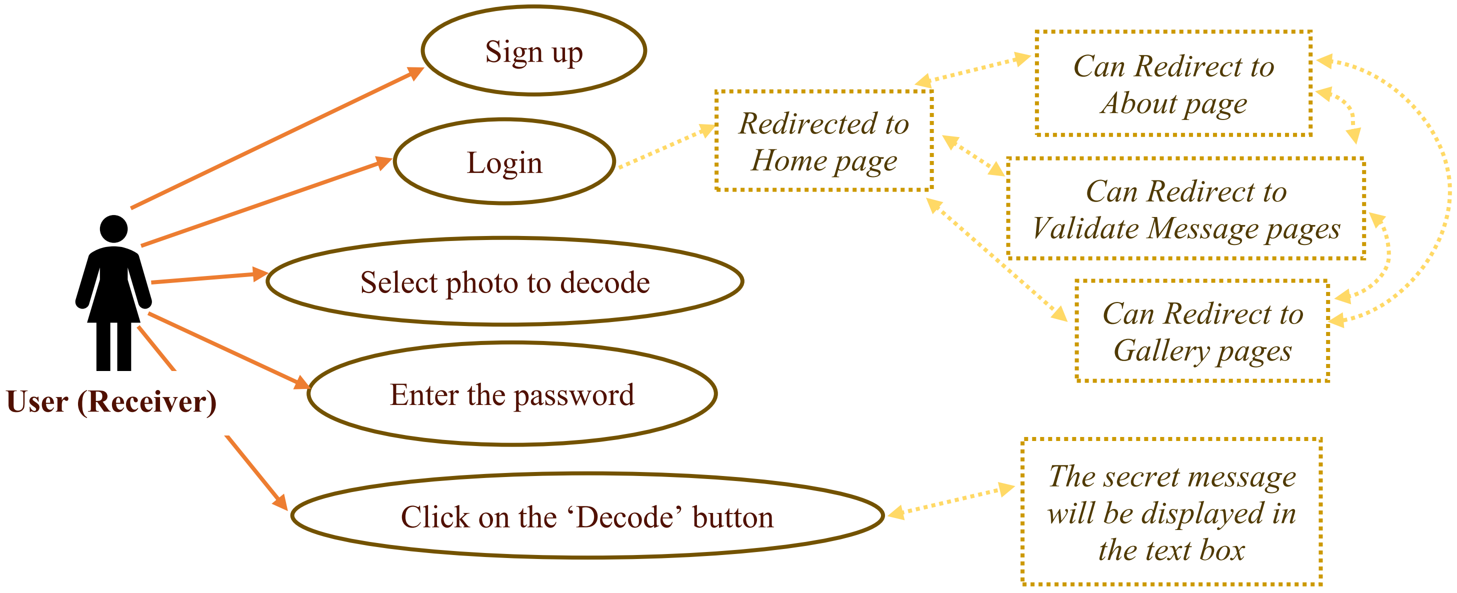
This combination of **LSB steganography** for secure embedding and **CSP-based AI filtering** for ethical validation makes *Hide & Seek* a complete, user-safe, and privacy-focused steganographic system.

User Diagram

For Encoding:

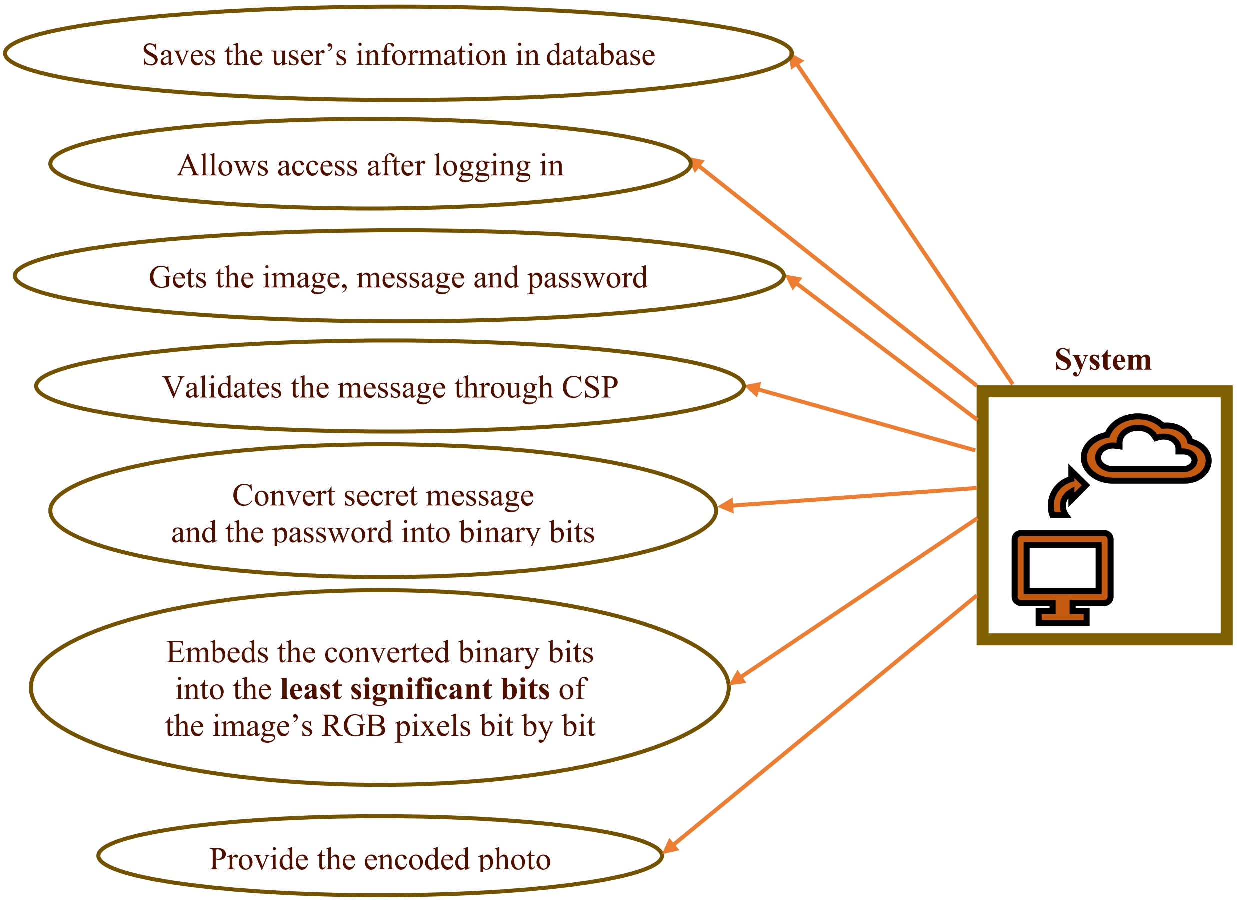

Can Redirect to About page
Sign up
Redirected to Home page
Can Redirect to Validate Message pages
Login
Can Redirect to Gallery pages
Select photo to encode
Enter the secret message along with the passwords (along with confirm password)
Get the message validation through CSP
User (Sender)
Click on the 'Encode' button
Download the encoded photo to share & send image

For Decoding:

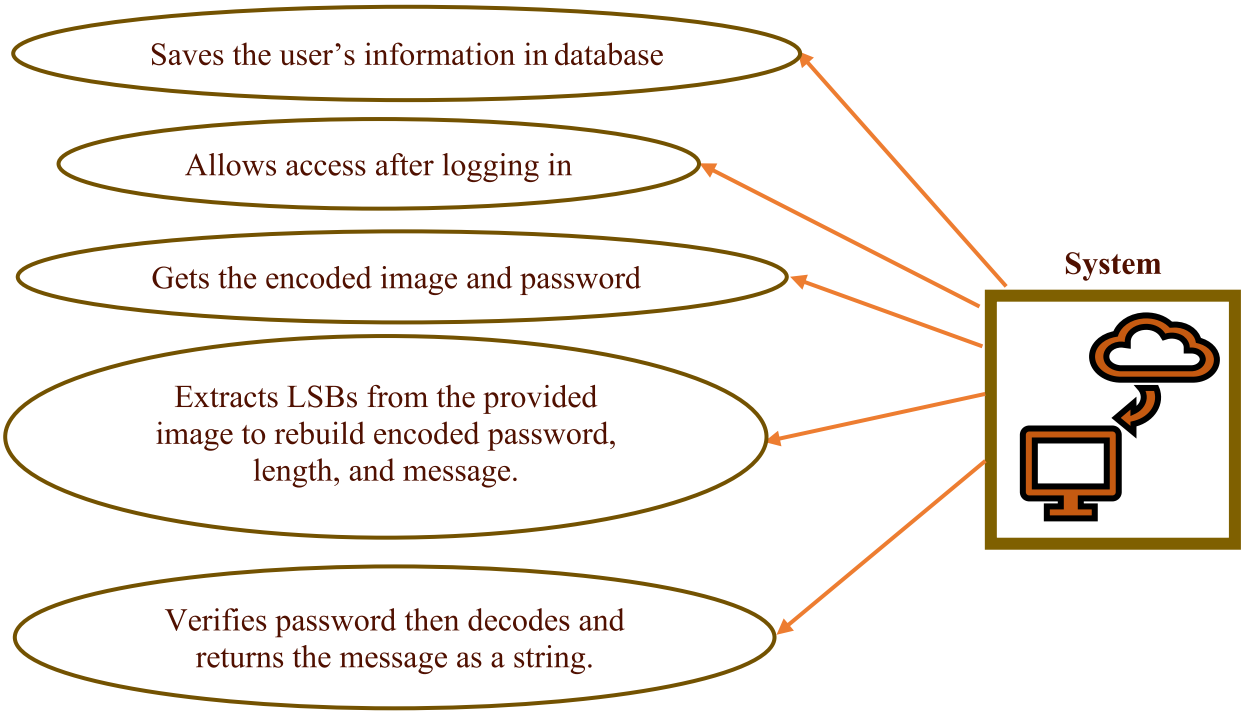

Sign up
Login
Select photo to decode
Enter the password
Click on the 'Decode' button
User (Receiver)
Redirected to Home page
Can Redirect to About page
Can Redirect to Validate Message pages
Can Redirect to Gallery pages
The secret message will be displayed in the text box

System Diagram

Encoding Process:


Saves the user's information in database
Allows access after logging in
Gets the image, message and password
Validates the message through CSP
Convert secret message and the password into binary bits
Embeds the converted binary bits into the **least significant bits** of the image's RGB pixels bit by bit
Provide the encoded photo
System

Decoding Process:



Saves the user’s information in database

Allows access after logging in

Gets the encoded image and password

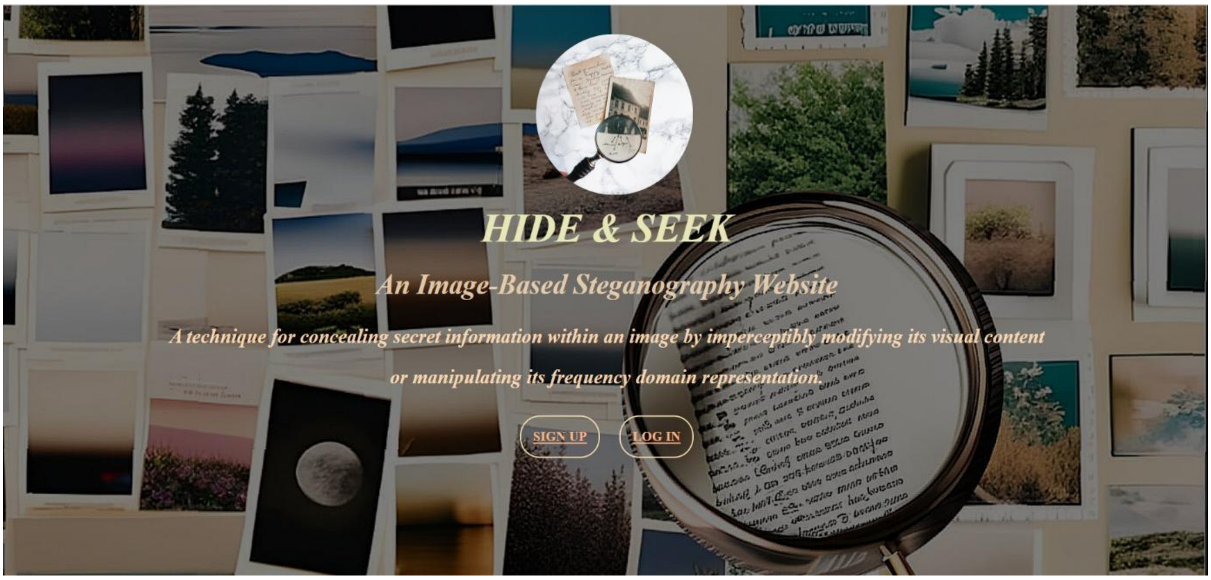
Extracts LSBs from the provided image to rebuild encoded password, length, and message.

Verifies password then decodes and returns the message as a string.

System

Website Demonstration

Front Page:

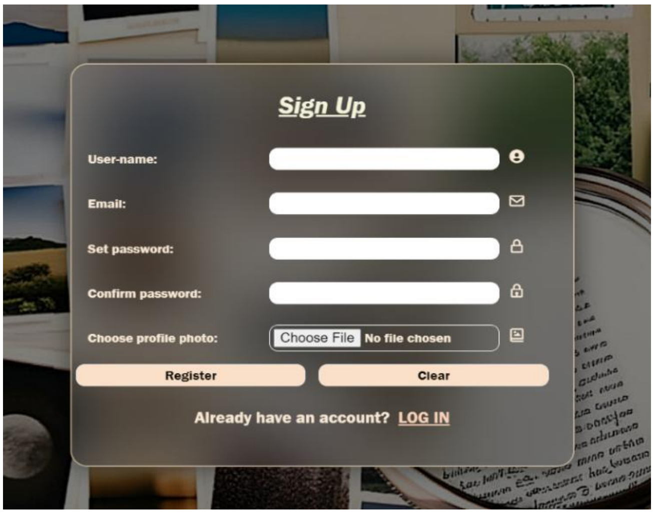


HIDE & SEEK

An Image-Based Steganography Website

A technique for concealing secret information within an image by imperceptibly modifying its visual content or manipulating its frequency domain representation.

Sign Up & Log In:



Sign Up

User-name:

Email:

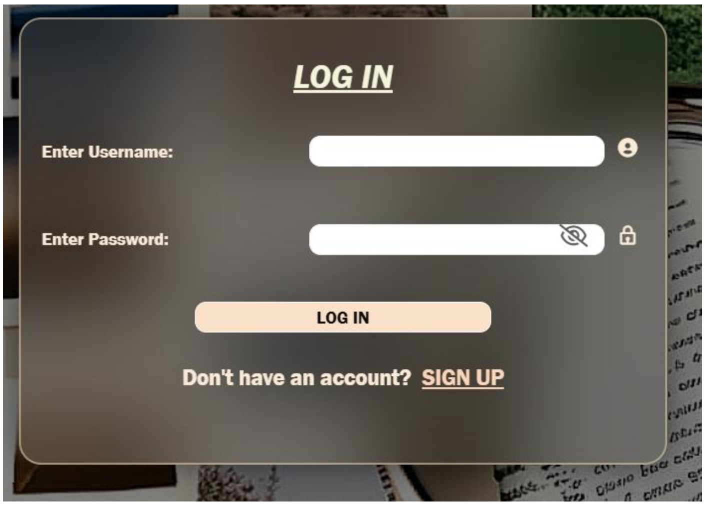
Set password:

Confirm password:

Choose profile photo: Choose File No file chosen

Register | Clear

Already have an account? LOG IN



LOG IN

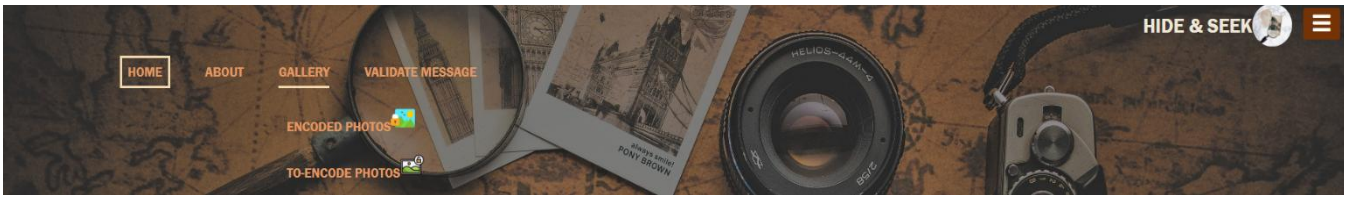
Enter Username:

Enter Password:

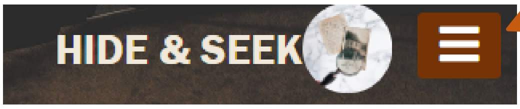
LOG IN

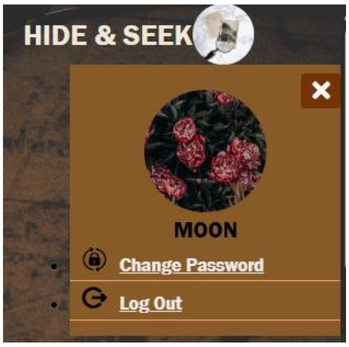
Don't have an account? SIGN UP

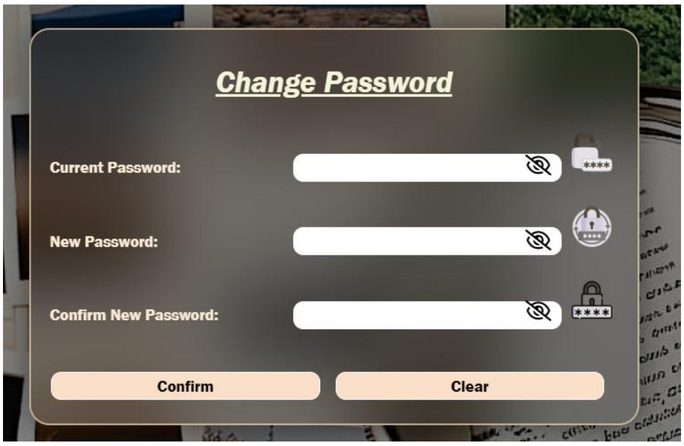
Navigation Bar:



Profile Display & Editing:







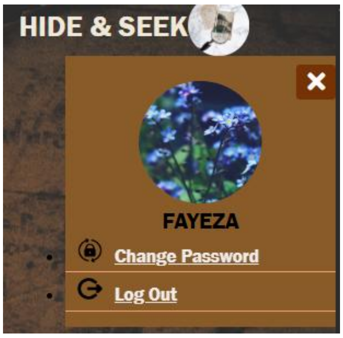
Change Password

Current Password:

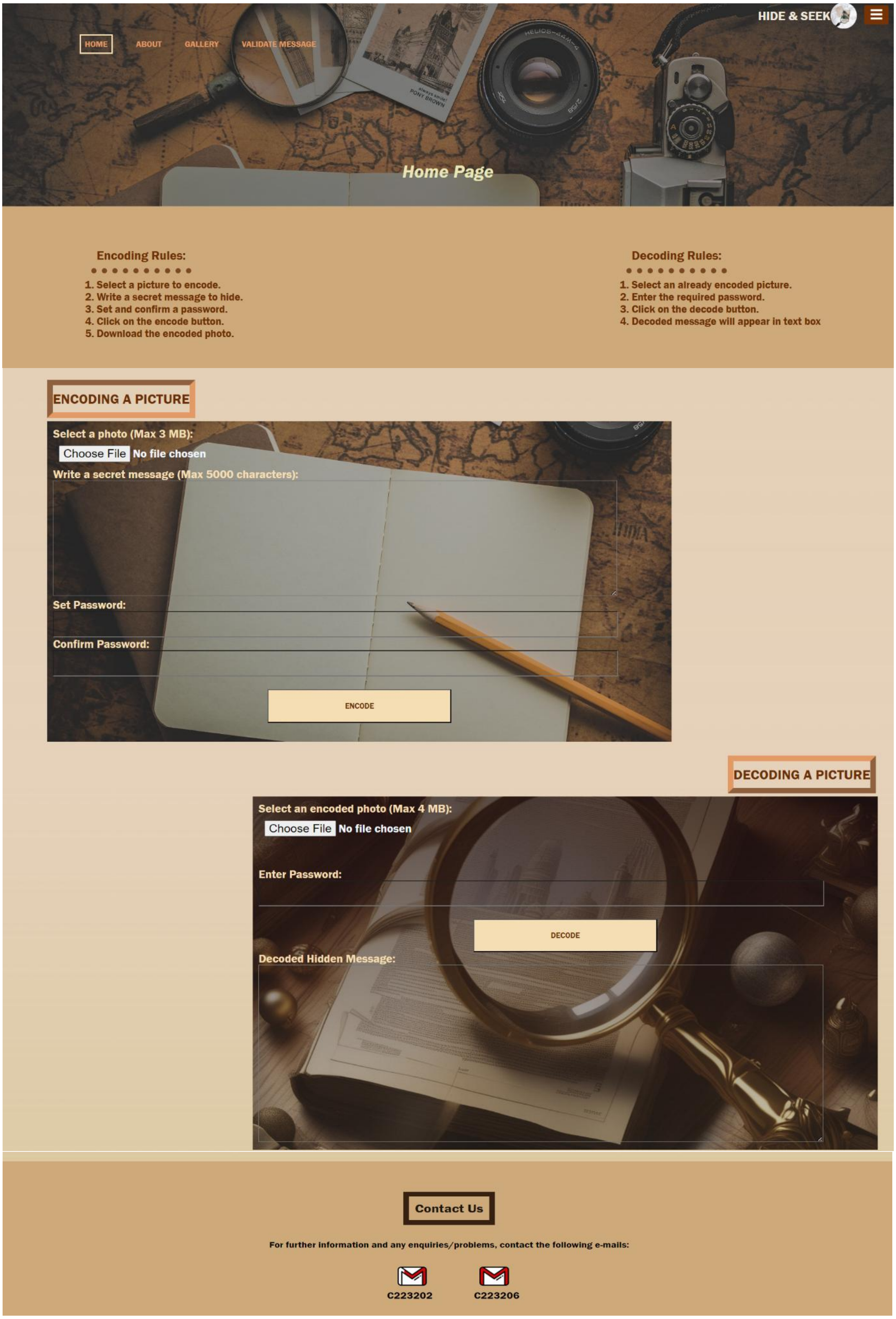
New Password:

Confirm New Password:

Confirm | Clear



Home Page:



Encoding Rules:

1. Select a picture to encode.
2. Write a secret message to hide.
3. Set and confirm a password.
4. Click on the encode button.
5. Download the encoded photo.

Decoding Rules:

1. Select an already encoded picture.
2. Enter the required password.
3. Click on the decode button.
4. Decoded message will appear in text box

ENCODING A PICTURE

Select a photo (Max 3 MB):

Choose File No file chosen

Write a secret message (Max 5000 characters):

Set Password:

Confirm Password:

DECODING A PICTURE

Select an encoded photo (Max 4 MB):

Choose File No file chosen

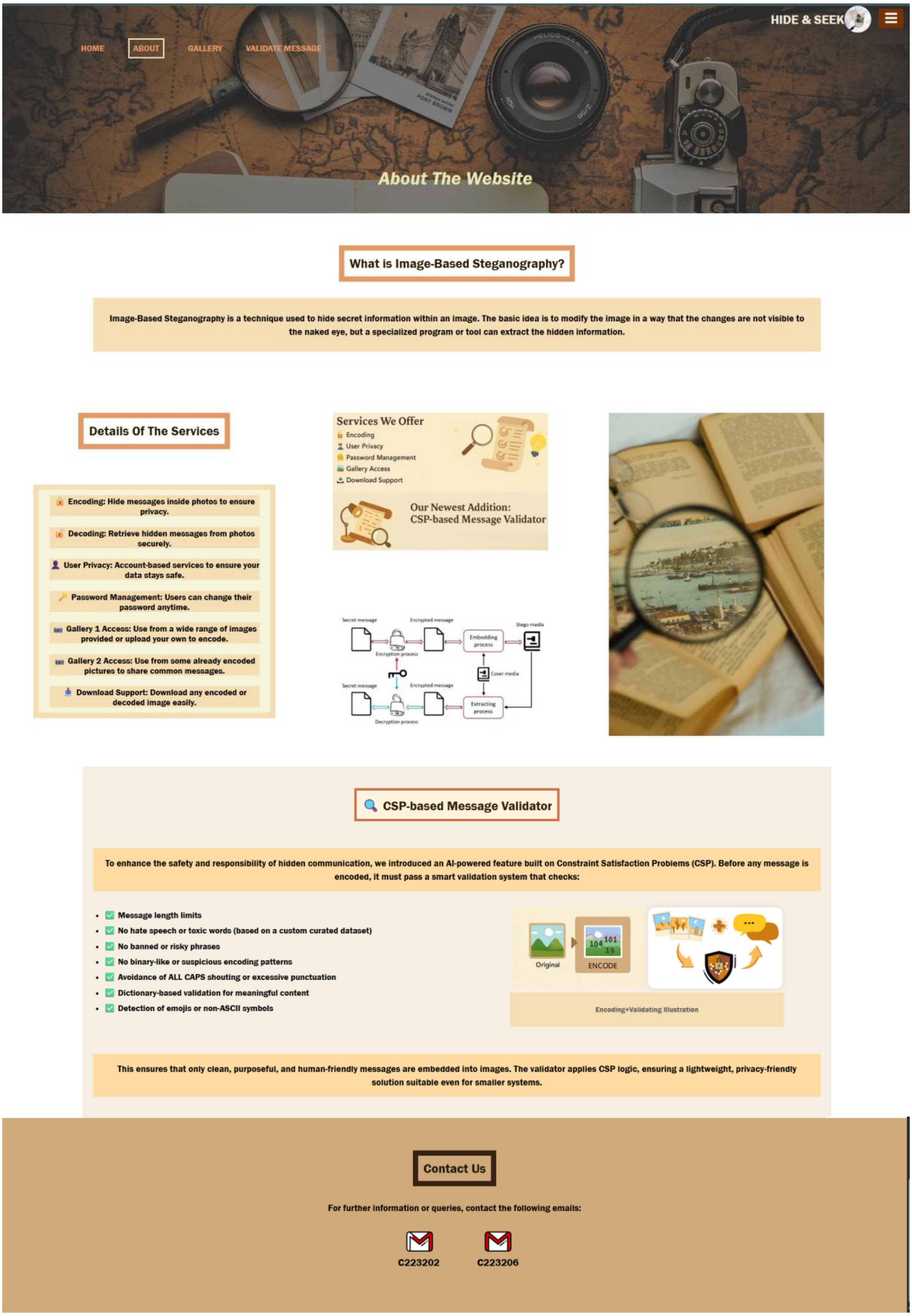
Enter Password:

Decoded Hidden Message:

Contact Us

For further information and any enquiries/problems, contact the following e-mails:

About Page:



About The Website

What is Image-Based Steganography?

Image-Based Steganography is a technique used to hide secret information within an image. The basic idea is to modify the image in a way that the changes are not visible to the naked eye, but a specialized program or tool can extract the hidden information.

Details Of The Services

- Encoding: Hide messages inside photos to ensure privacy.
- Decoding: Retrieve hidden messages from photos securely.
- User Privacy: Account-based services to ensure your data stays safe.
- Password Management: Users can change their password anytime.
- Gallery 1 Access: Use from a wide range of images provided or upload your own to encode.
- Gallery 2 Access: Use from some already encoded pictures to share common messages.
- Download Support: Download any encoded or decoded image easily.

Services We Offer

- Encoding
- User Privacy
- Password Management
- Gallery Access
- Download Support

Our Newest Addition:
CSP-based Message Validator

CSP-based Message Validator

To enhance the safety and responsibility of hidden communication, we introduced an AI-powered feature built on Constraint Satisfaction Problems (CSP). Before any message is encoded, it must pass a smart validation system that checks:

- ☑ Message length limits
- ☑ No hate speech or toxic words (based on a custom curated dataset)
- ☑ No banned or risky phrases
- ☑ No binary-like or suspicious encoding patterns
- ☑ Avoidance of ALL CAPS shouting or excessive punctuation
- ☑ Dictionary-based validation for meaningful content
- ☑ Detection of emojis or non-ASCII symbols

Encoding+Validating Illustration

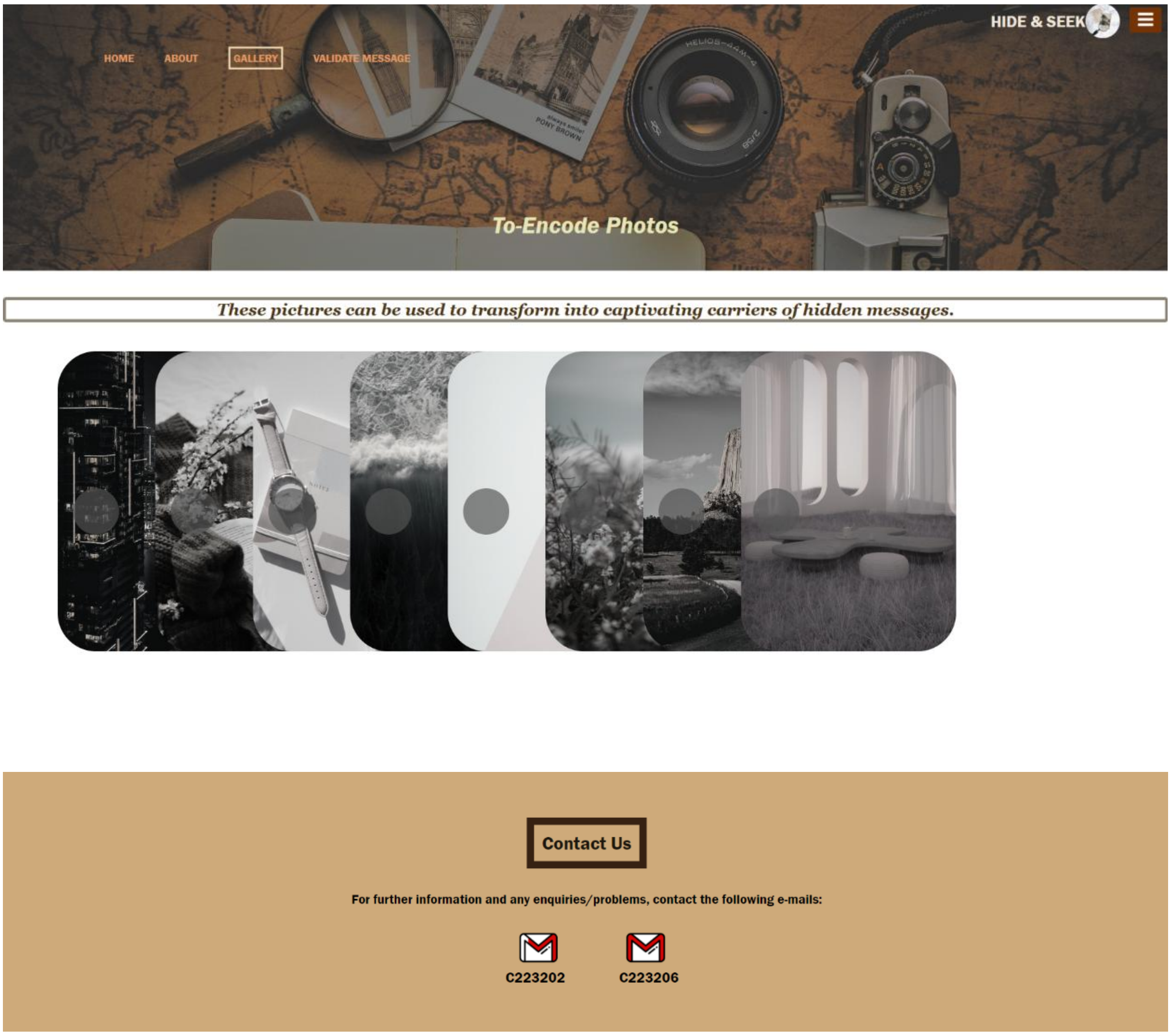
This ensures that only clean, purposeful, and human-friendly messages are embedded into images. The validator applies CSP logic, ensuring a lightweight, privacy-friendly solution suitable even for smaller systems.

Contact Us

For further information or queries, contact the following emails:

C223202 C223206

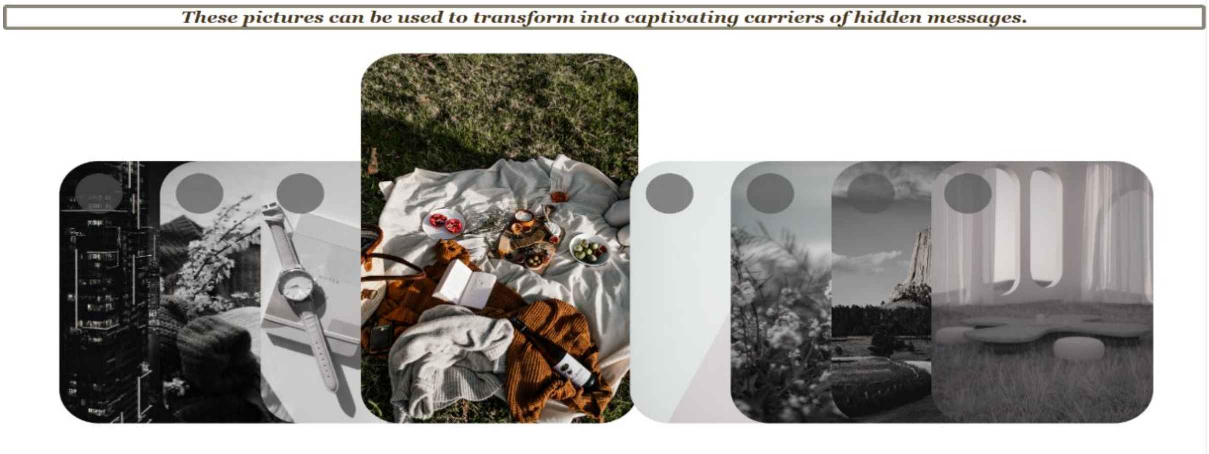
To-Encode Photo Gallery:



These pictures can be used to transform into captivating carriers of hidden messages.

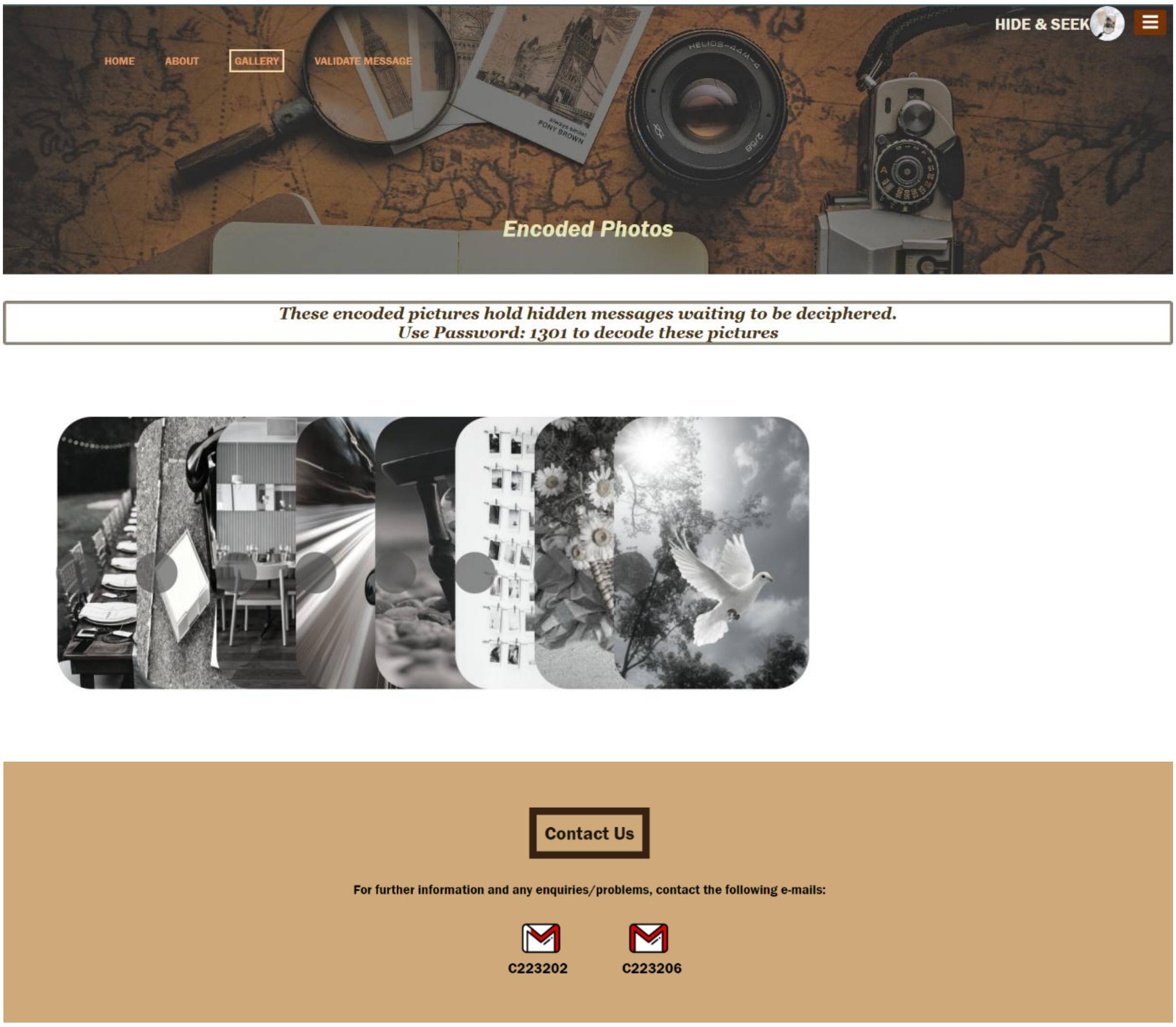
For further information and any enquiries/problems, contact the following e-mails:

Selected Photo View



These pictures can be used to transform into captivating carriers of hidden messages.

Encoded Photo Gallery:



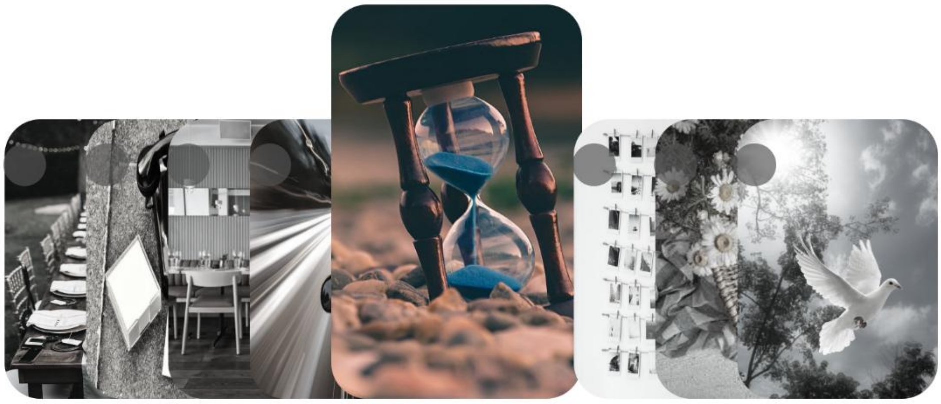
These encoded pictures hold hidden messages waiting to be deciphered.
Use Password: 1301 to decode these pictures

Contact Us

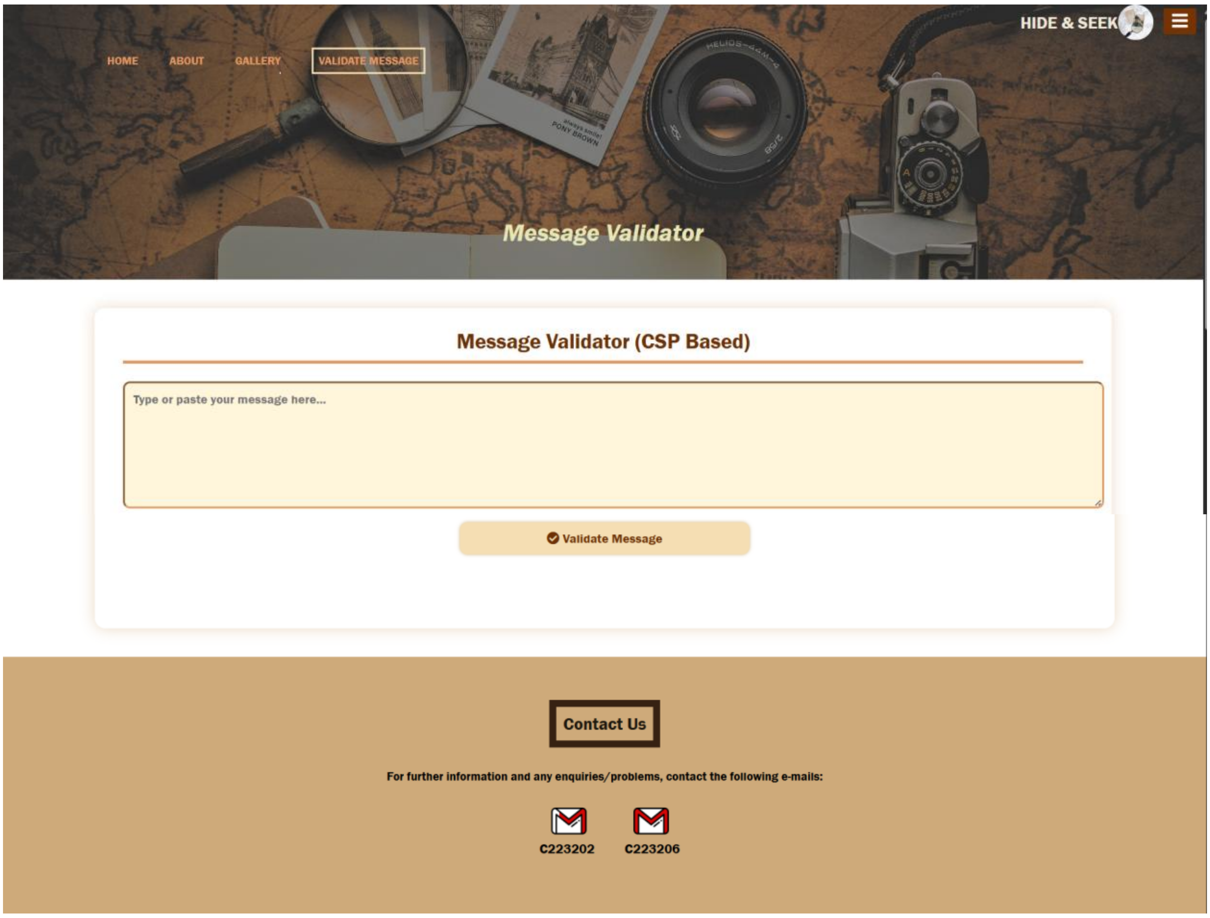
For further information and any enquiries/problems, contact the following e-mails:

C223202 C223206

Selected Photo View



Validate Message Page:



HOME ABOUT GALLERY VALIDATE MESSAGE

HIDE & SEEK

Message Validator

Message Validator (CSP Based)

Type or paste your message here...

Validate Message

Contact Us

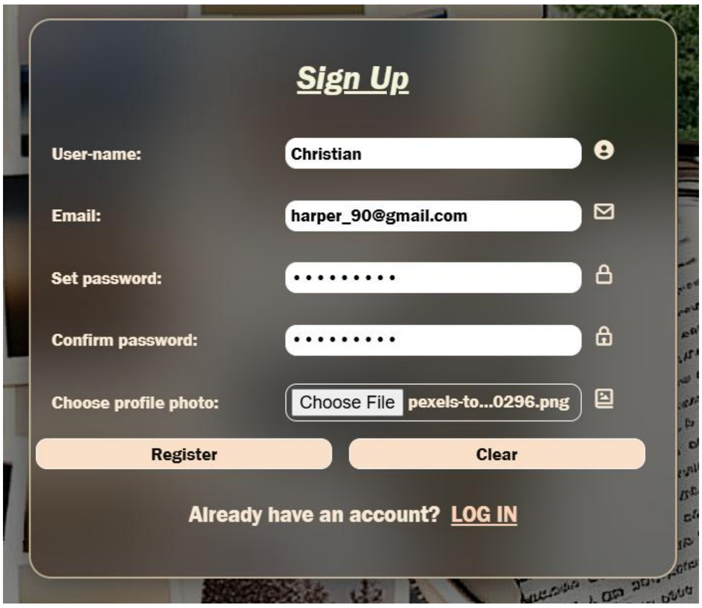
For further information and any enquiries/problems, contact the following e-mails:

C223202 C223206

The pictures provided here depict a primary overview of the whole website including the front page, profiles, account handling pages and the main five pages: Home, About, To-Encode Photos, Encoded Photos and Validate Message.

Working Demonstration

Account Registration & Logging In:



Sign Up

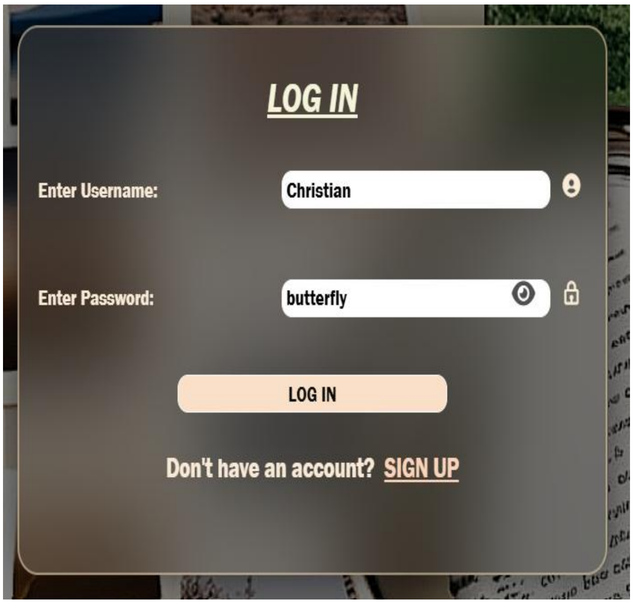
User-name: Christian

Email: harper_90@gmail.com

Set password: •••••••••

Confirm password: •••••••••

Choose profile photo: Choose File pexels-to...0296.png

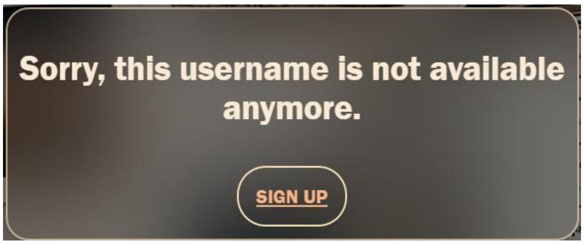


LOG IN

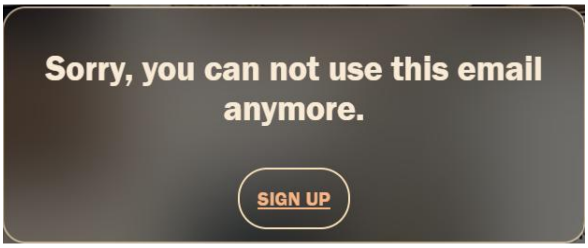
Enter Username: Christian

Enter Password: butterfly

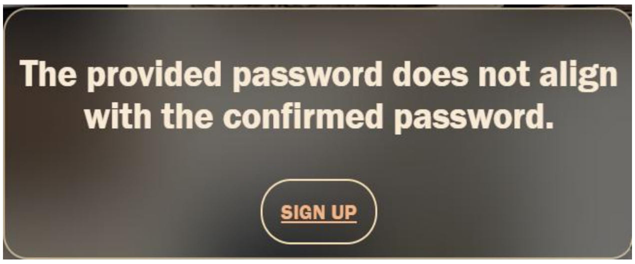
Error/Success Messages (Account Management):



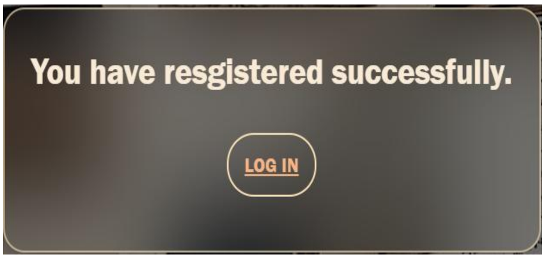
Sorry, this username is not available anymore.



Sorry, you can not use this email anymore.

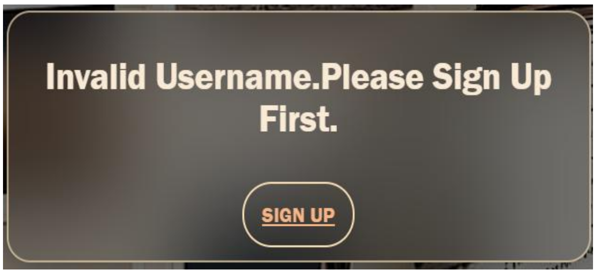


The provided password does not align with the confirmed password.

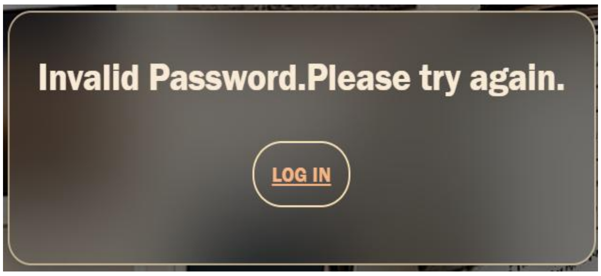


You have resgistered successfully.

During Signing Up

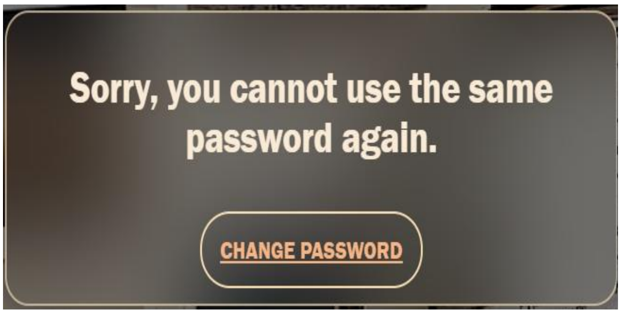


Invalid Username.Please Sign Up First.

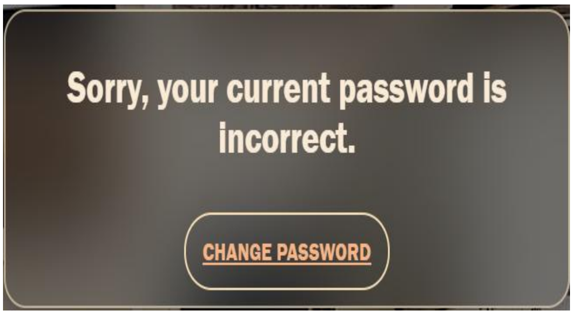


Invalid Password.Please try again.

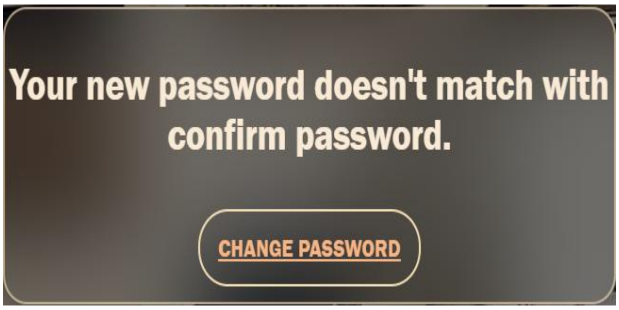
During Logging In



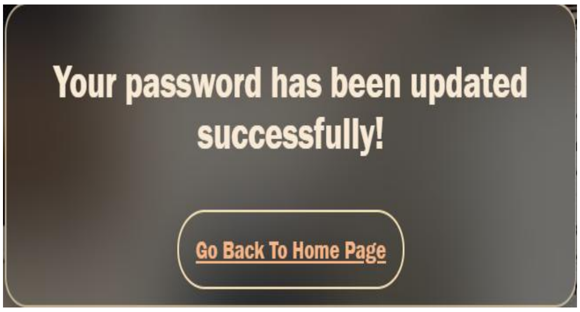
Sorry, you cannot use the same password again.



Sorry, your current password is incorrect.



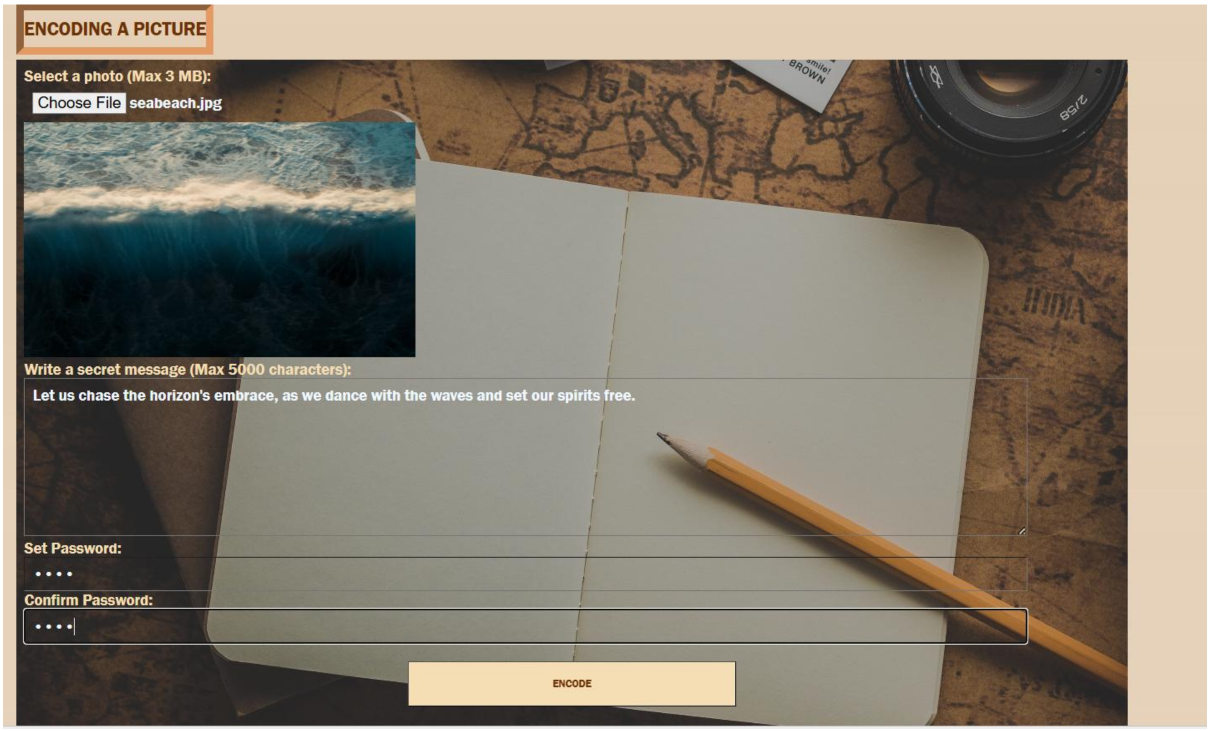
Your new password doesn't match with confirm password.



Your password has been updated successfully!

During Changing Password

Encoding Process:



ENCODING A PICTURE

Select a photo (Max 3 MB):

Choose File seabeach.jpg

Write a secret message (Max 5000 characters):

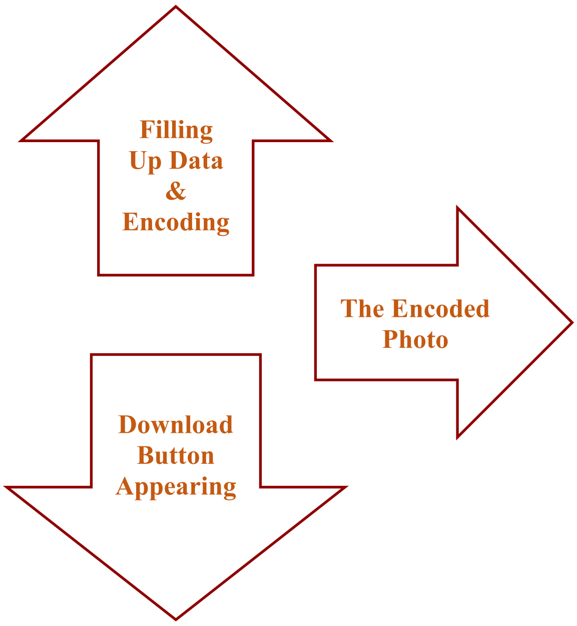
Let us chase the horizon's embrace, as we dance with the waves and set our spirits free.

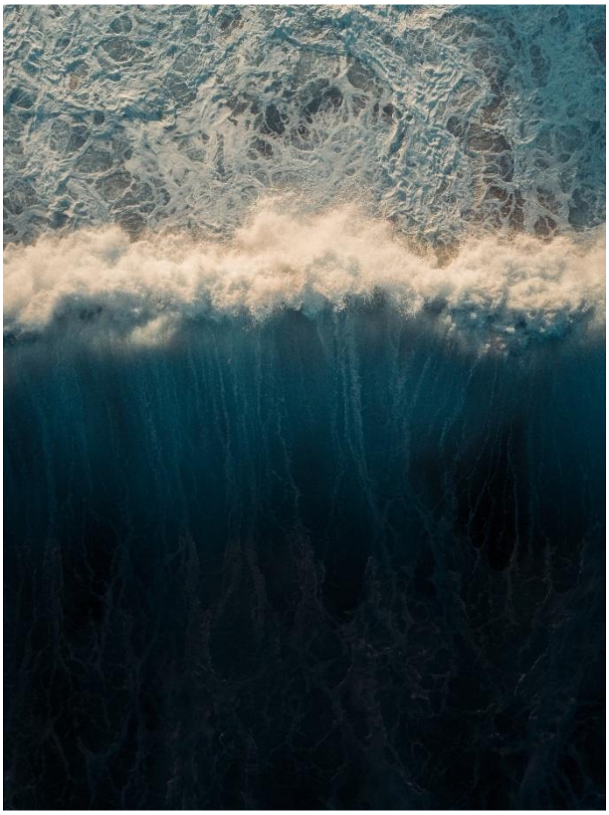
Set Password:

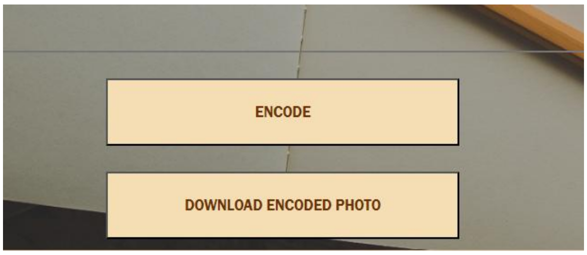
••••

Confirm Password:

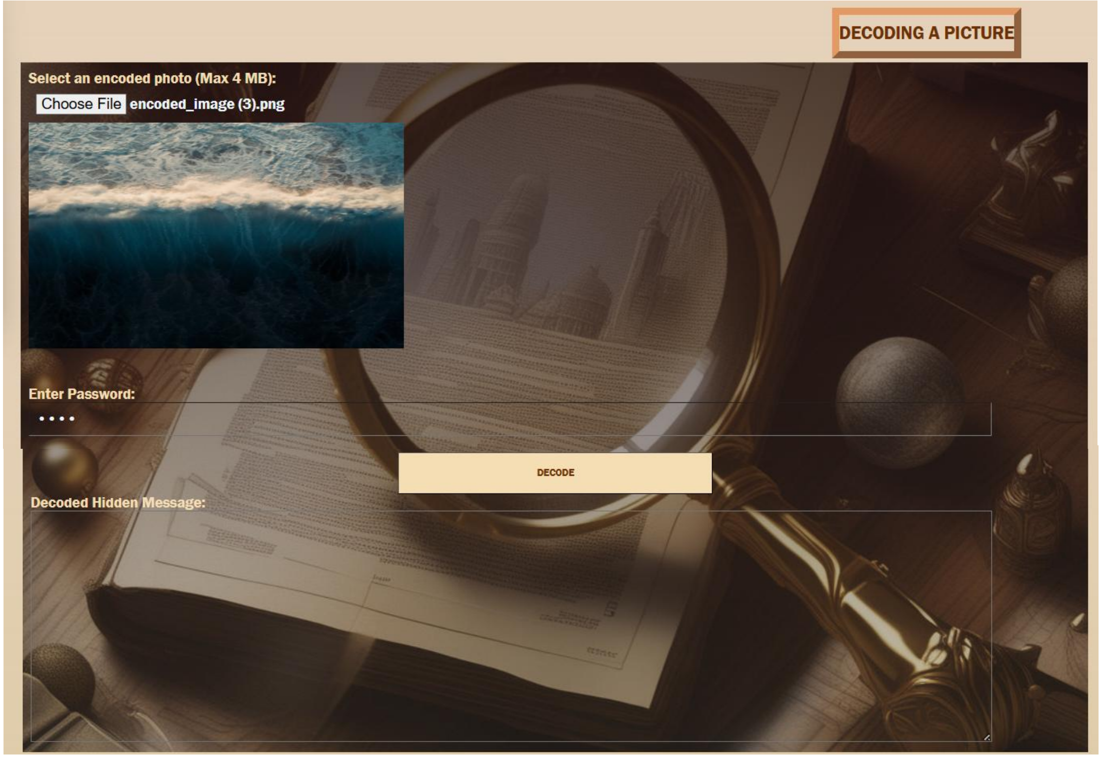
••••







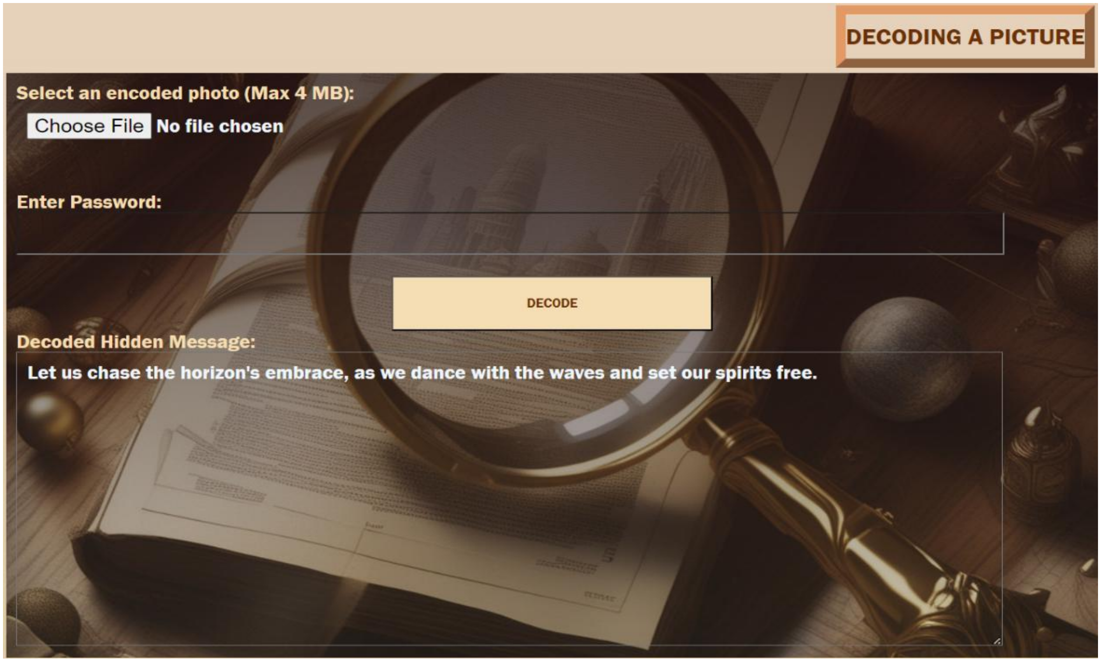
Decoding Process:



encoded_image (3).png

••••

Filling Up Data

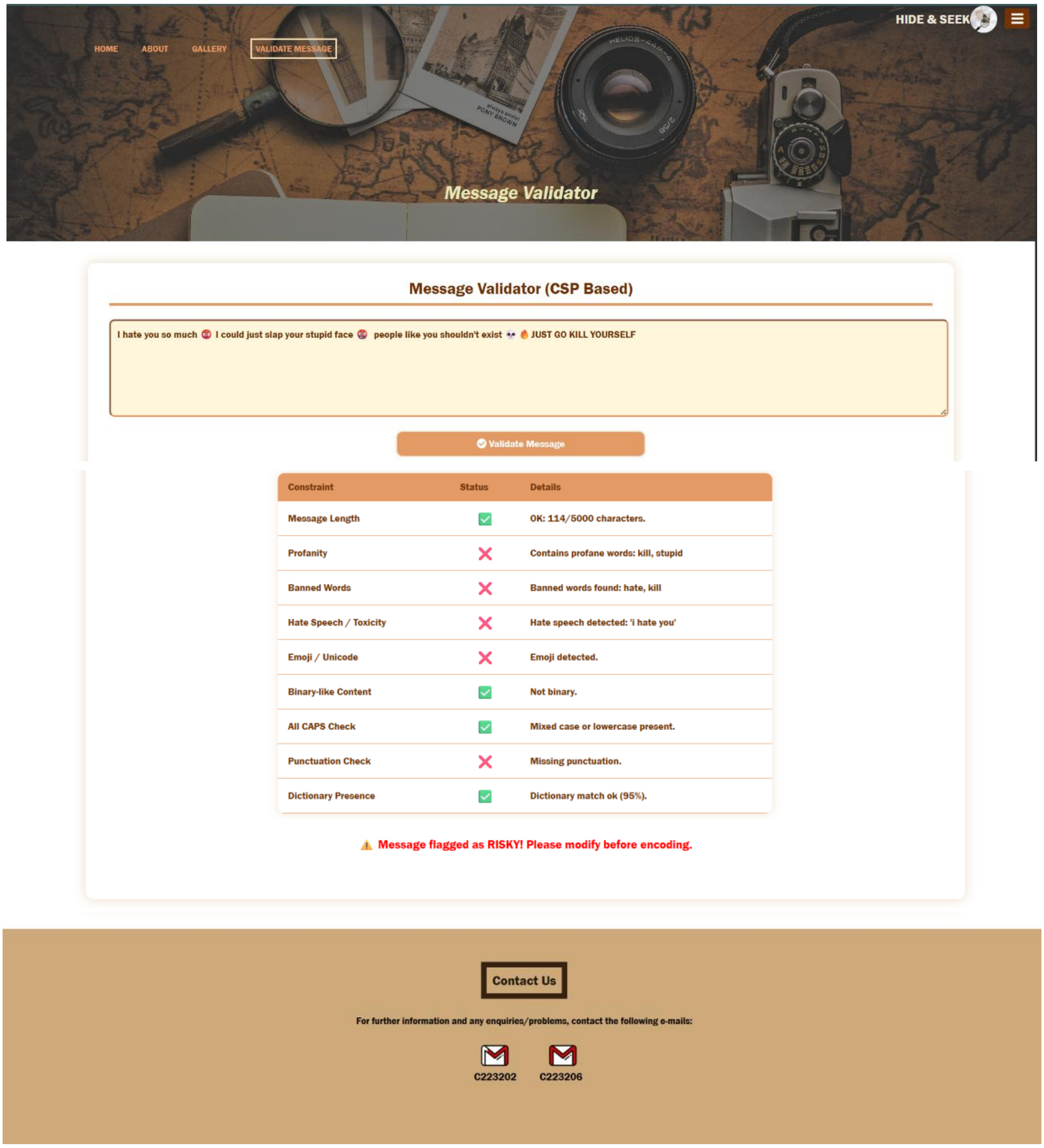


No file chosen

Let us chase the horizon's embrace, as we dance with the waves and set our spirits free.

The Decoded Message

Validating Message Process:



Message Validator (CSP Based)

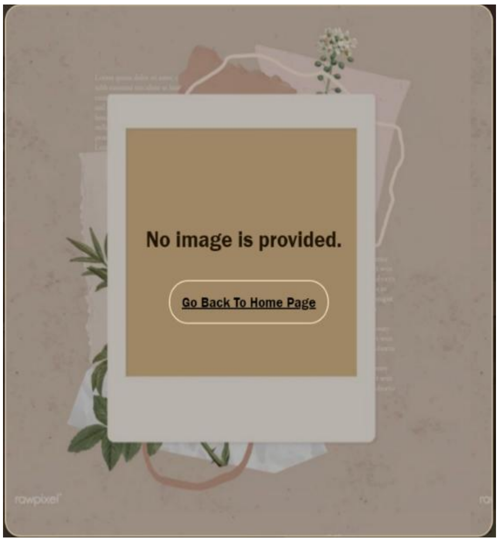
I hate you so much 😡 I could just slap your stupid face 🤬 people like you shouldn't exist 💀 🔥 JUST GO KILL YOURSELF

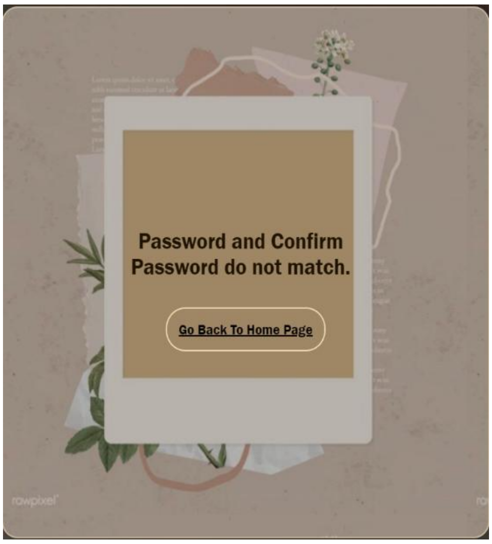
Constraint	Status	Details
Message Length	✅	OK: 114/5000 characters.
Profanity	✗	Contains profane words: kill, stupid
Banned Words	✗	Banned words found: hate, kill
Hate Speech / Toxicity	✗	Hate speech detected: 'i hate you'
Emoji / Unicode	✗	Emoji detected.
Binary-like Content	✅	Not binary.
All CAPS Check	✅	Mixed case or lowercase present.
Punctuation Check	✗	Missing punctuation.
Dictionary Presence	✅	Dictionary match ok (95%).

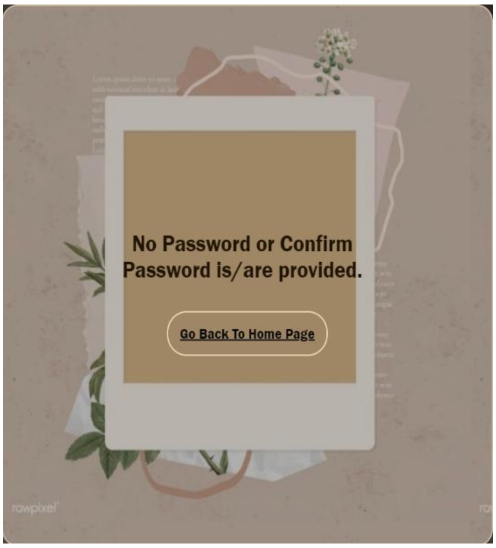
⚠️ Message flagged as RISKY! Please modify before encoding.

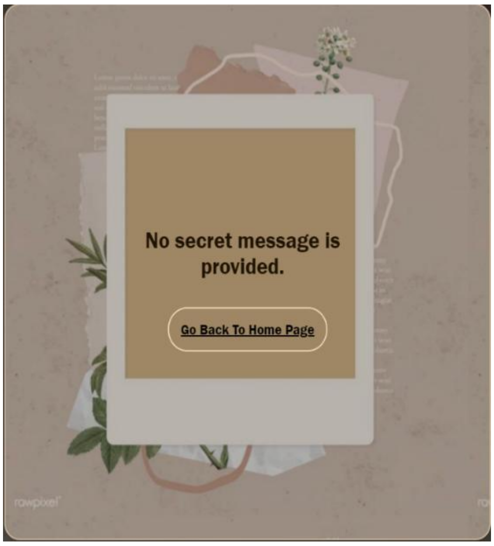
For further information and any enquiries/problems, contact the following e-mails:

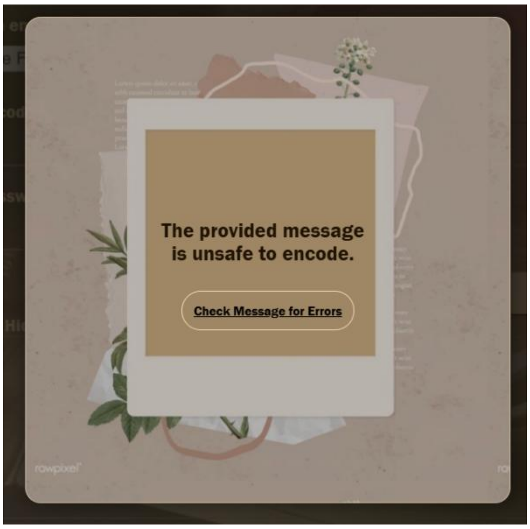
Error Messages (Encoding-Decoding):


No image is provided.

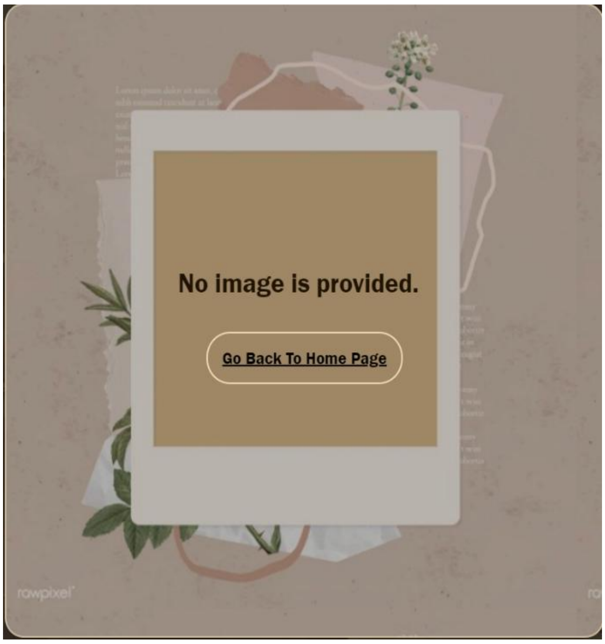

Password and Confirm Password do not match.


No Password or Confirm Password is/are provided.

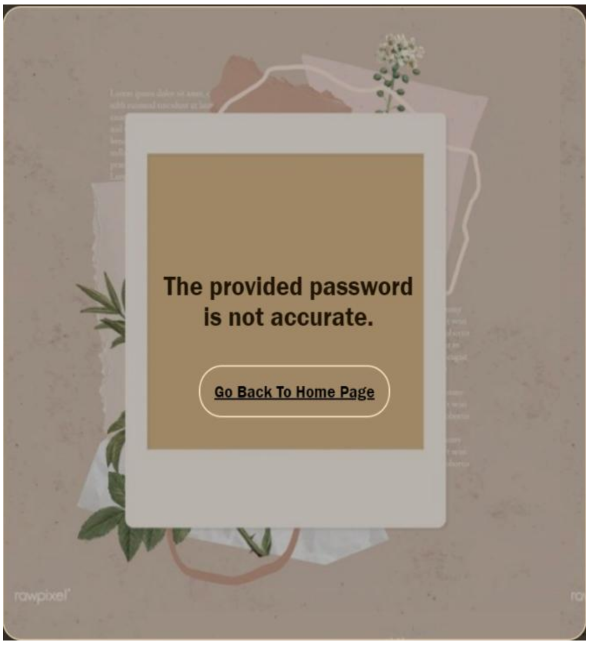

No secret message is provided.


The provided message is unsafe to encode.

During Encoding



No image is provided.



The provided password is not accurate.

During Decoding

The pictures provided here depict the overall working process of the website which are registering an account and logging in and the main works which are Encoding, Decoding Pictures and Validating Secret Messages.