A Project Report on

“STEGO TRIANGLE”

Submitted to partial fulfilment of the requirement of

Bachelors of Computer Application (BCA)
(Semester - VI)

Sant Gadge Baba Amravati University,
Amravati.

Submitted By:-
MS. MAYURI P. BOBADE
MS. HARSHIKA D. CHHOUNDIYA

Under the Guidance of:-
PROF. PRAVIN DHANDE

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who contributed to the successful completion of this project titled **“STEGO TRIANLGE”**.

First and foremost, I am deeply thankful to my project guide [Mr PRAVIN DHANDE SIR**]**, whose constant guidance, encouragement, and valuable suggestions helped me throughout the development of this project. Their technical insights and constructive feedback were instrumental in shaping this work.

I would also like to thank **[Prof. Avani Kulkarni Ma’am]**, Head of the Department of **[BCA]**, and all the faculty members of **[Smt. Maherbanu college of science & commerce, Akola]** for providing the necessary resources, support, and a conducive learning environment to carry out this project successfully.

I extend my heartfelt thanks to my friends and classmates for their cooperation, motivation, and constructive discussions during the course of this work. Their support made the learning process both enjoyable and productive.

Finally, I express my deepest gratitude to my parents and family members for their unwavering encouragement, patience, and moral support throughout my academic journey, without which this project would not have been possible.

THANK YOU

PREFACE

In the modern digital era, the rapid growth of information technology and multimedia communication has significantly increased the demand for secure data transmission. With the widespread use of audio and video files over the internet, ensuring the confidentiality and integrity of sensitive information has become a critical challenge. Traditional security mechanisms such as cryptography protect the content of data but often reveal the presence of encrypted information, which may attract unwanted attention.

Steganography provides an effective solution to this challenge by concealing secret information within digital media in such a way that the existence of the hidden data remains unnoticed. This project, titled **"Stego Triangle"**, focuses on embedding confidential information within audio and video files while maintaining their perceptual quality and usability. By combining steganography with encryption techniques, the project enhances security by not only hiding the data but also encrypting it to prevent unauthorized access.

The primary objective of this project is to design and implement a secure and user-friendly system that allows users to hide and retrieve secret data using audio and video files as carriers. The system ensures minimal distortion, high security, and ease of use through an interactive web-based interface. The implementation demonstrates how modern programming tools and algorithms can be used to strengthen secure communication in multimedia applications.

This project serves as an academic exploration of multimedia security concepts and highlights the practical application of steganography in real-world scenarios such as confidential communication, digital watermarking, and data protection. It also provides a foundation for further research and development in the field of secure data hiding and information security.

DECLARATION

We hear by declare that this project entitled

"STEGO TRIANGLE"

Submitted to Sant Gadge Baba Amravati University, for the degree of Bachelor of Computer Application, represents our original work. It was completed under the guidance of Prof. Pravin Dhande Sir.

All information, methods, and results presented are authentic and reflect our sincere efforts. No part of this work has been previously submitted for the award of any degree or diploma.

Ms. Mayuri Pravin Bobade
Ms. Harshika Dipchand Chhoundiya
Place:- Akola
Date:-01/02/2026

INDEX

1. INTRODUCTION

1.1 Introduction of the Project

In today's digital era, secure communication has become a critical requirement due to the exponential growth of data exchange over the internet. Audio and video files are extensively used for communication, entertainment, education, and business purposes. However, the open nature of digital networks makes sensitive information vulnerable to interception, unauthorized access, and cyber-attacks. Ensuring the confidentiality and security of transmitted information is therefore a major concern in modern communication systems.

Traditional security methods such as cryptography protect information by converting it into an unreadable format. Although encryption ensures data confidentiality, it does not conceal the existence of the message, which may attract attackers. To address this limitation, steganography provides an additional layer of security by embedding secret information within multimedia files such as audio and video in a manner that is imperceptible to human perception.

Steganography exploits the redundancy present in multimedia signals to hide confidential data without significantly affecting the quality of the carrier file. Audio and video steganography are particularly effective due to the large amount of data contained in multimedia files, making them ideal carriers for hidden information. When combined with encryption techniques, steganography ensures both secrecy and robustness, even if the hidden data is detected.

The project titled **"Stego Triangle"** focuses on designing and implementing a secure, efficient, and user-friendly system for hiding sensitive information within audio and video files. The system integrates encryption algorithms with steganographic techniques to enhance data security and prevent unauthorized access. A web-based interface is developed to simplify the encoding and decoding process for users.

1.2 Real-World Use Cases | Key Features

The applications of secure audio and video steganography extend across various domains, including:

1. Confidential Communication
Steganography can be used by individuals, organizations, and government agencies to transmit sensitive information secretly over public networks without revealing the existence of the message.

2. Military and Intelligence Services
Secure multimedia steganography plays a crucial role in covert communication, allowing classified information to be exchanged discreetly without raising suspicion.

3. Digital Watermarking and Copyright Protection
Audio and video steganography can embed ownership information, copyrights, or digital signatures into multimedia files to protect intellectual property and prevent piracy.

4. Secure Data Storage
Sensitive data such as passwords, authentication keys, or personal information can be hidden within multimedia files to provide an additional layer of security in data storage systems.

5. Medical and Healthcare Systems
Patient records and diagnostic information can be securely embedded within medical audio or video files, ensuring data privacy and compliance with healthcare regulations.

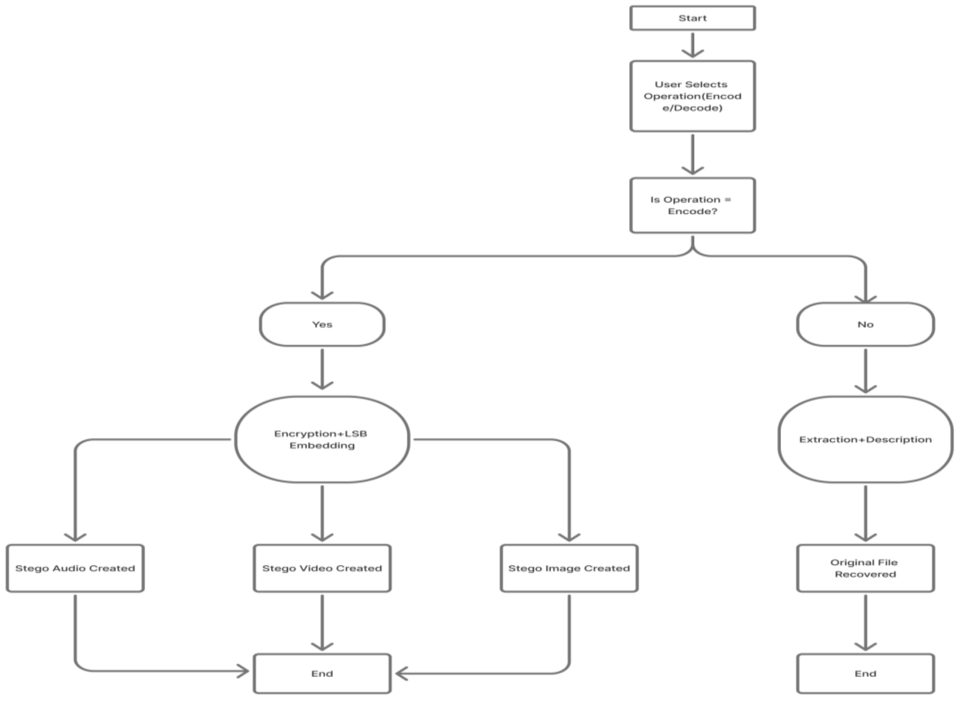
6. Journalism and Whistleblower Protection
Steganography can help journalists and whistleblowers securely transmit sensitive information while avoiding detection in hostile environments.

7. Cybersecurity and Forensic Applications
Steganography is used in cybersecurity research to analyze hidden communication channels and in digital forensics to detect unauthorized data hiding.

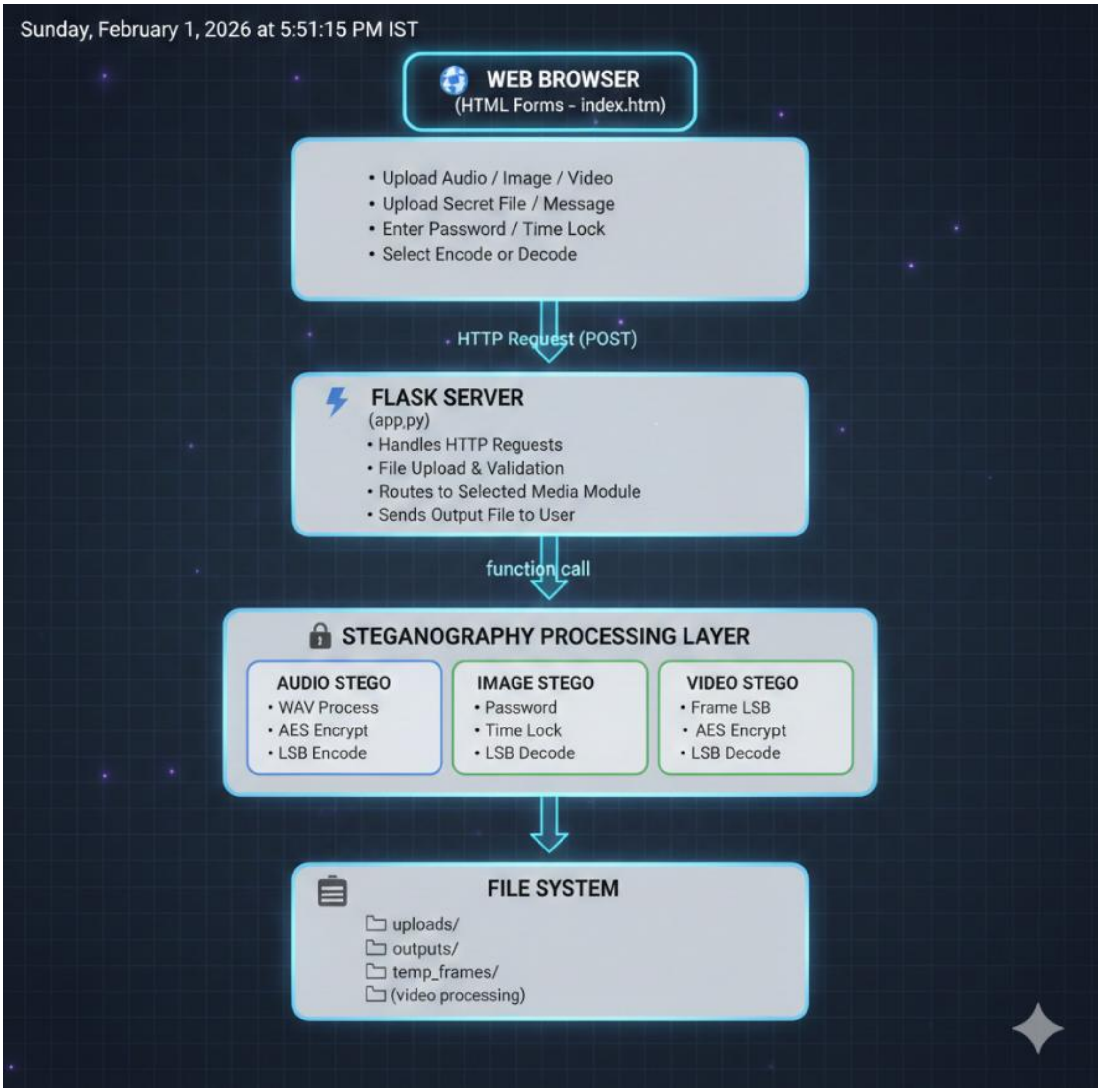
By addressing these real-world applications, this project demonstrates the practical relevance and importance of secure audio and video steganography in enhancing modern communication security. The system serves as a foundation for further research and development in multimedia security and covert communication technologies.

2. SYSTEM ANALYSIS & DESIGN

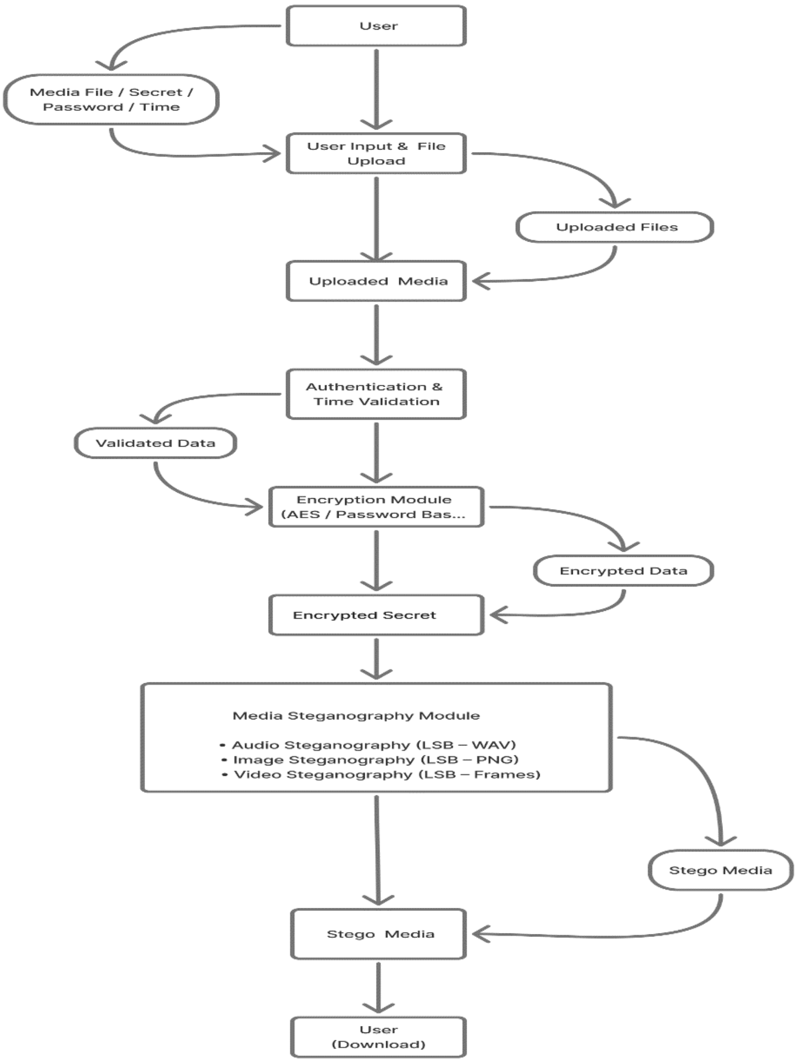
2.2 DATAFLOW DIAGRAM



2.3 HIGH LEVEL ARCHITECTURE



Level 1



User
Media File / Secret / Password / Time
User Input & File Upload
Uploaded Files
Uploaded Media
Authentication & Time Validation
Validated Data
Encryption Module (AES / Password Bas...
Encrypted Data
Encrypted Secret
Media Steganography Module
• Audio Steganography (LSB – WAV)
• Image Steganography (LSB – PNG)
• Video Steganography (LSB – Frames)
Stego Media
Stego Media
User (Download)

3. SYSTEM ARCHITECTURE

3.1 Frontend:-

- The frontend of the “STEGO TRIANGLE” is designed to provide users with a visually responsive interface, ensuring a experience throughout the process.
- Technologies Used:
- HTML (Hypertext Markup Language):
- Structure the content of web pages. Utilize semantic HTML for improved accessibility and search engine optimization.
- CSS3 (Glassmorphism UI)
- Implement styling to enhance the visual presentation of the interface. Ensure a consistent and aesthetically pleasing design across all pages.

3.2 Backend:-

- Flask (Python)

Flask is used in this project to act as the **backend framework** that connects the user interface with the steganography and encryption logic written in Python.

The core functionality of the project—such as hiding secret data, encrypting information, processing audio/video files, and extracting hidden data—is implemented in Python. Flask allows these Python functions to be executed through a **web-based interface**, making the system user-friendly and accessible.

Key Reasons for Using Flask

- **Lightweight and Simple**
  Flask is a micro web framework that is easy to understand and implement, making it ideal for academic and prototype-level projects.
- **Seamless Integration with Python Logic**
  Since the steganography and encryption algorithms are written in Python, Flask enables direct integration without the need for additional layers or complex configurations.
- **Web-Based User Interface**
  Flask allows users to interact with the system through a browser instead of using command-line commands, improving usability and accessibility.
- **File Upload and Download Handling**
  Flask efficiently manages uploading audio/video files, processing them on the server, and returning stego files or extracted secret data to the user.
- **Rapid Development**
  Flask supports fast development and easy debugging, which is suitable for student projects and research-based implementations.
- **Scalable for Future Enhancements**
  The project can be extended to support databases, authentication, cloud deployment, or REST APIs using Flask extensions.

3.3 Hardware & software Requirement | Minimum Hardware Requirements:-

- **Processor:** Intel/AMD processor (or equivalent)

- **RAM:** 4 GB or higher
- **Storage:** 20–30 GB free disk space
- **System Type:** Any standard desktop or laptop.

Software requirement:-

- Visual Studio, IDLE and Notepad
- Python Programming

4. PROJECT DESCRIPTION.

4.1 Purpose of the Project

The primary purpose of this project is to design and implement a secure system for hiding confidential information within audio and video files using steganography techniques combined with encryption. The project aims to ensure secure data transmission by concealing the existence of secret information, thereby preventing unauthorized detection and access.

In the current digital environment, where data interception and cyber threats are increasingly common, traditional encryption techniques alone are often insufficient, as they reveal the presence of sensitive data. This project addresses that limitation by embedding encrypted data within multimedia files, making the communication appear normal and harmless to unauthorized observers.

The project also aims to provide a user-friendly web-based interface that allows users to easily encode and decode secret information without requiring advanced technical knowledge. By maintaining the quality and integrity of the original audio and video files, the system ensures that the carrier media remains usable and visually or audibly indistinguishable from the original.

Additionally, the purpose of this project is to demonstrate the practical application of multimedia security concepts and to provide a foundation for further research in the field of information hiding, digital watermarking, and secure communication system.

4.2 Scope of the Project

The “STEGO TRIANGLE” project provides a strong foundation for secure multimedia communication; however, there are several areas where the system can be enhanced and extended in the future.

- **Support for Additional Media Formats**
  The system can be extended to support a wider range of audio and video formats, such as MP3, FLAC, MKV, and streaming media formats, improving compatibility and usability.
- **Advanced Steganography Techniques**
  More robust techniques such as frequency-domain steganography (DCT, DWT) can be implemented to improve resistance against steganalysis and compression attacks.
- **Stronger and Adaptive Encryption**
  The project can incorporate advanced encryption standards such as AES-256, hybrid encryption, or public-key cryptography to enhance data security.
- **User Authentication and Access Control**
  Adding user login, authentication, and role-based access control can help manage multiple users and prevent unauthorized access.

- **Cloud and Mobile Deployment**
  The system can be deployed on cloud platforms and extended into mobile applications to enable secure communication across devices and locations.
- **Machine Learning-Based Steganalysis Resistance**
  Machine learning techniques can be integrated to automatically adapt embedding patterns, making detection by steganalysis tools more difficult.
- **Real-Time Steganography**
  Future versions may support real-time audio or video streams, enabling secure live communication.
- **Digital Watermarking Applications**
  The project can be enhanced to support copyright protection, ownership verification, and tamper detection in multimedia content.

4.3 Overview of Project

The **STEGO TRIANGLE** project is designed to provide a secure and efficient method for concealing confidential information within multimedia files. The project integrates steganographic techniques with encryption mechanisms to ensure that sensitive data remains hidden and protected during transmission over public or unsecured networks

The system allows users to embed secret information into audio or video files in such a way that the presence of the hidden data is imperceptible to human perception. Before embedding, the secret information is encrypted using a password-based encryption algorithm, adding an extra layer of security. The encrypted data is then concealed within the carrier media using least significant bit (LSB) or similar steganographic techniques, ensuring minimal distortion to the original file.

The project is implemented as a web-based application with a simple and interactive graphical user interface. Users can upload an audio or video file, enter the secret message and encryption password, and generate a stego file containing the hidden information. Similarly, authorized users can extract and decrypt the hidden data from the stego file using the correct password.

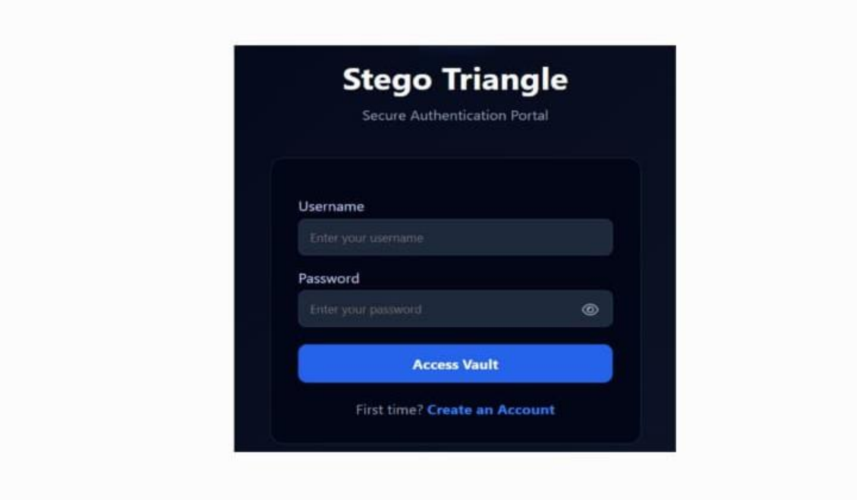
The system architecture consists of three main components:

- **User Interface Module** – Provides an easy-to-use interface for encoding and decoding operations.
- **Steganography Module** – Handles the embedding and extraction of encrypted data within audio and video files.
- **Encryption Module** – Ensures confidentiality of the hidden data through secure encryption and decryption techniques.

This project demonstrates the practical application of multimedia security concepts and highlights how steganography can be used alongside cryptography to enhance data protection. It serves as an effective solution for secure communication, digital watermarking, and information hiding, and provides a foundation for future enhancements in secure multimedia systems.

5. SYSTEM DESIGN, DEVELOPMENT & IMPLEMENTATION

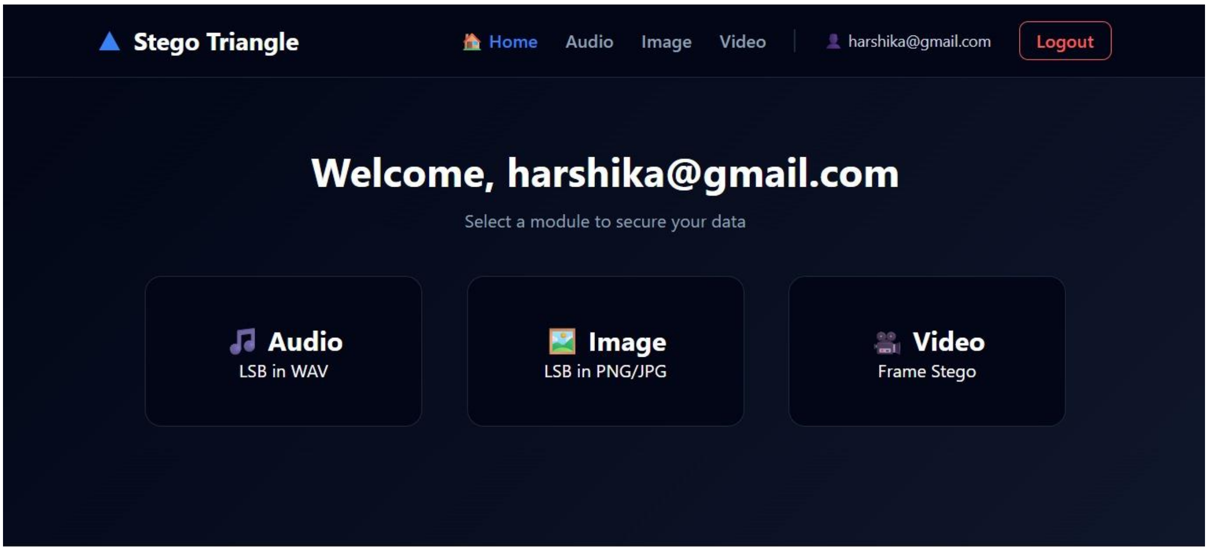
LOGIN PAGE



Overview of Login Page:-

- **Login page** of the *Stego Triangle* application
- Acts as a **secure authentication portal**
- Users enter **username and password** to access the system
- **Access Vault** button for secure login
- Option to **create a new account** for first-time users

<u>HOME PAGE</u>



Welcome, harshika@gmail.com

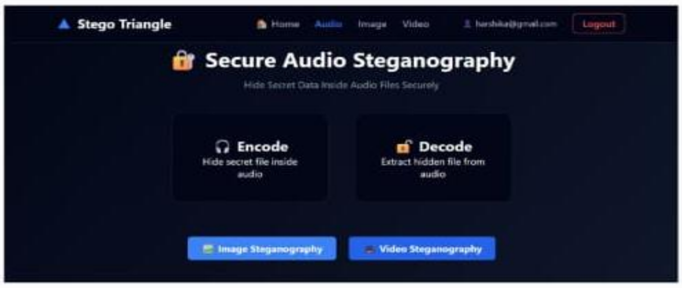
Select a module to secure your data

Home Page Description:-

- Acts as the main dashboard of the **Stego Triangle** application.
- Displays a navigation bar with options for **Home, Audio, Image, and Video** modules.
- Shows the logged-in user's email ID with a **Logout** option for session control.
- Provides a personalized welcome message to the user.
- Allows users to select a steganography module to secure data.
- Includes three module cards:
- **Audio** – LSB steganography in WAV files.

- **Image** – LSB steganography in PNG/JPG images.
- **Video** – Frame-based steganography in video files.
- Uses a dark-themed, modern interface for better usability and visual clarity.
- Ensures easy navigation and quick access to all core functionalities.

INPUT SCREEN:-

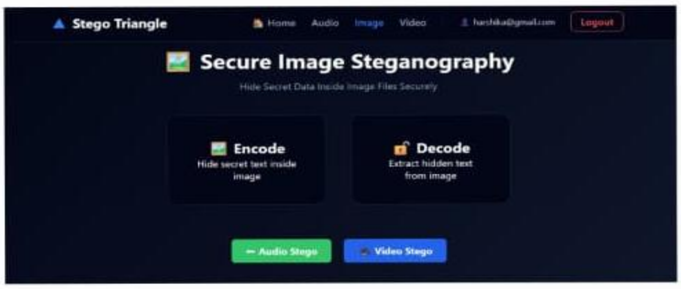


Secure Audio Steganography

Hide Secret Data Inside Audio Files Securely

Encode
Hide secret file inside audio

Decode
Extract hidden file from audio

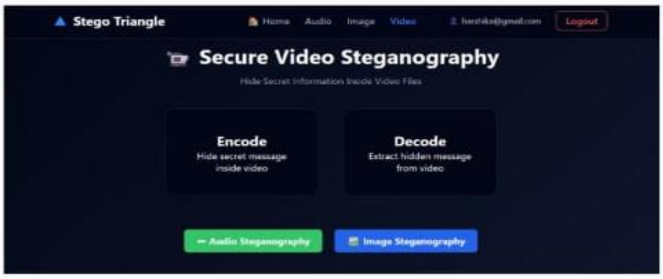


Secure Image Steganography

Hide Secret Data Inside Image Files Securely

Encode
Hide secret text inside image

Decode
Extract hidden text from image



Secure Video Steganography

Hide Secret Information Inside Video Files

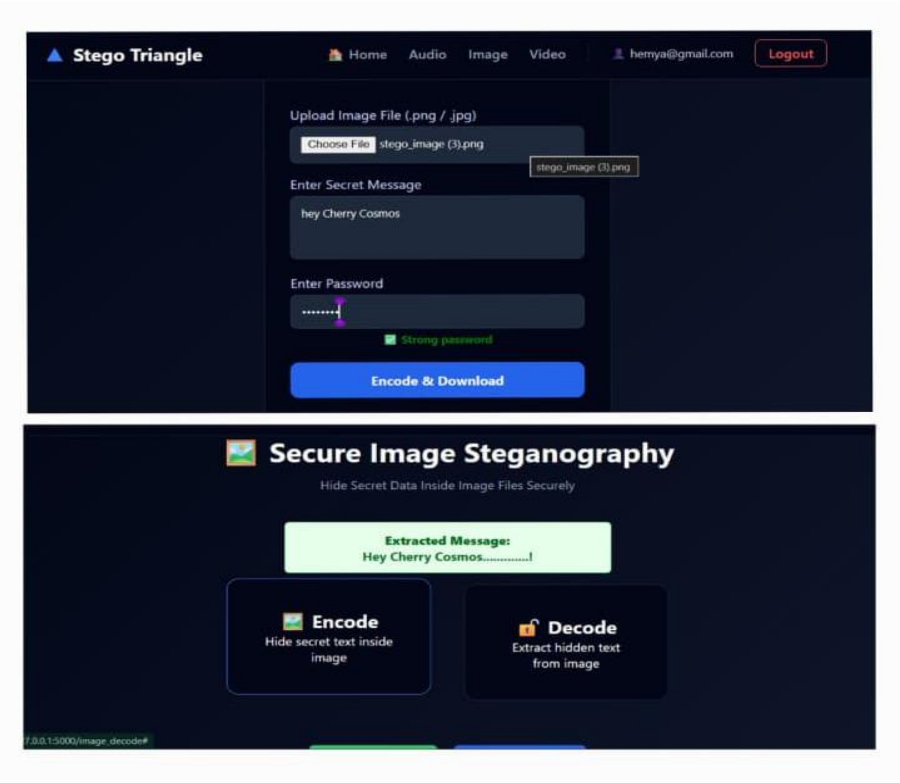
Encode
Hide secret message inside video

Decode
Extract hidden message from video

Input Screen Overview:-

- Displays **Stego Triangle** multimedia steganography interface
- Supports **Audio, Image, and Video** steganography
- Provides **Encode** and **Decode** options for each module

OUTPUT SCREEN:-



Upload Image File (.png / .jpg)
stego_image (3).png

Enter Secret Message
hey Cherry Cosmos

Enter Password
••••••••

Strong password

Secure Image Steganography

Hide Secret Data Inside Image Files Securely

Extracted Message:
Hey Cherry Cosmos..............!

7.0.0.1:5000/image_decode#

Output Image Description:-

- Displays the **Secure Image Steganography** module interface.
- Allows users to **upload an image file (PNG/JPG)**.
- Provides input fields for **secret message** and **password**.
- Shows **password strength indication** for security.
- Encodes the secret message into the image using steganography.
- Enables users to **download the stego image** after encoding.
- Displays the **extracted secret message** after decoding.
- Provides separate **Encode** and **Decode** options for clarity.
- Maintains a secure and user-friendly dark-themed interface.

6. CODING

App.py(main file)

```
from flask import Flask, render_template, request, send_file, redirect, url_for, session
import os
from audio_stego import encode_audio, decode_audio
from image_stego import encode_image, decode_image
from video_stego import encode_video, decode_video

app = Flask(__name__)
```

```
app.secret_key = "stego_triangle_secure_key_2026"  # Required for session management
UPLOAD_FOLDER = "uploads"
OUTPUT_FOLDER = "outputs"
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
os.makedirs(OUTPUT_FOLDER, exist_ok=True)
# ▲ In-memory user storage (Resets when server restarts)
USER_DATA = {"admin": "admin123"}
# ================= AUTHENTICATION ROUTES =================
@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        username = request.form.get("username")
        password = request.form.get("password")
                if username in USER_DATA and USER_DATA[username] == password:
            session["user"] = username
            # UPDATED: Redirect to dashboard instead of index
            return redirect(url_for("dashboard"))
        return render_template("login.html", error="❌ Invalid Username or Password")
    return render_template("login.html")
@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        u = request.form.get("username")
        p = request.form.get("password")
                if u in USER_DATA:
            return render_template("register.html", error="❌ User already exists!")
                USER_DATA[u] = p
        return render_template("login.html", success="✅ Registration Successful! Please
Login.")
    return render_template("register.html")
@app.route("/logout")
def logout():
    session.pop("user", None)
    return redirect(url_for("login"))
# ================= DASHBOARD (NEW HOME) =================
@app.route("/dashboard")
def dashboard():
    if "user" not in session:
        return redirect(url_for("login"))
    return render_template("dash.html")
# ================= AUDIO PAGE (index.html) =================
@app.route("/")
def index():
    if "user" not in session:
        return redirect(url_for("login"))
    return render_template("index.html")
# ================= IMAGE PAGE =================
@app.route("/image")
def image():
    if "user" not in session:
        return redirect(url_for("login"))
    return render_template("image.html")
# ================= VIDEO PAGE =================
@app.route("/video")
def video():
    if "user" not in session:
        return redirect(url_for("login"))
    return render_template("video_stego.html")
# ================= AUDIO ENCODE =================
@app.route("/encode", methods=["POST"])
def encode():
    if "user" not in session: return redirect(url_for("login"))
        audio = request.files["audio"]
    secret = request.files["secret"]
    password = request.form["password"]
    audio_path = os.path.join(UPLOAD_FOLDER, audio.filename)
    secret_path = os.path.join(UPLOAD_FOLDER, secret.filename)
    output_audio = os.path.join(OUTPUT_FOLDER, "stego.wav")
    audio.save(audio_path)
    secret.save(secret_path)
    encode_audio(audio_path, secret_path, password, output_audio)
    return send_file(output_audio, as_attachment=True)
# ================= AUDIO DECODE =================
@app.route("/decode", methods=["POST"])
```

```
def decode():
    if "user" not in session: return redirect(url_for("login"))
        audio = request.files["audio"]
    password = request.form["password"]
    audio_path = os.path.join(UPLOAD_FOLDER, audio.filename)
    output_file = os.path.join(OUTPUT_FOLDER, "extracted_secret")
    audio.save(audio_path)
    extracted_file = decode_audio(audio_path, password, output_file)
    if extracted_file is None:
        return render_template("index.html", error="❌ Wrong password or corrupted file!", highlight_password=True)
    return send_file(extracted_file, as_attachment=True)
# ================= IMAGE ENCODE =================
@app.route("/image_encode", methods=["POST"])
def image_encode():
    if "user" not in session: return redirect(url_for("login"))
        image = request.files["image"]
    message = request.form["message"]
    password = request.form["password"]
    if len(password) < 8:
        return render_template("image.html", error="❌ Password must be at least 8 characters long")
    image_path = os.path.join(UPLOAD_FOLDER, image.filename)
    image.save(image_path)
        output_image = os.path.join(OUTPUT_FOLDER, "stego_image.png")
    result = encode_image(image_path, message, password, output_image)
    if result and os.path.exists(result):
        return send_file(result, as_attachment=True)
    else:
        return render_template("image.html", error="❌ Encoding failed. Image may be too small.")
# ================= IMAGE DECODE =================
@app.route("/image_decode", methods=["POST"])
def image_decode():
    if "user" not in session: return redirect(url_for("login"))
   image = request.files["image"]
password = request.form["password"]
    image_path = os.path.join(UPLOAD_FOLDER, image.filename)
    image.save(image_path)
    secret = decode_image(image_path, password)
    if secret is None or "Error" in secret:
        return render_template("image.html", error="❌ Incorrect password or no hidden
message found.")
    return render_template("image.html", success="✅ Message Extracted Successfully",
secret_message=secret)
# ================= VIDEO ENCODE =================
@app.route("/video_encode", methods=["POST"])
def video_encode():
    if "user" not in session: return redirect(url_for("login"))
      video = request.files["video"]
    secret = request.form["secret"]
    password = request.form["password"]
    video_path = os.path.join(UPLOAD_FOLDER, video.filename)
    output_video = os.path.join(OUTPUT_FOLDER, "stego_video.avi")
    video.save(video_path)
    encode_video(video_path, secret, password, output_video)
    return send_file(output_video, as_attachment=True)
# ================= VIDEO DECODE =================
@app.route("/video_decode", methods=["POST"])
def video_decode():
    if "user" not in session: return redirect(url_for("login"))
        video = request.files["video"]
    password = request.form["password"]
    video_path = os.path.join(UPLOAD_FOLDER, video.filename)
    video.save(video_path)
    secret = decode_video(video_path, password)
    if secret is None:
        return render_template("video_stego.html", error="❌ Wrong password or invalid video
file.")
    output_file = os.path.join(OUTPUT_FOLDER, "video_secret.txt")
    with open(output_file, "w") as f:
        f.write(secret)
    return send_file(output_file, as_attachment=True)
# ================= RUN =================
if __name__ == "__main__":
    app.run(debug=True)
```

7. CONCLUSION

The scope of this project focuses on the development and implementation of a secure system that enables the hiding and extraction of confidential information within audio and video files using steganography techniques combined with encryption. The project is designed to ensure secure data transmission while preserving the quality and integrity of the carrier multimedia files.

The system supports embedding text-based secret information into commonly used audio and video formats, making it suitable for practical and real-world applications. Encryption mechanisms are incorporated to enhance security, ensuring that even if the hidden data is detected, it cannot be accessed without proper authentication credentials.

- This project primarily emphasizes:
- Secure transmission of sensitive information over public networks.
- Protection of confidential data from unauthorized access and detection.
- Implementation of multimedia steganography techniques using audio and video files.
- Development of a user-friendly web-based interface for encoding and decoding operations.
- Demonstration of practical applications of information hiding in multimedia security.

The scope of the project is limited to software-based implementation and does not include hardware-level security mechanisms. The project focuses on basic and efficient steganographic techniques suitable for academic and prototype-level applications rather than advanced industrial-grade solutions. Additionally, the system is designed for controlled environments and does not address real-time streaming or large-scale distributed systems.

Despite these limitations, the project provides a strong foundation for future enhancements such as support for additional media formats, improved robustness against steganalysis attacks, integration with cloud-based platforms, and application of advanced cryptographic and machine learning techniques.

8. BIBLIOGRAPHY

- Johnson, N. F., & Jajodia, S., *Exploring steganography: Seeing the unseen*, IEEE Computer, 1998.
- Provos, N., & Honeyman, P., *Hide and seek: An introduction to steganography*, IEEE Security & Privacy, 2003.
- Katzenbeisser, S., & Petitcolas, F., *Information Hiding Techniques*, Artech House, 2000.
- Python Documentation – https://docs.python.org/
- Flask Documentation – https://flask.palletsprojects.com/