ENCAP(Image Steganography)

MINOR PROJECT REPORT

Submitted in partial fulfillment of the requirement for the Degree of Bachelors of Engineering in Computer Science & Engineering

Submitted To:

[PARUL UNIVERSITY, VADODARA, GUJARAT (INDIA)]

Submitted By:

Chavda Sujitkumar J. (2303051057010)
Patel Meet D. (2303051057054)
Rohit Khushbu R. (2303051057066)
Bhojaviya Abhishekkumar B. (2303051057008)

Under The Guidance of:
Mrs. Frenisha Jaimish Digaswala
(Assistant Professor, Computer Science & Engineering)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PARUL INSTITUTE OF TECHNOLOGY VADODARA, GUJARAT

SESSION: AY 2024-2025

Parul University
Parul Institute of Technology

(Session: 2024 -2025)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that **("Chavda Sujitkumar J, Patel Meet D, Rohit Khushbu R, Bhojaviya Abhishekkumar B"),** Students of **CSE VI Semester** of **"Parul Institute of Technology, Vadodara"** has completed their **Minor Project** titled **"ENCAP"**, as per the syllabus and has submitted a satisfactory report on this project as a partial fulfillment towards the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** under **Parul University, Vadodara, Gujarat (India)**.

Mrs. Frenisha Digaswala	**Prof. Sumitra Menaria**	**Dr. Swapnil Parikh**
Assistant Professor (CSE)	**Head (CSE)**	**Principal**
PIT, Vadodara	**PIT, Vadodara**	**PIT, Vadodara**

DECLARATION

We the undersigned solemnly declare that the project report "**ENCAP"** is based on my own work carried out during the course of our study under the supervision of **Mrs. Frenisha Jaimish Digaswala, Assistant Professor, CSE**.

We assert the statements made and conclusions drawn are the outcomes of my own work. I further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree / diploma / certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.

Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

CHAVDA SUJITKUMAR J. [2303051057010] _____

PATEL MEET D. [2303051057054] _____

ROHIT KHUSHBU R. [2303051057066] _____

BHOJAVIYA ABHISHEKKUMAR B. [2303051057008] _____

ACKNOWLEDGEMENT

In this semester, we have completed our project on **ENCAP**. During this time, all the group members collaboratively worked on the project and learnt about the industry standards that how projects are being developed in IT Companies. We also understood the importance of teamwork while creating a project and got to learn the new technologies on which we are going to work in near future.

We gratefully acknowledge for the assistance, cooperation guidance and clarification provided by **Mrs. Frenisha Digaswala** during the development of our project. We would also like to thank our Head of Department **Prof. Sumitra Menaria** and our Principal **Dr. Swapnil Parikh** Sir for giving us an opportunity to develop this project. Their continuous motivation and guidance helped us overcome the different obstacles for completing the Project.

We perceive this as an opportunity and a big milestone in our career development. We will strive to use gained skills and knowledge in our best possible way and we will work to improve them.

CHAVADA SUJITKUMAR J. [2303051057010] _____

PATEL MEET D. [2303051057054] _____

ROHIT KHUSHBU R. [2303051057066] _____

BHOJAVIYA ABHISHEKKUMAR B. [2303051057008] _____

LIST OF FIGURES

ABSTRACT

The Encap API is a steganography-based encryption tool designed to securely hide and transfer sensitive data, including text, images, files, and audio, within image files. By combining advanced encryption techniques with steganography, Encap ensures that data remains confidential, tamper-proof, and undetectable to unauthorized parties. The API provides a simple, RESTful interface that can be easily integrated into various applications, making it ideal for secure messaging, data protection, and covert communication. This project focuses on enhancing privacy in digital communication by embedding encrypted data within innocuous images, enabling secure and covert transmission of information over potentially insecure networks. The Encap API offers a solution for users and organizations that require robust privacy measures for sensitive data, addressing the growing need for secure and encrypted communication in an increasingly digital world.

TABLE OF CONTENTS

CHAPTER I
INTRODUCTION

1.1 Overview

The **Encap API** is an innovative solution that leverages **steganography** and **encryption** techniques to enable secure data hiding and transfer. This API allows users to embed various types of sensitive data—such as text, images, files, and audio—within image files, ensuring both the **confidentiality** and **integrity** of the information. By combining **encryption** with **steganography**, Encap ensures that the data is protected from unauthorized access and remains undetectable during transmission.

The Encap API is designed with a **RESTful interface**, making it easy to integrate into any application, regardless of platform. It provides a seamless method for embedding encrypted data into an image file, which can then be securely transmitted. Only authorized recipients, with the appropriate decryption key, can extract and decrypt the hidden data from the image, ensuring secure communication.

This project addresses the growing need for privacy and data protection in digital communications, particularly for secure messaging and covert data transfer. Whether for personal or organizational use, the Encap API offers a reliable method to send and receive sensitive information without the risk of interception or tampering.

1.2 Problem Statement

In today's digital world, **data security and privacy** are major concerns. Traditional encryption methods help protect sensitive data, but **they are easily detectable**, making them a target for hackers and surveillance systems. **Steganography**, which involves hiding data within media files, offers an additional layer of security by making the data invisible to unauthorized users.

1.3 Objective of Project

- **Secure Data Hiding**: Enable users to securely embed text, files, and audio within image files, ensuring data confidentiality.

- **Encryption & Encoding**: Use advanced AES encryption to secure data before embedding it into images, ensuring that hidden information remains protected.
- **Decoding & Decryption**: Allow authorized users to extract and decrypt the hidden data, ensuring only intended recipients can access it.
- **Secure User Authentication**: Implement robust user authentication mechanisms to ensure that only authorized individuals can send or receive data through the API.
- **REST API Integration**: Provide a simple, RESTful API that allows developers to easily integrate secure data hiding and transmission features into their applications.
- **Cross-Platform Compatibility:** Ensure the API supports integration across multiple platforms, including web, mobile, and desktop applications.

1.4 Applications or Scope

- **Secure Messaging & Communication**: Encrypt and hide sensitive conversations within images to ensure private and untraceable communication.
- **Confidential File Transfers:** Securely share documents, passwords, keys, and other sensitive information over public or insecure networks.
- **Cybersecurity & Digital Forensics:** Safely conceal and retrieve forensic evidence, ensuring it is protected from unauthorized access during investigations.
- **Watermarking & Digital Rights Protection:** Embed hidden copyrights, ownership information, or digital signatures in media files, offering a method for protecting intellectual property.
- **Covert Intelligence & Military Communication:** Conceal classified or sensitive data within seemingly harmless images for secure and undetectable communication in intelligence or military operations.

1.5 Organization of Report

This report is structured as follows:

- **Chapter I** Introduces the project.
- **Chapter II** Covers the literature survey.
- **Chapter III** Explains the methodology, technologies, and system design.
- **Chapter IV** Discusses system requirements.
- **Chapter V** Presents expected outcomes and GUI snapshots.
- **Chapter VI** Provides the conclusion and future work.
- **Chapter VII** Includes references and a weekly progress.

CHAPTER II
LITERATURE SURVEY

In the field of digital communication, securing sensitive data is essential. Steganography and encryption have emerged as critical techniques for ensuring data confidentiality and integrity.

Steganography:

- Steganography involves hiding data within other innocuous data, such as embedding messages within image files. Early methods, such as Least Significant Bit (LSB) encoding, allow data to be embedded in image pixels but are often vulnerable to detection or tampering (Johnson & Jajodia, 1998). More advanced techniques, such as using the Discrete Cosine Transform (DCT) (Hossain et al., 2012), improve security by reducing visible alterations to the host image, making detection more difficult.

Encryption:

- AES encryption is one of the most widely used cryptographic techniques for securing digital data. It provides robust protection against unauthorized access and ensures the confidentiality of the information (Katz & Lindell, 2007). The combination of steganography and encryption ensures both secrecy and security, making the data both hidden and protected from decryption without the appropriate key.

CHAPTER III
METHODOLOGY

3.1 Background / Overview of Methodology

The Encap API is designed to provide a secure and efficient way to encapsulate data (text, images, audio, files) within images using steganography. Steganography is a technique of hiding information inside a carrier medium, such as an image, to ensure covert communication and data security.

- **Secure User Authentication** – Only authorized users can encode and decode data.
- **Efficient Data Hiding** – Implementing Least Significant Bit (LSB) Steganography to conceal data within images.
- **Controlled Data Sharing** – The encapsulated image can only be decoded within the application.
- **Integrity & Performance Testing** – Ensuring the hidden data is extracted without loss or distortion.

The methodology is implemented using **Python-based backend APIs**, ensuring scalability, security, and performance.

3.2 Project Platforms used in Project

- **Backend:**
  - Python (Django Framework): Django, a powerful and secure web framework, is used to build the API. It provides a robust platform for handling encryption, steganography, and user authentication efficiently.
- **Database:**
  - MySQL: MySQL is used for storing user details, encrypted data, and transaction logs. It ensures secure and structured management of sensitive information and API interactions.

- **Steganography Libraries:**
  - PIL (Python Imaging Library): Used for handling image processing tasks such as opening, modifying, and saving images, which is essential for steganographic operations.
- **Encryption Libraries:**
  - Cryptography (Python Library - Cipher, Algorithms, and Modes): Provides low-level cryptographic primitives to implement encryption techniques for securing data before embedding it into images
- **API Testing:**
  - Postman: Postman is used to test the API endpoints and ensure that the encryption, embedding, extraction, and decryption functionalities work as expected. It also helps in validating secure data transfer between client and server.
- **Development Environment:**
  - VS Code: Visual Studio Code is used as the main IDE for writing and debugging the API code.
- **Version Control:**
  - Git and GitHub: Git is used for version control, and GitHub hosts the project repository, allowing team collaboration and project tracking.

3.3 Proposed Methodology

The Encap API is designed to provide a secure and efficient method for hiding and extracting sensitive data using steganography and encryption. The proposed methodology follows a structured approach to ensure data security, system efficiency, and seamless integration for users and developers.

- **Step-by-Step Methodology**

  - User Authentication & Authorization
  - Data Encryption & Steganography Encoding
  - Steganography Decoding & Data Extraction
  - API Integration & Request Processing
  - Logging & Monitoring

3.4 Project Modules

1. User Authentication Module (Login & Logout)

- User Registration – Users sign up with email and password.
- Login – Authenticate users using email and password.
- Logout – Securely log out users and clear sessions.
- Account Management – Handle password resets and user profiles.

2. API Key Management Module

- Generate API Key – Users can request an API key.
- Validate API Key – Each request must include a valid API key.
- Revoke API Key – Admins can deactivate API keys if needed.
- Monitor API Usage – Track the number of requests per key.

3. Encryption (Data Hiding) Module

- Upload Data – Users upload data to be hidden.
- Select Host Image – Choose an image for steganography.
- Encode & Encrypt – Securely hide the data inside the image.
- Generate Stego Image – Download the final image with hidden data.

4. Decryption (Data Extraction) Module

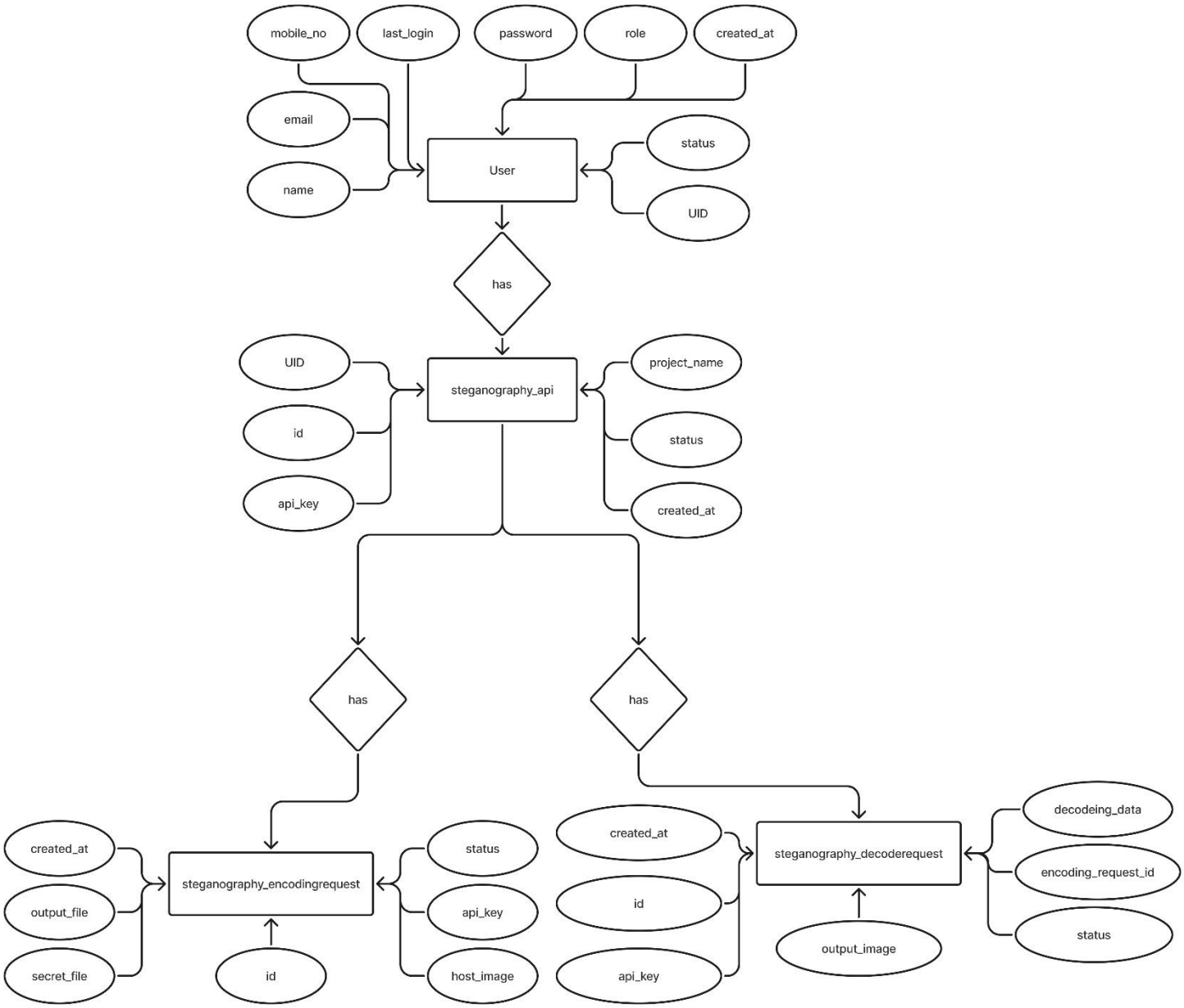
- Upload Stego Image – Users upload the encoded image.
- Extract Data – The system retrieves hidden text, files, or media.
- Decrypt if Necessary – If encryption was used, decrypt the data.
- Download Extracted Data – Users can retrieve their hidden content.

5. Request Processing & Logging Module

- Track Encoding & Decoding Requests – Each request is logged.
- Monitor Status – Pending, Processing, Completed, or Failed.
- Error Handling – If something goes wrong, logs the issue.

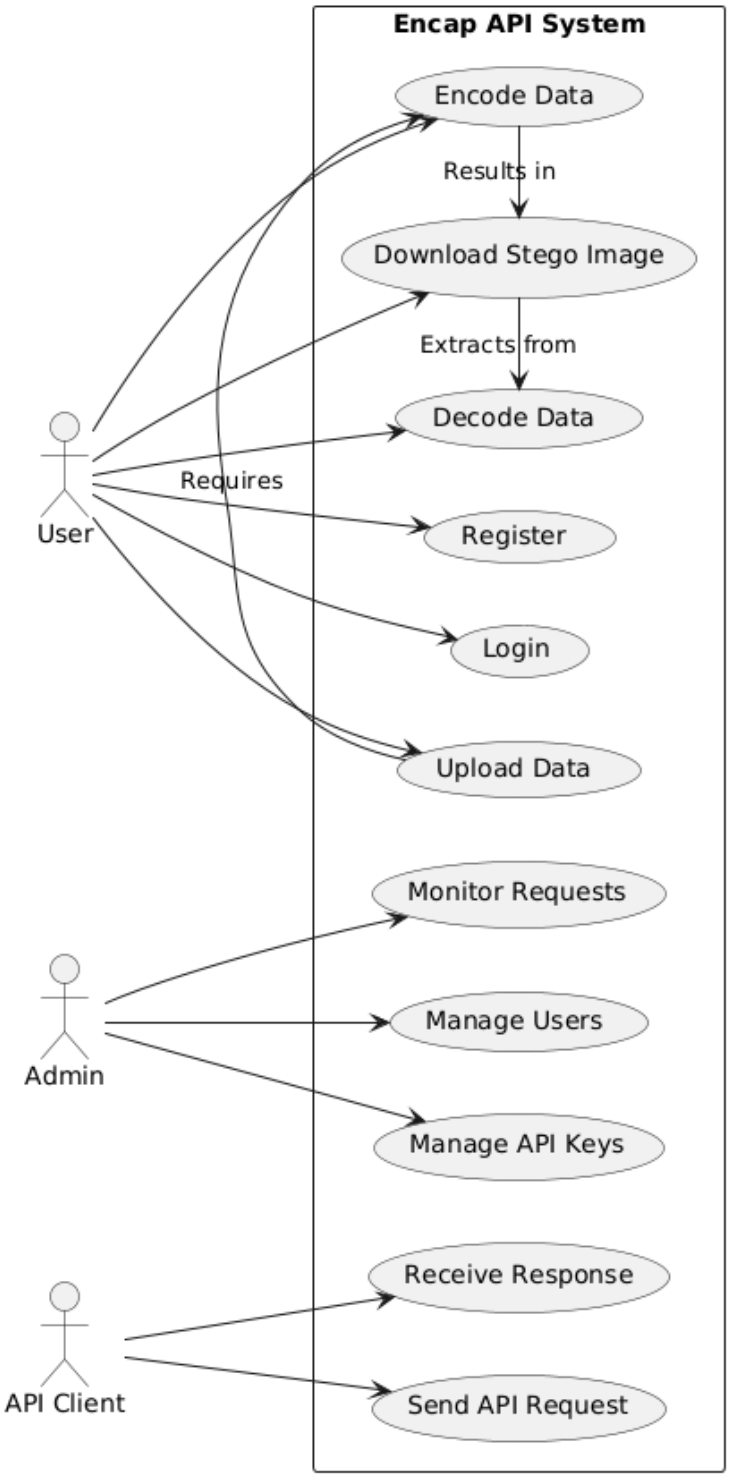
3.4 Diagrams

- **ER Diagram**



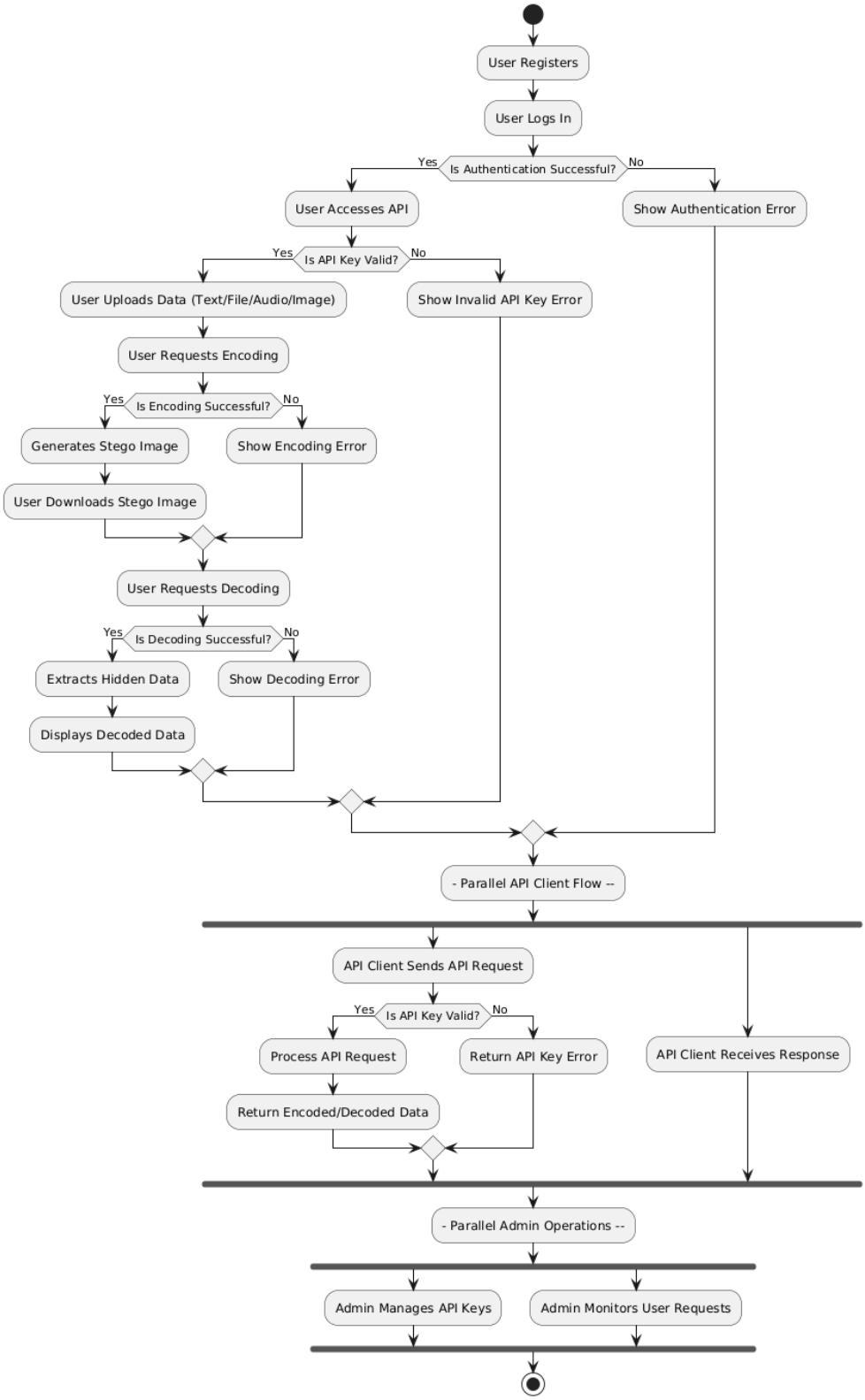
(Fig 1.1 ER Diagram **)**

- **Use Case Diagram**



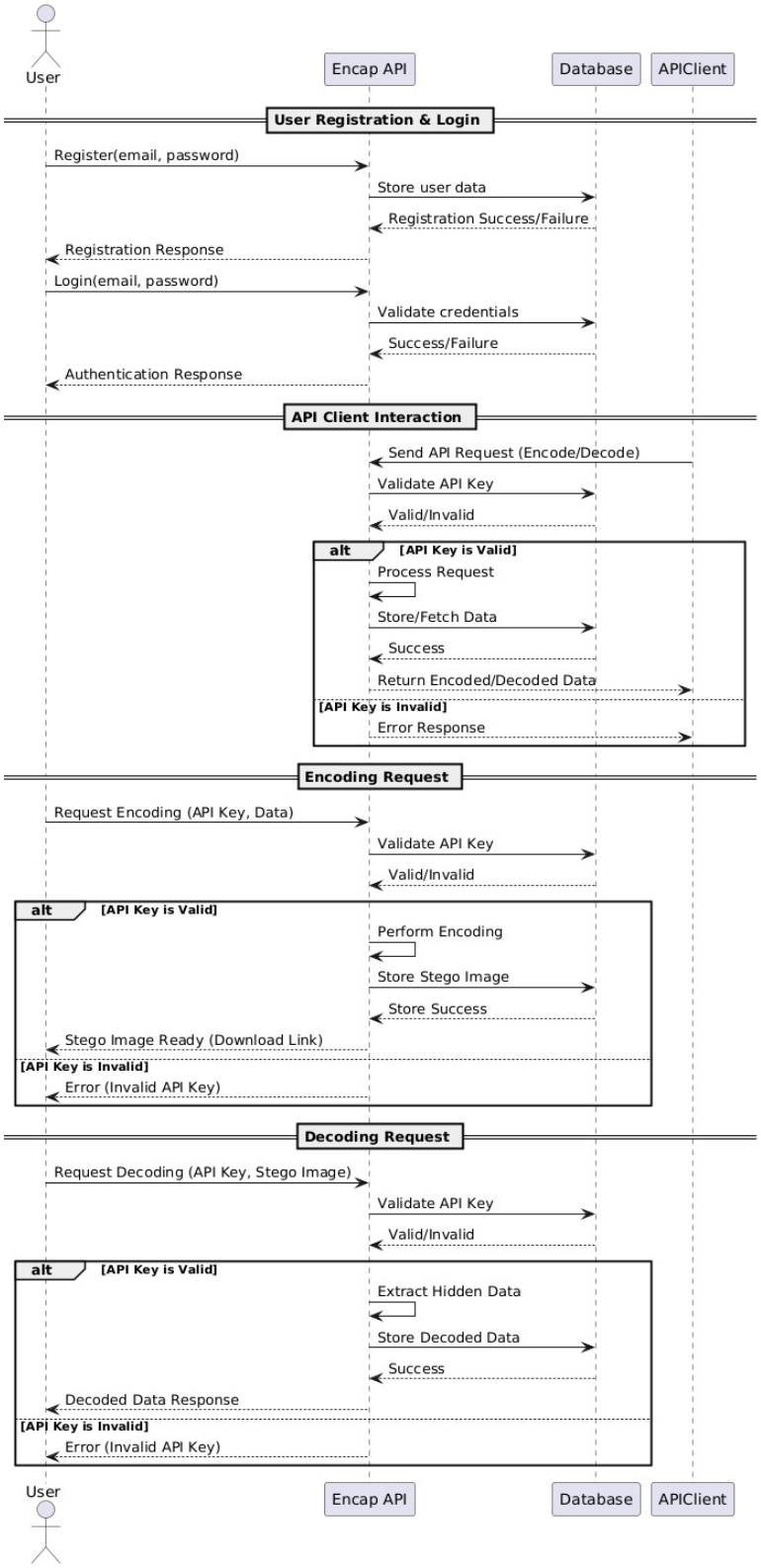
(Fig 1.2 Use Case Diagram)

- **Activity Diagram**



(Fig 1.3 Activity Diagram **)**

- **Flow Diagram**



(Fig 1.4 Flow Diagram)

CHAPTER IV
SYSTEM REQUIREMENTS

4.1 Software Requirements

➢ **Server-Side (Backend)**

- Operating System: Linux (Ubuntu 20.04+), Windows Server 2019+
- Programming Language: Python 3.8+
- Web Framework: Django
- Database: SQLLite/MySQL
- Image Processing: OpenCV, PIL (Pillow)
- Steganography Library: Stegano, Steganopy
- Web Server: Nginx / Apache
- Containerization (Optional): Docker

➢ **Client-Side (Frontend & Testing)**

- Operating System: Windows 10/10+, macOS, Linux
- Browser: Google Chrome, Mozilla Firefox, Edge
- Development Tools: VS Code, PyCharm
- API Testing: Postman, cURL
- Frontend (if applicable): HTML,CSS,JS

4.2 Hardware Requirements

➢ **Server Requirements (For Hosting API)**

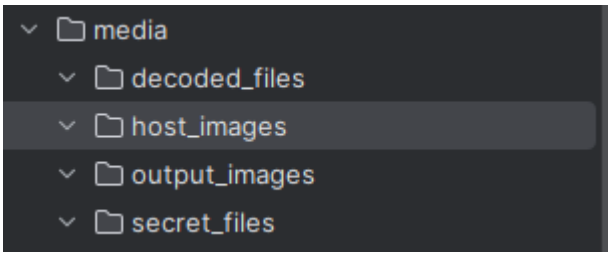
- Processor: Intel Core i5 or higher
- RAM: 8GB (Recommended: 16GB)
- Storage: 256GB SSD (Recommended: 512GB SSD)
- Network: Minimum 100 Mbps Internet Speed
- GPU (if needed for image processing): NVIDIA GTX 1050 or higher

➢ **Client Requirements (For Users & Developers)**

- Processor: Intel Core i3 or higher
- RAM: 4GB (Recommended: 8GB)
- Storage: 128GB (Recommended: 256GB SSD)
- Display: Full HD (1920x1080) recommended
- Internet: Minimum 2 Mbps

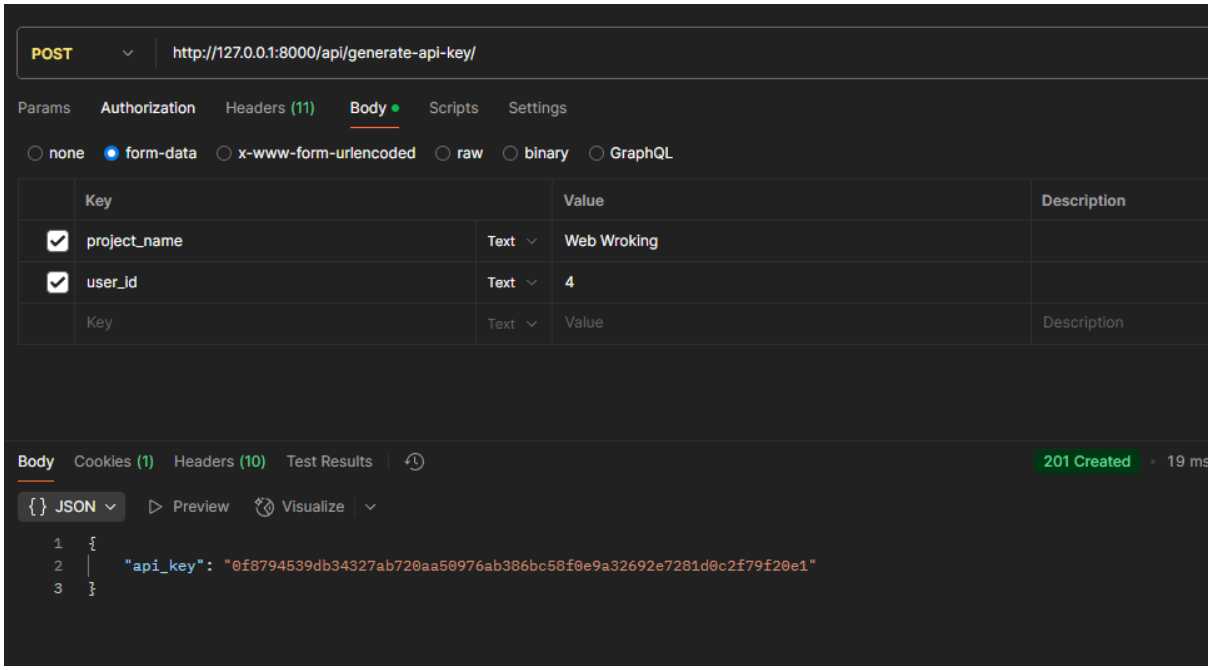
CHAPTER V
EXPECTED OUTCOMES

➢ File Structure



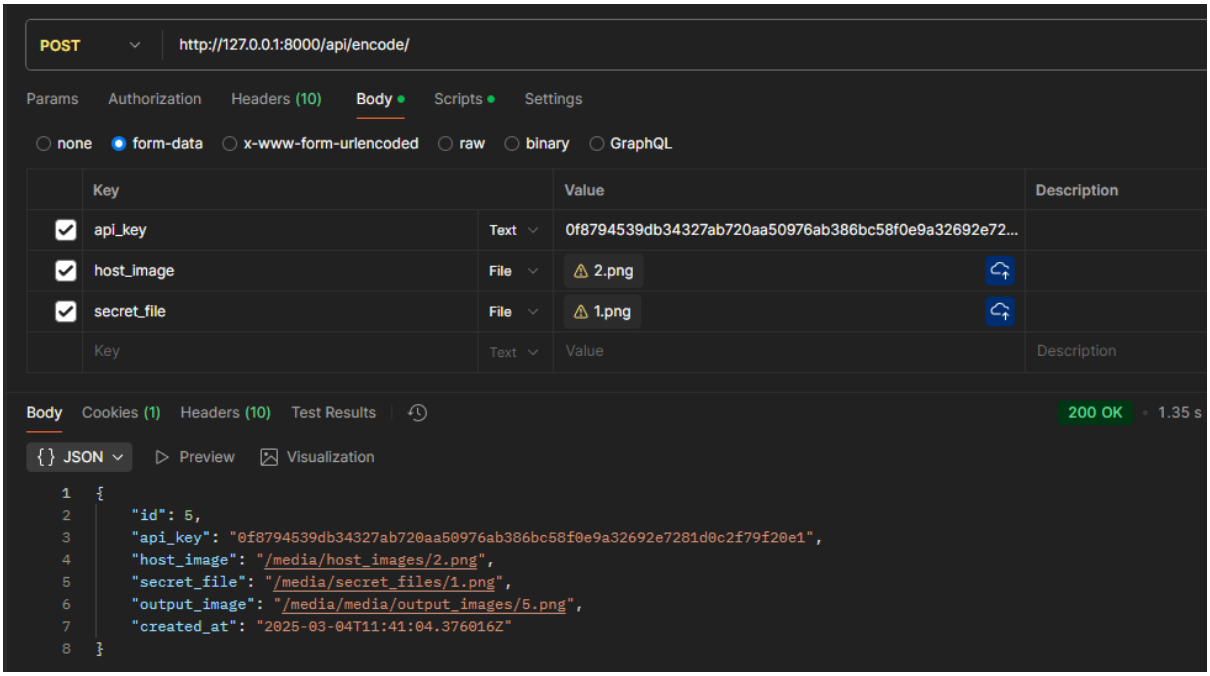
(Fig 2.1 File Structure)

➢ API Key Generation



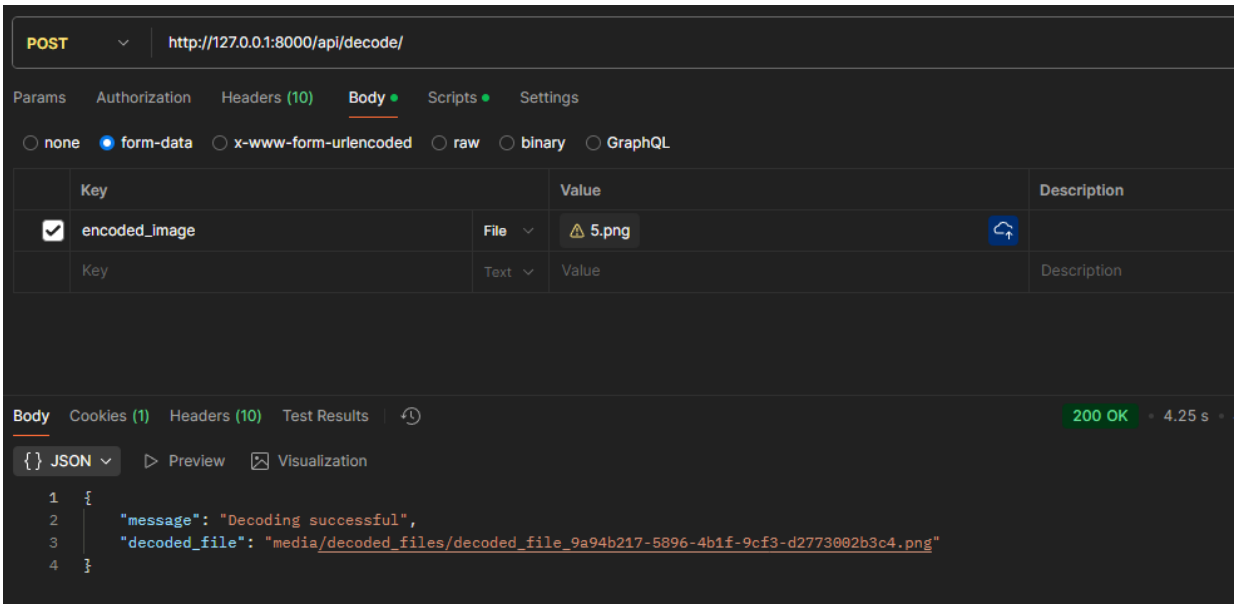
(Fig 2.2 Generate API Key)

➢ Encoding Data



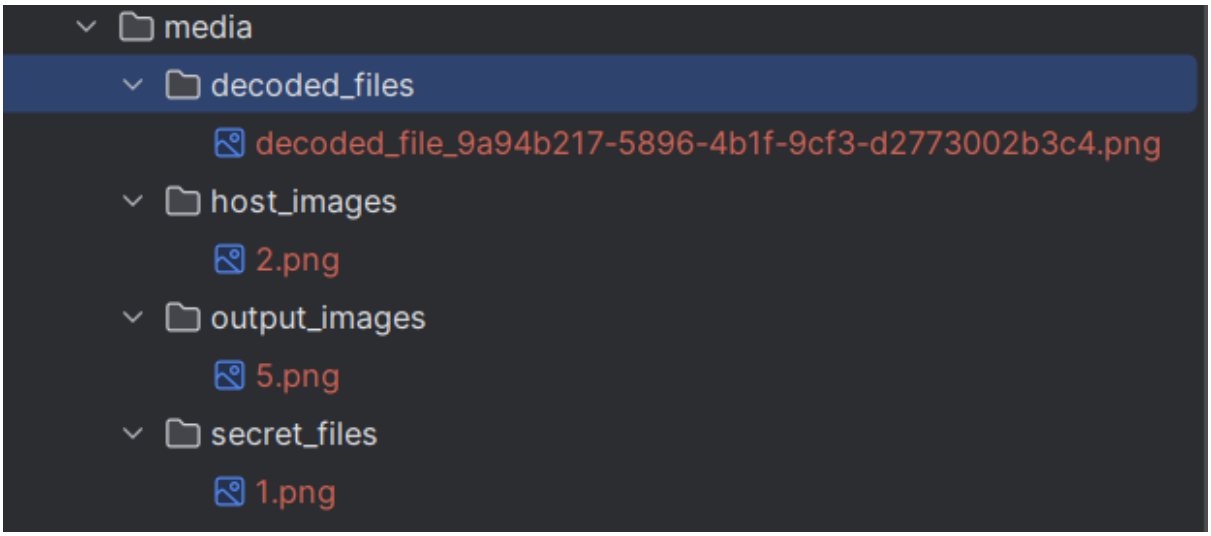
(Fig 2.3 Data Encoding)

➢ Decoding Data



(Fig 2.4 Data Decoding)

- **Decoded File Storing Structure**



(Fig 2.5 Decoded File Structure)

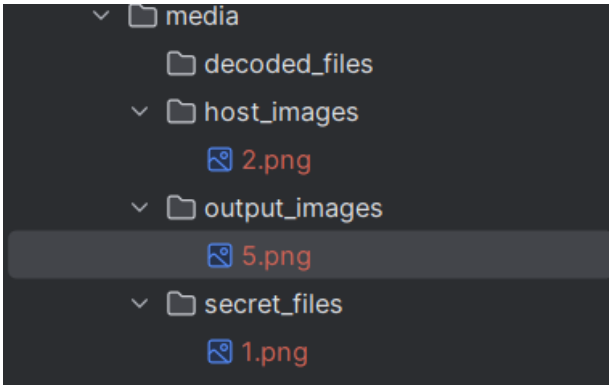
- **Secret Data**

(Fig 2.6 Secret Data)

- **Host Image**

(Fig 2.7 Host Image)

- **Output File Storing Structure**



(Fig 2.8 Output Image File Structure)

➢ Output Image

(Fig 2.9 Output Image)

CHAPTER VI
CONCLUSION & FUTURE SCOPE

6.1 Conclusion

The Encap API provides a secure and efficient solution for data encapsulation using steganography, allowing users to hide and extract information within images while ensuring confidentiality and controlled access. By integrating Least Significant Bit (LSB) steganography, AES encryption, and user authentication, the system ensures that only authorized users can encode and decode data, preventing unauthorized access.

The project successfully implements:

- User Authentication & Access Control to restrict data operations to registered users.
- Steganographic Encoding & Decoding to securely hide and extract text, audio, files, and images.
- Data Integrity & Security Measures to protect sensitive information through encryption and controlled sharing.
- Scalable API Design that enables seamless integration with various applications requiring secure data transmission.

The testing and validation processes confirm that the Encap API is reliable, efficient, and secure, making it a practical solution for applications that require covert data communication. The project demonstrates how steganography can be effectively used to enhance data security and privacy, providing a robust mechanism for secure information exchange.

6.2 Future Scope

As technology evolves, the Encap API can be enhanced with several advanced features and improvements to increase security, usability, and efficiency. Below are some key future work ideas that align with the core principles of the project:

1. Video Steganography Support

- Extend the LSB steganography technique to support video files as carrier media.
- Allow users to hide and extract data within video frames.
- Implement frame selection algorithms to optimize data embedding without compromising video quality.

2. AI-Powered Steganalysis Resistance

- Implement deep learning-based adaptive steganography, which dynamically selects optimal pixels for data hiding.
- Use adversarial AI techniques to make the hidden data undetectable against modern steganalysis tools.

3. Blockchain Integration for Security & Verification

- Store steganographic transaction logs on a blockchain to ensure tamper-proof records.
- Verify and track encoded images using smart contracts, ensuring authenticity.
- Prevent man-in-the-middle attacks by securing API requests via decentralized encryption keys.

4. Mobile Application for On-the-Go Data Hiding

- Develop a cross-platform mobile app (Android & iOS) for easy access to Encap API functionalities.
- Enable users to hide and extract data directly from their smartphones.
- Integrate with cloud storage services for encrypted data backup.

5. Cloud-Based Steganography as a Service (SaaS)

- Deploy Encap API as a cloud-based service, allowing businesses and individuals to use steganography securely online.
- Implement subscription-based plans for secure data hiding & retrieval.
- Provide RESTful API access for third-party applications to integrate steganographic functionality.

6. Support for Additional File Formats

- Extend the API to support GIF, BMP, and TIFF formats for steganography.
- Research audio-based steganography techniques to hide data within sound waves.

7. Steganography-Based Digital Watermarking

- Implement invisible watermarking techniques to protect copyrighted digital assets.
- Allow content creators to embed unique identifiers within images and videos.
- Use blockchain verification to authenticate ownership.

CHAPTER VII
REFERENCES

GitHub - Open Source Steganography Projects.
https://github.com/topics/steganography

- A collection of open-source projects on image steganography that can provide inspiration for future improvements in Encap API.

Python PIL Library for Image Processing.
https://pillow.readthedocs.io/en/stable/

- Documentation for the Python PIL (Pillow) library used in image processing for steganography.

OpenCV Library (for Image Manipulation).
https://docs.opencv.org/

- Official OpenCV documentation, which is useful for handling image processing tasks in Encap API.

Flask Web Framework Documentation.
https://flask.palletsprojects.com/

- Flask framework documentation for API development and implementation of Encap API.

JSON Web Token (JWT) for Secure Authentication.
https://jwt.io/introduction/

- Explains the JWT authentication mechanism, which is used for user authentication in Encap API.

PyCryptodome Library for AES Encryption.
https://pycryptodome.readthedocs.io/en/latest/

- Python's cryptography library used for encryption before embedding data in images.

Django REST Framework (if used instead of Flask).
https://www.django-rest-framework.org/

- Provides a structured way to build secure APIs for web applications.