National College of Ireland

Configuration Manual

MSc Research Project
Cyber Security

Nikhil Adhikrao Yadav
Student ID: X22187693

School of Computing
National College of Ireland

Supervisor: Prof. Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nikhil Adhikrao Yadav
Student ID:	X22187693
Programme:	MSc in Cyber Security **Year:** 2023-24
Module:	Practicum
Lecturer:	Prof. Michael Pantridge
Submission Due Date:	12/08/2024
Project Title:	Password Storage protection using Audio Steganography with AES Encryption
Word Count:	1178 **Page Count:** 10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Nikhil.A.Yadav

Date: 12/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Nikhil Adhikrao Yadav
X22187693

1 Introduction

This manual contains a brief description of the architecture of the Flask-based audio steganography application, which includes AES encryption, hashing, and steganography. effectively encrypts and decrypt password embedded in audio streams.

2 Hardware Configuration

A. Operating system: Windows >=10
B. Processor: Intel >=i5
C. System Compatibility: 64-bit
D. Hard Disk: 1 TB
E. RAM: >=4 GB

3 Software Configuration

3.1 Python 3.10

Python is one of the most popular languages to code that is quite easy for any level programmer to understand and work with. The readability of the language is high; it uses fewer lines of code to express concepts as compared to most other languages not just sparing time but also reducing the chances of making a mistake., supports multiple paradigms including procedural, object oriented, and functional programming and there are many libraries and frameworks for specific tasks ranging from web development to data analysis and artificial intelligence. This flexibility and robustness have made Python one of the most sought-after languages in various fields including software engineering.

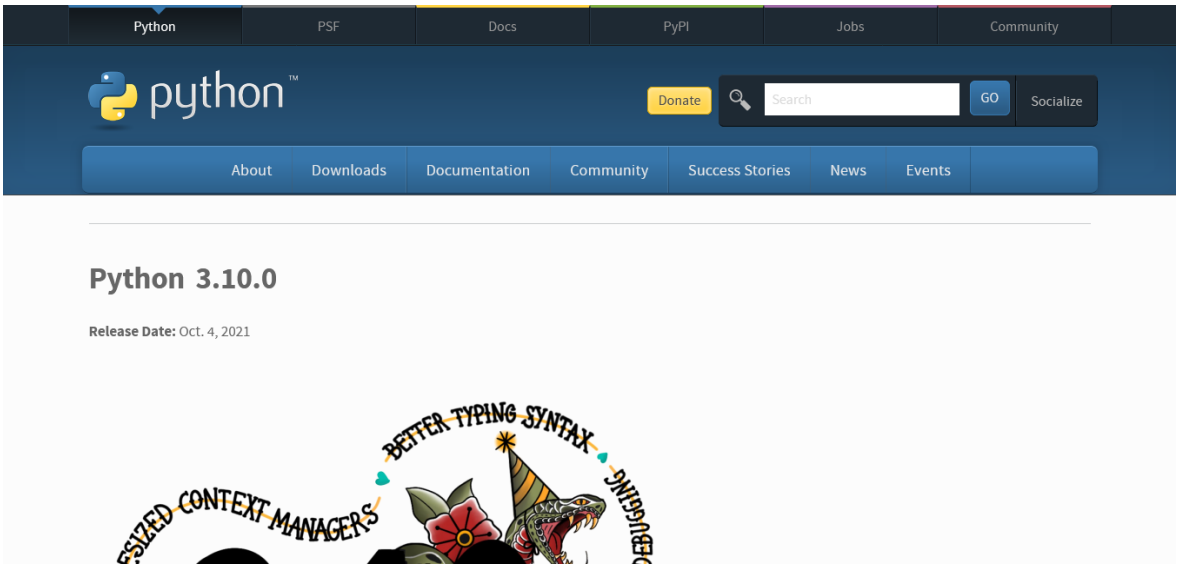

Figure 3.1: Python application

3.2 Visual Studio Code (VS Code)

Microsoft Visual Studio Code is a versatile yet light-weighted source code editor, which is supported cross-platform. It may be noted that it can offer sturdiness to a vast number of programming languages with the help of extensions, which it is famous for. Some of the facilities that VS Code has include code intelligence, debugging, terminal, and Git support, which enables easy development. Spread across its vast marketplace, developers can design their productivity with different extensions and themes for codes. Thanks to the very engaged community and continuous updates, VS Code continues to be the go-to solution for developers who prefer a fresh and fast working environment (Visual Studio Code, 2023).

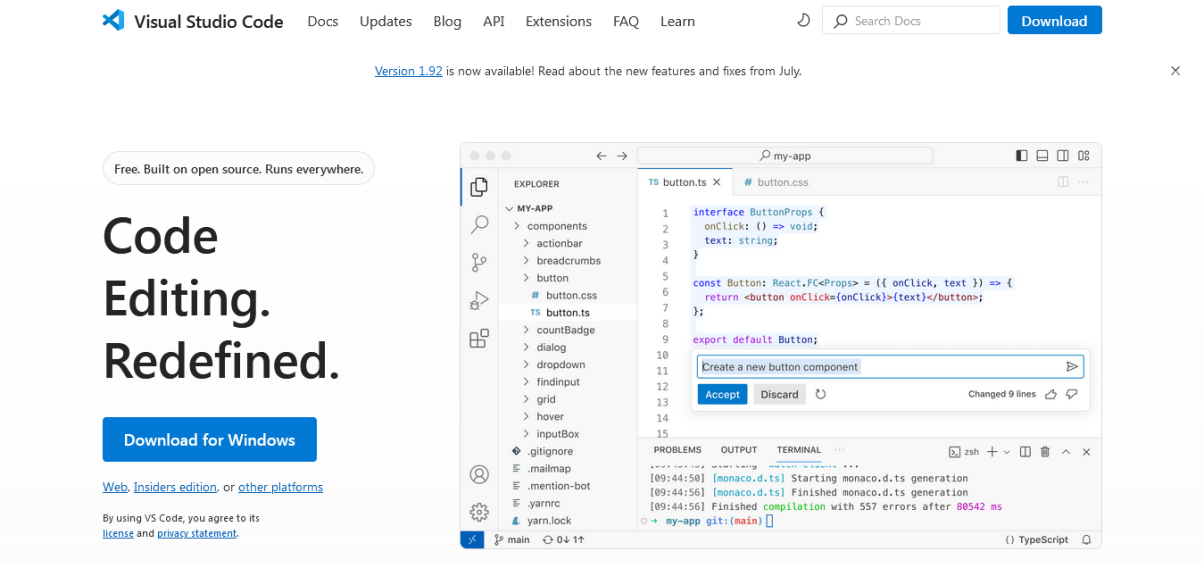

Figure 3.3: Visual Studio Code Application

4 LIBRARIES OVERVIEW

- **os**: Provides a way to interact with the operating system, allowing file and directory manipulation, environment variable access, and more.
- **wave**: Used for reading and writing WAV audio files, enabling audio data manipulation for steganography.
- **time**: Used for time-related functions, including measuring the duration of operations.
- **scrypt**: Implements the scrypt password-based key derivation function to securely hash passwords with a salt.
- **random**: Provides functions for generating random numbers, useful for generating random keys or salts.
- **base64**: Used for encoding binary data into a base64 representation, often for safe transmission of binary data as text.
- **pickle**: Serializes and deserializes Python objects to and from byte streams, used for saving and loading objects.
- **smtplib**: Implements the Simple Mail Transfer Protocol (SMTP) client, allowing for sending emails.
- **numpy**: Provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on them.
- **pandas**: Offers data structures and data analysis tools, including DataFrames for handling and analyzing data efficiently.
- **seaborn**: Provides a high-level interface for drawing attractive and informative statistical graphics based on Matplotlib.
- **pprint**: Pretty-prints Python data structures, making them easier to read.
- **email**: Includes modules for managing email messages, including composing and sending emails.
- **datetime**: Supplies classes for manipulating dates and times, used for timestamping events and measuring durations.
- **matplotlib**: A plotting library for creating static, animated, and interactive visualizations in Python.
- **cryptodome.cipher**: Part of the PyCryptodome library, used for cryptographic operations like AES encryption and decryption.
- **cryptodome.random**: Provides secure random number generation for cryptographic purposes.
- **cryptography.hazmat**: Contains low-level cryptographic primitives and utilities, such as padding and cipher algorithms.
- **flask**: A micro web framework for building web applications, used for creating and managing routes and handling web requests.

5 Usage

Initialization: Set up file paths and Flask app.
File Handling: Receive and save the audio file uploaded by the user.
Encryption: Hash the input message and encrypt it using AES.
Steganography: Encode the encrypted message into the WAV file.
Timing: Measure the time taken for encryption, encoding, decoding, and decryption.
Decryption: Decode the message from the audio file and decrypt it.
Rendering: Display results on the web interface, including timing and success messages.

6 Project Implementation

In this figure website interface is shown below

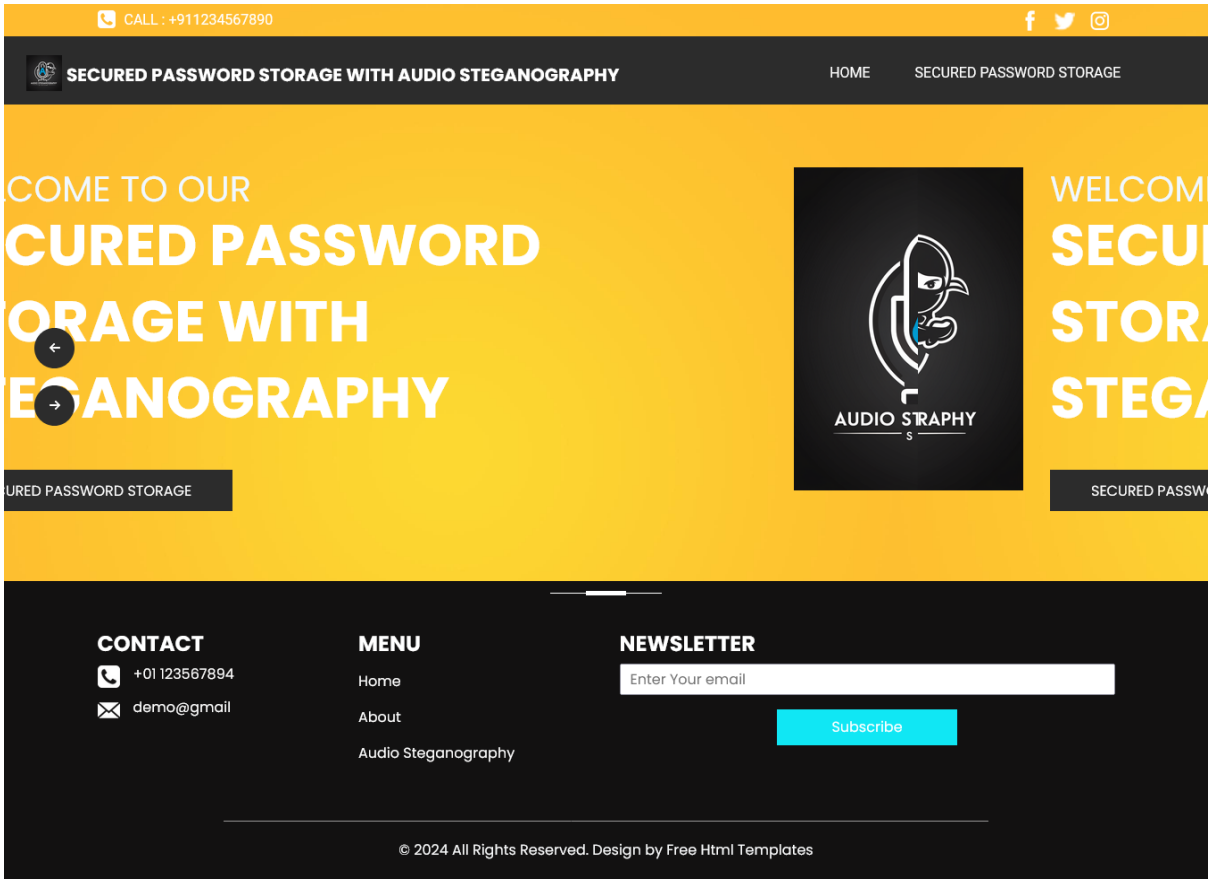

Figure 6.1: Web Application Interface

In this figure tab is shown where user can upload the audio file and the text password which will encrypt in once user will click on submit tab.

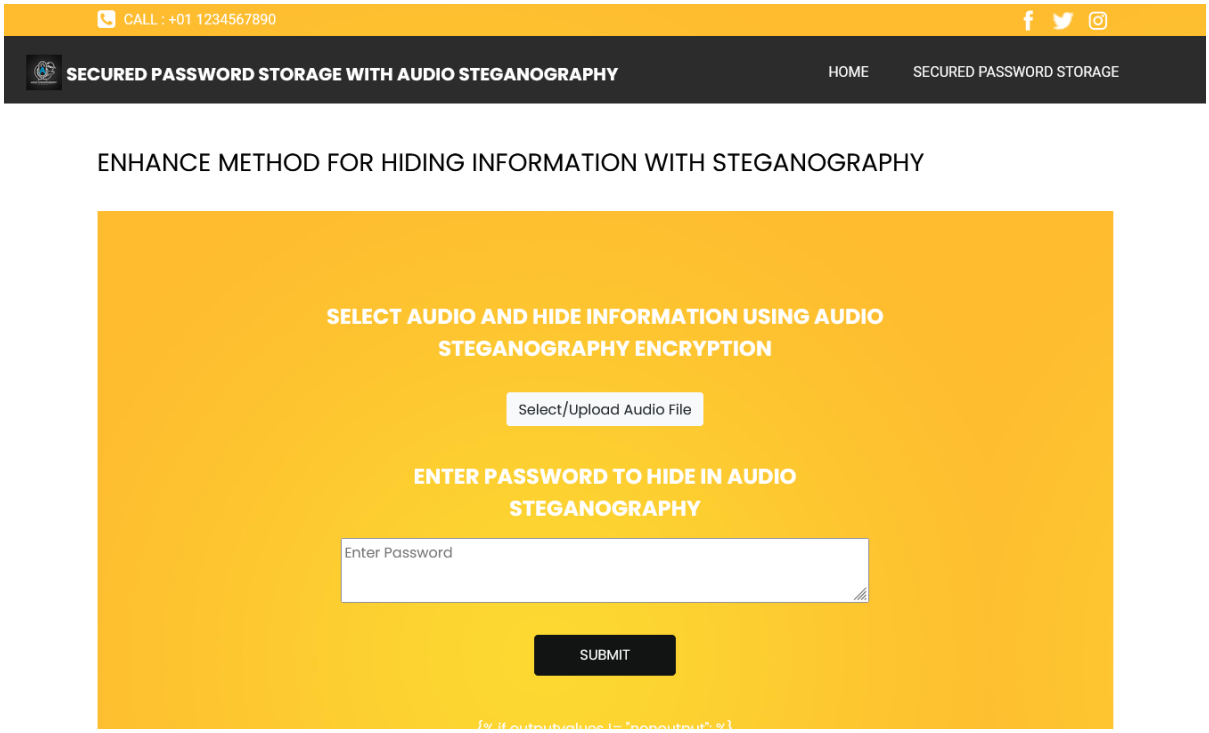

Figure 6.2: Interface for Audio Interface

In the following figure code design to check avalanche score

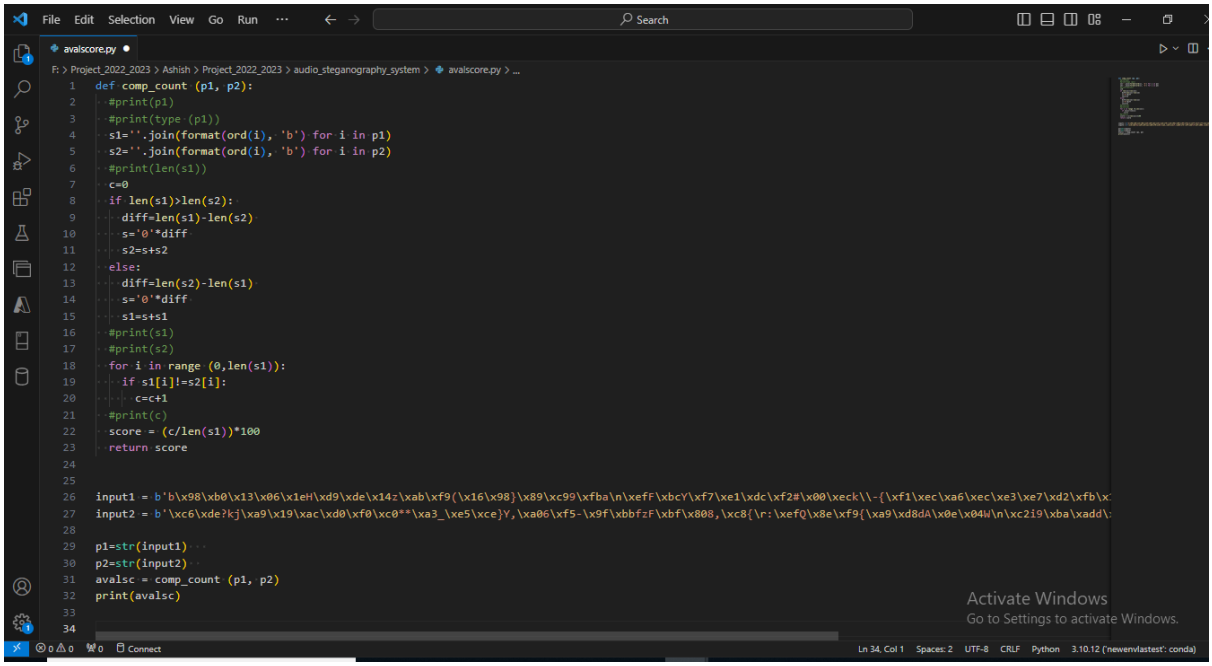

Figure 6.3: Avalanche Score Code

For testing purpose in the following figure audio file is selected names Audio 1 (file_example_WAV_1MB.wav) and as password ‘Test@12345$’ is given and this is the first example

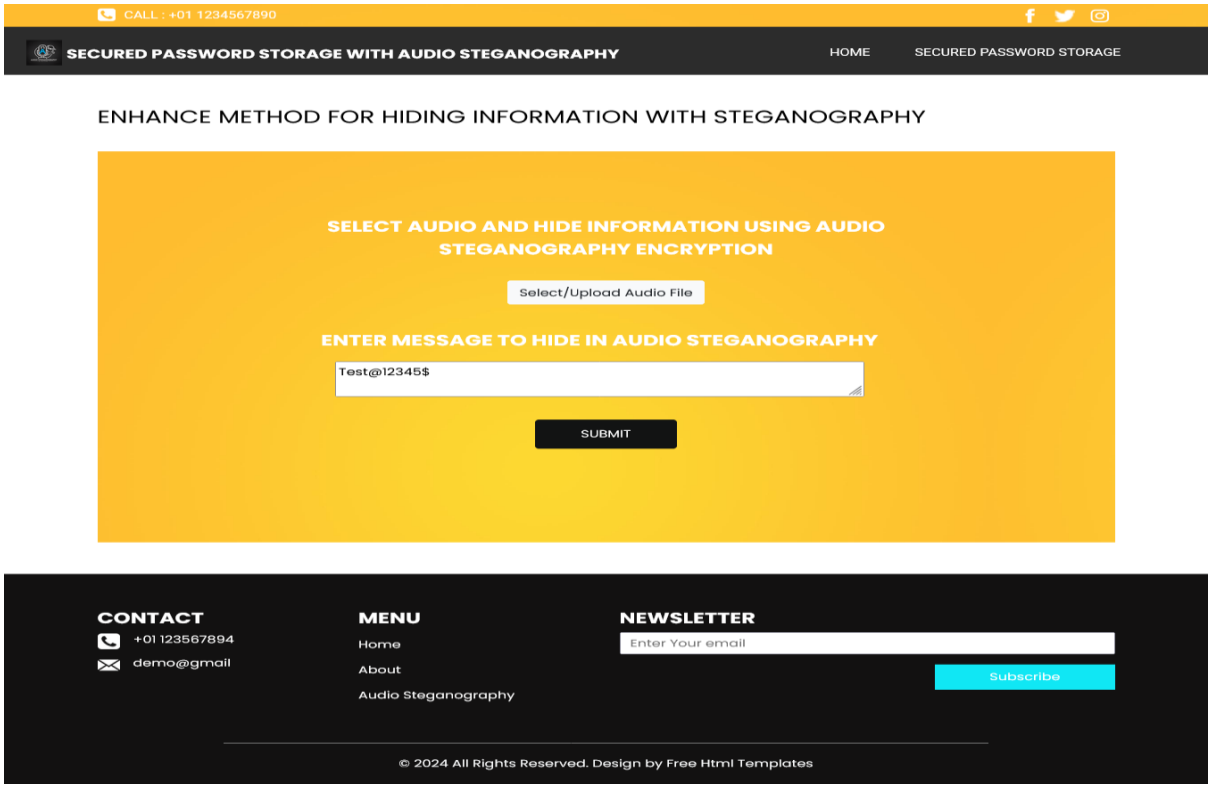

Figure 6.4: User Interface while giving inputs.

In this figure we get encrypted hash value cypher text & decrypted as well with their timing.

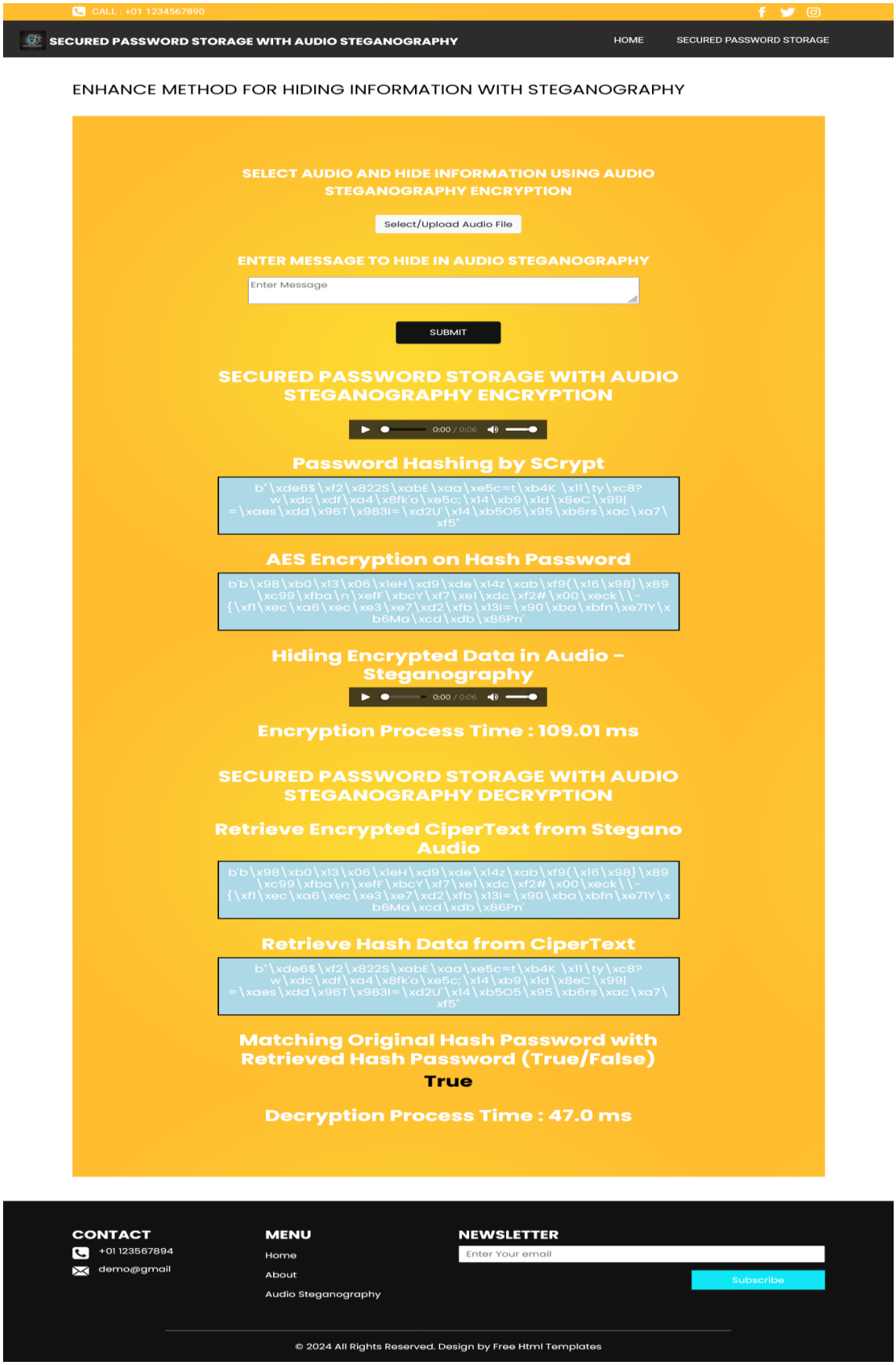

Fig 6.5: Interface after getting all results.

In this second example we change the password slightly to check performance as well as accuracy of our model. Here is changed password - Teap@12345%

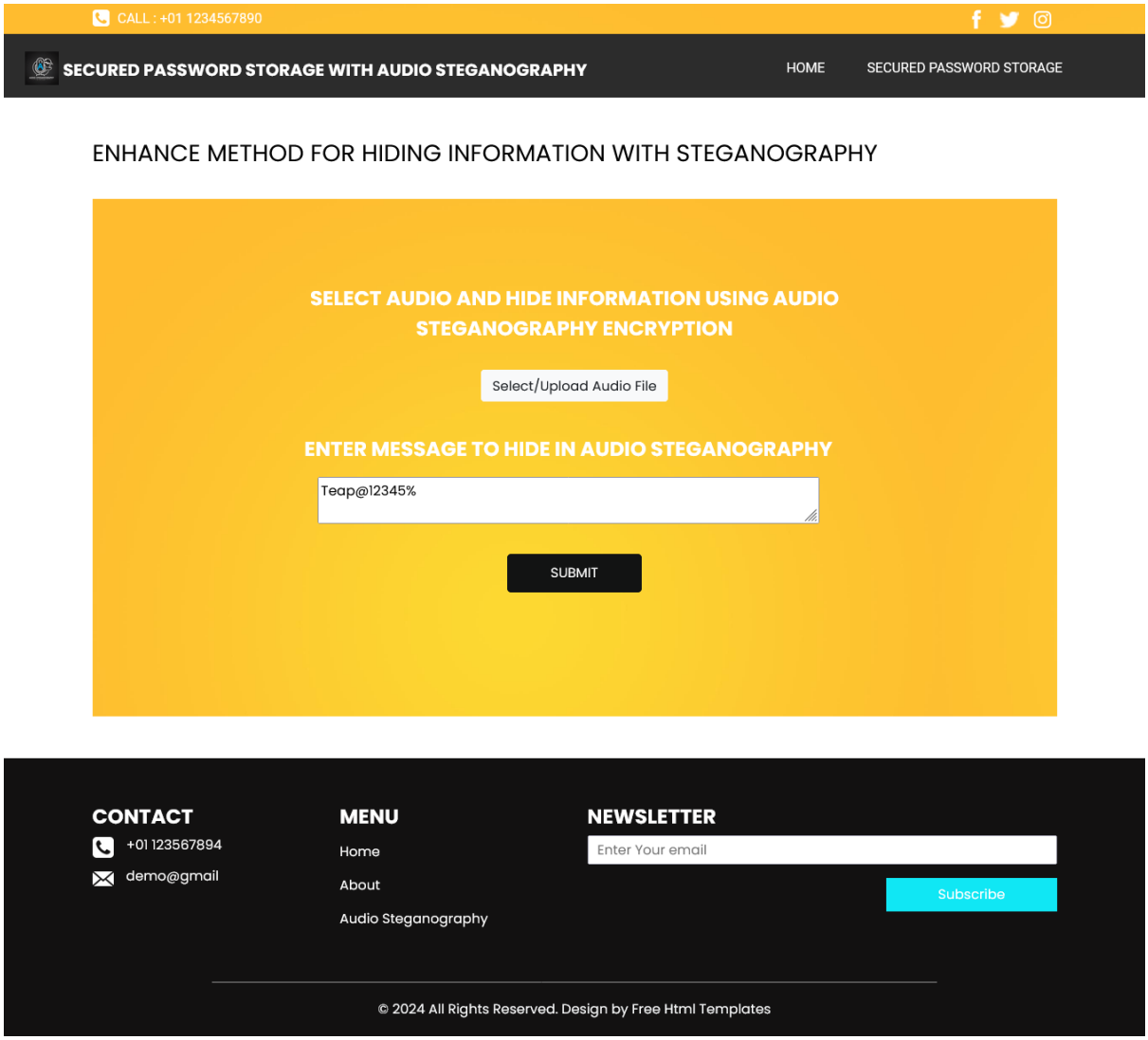

Fig 6.6: Interface while giving second inputs

In this following figure we get result of second example.

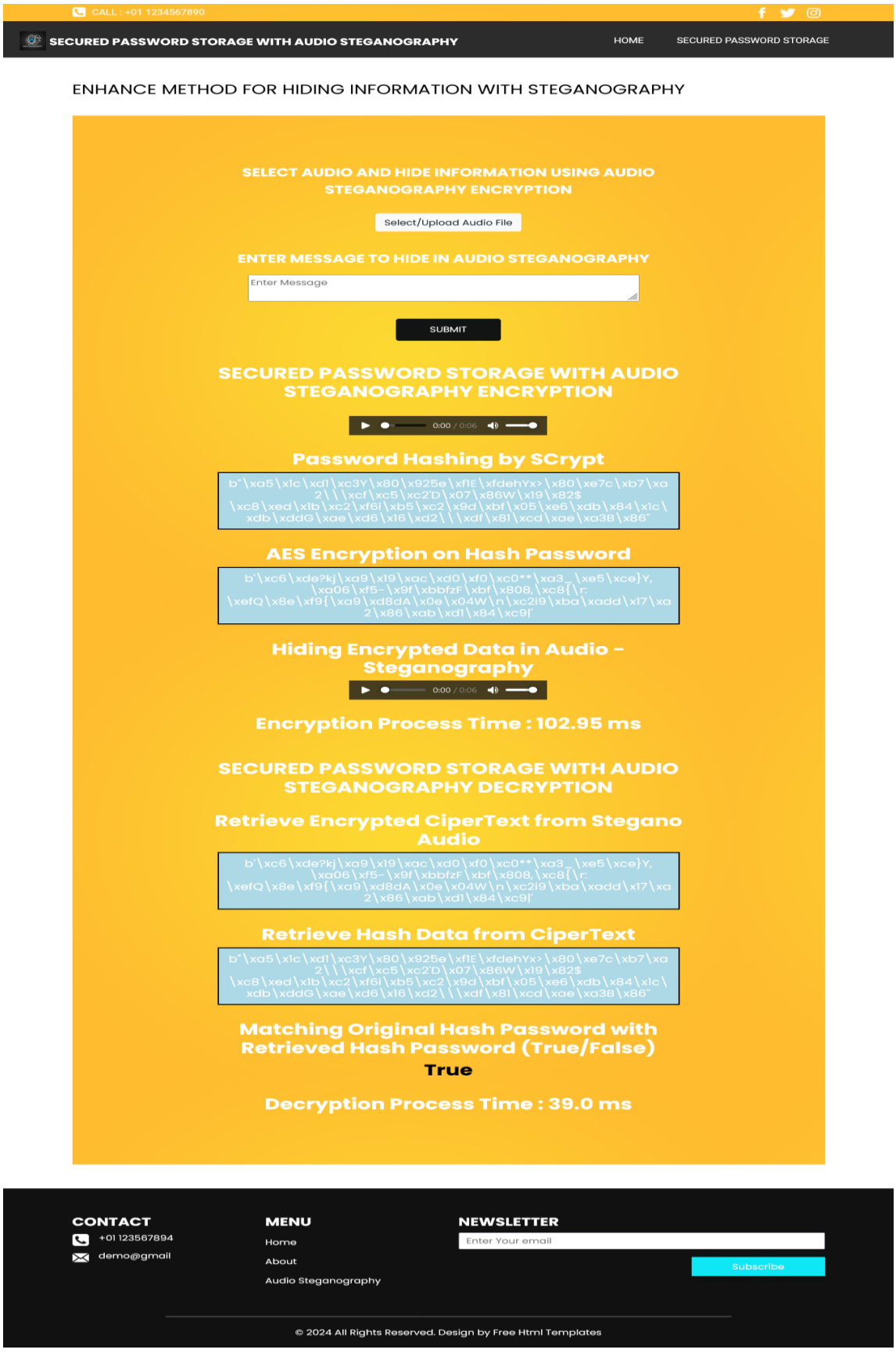

Fig 6.7 Interface after getting second result

This is output we get after comparing hash values of both results, we get Avalanche Score - 51.40845070422535

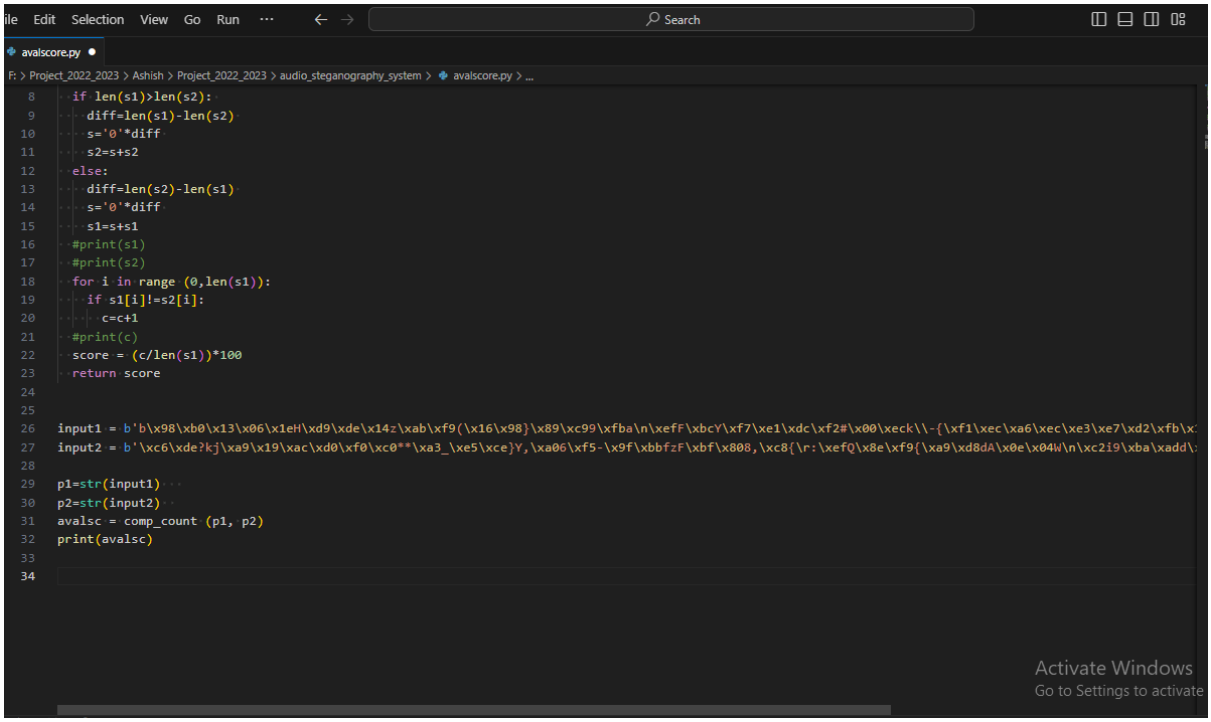

Fig 6.8: Avalanche Score

This is the snapshot of overall comparison of avalanche score with various passwords as well as with different sizes of audio files.

A	B	C	D	E	F	G	H	I	J
Audio File Name	Original Password	Cipertext	Enc Process Time(ms)	Dec Process Time(ms)	Changed Password	Cipertext	Enc Process Time(ms)	Dec Process Time(ms)	Avalanche Score
file_example_WAV_1MG.wav	Test@12345$	b'b\x98\xb0\x13\x	109.01	47	Teap@12345%	b'\xc6\xde?kj\xa9\x	102.95	39	51.4084507
file_example_WAV_1MG.wav	tangoCharlie1121	b"\x95\xde\x94J\\\	98.01	46.99	tengOCharlie1213	b"\x95y\xb7\x02\xd	113	44	48.65269461
file_example_WAV_1MG.wav	dummy@abc	b"\x9aJh\x85\xa4\	116.01	48.01	tummp@bcd123	b"\xa3Q\xd3\xc8S\x	95.01	46	51.41122214
file_example_WAV_2MG.wav	Test@12345$	b"x'\xd4\xa6U\xc8	134.01	73.01	Teap@12345%	b"F\xa0\xc8\xbf\xb8	125.97	72	50.23359769
file_example_WAV_2MG.wav	tangoCharlie1121	b'1o\xea\xa0\x9d\'	132.01	71.01	tengOCharlie1213	b"\xcf\xcb\x90O\x1	141.01	92.01	49.43014129
file_example_WAV_2MG.wav	passwOrd111$	b'\xc3F\x82\x8d~+'	133.01	76.01	pasawOrd121$	b"\xe9\x98\x01`\x13	157.01	82.01	48.23172628
file_example_WAV_10MG.wav	Test@12345$	b"\xc2b\x81c\x81*	419.03	292.02	Teap@12345%	b'-\xa7\x7fA\xdc\x0	428.04	271.02	51.62002946
file_example_WAV_10MG.wav	tangoCharlie1121	b'\xd7\x99\xf9R\xa	387.03	280.02	tengOCharlie1213	b'8[\xa3\x13\xbe,\x	465.08	314.02	50.36603221
file_example_WAV_10MG.wav	passwOrd111$	b'\x85\x88\xcd\x0	433.03	309.02	pasawOrd121$	b'S\xbc\x90\xdb\xa	405.48	297.03	49.80930587

Fig. 6.9: Avalanche Score Comparison Table.

References

1. Python (2019). Python For Beginners. [online] Python.org. Available at: https://www.python.org/about/gettingstarted/.
2. Visual Studio Code (2023). *Documentation for Visual Studio Code*. [online] code.visualstudio.com. Available at: https://code.visualstudio.com/docs.