National College of Ireland

Configuration Manual

MSc Research Project

Muhammad Ahssan Sajjad
Student ID: 23156767

School of Computing
National College of Ireland

Supervisor:Liam Mccabe

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: .....Ahssan Sajjad.....

Student ID: .....23156767.....

Programme: .....MSC in Cybersecurity..... **Year:** .....

Module: .....Thesis Research Project.....

Lecturer: .....Liam Mccabe.....

Submission Due Date: .....15/09/2024.....

Project Title: .....Configuration Manual.....

Word Count: ..... **Page Count:** .....

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: .....*Ahssan Sajjad*.....

Date: .....11/08/2024.....

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

Muhammad Ahssan
Sajjad

23156767

1 Introduction

This configuration manual aims at giving a step-by-step procedure on the physical layout of the Network-Specific Data Decryption Tool for Enhancing User Profile Security, its actual setup into a working system and the way it should be controlled and maintained. The project entails the development of a web application and several features include user sign on, encryption of data and logging. The main target audience of this document is system administrators, developers, and IT specialists who are to install the system on a server.

2 System overview

The system is a web base application, which has a goal of authorizing users' accounts, managing data through encryption, or logs. You get a front end part (HTML/CSS) along with the backend part python Flask.

2.1 Hardware & Software requirements

Hardware:

- A server or Virtual Machine that has a minimum of two cores on the CPU and 4 GB of RAM.
- At least 20 GB disk space is to be maintained.
- A reliable internet connection.

Software:

- Python 3. x
- Flask Web Framework [1]
- Other prerequisites that include but not limit ed to; flask , werkzeug, cryptography and other Python related libraries.
- A web server such as the Apache or the Nginx [2]
- Ssl certificates for security in communication [3]
- An optional database if persistent data storage is required such as SQLite, MySQL, or PostgreSQL [4].

3 Installation

3.1 Pre-installation

Ensure that Python 3. Indeed an application x is installed on the server. This, you can verify it by using the command; python3 –version [5].

- Install a virtual environment for the project by running:Install a virtual environment for the project by running:

```
python3 -m venv venv
```

- Activate the virtual environment:

On Linux/Mac: source venv/bin/activate

On Windows: venv¥Scripts¥activate

- Install the required dependencies for Python language execution by using following command replacing actual library name at 'name':

  Pip install <name>

3.2 Installation procedure

Add environment variables like database username, password, keys, etc. These should be done in a . env file [6].

If your setup uses a database, configure the database (if you are using LAMP stack than you might need to install 'mysql-server' also) and run migrations to create tables which are required to run the application (if you are using 'laravel' then you can use 'php artisan migrate' command to make a migration) [7]. Start the Flask application by running:

Pip install flask

Python app.py

3.3 Post installation

Web Server Configuration:

Setting up a web server such as Apache or Nginx to host the Flask application. Make sure HTTP communication is using S for SSL to ensure safe HTTP transmission [8].

Testing:

It is recommended to check the installation by opening the created web application in a browser. Check all paths and features, making special attentiveness to whether the app is working properly or not.

4 Configuration

4.1 Initial Configuration

Configure config. py:

Modify config.py used to establish such parameters as secret keys, database uniform resource identifier, and debug settings [9].

Authentication Configuration:

Set the user authentication rules inside auth. py. Make sure that the passwords' hashing and users' sessions' management are configured properly [10].

Data Encryption:

Establish data security measures on encryption. py. This involves key definition and selection of suitable encryption algorithms [11].

4.2 Advanced Configuration

Logging Configuration:

Use logging_util. ble py to set up logging schemes. They left the ways of constructing log file paths, log rotation policies, and log levels [12].

Machine Learning Model Integration:

If your application implies working with machine learning, make sure that the trained model is launched in the right way. You will now have to update the paths of your models in your code and possibly configure any more predictions settings [13].

5 Glossary

1. **Flask:** WSGI based web application framework for Python that is relatively light weight [1].
2. **SSL:** Secure Sockets Layer a common protocol used in Internet secure connection between web server and a browser [3].
3. **WSGI:** Web Server Gateway Interface, a standard relating to the construction of Python web applications [5].

6 References

1. Grinberg, M. (2018). Flask web development: Developing web applications with Python (2nd ed.). O'Reilly Media.
2. Apache Software Foundation. (2024). Apache HTTP Server Project. https://httpd.apache.org/
3. SSL.com. (2024). SSL Certificates. https://www.ssl.com/
4. Chappell, D. (2019). Understanding SQL and MySQL: Harnessing the power of SQL and NoSQL databases (1st ed.). Addison-Wesley Professional.
5. Python Software Foundation. (2024). Python 3.x documentation. https://docs.python.org/3/
6. Flask Documentation. (2024). Configuration handling. https://flask.palletsprojects.com/en/2.0.x/config/
7. Flask-Migrate Documentation. (2024). Flask-Migrate. https://flask-migrate.readthedocs.io/en/latest/
8. Nginx. (2024). NGINX Docs. https://nginx.org/en/docs/
9. Flask Documentation. (2024). Application setup. https://flask.palletsprojects.com/en/2.0.x/config/
10. Flask-Login Documentation. (2024). Flask-Login. https://flask-login.readthedocs.io/en/latest/
11. Stallings, W. A. (2020). Cryptography and network security: Principles and practice (8th ed.). Pearson.
12. Python Documentation. (2024). Logging HOWTO. https://docs.python.org/3/howto/logging.html
13. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.
14. Axelsson, S. (1999). Research in intrusion detection systems: A survey. ACM Computing Surveys, 31(3), 264-325.