DOCUMENTATION

RAAZ ENCRYPT - SAFEGUARDING SECRETS WITH INTELLIGENT ENCRYPTION

1. INTRODUCTION

"**Raaz Encrypt**" is a secure messaging app that protects your private messages by encrypting them. It ensures that only the right people can read the messages, and only if they have the correct key. The app is easy to use, making sure your conversations stay private and secure.

2. PROJECT CONTRIBUTORS

This project was developed as part of the Bachelor of Science in Computer Science (BSCS) program at the University of Karachi.

- **Group Leader**: Syed Zayan Ali (mailto:zayanali2003@gmail.com)
- **Team Members**:
  - Yahya Arif Butt
  - Anas Shoaib
  - Hamza Wahaj
- **Supervisor**: SIR TAUSEEF MUBEEN

3. PROJECT FILES OVERVIEW

Simple explanation of what each file in the project does:

- **app.py**: This is the main file that runs the entire app. It brings all the parts of the program together and makes sure everything works smoothly. You start the app by running this file.

- **auth.py**: This file handles user accounts. It manages the process of signing up new users, logging them in, and making sure that passwords are correct. It's like the security system that checks who can use the app.
- **database.py**: This file takes care of storing all the data. It uses a database to save user information and encrypted messages. Think of it as the app's storage system, keeping everything safe and organized.
- **encrypted_messages.db**: This is the actual database file where all the important data is stored. It holds all the encrypted messages and user details securely, like a vault for your app's information.
- **messages.py**: This file is responsible for encrypting and decrypting messages. It takes a message and scrambles it so only someone with the right key can read it. This is the core of your app's security.
- **RAAZENCRYPT.jpg:** This is the images used in the app.
- **ui.py**: This file manages the user interface (UI), which is what users see and interact with. It designs the layout, buttons, and screens that make the app easy to use and navigate.
- **init.py**: This file is part of the setup that lets Python know all these files work together as one project. It helps keep the project organized.

4. REQUIRED LIBRARIES

To run "Raaz Encrypt," you'll need to install the following Python libraries:

- **tkinter**: For creating the graphical user interface (GUI).
- **sqlite3**: For managing the SQLite database (included with Python, no need to install separately).

5. HOW TO USE RAAZ ENCRYPT

1. **Setup**:
    - Make sure Python is installed on your computer.
    - Unzip the project files and open the project folder.
2. **Running the App**:

- To start the app, double-click on `app.py` or run it from your Python environment. This will launch the program.

3. **User Registration**:
    - If you're a new user, create an account by choosing a username and password. Your information will be stored securely in the database.
4. **Login**:
    - If you've already registered, simply log in with your username and password to access the app's features.
5. **Encrypting a Message**:
    - After logging in, you can type a message that you want to keep private. The app will automatically encrypt it using the Caesar cipher and store it securely.
6. **Decrypting a Message**:
    - To read a stored message, enter the correct decryption key. The app will unscramble the message and display the original text.

6. TECHNICAL DETAILS

- **Language Used**: Python
- **Database**: SQLite (`encrypted_messages.db`) for storing user data and encrypted messages.
- **Encryption Method**: The Caesar cipher, a simple yet effective way to keep messages private.
- **User Interface**: Managed by `ui.py`, which provides a simple and user-friendly experience.

7. SECURITY FEATURES

- **User Authentication**: Only registered users can access the app, and passwords are checked to ensure security.
- **Message Encryption**: Messages are encrypted to keep them safe from unauthorized access.

- **Data Storage**: All information is securely stored in a local database, ensuring that your data is protected.

8. FUTURE IMPROVEMENTS

Here are some ideas for making Raaz Encrypt even better in the future:

- **Advanced Encryption**: Implementing stronger encryption methods to enhance security.
- **User Management**: Adding features like password recovery, user roles, and activity tracking.
- **Mobile App Version**: Developing a version of the app that works on smartphones and tablets.
- **Cloud Storage**: Allowing users to store encrypted messages in the cloud for easier access from different devices.