**Hyper Chaotic Encryption Tool - Documentation**

**Project Overview**

The Hyper Chaotic Encryption Tool is a Flask-based web application that provides encryption and decryption functionalities using a hyper-chaotic algorithm. The tool is designed to be user-friendly, offering a web interface for users to input messages and encryption keys.

**Features**

1. **Encrypt and Decrypt Messages**:
   * Users can input a message and a key to encrypt or decrypt the message.
   * The result is displayed sequentially: original message, key, and the final result (encrypted/decrypted message).
2. **Download Result**:
   * Users can download the result of the operation as a .txt file.
3. **Reset Functionality**:
   * A reset button clears all inputs and results.
4. **Hover-Over Info**:
   * When users hover over the "All rights reserved" footer, additional information about the author, including name and email, is displayed.

**Technologies Used**

1. **Backend**:
   * Python
   * Flask
2. **Frontend**:
   * HTML
   * CSS
   * JavaScript (Fetch API for communication with the backend)
3. **Deployment**:
   * Render (Free-tier hosting platform)

**Project Structure**

.

├── app.py # Main Flask application

├── requirements.txt # Dependencies for the project

├── Procfile # Render deployment configuration

├── templates/

│ └── index.html # HTML file for the web interface

├── static/

│ ├── styles.css # Custom CSS (if applicable)

│ ├── script.js # JavaScript for frontend (if applicable)

└── README.md # Project documentation

**Setting Up the Project Locally**

**Prerequisites**

* Python (3.7 or later)
* Git

**Installation Steps**

1. **Clone the Repository**:
2. git clone https://github.com/shrirambansal/hyper.git
3. cd hyper
4. **Set Up a Virtual Environment**:
5. python -m venv venv
6. source venv/bin/activate # On Windows: venv\Scripts\activate
7. **Install Dependencies**:
8. pip install -r requirements.txt
9. **Run the Application**:
10. python app.py

The app will be available at http://127.0.0.1:5000.

**Deploying to Render**

**Steps to Deploy**

1. **Push to GitHub**:
   * Ensure your code is on a GitHub repository.
2. **Create a Render Account**:
   * Go to [Render](https://render.com/) and log in or create an account.
3. **Deploy Web Service**:
   * Click **New > Web Service**.
   * Connect your GitHub repository.
   * Set the following configurations:
     + **Environment**: Python
     + **Build Command**: pip install -r requirements.txt
     + **Start Command**: gunicorn app:app
4. **Trigger Deployment**:
   * Render will automatically build and deploy your app.
   * Access the deployed app via the URL provided by Render.

**API Endpoints**

**/process - POST**

Handles encryption and decryption requests.

**Request Body:**

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| action | String | encrypt or decrypt |
| message | String | The message to encrypt or decrypt |
| key | String | Encryption key (float between 0 and 1) |

**Response:**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| result | String | The encrypted or decrypted message |
| error | String | Error message (if any, e.g., invalid input) |

**Frontend Design**

**index.html**

* **Input Fields**:
  + Message text area for the input message.
  + Input box for the encryption key.
* **Buttons**:
  + Encrypt, Decrypt, Reset.
  + Download result functionality.

**CSS Styling**

* Styled for a clean and responsive design.
* Box shadow and margins for a centered container.

**JavaScript**

* Handles user interactions and communicates with the backend using the Fetch API.

**Common Errors and Debugging**

**Error: gunicorn: command not found**

* Ensure gunicorn is added to requirements.txt.
* Re-deploy after updating dependencies.

**Error: refusing to merge unrelated histories**

* Use the --allow-unrelated-histories flag while pulling remote changes:
* git pull origin main --allow-unrelated-histories

**Future Improvements**

1. Add more encryption techniques for enhanced functionality.
2. Implement user authentication for personalized encryption histories.
3. Improve UI with additional styling and animations.

**Author**

**Shriram Bansal**  
*Email*: [shrirambansal13@gmail.com](mailto:shrirambansal13@gmail.com)  
*GitHub*: [github.com/shrirambansal](https://github.com/shrirambansal)/

This concludes the documentation for the Hyper Chaotic Encryption Tool. For further queries or contributions, feel free to reach out!