**Implementation of DES Encryption Algorithm**

Supervisor Assignment by,

Dr. Narendran Rajagopalan, Polamreddy Dinesh Reddy

Associate Professor, (CS22B1043)

NITPY.

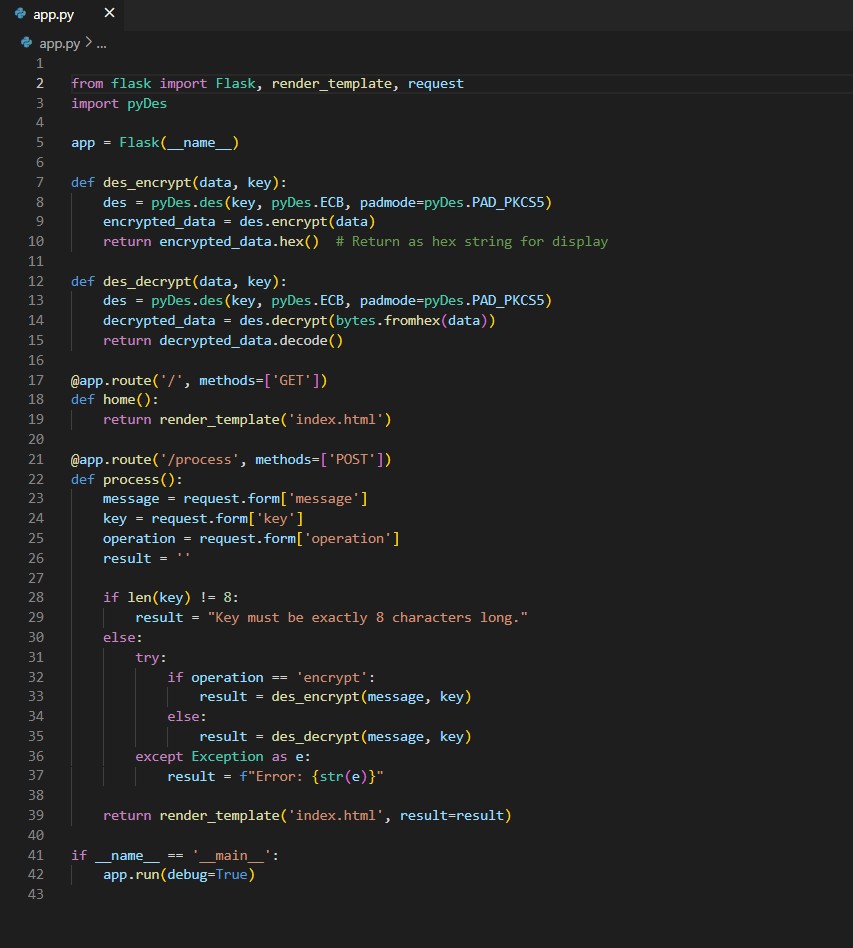
**Introduction:**

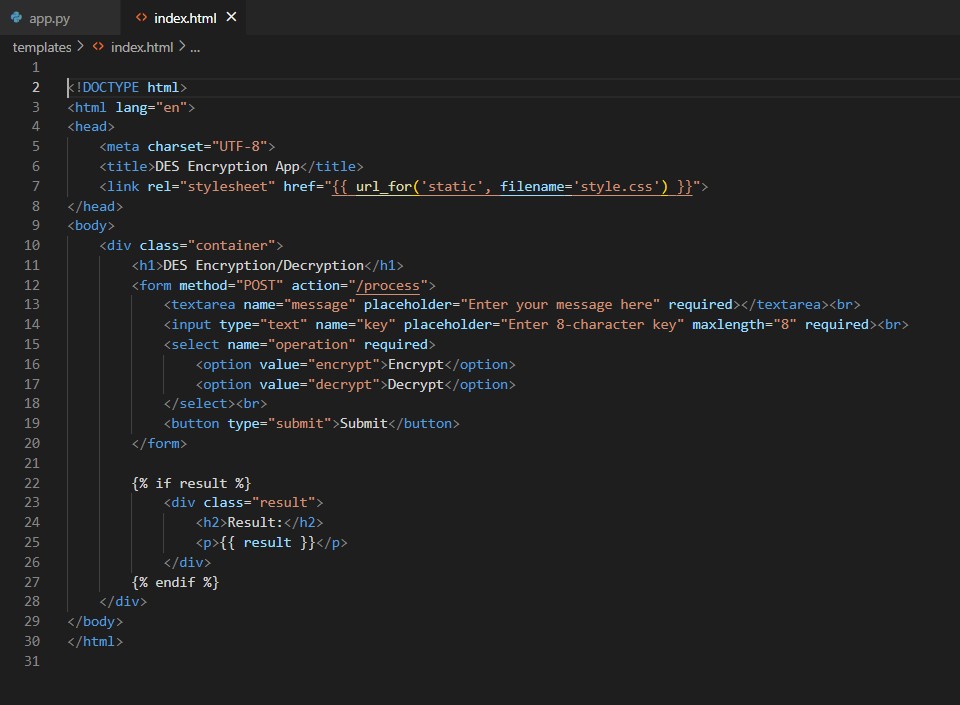
In today’s world, information security plays a critical role in protecting data from unauthorized access and tampering. One of the earliest and most well-known encryption techniques is the Data Encryption Standard (DES), a symmetric-key algorithm for encrypting and decrypting information.

This project presents a simple web application that demonstrates the DES encryption and decryption process using Python (Flask framework) for backend logic and HTML/CSS for the frontend interface.

Users can input a message, provide an 8-character key, and choose whether to encrypt or decrypt their message, helping them understand how classical encryption algorithms work in practice.

**Implementation:**

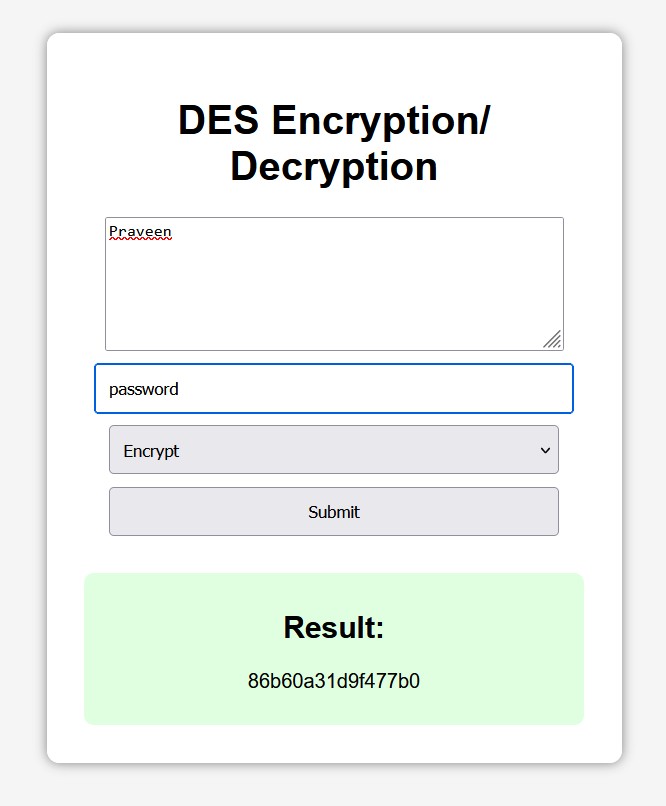






**Output:**





**Explanation:**

This project is built with three main components:

1. **Frontend (HTML/CSS)** 
   * A clean, simple webpage is designed to accept:
     + The **message** to encrypt or decrypt
     + An **8-character key** (as required by DES) o The operation: **Encrypt** or **Decrypt**
   * It provides a user-friendly interface styled with CSS to make the interaction smooth.
2. **Backend (Python Flask)** 
   * **Flask**, a lightweight Python web framework, handles the form submission.
   * After receiving the input, it uses the **pyDes** library to perform DES encryption or decryption:
     + **Encryption**: Converts plain text into a secure hex string.
     + **Decryption**: Converts the hex string back into readable plain text.
   * The backend then sends the result back to the frontend for display.
3. **Encryption Logic** 
   * The DES algorithm requires a **fixed 8-character key**.
   * Messages are padded automatically to match DES block size requirements.
   * Outputs are presented in a readable **hexadecimal format** after encryption, making it easier for users to copy and store.

**Conclusion:**

This project successfully demonstrates the implementation of a classic DES encryption system inside a web application, combining web development (HTML, CSS) with cybersecurity techniques (encryption algorithms).

It helps users interactively understand how messages can be encrypted and decrypted using symmetric-key cryptography.

While DES is no longer recommended for high-security systems due to its shorter key length and vulnerability to brute-force attacks, it remains a fundamental learning tool for cryptography basics.

This project serves as a foundation for further explorations into more advanced encryption algorithms like AES or RSA and introduces how web applications can be used for real-time security operations.