AES Encryption Web Application

1. Project Title

AES-Based Secure Text Encryption Web Application

---

2. Project Overview

This project is a simple web-based application that demonstrates how data can be protected using **AES (Advanced Encryption Standard)** encryption. The system allows users to enter plain text, encrypt it securely using AES, and then decrypt it back to its original form.

The main goal of the project is to show a practical implementation of symmetric encryption in the field of **Information Security**, focusing on simplicity, correctness, and clear understanding rather than complexity.

---

3. Objectives of the Project

- Demonstrate how AES encryption works in practice.
- Protect text data from unauthorized access.
- Show the process of encryption and decryption using the same secret key.
- Build a simple and clear web interface for user interaction.
- Apply theoretical security concepts in a real working system.

---

4. Why AES?

AES (Advanced Encryption Standard) is one of the most widely used encryption algorithms in the world.

Reasons for choosing AES: - Strong security and resistance to brute-force attacks. - Used by governments and security organizations. - Faster and more secure than older algorithms like DES. - Supports different key sizes (128, 192, 256 bits).

In this project, **AES-128** is used for simplicity and efficiency.

---

5. System Architecture

The system is divided into four main components:

1. **encrypt.py** – Handles AES encryption logic.
2. **decrypt.py** – Handles AES decryption logic.
3. **app.py** – Flask backend that connects encryption logic with the web interface.
4. **index.html** – Frontend web page for user interaction.

6. Technologies Used

- **Programming Language:** Python
- **Web Framework:** Flask
- **Encryption Library:** PyCryptodome
- **Frontend:** HTML + Bootstrap
- **Encryption Algorithm:** AES (Symmetric Encryption)

7. Encryption Process (How It Works)

1. The user enters plain text in the web interface.
2. The text is sent to the Flask backend.
3. The encryption function uses:
4. AES algorithm
5. Secret key
6. Secure encryption mode
7. The output is encrypted binary data.
8. The encrypted data is encoded using **Base64** to make it readable and transferable.
9. The encrypted text is displayed to the user.

8. Decryption Process

1. The user enters the encrypted Base64 text.
2. The backend decodes it back to binary data.
3. The same AES secret key is used.
4. The original plain text is restored.
5. The decrypted text is displayed.

9. Role of Base64 Encoding

Base64 is not encryption. It is used to: - Convert binary encrypted data into readable text. - Allow encrypted data to be safely displayed and transferred in web applications.

AES provides security, while Base64 provides compatibility.

10. Security Discussion

- AES is a symmetric encryption algorithm, meaning the same key is used for encryption and decryption.
- If the secret key is kept secure, the data remains protected.
- AES is resistant to brute-force attacks due to large key space.
- The project demonstrates confidentiality, which is a core principle of information security.

13. Conclusion

This project successfully demonstrates how AES encryption can be implemented in a real web application. It combines theoretical security concepts with practical coding, making it a strong example of applying cryptography in information security systems.

The system is simple, secure, and effective for educational purposes and project discussion.