Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 4
з курсу: «*Безпека програмного забезпечення*»

Виконав:
студент 4-го курсу,
групи ТВ-21
Цвігун Богдан

Київ 2025

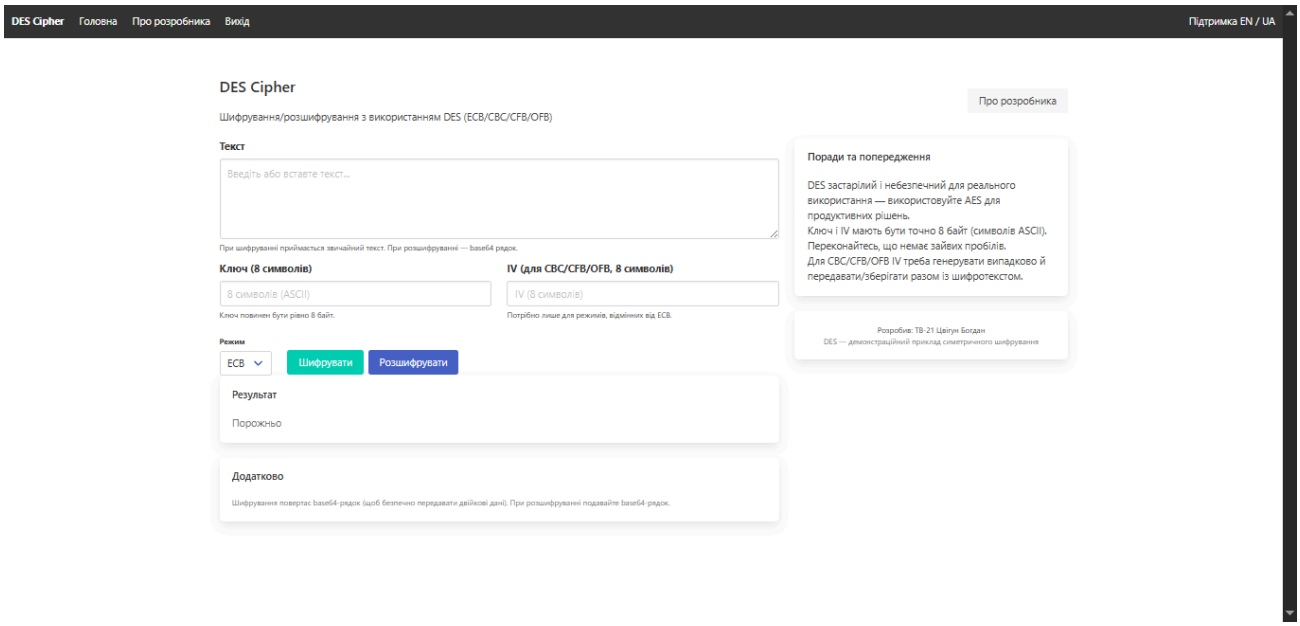
Практична робота № 4

Завдання:

1. Розробіть інтерфейс криптографічної системи для шифрування за допомогою DES з використанням всіх можливих режимів.
2. Ознайомтесь з описом класів CryptographicServiceProvider і CryptoStream бібліотеки .NET Framework.
3. Реалізуйте шифрування DES, використовуючи класи .NET Framework.
4. Виконайте тестування роботи системи.

Хід виконання:

Для розробки веб-додатку було використано фреймворк Flask, який забезпечує ефективну взаємодію між серверною частиною та інтерфейсом користувача. Інтерфейс реалізовано з використанням сучасних веб-технологій, зокрема CSS-фреймворку Bulma, що надає адаптивний і естетичний дизайн, а також шаблонізатора Jinja2, який відповідає за динамічне формування вмісту HTML-сторінок.

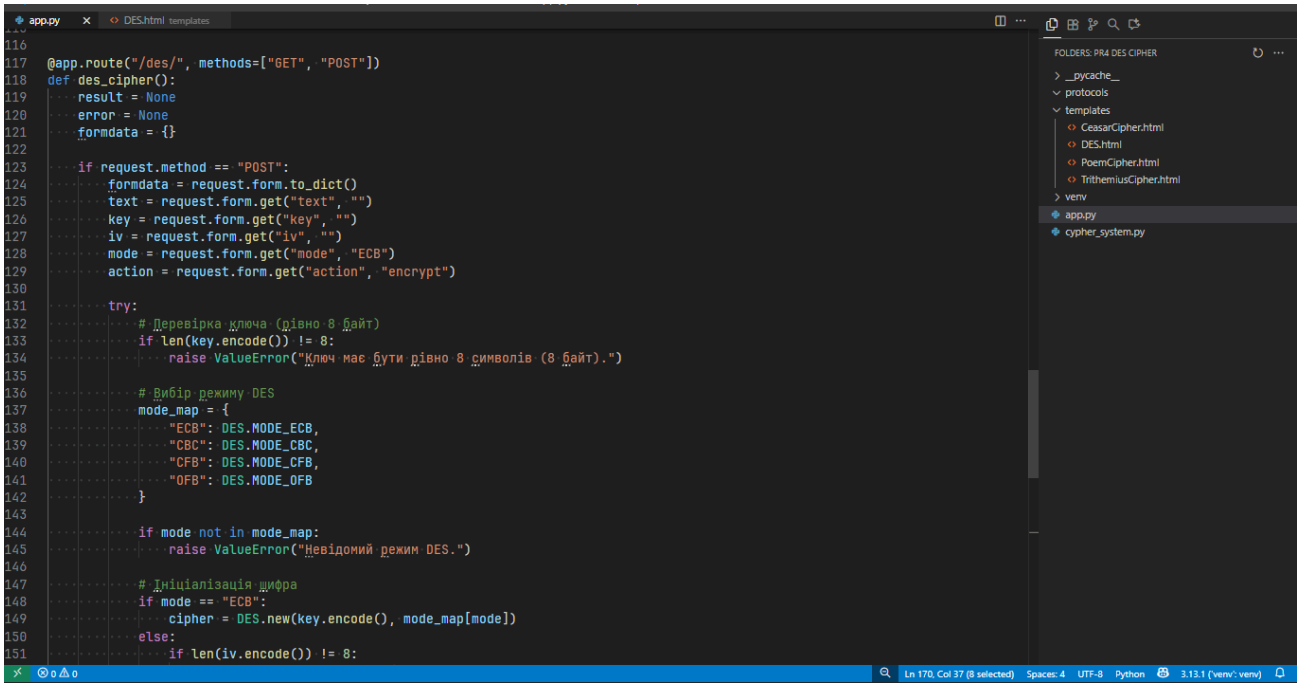


Шифр DES (Data Encryption Standard) є класичним алгоритмом симетричного блочного шифрування, розробленим у 1970-х роках компанією IBM та затвердженим Національним інститутом стандартів і технологій США (NIST) у 1977 році як федеральний стандарт. DES працює з блоками даних розміром 64 біти та використовує ключ довжиною 56 біт, за допомогою якого виконується послідовність із 16 раундів перестановок і підстановок.

Основний принцип DES полягає у поєднанні операцій заміни (S-box) та перестановки (P-box), що забезпечує як дифузію, так і плутанину — ключові властивості надійного шифрування.

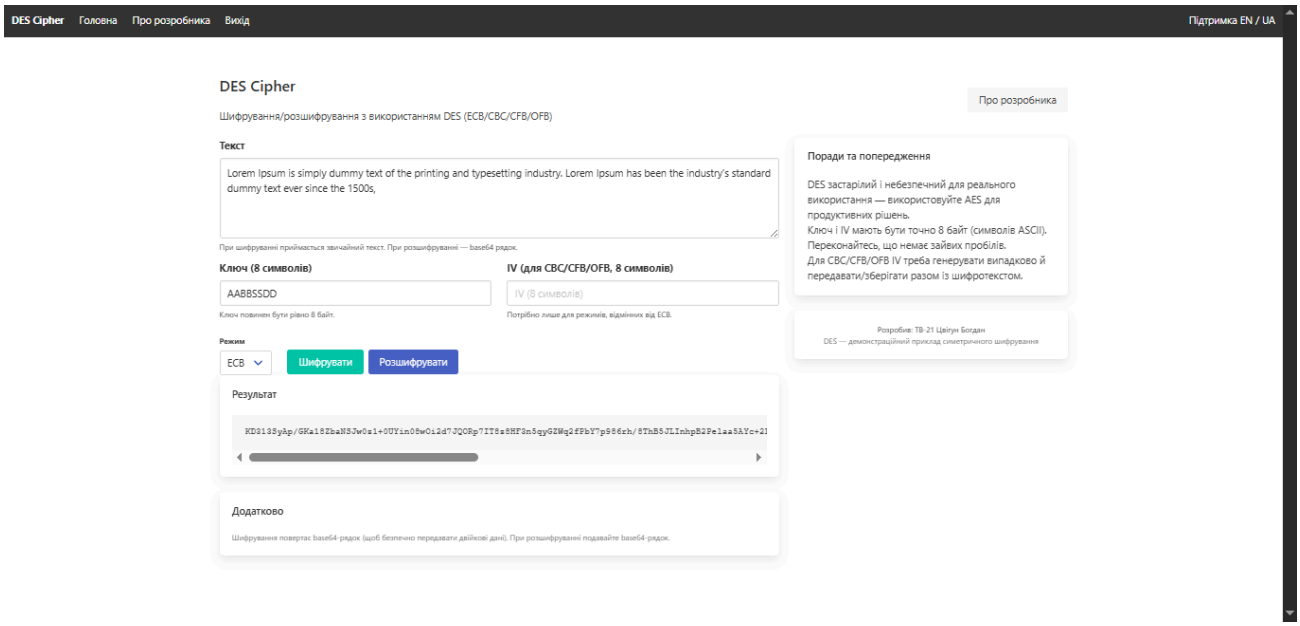
Попри історичну важливість і довготривале використання у фінансових та урядових системах, DES нині вважається криптографічно застарілим, оскільки його ключ можна зламати за допомогою сучасних обчислювальних засобів методом повного перебору.

Для підвищення безпеки було створено модифікацію Triple DES (3DES), у якій алгоритм виконується тричі з різними ключами. Проте сьогодні для практичного використання рекомендується застосовувати сучасні стандарти, зокрема AES (Advanced Encryption Standard), який забезпечує значно вищий рівень криптостійкості.



```python
@app.route("/des/", methods=["GET", "POST"])
def des_cipher():
    result = None
    error = None
    formdata = {}

    if request.method == "POST":
        formdata = request.form.to_dict()
        text = request.form.get("text", "")
        key = request.form.get("key", "")
        iv = request.form.get("iv", "")
        mode = request.form.get("mode", "ECB")
        action = request.form.get("action", "encrypt")

        try:
            # Перевірка ключа (рівно 8 байт)
            if len(key.encode()) != 8:
                raise ValueError("Ключ має бути рівно 8 символів (8 байт).")

            # Вибір режиму DES
            mode_map = {
                "ECB": DES.MODE_ECB,
                "CBC": DES.MODE_CBC,
                "CFB": DES.MODE_CFB,
                "OFB": DES.MODE_OFB
            }

            if mode not in mode_map:
                raise ValueError("Невідомий режим DES.")

            # Ініціалізація шифра
            if mode == "ECB":
                cipher = DES.new(key.encode(), mode_map[mode])
            else:
                if len(iv.encode()) != 8:
```

Повний вихідний код проєкту доступний у репозиторії, що буде додано до завдання.

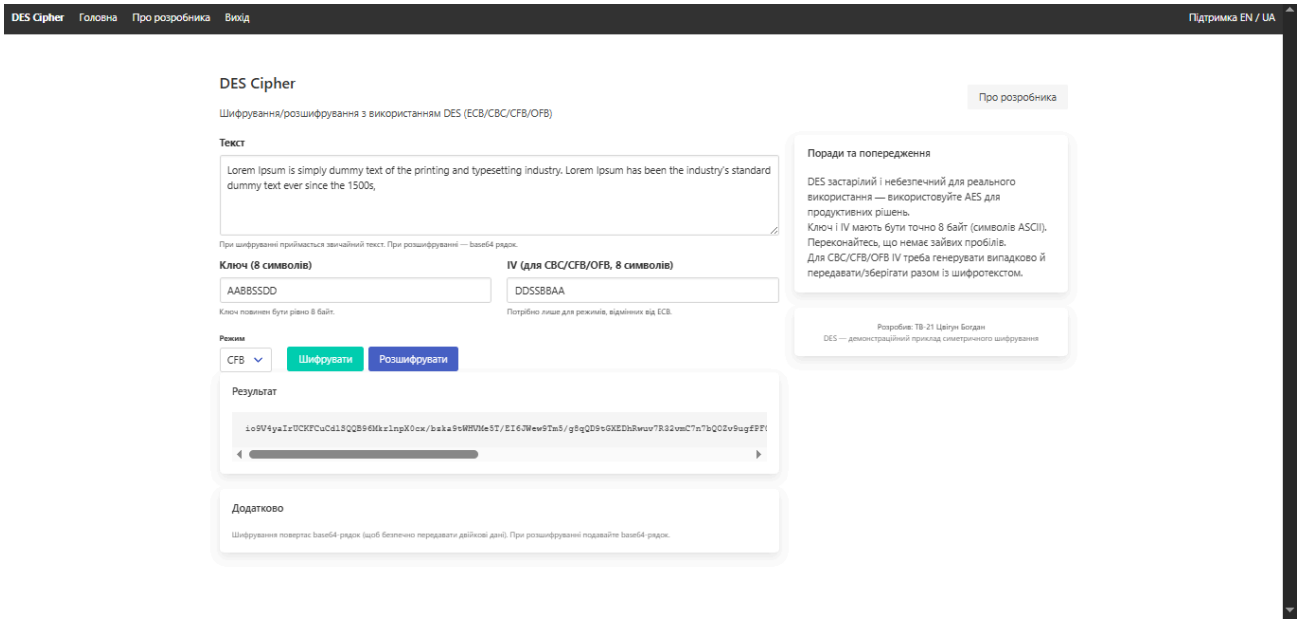


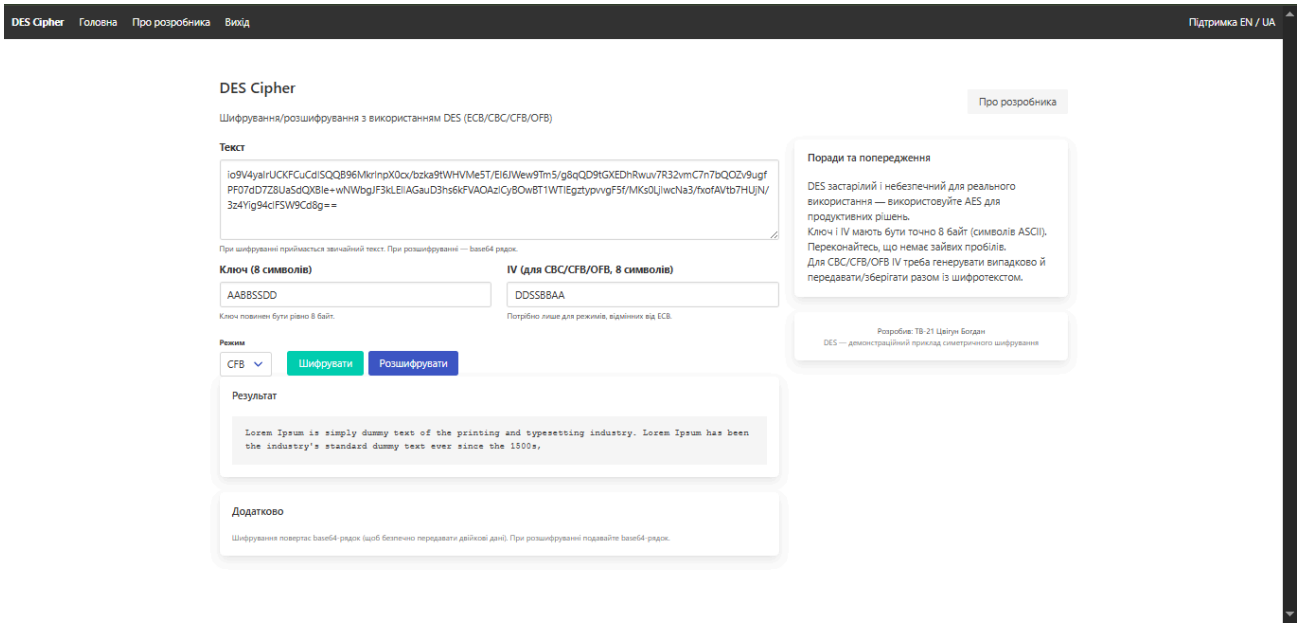
Ендпойнт /des/ реалізує веб-інтерфейс для симетричного блочного шифру DES (Data Encryption Standard) у межах Flask-застосунку. Його основне призначення полягає у забезпеченні взаємодії між користувачем і серверною частиною для виконання процесів шифрування та розшифрування тексту. Під час обробки запиту методом POST ендпойнт отримує з форми дані — вхідний

текст, ключ, ініціалізаційний вектор, режим роботи та обрану дію. Перед початком обчислень виконується перевірка коректності введених параметрів: ключ повинен містити рівно вісім байт, а ініціалізаційний вектор (IV), якщо він використовується, також має відповідати цій довжині.

Далі відбувається створення об’єкта шифра відповідно до вибраного режиму, серед яких підтримуються ECB, CBC, CFB та OFB. У режимі шифрування текст доповнюється до довжини, кратної розміру блоку, після чого шифрується і кодується у формат Base64 для коректного відображення в інтерфейсі. У режимі розшифрування виконується зворотна процедура — вхідні дані декодуються, розшифровуються та очищуються від заповнення.

Результат операції або повідомлення про можливі помилки передаються у шаблон DES.html, який відображає відповідну інформацію користувачеві. Така реалізація забезпечує зручну демонстрацію принципів роботи блочного симетричного шифру DES та дозволяє на практиці спостерігати відмінності між різними режимами його роботи.





Висновок:

У результаті виконання практичної роботи було створено веб-інтерфейс для симетричного блочного шифрування за допомогою алгоритму DES з підтримкою всіх основних режимів роботи (ECB, CBC, CFB, OFB). Розроблена система забезпечує коректну обробку ключів та ініціалізаційних векторів, виконує шифрування і розшифрування тексту та відображає результат користувачеві у зручному форматі. Реалізація на базі Flask і використання сучасних веб-технологій дозволяє продемонструвати принципи роботи DES у навчальних цілях, а також надає можливість тестування різних режимів шифрування та їхнього впливу на результат, забезпечуючи практичне розуміння роботи симетричних криптографічних алгоритмів.