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

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 3
з курсу: «*Безпека програмного забезпечення*»

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

Київ 2025

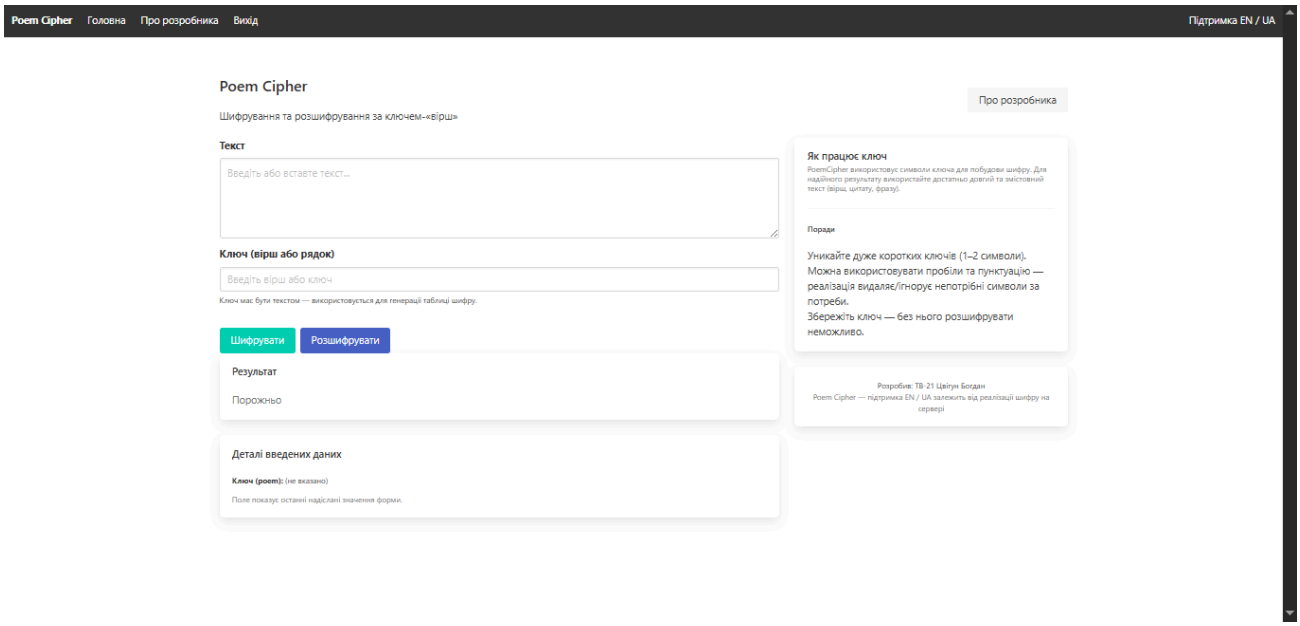
Практична робота № 3

Завдання:

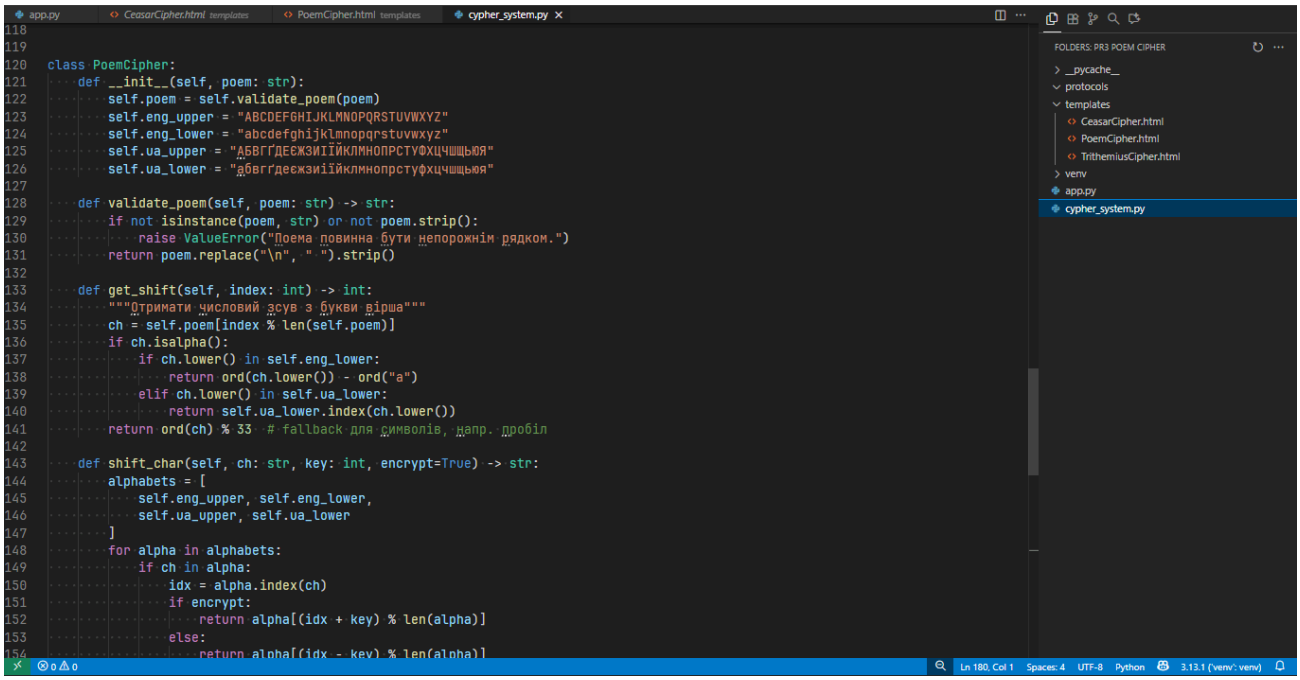
1. Розробіть інтерфейс криптографічної системи для реалізації шифрування з використанням вірша.
2. Доповніть систему класів з попередніх лабораторних робіт класами та методами, необхідними для шифрування і розшифрування віршованим шифром.
3. Виконати тестування роботи системи.

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 Поеми є поліалфавітним методом шифрування, в якому ключ формується на основі заданого вірша або текстового фрагмента. Кожна літера ключа визначає зсув для відповідної літери відкритого тексту, а при завершенні ключа процес повторюється циклічно. Такий підхід робить шифр більш стійким до простого частотного аналізу, оскільки одна й та сама літера може бути зашифрована різними символами залежно від позиції у тексті. Попри це, через передбачуваність структури ключа та обмежену довжину вірша шифр Поеми залишається вразливим до статистичних методів розкриття і використовується переважно в навчальних цілях для демонстрації принципів поліалфавітного шифрування.



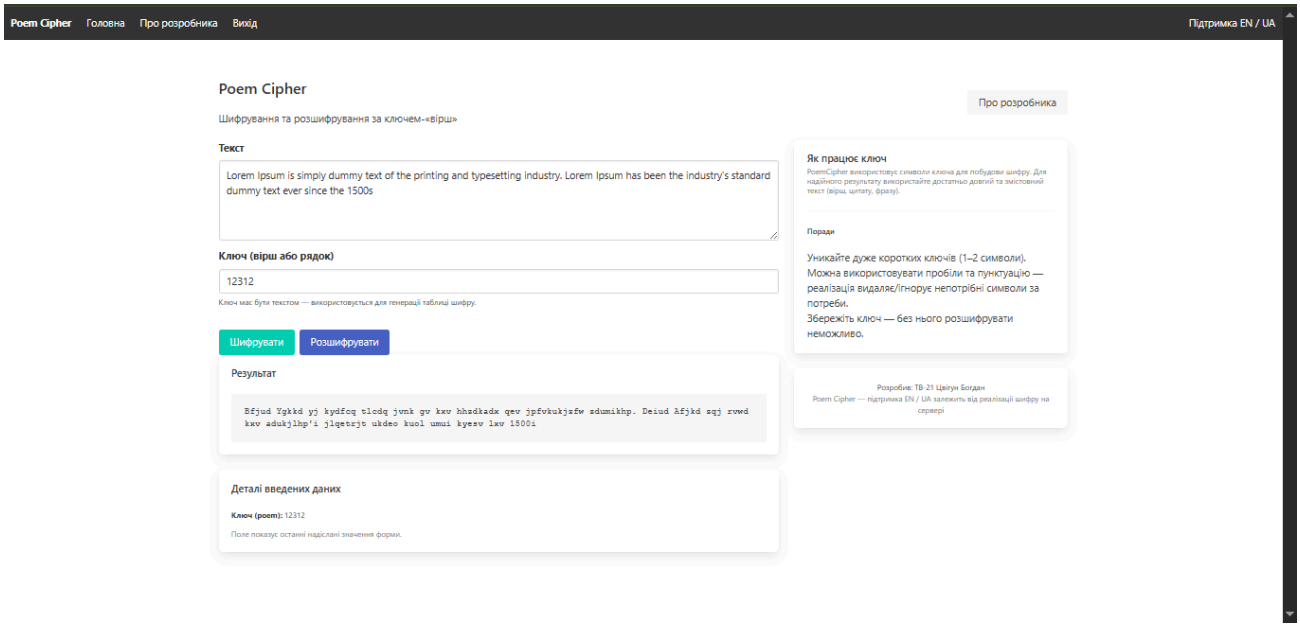

```python
class PoemCipher:
    def __init__(self, poem: str):
        self.poem = self.validate_poem(poem)
        self.eng_upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        self.eng_lower = "abcdefghijklmnopqrstuvwxyz"
        self.ua_upper = "АБВГҐДЕЄЖЗИІЇЙКЛМНОПРСТУФХЦЧШЩЬЮЯ"
        self.ua_lower = "абвгґдеєжзиіїйклмнопрстуфхцчшщьюя"

    def validate_poem(self, poem: str) -> str:
        if not isinstance(poem, str) or not poem.strip():
            raise ValueError("Поема повинна бути непорожнім рядком.")
        return poem.replace("\n", " ").strip()

    def get_shift(self, index: int) -> int:
        """Отримати числовий зсув з букви вірша"""
        ch = self.poem[index % len(self.poem)]
        if ch.isalpha():
            if ch.lower() in self.eng_lower:
                return ord(ch.lower()) - ord("a")
            elif ch.lower() in self.ua_lower:
                return self.ua_lower.index(ch.lower())
        return ord(ch) % 33  # fallback для символів, напр. пробіл

    def shift_char(self, ch: str, key: int, encrypt=True) -> str:
        alphabets = [
            self.eng_upper, self.eng_lower,
            self.ua_upper, self.ua_lower
        ]
        for alpha in alphabets:
            if ch in alpha:
                idx = alpha.index(ch)
                if encrypt:
                    return alpha[(idx + key) % len(alpha)]
                else:
                    return alpha[(idx - key) % len(alpha)]
```

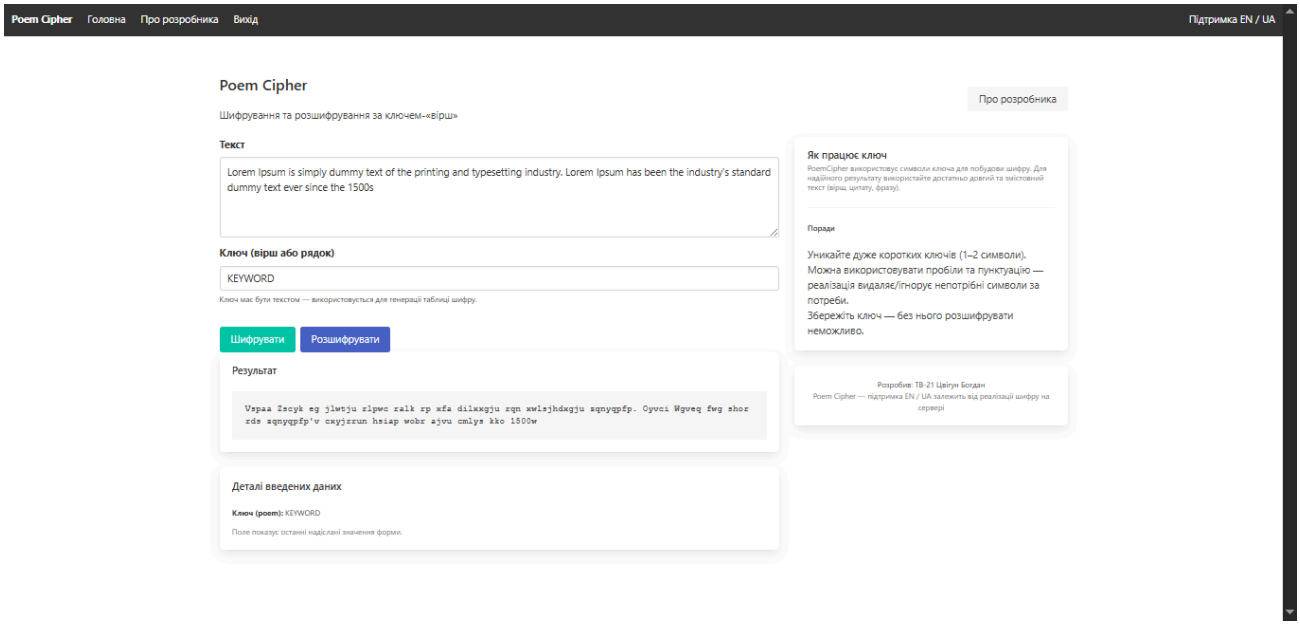
Повний вихідний код проєкту доступний у репозиторії, що буде додано до завдання.

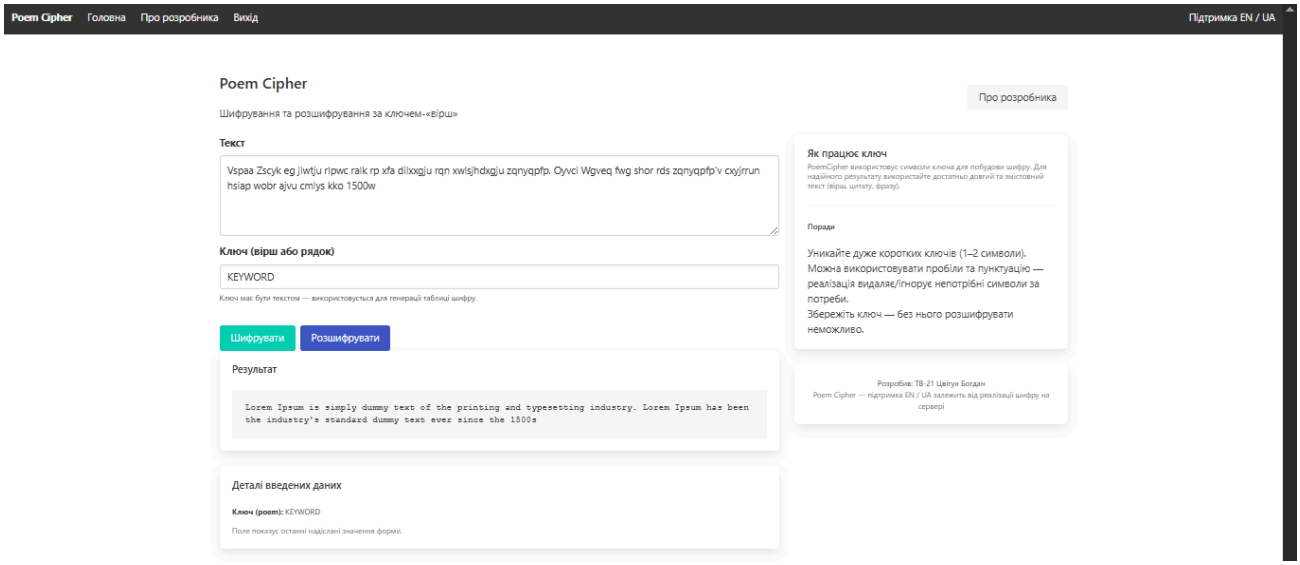


Клас PoemCipher реалізує симетричний метод поліалфавітного шифрування, у якому ключем виступає вірш або довільний текстовий фрагмент. Кожна літера ключа визначає числовий зсув, який використовується для шифрування відповідного символу у вхідному повідомленні. Коли кінець ключа досягається, зсуви повторюються циклічно, що забезпечує рівномірне застосування шаблону шифрування до всього тексту.

Під час ініціалізації виконується перевірка валідності вхідного ключа-вірша та підготовка алфавітів — українського та англійського у верхньому й нижньому регістрах. Метод get_shift() обчислює величину зсуву для певної позиції, перетворюючи символ вірша у числове значення. Процедура shift_char() виконує безпосереднє зміщення літери у відповідному алфавіті під час шифрування або розшифрування, залишаючи неалфавітні символи (цифри, пробіли, розділові знаки) без змін.

Методи encrypt() і decrypt() реалізують двосторонній процес перетворення тексту, синхронно використовуючи символи ключа для кожної позиції вхідного рядка. Такий підхід забезпечує різні зсуви для однакових літер залежно від їхнього розташування у тексті, що підвищує криптографічну стійкість порівняно з простими моноалфавітними шифрами.





Висновок:

У результаті виконання роботи було розроблено веб-застосунок, який реалізує шифрування та розшифрування текстів за допомогою віршованого (Poem) шифру. Система побудована на основі фреймворку Flask із використанням шаблонізатора Jinja2 та CSS-фреймворку Bulma, що забезпечило зручний інтерфейс і коректну взаємодію між клієнтською та серверною частинами. Реалізований клас PoemCipher продемонстрував принципи поліалфавітного шифрування, а проведене тестування підтвердило працездатність і правильність реалізації методів шифрування та розшифрування.