**🧾 Short Report: Web-Based Text Encryption Tool**

**1. Introduction**

This project is a **web-based text encryption tool** designed to demonstrate classical cryptographic techniques using JavaScript.  
Users can input plaintext, select an encryption algorithm, provide a key (if required), and view the resulting ciphertext.  
The goal is to illustrate how different encryption methods transform data to protect it from unauthorized access.

**2. Implemented Encryption Techniques**

The system includes **three classical algorithms**:  
**Transposition Cipher**, **Polyalphabetic Cipher**, and **Vigenère Cipher**.  
Each represents a different approach to encryption — rearrangement or substitution.

**🔹 2.1 Transposition Cipher**

**Type:** Rearrangement Cipher  
**Concept:**  
In a transposition cipher, the characters of the plaintext are **rearranged** according to a specific pattern or key, without changing the actual letters.

**Working Principle:**

1. The plaintext is written into a grid with columns equal to the key length.
2. The columns are then read in alphabetical order of the key letters.
3. The resulting sequence of letters forms the ciphertext.

**Example:**  
Plaintext: HELLO WORLD  
Key: ZEBRA

Grid (rows):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Z | E | B | R | A |
| H | E | L | L | O |
| W | O | R | L | D |

Alphabetical order of key: **A, B, E, R, Z**  
Ciphertext (read column by column): OLLERDHELOW

**Decryption:**  
The reverse process reconstructs the grid column-wise and reads it row by row.

**🔹 2.2 Polyalphabetic Cipher**

**Type:** Substitution Cipher  
**Concept:**  
This cipher uses **multiple substitution alphabets**, controlled by a repeating key, to make frequency analysis attacks more difficult.

**Working Principle:**

1. Each letter in the plaintext is shifted by a number determined by the corresponding key letter.
2. For example, key letter **A** means no shift, **B** means shift by 1, and so on.

**Example:**  
Plaintext: HELLO  
Key: KEY  
Shifts: K(+10), E(+4), Y(+24)  
Ciphertext: RIJVS

**Decryption:**  
Subtract the key shifts from each ciphertext letter to get back the original text.

**🔹 2.3 Vigenère Cipher**

**Type:** Substitution Cipher (Special Case of Polyalphabetic)  
**Concept:**  
The Vigenère cipher is a refined version of the polyalphabetic cipher.  
It uses the same principle of shifting each letter by the amount indicated by the key but skips non-alphabetic characters so the key aligns only with letters.

**Working Principle:**

1. Repeat the key to match the length of the plaintext.
2. Shift each plaintext letter by the alphabetical index of the corresponding key letter.

**Example:**  
Plaintext: ATTACK AT DAWN  
Key: LEMON  
Ciphertext: LXFOPV EF RNHR

**Decryption:**  
Reverse the shifts using the same key.

**3. Validation and User Interaction**

* Input validation ensures users provide both text and a valid key.
* A simple interface allows encrypting, decrypting, and copying results.
* JavaScript handles all encryption logic directly in the browser (no server-side code required).

**4. Conclusion**

This project demonstrates fundamental encryption methods that laid the foundation for modern cryptography.  
While classical ciphers like **Transposition**, **Polyalphabetic**, and **Vigenère** are no longer secure for practical use, they effectively illustrate core cryptographic concepts such as **permutation**, **substitution**, and **key-based transformation**.  
The web tool provides an educational and interactive way to explore how encryption works at a conceptual level.