Name: Arham Sharif

Seat No.: EB21102022

Section: B

Subject: Network Security & Cryptography

Language: JavaScript

LAB# 1

CESEAR CIPHER

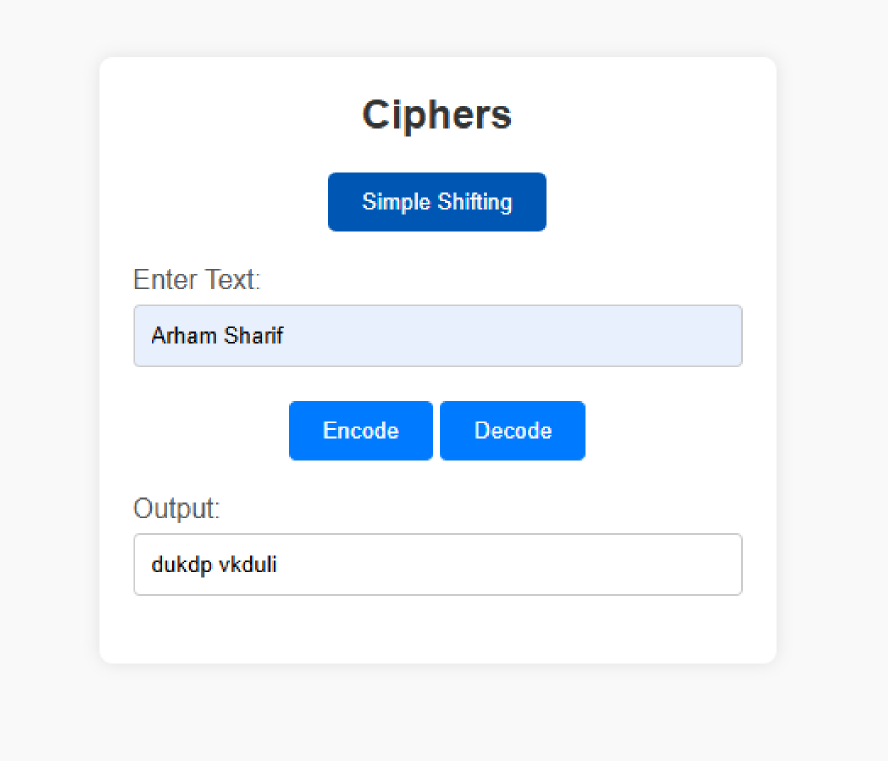
Objective: Design and implement a simple encoding and decoding program in JavaScript that allows users to input a string, encode it using a specified algorithm, and then decode it back to the original form. This lab aims to reinforce understanding of string manipulation, algorithmic concepts, and basic programming skills while demonstrating the principles of encoding and decoding.

Technique: Substitution

Code:

```
const simpleInc = 3;

const simpleCharArr = Array.from({ length: 26 }, (_, i) =>
String.fromCharCode(97 + i));

const simpleLenCharArr = simpleCharArr.length;

"""-----------------ENCODE------------------"""

# Function Can Encode Char

const encodeCharSimple = (char) => {

    let encodeChar = '';

    let index = -1;

    for (let i = 0; i < simpleLenCharArr; i++) {

        if (simpleCharArr[i] === char) {

            index = i + simpleInc;

            if (index >= simpleLenCharArr) {

                index %= simpleLenCharArr;

            }

            encodeChar = simpleCharArr[index];

            break;
        }
    }
    if (index !== -1) {
        return encodeChar;
    } else {
        return char;
    }
}
```

Output:



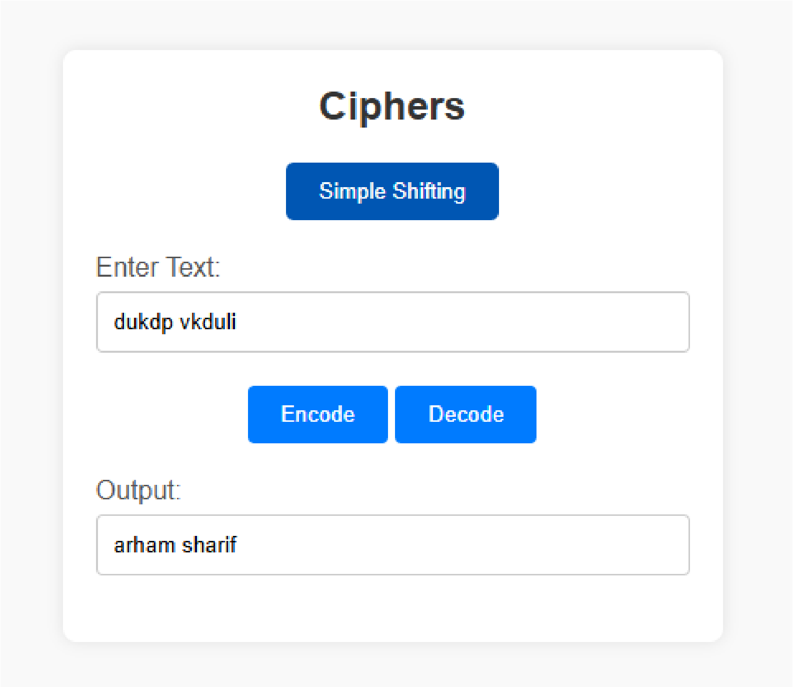
Enter Text: Arham Sharif

Output: dukdp vkduli

Code:

```
"""-----------------DECODE------------------"""
# Function to Decode Char
const decodeCharSimple = (char) => {
    let decodeChar = '';
    let index = -1;
    for (let i = 0; i < simpleLenCharArr; i++) {
        if (simpleCharArr[i] === char) {
            index = i - simpleInc;
            if (index < 0) {
                index += simpleLenCharArr;
            }
            decodeChar = simpleCharArr[index];
            break;
        }
    }
    if (index !== -1) {
        return decodeChar;
    } else {
        return char;
    }
}
```

Output:



Enter Text: dukdp vkduli

Output: arham sharif

LAB# 2

VIGENÈRE CIPHER

Objective: Design and implement the Vigenère cipher, a classical encryption technique that uses a keyword to encrypt and decrypt messages.

Key: zautnq

Matrix:

zabcdefghijklmnopqrstuvwxy

abcdefghijklmnopqrstuvwxyz

uvwxyzabcdefghijklmnopqrst

tuvwxyzabcdefghijklmnopqrs

nopqrstuvwxyzabcdefghijklm

qrstuvwxyzabcdefghijklmnop

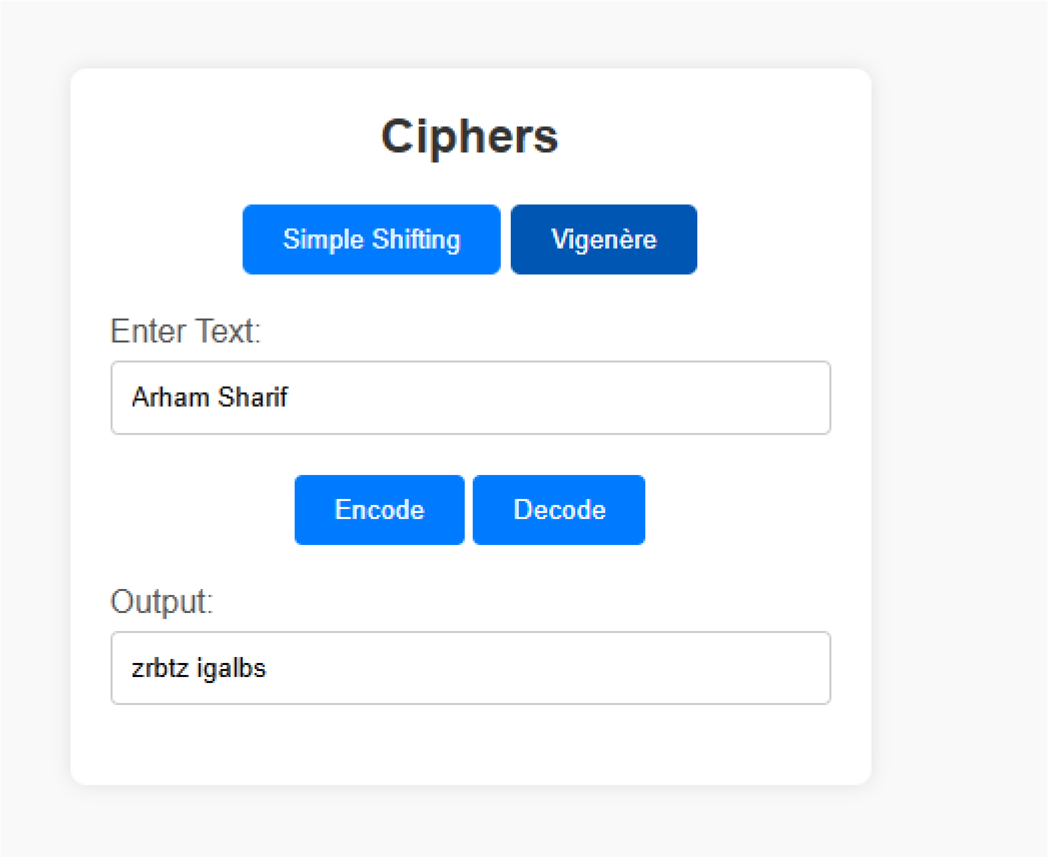
Code:

```
const vigenereCharArr = Array.from({ length: 26 }, (_, i) =>
String.fromCharCode(97 + i));

function generateVigenereRandomKey(length) {

    const charset = vigenereCharArr.join("");

    let key = '';

    for (let i = 0; i < length; i++) {

        const randomIndex = Math.floor(Math.random() *
charset.length);

        key += charset[randomIndex];

    }

    return key;

}
```

```
// Function to generate the Vigenère character array based on the key
function generateVigenereCharArr(key) {
    const charArr = [];
    for (let i = 0; i < key.length; i++) {
        const shift = key.charCodeAt(i) - 97; // Get the shift amount for each character in the key
        const shiftedChars = vigenereCharArr.slice(shift).concat(vigenereCharArr.slice(0, shift));
        charArr.push(shiftedChars);
    }
    return charArr;
}


// Function to save Vigenère key and character array to local storage
function saveVigenereToLocalStorage(key, charArr) {
    localStorage.setItem('vigenereKey', key);
    localStorage.setItem('vigenereCharArr', JSON.stringify(charArr));
}


// Function to retrieve Vigenère key and character array from local storage
function getVigenereFromLocalStorage() {
    const key = localStorage.getItem('vigenereKey');
    const charArr = JSON.parse(localStorage.getItem('vigenereCharArr'));
    return { key, charArr };
}
```

```
// Generate random key and character array
const randomKey = generateVigenereRandomKey(6); // Change the length as needed
const randomCharArr = generateVigenereCharArr(randomKey);

// Save them to local storage
saveVigenereToLocalStorage(randomKey, randomCharArr);
"""-----------------ENCODE------------------"""
# Function Can Encode Char
function encryptVigenereShifting(message) {
    const { key, charArr } = getVigenereFromLocalStorage();

    let result = '';

    for (let i = 0, j = 0; i < message.length; i++) {
        const c = message.charAt(i);
        const index = vigenereCharArr.indexOf(c);
        if (index !== -1) {
            result += charArr[j % key.length][index];
            j++;
        } else {
            result += c;
        }
    }
    return result;
}
```

Output:



Enter Text:

Arham Sharif

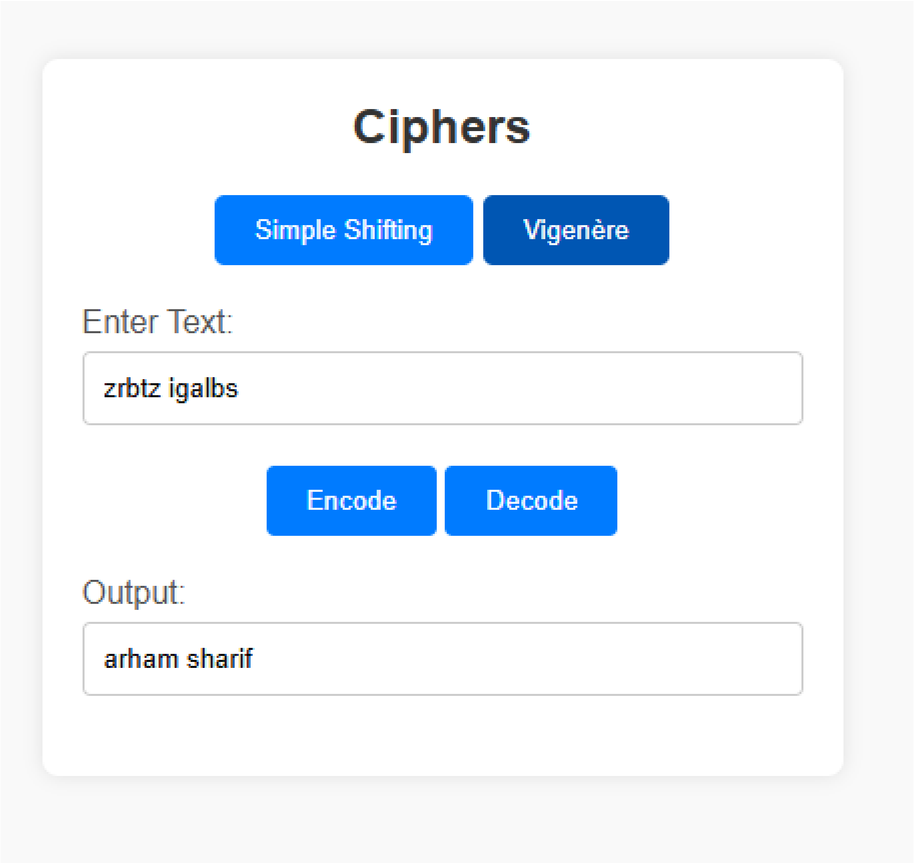
Output:

zrbtz igalbs

Code:

```
"""------------------DECODE-------------------"""
function decryptVigenereShifting(message) {
    const { key, charArr } = getVigenereFromLocalStorage();

    let result = '';

    for (let i = 0, j = 0; i < message.length; i++) {
        const c = message.charAt(i);
        const rowIndex = j % key.length;
        const charIndex = charArr[rowIndex].indexOf(c);
        if (charIndex !== -1) {
            result += vigenereCharArr[charIndex];
            j++;
        } else {
            result += c;
        }
    }
    return result;
}
```

Output:



Ciphers

Enter Text:

zrbtz igalbs

Output:

arham sharif