CSE – 537

Network Security

Practical Assignment – 0(PA0)

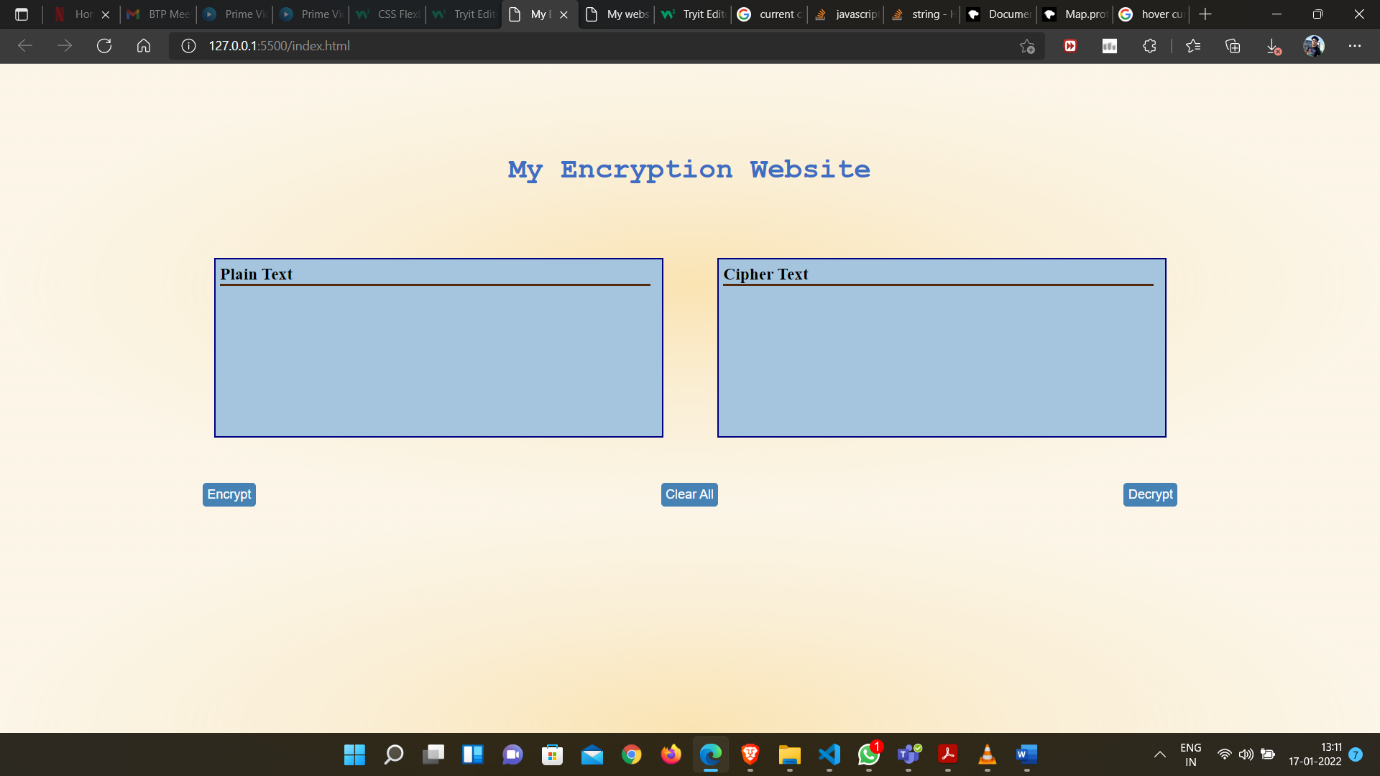
Name: Mohd. Rajeen

Roll No.: 19095065

Department: ECE

e-mail: [mohd.rajeen.ece19@iitbhu.ac.in](mailto:mohd.rajeen.ece19@iitbhu.ac.in)

**Web Interface:**

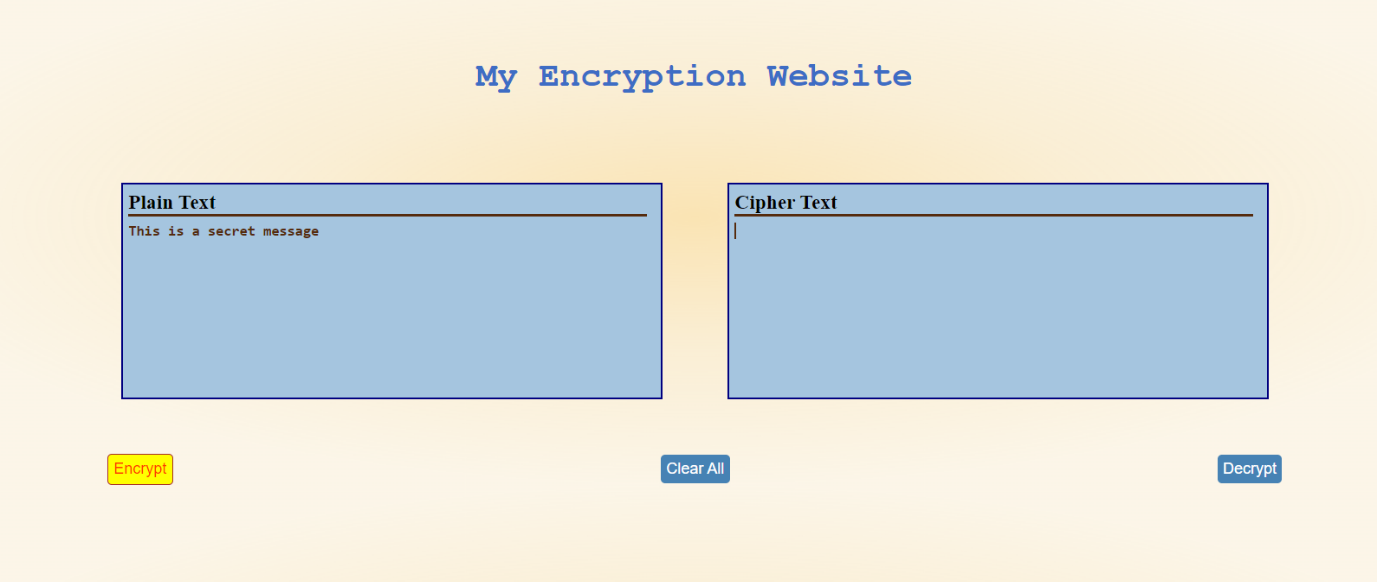


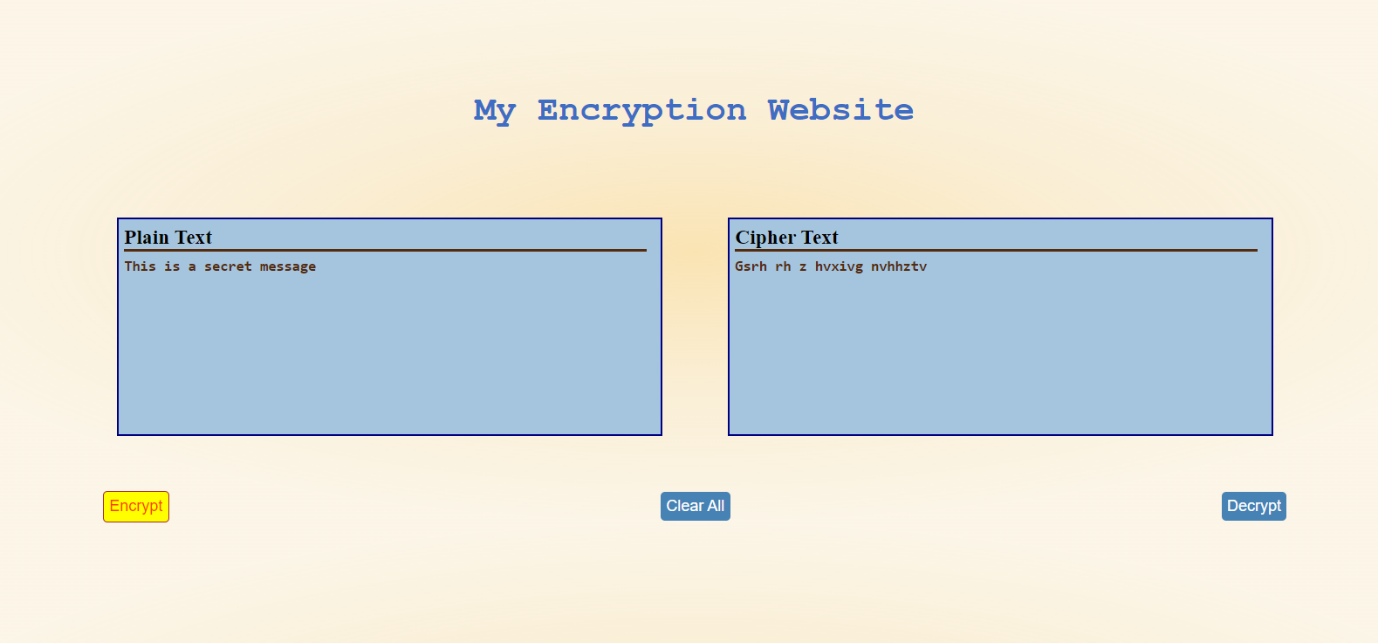
Available Utilities:

* 2 text boxes are available: one for plain text and another for cipher text
* Encrypt button reads the text from plain textbox and outputs the encrypted text in cipher textbox.
* Decrypt button reads the text from cipher textbox and outputs the decrypted text in plain textbox.
* Clear button can be used to clear both the textboxes for a fresh start.

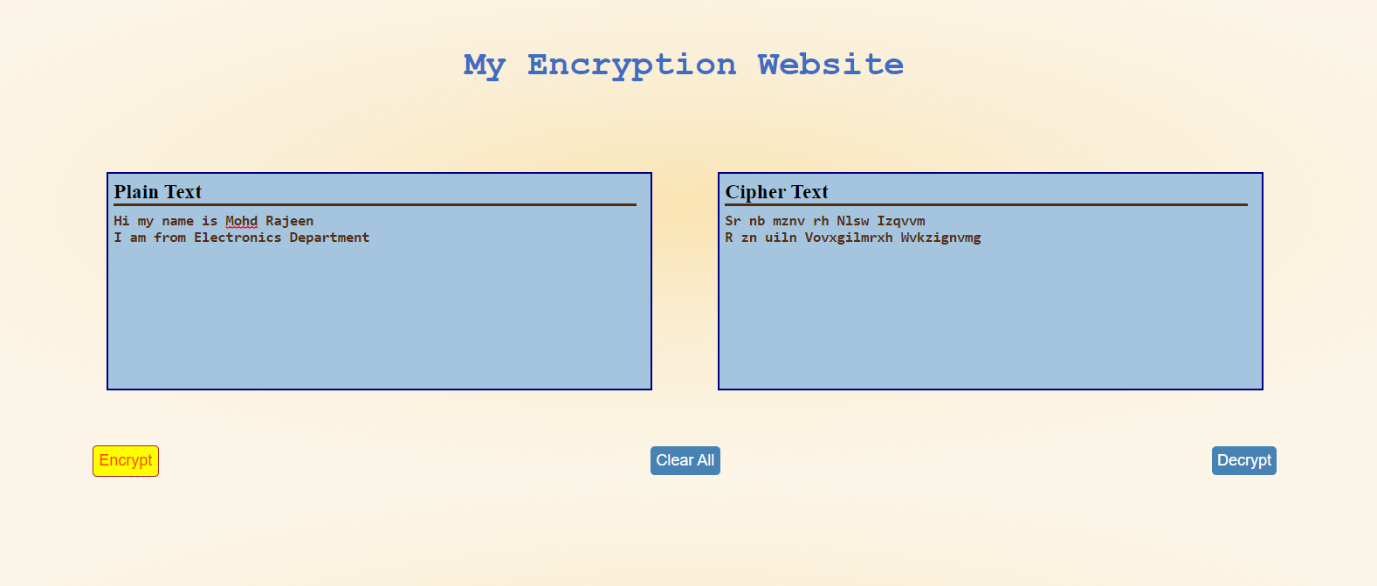
Encryption Examples:

1.



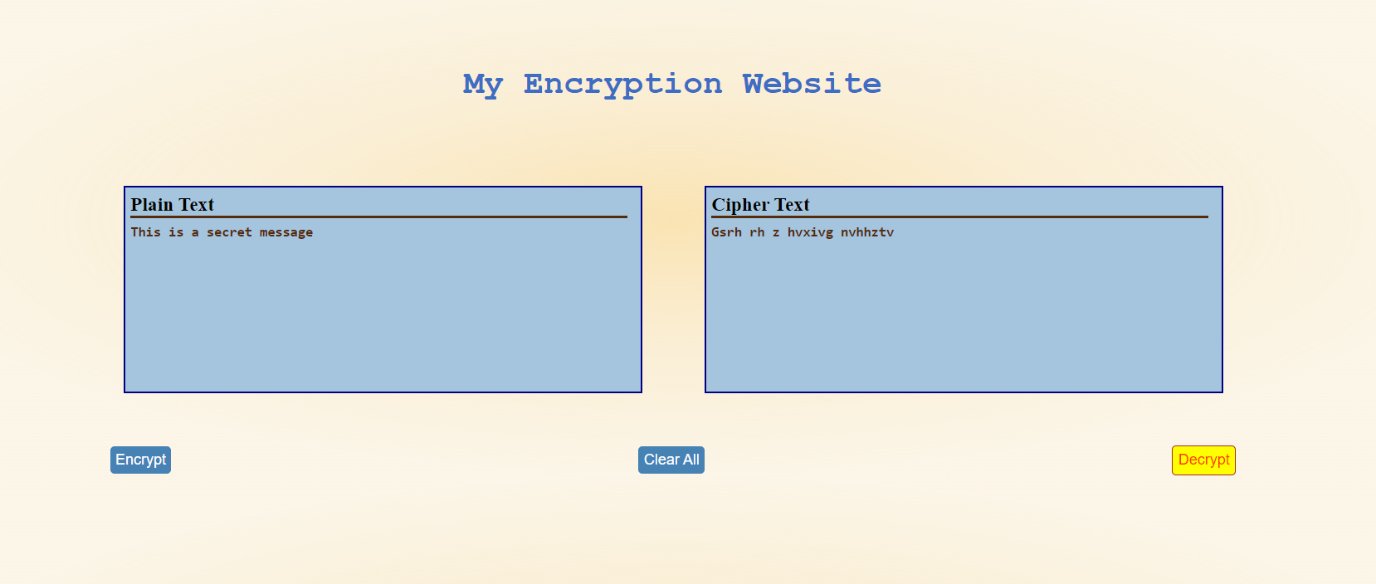
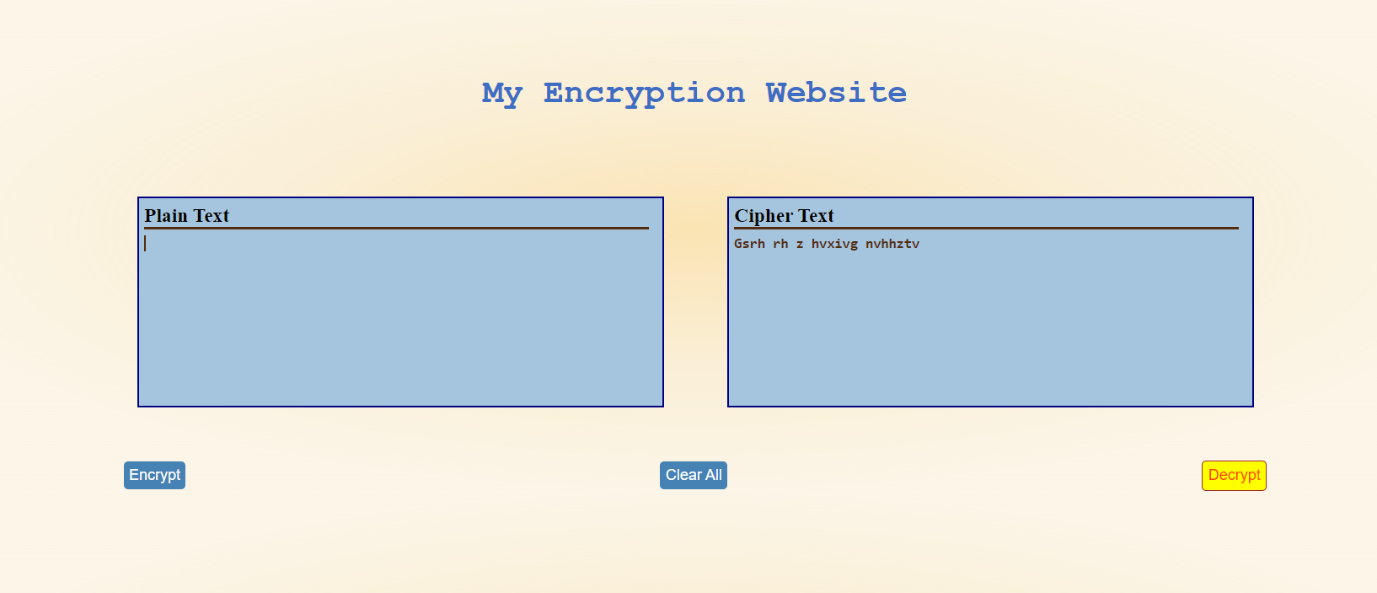


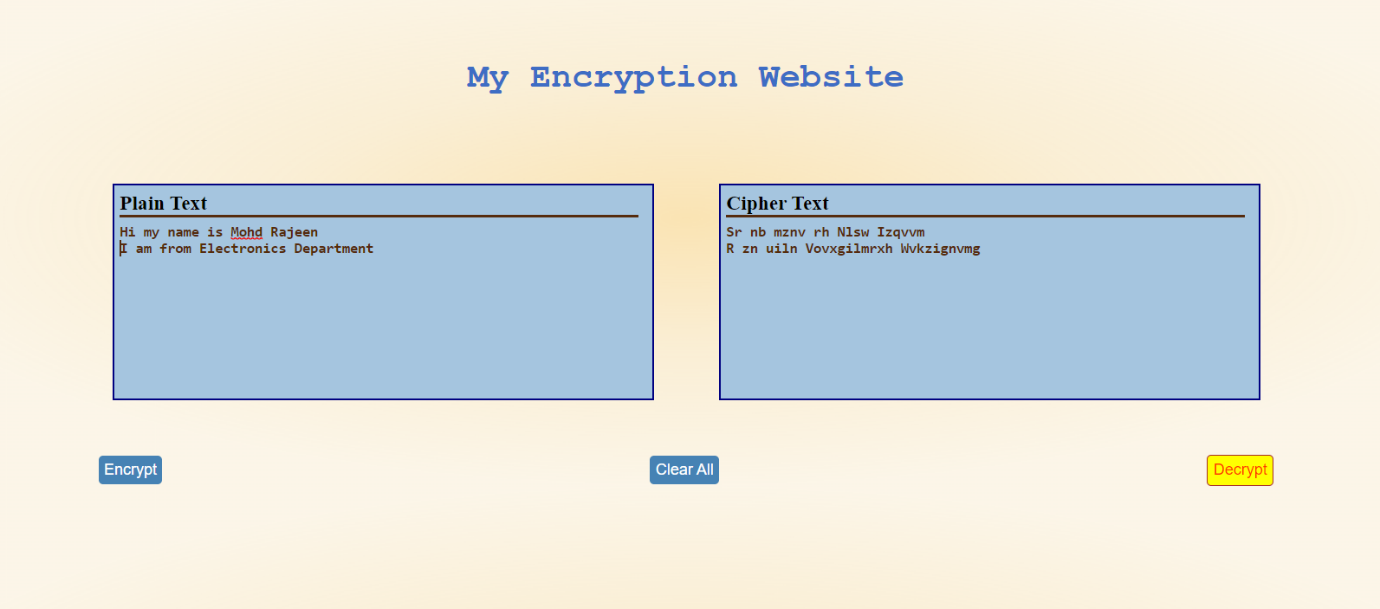
2.



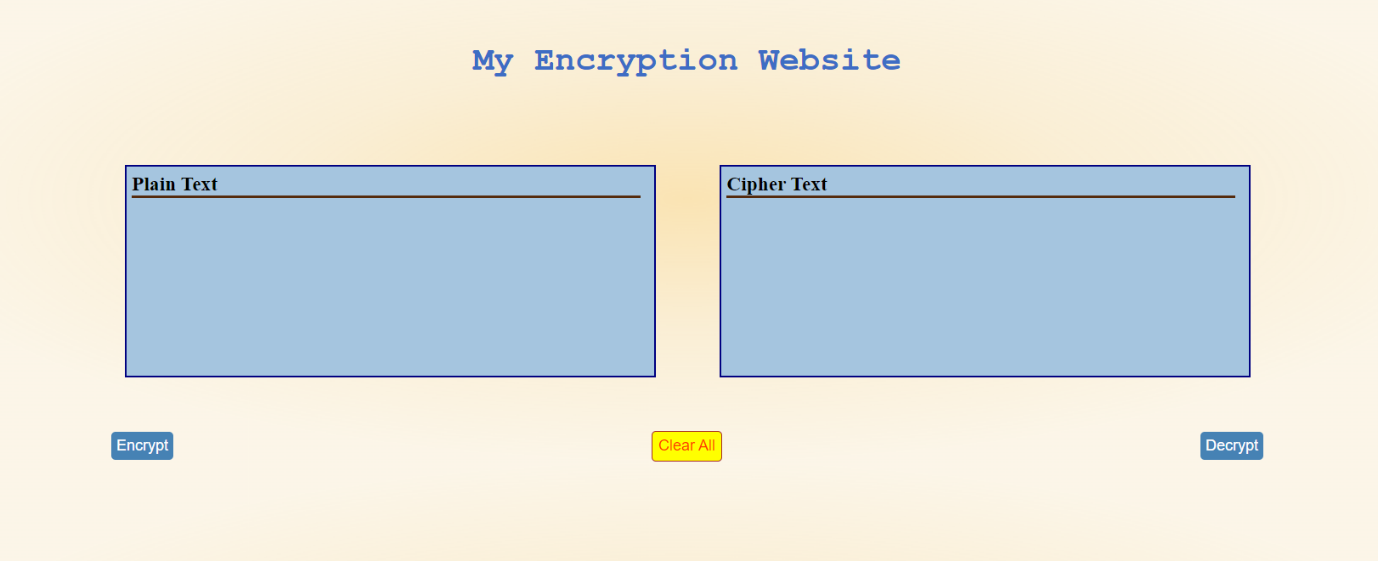
Decryption Examples:

1.



2.

Clear button:



Source Code:

The source code consists of 3 different files : HTML, CSS and JavaScript.

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="style.css">

    <title>My Encryption website</title>

</head>

<body>

    <h1 class="heading">My Encryption Website</h1>

    <div class="main">

        <div class="container">

            <div class="plain box">

                <h3>Plain Text</h3>

                <textarea rows="10" cols="63" name="comment" form="usrform" id="myTextArea"></textarea>

            </div>

            <div class="cipher box">

                <h3>Cipher Text</h3>

                <textarea rows="10" cols="63" name="comment" form="usrform" id="cipherTextArea"></textarea>

            </div>

        </div>

        <div class="butDiv">

            <button id="but1" class="but">Encrypt</button>

            <button id="clr" class="but">Clear All</button>

            <button id="but2" class="but">Decrypt</button>

        </div>

    </div>

    <script src="encrypt.js"></script>

</body>

</html>

JS:

//map1 is used for encryption

// it has key value pairs as :

// a: z , b:y  and so on

// it also has A:Z, B:Y and so on

// size of map1 is 52

var map1 = new Map();

for (let i = 0; i < 26; i++)

{

    let k1 = String.fromCharCode(65+i)

    let v1 = String.fromCharCode(65+25-i)

    let k2 = String.fromCharCode(97+i)

    let v2 = String.fromCharCode(97+25-i)

    map1.set(k1,v1);

    map1.set(k2,v2);

}

// similar to map1 we have map2, only change is that the values corresponding to keys in map1 are keys for map2 and vice- versa

var map2 = new Map();

for (let i = 0; i < 26; i++)

{

    let k1 = String.fromCharCode(65+i)

    let v1 = String.fromCharCode(65+25-i)

    let k2 = String.fromCharCode(97+i)

    let v2 = String.fromCharCode(97+25-i)

    map2.set(v1,k1);

    map2.set(v2,k2);

}

var but1= document.getElementById("#but1")

var but2= document.getElementById("#but2")

function encrypt\_fun(t)

{

    // console.log(t)

    var newtext="";

    for (var i = 0; i < t.length; i++)

    {

        if(t.charAt(i)==' ' || map1.get(t.charAt(i))==undefined )

            newtext+=t.charAt(i);

        else

            newtext+= map1.get(t.charAt(i))

    }

    // console.log(newtext)

    document.querySelector("#cipherTextArea").value=newtext;

}

function decrypt\_fun(t)

{

    // console.log(t)

    var newtext="";

    for (var i = 0; i < t.length; i++)

    {

        if(t.charAt(i)==' ' || map2.get(t.charAt(i))==undefined)

            newtext+=" ";

        else

            newtext+= map2.get(t.charAt(i))

    }

    // console.log(newtext)

    document.querySelector("#myTextArea").value=newtext;

}

document.querySelector("#but1").onclick = function(){

    var t= document.querySelector("#myTextArea").value;

    // console.log(t)

    encrypt\_fun(t);

}

document.querySelector("#but2").onclick = function(){

    var t= document.querySelector("#cipherTextArea").value;

    decrypt\_fun(t);

}

document.querySelector("#clr").onclick = function(){

    document.querySelector("#myTextArea").value="";

    document.querySelector("#cipherTextArea").value="";

}

CSS:

\*{

    margin:0;

    padding: 0;

    box-sizing: border-box;

    /\* background-color: rgb(192, 192, 192); \*/

}

body{

    background-image: radial-gradient(closest-side at 50% 50%,rgba(247, 207, 121, 0.562),rgba(243, 214, 150, 0.397), rgba(243, 219, 166, 0.26));

}

.heading{

    margin: 100px 0 80px 0 ;

    text-align: center;

    font-family: 'Courier New', Courier, monospace;

    color: rgb(62, 109, 196);

}

.container{

    margin-top: 50px;

    display:flex;

    justify-content: center;

}

.box{

    height: 200px;

    width: 500px;

    margin: auto 30px;

    padding: 5px;

    background-color: rgb(165, 197, 223);

}

textarea{

    padding: 5px 0 0 0 ;

    font-weight: 800;

    color: rgb(85, 43, 13);

    resize: none;

    border: none;

    border-top: 3px solid rgb(85, 43, 13);

    outline: none;

    background-color: rgb(165, 197, 223);

}

h3{

    background-color: rgb(165, 197, 223);

}

.plain{    border: 2px solid navy;

}

.cipher{    border: 2px solid navy;

}

.butDiv{

    margin-top: 50px;

    align-items: center;

    /\* max-width: 800px; \*/

    display: flex;

    justify-content: space-around;

}

.but{

    /\* margin: 30px 0 0 235px; \*/

    border: none;

    outline: none;

    padding: 5px;

    background-color: steelblue;

    color: white;

    border-radius: 4px;

    font-size: 0.9rem;

}

.but:hover{

    /\* background-color: rgb(12, 34, 53); \*/

    background-color: yellow;

    border: 1px solid brown;

    color: rgb(255, 60, 0);

    cursor: pointer;

}