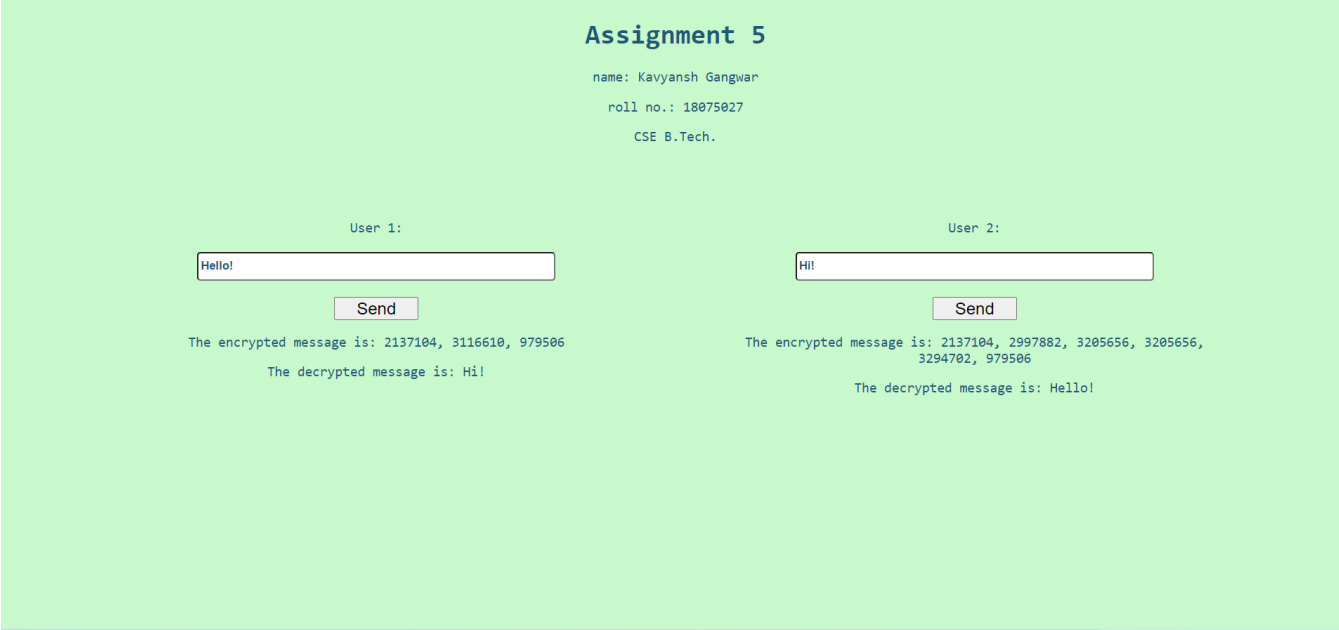
Assignment 5

Name - Kavyansh Gangwar
Roll no. - 18075027
CSE B. Tech

Screen Shots



Source Code

assignment.html →

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>El Gammal</title>
    <link rel="stylesheet" href="assignment.css">
  </head>
  <body>
    <div class="container center">
      <h1>Assignment 5</h1>
```

```
      <p>name: Kavyansh Gangwar</p>
      <p>roll no.: 18075027</p>
      <p>CSE B.Tech.</p>
      <table>


      <tr>


        <!-- user 1 -->
        <td class="user1" valign='top'>
          User 1:
          <br>
          <br>
          <input type="text" name="" value="" id="user1msg">
          <br>
          <br>
          <button type="button"
name="send"id="user1send">Send</button>
          <br>
          <p id="user1encrec">The encrypted message is: </p>
          <p id='user1decrec'>The decrypted message is: </p>
        </td>
        <!-- user 2 -->
        <td class="user2" valign='top'>
          User 2:<br><br>
          <input type="text" name="" value="" id="user2msg"><br><br>
          <button type="button"
name="send"id="user2send">Send</button><br>
          <p id="user2encrec">The encrypted message is:</p>
          <p id='user2decrec'>The decrypted message is:</p>
        </td>
        </tr>
        </table>
      </div>

    <script src="assignment.js" charset="utf-8"></script>
  </body>
</html>
```

assignment.css →

```
body{
  background-color: #C7F9CC;
  color: #22577A;
  font-family: monospace;
  font-size: 1.2em;

}

.container{
  margin-left: 5em;
  margin-right: 5em;
  margin-top: auto;
  margin-bottom: auto;
}

input{
  height: 2em;
  width: 30em;
  border-radius: 0.3em;
  color: inherit;
  font-weight: bold;
}
.center{
  text-align: center;
}

button{
  width: 5em;
  height: 1.4em;
  font-size: 1em;
}
table{
  width: 100%;
}
td{
  padding: 5%;
  width: 50%;
}
```

assignment.js →

```
// user 1 variables
var user1Msg = document.getElementById("user1msg");
var user1Send = document.getElementById("user1send");
var user1EncRec = document.getElementById("user1encrec");
var user1DecRec = document.getElementById("user1decrec");

var user1PrivateKey;
var user1PublicKey;

// user 2 variables
var user2Msg = document.getElementById("user2msg");
var user2Send = document.getElementById("user2send");
var user2EncRec = document.getElementById("user2encrec");
var user2DecRec = document.getElementById("user2decrec");

var user2PrivateKey;
var user2PublicKey;

// function to compute gcf of two numbers
const gcd = function(a, b){
  if(a<b){
    return gcd(b,a);
  }
  if(a%b === 0){
    return b;
  }
  return gcd(b,a%b);
}

const genKey = function(q){
  var key = Math.random()*(q-Math.pow(2,6));
  key+=Math.pow(2,6);
  return Math.trunc(key);
}
// function to generate key
const generateKey = function(q){
  var key = genKey(q);
  while(gcd(q,key)≠1){
```

```
    key=genKey(q);
  }
  return key;
}

// modular exponentiation
const power = function(a,b,c){
  var x = 1;
  var y = Math.trunc(a);
  while(b>0){
    if(b%2≠0){
      x=(x*y)%c;
    }
    y=(y*y)%c;
    b=Math.trunc(b/2);
  }
  return x%c;
}

// encryption function
const encrypt = function(msg,q,h,g,user){
  var encMsg = [];
  var k;
  if(user==1){
    k=user1PrivateKey;
  }else{
    k=user2PrivateKey;
  }
  s = power(h,k,q);
  p = power(g,k,q);

  for(var i=0;i<msg.length;i++){
    encMsg.push(s*msg.charCodeAt(i));
  }
  return encMsg;
}

// decryption function
const decrypt = function(encMsg,p,key,q){
```

```
  decMsg = "";
  var h = power(p,key,q);
  for(var i=0;i<encMsg.length;i++){
    decMsg+=String.fromCharCode(Math.trunc(encMsg[i]/h));
  }
  return decMsg;
}

const generateQ = function(){
  var primes = [];
  var flag = true;
  for(var i=Math.pow(2,6)+1;i<Math.pow(2,16);i+=2){
    flag =true;
    for(j=2;j<i;j++){
      if(i%j==0){
        flag = false;
        break;
      }
    }
    if(flag){

      primes.push(i);
    }
  }
  return primes[Math.trunc(Math.random()*primes.length)];
}

// generate q
var q = generateQ();

// generate g
var g = Math.trunc((Math.random()*(q-2))+2);
// private key of user 2
user2PrivateKey = generateKey(q);
// h of user 2
var h = power(g,user2PrivateKey,q);
// public key of user 2
user2PublicKey = {'g':g,'h':h,'q':q};
// private key of user 1
```

```
user1PrivateKey = generateKey(q);
// public key of user 1
user1PublicKey = {'g':g,'h':power(g,user1PrivateKey,q),'q':q};
// var encMsg = encrypt(msg,q,user2PublicKey['h'],g,1);
// var decMsg =
decrypt(encMsg['encMsg'],user1PublicKey['h'],user2PrivateKey,q);
// console.log(decMsg);

user1Send.addEventListener('click',()⇒{
  var msg = user1Msg.value;
  var encMsg = encrypt(msg,q,user2PublicKey['h'],g,1);
  var decMsg = decrypt(encMsg,user1PublicKey['h'],user2PrivateKey,q);
  user2EncRec.innerText = "The encrypted message is: "+encMsg.join(',
');
  user2DecRec.innerText = "The decrypted message is: "+decMsg;
});

user2Send.addEventListener('click',()⇒{
  var msg = user2Msg.value;
  var encMsg = encrypt(msg,q,user1PublicKey['h'],g,2);
  var decMsg = decrypt(encMsg,user2PublicKey['h'],user1PrivateKey,q);
  user1EncRec.innerText = "The encrypted message is: "+encMsg.join(',
');
  user1DecRec.innerText = "The decrypted message is: "+decMsg;
});
```

Github link →

https://github.com/kavyanshgangwar/netsec_assignment5