Lab 4

Question 1)
Approach : In order to call the ‘bkc_logging’ function we would create an event ‘log_data’ and would add an event listener to that event which will call the bkc_logging function. Full code is attached with the pdf.

```
var events = require('events');
var em = new events.EventEmitter();

em.on('logData',function(data){
  bkc_logging(data);
});
```

We will emit an event whenever user_login,user_logout,file_access ,file_permission_set is called.
Code :

1)User login :

```
this.user_Login = function(userId,pwd) {
   if(user[userId] == pwd)
   {
     loginStatus[userId] = 1;
     em.emit('logData','user with id '+userId+' successfully logged in');
   }
 }
```

2)user logout:

```
this.user_Logout = function(userId) {
   loginStatus[userId] = 0;
   em.emit('logData','user with id '+userId+' successfully logged in');
 }
```

3) File_Access :

```
this.file_Access = function(user) {
   if(loginStatus[user] == 1 && filePermissionBit[user] == 1)
   {
     em.emit('logData','user with id '+user+' accessed the file');
```

```
    return fileX;
  }
  em.emit('logData','user with id '+user+' failed to access file');
  return "You are not authorized to read this file.";
 }
}
```

4) `file_permission_set`

```
this.file_permission_set = function(user) {
  filePermissionBit[user] = 1;
  em.emit('logData','user with id '+user+' got the permission to read the file');
 }
```

Question 2)

We will use the web3.toHex method in order to convert a given string to hexa-decimal . After that we will use web3.eth.sendTransaction to send the transaction with data as a hex of the log.

Code :

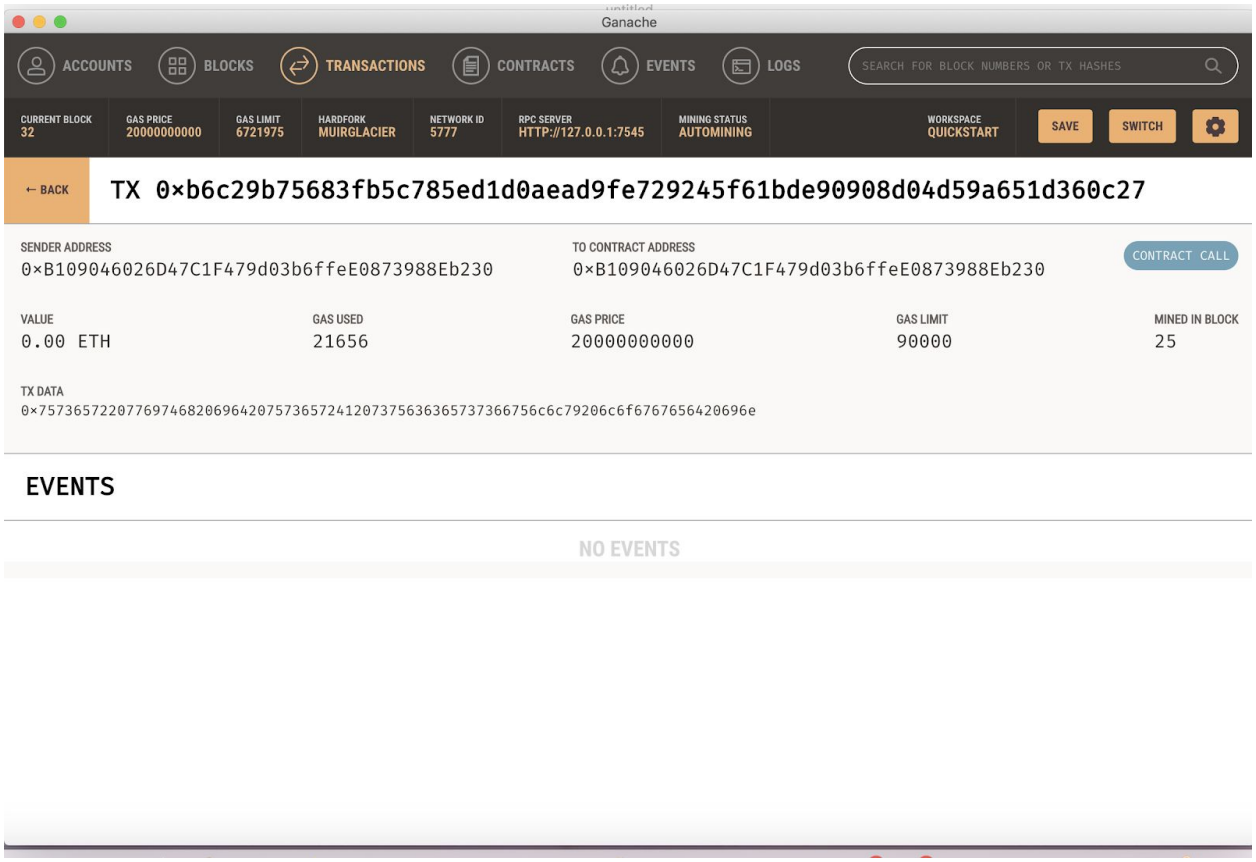
```
/*Function to generate hex encoded value for input string & sending transaction to
blockchain for logging puropse*/
function bkc_logging(str){
 let str_hex = web3.toHex(str);


web3.eth.sendTransaction({from:defaultAcc,data:str_hex,to:defaultAcc},function(err,suc
cess) {
  if (err)
    console.log("Failed to Send Transcation");
  else {
    console.log("sucsess");
    var receipt = web3.eth.getTransactionReceipt(success,function(err, transaction) {
      console.info(transaction);
    });
  }
 });
}
```

Execution:

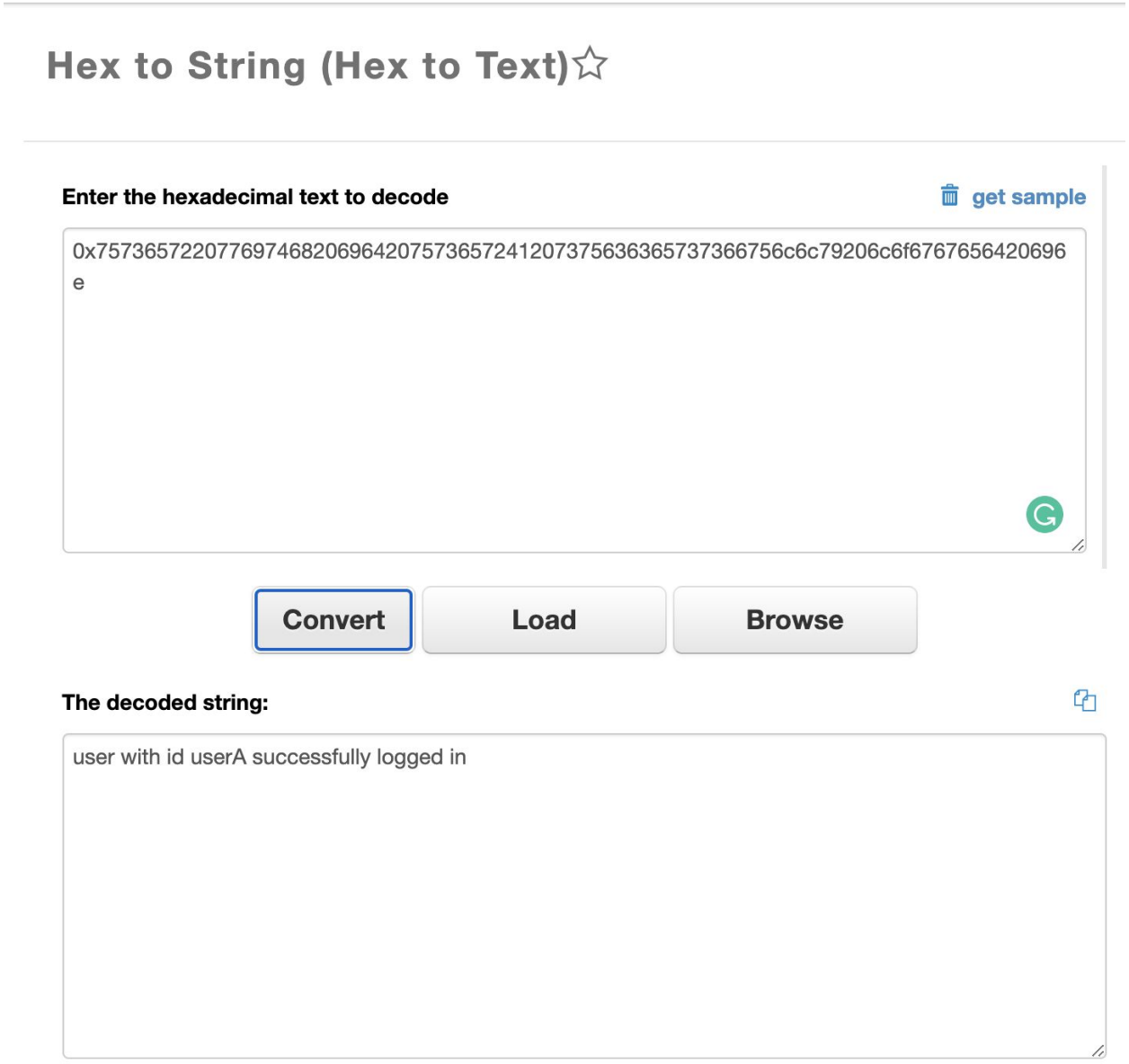
step1) Run code using node.js

```
Dhrumils-MBP:lab4-final dhrumil$ node question2.js
(node:4771) Warning: N-API is an experimental feature and could change at any time.
Response after userA reading file:This is the file from server
Response after userB reading file:You are not authorized to read this file.
userA giving file-read permission to userB
Response after userB reading file :This is the file from server
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
{ transactionHash: '0xb6c29b75683fb5c785ed1d0aead9fe729245f61bde90908d04d59a651d360c27',
  transactionIndex: 0,
  blockHash: '0x797064ee08fb6a5bfb855035fe2c19bc4ed5d0696634160d3814b2d8922c07e0',
  blockNumber: 25,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21656,
  cumulativeGasUsed: 21656,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0xf57879d3c95bd3a0e1289f2999abc09f20822a06c7f626d266885cf41f0d743a',
  transactionIndex: 0,
  blockHash: '0x994e315372a57f645fde34eeb09d9c160b0f2adb1766a3d927ec789fe137c22a',
  blockNumber: 26,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21656,
  cumulativeGasUsed: 21656,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x67b15b39faf29a1028aec7d725d5666e015e458d044cc0fb189d825f30e40b79',
  transactionIndex: 0,
  blockHash: '0xef66d9a89f643d639c6881cbd658db199f131b0084510fb0581c1ea97e7c57e8',
  blockNumber: 27,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21864,
  cumulativeGasUsed: 21864,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x40761d6d977eaac067698d167df17e5a8171ab9f041cfef0b65123cdbf39f51f',
  transactionIndex: 0,
  blockHash: '0xbafd317f596fdca6663c9747eaf795daa225d756e2939288d2a858bfd2e4d4da',
  blockNumber: 28,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21576,
```

Step2) check the given transaction on Ganache



step3) convert given hex data to string inorder to verify.



Question 3) In question 3 we will first convert the current log string to hexadecimal string . (using approach number 2 from lab-4 description). After that we would take 40 characters from string until possible and when string length is less than 0 then we would use padding to convert that remaining part to a valid address. Here is a code segment.

```
/*Function to generate hex encoded value for input string & sending transaction to
blockchain for logging puropse*/
function bkc_logging(str){
 let str_hex = web3.utils.toHex(str);
 var len = str_hex.length;
 var i = 2;
```

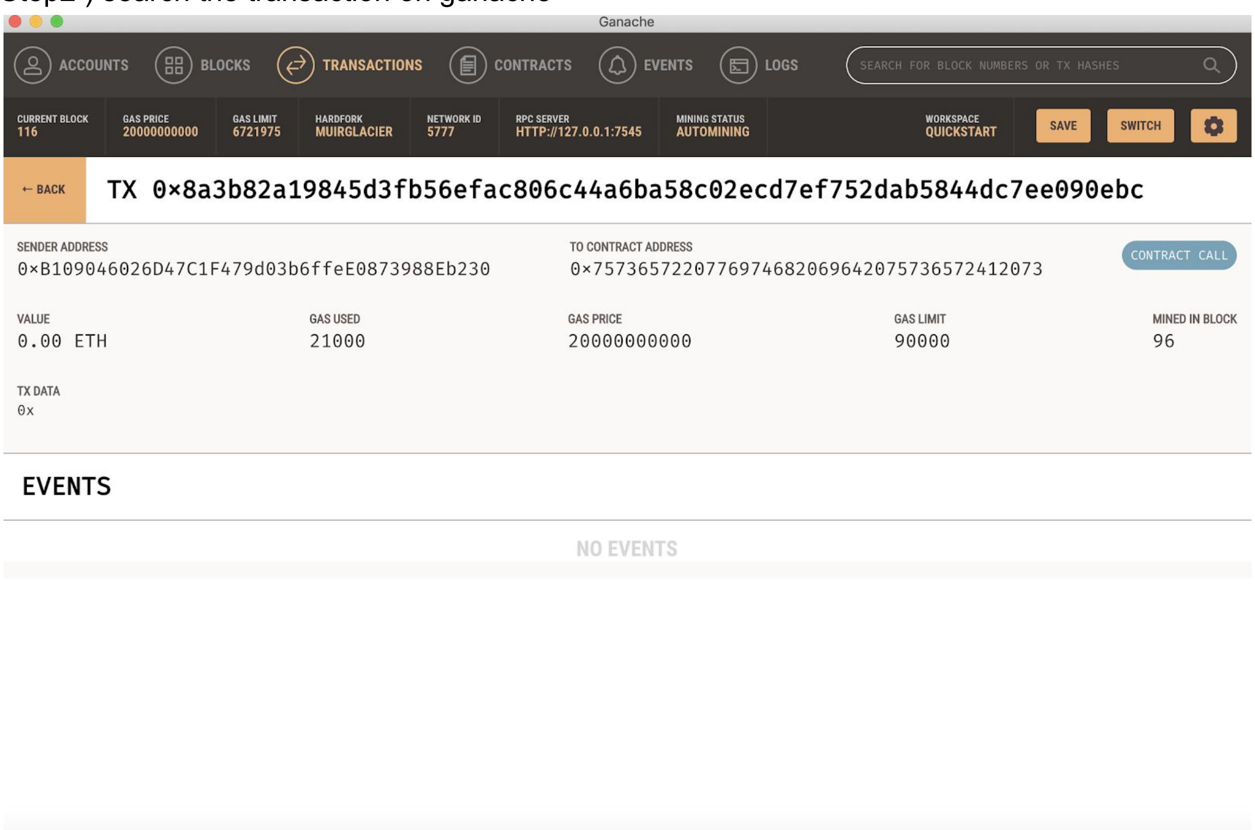
```
 while(i<len){
   var current = Math.min(i+40,len);
   var current_substring = str_hex.slice(i,current);
   var pdded_address = web3.utils.padLeft(current_substring, 40);
   var to_address = '0x'+pdded_address;
   console.log("default account");
   console.log(defaultAcc);
   console.log("To address");
   console.log(to_address);
   web3.eth.sendTransaction({from:defaultAcc,to:to_address},function(err,success) {
       if (err)
         console.log("Failed to Send Transcation");
       else {
         console.log("sucsess");
         var receipt = web3.eth.getTransactionReceipt(success,function(err,
transaction) {
           console.info(transaction);
         });
       }
     });
   i+=40;
 }
}
```

Execution:

step-1)

```
Dhrumils-MBP:lab4-final dhrumil$ node question3.js
this is program 3
(node:6521) Warning: N-API is an experimental feature and could change at any time.
Response after userA reading file:This is the file from server
Response after userB reading file:You are not authorized to read this file.
userA giving file-read permission to userB
Response after userB reading file :This is the file from server
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
{ transactionHash: '0x8a3b82a19845d3fb56efac806c44a6ba58c02ecd7ef752dab5844dc7ee090ebc',
  transactionIndex: 0,
  blockHash: '0x8730361706530b1e72ca7e9c871714c520e3578c714ff74dee8d2143f7d2bb60',
  blockNumber: 96,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0x7573657220776974682069642075736572412073',
  gasUsed: 21000,
  cumulativeGasUsed: 21000,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x6a3002065cf259655b5679dc36b1345379e61c0575165a3b4d1c6e9117edb4e3',
  transactionIndex: 0,
  blockHash: '0xcf57737b3dd8b5fbb5a875e99adcb5c7e2666a9ffe2b8804b1ee280a697576dd',
  blockNumber: 97,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0x75636365737366756c6c79206c6f676765642069',
  gasUsed: 21000,
  cumulativeGasUsed: 21000,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x76c0b479f082ce8be531665c8532495ac9623ea62a9aa381e4a8f258cc0cc1f8',
  transactionIndex: 0,
  blockHash: '0x459c85a8b0511a1acd174b598576765dfd5aebc7b947f8e8c389ea611b493458',
  blockNumber: 98,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
```

Step2) search the transaction on ganache



step-3) use to-contract address and convert that hex to string. As we can see in the image first part of the log is added as a to-address.

Question-4)
First we will change the client function to include only login and logout events.

```
function client(){
 server1=new server();
 this.execute = function() {
   server1.user_Login("userA","pwd123");
   server1.user_Login("userB","pwd456");

   server1.user_Logout("userA");
   server1.user_Logout("userB");
 }
}
```

After that we can use the bkc_logging function as implemented in question2 and question 3 in order to compare the gas cost.

Execution :
step1) using data field

```
Dhrumils-MBP:lab4-final dhrumil$ node question4.js
this is program 3
(node:6958) Warning: N-API is an experimental feature and could change at any time.
sucsess
sucsess
sucsess
sucsess
{ transactionHash: '0xc936c34e33493314e751c14dba41080cb7a021ac54f975985b5a96b884b6df78',
  transactionIndex: 0,
  blockHash: '0x3cf0d8c4fe23340e64d686b70883e7f03a5764af572de5d3cea2f94b9b5a1f63',
  blockNumber: 117,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21656,
  cumulativeGasUsed: 21656,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x15ffb4a302b75488fd4e8a6dbd801327e26edd746768d86c6e400be357218ae4',
  transactionIndex: 0,
  blockHash: '0xe28d0701b7248ee8076a65cc0864e2328e10bac894f2399bdea0ae6b84e78da5',
  blockNumber: 118,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21656,
  cumulativeGasUsed: 21656,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x0185b3a6813866e506dfee8e34fc1cbfb4ca5de0027ba9d8e211ba09792e5b59',
  transactionIndex: 0,
  blockHash: '0xfce31c3c6eac5656a4a750d4ecc42efa75abf013b70923222fed25dd281a8d98',
  blockNumber: 119,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21656,
  cumulativeGasUsed: 21656,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x772d9e5ab158eeb09370f779b7f6295fef19f21350d5f2ec4a5c6ba6df02b698',
  transactionIndex: 0,
  blockHash: '0xed2c6a1479e85d319fb6e0ca65f4c7363d001d652bb4c92f7c4b0c89a59d7999',
  blockNumber: 120,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
```

As we can see for all the cases total gas used was 21656.

step2) using to-address field

```
Dhrumils-MBP:lab4-final dhrumil$ node question4.js
this is program 3
(node:7059) Warning: N-API is an experimental feature and could change at any time.
user A log-in completed
user B log-in completed
user A log-out completed
user B log-out completed
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
sucsess
{ transactionHash: '0xdb26adad7b4d2981e4e1cf793df0358c96b42985d1351650764a01a64e613c8d',
  transactionIndex: 0,
  blockHash: '0x3f1c9a128f68a16e6536e0f4e6e5f0d5f5e899c8a14b85db5abb4514c31b64c1',
  blockNumber: 133,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0x7573657220776974682069642075736572412073',
  gasUsed: 21000,
  cumulativeGasUsed: 21000,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x1eea632fea58b6ae5c96cdfc96cd734e017c0b3798150c0563cc6876e599c580',
  transactionIndex: 0,
  blockHash: '0xf45e9a4545328fbbf75c7e96e41070d54724dd445b92950c57a5542a86a11be2',
  blockNumber: 134,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0x7563636573736756c6c79206c6f676765642069',
  gasUsed: 21000,
  cumulativeGasUsed: 21000,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
{ transactionHash: '0x0c02fed954b060d32e6dd0bce02f7d7c31913d662de35b8972f53fbdcd1cfd99',
  transactionIndex: 0,
  blockHash: '0xaa33551d456d484cb4c6a84253951a4ddcd2ade89b99c02975609f22a9db3260',
  blockNumber: 135,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0x000000000000000000000000000000000000006e',
  gasUsed: 21000,
  cumulativeGasUsed: 21000,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
```

As we can see all the transactions cost 21000 gas . So depending upon the length of the log data total cost could vary . But one conclusion we can derive is that transactions with data are more expensive.

Question 5)
Define a global variable named concatenated_log.

```
var concatenated_log = '';
```

Update the event handler to a method which appends a log to concatenated_log global string.

```
em.on('logData',function(data){
 append_log(data);
});
```

```
function append_log(str){
 concatenated_log = concatenated_log.concat(str);
}
```

Now update the client function which calls bkc logging at the end of all the function calls.

```
function client(){
 server1=new server();
 this.execute = function() {
   server1.user_Login("userA","pwd123");
   server1.user_Login("userB","pwd456");

   server1.file_permission_set("userA");
   var response = server1.file_Access("userA");
   console.log("Response after userA reading file:"+response);
   response=server1.file_Access("userB");
   console.log("Response after userB reading file:"+response);

   server1.file_delegate("userA","userB");
   response = server1.file_Access("userB");
   console.log("Response after userB reading file :"+response);

   server1.user_Logout("userA");
   server1.user_Logout("userB");
   bkc_logging();
 }
}
```

Now instead of parameter bkc_logging will use global concatenated log variable.

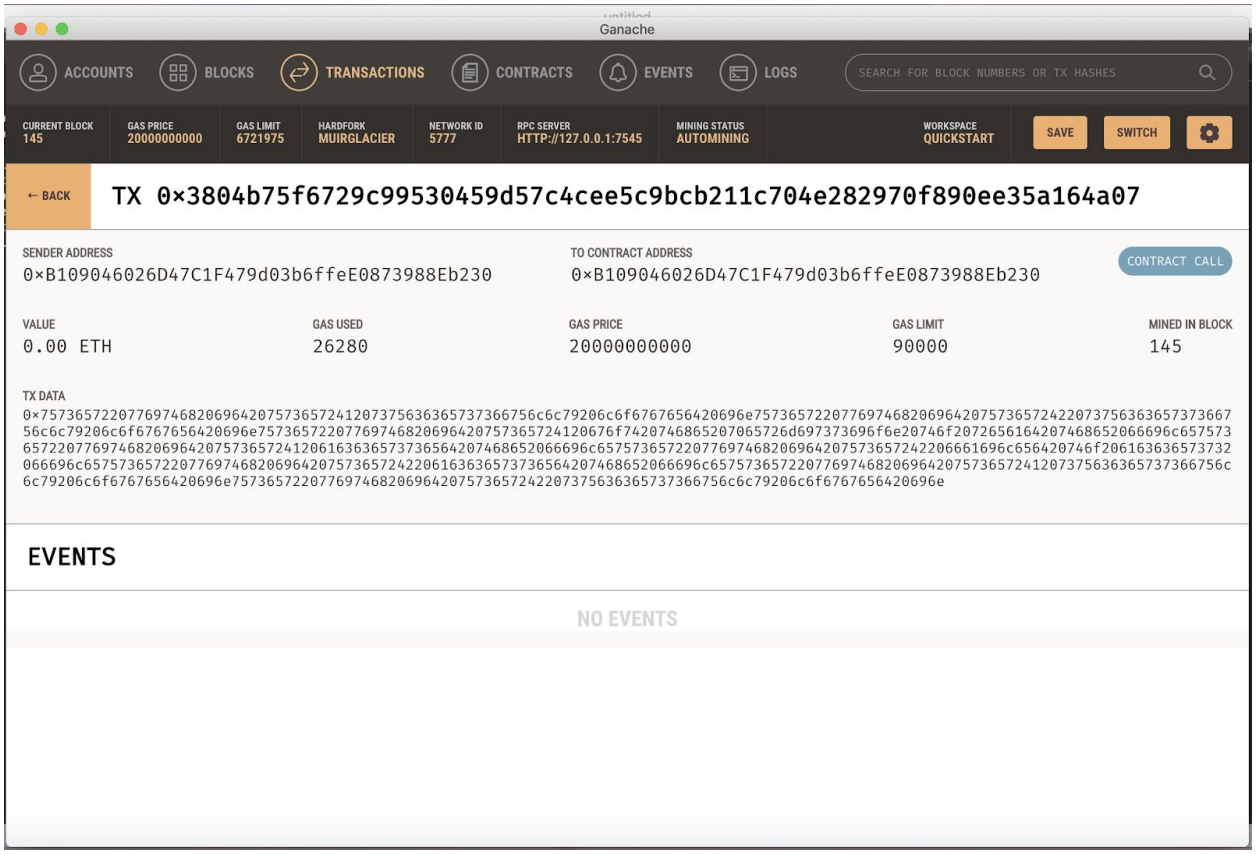
```
/*Function to generate hex encoded value for input string & sending transaction to
blockchain for logging puropse*/
function bkc_logging(){
 let str_hex = web3.utils.toHex(concatenated_log);

web3.eth.sendTransaction({from:defaultAcc,data:str_hex,to:defaultAcc},function(err,suc
cess) {
   if (err)
     console.log("Failed to Send Transcation");
   else {
     console.log("sucsess");
     var receipt = web3.eth.getTransactionReceipt(success,function(err, transaction) {
       console.info(transaction);
     });
   }
 });
}
```

Execution:

step1) running node command

```
(node:7188) Warning: N-API is an experimental feature and could change at any time.
Response after userA reading file:This is the file from server
Response after userB reading file:You are not authorized to read this file.
userA giving file-read permission to userB
Response after userB reading file :This is the file from server
sucsess
{ transactionHash: '0x3804b75f6729c99530459d57c4cee5c9bcb211c704e282970f890ee35a164a07',
  transactionIndex: 0,
  blockHash: '0x2184207d544baeaf2997a66925a162d10579cf49b49b1a77c50c64f54d63d35a',
  blockNumber: 145,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 26280,
  cumulativeGasUsed: 26280,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
Dhrumils-MBP:lab4-final dhrumil$ 
```

step2) verify data in Ganache



step3) Gas cost :
Total 26280 gas was used during the entire transaction.

Question 6)
As we did in the previous question we will create an event handler append log which will merge the current hash and log and rehash it again.

```
function append_log(str){
   concatenated_log = concatenated_log.concat(str);
   concatenated_log =
crypto.createHash('md5').update(concatenated_log).digest('hex');
}
```

Bkc_log function would use the concatenated log to send the transaction.

```
/*Function to generate hex encoded value for input string & sending transaction to
blockchain for logging puropse*/
function bkc_logging(){
 let str_hex = concatenated_log;
```

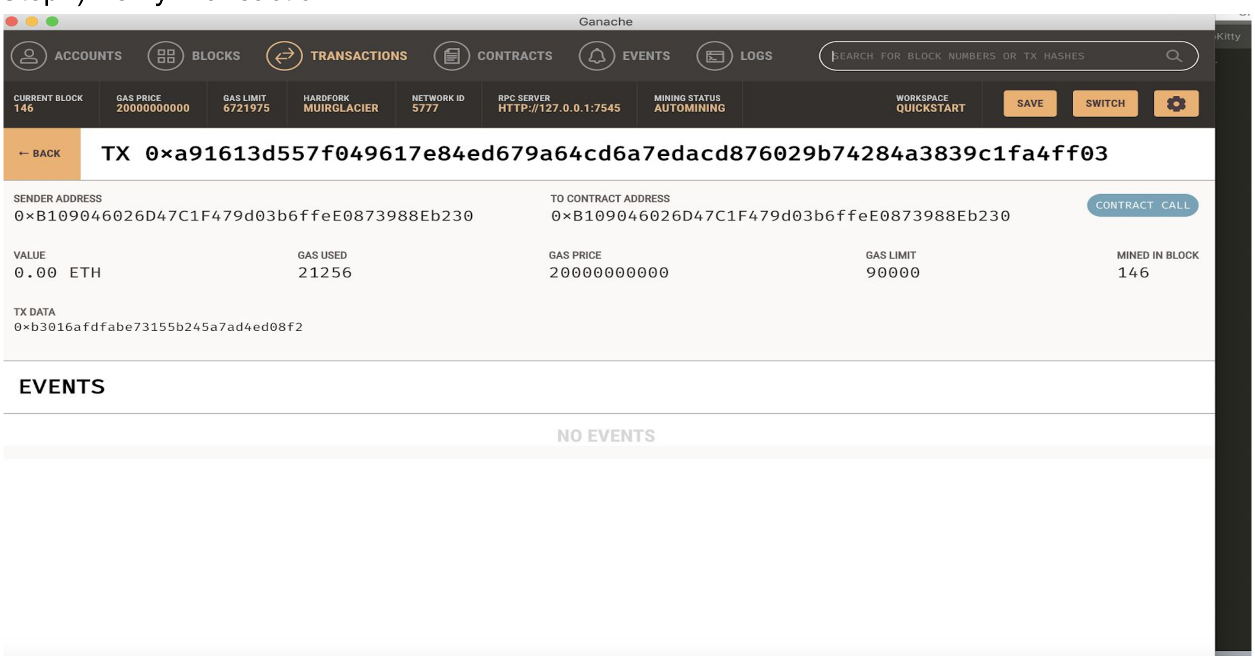
```
web3.eth.sendTransaction({from:defaultAcc,data:str_hex,to:defaultAcc},function(err,suc
cess) {
   if (err)
     console.log("Failed to Send Transcation");
   else {
     console.log("sucsess");
     var receipt = web3.eth.getTransactionReceipt(success,function(err, transaction) {
       console.info(transaction);
     });
   }
 });
}
```

Execution :

Step1)

```
Dhrumils-MBP:lab4-final dhrumil$ node question6.js
(node:7311) Warning: N-API is an experimental feature and could change at any time.
Response after userA reading file:This is the file from server
Response after userB reading file:You are not authorized to read this file.
userA giving file-read permission to userB
Response after userB reading file :This is the file from server
sucsess
{ transactionHash: '0xa91613d557f049617e84ed679a64cd6a7edacd876029b74284a3839c1fa4ff03',
  transactionIndex: 0,
  blockHash: '0x7f26f4cc159a0b025967c52365bd3773e0f23d8909a535a162db89eb3e4fb51a',
  blockNumber: 146,
  from: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  to: '0xb109046026d47c1f479d03b6ffee0873988eb230',
  gasUsed: 21256,
  cumulativeGasUsed: 21256,
  contractAddress: null,
  logs: [],
  status: true,
  logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' }
Dhrumils-MBP:lab4-final dhrumil$ 
```

Step2) Verify Transaction



Conclusion : As we can see gas cost reduced from 26280 to 21256 due to reduced data size thanks to hashing.