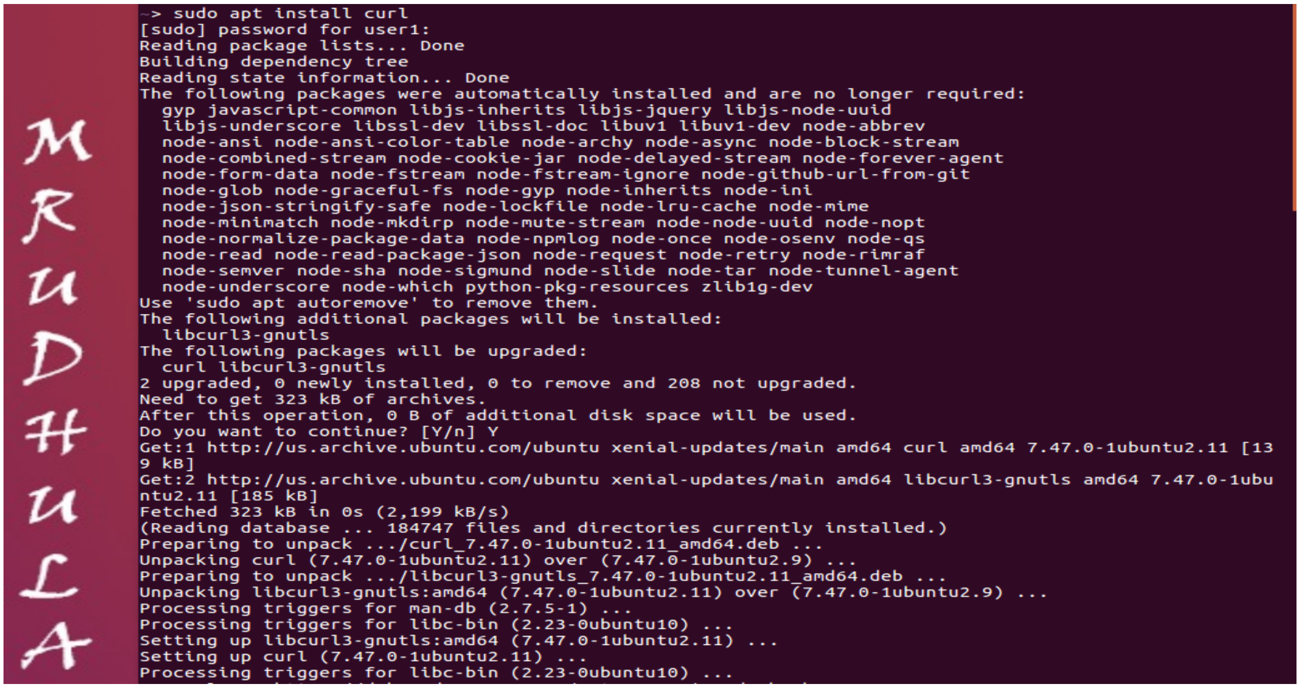
Mrudhula Ashok Shenava 265469976

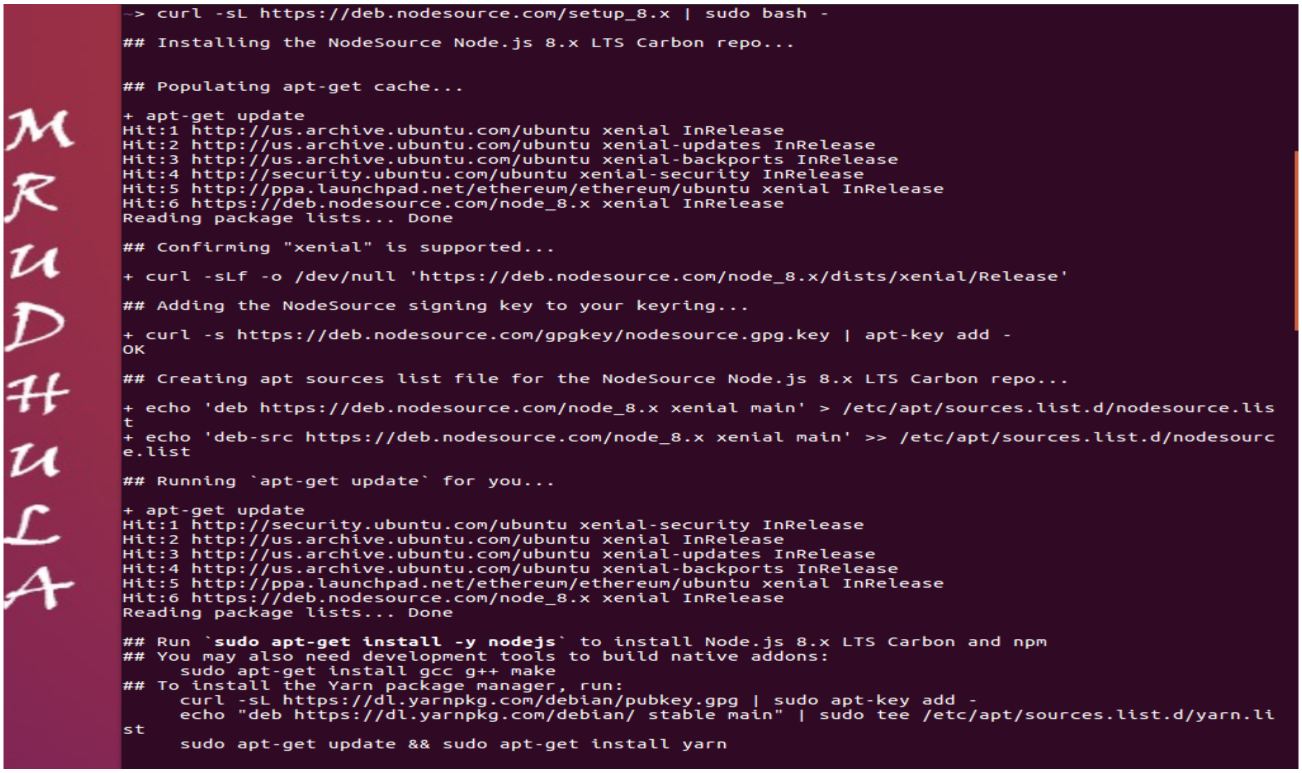
BLOCKCHAIN Applications: LOGGING REMOTE FILE STORAGE

INITIAL SETUP

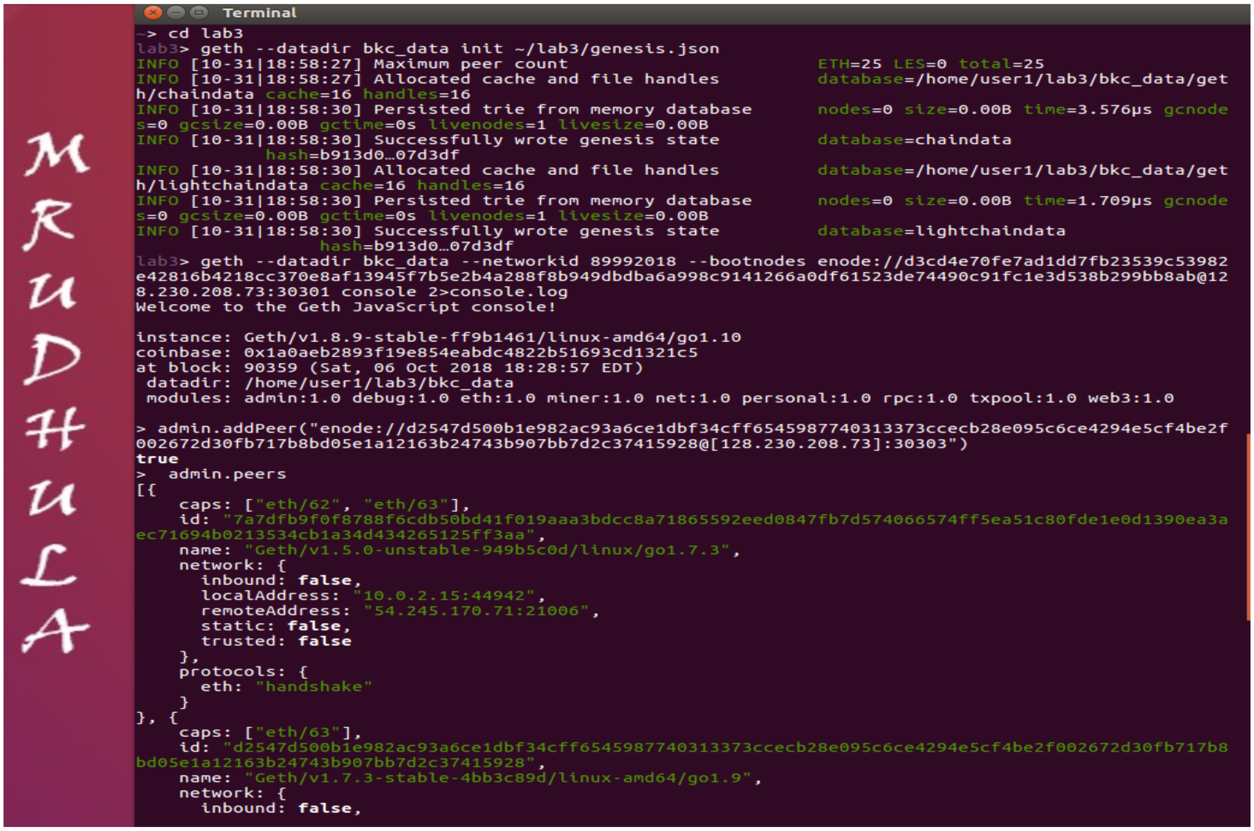
TERMINAL1



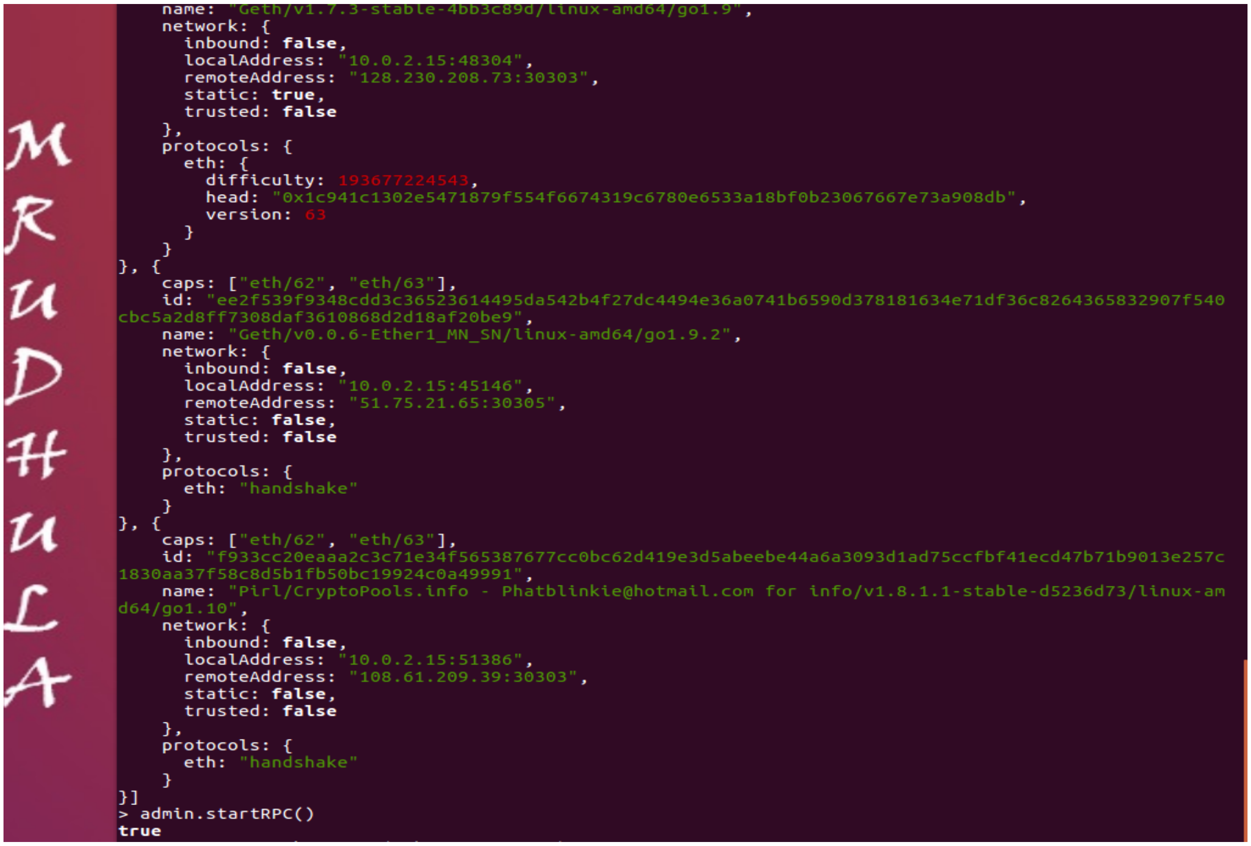
```
~> sudo apt install curl
[sudo] password for user1:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gyp javascript-common libjs-inherits libjs-jquery libjs-node-uuid
  libjs-underscore libssl-dev libssl-doc libuv1 libuv1-dev node-abbrev
  node-ansi node-ansi-color-table node-archy node-async node-block-stream
  node-combined-stream node-cookie-jar node-delayed-stream node-forever-agent
  node-form-data node-fstream node-fstream-ignore node-github-url-from-git
  node-glob node-graceful-fs node-gyp node-inherits node-ini
  node-json-stringify-safe node-lockfile node-lru-cache node-mime
  node-minimatch node-mkdirp node-mute-stream node-node-uuid node-nopt
  node-normalize-package-data node-npmlog node-once node-osenv node-qs
  node-read node-read-package-json node-request node-retry node-rimraf
  node-semver node-sha node-sigmund node-slide node-tar node-tunnel-agent
  node-underscore node-which python-pkg-resources zlib1g-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcurl3-gnutls
The following packages will be upgraded:
  curl libcurl3-gnutls
2 upgraded, 0 newly installed, 0 to remove and 208 not upgraded.
Need to get 323 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 curl amd64 7.47.0-1ubuntu2.11 [13
9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libcurl3-gnutls amd64 7.47.0-1ubu
ntu2.11 [185 kB]
Fetched 323 kB in 0s (2,199 kB/s)
(Reading database ... 184747 files and directories currently installed.)
Preparing to unpack .../curl_7.47.0-1ubuntu2.11_amd64.deb ...
Unpacking curl (7.47.0-1ubuntu2.11) over (7.47.0-1ubuntu2.9) ...
Preparing to unpack .../libcurl3-gnutls_7.47.0-1ubuntu2.11_amd64.deb ...
Unpacking libcurl3-gnutls:amd64 (7.47.0-1ubuntu2.11) over (7.47.0-1ubuntu2.9) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Setting up libcurl3-gnutls:amd64 (7.47.0-1ubuntu2.11) ...
Setting up curl (7.47.0-1ubuntu2.11) ...
Processing triggers for libc-bin (2.23-0ubuntu10) ...
```



```
~> curl -sL https://deb.nodesource.com/setup_8.x | sudo bash -

## Installing the NodeSource Node.js 8.x LTS Carbon repo...


## Populating apt-get cache...

+ apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:5 http://ppa.launchpad.net/ethereum/ethereum/ubuntu xenial InRelease
Hit:6 https://deb.nodesource.com/node_8.x xenial InRelease
Reading package lists... Done

## Confirming "xenial" is supported...

+ curl -sLf -o /dev/null 'https://deb.nodesource.com/node_8.x/dists/xenial/Release'

## Adding the NodeSource signing key to your keyring...

+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | apt-key add -
OK

## Creating apt sources list file for the NodeSource Node.js 8.x LTS Carbon repo...

+ echo 'deb https://deb.nodesource.com/node_8.x xenial main' > /etc/apt/sources.list.d/nodesource.lis
t
+ echo 'deb-src https://deb.nodesource.com/node_8.x xenial main' >> /etc/apt/sources.list.d/nodesourc
e.list

## Running `apt-get update` for you...

+ apt-get update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:5 http://ppa.launchpad.net/ethereum/ethereum/ubuntu xenial InRelease
Hit:6 https://deb.nodesource.com/node_8.x xenial InRelease
Reading package lists... Done

## Run `sudo apt-get install -y nodejs` to install Node.js 8.x LTS Carbon and npm
## You may also need development tools to build native addons:
     sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
     curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
     echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.li
st
     sudo apt-get update && sudo apt-get install yarn
```

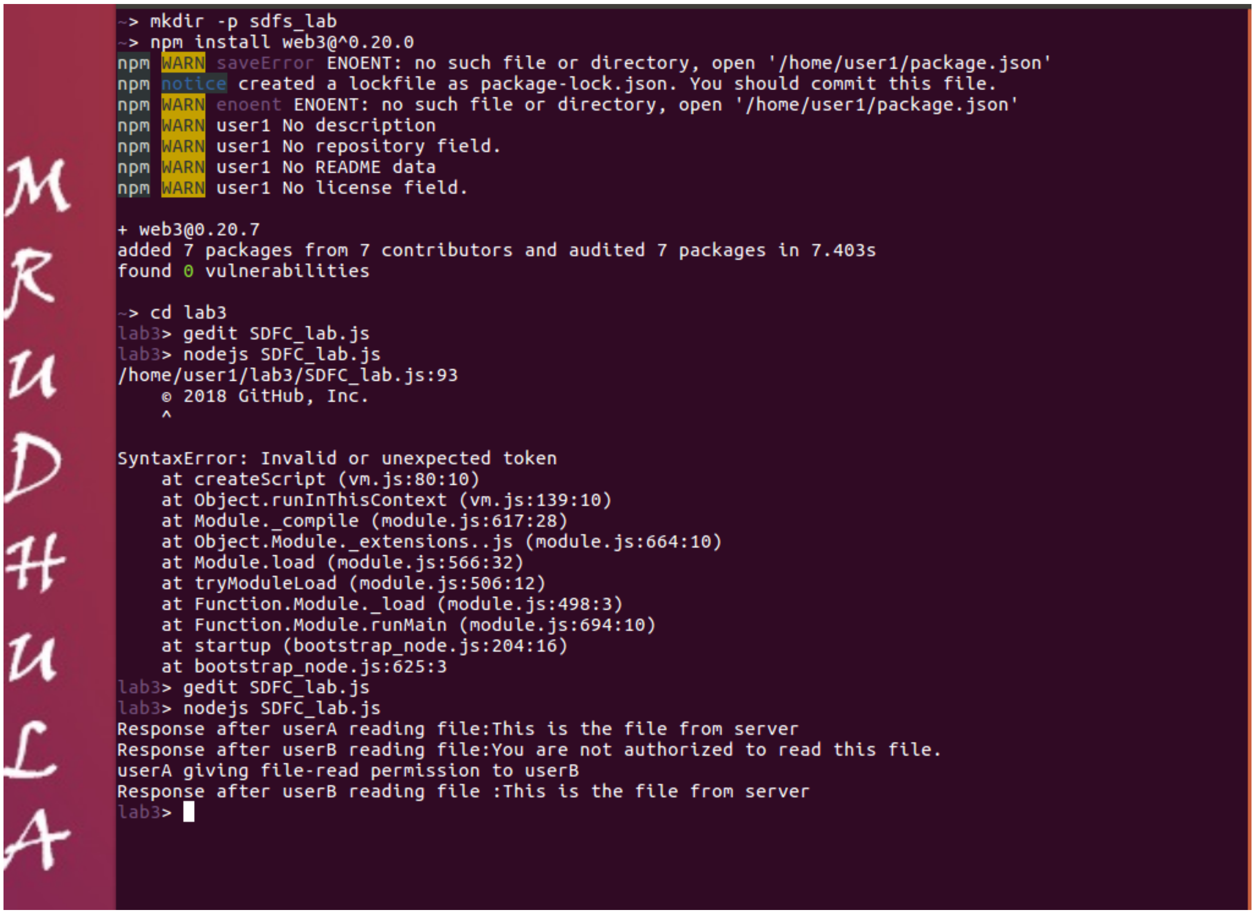


```
~> cd lab3
lab3> geth --datadir bkc_data init ~/lab3/genesis.json
INFO [10-31|18:58:27] Maximum peer count                       ETH=25 LES=0 total=25
INFO [10-31|18:58:27] Allocated cache and file handles         database=/home/user1/lab3/bkc_data/get
h/chaindata cache=16 handles=16
INFO [10-31|18:58:30] Persisted trie from memory database      nodes=0 size=0.00B time=3.576µs gcnode
s=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [10-31|18:58:30] Successfully wrote genesis state         database=chaindata
            hash=b913d0…07d3df
INFO [10-31|18:58:30] Allocated cache and file handles         database=/home/user1/lab3/bkc_data/get
h/lightchaindata cache=16 handles=16
INFO [10-31|18:58:30] Persisted trie from memory database      nodes=0 size=0.00B time=1.709µs gcnode
s=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [10-31|18:58:30] Successfully wrote genesis state         database=lightchaindata
                 hash=b913d0…07d3df
lab3> geth --datadir bkc_data --networkid 89992018 --bootnodes enode://d3cd4e70fe7ad1dd7fb23539c53982
e42816b4218cc370e8af13945f7b5e2b4a288f8b949dbdba6a998c9141266a0df61523de74490c91fc1e3d538b299bb8ab@12
8.230.208.73:30301 console 2>console.log
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.9-stable-ff9b1461/linux-amd64/go1.10
coinbase: 0x1a0aeb2893f19e854eabdc4822b51693cd1321c5
at block: 90359 (Sat, 06 Oct 2018 18:28:57 EDT)
 datadir: /home/user1/lab3/bkc_data
 modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> admin.addPeer("enode://d2547d500b1e982ac93a6ce1dbf34cff6545987740313373ccecb28e095c6ce4294e5cf4be2f
002672d30fb717b8bd05e1a12163b24743b907bb7d2c37415928@[128.230.208.73]:30303")
true
>  admin.peers
[{
    caps: ["eth/62", "eth/63"],
    id: "7a7dfb9f0f8788f6cdb50bd41f019aaa3bdcc8a71865592eed0847fb7d574066574ff5ea51c80fde1e0d1390ea3a
ec71694b0213534cb1a34d434265125ff3aa",
    name: "Geth/v1.5.0-unstable-949b5c0d/linux/go1.7.3",
    network: {
      inbound: false,
      localAddress: "10.0.2.15:44942",
      remoteAddress: "54.245.170.71:21006",
      static: false,
      trusted: false
    },
    protocols: {
      eth: "handshake"
    }
}, {
    caps: ["eth/63"],
    id: "d2547d500b1e982ac93a6ce1dbf34cff6545987740313373ccecb28e095c6ce4294e5cf4be2f002672d30fb717b8
bd05e1a12163b24743b907bb7d2c37415928",
    name: "Geth/v1.7.3-stable-4bb3c89d/linux-amd64/go1.9",
    network: {
      inbound: false,
```



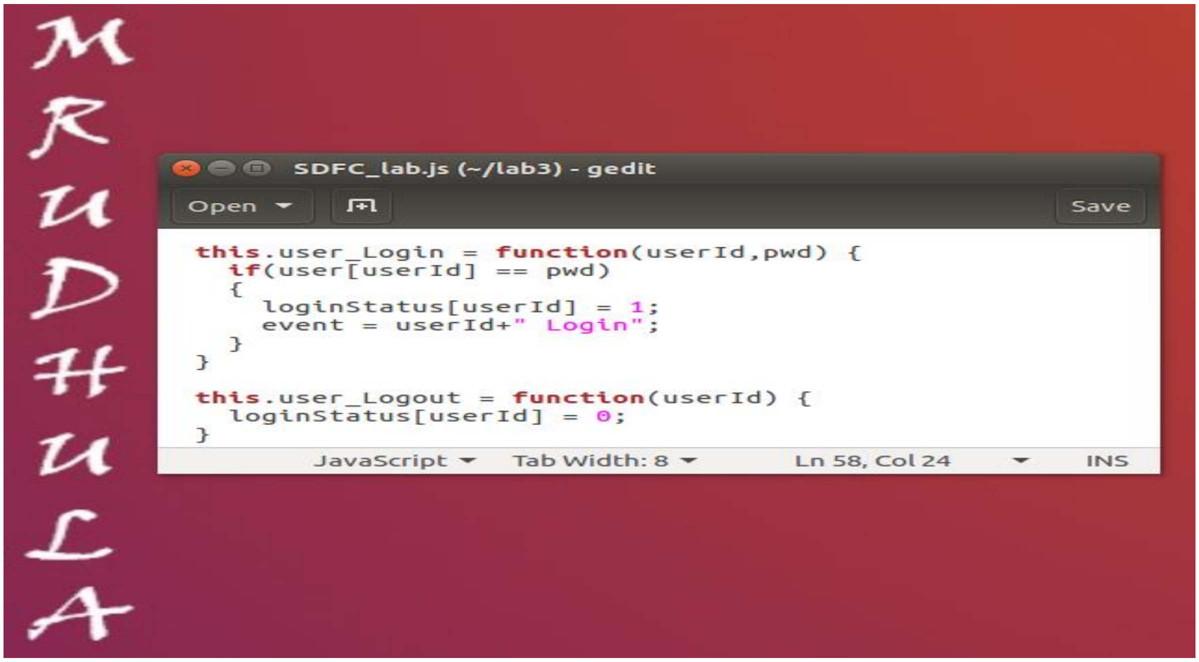
```
    name: "Geth/v1.7.3-stable-4bb3c89d/linux-amd64/go1.9",
    network: {
      inbound: false,
      localAddress: "10.0.2.15:48304",
      remoteAddress: "128.230.208.73:30303",
      static: true,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 193677224543,
        head: "0x1c941c1302e5471879f554f6674319c6780e6533a18bf0b23067667e73a908db",
        version: 63
      }
    }
}, {
    caps: ["eth/62", "eth/63"],
    id: "ee2f539f9348cdd3c36523614495da542b4f27dc4494e36a0741b6590d378181634e71df36c8264365832907f540
cbc5a2d8ff7308daf3610868d2d18af20be9",
    name: "Geth/v0.0.6-Ether1_MN_SN/linux-amd64/go1.9.2",
    network: {
      inbound: false,
      localAddress: "10.0.2.15:45146",
      remoteAddress: "51.75.21.65:30305",
      static: false,
      trusted: false
    },
    protocols: {
      eth: "handshake"
    }
}, {
    caps: ["eth/62", "eth/63"],
    id: "f933cc20eaaa2c3c71e34f565387677cc0bc62d419e3d5abeebe44a6a3093d1ad75ccfbf41ecd47b71b9013e257c
1830aa37f58c8d5b1fb50bc19924c0a49991",
    name: "Pirl/CryptoPools.info - Phatblinkie@hotmail.com for info/v1.8.1.1-stable-d5236d73/linux-am
d64/go1.10",
    network: {
      inbound: false,
      localAddress: "10.0.2.15:51386",
      remoteAddress: "108.61.209.39:30303",
      static: false,
      trusted: false
    },
    protocols: {
      eth: "handshake"
    }
}]
> admin.startRPC()
true
```

TERMINAL2



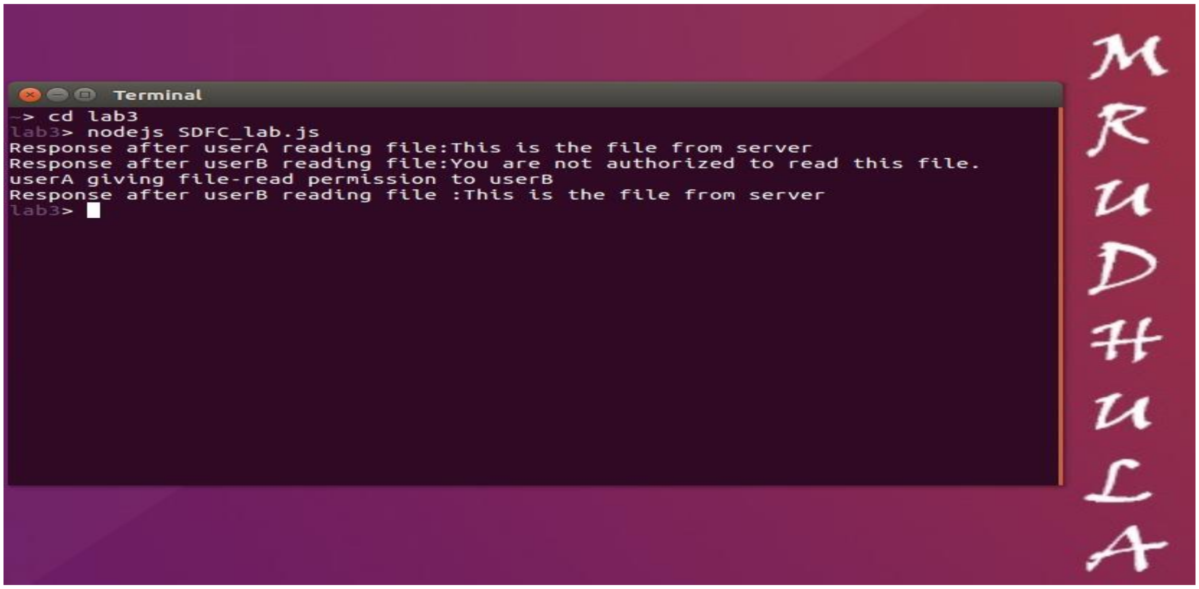
```
~> mkdir -p sdfs_lab
~> npm install web3@^0.20.0
npm WARN saveError ENOENT: no such file or directory, open '/home/user1/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open '/home/user1/package.json'
npm WARN user1 No description
npm WARN user1 No repository field.
npm WARN user1 No README data
npm WARN user1 No license field.

+ web3@0.20.7
added 7 packages from 7 contributors and audited 7 packages in 7.403s
found 0 vulnerabilities

~> cd lab3
lab3> gedit SDFC_lab.js
lab3> nodejs SDFC_lab.js
/home/user1/lab3/SDFC_lab.js:93
    © 2018 GitHub, Inc.
    ^

SyntaxError: Invalid or unexpected token
    at createScript (vm.js:80:10)
    at Object.runInThisContext (vm.js:139:10)
    at Module._compile (module.js:617:28)
    at Object.Module._extensions..js (module.js:664:10)
    at Module.load (module.js:566:32)
    at tryModuleLoad (module.js:506:12)
    at Function.Module._load (module.js:498:3)
    at Function.Module.runMain (module.js:694:10)
    at startup (bootstrap_node.js:204:16)
    at bootstrap_node.js:625:3
lab3> gedit SDFC_lab.js
lab3> nodejs SDFC_lab.js
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
lab3> 
```

TASK 1: USER LOGIN/LOGOUT



```
this.user_Login = function(userId,pwd) {
  if(user[userId] == pwd)
  {
    loginStatus[userId] = 1;
    event = userId+" Login";
  }
}

this.user_Logout = function(userId) {
  loginStatus[userId] = 0;
}
```

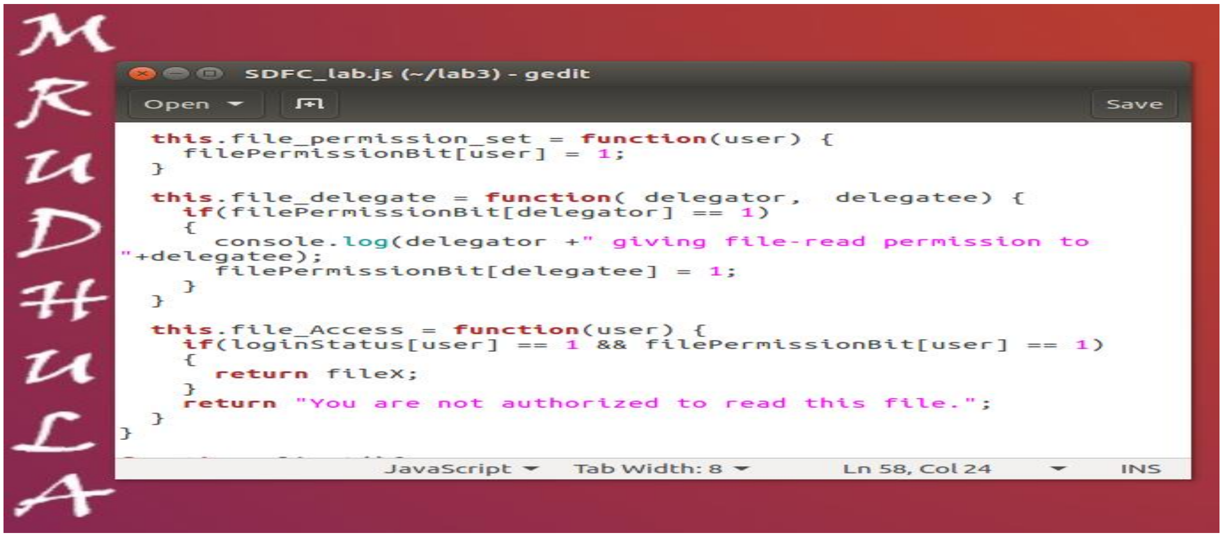
We run the SDFC_lab.js using the command line nodejs SDFC_lab.js. The screenshot of the code is given above for this task.



```
~> cd lab3
lab3> nodejs SDFC_lab.js
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
lab3> 
```

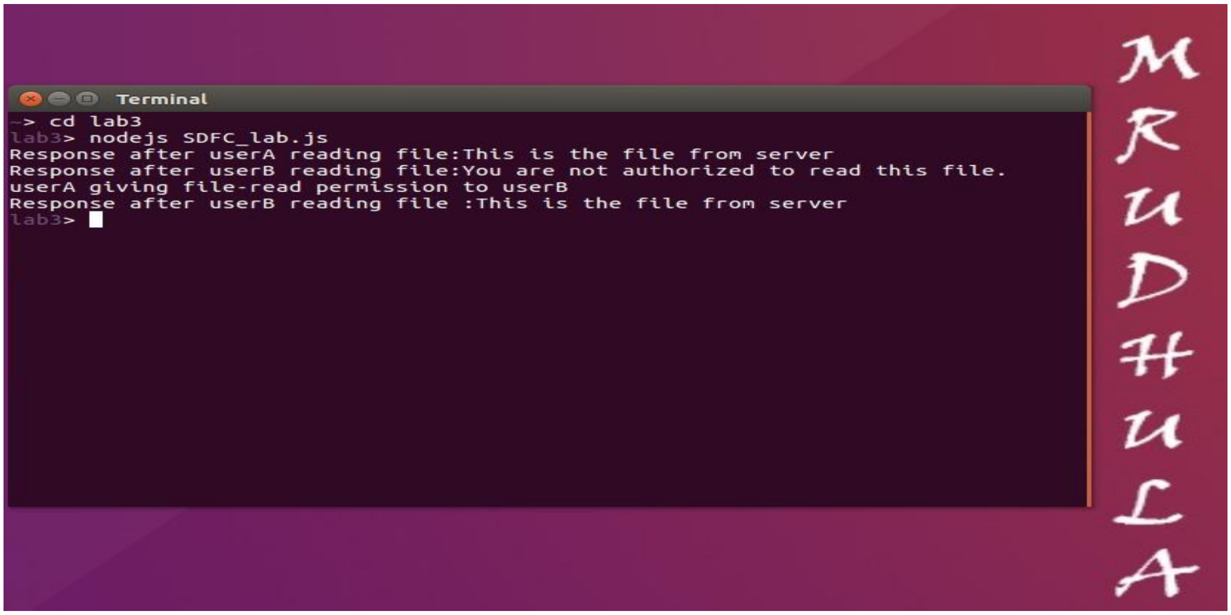
This program, SDFC_lab.js successfully executes user_login and user_logout functions. UserA and UserB successfully logged in remotely and logged out in the sequence of User_login, file_access, file_permission_bit, user_logout.

TASK 2: ACCESS CONTROL BY PERMISSION



```
    this.file_permission_set = function(user) {
      filePermissionBit[user] = 1;
    }

    this.file_delegate = function( delegator,  delegatee) {
      if(filePermissionBit[delegator] == 1)
      {
        console.log(delegator +" giving file-read permission to
"+delegatee);
        filePermissionBit[delegatee] = 1;
      }
    }

    this.file_Access = function(user) {
      if(loginStatus[user] == 1 && filePermissionBit[user] == 1)
      {
        return fileX;
      }
      return "You are not authorized to read this file.";
    }
}
```

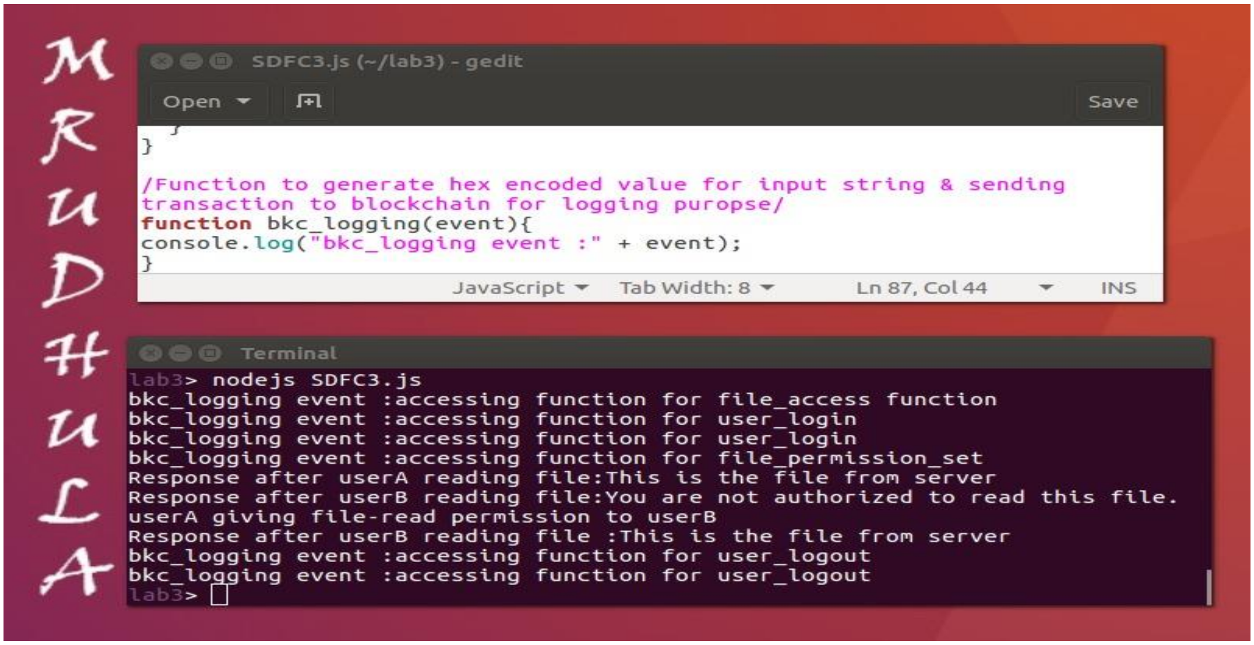
Above screenshot is the part of code used for this task.



```
~> cd lab3
lab3> nodejs SDFC_lab.js
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
lab3> 
```

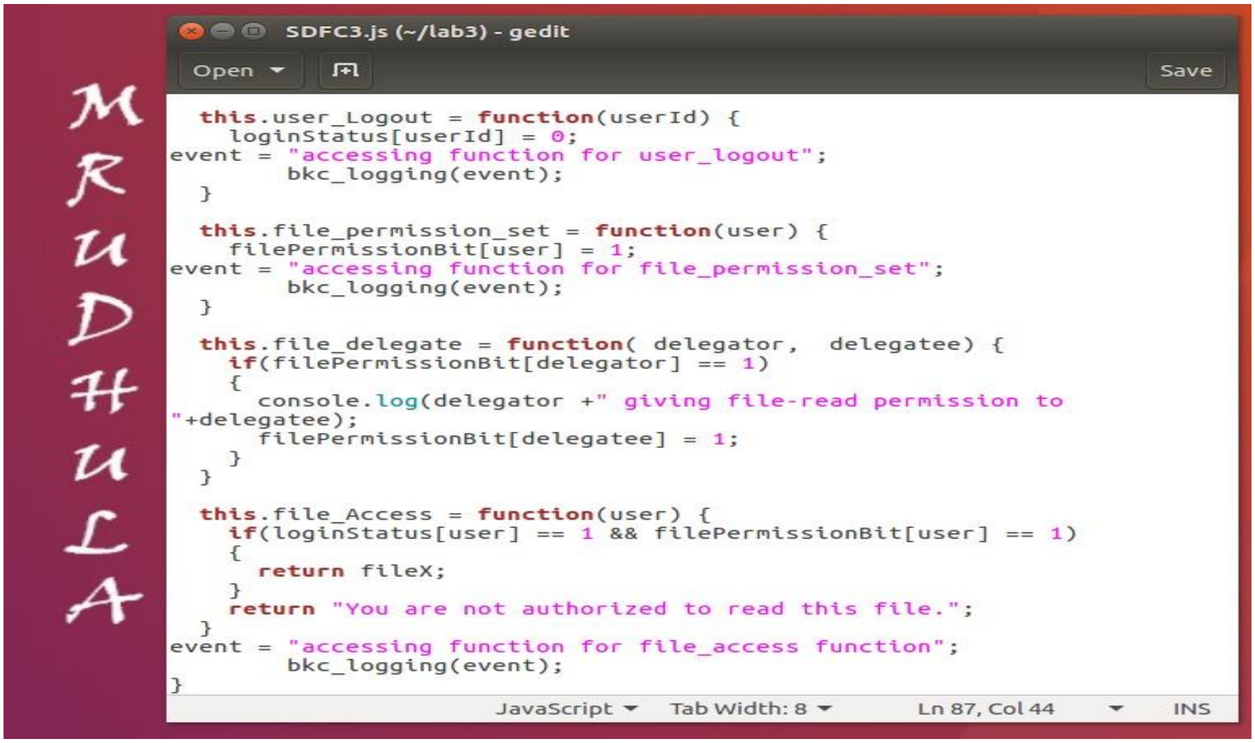
For UserA file access bit was set to 1 in the code which is printed on the console for access to the file and UserB does not have access to the file. for UserA file access bit was set to 1 and file-read permission was given for UserB and therefore, userB has access to read file now as seen in above screenshot.

TASK 3: TRACING REMOTE REQUESTS



```
}

/Function to generate hex encoded value for input string & sending
transaction to blockchain for logging puropse/
function bkc_logging(event){
console.log("bkc_logging event :" + event);
}
```

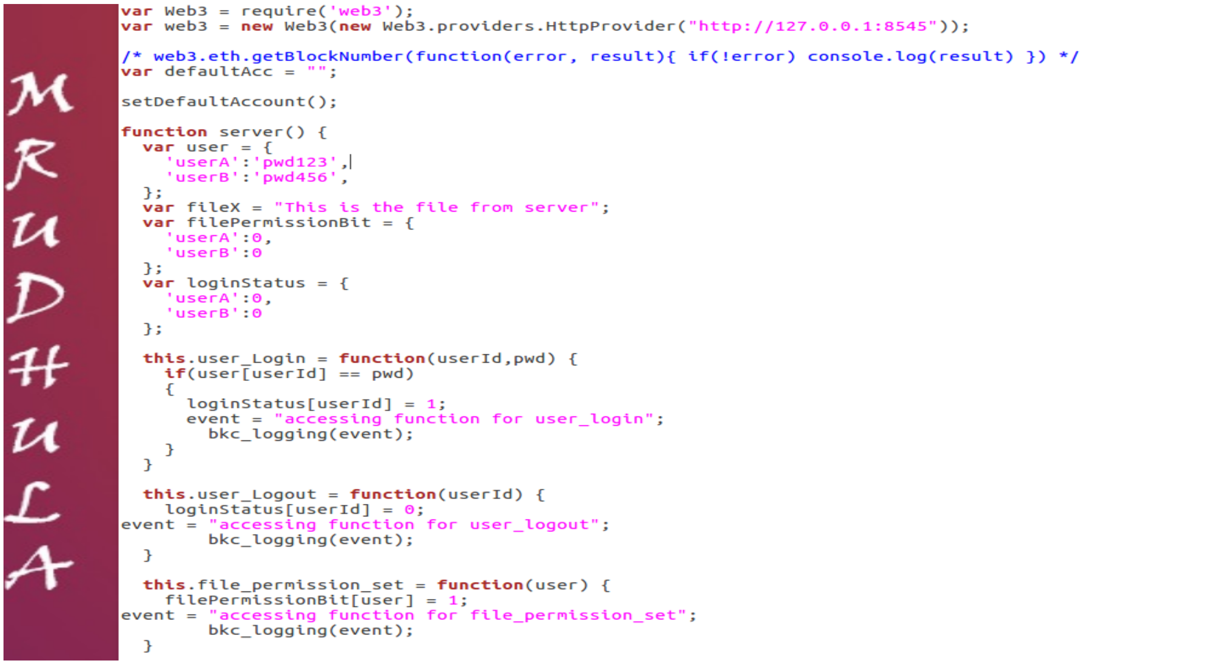
```
lab3> nodejs SDFC3.js
bkc_logging event :accessing function for file_access function
bkc_logging event :accessing function for user_login
bkc_logging event :accessing function for user_login
bkc_logging event :accessing function for file_permission_set
Response after userA reading file:This is the 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
bkc_logging event :accessing function for user_logout
bkc_logging event :accessing function for user_logout
lab3> 
```



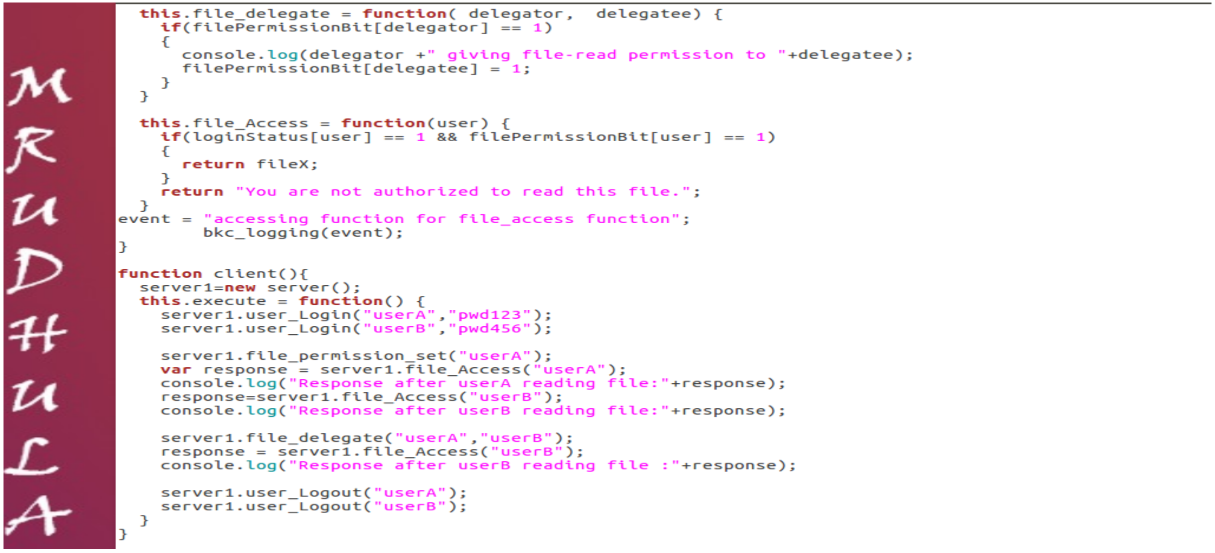
```
  this.user_Logout = function(userId) {
    loginStatus[userId] = 0;
event = "accessing function for user_logout";
        bkc_logging(event);
  }

  this.file_permission_set = function(user) {
    filePermissionBit[user] = 1;
event = "accessing function for file_permission_set";
        bkc_logging(event);
  }

  this.file_delegate = function( delegator,  delegatee) {
    if(filePermissionBit[delegator] == 1)
    {
      console.log(delegator +" giving file-read permission to
"+delegatee);
      filePermissionBit[delegatee] = 1;
    }
  }

  this.file_Access = function(user) {
    if(loginStatus[user] == 1 && filePermissionBit[user] == 1)
    {
      return fileX;
    }
    return "You are not authorized to read this file.";
  }
event = "accessing function for file_access function";
        bkc_logging(event);
}
```

The above screenshot we notice that the event for given call functions user_login(), user_logout(), file_access() and file_permission_set() was created and passed to call function bkc_logging() as an event. This logs the event and we see this is printed on console to show calls of different functions.

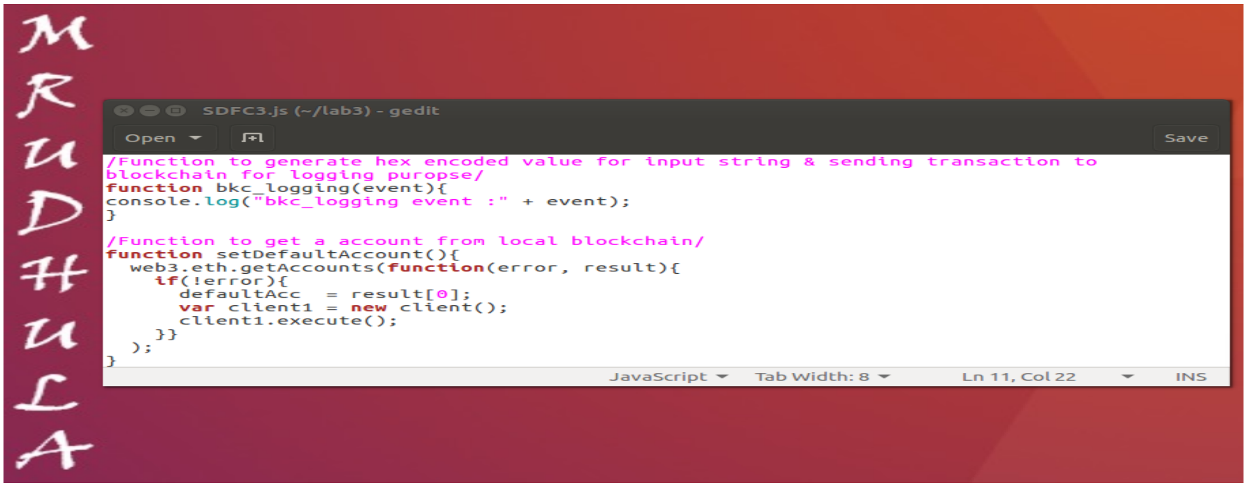
Below is the complete code used for this task.



```
var Web3 = require('web3');
var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8545"));

/* web3.eth.getBlockNumber(function(error, result){ if(!error) console.log(result) }) */
var defaultAcc = "";

setDefaultAccount();

function server() {
  var user = {
    'userA':'pwd123',
    'userB':'pwd456',
  };
  var fileX = "This is the file from server";
  var filePermissionBit = {
    'userA':0,
    'userB':0
  };
  var loginStatus = {
    'userA':0,
    'userB':0
  };

  this.user_Login = function(userId,pwd) {
    if(user[userId] == pwd)
    {
      loginStatus[userId] = 1;
      event = "accessing function for user_login";
        bkc_logging(event);
    }
  }

  this.user_Logout = function(userId) {
    loginStatus[userId] = 0;
event = "accessing function for user_logout";
        bkc_logging(event);
  }

  this.file_permission_set = function(user) {
    filePermissionBit[user] = 1;
event = "accessing function for file_permission_set";
        bkc_logging(event);
  }
```

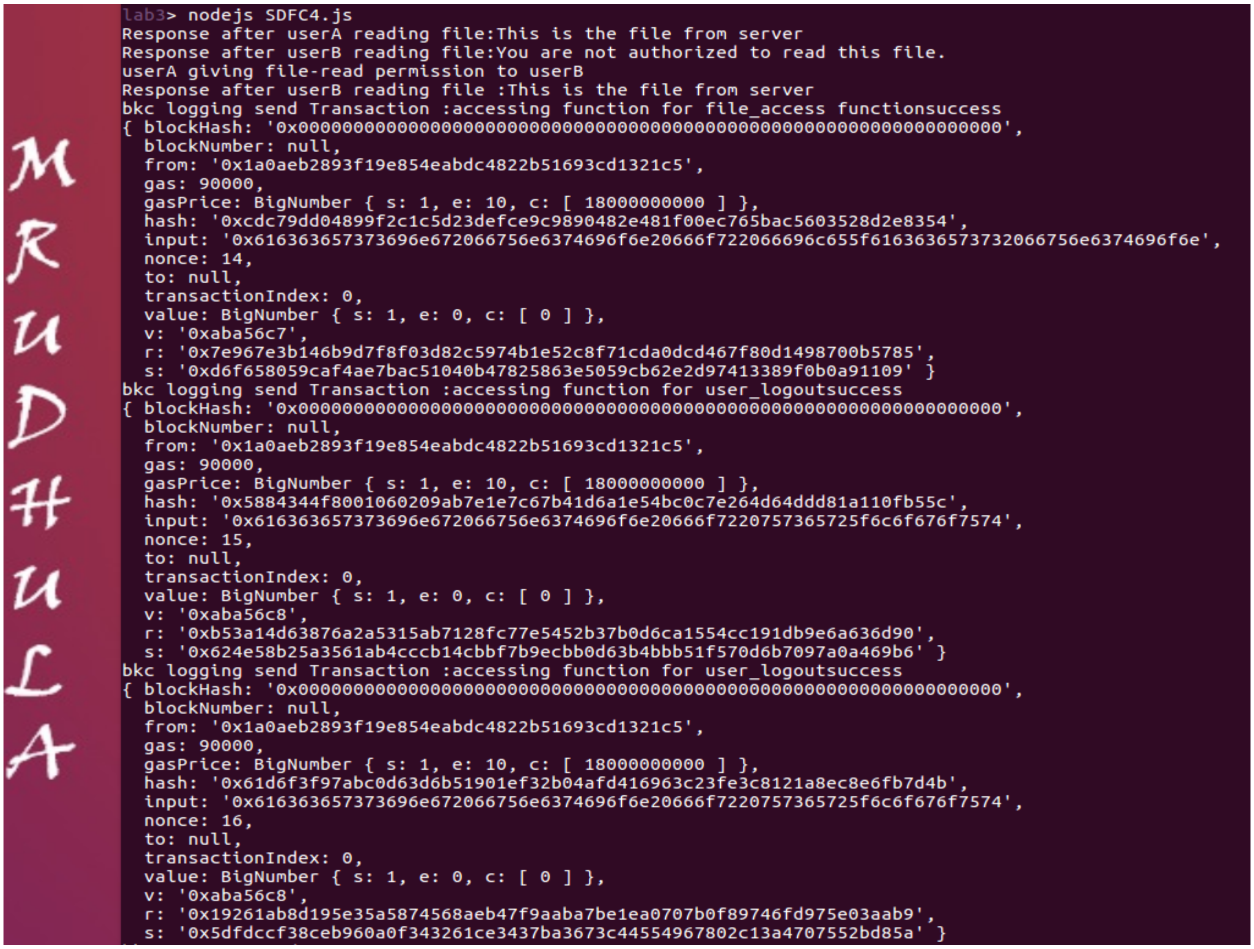


```
  this.file_delegate = function( delegator,  delegatee) {
    if(filePermissionBit[delegator] == 1)
    {
      console.log(delegator +" giving file-read permission to "+delegatee);
      filePermissionBit[delegatee] = 1;
    }
  }

  this.file_Access = function(user) {
    if(loginStatus[user] == 1 && filePermissionBit[user] == 1)
    {
      return fileX;
    }
    return "You are not authorized to read this file.";
  }
event = "accessing function for file_access function";
        bkc_logging(event);
}

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");
  }
}
```

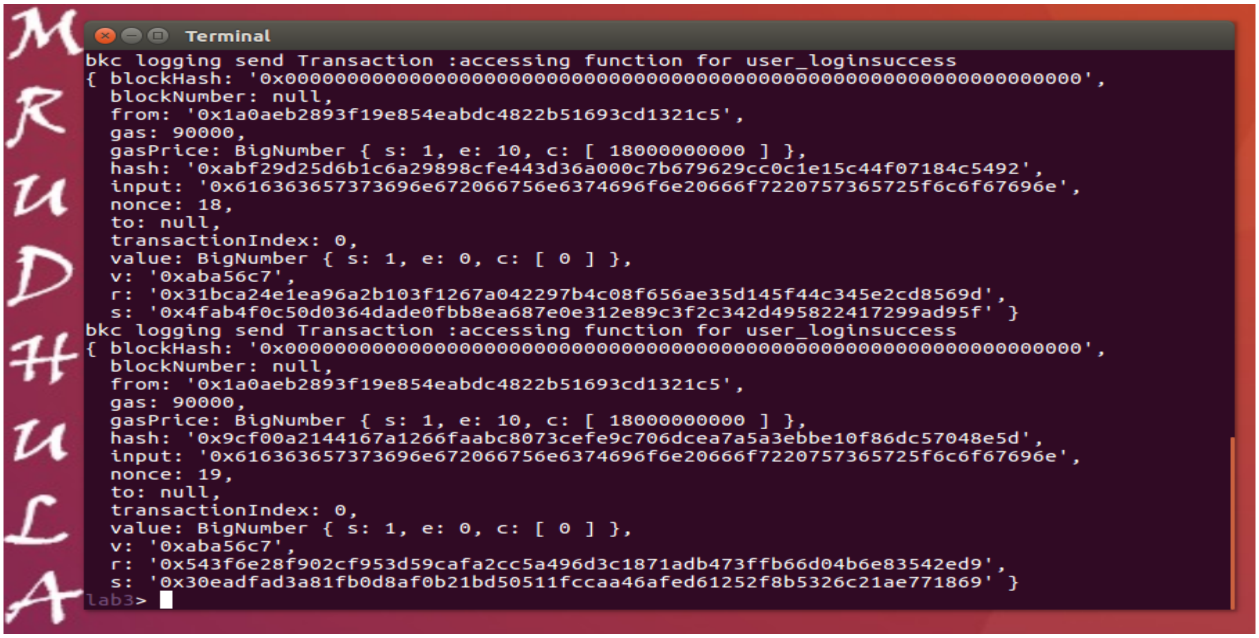


```
/Function to generate hex encoded value for input string & sending transaction to
blockchain for logging puropse/
function bkc_logging(event){
console.log("bkc_logging event :" + event);
}

/Function to get a account from local blockchain/
function setDefaultAccount(){
  web3.eth.getAccounts(function(error, result){
    if(!error){
      defaultAcc  = result[0];
      var client1 = new client();
      client1.execute();
    }}
  );
}
```

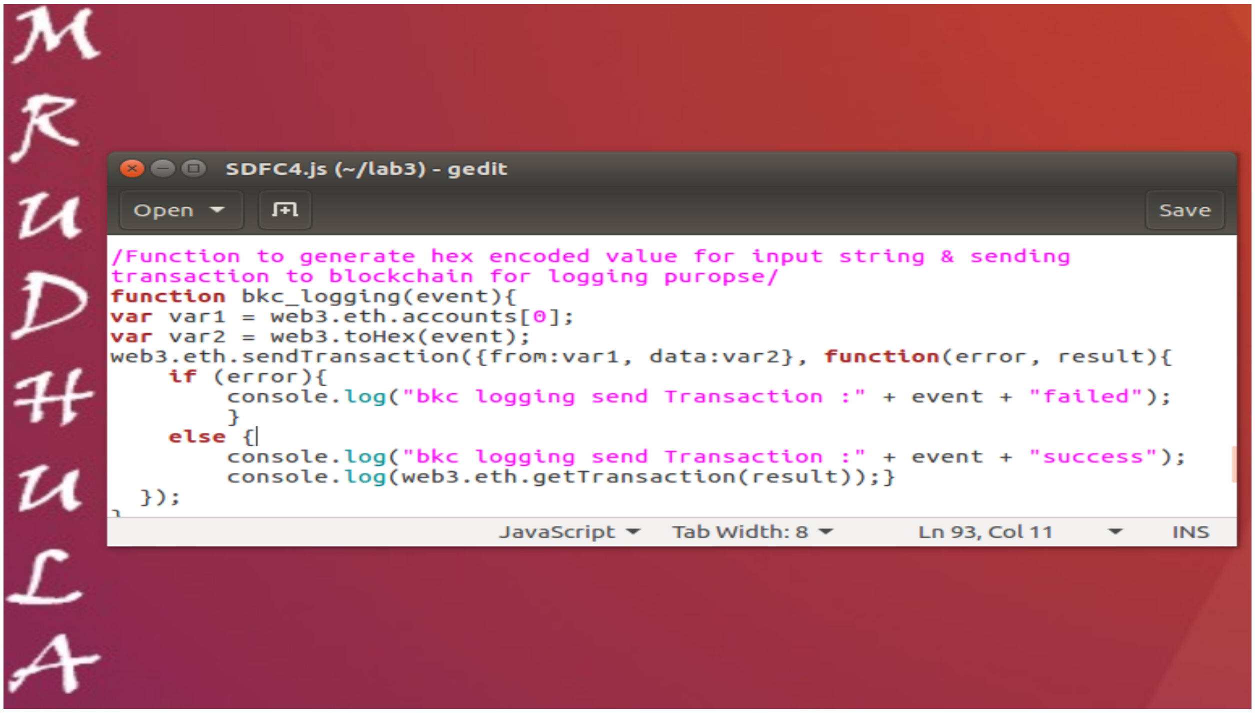
TASK 4: BLOCKCHAIN LOGGING



```
lab3> nodejs SDFC4.js
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
bkc logging send Transaction :accessing function for file_access functionsuccess
{ blockHash: '0x0000000000000000000000000000000000000000000000000000000000000000',
  blockNumber: null,
  from: '0x1a0aeb2893f19e854eabdc4822b51693cd1321c5',
  gas: 90000,
  gasPrice: BigNumber { s: 1, e: 10, c: [ 18000000000 ] },
  hash: '0xcdc79dd04899f2c1c5d23defce9c9890482e481f00ec765bac5603528d2e8354',
  input: '0x616363657373696e672066756e6374696f6e20666f722066696c655f616363657373732066756e6374696f6e',
  nonce: 14,
  to: null,
  transactionIndex: 0,
  value: BigNumber { s: 1, e: 0, c: [ 0 ] },
  v: '0xaba56c7',
  r: '0x7e967e3b146b9d7f8f03d82c5974b1e52c8f71cda0dcd467f80d1498700b5785',
  s: '0xd6f658059caf4ae7bac51040b47825863e5059cb62e2d97413389f0b0a91109' }
bkc logging send Transaction :accessing function for user_logoutsuccess
{ blockHash: '0x0000000000000000000000000000000000000000000000000000000000000000',
  blockNumber: null,
  from: '0x1a0aeb2893f19e854eabdc4822b51693cd1321c5',
  gas: 90000,
  gasPrice: BigNumber { s: 1, e: 10, c: [ 18000000000 ] },
  hash: '0x5884344f8001060209ab7e1e7c67b41d6a1e54bc0c7e264d64ddd81a110fb55c',
  input: '0x616363657373696e672066756e6374696f6e20666f7220757365725f6c6f676f7574',
  nonce: 15,
  to: null,
  transactionIndex: 0,
  value: BigNumber { s: 1, e: 0, c: [ 0 ] },
  v: '0xaba56c8',
  r: '0xb53a14d63876a2a5315ab7128fc77e5452b37b0d6ca1554cc191db9e6a636d90',
  s: '0x624e58b25a3561ab4cccb14cbbf7b9ecbb0d63b4bbb51f570d6b7097a0a469b6' }
bkc logging send Transaction :accessing function for user_logoutsuccess
{ blockHash: '0x0000000000000000000000000000000000000000000000000000000000000000',
  blockNumber: null,
  from: '0x1a0aeb2893f19e854eabdc4822b51693cd1321c5',
  gas: 90000,
  gasPrice: BigNumber { s: 1, e: 10, c: [ 18000000000 ] },
  hash: '0x61d6f3f97abc0d63d6b51901ef32b04afd416963c23fe3c8121a8ec8e6fb7d4b',
  input: '0x616363657373696e672066756e6374696f6e20666f7220757365725f6c6f676f7574',
  nonce: 16,
  to: null,
  transactionIndex: 0,
  value: BigNumber { s: 1, e: 0, c: [ 0 ] },
  v: '0xaba56c8',
  r: '0x19261ab8d195e35a5874568aeb47f9aaba7be1ea0707b0f89746fd975e03aab9',
  s: '0x5dfdccf38ceb960a0f343261ce3437ba3673c44554967802c13a4707552bd85a' }
```



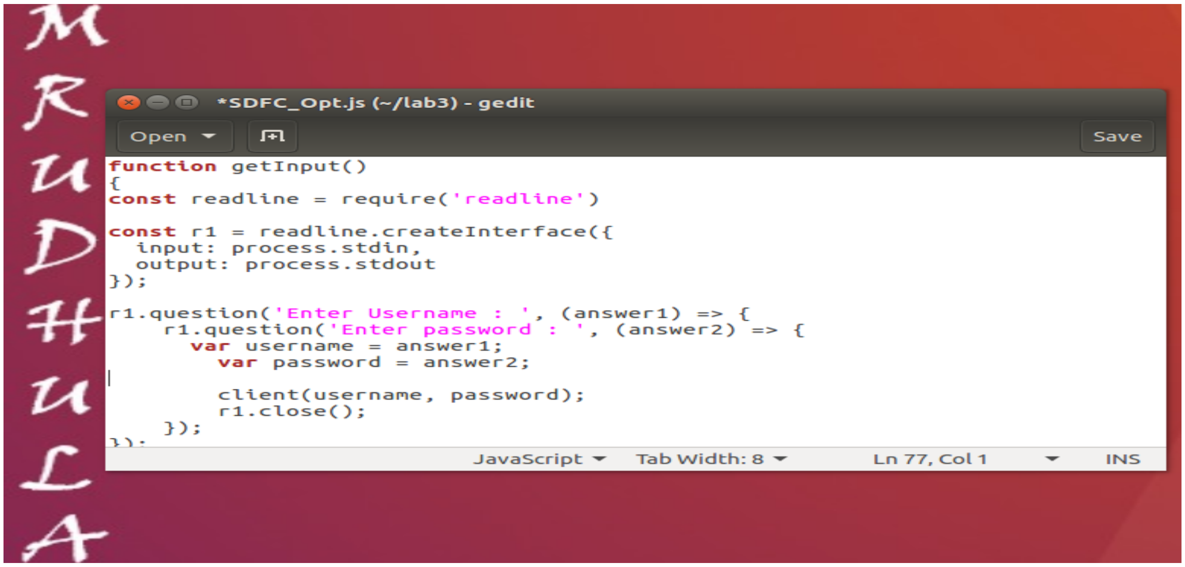
```
bkc logging send Transaction :accessing function for user_loginsuccess
{ blockHash: '0x0000000000000000000000000000000000000000000000000000000000000000',
  blockNumber: null,
  from: '0x1a0aeb2893f19e854eabdc4822b51693cd1321c5',
  gas: 90000,
  gasPrice: BigNumber { s: 1, e: 10, c: [ 18000000000 ] },
  hash: '0xabf29d25d6b1c6a29898cfe443d36a000c7b679629cc0c1e15c44f07184c5492',
  input: '0x616363657373696e672066756e6374696f6e20666f7220757365725f6c6f67696e',
  nonce: 18,
  to: null,
  transactionIndex: 0,
  value: BigNumber { s: 1, e: 0, c: [ 0 ] },
  v: '0xaba56c7',
  r: '0x31bca24e1ea96a2b103f1267a042297b4c08f656ae35d145f44c345e2cd8569d',
  s: '0x4fab4f0c50d0364dade0fbb8ea687e0e312e89c3f2c342d495822417299ad95f' }
bkc logging send Transaction :accessing function for user_loginsuccess
{ blockHash: '0x0000000000000000000000000000000000000000000000000000000000000000',
  blockNumber: null,
  from: '0x1a0aeb2893f19e854eabdc4822b51693cd1321c5',
  gas: 90000,
  gasPrice: BigNumber { s: 1, e: 10, c: [ 18000000000 ] },
  hash: '0x9cf00a2144167a1266faabc8073cefe9c706dcea7a5a3ebbe10f86dc57048e5d',
  input: '0x616363657373696e672066756e6374696f6e20666f7220757365725f6c6f67696e',
  nonce: 19,
  to: null,
  transactionIndex: 0,
  value: BigNumber { s: 1, e: 0, c: [ 0 ] },
  v: '0xaba56c7',
  r: '0x543f6e28f902cf953d59cafa2cc5a496d3c1871adb473ffb66d04b6e83542ed9',
  s: '0x30eadfad3a81fb0d8af0b21bd50511fccaa46afed61252f8b5326c21ae771869' }
lab3> 
```



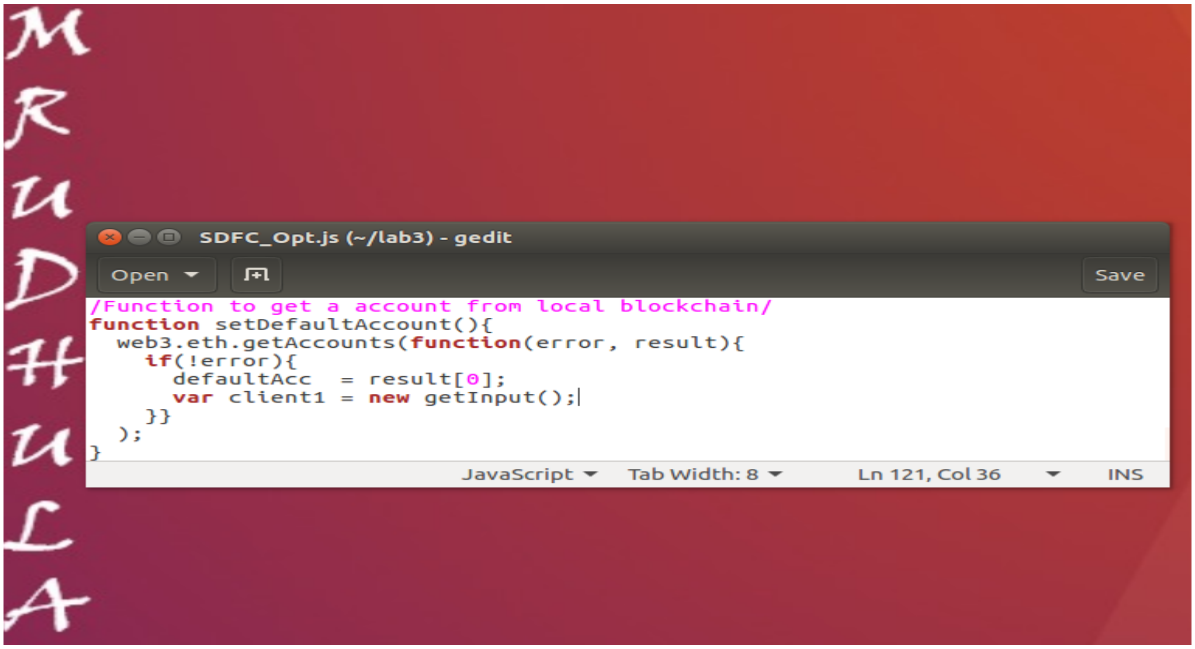
```
/Function to generate hex encoded value for input string & sending
transaction to blockchain for logging puropse/
function bkc_logging(event){
var var1 = web3.eth.accounts[0];
var var2 = web3.toHex(event);
web3.eth.sendTransaction({from:var1, data:var2}, function(error, result){
        if (error){
                console.log("bkc logging send Transaction :" + event + "failed");
                }
        else {
                console.log("bkc logging send Transaction :" + event + "success");
                console.log(web3.eth.getTransaction(result));}
  });
```

The above screenshot we notice the event was logged and transaction was sent. The logging on Ethereum Blockchain is done using web3.eth.sendTransaction() for sending data from the account. Web3.eth.getTransaction() prints the transaction block.

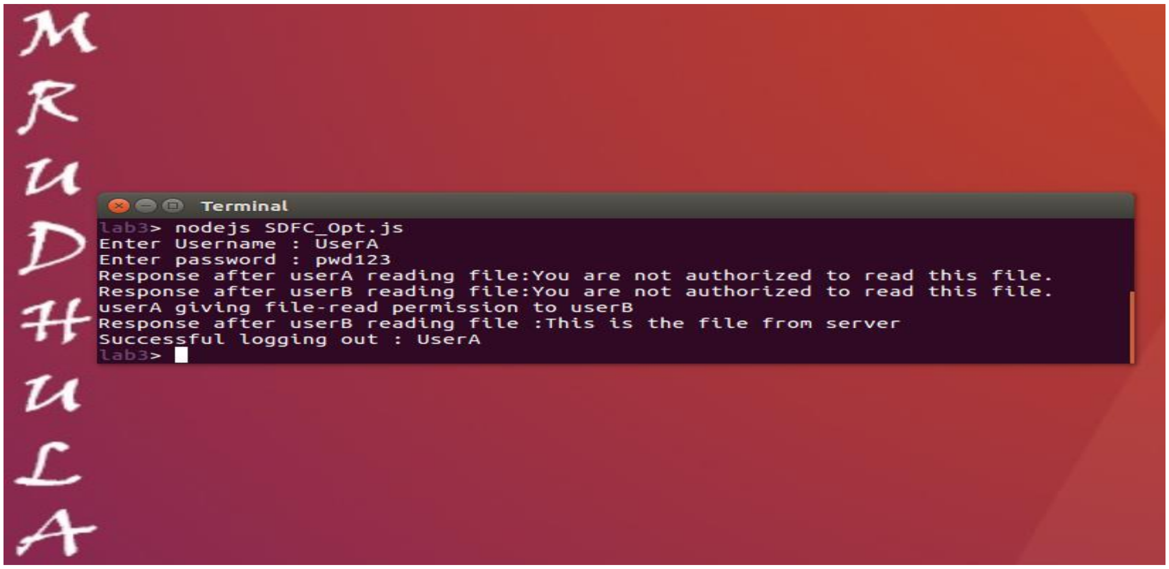
TASK 4: DESIGN OPTIMIZATION



```
function getInput()
{
const readline = require('readline')

const r1 = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

r1.question('Enter Username : ', (answer1) => {
    r1.question('Enter password : ', (answer2) => {
      var username = answer1;
        var password = answer2;

        client(username, password);
        r1.close();
    });
});
```



```
/Function to get a account from local blockchain/
function setDefaultAccount(){
  web3.eth.getAccounts(function(error, result){
    if(!error){
      defaultAcc  = result[0];
      var client1 = new getInput();
    }}
  );
}
```



```
lab3> nodejs SDFC_Opt.js
Enter Username : UserA
Enter password : pwd123
Response after userA reading file:You are not authorized to read this file.
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
Successful logging out : UserA
lab3>
```

The above code takes input from console from User for user_login() i.e. username and password by using readline() function and authenticates particular user for login and sets their file access bit to 1 and also the user gives access of file to other user. It also logs out the user who logged in.