JavaScript crud operation =>
https://www.youtube.com/watch?v=k4HRR85Af-I

MERN Stack =>
https://www.youtube.com/watch?v=nu0z_RVs7r8

MongoDB Module in Node js
https://www.youtube.com/watch?v=fbYExfeFsI0 ⇒ Connecting & CRUD Operations (Part 1 of 4)

Topic	**Reference Links**
Git	https://www.w3schools.com/git/default.asp
Linux	https://www.hostinger.in/tutorials/linux-commands
HTML	https://www.w3schools.com/html/default.asp
CSS	https://www.w3schools.com/css/default.asp
Boostrap	https://www.w3schools.com/bootstrap5/index.php
Javascript	https://www.youtube.com/watch?v=PkZNo7MFNFg&t=30s https://www.w3schools.com/js/default.asp
Typescript	https://youtu.be/gp5H0Vw39yw
ES6/ES7/ES8	- https://www.youtube.com/watch?v=NCwa_xi0Uuc&t=18 - https://www.youtube.com/watch?v=nZ1DMMsyVyI&t=9s
ReactJS	- https://www.w3schools.com/react/default.asp - https://youtu.be/QFaFIcGhPoM
Redux	https://redux.js.org/introduction/getting-started https://www.youtube.com/watch?v=poQXNp9ItL4&t=3970s
Material UI	https://mui.com/getting-started/installation/
React Styled Components	https://youtu.be/FSCSdAlLsYM
NextJS	- https://nextjs.org/docs/getting-started - https://www.youtube.com/watch?v=mTz0GXj8NN0 - https://www.youtube.com/watch?v=MFuwkrseXVE
React Boostrap	- https://react-bootstrap.github.io/getting-started/introduction
NodeJS	https://youtu.be/TIB_eWDSMt4 https://youtu.be/RLtyhwFtXQA
Sequelize ORM	https://youtu.be/ExTZYpyAn6s

Topic	**Reference Links**
Git	https://www.w3schools.com/git/default.asp
Linux	https://www.hostinger.in/tutorials/linux-commands
NestJS	https://docs.nestjs.com/ https://www.youtube.com/watch?v=wqhNoDE6pb4&t=2496s https://www.youtube.com/watch?v=F_oOtaxb0L8&t=1522s
GoLang	https://go.dev/tour/welcome/1 https://gobyexample.com/
Go SQL	https://pkg.go.dev/database/sql https://www.youtube.com/watch?v=Y0lDGjwRYKw&list=PL4cUxeGkcC9iK6Qhn-QLcXCXPQUov1U7f
Blockchain	
Ethereum Blockchain	
Ethereum Smart Contracts	
Smart Contract Interaction	https://youtu.be/t3wM5903ty0
Solidity	

Node express MongoDB (CRUD-Operaration) => **https://www.youtube.com/watch?v=W1Kttu53qTg**

Express JS Code with Harry => https://www.youtube.com/watch?v=7H_QH9nipNs

Unit testing => https://www.youtube.com/watch?v=u5cLK1UrFyQ
https://www.youtube.com/watch?v=MLTRHc5dk6s

```
1.
//command to list directories and files
ls
//command to see all dirs and files with hidden also
ls -a

ls -R //will list all down all the
2.
//to find path of current directories
pwd

3.
//command to change the directories

cd Desktop //will bring you from there to desktop

cd Desktop/Practice //skip multiple directories

cd .. //to go to 1 level up towards root directories

cd - //to go to the previous directories

4.
cat fileName.txt //to show the contents of the file

cat > fileName //creates a file

cat fileName1 fileName2 > fileName3 //file1 and file2 will be added and goes into a new file file3

cat filename | tr a-z A-Z >output.txt //converts all chars into uppercase and put into a new file output.txt


cp scenery.jpg /home/username/Pictures //creates a copy
```

5. To kill a process running on port 9090
npx kill-port 9090
6. How to edit a file from terminal
Vim filename.extention +line_no.

7. To exit from vim type command

a. **Save and exit**
   → press esc and then type ":wq"
b. **Exit without save**
   →type ":qa!"

8.

—---------------------------------MongoDB Cheat Sheet -------------------------------------

Open terminal and type 'mongo' to go into the mongodb environment

1. **To see the current database**
   → db
2. **To see all the database**
   → show dbs

3. To go to a particular database
   →use dbName
4. To create a database
   →use dbName
   //a database will be created
5. To see collections of a db
   → show collections
6. To find all the documents in a database
   →db.collectionName.find().pretty()
7. To find the particular document in db
   → db.collectionName.find({“fieldName”:”fieldValue”})
8. .pretty() will show the documents in a user friendly way
9. To delete a document
   →

-- How to server in terminal and how to start, stop, and debug in server

—-----------------------------------Start from here —-------------------------------

// to find ssh key in terminal

```
cat ~/.ssh/id_rsa
```

--> output like below

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDD7o7Dp+GEm/guJ85SWGGP5esW3pYcnYm
HfowOm2Qq2jJXyG3HYfhf4+unTRSWCkjHkRn0f5Z/UbQKjIpFIkpqpQUKwNA8z3hAiY1r6n
JjxWRGj/N0I6V5SZH4dfNDR0GkpjBKKHYW6wJrKwoP740UyWeaBmkV79rYRuGNcjXwzW
h/tqHEEMsCTsSIg+5NOzEZICyKZd6WGIpcvRK+kf+44KuVbZypKmVbc5vhIaq9FPetrWUH
XSn3QjB401E8pH5h82jVoZ9W65hmw6t9qzSV/P7673+embtTfgCASLpoaThmk6xDNHDbMs
BDIaZPGHzqsCnqa8fks+aBGmCvn48RNECxIm+2YkNwuI5mTwaFmK8VQwQDs0/+srqoBY
hfoW8gss2FiwL6TIpOf0tjajo0+F13zfRGA7gRcqve4aSIPnhNWgdakA5InM79bk4Agu1b8A3
PTPnhj7HnM9EmjCrEyY0gA8G/gMCVNdww8DxwY6eNA/oz7QFZcL2LQI5yMhnddmg2Rzm
HvN7bRDpBYsIim/VCU/e86Y+0TSCjsF9eZH+L0Hw1STCpbVWHzRTVftxZp/p5kgAIqI2wsy
UkkLrBIt+ZNXxP+tniEOX12526ZO8LOpxBbGN5kARpcXpKoFTQCwkEyLyOd19RUf30dSkJ
8mgjaTBASQL+HVX7Y70c5w== arvind.s@knackroot.com
```

// to connect to server type below command which will be granted by admin
ssh sandip@34.93.107.95

// command to see the log of server from local
type "fish" and then type "pm2 status" will show output like below

```
┌────┬───────────────────────────────────────┬──────────────────┬──────────
─┬────────────┬─────────────┬───────────┬────────┬───────────────┬──────────
─┬─────────────┬─────────────┬─────────────┐
│ id │ name                 │ namespace   │ version │ mode    │ pid     │ uptime │ ↺    │
status    │ cpu      │ mem      │ user     │ watching │
├────┼───────────────────────────────────────┼──────────────────┼──────────
─┼────────────┼─────────────┼───────────┼────────┼───────────────┼──────────
─┼─────────────┼─────────────┼─────────────┤
│ 0  │ DecentraName-Backend-prod    │ default    │ 0.39.1 │ fork    │ 23888    │ 3h    │
146 │ online    │ 0%      │ 50.2mb │ sandip  │ disabled │
└────┴───────────────────────────────────────┴──────────────────┴──────────
─┴────────────┴─────────────┴───────────┴────────┴───────────────┴──────────
─┴─────────────┴─────────────┴─────────────┘
```

then type "pm2 logs --raw 0 || pino-pretty -t" will show some logs and then send somerequest to the server and then you can see logs of that request in your terminal

—---------------------log will be something like this—----------------------------

----------------log of server------------------------

```
[2022-04-18 09:08:08.328 +0000] INFO (23923 on dweb-testnet):
approve-subdomain-registration request body: {
  "name": [
    "qw2"
  ],
  "label": [
    "arvind1"
  ],
  "owner": [
    "0xb66Fb7DDe8317c9758f611237A41AbB4aBD596DA"
  ],
  "sender": "0xb66Fb7DDe8317c9758f611237A41AbB4aBD596DA",
  "chainid": 4,
  "signature":
"0xa9e1acc25b8b6ac1d718ea883b730bd987e64328c65f2c58a10252a7a1c2b7ba48c6d0823
6db849f0106dd22e8dcbff6c477ea13772492db88601ef545a7c3c21c"
}
[2022-04-18 09:08:09.982 +0000] DEBUG (23923 on dweb-testnet): Generated subdomain
registration commitment is
0x274cc6f1a3fea38d8a247326fe1ef2c39d67e699e87d16b0068bd1659b778b87
[2022-04-18 09:08:09.986 +0000] DEBUG (23923 on dweb-testnet): request completed
    req: {
      "id": 583,
      "method": "POST",
      "url": "/api/v1/approve-subdomain-registration",
      "query": {},
      "params": {},
      "headers": {
        "host": "api-hotfix.decentraweb.org",
        "x-real-ip": "172.70.218.69",
        "x-forwarded-for": "27.109.10.98, 172.70.218.69",
        "connection": "close",
        "content-length": "304",
        "accept-encoding": "gzip",
        "cf-ipcountry": "IN",
        "cf-ray": "6fdc410fefdb8489-BOM",
        "x-forwarded-proto": "https",
        "cf-visitor": "{\"scheme\":\"https\"}",
        "sec-ch-ua": "\" Not A;Brand\";v=\"99\", \"Chromium\";v=\"100\", \"Google
Chrome\";v=\"100\"",
        "accept": "application/json, text/plain, */*",
        "content-type": "application/json",
        "sec-ch-ua-mobile": "?0",
        "user-agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/100.0.4896.75 Safari/537.36",
        "sec-ch-ua-platform": "\"Linux\"",
        "origin": "https://dns-hotfix.decentraweb.org",
        "sec-fetch-site": "same-site",
        "sec-fetch-mode": "cors",
        "sec-fetch-dest": "empty",
        "referer": "https://dns-hotfix.decentraweb.org/",
        "accept-language": "en-GB,en-US;q=0.9,en;q=0.8",
        "cf-connecting-ip": "27.109.10.98",
        "cdn-loop": "cloudflare"
      },
      "remoteAddress": "::ffff:127.0.0.1",
      "remotePort": 40806,
      "body": {
        "name": [
          "qw2"
        ],
        "label": [
          "arvind1"
        ],
        "owner": [
          "0xb66Fb7DDe8317c9758f611237A41AbB4aBD596DA"
        ],
        "sender": "0xb66Fb7DDe8317c9758f611237A41AbB4aBD596DA",
        "chainid": 4,
        "signature":
"0xa9e1acc25b8b6ac1d718ea883b730bd987e64328c65f2c58a10252a7a1c2b7ba48c6d0823
6db849f0106dd22e8dcbff6c477ea13772492db88601ef545a7c3c21c"
      }
    }
    res: {
      "statusCode": 200,
      "headers": {
        "x-powered-by": "Express",
        "access-control-allow-origin": "*",
        "content-type": "application/json; charset=utf-8",
        "content-length": "250",
        "etag": "W/\"fa-nRBIMQDzQ29g1NIlCxyOaMK98Bg\""
      }
    }
    responseTime: 1658


    ----------------------------------------------log end----------------------------------------------------------
```

you can also use here command like ls, cd ...etc

you can here go in other directory from one directory

vim server.js +882 ==>> will take you directly on line no. 882 in server.js file

//how to come out of file after editing it
first press esc and then write ":wq" //file will be saved and you will come out of it

:qa! //will revert all changes and come out of file

```
 //command to restart the server is
 pm restart 0
```

-------------------------- to open server databases ---

```
// to open the database
mongo 'url_of_dataabse'
```

and all commands are same as on command line

////////////////////////////////// failed metadata and some logs ////////////////////////////////

http://api-hotfix.decentraweb.org/api/v1/metadata/0xd860Cda0E428C7C7A905739B81Ca81A3e2e359A5/11178919336686568857287642370583339281059634224727941742481381024852090048579

/api/v1/metadata/0xd860Cda0E428C7C7A905739B81Ca81A3e2e359A5/1141588560161583952051339263063757306877962721074147333520476116451224695077 91

/api/v1/metadata/0xd860Cda0E428C7C7A905739B81Ca81A3e2e359A5/9034308898612262306433862315208010449105449577513748088543160482570480027572 4

/api/v1/metadata/0xd860Cda0E428C7C7A905739B81Ca81A3e2e359A5/3613218069913091958546107991700187543775549418246328297204463728457923477309 6

/api/v1/metadata/0xd860Cda0E428C7C7A905739B81Ca81A3e2e359A5/6319605719346058985554585144985544923186354753094877341187084614455824950769 5

/api/v1/metadata/0xd860Cda0E428C7C7A905739B81Ca81A3e2e359A5/4795073776890453389465015958784678663965520302870520592574996224698967877753 7

```
req: {
  "id": 366,
  "method": "GET",
  "url": "/api/v1/image/6eeb23a7-bc50-4d4b-a795-a4b40fbe47b6",
  "query": {},
  "params": {},
  "headers": {
    "host": "api-hotfix.decentraweb.org",
    "x-real-ip": "172.68.118.49",
    "x-forwarded-for": "35.77.187.86, 172.68.118.49",
    "connection": "close",
    "accept-encoding": "gzip",
    "cf-ipcountry": "JP",
    "cf-ray": "6fe5911ebc14341a-NRT",
    "x-forwarded-proto": "https",
    "cf-visitor": "{\"scheme\":\"https\"}",
    "user-agent": "okhttp/4.9.3",
    "cf-connecting-ip": "35.77.187.86",
    "cdn-loop": "cloudflare"
    },
    "remoteAddress": "::ffff:127.0.0.1",
    "remotePort": 49854
  }
  res: {
    "statusCode": 200,
    "headers": {
      "x-powered-by": "Express",
      "access-control-allow-origin": "*",
      "content-type": "image/png",
      "content-length": 21558
    }
  }
  responseTime: 135
Failed to generate metadata image Error: Could not find MIME for Buffer <null>
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/utils/image-bitmap
.js:187:15)
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:431:32)
    at
/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:373:15
    at FSReqCallback.readFileAfterClose [as oncomplete]
(internal/fs/read_file_context.js:71:3) {
  methodName: 'constructor'
}
Failed to generate metadata image Error: Could not find MIME for Buffer <null>
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/utils/image-bitmap
.js:187:15)
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:431:32)
    at
/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:373:15
    at FSReqCallback.readFileAfterClose [as oncomplete]
(internal/fs/read_file_context.js:71:3) {
  methodName: 'constructor'
}
Failed to generate metadata image Error: Could not find MIME for Buffer <null>
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/utils/image-bitmap
.js:187:15)
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:431:32)
```

```
    at
/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:373:15
    at FSReqCallback.readFileAfterClose [as oncomplete]
(internal/fs/read_file_context.js:71:3) {
  methodName: 'constructor'
}
Failed to generate metadata image Error: Could not find MIME for Buffer <null>
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/utils/image-bitmap
.js:187:15)
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:431:32)
    at
/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:373:15
    at FSReqCallback.readFileAfterClose [as oncomplete]
(internal/fs/read_file_context.js:71:3) {
  methodName: 'constructor'
}
Failed to generate metadata image Error: Could not find MIME for Buffer <null>
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/utils/image-bitmap
.js:187:15)
    at Jimp.parseBitmap
(/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:431:32)
    at
/home/sandip/decentraname-backend/node_modules/@jimp/core/dist/index.js:373:15
    at FSReqCallback.readFileAfterClose [as oncomplete]
(internal/fs/read_file_context.js:71:3) {
  methodName: 'constructor'
}
[2022-04-19 12:15:42.504 +0000] ERROR (17310 on dweb-testnet): Failed to add image:;
{"imageid":"6eeb23a7-bc50-4d4b-a795-a4b40fbe47b6"}
[2022-04-19 12:15:42.507 +0000] DEBUG (17310 on dweb-testnet): Error encountered in
getting asset metadata: {"type":"Error","message":"Failed to add image","stack":"Error:
Failed to add image\n    at Object.addImage
(/home/sandip/decentraname-backend/database/assets.js:61:15)\n    at runMicrotasks
(<anonymous>)\n    at processTicksAndRejections
(internal/process/task_queues.js:95:5)\n    at async
/home/sandip/decentraname-backend/server.js:255:13"}
[2022-04-19 12:15:42.515 +0000] ERROR (17310 on dweb-testnet): Error encountered in
getting asset metadata: 'Failed to add image'
[2022-04-19 12:15:42.519 +0000] DEBUG (17310 on dweb-testnet): request errored
```

=======------------------------------------ｰ Schema of DB —------------------------------------=========

```
db.createCollection( "metadata", {
  validator: {
      $jsonSchema: {
       required: [ "tokenid", "imageid" ],
       properties: {
          tokenid: {
             bsonType: "string",
             description: "must be a string and is required"
          },
          imageid: {
            bsonType: "string",
            description: "must be a string and is required"
          }
       }
    }
  }
}
);


//

db.metadata.createIndex( { "tokenid": 1 }, { unique: true } );

db.createCollection( "image", {
  validator: {
      $jsonSchema: {
       required: [ "imageid", "image" ],
       properties: {
          imageid: {
             bsonType: "string",
             description: "must be a string and is required"
          },
          image: {
             bsonType: "binData",
             description: "must be a string and is required"
          }
       }
    }
  }
}
);

db.image.createIndex( { "imageid": 1 }, { unique: true } );

db.createCollection( "hash", {
  validator: {
      $jsonSchema: {
       required: [ "hash", "name" ],
       properties: {
          hash: {
             bsonType: "string",
             description: "must be a string and is required"
          },
          name: {
            bsonType: "string",
            description: "must be a string and is required"
          }
       }
    }
  }
}
);


db.createCollection( "nameIdMap", {
  validator: {
      $jsonSchema: {
       required: [ "id", "name" ],
       properties: {
          hash: {
             bsonType: "string",
             description: "must be a string and is required"
          },
          name: {
            bsonType: "string",
            description: "must be a string and is required"
          }
       }
    }
  }
}
);

db.nameIdMap.createIndex( { "id": 1 }, { unique: true } );




// Schema for the address and email form

db.createCollection( "profile", {
    validator: {
        $jsonSchema: {
            required: [ "address"],
            properties: {
                address: {
                    bsonType: "string",
                    description: "etheriumAddress should not be duplicate"
                },
                email: {
                    bsonType: "string",
                    description: "email id associated with the etheriumaddress but it is not required"
                }
            }
        }
    }
})

db.profile.createIndex({ "address": 1}, { unique: true})
```