[illegible]

[illegible] unknown address where the [illegible]

[illegible] [illegible] after this [illegible]

[illegible] the [illegible] account

[illegible] [illegible] These like having

[illegible]

Node js

Node js is a [illegible]

[illegible]

[illegible]

[illegible]

Module

It is a package [illegible]

node filename

3rd party modules

[illegible]

[illegible]

Core module Custom module

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] where the bank [illegible]

[illegible] quite than phone so done

[illegible]

[illegible] these like having

Node.js

Node.js is a library
One more library is Express.js
It will connect frontend with database.
It also like middleware

Module

It is a package like (libraries)

3rd party modules:
numpy
pandas

node filename

Math module custom module:

math.sqrt

To run node files => node filename.

In node we can use express.js as middleware

uestion:
OD:
filter only yes
Display only the customers address where the branch name
e same.
Current balance 10,000 to 20,000 filter their phone.no alone
filter only the savings account
add the column name status: same those who having
me IFSC code ~~same~~

Node.js

Node.js is a library
One more library is Express.js
It will connect frontend with data base.
It also like middleware.

odule

It is a package like (libraries)

node filename

rd party modules
numpy
Pandas

ath module custom module:

math.start

To run node files -> node filename

[illegible]

Display only the customers address where the branch name

[illegible] balance 10,000 to 20,000 filter their phone.no alone

[illegible] only the savings account

[illegible] the column name status: name those who having

[illegible] IFSC code [illegible]

Node.js

Node.js is a library
One more library is Express.js
It will connect frontend with database.
It also acts middleware

Module
It is a package (i.e libraries)

node filename

[illegible] party modules:
numpy
pandas
Math module custom module:

math.sqrt

To [illegible] node files => node filename

[illegible] we can use express.js as middleware ation:

filter only OD: yes

Display only the customers address where the branch name same.

Current balance 10,000 to 20,000 filter their phone no alone

ilter only the savings account

dd the column name status: ~~name~~ those who having

ne IFSC code ~~[illegible]~~

Node.js

Node.js is a library

One more library is Express.js

It will connect frontend with database

It also like middleware

odule

It is a package ela (libraries)

node filename

rd party modules

numpy

pandas

[illegible] custom module:

Question:

i) filter only OD: yes
(ii) Display only the customers address where the branch names are same.
(iii) Current balance 10,000 to 20,000 filter their phone.no alone
(iv) filter only the savings account
(v) add the column name status: same those Who having same IFSC code ~~[illegible]~~

Node.js

Node.js is a library
One more library is Express.js
It will connect frontend with database.
It also like middleware.

Module

It is a package like (libraries)

3rd party modules:

numpy
pandas

node filename

Math module **custom module:**

math.sqrt

To run node files => node filename

In node we can use express.js as middleware

Node:

module

require from node js - helps to include modules in our code
below example custom module

```
Const http = require('http'); // Built-in module
const express = require('express'); // Third party module
const sayHello = require('./greet'); // custom module
```

Node
client -> request
server -> response

To run server (node) => node ~~Appname~~ file

Note:
Always start a server (first)

Nodemon -> It is used to simply the work of refreshing the ~~to~~ browser.

Node.js:
Runtime environment that allows you to run javascript on the server-side

Note:
Maintain split terminals in VS code in order to use client and server.

Run command:
Always starts the server first

```
node filename
(server.js)
```

Command to run client/ (frontend)

```
npm start
```

To install express:

```
npm i express
```

to getting all packages from
we use require() it is from node.js
↳ keyword

sending request → post()
getting request → get()

Server always respond in the JSON format

To activate port ⇒ listen(port, ()=>{})

When we delete package.json.
To reinstall the package use command
⇓

```
npm init -y
```

Run the server:

Server console is below the our command not in browser inspect→console

Axioms:

It's a popular js library used to make http request from the browser/server

Axioms is known for easy and clean syntax also flexibility exspecially works well with API s and Rest API

Rest API

When we ~~right~~ write an API for exclusive purpose we call it an Rest API.

CORS:

(Cross Origin Resource Sharing)

When a web page request information from resource (from any other sites) whether to accept the request and process it or not. will be defined in a Rule book.

To install axiom

```
npm i axioms
```

To install CORS:

```
npm i CORS
```

In node we don't need to use import ~~be~~ instead we use require

To install nodemon

```
npm i -g nodemon
```

In the given example we are requesting data from server

In datacomponent.js as client using 'http' get method via ~~API /~~ api/data
Server responds as json from the json i want to filter only the msg, so we are using cors from cors & responds.data.message

---

Create mongo-crud App
inside src
Users.js
Create_User.js
Update_Users.js

Model:
Name
Age
Email
[button] Add user
} Store them in MongoDB

Dependencies:

install => Router
axios
Express, cors

App.js
Routing in App.js for three page

Create server folder
inside folder path. npm init

Write in ~~index.js~~ backend code in index.js

Create new folder inside server
named as model
Create Users.js There write server code
In terminal run npm i mongoose

```
app.post("CreateUser", (req, res) => {
    UserModel.create(req.body) → This is in cors
```

database

Create empl:
- emp_id
- emp_name
- emp_salary
- contact : array

create empl component

classmate
Date
Page

MySQL => GroupBy
MongoDB => aggregation

localField => like primarykey
foreignField => like foreignkey
lookup => It's look for particular task
from => another collection

```
db.companies.aggregate([{$lookUp:{from:'employees',
    localField:'_id',foreignField:'company_id',as:'employees'}}])
```

Here, local field belongs to employees
foreignfield belongs to companies
new matched document will be added as an array
in companies and the array name is employees.

Create a database called bank, create two collection under
That named as customer ~~and~~ personal and customer account

Datamodel: (1)
name : string
address : array
PhoneNo : Object (2 phonenum) num1, num2
Age :

Datamodel
Account number :
Branch ; string
Bank Name ; string
IFSC code : string
current Balance : float
Account type : saving and current
OD : Yes or No