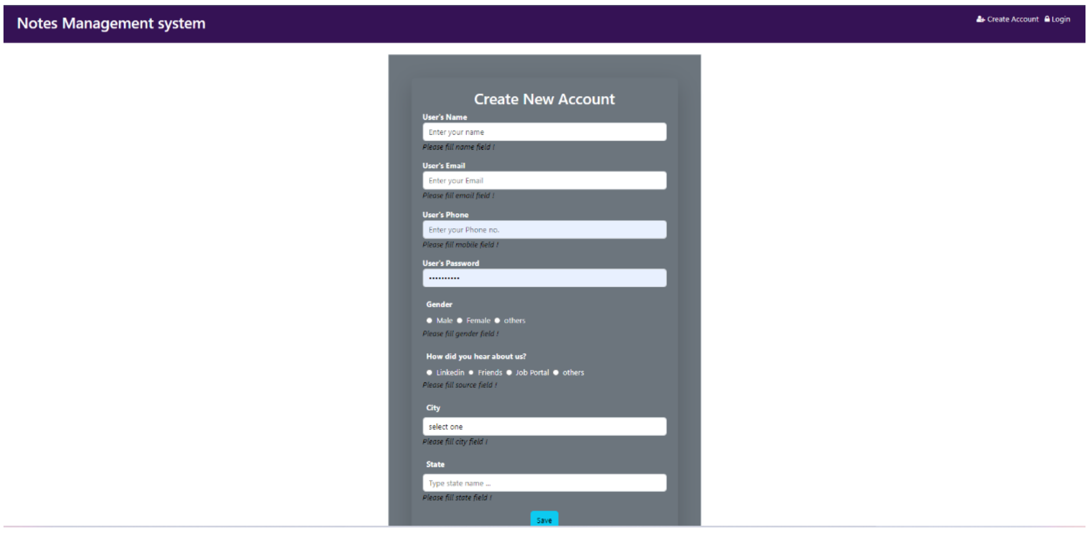
Notes Management system Design Document

Introduction- Welcome to our Note taking Portal (Notes Management system). Enjoy a seamless sign up and login experience. Once logged in, explore all the notes effortlessly. You can add notes by clicking on "Add Note" button. This application offers a user-friendly interface for efficient navigation. After adding and viewing notes securely log out using the designated button.

UI design-

1. **Create account Page-** I provided a form with some fields like name, email, mobile, gender, password, source, city & state. All fields are necessary to fill. For it I used validation in the form. After successfully account creation user can go to login page and can log in.



Create New Account

User's Name
Enter your name
Please fill name field !

User's Email
Enter your Email
Please fill email field !

User's Phone
Enter your Phone no.
Please fill mobile field !

User's Password
••••••••••

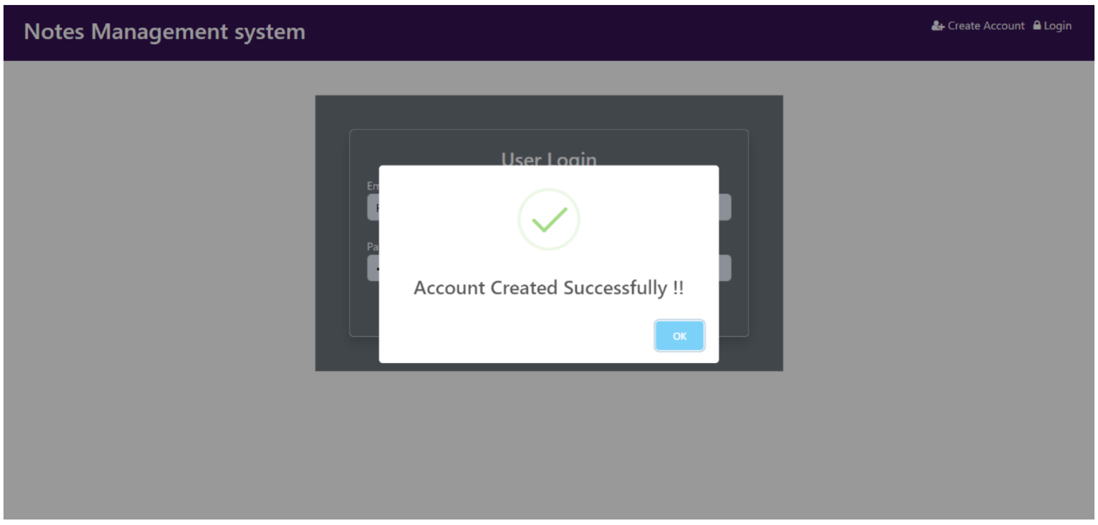
Gender
● Male ● Female ● others
Please fill gender field !

How did you hear about us?
● Linkedin ● Friends ● Job Portal ● others
Please fill source field !

City
select one
Please fill city field !

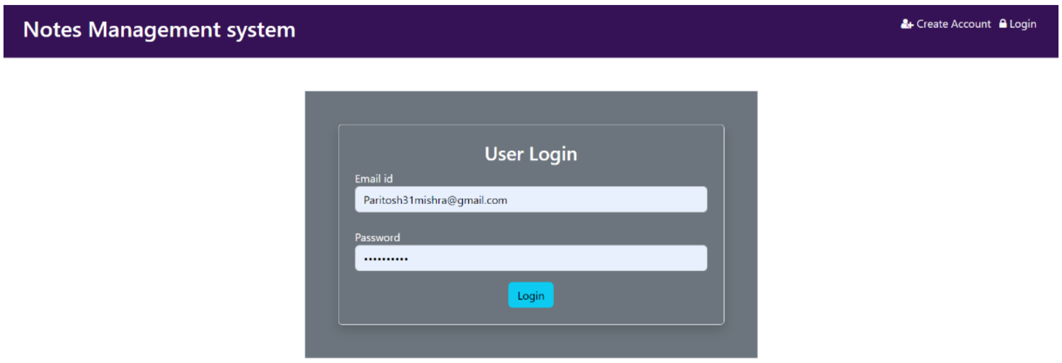
State
Type state name ...
Please fill state field !

Save



Account Created Successfully !!

2. **Login Page- -** I provided a form with two fields like email & password. All fields are necessary to fill. For it I used validation in the form. User can login by filling both fields. Once logged in, you will be directed to home page where you can explore all the notes and perform crud (create, retrieve, update & delete) operations.



Email id

Paritosh31mishra@gmail.com

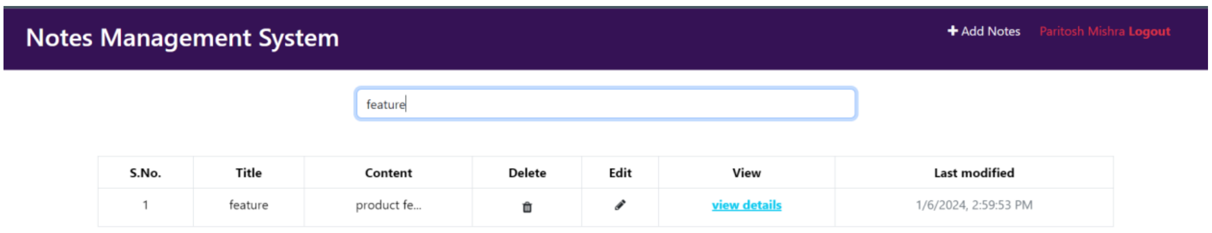
Password

••••••••••

3. **Notes Page (home page) -** Once you successfully logged in you will directed to notes page. On notes page You can explore all the notes effortlessly. All the available notes are fetched and shown in form of table with fields title, content, edit, delete. Each row represent a note. By clicking on edit button user can edit the note and by clicking on delete button user can delete a note. And by clicking on the link view details user can see the note in details.

Features:

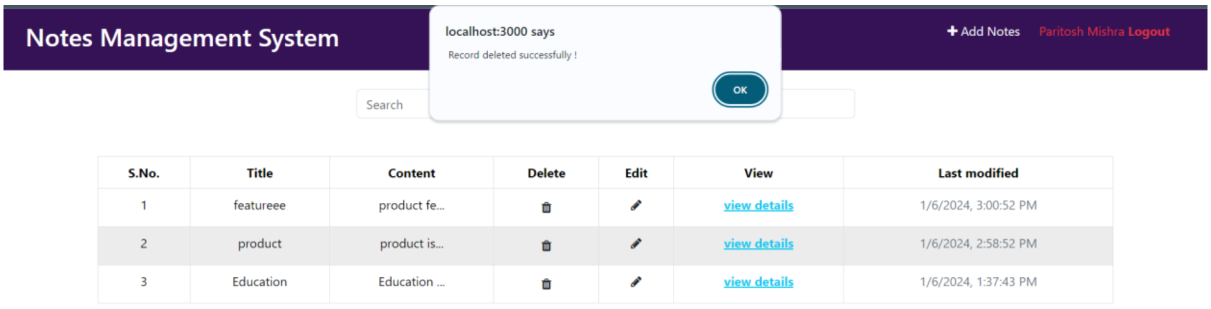
1. **Search box** - For searching desire note by it's title name.



feature

S.No.	Title	Content	Delete	Edit	View	Last modified
1	feature	product fe...	🗑	✎	view details	1/6/2024, 2:59:53 PM

2. Delete - For deleting note from database.

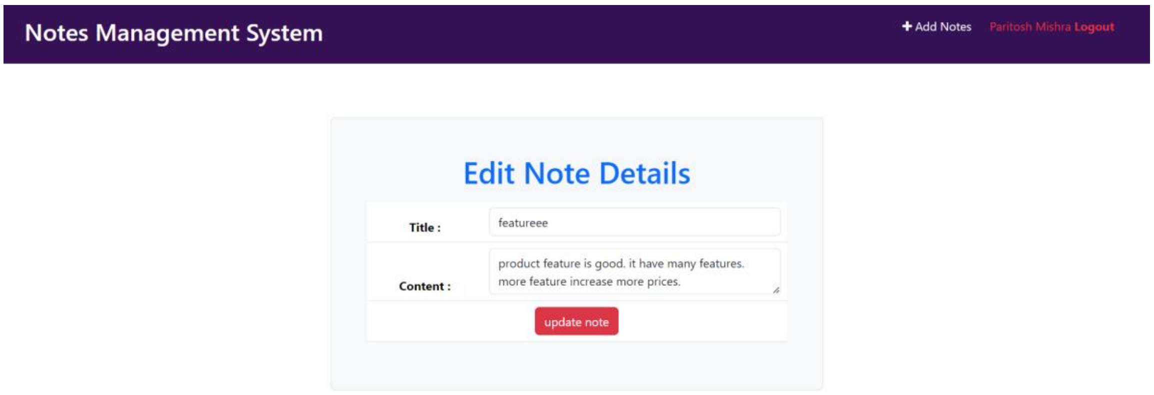


localhost:3000 says

Record deleted successfully !

S.No.	Title	Content	Delete	Edit	View	Last modified
1	featureee	product fe...	🗑	✎	view details	1/6/2024, 3:00:52 PM
2	product	product is...	🗑	✎	view details	1/6/2024, 2:58:52 PM
3	Education	Education ...	🗑	✎	view details	1/6/2024, 1:37:43 PM

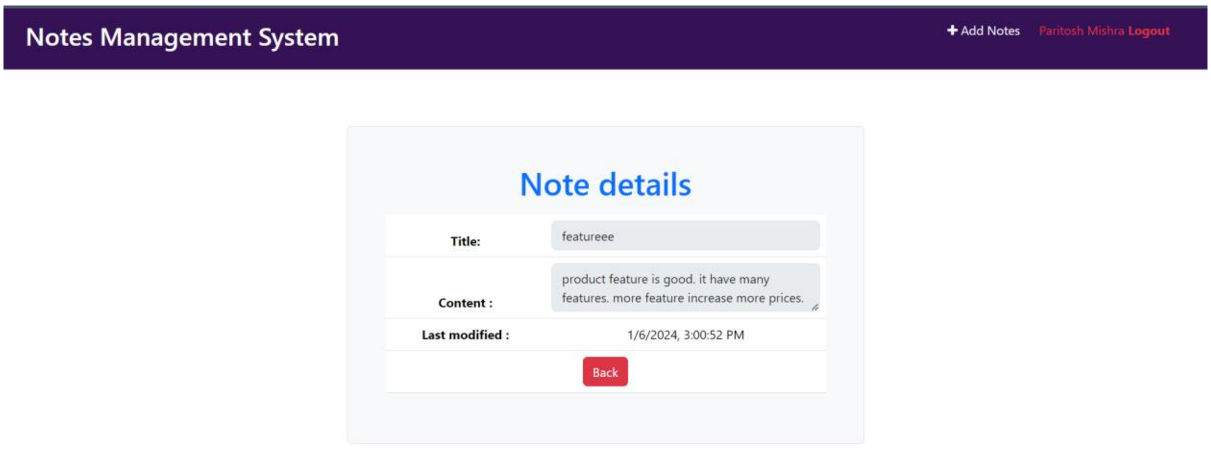
3. Edit - For Editing existing note.



Title : featureee

Content : product feature is good. it have many features. more feature increase more prices.

4. **View details** - view note in details.

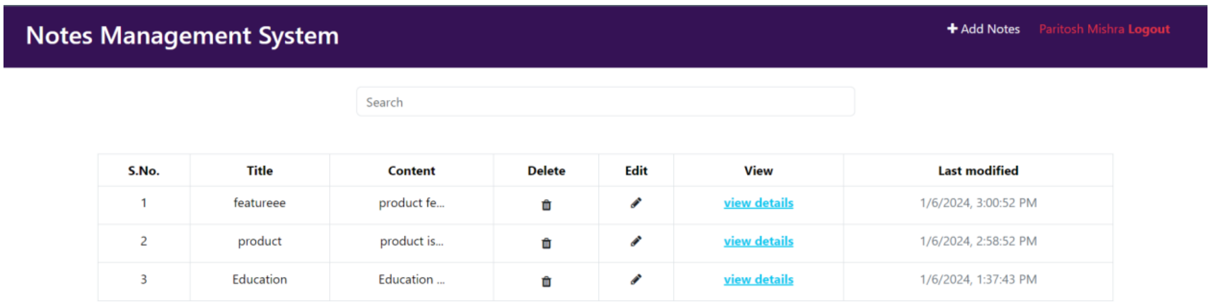


Note details

Title:	featureee
Content :	product feature is good. it have many features. more feature increase more prices.
Last modified :	1/6/2024, 3:00:52 PM

In right side above I provided two buttons **Addnote** and **logout**.

- Addnote for adding new note.
- Logout for logged out.

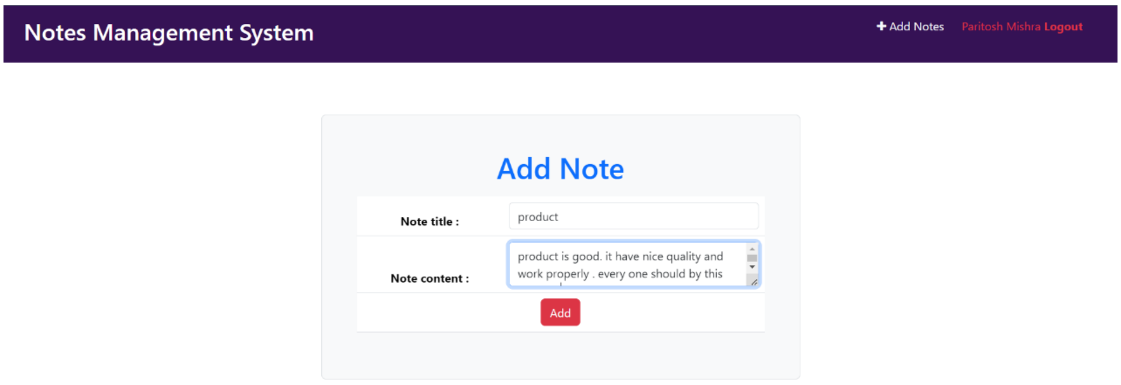


S.No.	Title	Content	Delete	Edit	View	Last modified
1	featureee	product fe...	🗑	✎	view details	1/6/2024, 3:00:52 PM
2	product	product is...	🗑	✎	view details	1/6/2024, 2:58:52 PM
3	Education	Education ...	🗑	✎	view details	1/6/2024, 1:37:43 PM

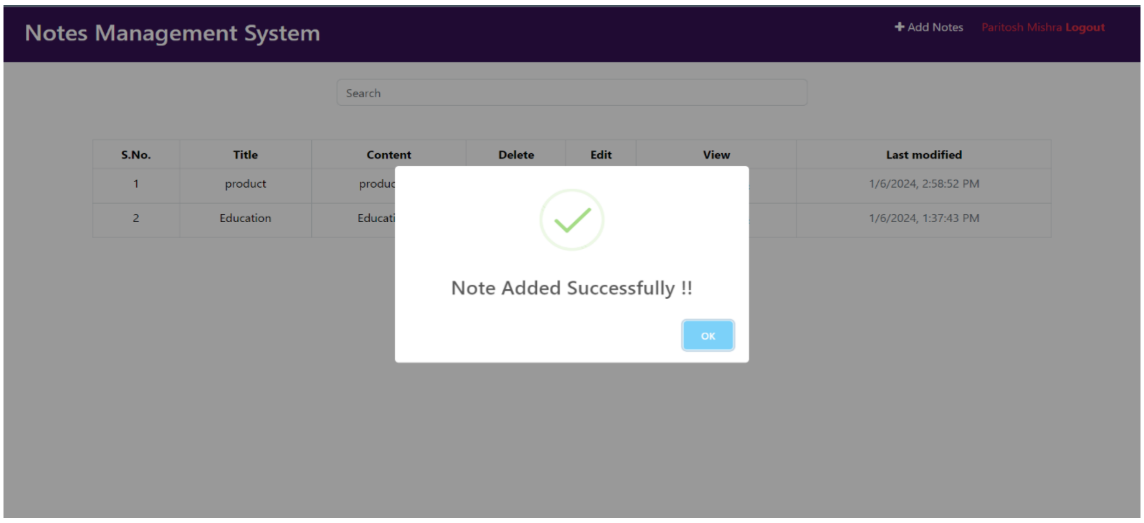
Api endpoints used –

- Get all notes - "http://localhost:4444/retrievenote";
- For edit note - "http://localhost:4444/editnote";
- For deleting note – " http:/localhost:4444/deletenote/"+id;
- View details- "http:/localhost:4444/note/"+userid

4. **Addnote -** When you click on Addnote button you will be directed to add note page that have a form with fields Title & content.
I have been used validation in this form, title & content with atleast 50 characters length is required fields & required to submit the form.



Note title :	product
Note content :	product is good. it have nice quality and work properly . every one should by this



S.No.	Title	Content	Delete	Edit	View	Last modified
1	product	produc				1/6/2024, 2:58:52 PM
2	Education	Educati				1/6/2024, 1:37:43 PM

Note Added Successfully !!

While submitting the form /addnote Api have been called from the backend nodejs.

Api endpoints used –

- Url = "http://localhost:4444/addnote"
- Method: Post
- Body: { name, content }
- author field get from localStorage.getItem("fullname");

Backend Services –

1. **/account –**

```
const accountapi = require("./accountapi");
app.use("/account", accountapi);
```

- **Description –**

- RestApis are the secure apis. So path of the api call /account has been changed to /accountapi. It protect database from direct access.

2. **/retrievenote –**

```
const manageuser = require("./userapi");
app.use("/retrievenote", manageuser);
```

- **Description –**

- RestApis are the secure apis. So path of the api call /retrievenote has been changed to /manageuser. It protect database from direct access.

3. **/**editnote –

```
const manageuser = require("./userapi");
 app.use("/editnote", manageuser);
```

- **Description –**

- RestApis are the secure apis. So path of the api call /editnote has been changed to /manageuser. It protect database from direct access.

4. **/** deletenote **–**

```
const manageuser = require("./userapi");
 app.use("/deletenote", manageuser);
```

- **Description –**

- RestApis are the secure apis. So path of the api call /deletenote has been changed to /manageuser. It protect database from direct access.

5. **/addnote –**

```
const manageuser = require("./userapi");
 app.use("/addnote", manageuser);
```

- **Description –**

- RestApis are the secure apis. So path of the api call /addnote has been changed to /manageuser. It protect database from direct access.

6. **/note –**

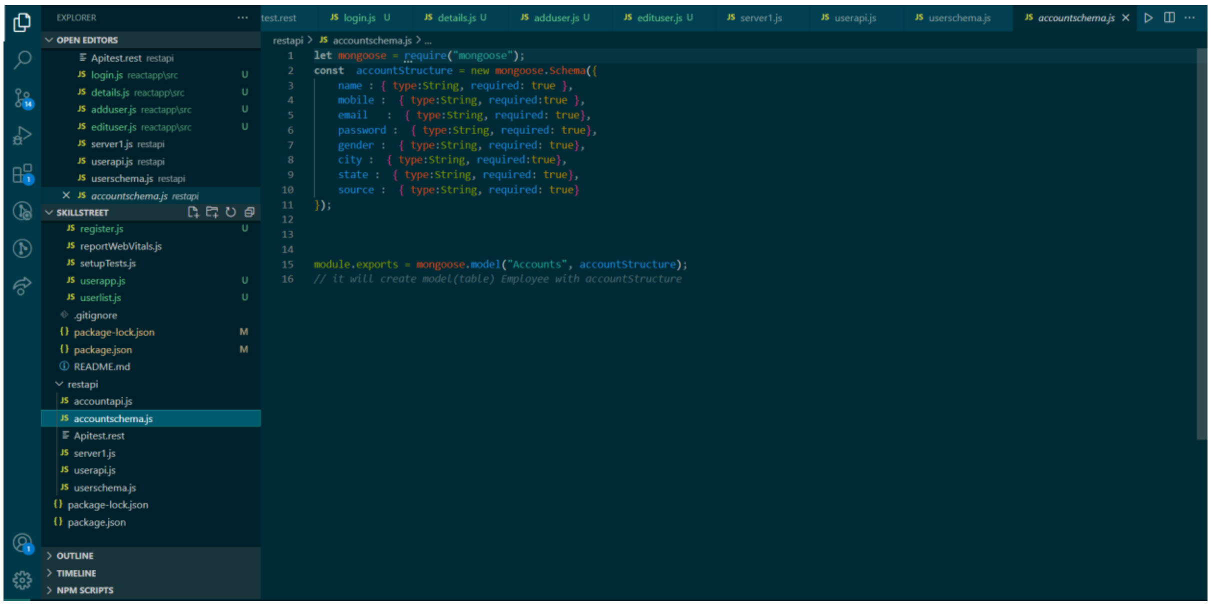
```
const manageuser = require("./userapi");
 app.use("/note", manageuser);
```

- **Description –**

- RestApis are the secure apis. So path of the api call /note has been changed to /manageuser. It protect database from direct access.

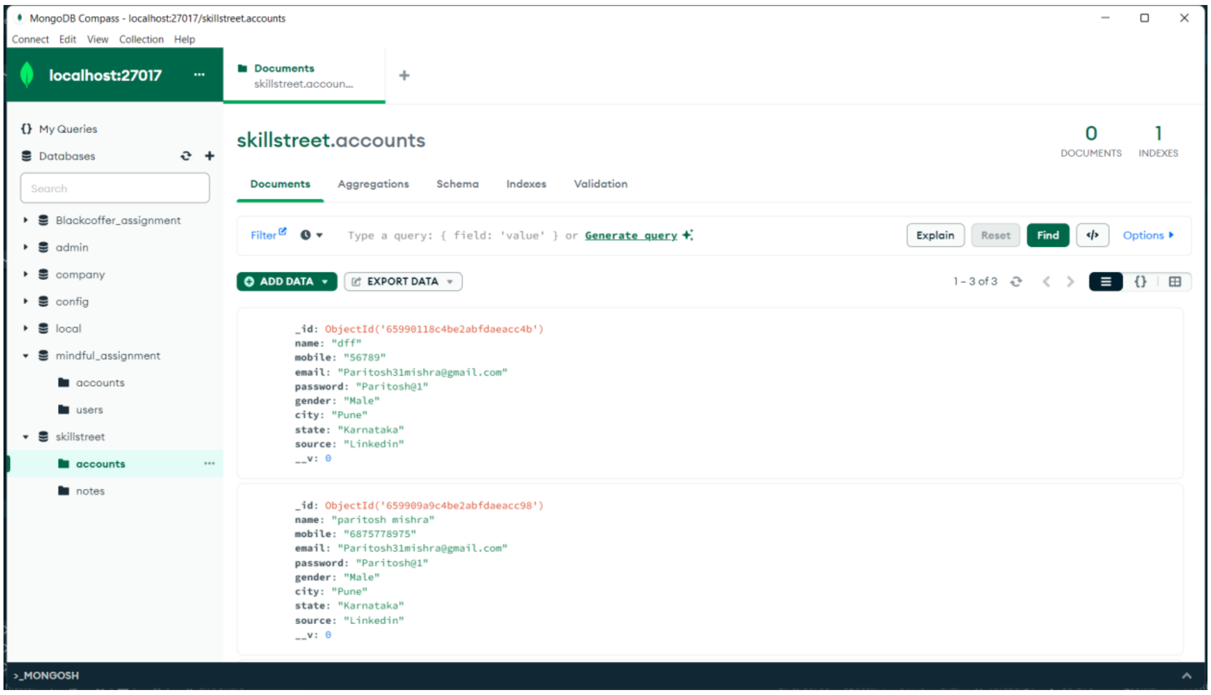
Mongo DB -

- **Account –**



```
let mongoose = require("mongoose");
const  accountStructure = new mongoose.Schema({
    name : { type:String, required: true },
    mobile :  { type:String, required:true },
    email   :  { type:String, required: true},
    password :  { type:String, required: true},
    gender :  { type:String, required: true},
    city :  { type:String, required:true},
    state :  { type:String, required: true},
    source :  { type:String, required: true}
});



module.exports = mongoose.model("Accounts", accountStructure);
// it will create model(table) Employee with accountStructure
```

Account schema



MongoDB Compass - localhost:27017/skillstreet.accounts

skillstreet.accounts

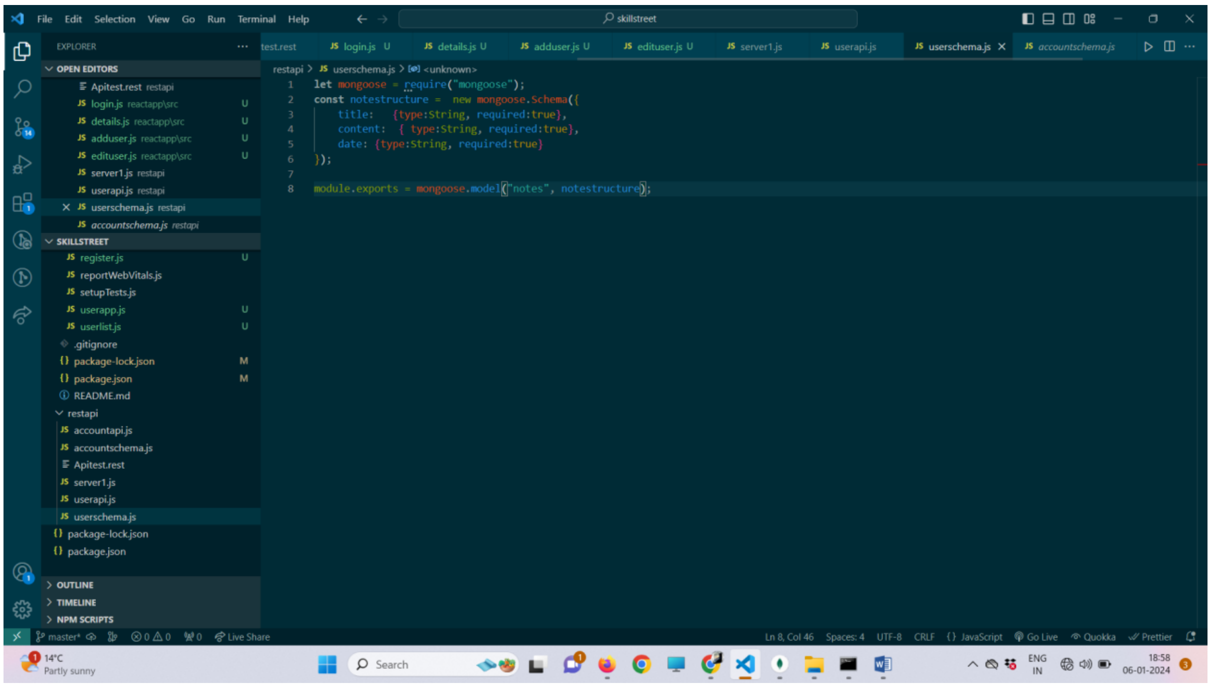
0 DOCUMENTS 1 INDEXES

```
_id: ObjectId('65990118c4be2abfdaeacc4b')
name: "dff"
mobile: "56789"
email: "Paritosh31mishra@gmail.com"
password: "Paritosh@1"
gender: "Male"
city: "Pune"
state: "Karnataka"
source: "Linkedin"
__v: 0
```

```
_id: ObjectId('659909a9c4be2abfdaeacc98')
name: "paritosh mishra"
mobile: "6875778975"
email: "Paritosh31mishra@gmail.com"
password: "Paritosh@1"
gender: "Male"
city: "Pune"
state: "Karnataka"
source: "Linkedin"
__v: 0
```

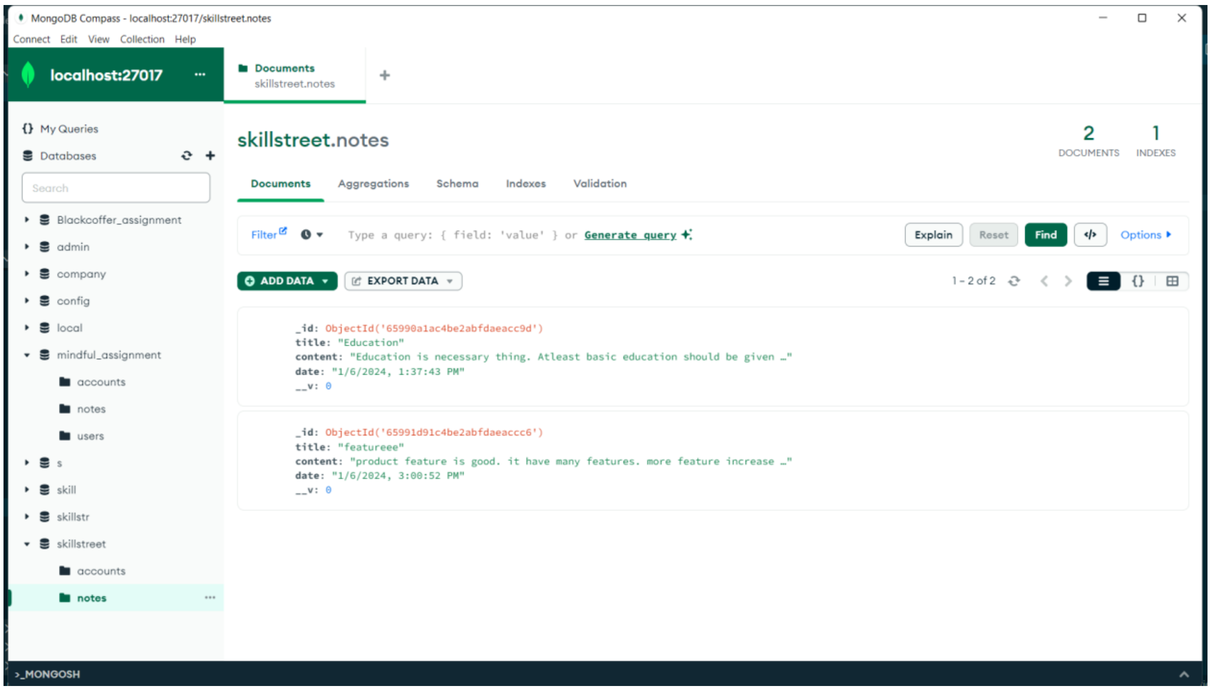
Account database

- **Notes –**



```
let mongoose = require("mongoose");
const notestructure =  new mongoose.Schema({
    title:   {type:String, required:true},
    content:  { type:String, required:true},
    date: {type:String, required:true}
});

module.exports = mongoose.model("notes", notestructure);
```

Note schema

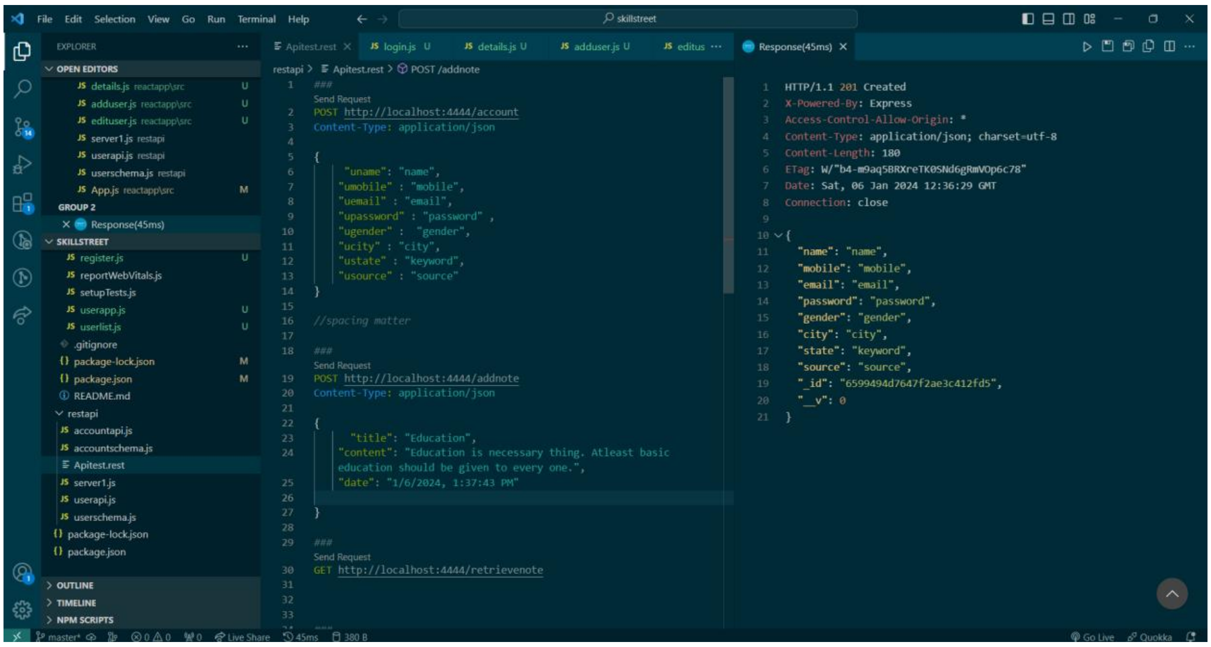


```
_id: ObjectId('65990a1ac4be2abfdaeacc9d')
title: "Education"
content: "Education is necessary thing. Atleast basic education should be given …"
date: "1/6/2024, 1:37:43 PM"
__v: 0

_id: ObjectId('65991d91c4be2abfdaeaccc6')
title: "featureee"
content: "product feature is good. it have many features. more feature increase …"
date: "1/6/2024, 3:00:52 PM"
__v: 0
```

Note database

APIs testing -

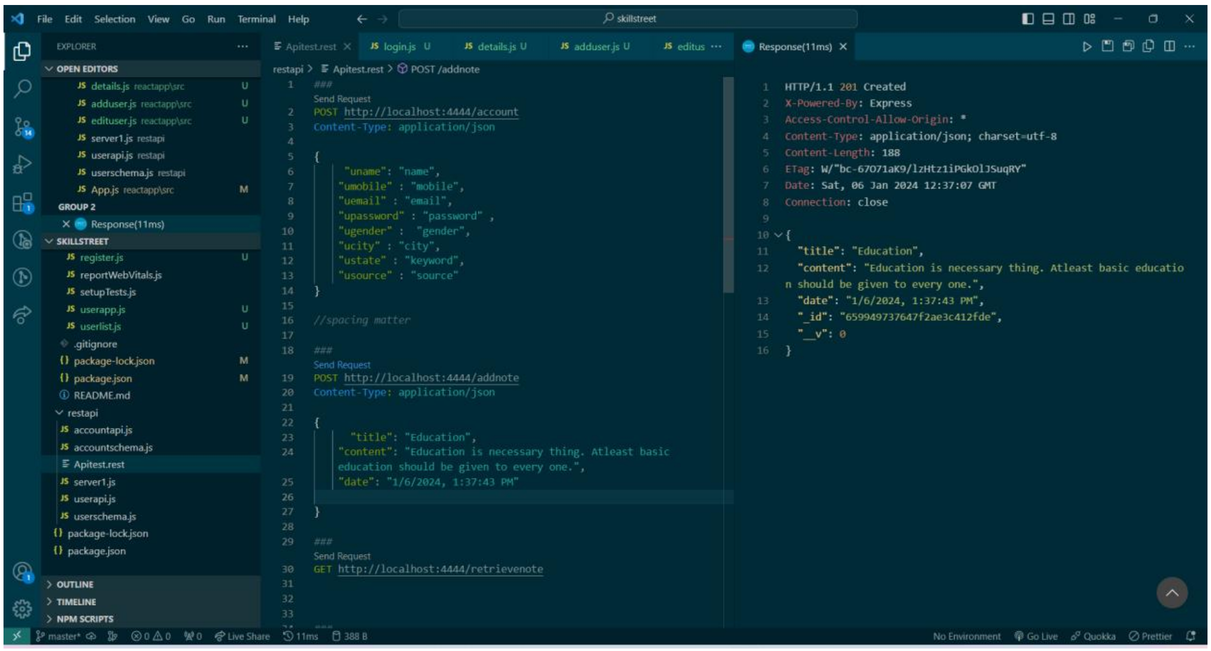
- **/account –**



```
###
Send Request
POST http://localhost:4444/account
Content-Type: application/json

{
    "uname": "name",
   "umobile" : "mobile",
   "uemail" : "email",
   "upassword" : "password" ,
   "ugender" :  "gender",
   "ucity" : "city",
   "ustate" : "keyword",
   "usource" : "source"
}

//spacing matter

###
Send Request
POST http://localhost:4444/addnote
Content-Type: application/json

{
     "title": "Education",
   "content": "Education is necessary thing. Atleast basic
   education should be given to every one.",
   "date": "1/6/2024, 1:37:43 PM"

}

###
Send Request
GET http://localhost:4444/retrievenote
```

```
HTTP/1.1 201 Created
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 180
ETag: W/"b4-m9aq5BRXreTK0SNd6gRmVOp6c78"
Date: Sat, 06 Jan 2024 12:36:29 GMT
Connection: close

{
  "name": "name",
  "mobile": "mobile",
  "email": "email",
  "password": "password",
  "gender": "gender",
  "city": "city",
  "state": "keyword",
  "source": "source",
  "_id": "6599494d7647f2ae3c412fd5",
  "__v": 0
}
```

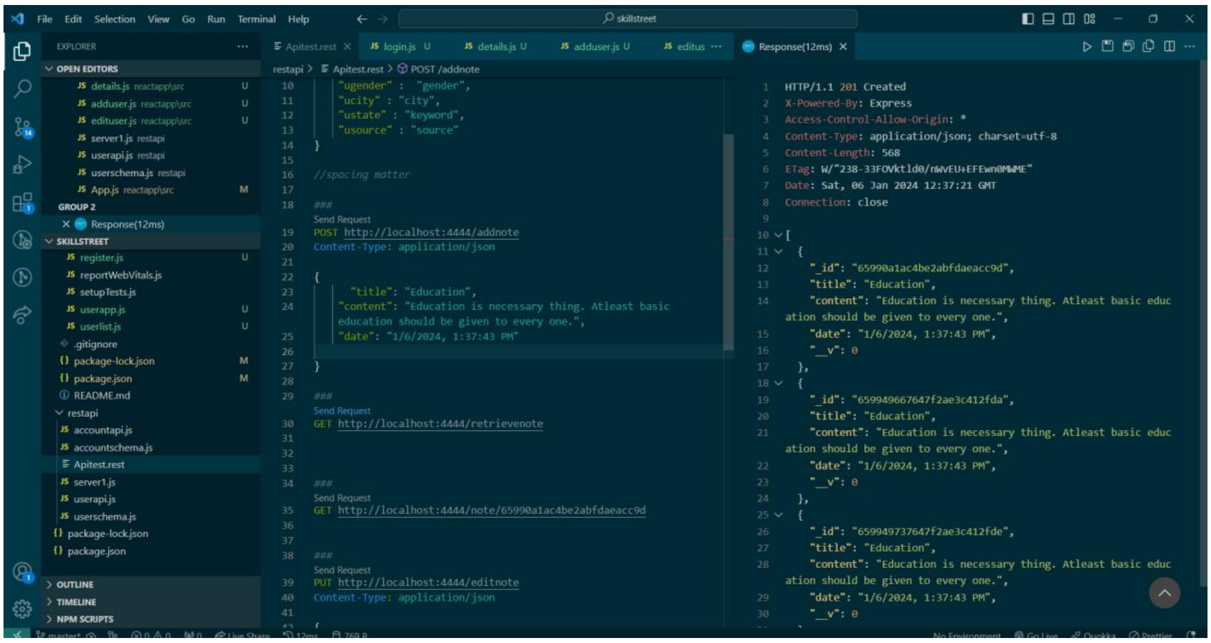
- **/addnote –**



```
###
Send Request
POST http://localhost:4444/account
Content-Type: application/json

{
    "uname": "name",
   "umobile" : "mobile",
   "uemail" : "email",
   "upassword" : "password" ,
   "ugender" :  "gender",
   "ucity" : "city",
   "ustate" : "keyword",
   "usource" : "source"
}

//spacing matter

###
Send Request
POST http://localhost:4444/addnote
Content-Type: application/json

{
     "title": "Education",
   "content": "Education is necessary thing. Atleast basic
   education should be given to every one.",
   "date": "1/6/2024, 1:37:43 PM"

}

###
Send Request
GET http://localhost:4444/retrievenote
```

```
HTTP/1.1 201 Created
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 188
ETag: W/"bc-67O71aK9/lzHtz1iPGkOlJSuqRY"
Date: Sat, 06 Jan 2024 12:37:07 GMT
Connection: close

{
  "title": "Education",
  "content": "Education is necessary thing. Atleast basic educatio
n should be given to every one.",
  "date": "1/6/2024, 1:37:43 PM",
  "_id": "659949737647f2ae3c412fde",
  "__v": 0
}
```

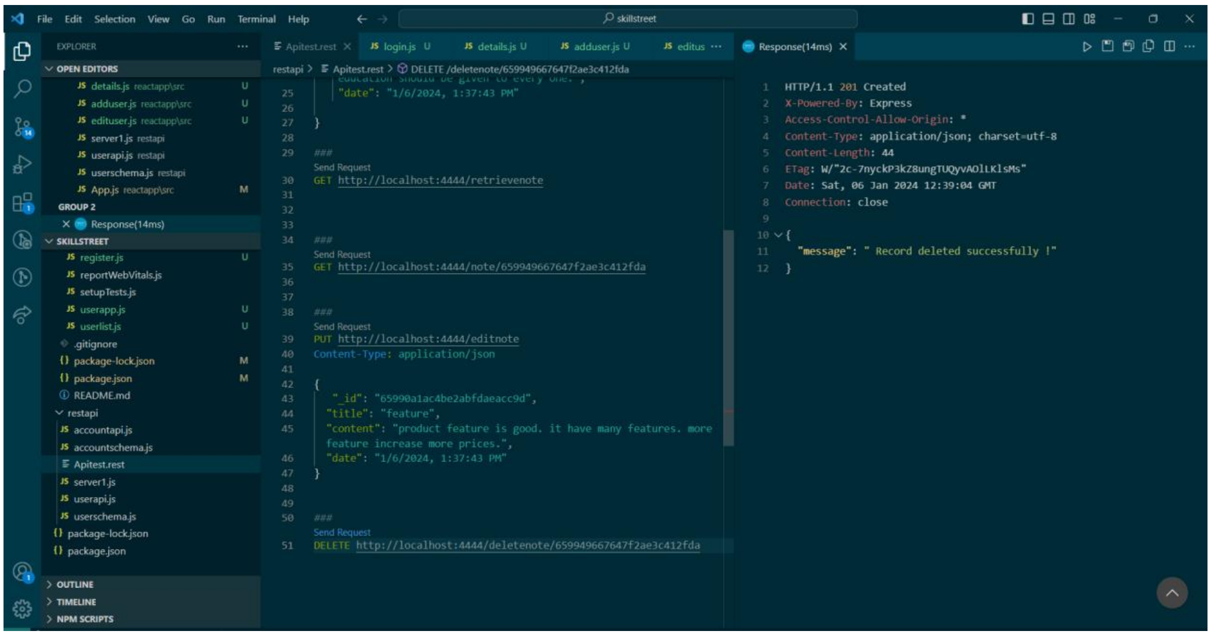
- **/retrievenote-**



```
    "ugender" :  "gender",
    "ucity" : "city",
    "ustate" : "keyword",
    "usource" : "source"
}

//spacing matter

###
Send Request
POST http://localhost:4444/addnote
Content-Type: application/json

{
      "title": "Education",
    "content": "Education is necessary thing. Atleast basic
    education should be given to every one.",
    "date": "1/6/2024, 1:37:43 PM"

}

###
Send Request
GET http://localhost:4444/retrievenote



###
Send Request
GET http://localhost:4444/note/65990a1ac4be2abfdaeacc9d


###
Send Request
PUT http://localhost:4444/editnote
Content-Type: application/json
```

```
HTTP/1.1 201 Created
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 568
ETag: W/"238-33FOVktld0/nWvEU+EFEwn0MWME"
Date: Sat, 06 Jan 2024 12:37:21 GMT
Connection: close

[
  {
    "_id": "65990a1ac4be2abfdaeacc9d",
    "title": "Education",
    "content": "Education is necessary thing. Atleast basic educ
ation should be given to every one.",
    "date": "1/6/2024, 1:37:43 PM",
    "__v": 0
  },
  {
    "_id": "659949667647f2ae3c412fda",
    "title": "Education",
    "content": "Education is necessary thing. Atleast basic educ
ation should be given to every one.",
    "date": "1/6/2024, 1:37:43 PM",
    "__v": 0
  },
  {
    "_id": "659949737647f2ae3c412fde",
    "title": "Education",
    "content": "Education is necessary thing. Atleast basic educ
ation should be given to every one.",
    "date": "1/6/2024, 1:37:43 PM",
    "__v": 0
```

- **/deletenote-**



```
    "date": "1/6/2024, 1:37:43 PM"

}

###
Send Request
GET http://localhost:4444/retrievenote



###
Send Request
GET http://localhost:4444/note/659949667647f2ae3c412fda


###
Send Request
PUT http://localhost:4444/editnote
Content-Type: application/json

{
   "_id": "65990a1ac4be2abfdaeacc9d",
  "title": "feature",
  "content": "product feature is good. it have many features. more
  feature increase more prices.",
  "date": "1/6/2024, 1:37:43 PM"
}



###
Send Request
DELETE http://localhost:4444/deletenote/659949667647f2ae3c412fda
```

```
HTTP/1.1 201 Created
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 44
ETag: W/"2c-7nyckP3kZ8ungTUQyvAOlLKlsMs"
Date: Sat, 06 Jan 2024 12:39:04 GMT
Connection: close

{
  "message": " Record deleted successfully !"
}
```

Setup Details:

- Clone the repository <link> to a directory.
- Start the react server by going to the directory of the skillstreet > reactapp and run ‘npm start’
- Start the node js server by going to the directory of skillstreet > nodeapp and run ‘nodemon server1.js’ . Here server1.js is the name of the node js service file.