Task

Aim: To implement CRUD operations for managing student details using MongoDB, an Express-based REST API will be developed, leveraging the Mongoose package for database interaction. Test REST-API using POSTMAN Tool. Send Email to admin email, Upon successful deletion of a record via the REST API, an email notification will be dispatched using the nodemailer package.

Description:

A REST API for managing student details using MongoDB and Express, with Mongoose for database interaction, will be developed. CRUD operations can be tested using the POSTMAN tool. Additionally, upon successful deletion of a record via the API, an email notification will be sent to the admin email using the nodemailer package, enhancing the system's functionality.

Source Code:

Exdemo.js

```
const express = require('express')
const main = require('./Email.js');
const mongoose = require('mongoose');
const app = express()
const port=3000
app.use(express.static('public'))
app.use(express.json())
//app.use(express.static('public'))
const UserModel = require('./models/Users')
mongoose.connect('mongodb://localhost:27017/mydb')
.then(()=>{console.log("Database connected successfully!");})
.catch((err)=>{console.log("Database does not connected!");})

app.get('/addStatic',(req,res)=>
{
   var mydata =
   {
      User_name:'Trushang',
      User_gender:'Male'
   }

   UserModel.create(mydata)
   .then(()=>{console.log("Record added")})
   .catch((err)=>{console.log("NO record Inserted"+err)})
   res.send("record added")
```

```
})

app.post('/api/add',(req,res)=>
{
   console.log(req.body)
   var mydata = req.body;
   UserModel.create(mydata)
   .then(()=>{console.log("Record Added")})
   .catch((err)=>{console.log("Record not inserted"+err)})
   res.send(JSON.stringify("RECord added"))
})

app.post('/register',(req,res)=>{
      console.log(req.body);
      UserModel.find()
      .then(data=>res.json(data))
      .catch(err=>console.log(err))
})
app.get('/',(req,res)=>{
    res.send('Hello World!')
})
 app.get('/home',(req,res)=>
 {
    res.send(__dirname+'/home.html')
 })

 app.get('/display',(req,res)=>
 {
   UserModel.find()
   .then(data=>res.json(data))
   .catch(err=>console.log(err))
 });

 main();
 app.get('/update', (req, res) => {
   UserModel.updateOne({ User_name: 'Trushang Patel' }, { User_name: 'Trushang'
})
       .then(() => { console.log("Record Updated") })
       .catch((err) => { console.log("Record not updated" + err) })
   res.send("Record Updated");
});
```

```
app.get('/delete', (req, res) => {
   UserModel.deleteOne({ User_name: 'Trushang' })
        .then(() => { console.log("Record Deleted") })
        .catch((err) => { console.log("Record not Deleted" + err) })
   res.send("Record Deleted");
});

 app.listen(port,()=>{
    console.log(`Example app listening on port ${port}`)
 });
```

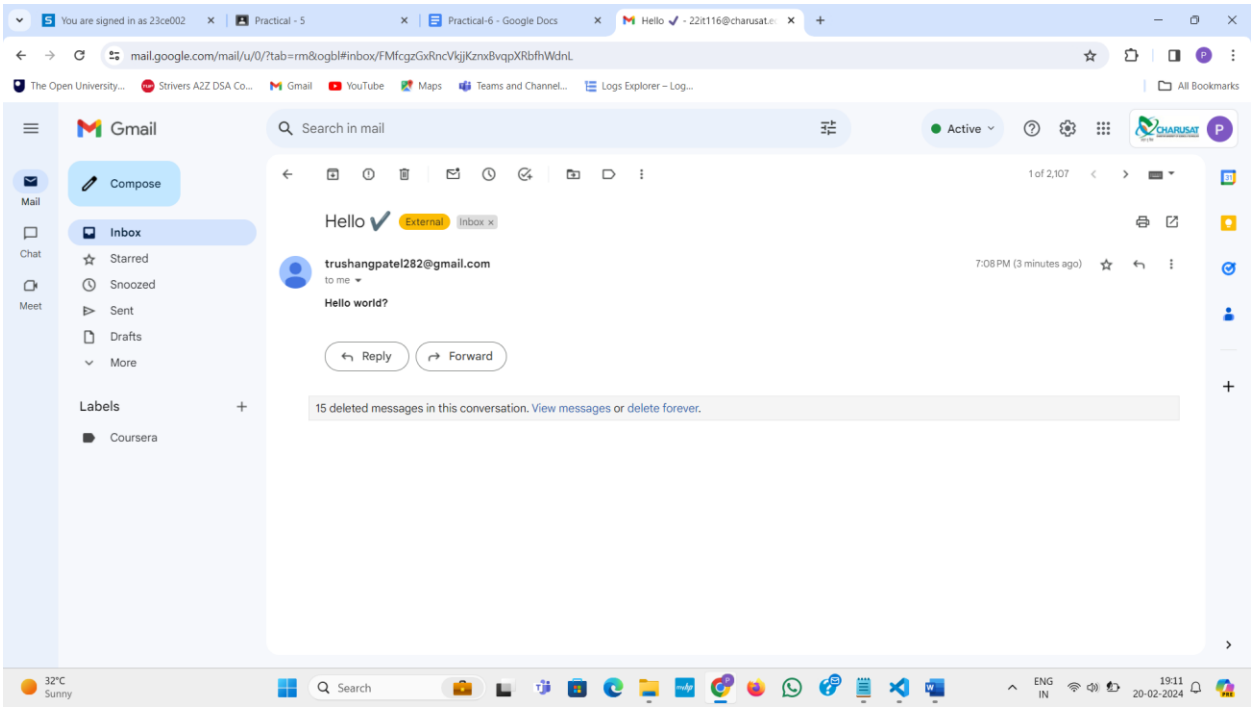
Email.js:

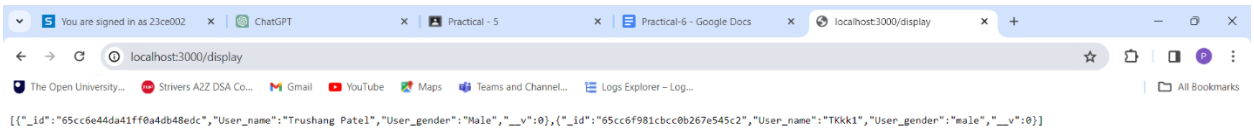
```
const nodemailer = require("nodemailer");
// Create a nodemailer transporter
const transporter = nodemailer.createTransport({
host: "smtp.gmail.com",
port: 465,
secure: true,
auth: {
user: "trushangpatel282@gmail.com",
pass: "aqfzngcdkrimhzbu",
},
});
// Define the email content
const emailOptions = {
from: 'trushangpatel282@gmail.com>', // sender address
to: "22it116@charusat.edu.in", // list of receivers
subject: "Hello ✓", // Subject line
text: "Hello world?", // plain text body
html: "<b>Hello world?</b>", // html body
};
// async..await is not allowed in global scope, must use a wrapper
const main=async ()=> {
try {
// Send mail with the defined transport object
const info = await transporter.sendMail(emailOptions);
console.log("Message sent: %s", info.messageId);
// Message sent: <b658f8ca-6296-ccf4-8306-87d57a0b4321@example.com
// NOTE: You can check your email delivery status and preview at
```

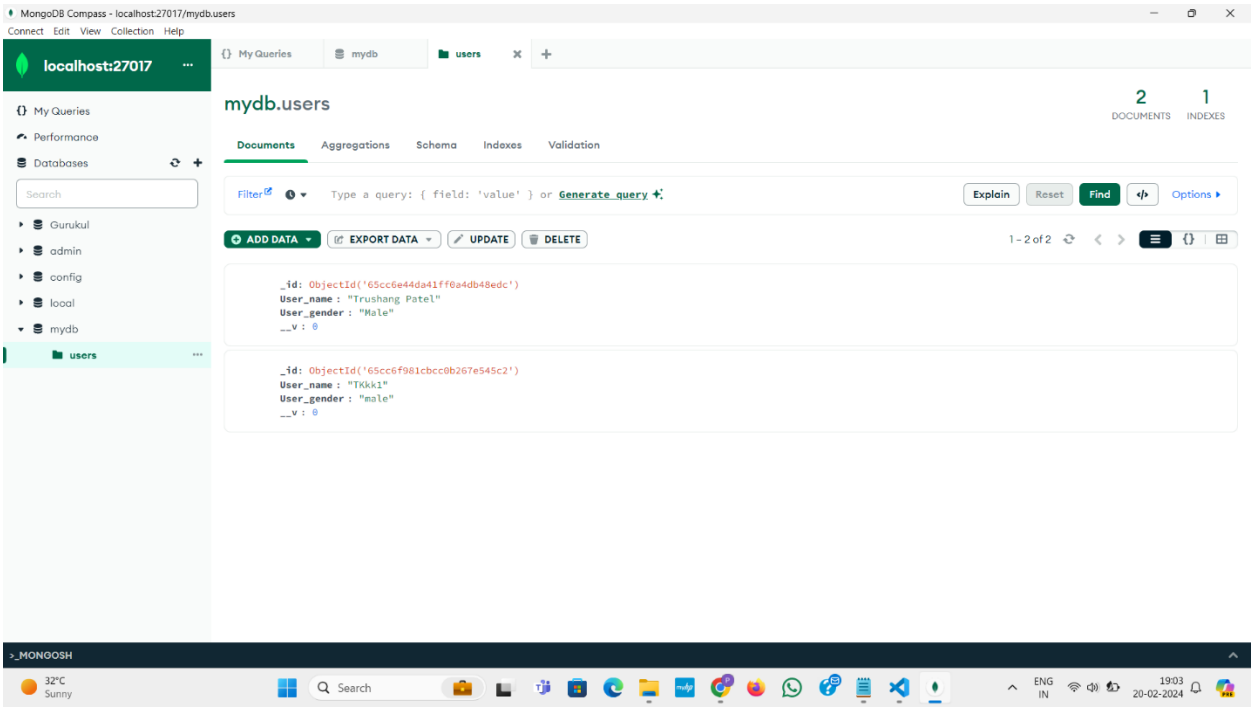
```
//https://forwardemail.net/my-account/emails
} catch (error) {
console.error("Error sending email:", error);
}
}
// Call the main function


module.exports = main;
```

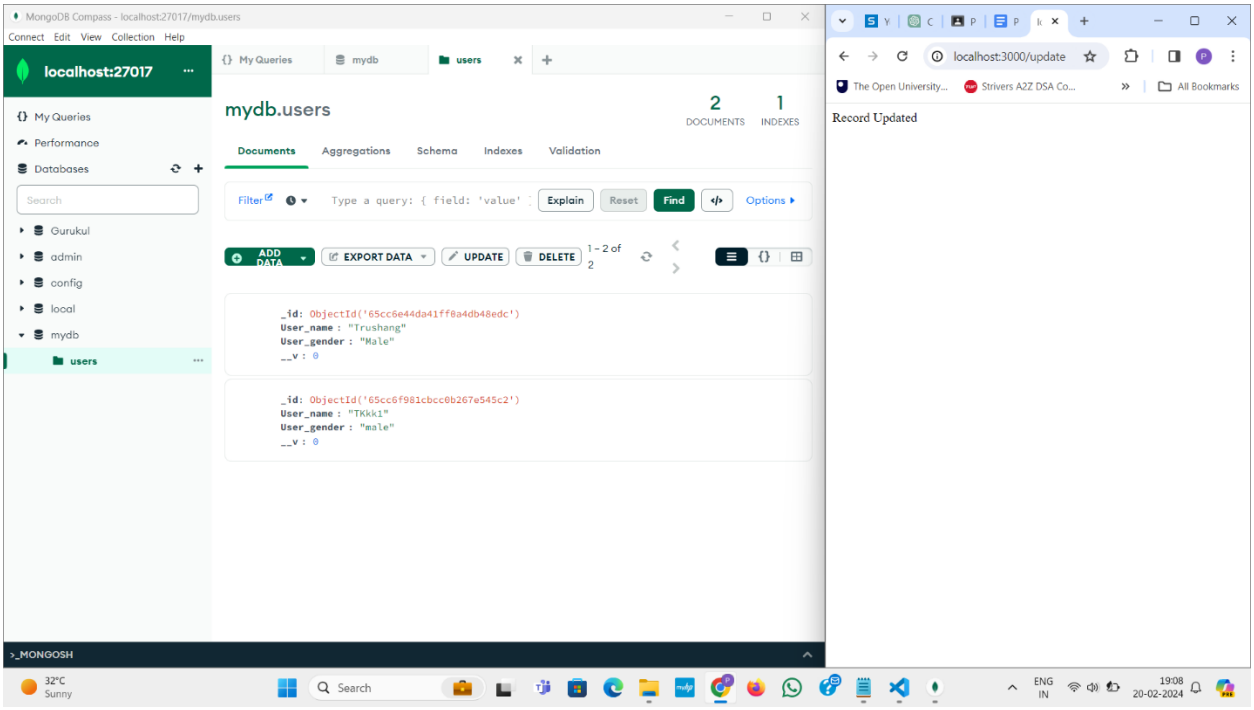
Output:

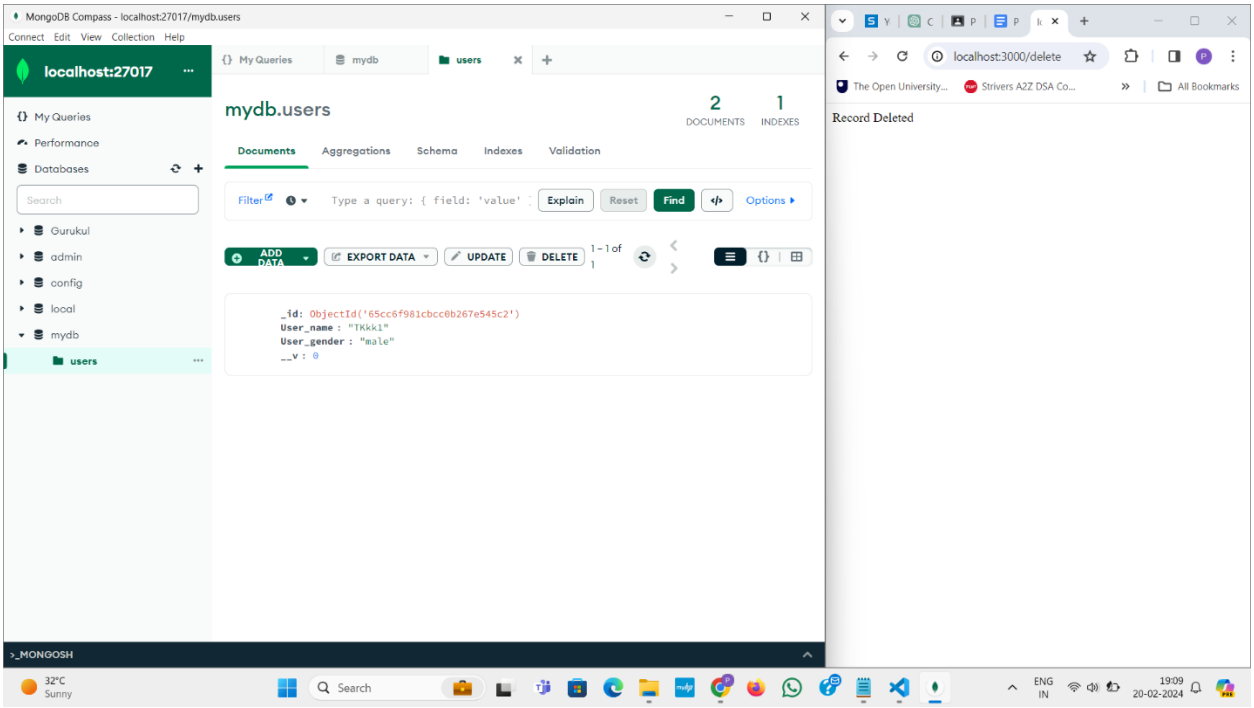
```
C:\SEM-4\FSWD\WEEK-5>nodemon exdemo
[nodemon] 3.0.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node exdemo.js`
Example app listening on port 3000
Database connected successfully!
Message sent: <8e5362bc-0af2-61ed-7af9-1b46e45b4105@gmail.com>
Record Updated
[nodemon] restarting due to changes...
[nodemon] starting `node exdemo.js`
Example app listening on port 3000
Database connected successfully!
[nodemon] restarting due to changes...
[nodemon] starting `node exdemo.js`
Example app listening on port 3000
Database connected successfully!
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node exdemo.js`
Example app listening on port 3000
Database connected successfully!
Message sent: <70797e13-8df6-e853-428c-bddee0b190c3@gmail.com>
Record Updated
Record Deleted
```


Gmail
Search in mail
Compose
Inbox
Starred
Snoozed
Sent
Drafts
More
Labels
Coursera
Hello
External
Inbox
trushangpatel282@gmail.com
to me
7:08 PM (3 minutes ago)
Hello world?
Reply
Forward
15 deleted messages in this conversation. View messages or delete forever.


localhost:3000/display
[{"_id":"65cc6e44da41ff0a4db48edc","User_name":"Trushang Patel","User_gender":"Male","__v":0},{"_id":"65cc6f981cbcc0b267e545c2","User_name":"TKkk1","User_gender":"male","__v":0}]


MongoDB Compass - localhost:27017/mydb.users
Connect Edit View Collection Help
localhost:27017
My Queries
Performance
Databases
Search
Gurukul
admin
config
local
mydb
users
My Queries
mydb
users
mydb.users
2 DOCUMENTS
1 INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
1 - 2 of 2
_id: ObjectId('65cc6e44da41ff0a4db48edc')
User_name : "Trushang Patel"
User_gender : "Male"
__v : 0
_id: ObjectId('65cc6f981cbcc0b267e545c2')
User_name : "TKkk1"
User_gender : "male"
__v : 0
>_MONGOSH
32°C Sunny
Search
ENG IN
19:03 20-02-2024


MongoDB Compass - localhost:27017/mydb.users
localhost:27017
mydb.users
2 DOCUMENTS
1 INDEXES
Documents
Aggregations
Schema
Indexes
Validation
ADD DATA
EXPORT DATA
UPDATE
DELETE
1 - 2 of 2
_id: ObjectId('65cc6e44da41ff0a4db48edc')
User_name : "Trushang"
User_gender : "Male"
__v : 0
_id: ObjectId('65cc6f981cbcc0b267e545c2')
User_name : "TKkk1"
User_gender : "male"
__v : 0
>_MONGOSH
localhost:3000/update
Record Updated
32°C Sunny
Search
ENG IN
19:08 20-02-2024



Learning Outcome:

In this task we learn about how to connect MongoDB with node js and how to send email using node js.