

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

Email.js:

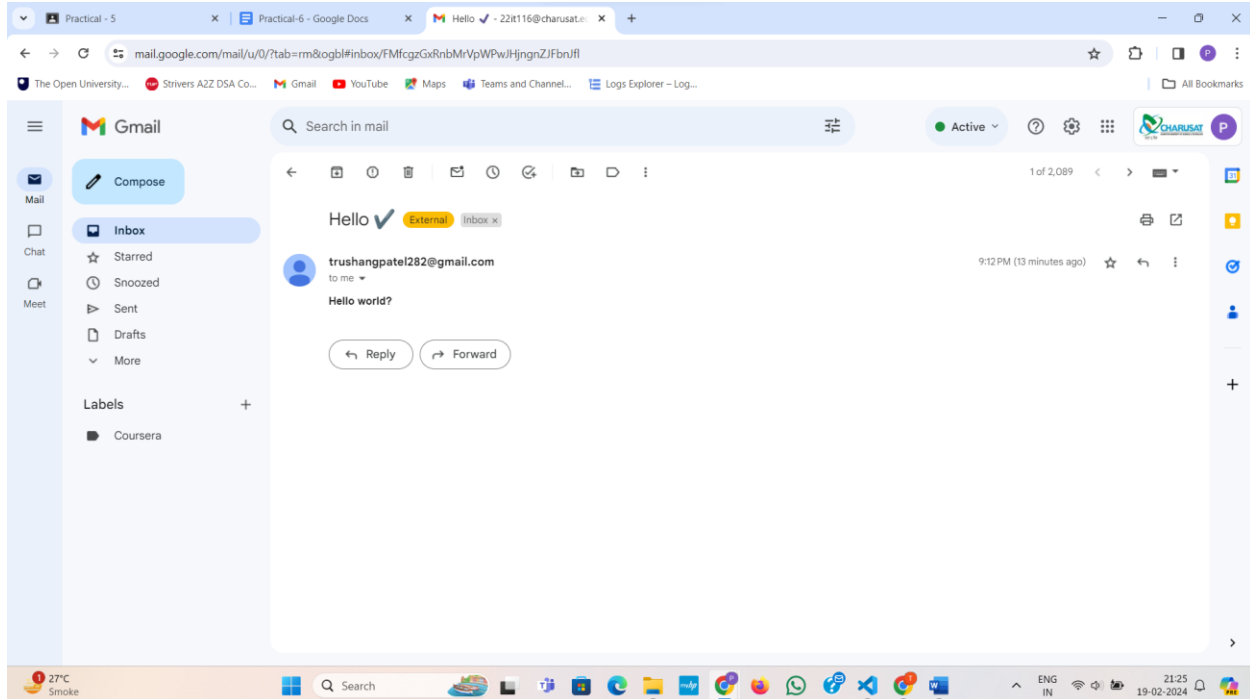
```
const nodemailer = require("nodemailer");
// Create a nodemailer transporter
const transporter = nodemailer.createTransport({
  host: "smtp.gmail.com",
  port: 465,
  secure: true,
  auth: {
    user: "trushangpatel282@gmail.com",
    pass: "aqfzngcdkrimhzb",
  },
});
// Define the email content
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:

```
PROBLEMS PORTS OUTPUT DEBUG CONSOLE TERMINAL

[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!
Message sent: <c6088987-d031-f0d3-1576-1c1b265320ec@gmail.com>
[nodemon] restarting due to changes...
[nodemon] starting `node exdemo.js`
Example app listening on port 3000
Database connected successfully!
Message sent: <4b2bebcf-737b-988d-0a45-bbf0b59b9a0d@gmail.com>
```



Learning Outcome:

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