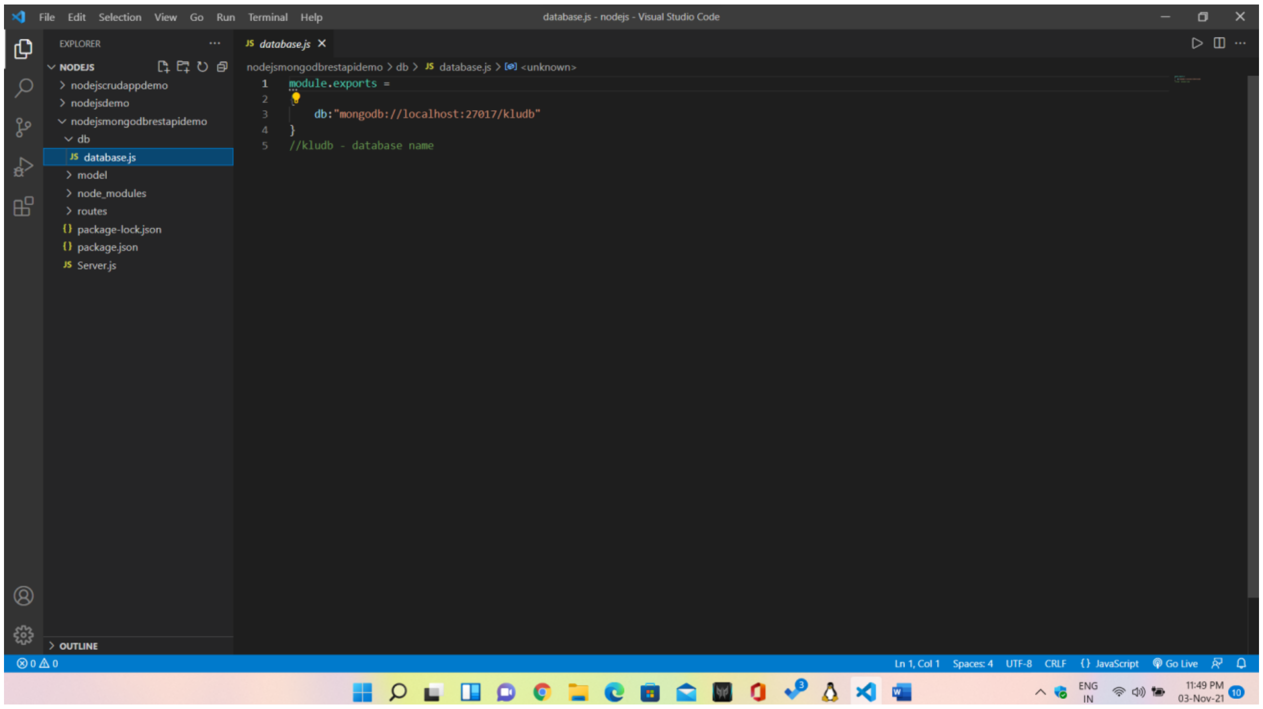
<u>Skill 10</u>

ID: 2000030384

Name: Jatla Nikhil Sai Lalith

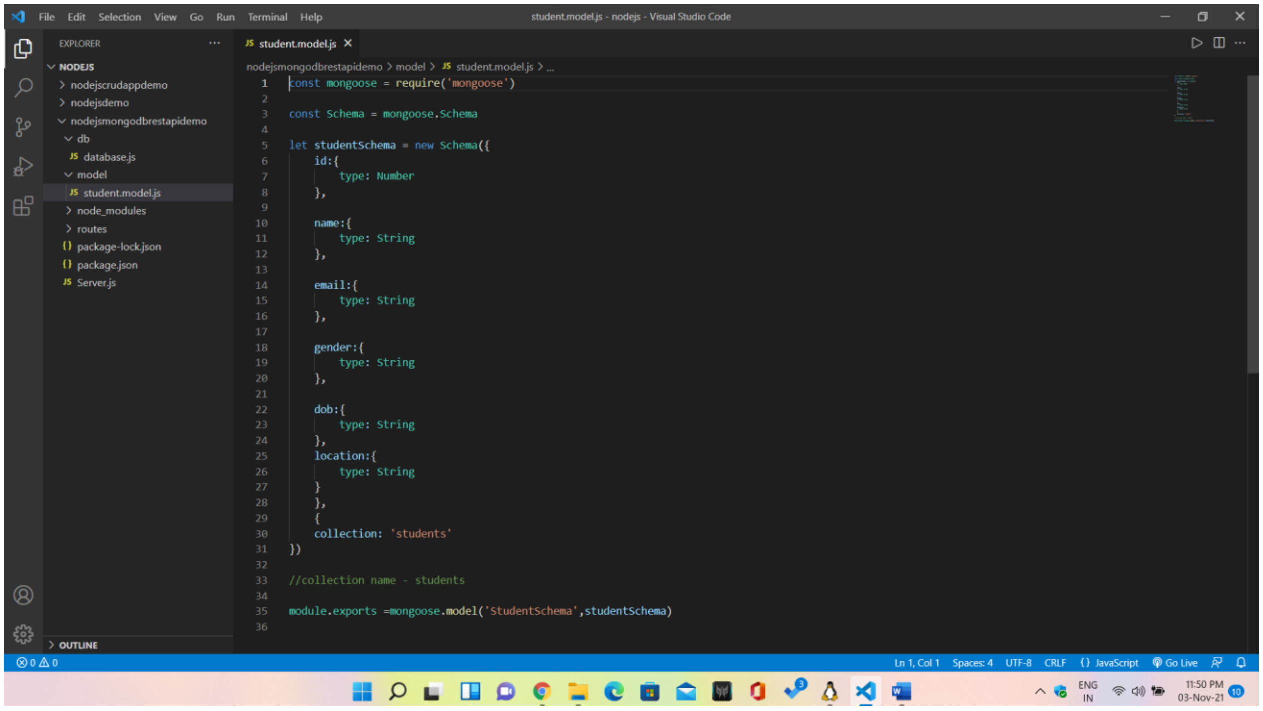
Section No: S09

Database.js:



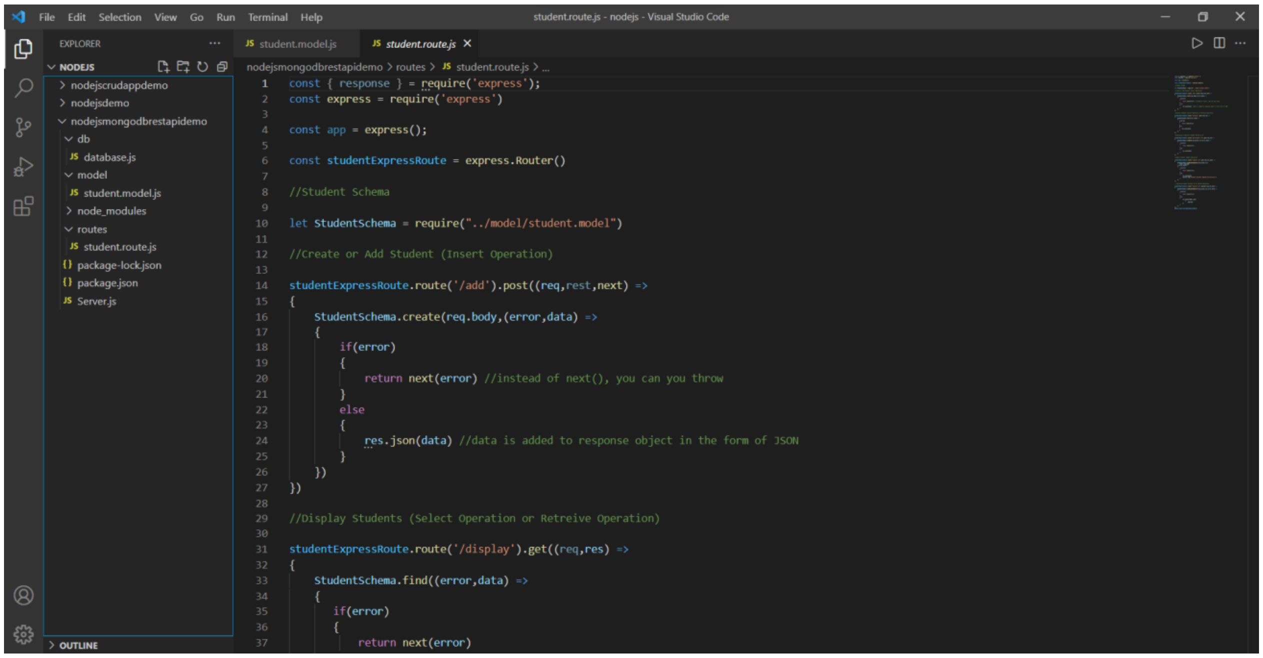
```
module.exports =
{
    db:"mongodb://localhost:27017/kludb"
}
//kludb - database name
```

Student.model.js:

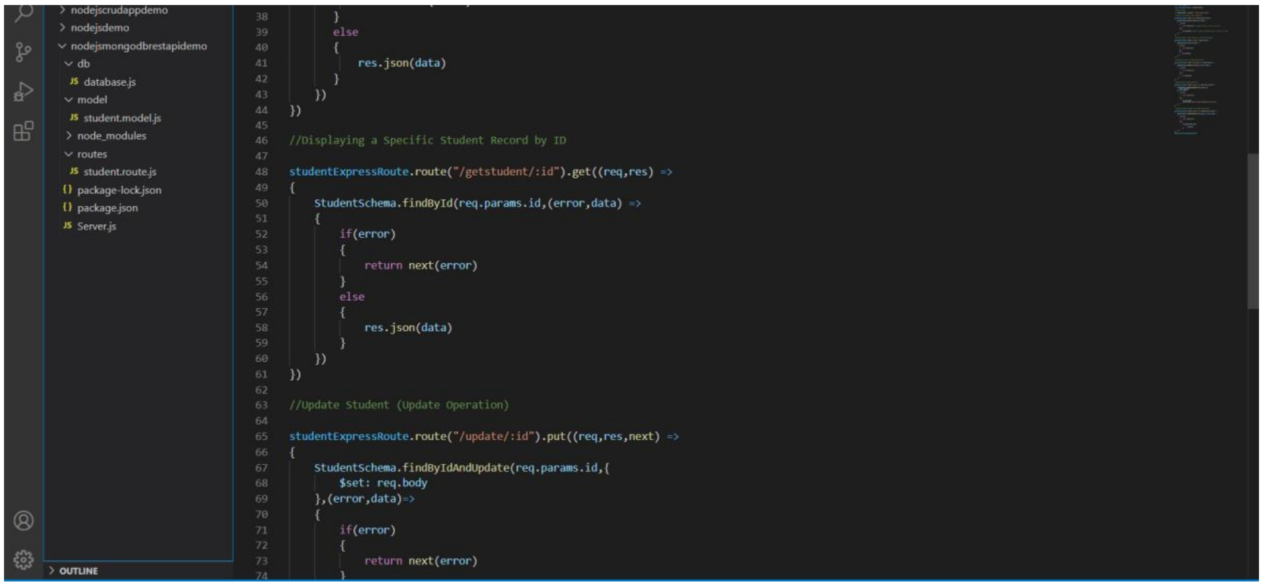


```
const mongoose = require('mongoose')

const Schema = mongoose.Schema

let studentSchema = new Schema({
    id:{
        type: Number
    },

    name:{
        type: String
    },

    email:{
        type: String
    },

    gender:{
        type: String
    },

    dob:{
        type: String
    },
    location:{
        type: String
    }
    },
    {
    collection: 'students'
})

//collection name - students

module.exports =mongoose.model('StudentSchema',studentSchema)
```

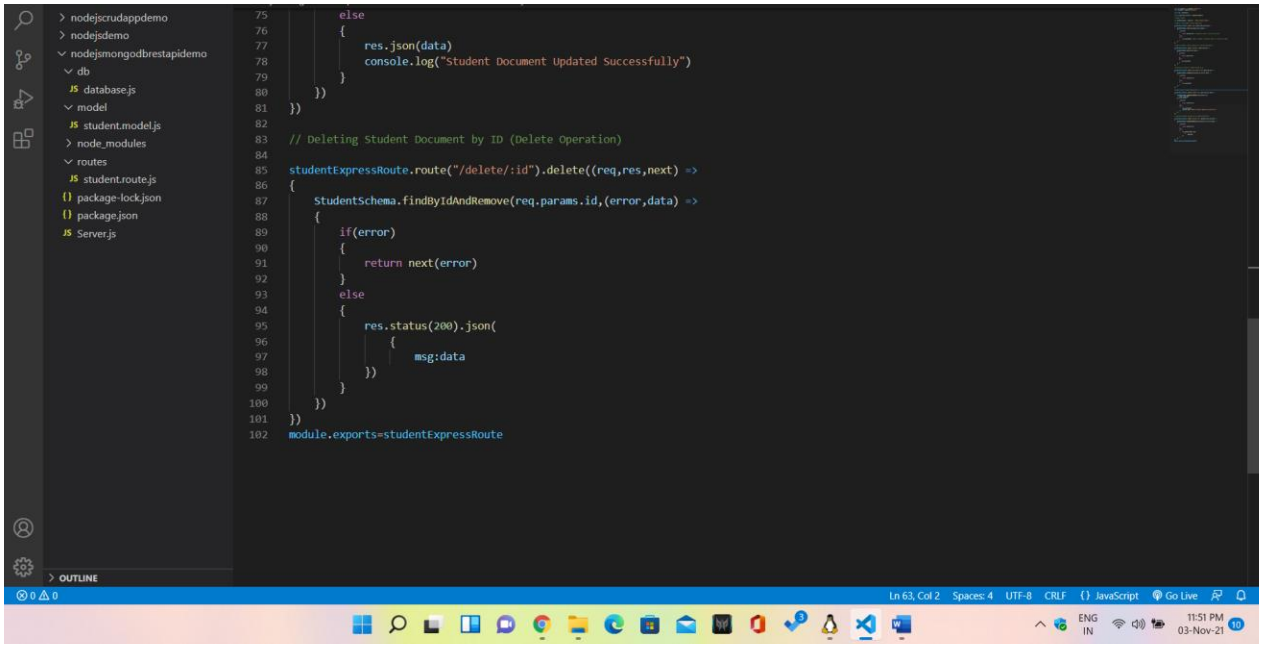
Student.route.js:



```
const { response } = require('express');
const express = require('express')

const app = express();

const studentExpressRoute = express.Router()

//Student Schema

let StudentSchema = require("../model/student.model")

//Create or Add Student (Insert Operation)

studentExpressRoute.route('/add').post((req,rest,next) =>
{
    StudentSchema.create(req.body,(error,data) =>
    {
        if(error)
        {
            return next(error) //instead of next(), you can you throw
        }
        else
        {
            res.json(data) //data is added to response object in the form of JSON
        }
    })
})

//Display Students (Select Operation or Retreive Operation)

studentExpressRoute.route('/display').get((req,res) =>
{
    StudentSchema.find((error,data) =>
    {
       if(error)
       {
           return next(error)
```

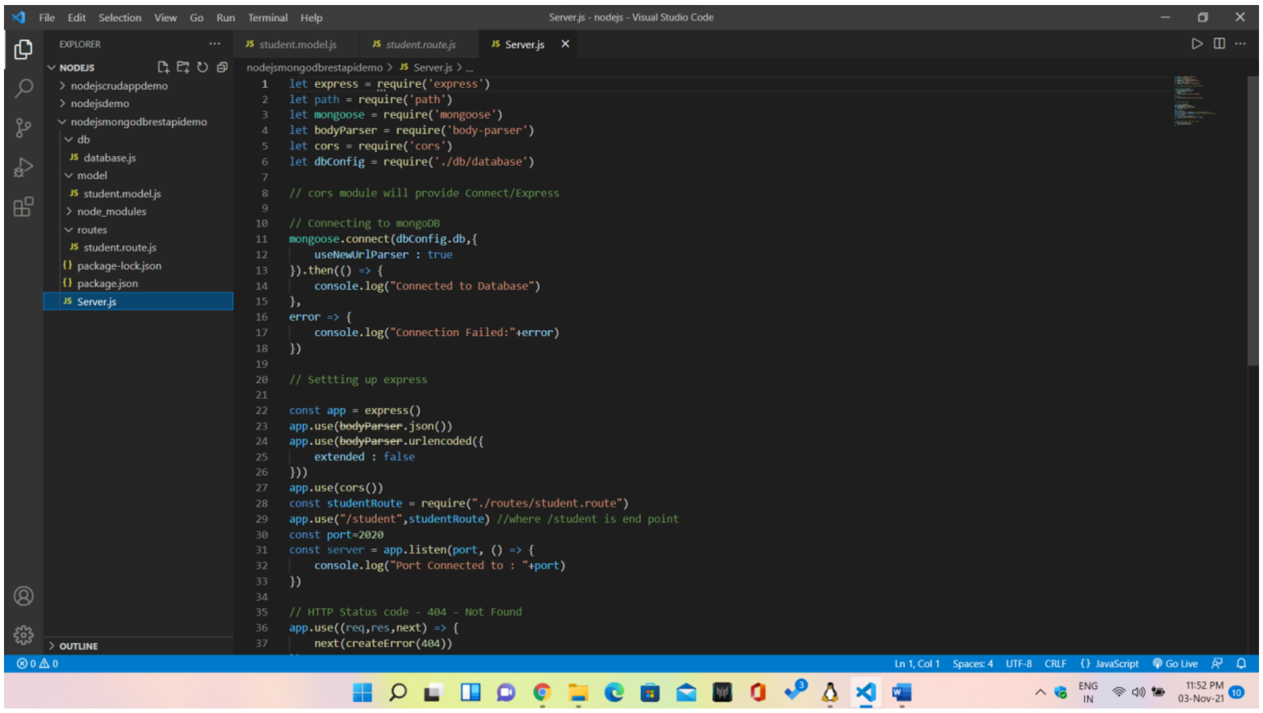


```
        }
        else
        {
            res.json(data)
        }
    })
})

//Displaying a Specific Student Record by ID

studentExpressRoute.route("/getstudent/:id").get((req,res) =>
{
    StudentSchema.findById(req.params.id,(error,data) =>
    {
        if(error)
        {
            return next(error)
        }
        else
        {
            res.json(data)
        }
    })
})

//Update Student (Update Operation)

studentExpressRoute.route("/update/:id").put((req,res,next) =>
{
    StudentSchema.findByIdAndUpdate(req.params.id,{
        $set: req.body
    },(error,data)=>
    {
        if(error)
        {
            return next(error)
```



```
        else
        {
            res.json(data)
            console.log("Student Document Updated Successfully")
        }
    })
})

// Deleting Student Document by ID (Delete Operation)

studentExpressRoute.route("/delete/:id").delete((req,res,next) =>
{
    StudentSchema.findByIdAndRemove(req.params.id,(error,data) =>
    {
        if(error)
        {
            return next(error)
        }
        else
        {
            res.status(200).json(
                {
                    msg:data
                })
        }
    })
})
module.exports=studentExpressRoute
```

Server.js:



```
let express = require('express')
let path = require('path')
let mongoose = require('mongoose')
let bodyParser = require('body-parser')
let cors = require('cors')
let dbConfig = require('./db/database')

// cors module will provide Connect/Express

// Connecting to mongoDB
mongoose.connect(dbConfig.db,{
    useNewUrlParser : true
}).then(() => {
    console.log("Connected to Database")
},
error => {
    console.log("Connection Failed:"+error)
})

// Settting up express

const app = express()
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({
    extended : false
}))
app.use(cors())
const studentRoute = require("./routes/student.route")
app.use("/student",studentRoute) //where /student is end point
const port=2020
const server = app.listen(port, () => {
    console.log("Port Connected to : "+port)
})

// HTTP Status code - 404 - Not Found
app.use((req,res,next) => {
    next(createError(404))
```