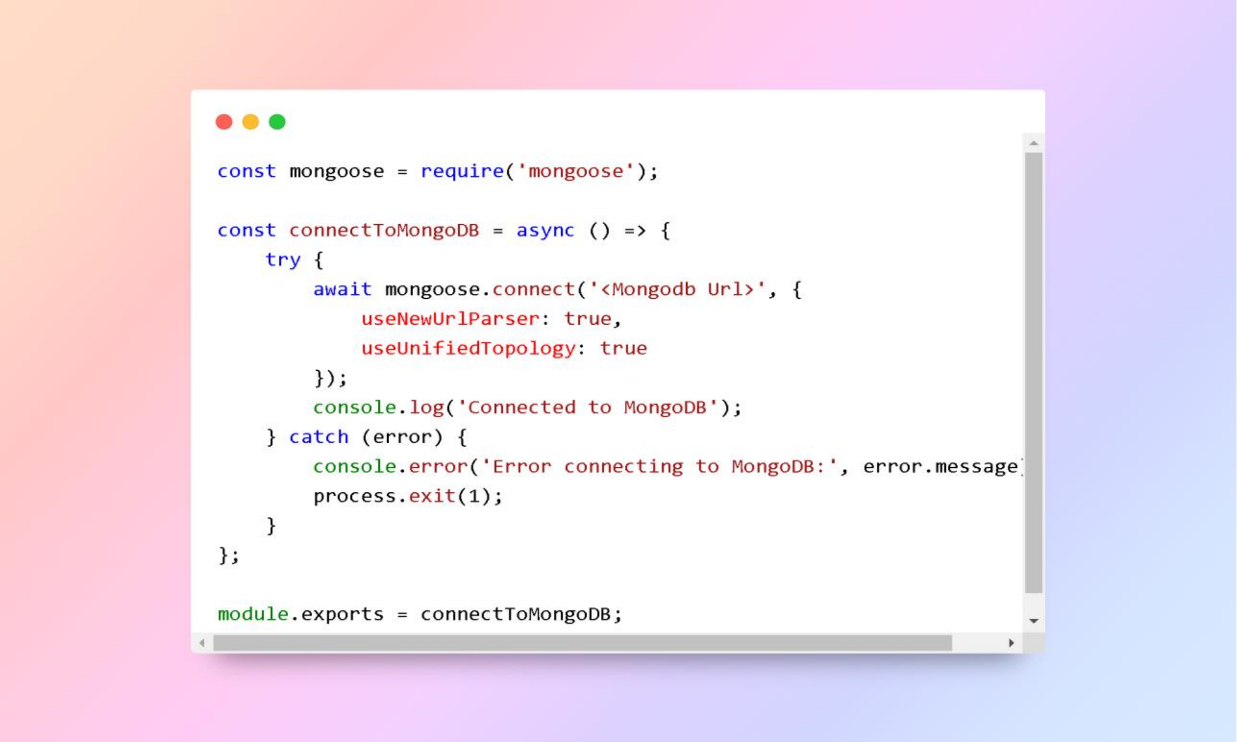
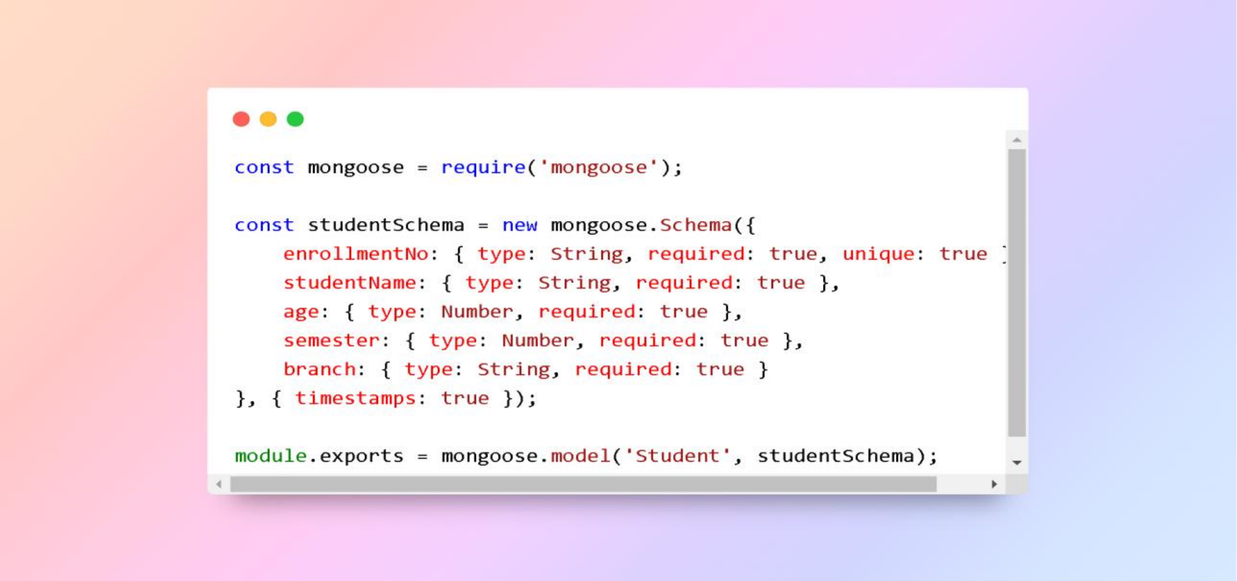
- **Lab-15**
- **1. Create a restful CRUD API using NodeJS, Express and MongoDB for student.**

connectdb.js



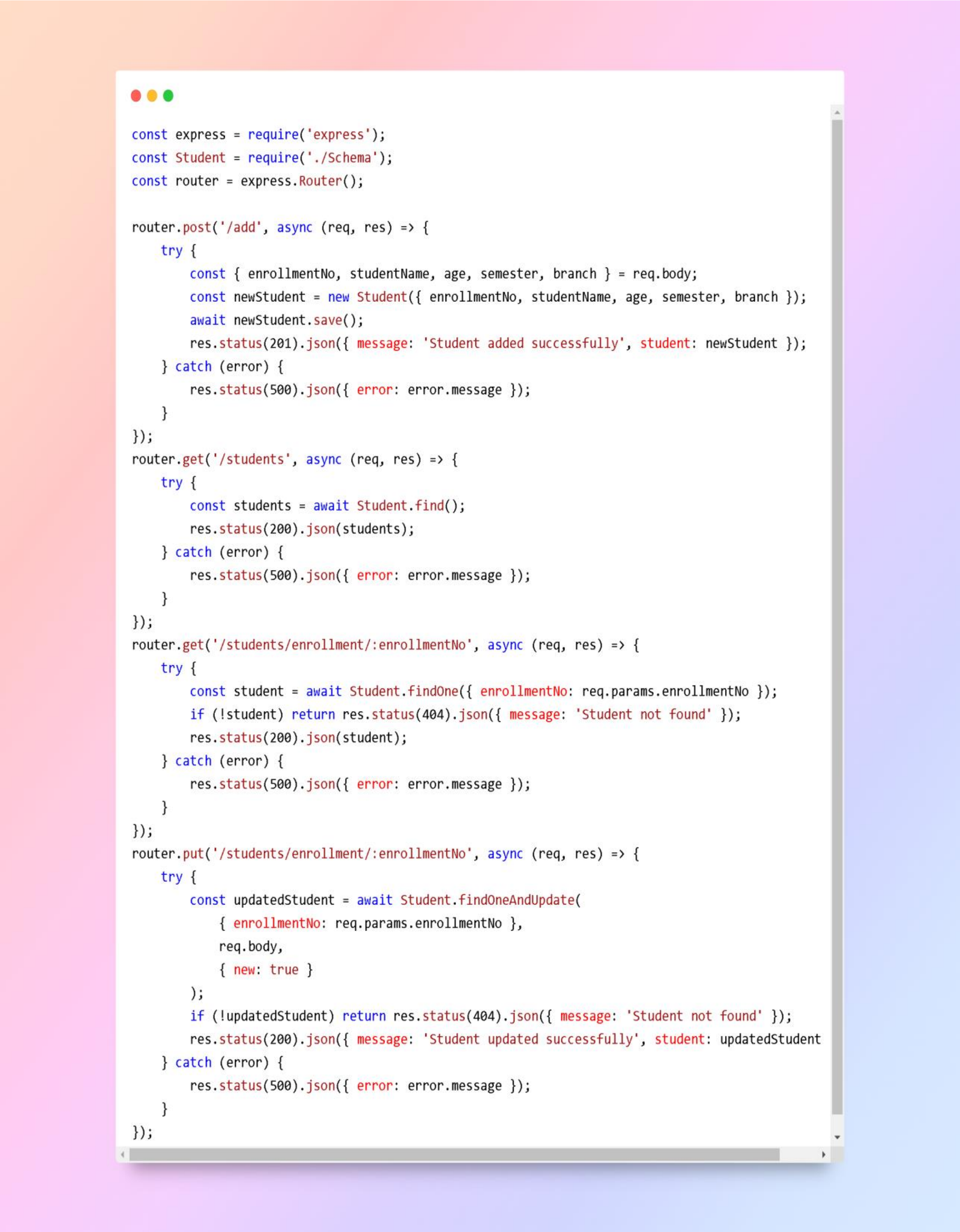
```
const mongoose = require('mongoose');

const connectToMongoDB = async () => {
    try {
        await mongoose.connect('<Mongodb Url>', {
            useNewUrlParser: true,
            useUnifiedTopology: true
        });
        console.log('Connected to MongoDB');
    } catch (error) {
        console.error('Error connecting to MongoDB:', error.message
        process.exit(1);
    }
};

module.exports = connectToMongoDB;
```

Schema.js

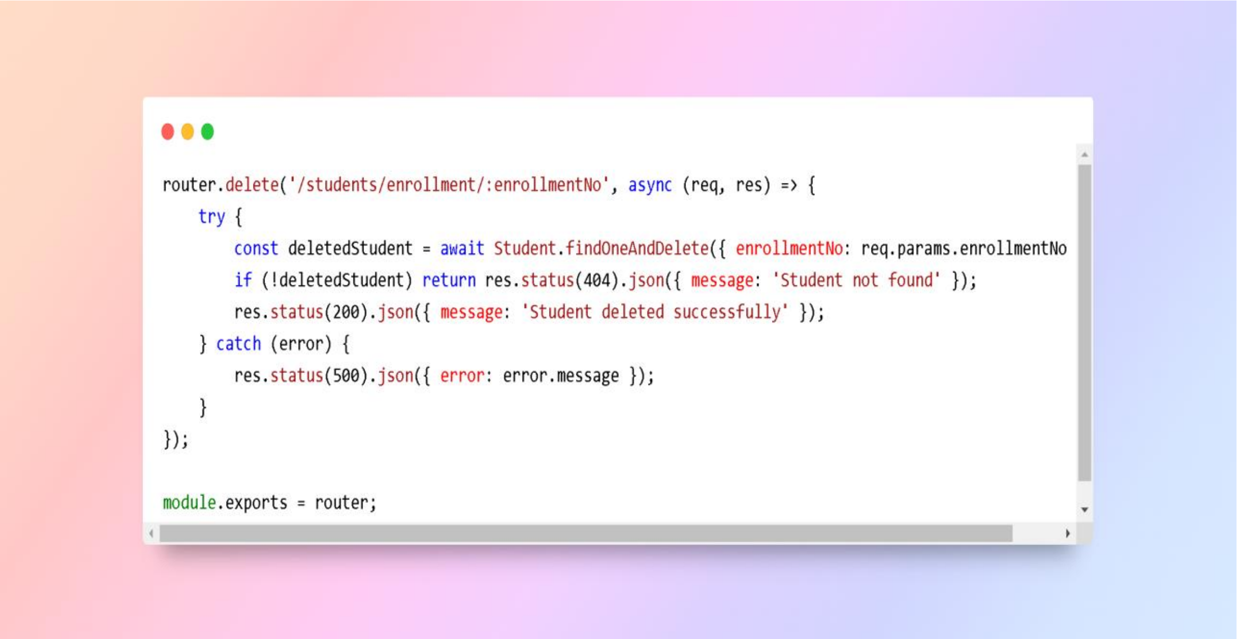


```
const mongoose = require('mongoose');

const studentSchema = new mongoose.Schema({
    enrollmentNo: { type: String, required: true, unique: true
    studentName: { type: String, required: true },
    age: { type: Number, required: true },
    semester: { type: Number, required: true },
    branch: { type: String, required: true }
}, { timestamps: true });

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

routes.js

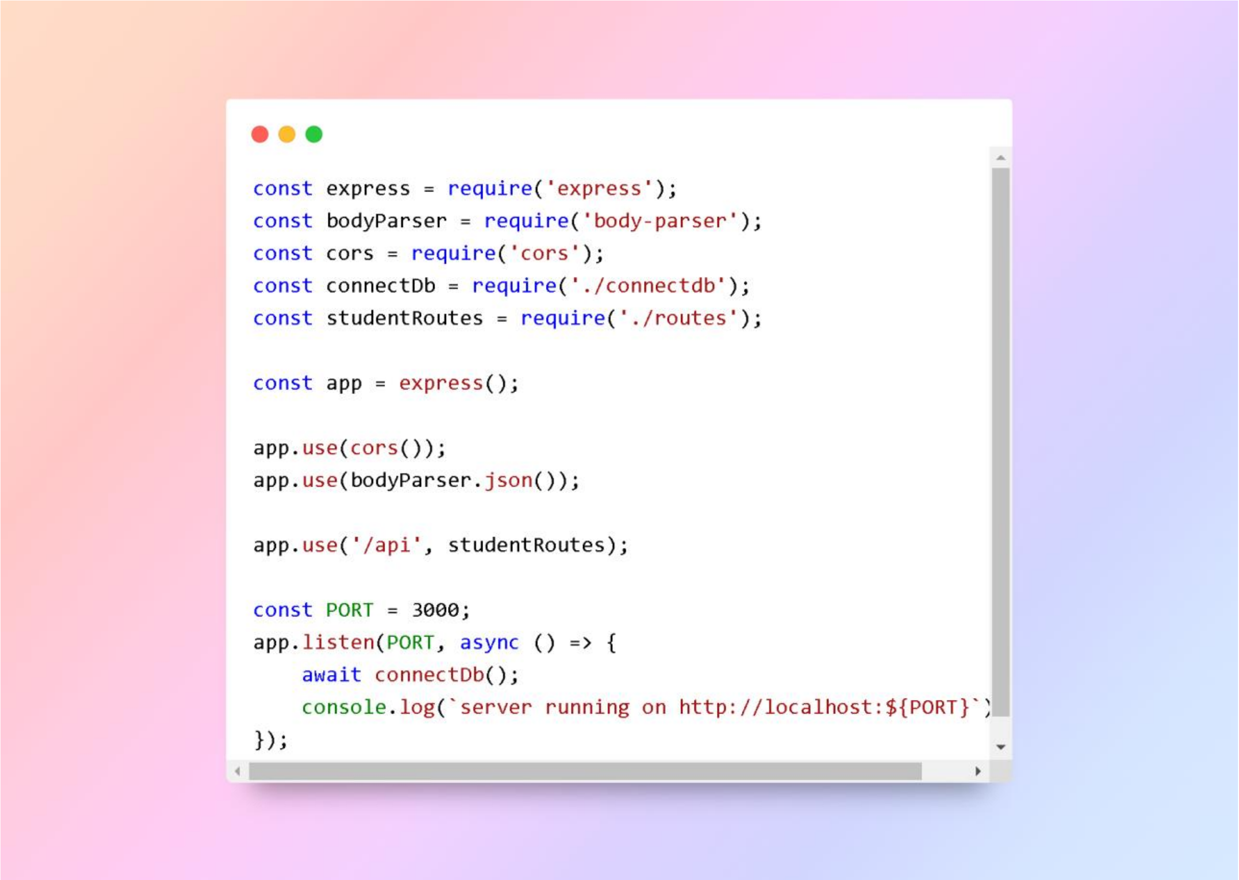


```
const express = require('express');
const Student = require('./Schema');
const router = express.Router();

router.post('/add', async (req, res) => {
    try {
        const { enrollmentNo, studentName, age, semester, branch } = req.body;
        const newStudent = new Student({ enrollmentNo, studentName, age, semester, branch });
        await newStudent.save();
        res.status(201).json({ message: 'Student added successfully', student: newStudent });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});
router.get('/students', async (req, res) => {
    try {
        const students = await Student.find();
        res.status(200).json(students);
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});
router.get('/students/enrollment/:enrollmentNo', async (req, res) => {
    try {
        const student = await Student.findOne({ enrollmentNo: req.params.enrollmentNo });
        if (!student) return res.status(404).json({ message: 'Student not found' });
        res.status(200).json(student);
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});
router.put('/students/enrollment/:enrollmentNo', async (req, res) => {
    try {
        const updatedStudent = await Student.findOneAndUpdate(
            { enrollmentNo: req.params.enrollmentNo },
            req.body,
            { new: true }
        );
        if (!updatedStudent) return res.status(404).json({ message: 'Student not found' });
        res.status(200).json({ message: 'Student updated successfully', student: updatedStudent
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});
```



```
router.delete('/students/enrollment/:enrollmentNo', async (req, res) => {
    try {
        const deletedStudent = await Student.findOneAndDelete({ enrollmentNo: req.params.enrollmentNo
        if (!deletedStudent) return res.status(404).json({ message: 'Student not found' });
        res.status(200).json({ message: 'Student deleted successfully' });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});

module.exports = router;
```

Index.js



```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');
const connectDb = require('./connectdb');
const studentRoutes = require('./routes');

const app = express();

app.use(cors());
app.use(bodyParser.json());

app.use('/api', studentRoutes);

const PORT = 3000;
app.listen(PORT, async () => {
    await connectDb();
    console.log(`server running on http://localhost:${PORT}`)
});
```