

Models\student.js

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

const studentSchema = new mongoose.Schema({
  registrationNumber: {
    type: String,
    required: true,
    unique: true
  },
  name: {
    type: String,
    required: true
  },
  age: Number,
  email: String,
  department: String,
});

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

Routes\student.js

```
// routes/student.js

const express = require('express');
const router = express.Router();
const Student = require('../models/student');

// Get student details by ID
router.get('/students/:id', async (req, res) => {
  try {
    const student = await Student.findById(req.params.id);
    if (!student) {
      return res.status(404).json({ message: 'Student not found' });
    }
  }
});
```

```
        res.json(student);
    } catch (error) {
        res.status(500).json({ message: error.message });
    }
});

// Create a new student
router.post('/students', async (req, res) => {
    const { registrationNumber, name, age, email, department } = req.body;

    if (!registrationNumber || !name) {
        return res.status(400).json({ message: 'Registration number and name are required' });
    }

    const student = new Student({
        registrationNumber,
        name,
        age,
        email,
        department,
    });

    try {
        const savedStudent = await student.save();
        res.status(201).json(savedStudent);
    } catch (error) {
        res.status(500).json({ message: error.message });
    }
});

module.exports = router;
```

Server.js

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

dotenv.config();

const app = express();
const port = process.env.PORT || 3000;

// Connect to MongoDB
mongoose.connect(process.env.MONGODB_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
  .then(() => console.log('Connected to MongoDB'))
  .catch(err => console.error('Failed to connect to MongoDB', err));

// Middleware
app.use(express.json());

// Routes
const studentRoutes = require('./routes/student');
app.use('/api', studentRoutes);

// Start the server
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

OUTPUT:-

The screenshot displays a REST client interface with a GET request to `http://localhost:3000/api/students/66c87af14fb1c52551d56cda`. The request body is raw JSON. The response is shown in two panels, both displaying the same JSON object in a pretty-printed format.

Request:

```
1 {
2   ... "registrationNumber": "2021bcs107",
3   ... "name": "prerna raut",
4   ... "age": 21,
5   ... "email": "prernaraut@gmail.com",
6   ... "department": "cse"
7   ...
8 }
```

Response (Top Panel):

```
1 {
2   "_id": "66c87af14fb1c52551d56cda",
3   "registrationNumber": "2021bcs107",
4   "name": "prerna",
5   "age": 21,
6   "email": "prernaraut@gmail.com",
7   "department": "cse",
8   "__v": 0
9 }
```

Response (Bottom Panel):

```
1 {
2   "_id": "66c87af14fb1c52551d56cda",
3   "registrationNumber": "2021bcs107",
4   "name": "prerna",
5   "age": 21,
6   "email": "prernaraut@gmail.com",
7   "department": "cse",
8   "__v": 0
9 }
```