**Day 15 – Mongoose Validation and Schema Rules**

**Why validation is important:**

Validation helps to ensure that the data entered is correct, complete, and safe before saving to the database.

**Mongoose Schema Validation Rules:**

required: field must be filled

minlength: sets minimum characters

unique: prevents duplicate dat

type: ensures correct data type

**How .validate() works in Mongoose:**

When we use .save() or .create(), Mongoose automatically calls .validate() to check schema rules. If the data is invalid, it stops the save and shows an error.

**Code:**

**Server.js:**

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const Student = require('./models/Student');

const app = express();

app.use(bodyParser.json());

mongoose.connect('mongodb://127.0.0.1:27017/webteamdb', {

useNewUrlParser: true,

useUnifiedTopology: true

}).then(() => console.log("MongoDB connected"))

.catch(err => console.log(err));

app.post('/add-student', async (req, res) => {

try {

const student = new Student(req.body);

await student.save();

res.status(201).json({ message: "Student added successfully!", student });

} catch (err) {

res.status(400).json({ error: err.message });

}

});

app.listen(3000, () => {

console.log("Server running on http://localhost:3000");

});

**Models\ Student.js:**

const mongoose = require('mongoose');

const studentSchema = new mongoose.Schema({

name: {

type: String,

required: [true, "Name is required"],

minlength: [3, "Name must be at least 3 characters long"]

},

email: {

type: String,

required: [true, "Email is required"],

unique: true

},

course: {

type: String,

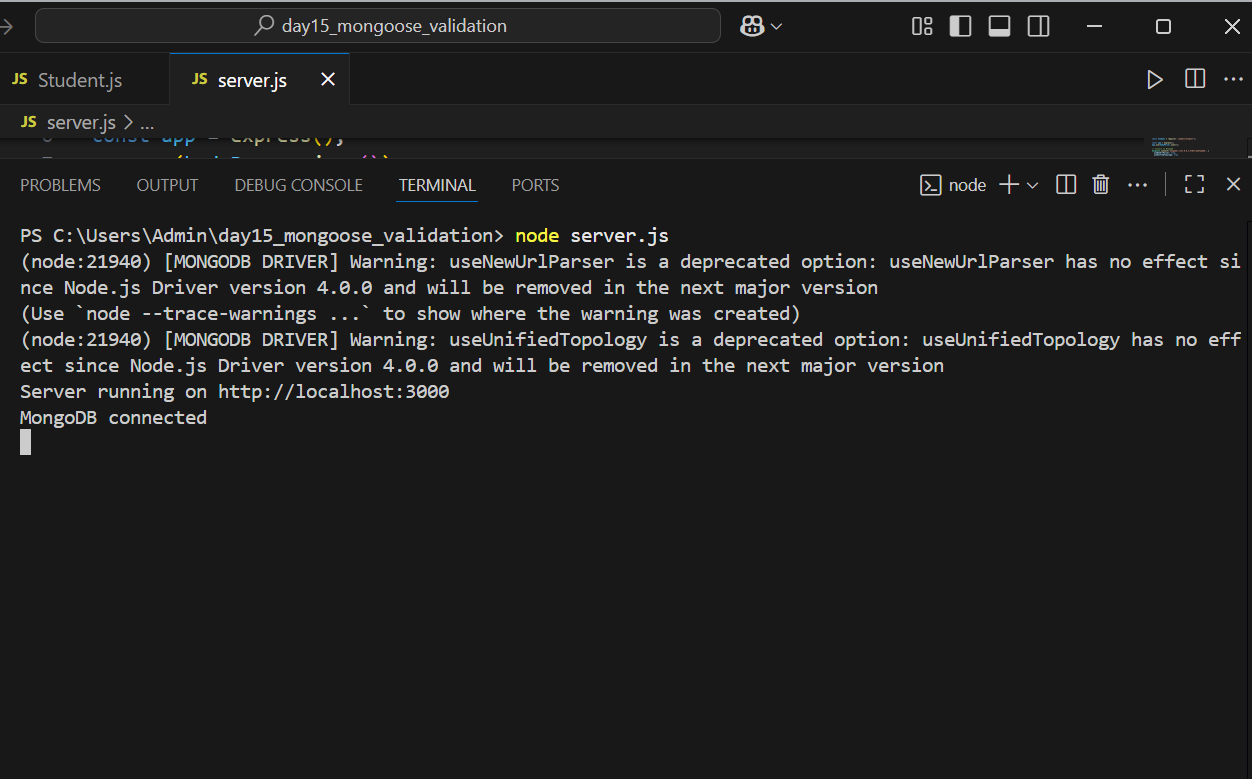
required: [true, "Course is required"]

}

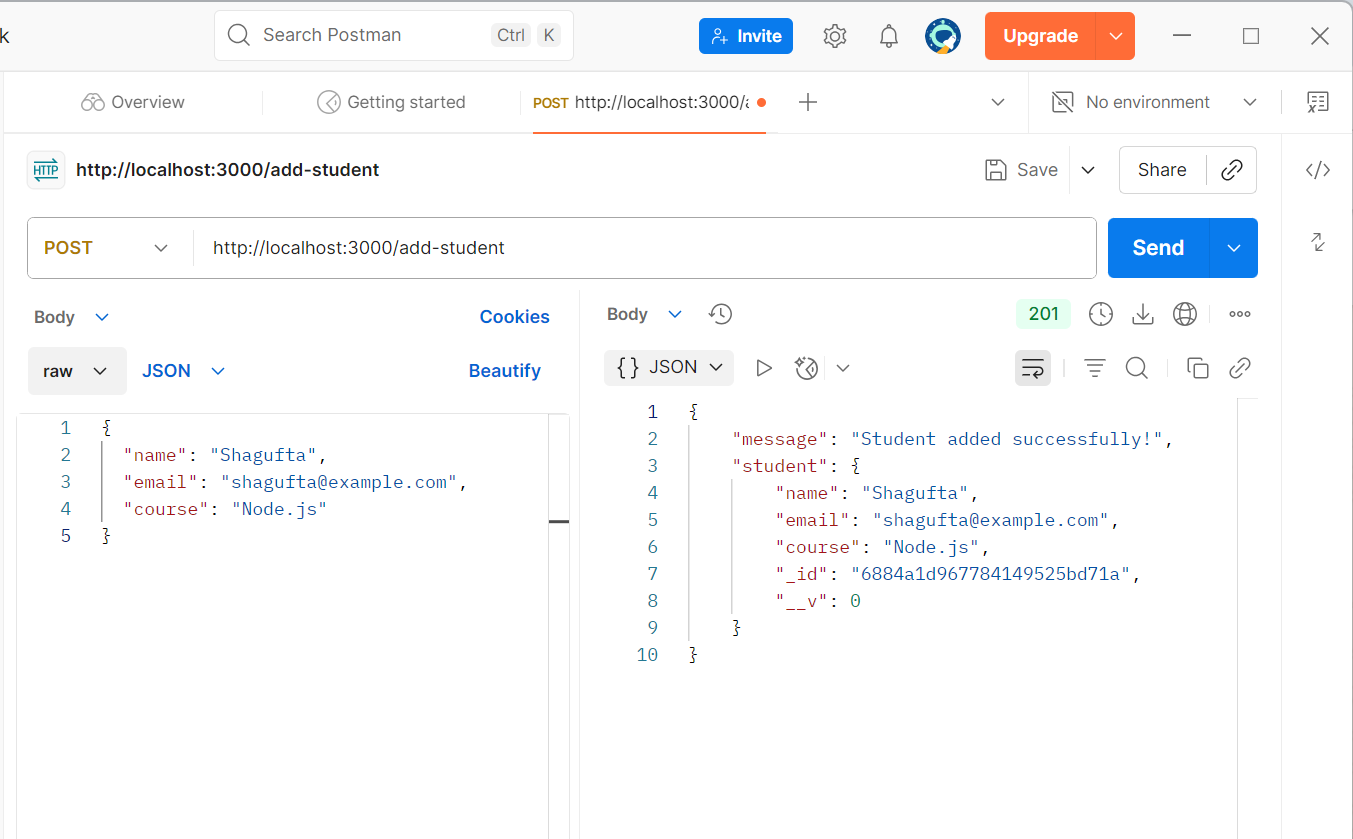
});

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

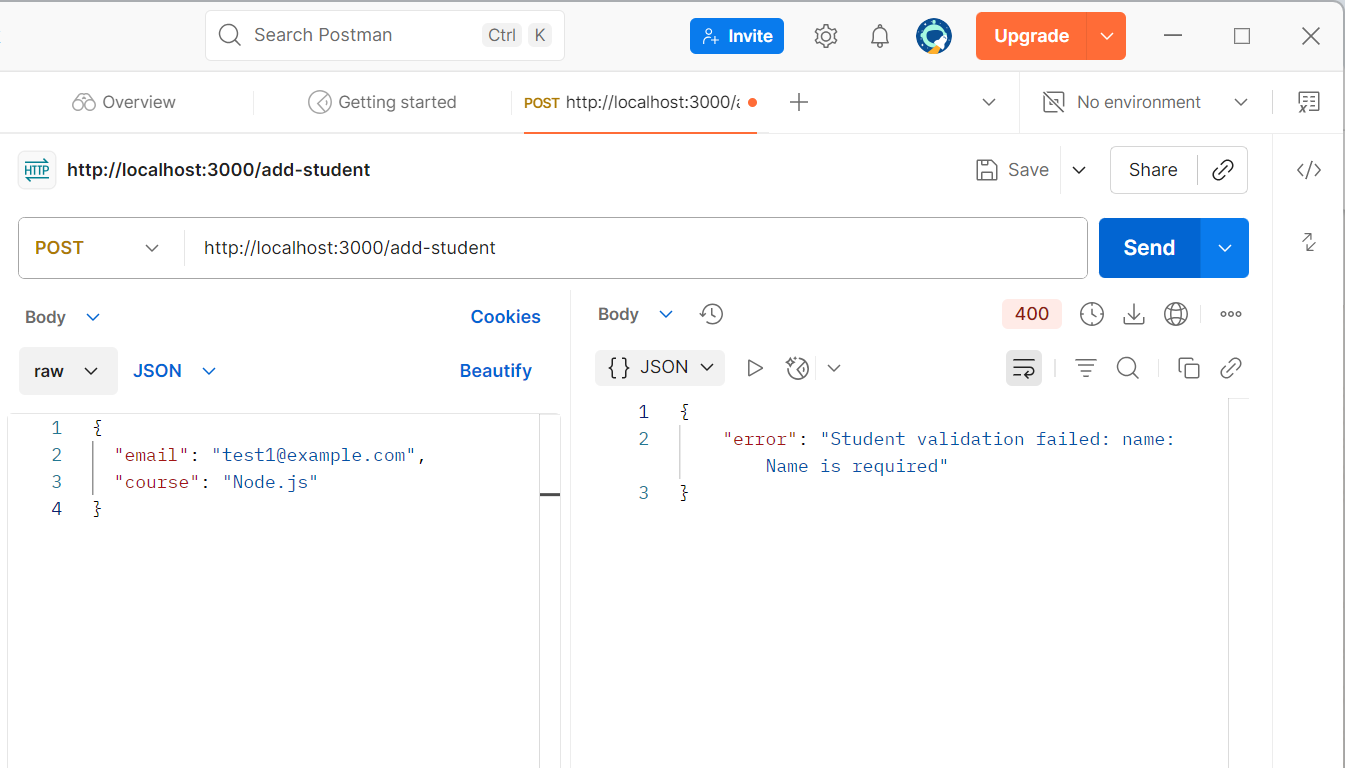
**Output:**

****

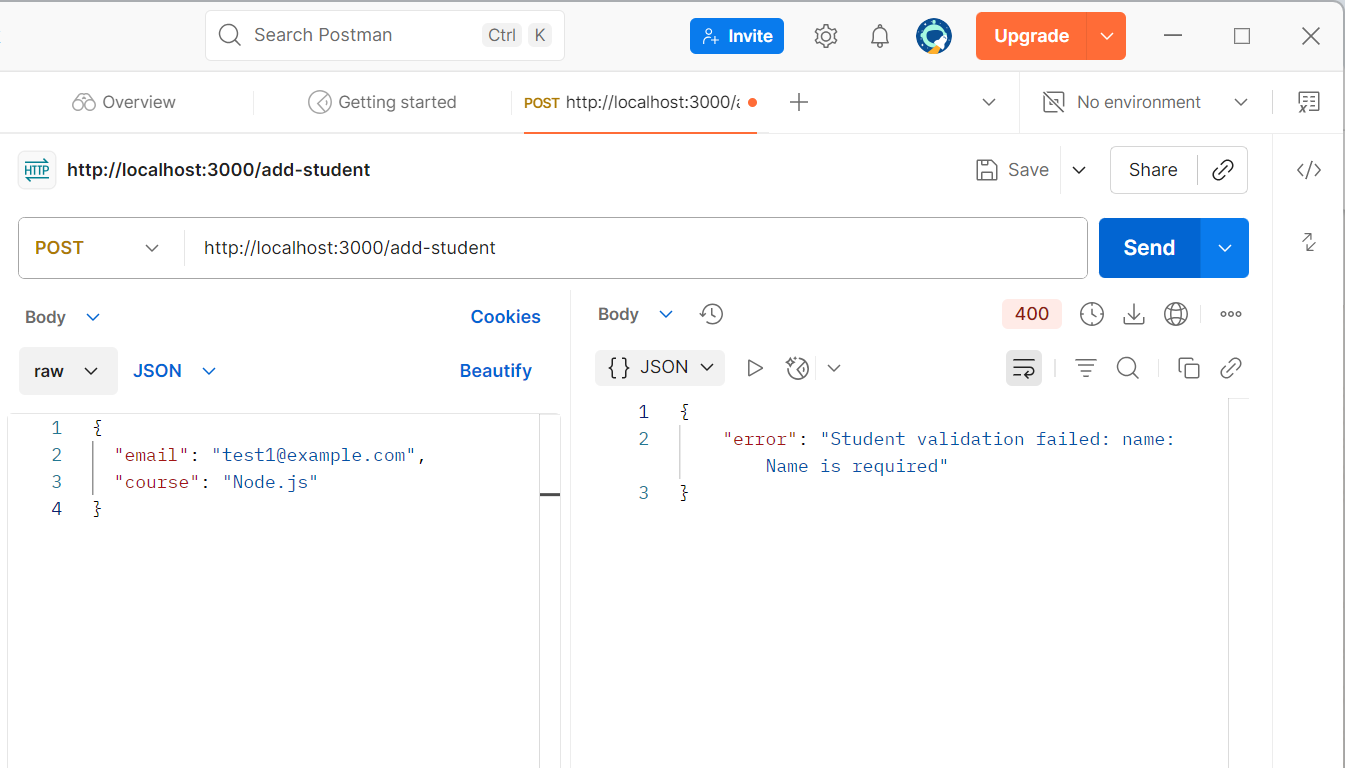
**Test in Postman:**

****

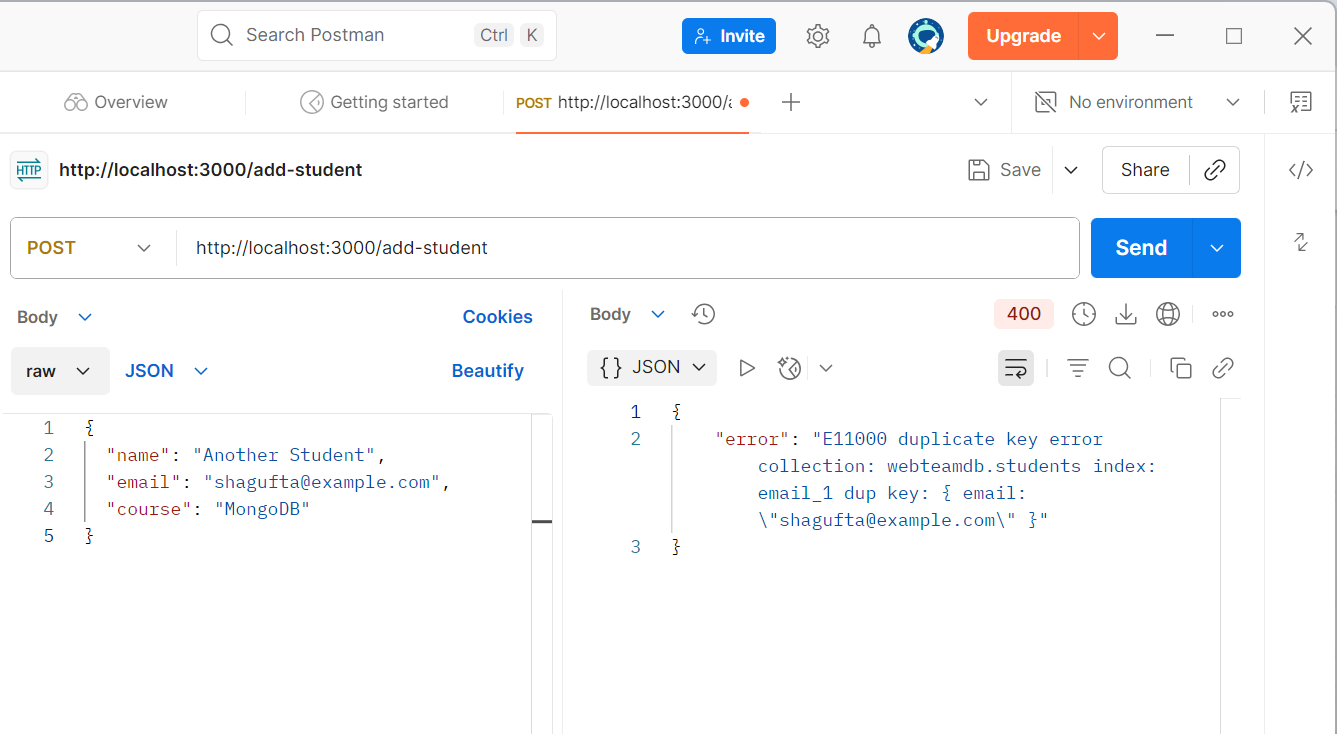
**Test1:Valid Student:**

****

**Test2: Missing name:**

****

**Test3: Duplicate Email:**

****