Lab Assignment-8

Simple API using Express Js, Node Js, MongoDB and Postman

1) Index.js (server):
- Connecting to mongo dB, mongoose and express

```javascript
const express=require("express");
const mongoose=require("mongoose");
const url='mongodb://127.0.0.1:27017/studentDB';


const app=express();

mongoose.connect(url,{
    useNewUrlParser:true
})

const con =mongoose.connection

con.on('open',function(){
    console.log("connected to mongodb database")
})

app.use(express.json())

const studentRouter=require('./routes/students')
app.use('/students',studentRouter)

app.listen(3000,function(){
    console.log("Server started")
})
```

2) students.js
- Creating routes (GET, PATCH, GET single object by ID, POST)

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

router.get('/',async(req,res)=>{
    try{
        const  stud=await Student.find()
        res.json(stud)
```

```javascript
    }catch(err){
        res.send("Error")
    }
    res.send("Get request made")
})

router.get('/:id',async(req,res)=>{
    try{
        const stud1=await Student.findById(req.params.id)
        res.json(stud1)
    }catch(err){
        res.send("Error")
    }
})
router.patch('/:id', async (req, res) => {
    try {
        const studPatch = await Student.findById(req.params.id);
        studPatch.name = req.body.name;
        const s = await studPatch.save();
        res.json(studPatch);
    } catch (err) {
        res.status(500).send("Error"); // Sending an error response
    }
});

router.post('/',async(req,res)=>{
    const student=new Student({
        name: req.body.name,
        course: req.body.course
    })
    try{
        const s=await student.save()
        res.json(s)
    }catch(err){
        res.send("Error")
    }
})


module.exports=router
```

3) student.js

- Creating mongoose schema for a single object (student here)

```javascript
const mongoose=require("mongoose")

const studSchema=new mongoose.Schema({
    name:{
        type: String,
        required:true
    },
    course:{
        type: String,
        required:true
    }
})

module.exports=mongoose.model('Student',studSchema)
```
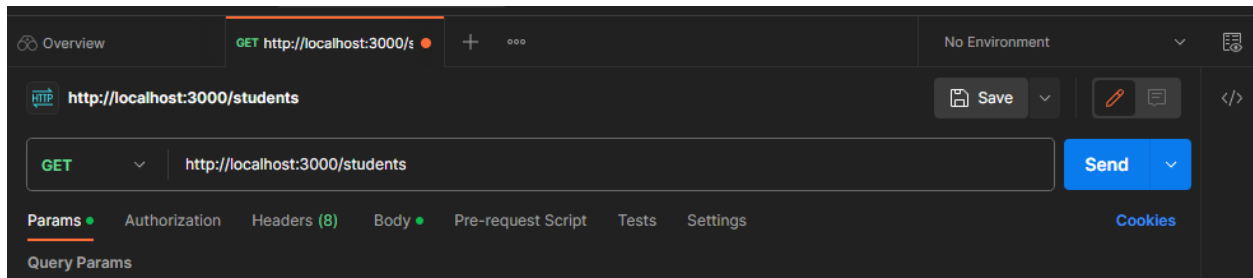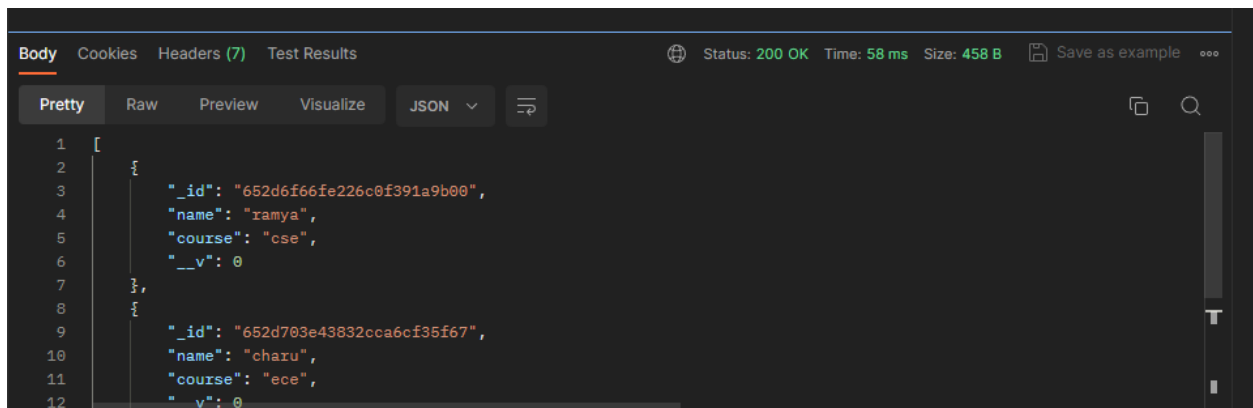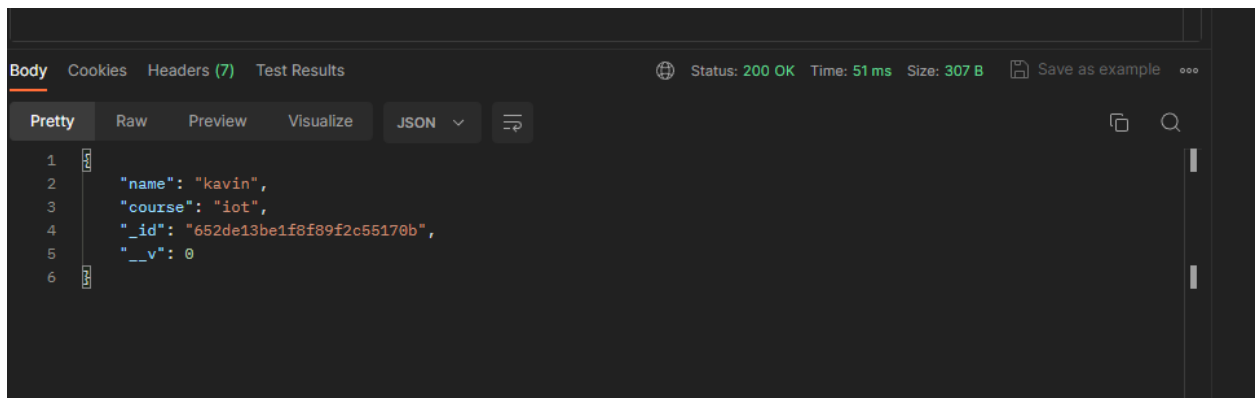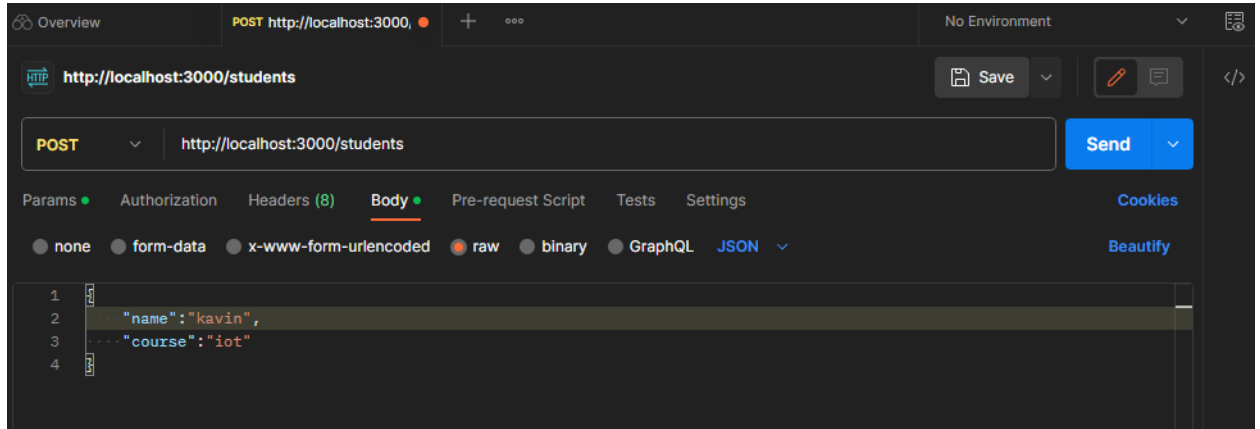
Output:
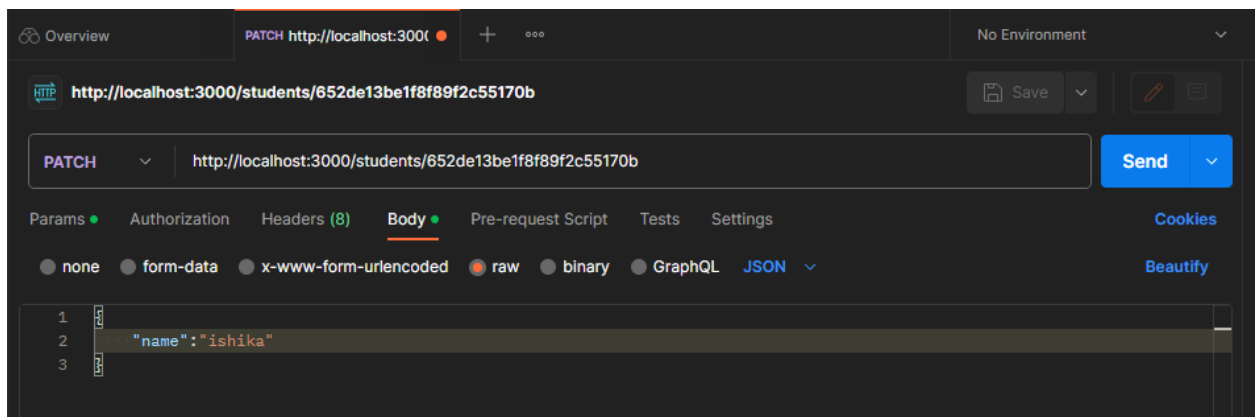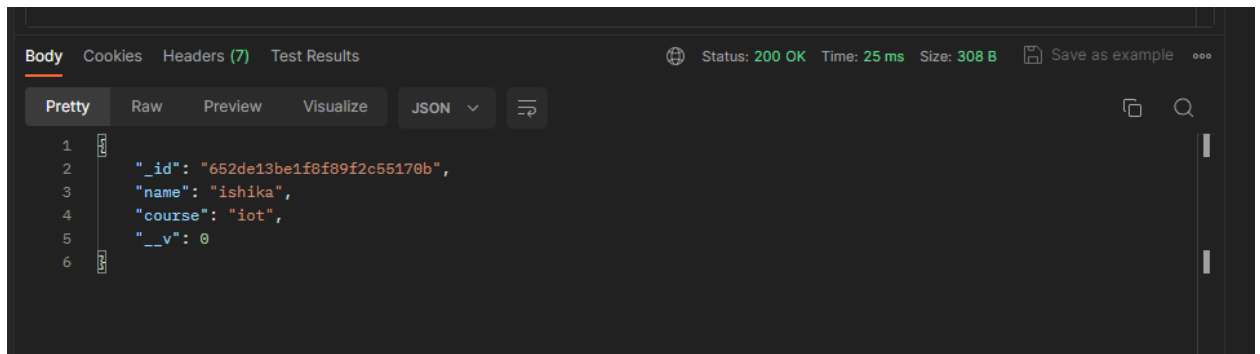
Requests made from Postman API

1) Get request

o/p:

2) Post request

3) Patch request

Body Cookies Headers (7) Test Results   Status: 200 OK Time: 25 ms Size: 308 B   Save as example ooo

Pretty Raw Preview Visualize   JSON ∨

1 {
2   "_id": "652de13be1f8f89f2c55170b",
3   "name": "ishika",
4   "course": "iot",
5   "__v": 0
6 }

4) Get by Id

Overview   GET http://localhost:3000/s ●   + ooo   No Environment ∨

HTTP http://localhost:3000/students/652d6f66fe226c0f391a9b00   Save ∨

GET ∨   http://localhost:3000/students/652d6f66fe226c0f391a9b00   Send ∨

Params ● Authorization Headers (8) Body ● Pre-request Script Tests Settings   Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON ∨   Beautify

Body Cookies Headers (7) Test Results   Status: 200 OK Time: 39 ms Size: 307 B   Save as example ooo

Pretty Raw Preview Visualize   JSON ∨

1 {
2   "_id": "652d6f66fe226c0f391a9b00",
3   "name": "ramya",
4   "course": "cse",
5   "__v": 0
6 }