Lab Report SE

CS-165L

Submitted To:

Sir Laeeq Khan Niazi

Submitted By:

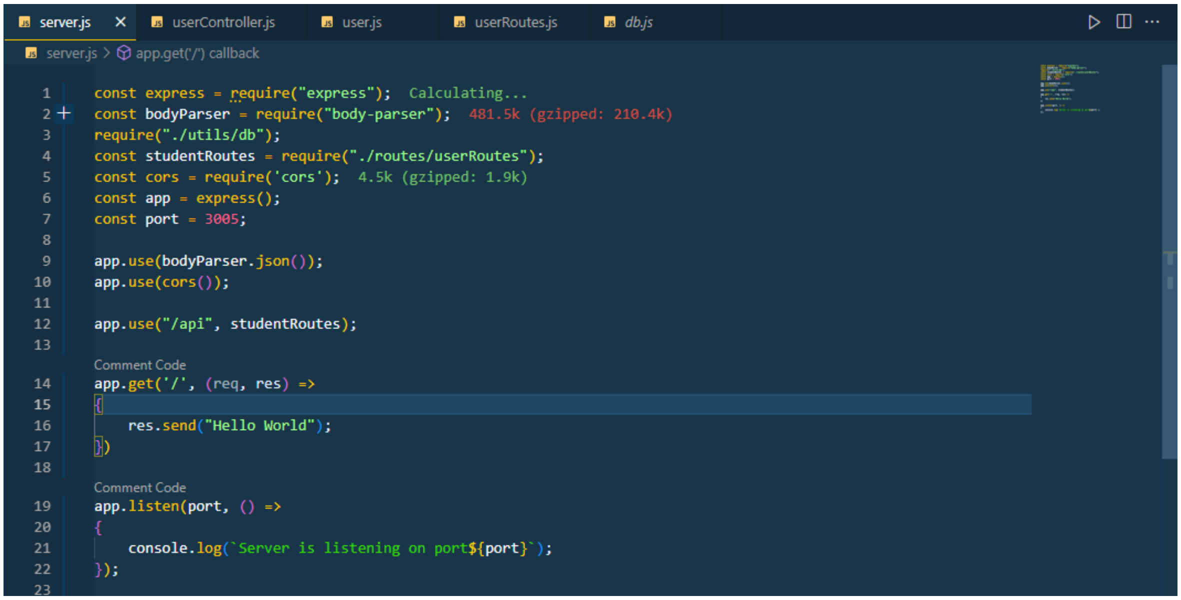
Muhammad Moazam 2021-CS-53

` University of Engineering Technology

Lahore

1. Complete Code

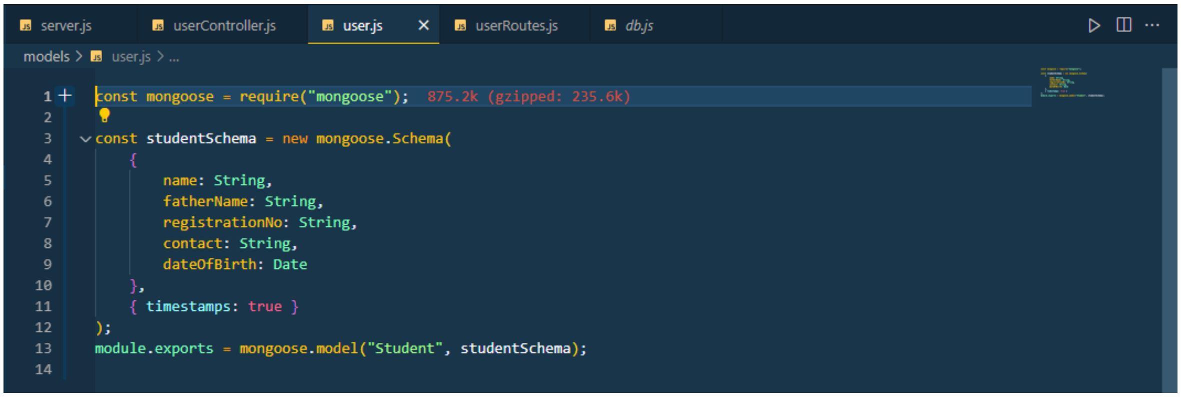
1.1 Server



```
const express = require("express");  Calculating...
const bodyParser = require("body-parser");  481.5k (gzipped: 210.4k)
require("./utils/db");
const studentRoutes = require("./routes/userRoutes");
const cors = require('cors');  4.5k (gzipped: 1.9k)
const app = express();
const port = 3005;

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

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

Comment Code
app.get('/', (req, res) =>
{
    res.send("Hello World");
})

Comment Code
app.listen(port, () =>
{
    console.log(`Server is listening on port${port}`);
});

```

2. Models

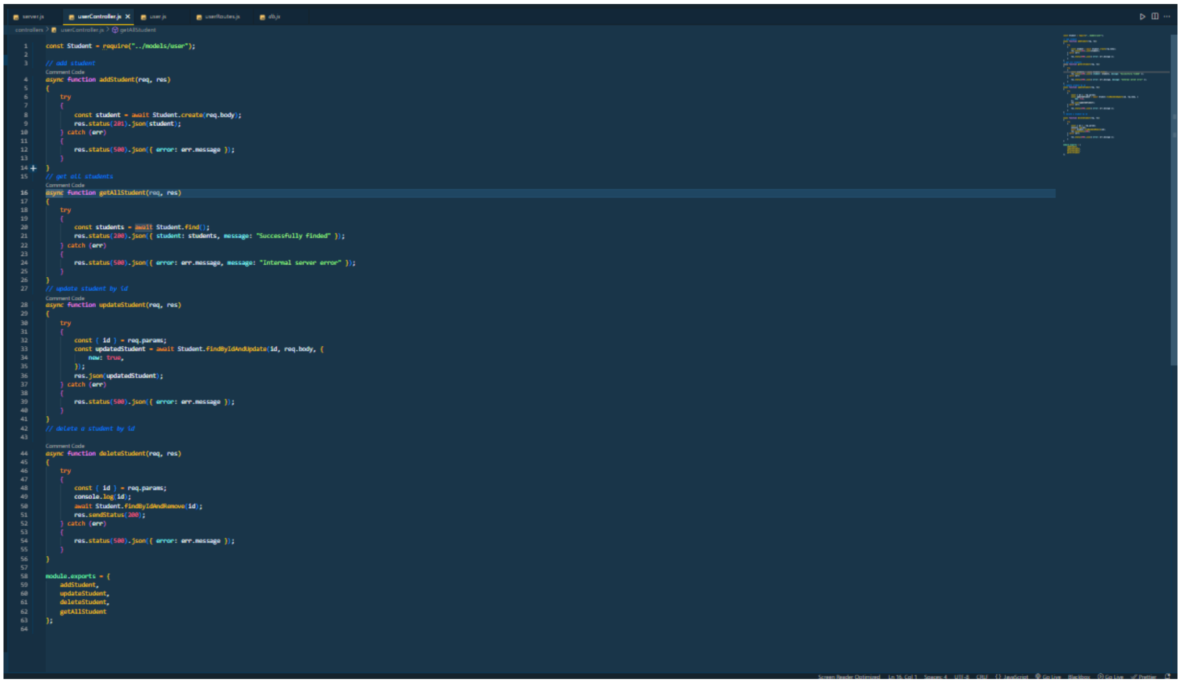
2.1 User Model



```
const mongoose = require("mongoose");  875.2k (gzipped: 235.6k)

const studentSchema = new mongoose.Schema(
    {
        name: String,
        fatherName: String,
        registrationNo: String,
        contact: String,
        dateOfBirth: Date
    },
    { timestamps: true }
);
module.exports = mongoose.model("Student", studentSchema);

```

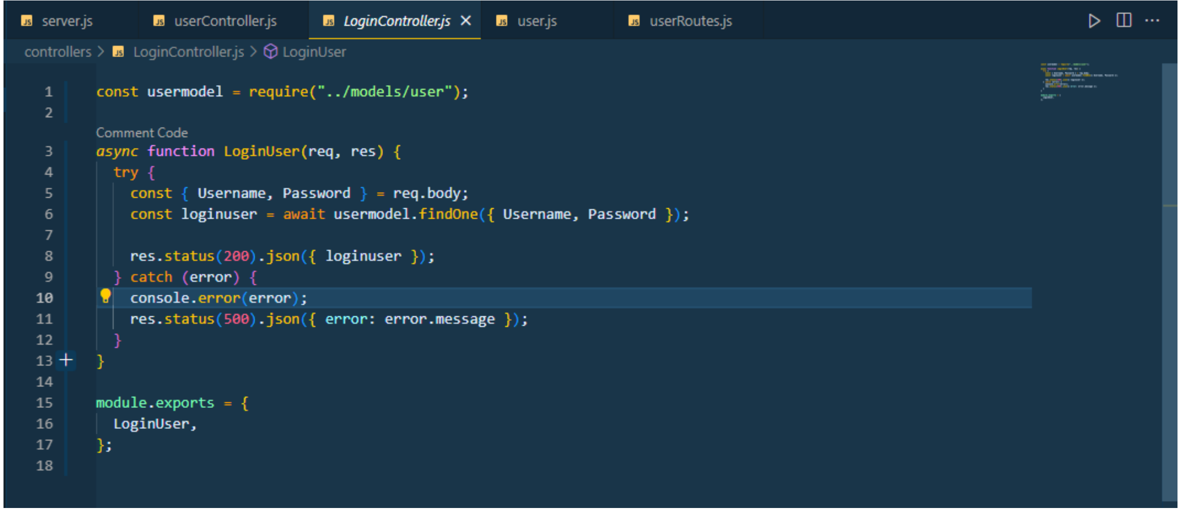
3. Controllers

3.1 UserController



```
const Student = require("../models/user");

// add student
async function addStudent(req, res)
{
    try
    {
        const student = await Student.create(req.body);
        res.status(201).json(student);
    } catch (err)
    {
        res.status(500).json({ error: err.message });
    }
}
// get all students
async function getAllStudent(req, res)
{
    try
    {
        const students = await Student.find();
        res.status(200).json({ student: students, message: "Successfully finded" });
    } catch (err)
    {
        res.status(500).json({ error: err.message, message: "Internal server error" });
    }
}
// update student by id
async function updateStudent(req, res)
{
    try
    {
        const { id } = req.params;
        const updatedStudent = await Student.findByIdAndUpdate(id, req.body, {
            new: true,
        });
        res.json(updatedStudent);
    } catch (err)
    {
        res.status(500).json({ error: err.message });
    }
}
// delete a student by id

async function deleteStudent(req, res)
{
    try
    {
        const { id } = req.params;
        console.log(id);
        await Student.findByIdAndRemove(id);
        res.sendStatus(200);
    } catch (err)
    {
        res.status(500).json({ error: err.message });
    }
}

module.exports = {
    addStudent,
    updateStudent,
    deleteStudent,
    getAllStudent
};
```

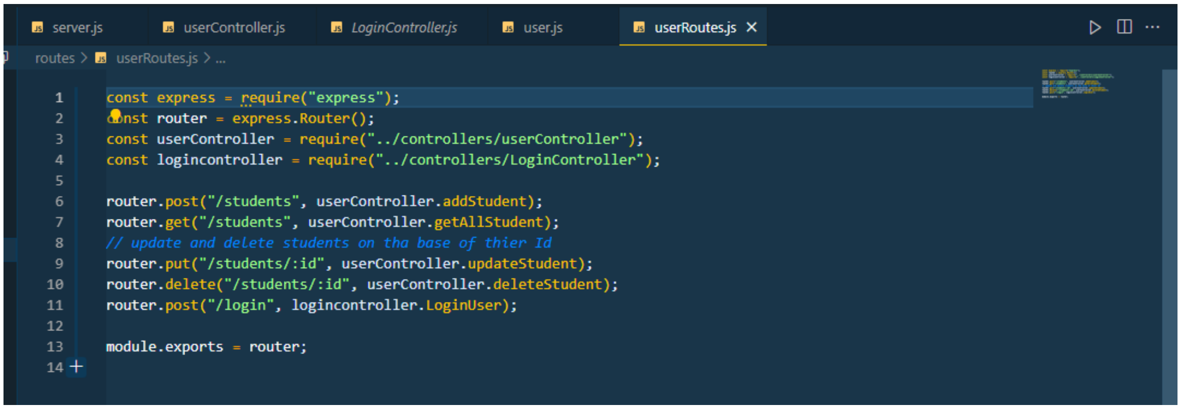
3.2 Login Controller



```
const usermodel = require("../models/user");

async function LoginUser(req, res) {
  try {
    const { Username, Password } = req.body;
    const loginuser = await usermodel.findOne({ Username, Password });

    res.status(200).json({ loginuser });
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: error.message });
  }
}

module.exports = {
  LoginUser,
};
```

4. Routes

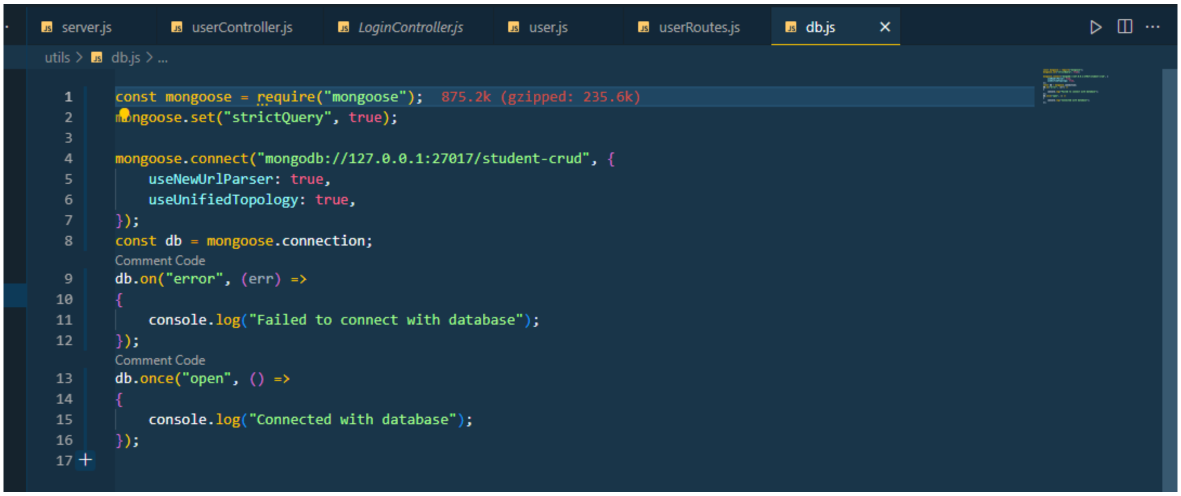
4.1 User Routes



```javascript
const express = require("express");
const router = express.Router();
const userController = require("../controllers/userController");
const logincontroller = require("../controllers/LoginController");

router.post("/students", userController.addStudent);
router.get("/students", userController.getAllStudent);
// update and delete students on tha base of thier Id
router.put("/students/:id", userController.updateStudent);
router.delete("/students/:id", userController.deleteStudent);
router.post("/login", logincontroller.LoginUser);

module.exports = router;
```

5. Utils

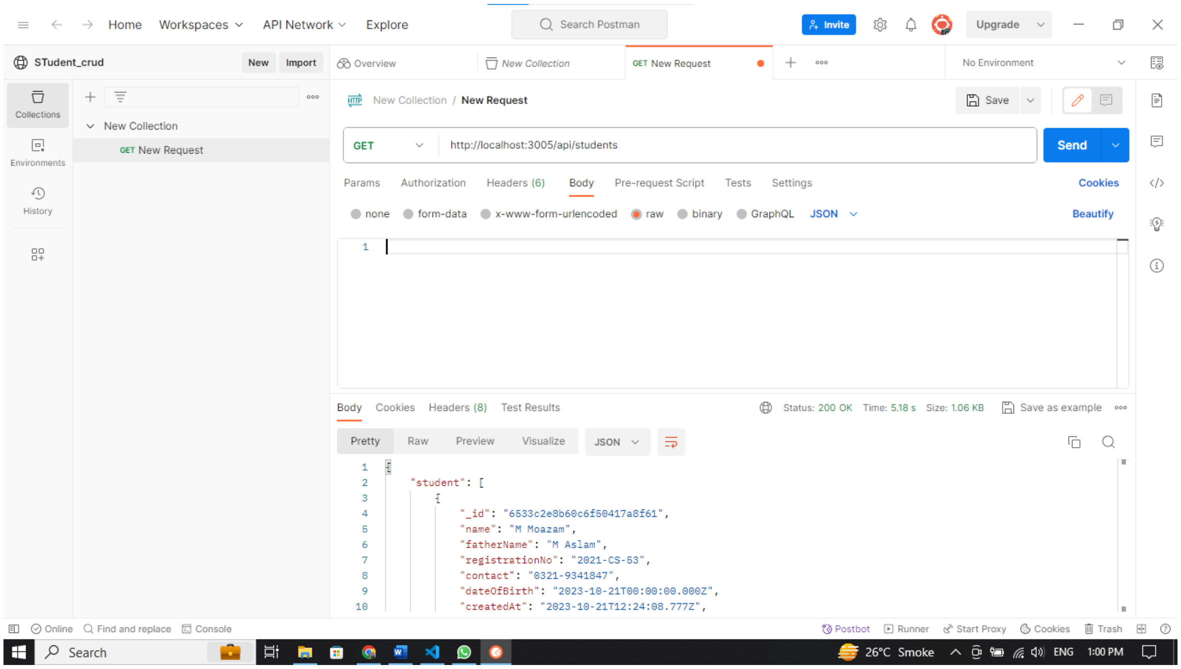
5.1 DB



```javascript
const mongoose = require("mongoose");
mongoose.set("strictQuery", true);

mongoose.connect("mongodb://127.0.0.1:27017/student-crud", {
    useNewUrlParser: true,
    useUnifiedTopology: true,
});
const db = mongoose.connection;
db.on("error", (err) =>
{
    console.log("Failed to connect with database");
});
db.once("open", () =>
{
    console.log("Connected with database");
});
```

6. Testing

6.1 Post API

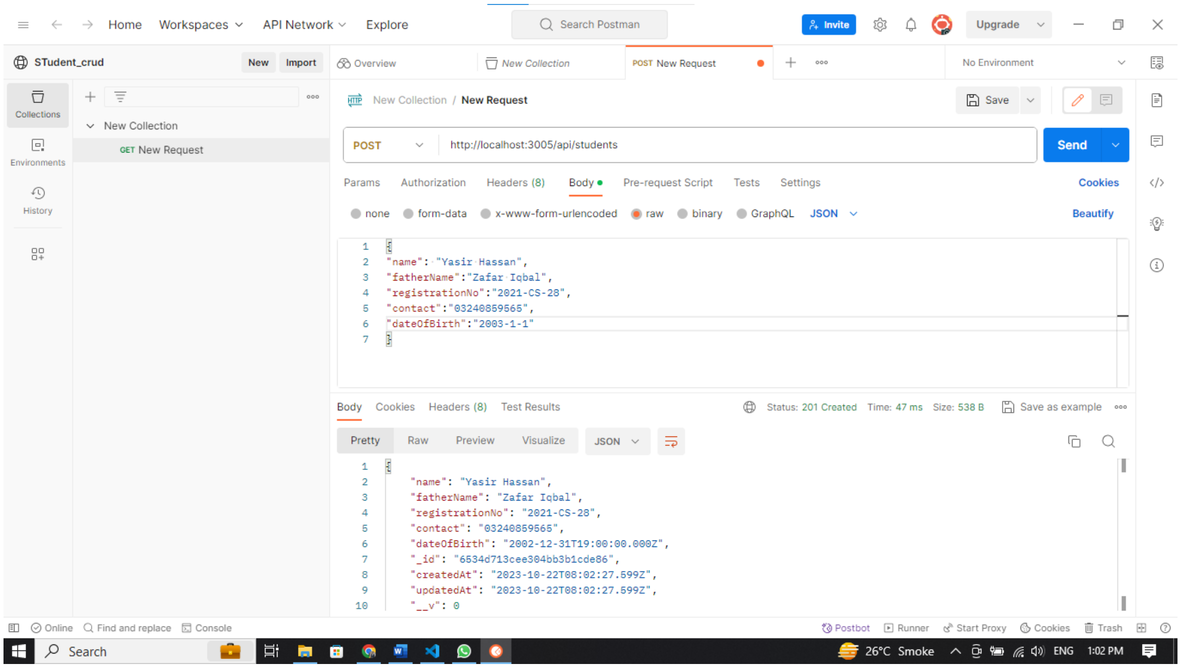


GET http://localhost:3005/api/students

Status: 200 OK Time: 5.18 s Size: 1.06 KB

```
{
    "student": [
        {
            "_id": "6533c2e8b60c6f50417a8f61",
            "name": "M Moazam",
            "fatherName": "M Aslam",
            "registrationNo": "2021-CS-53",
            "contact": "0321-9341847",
            "dateOfBirth": "2023-10-21T00:00:00.000Z",
            "createdAt": "2023-10-21T12:24:08.777Z",
```

6.2 Get API



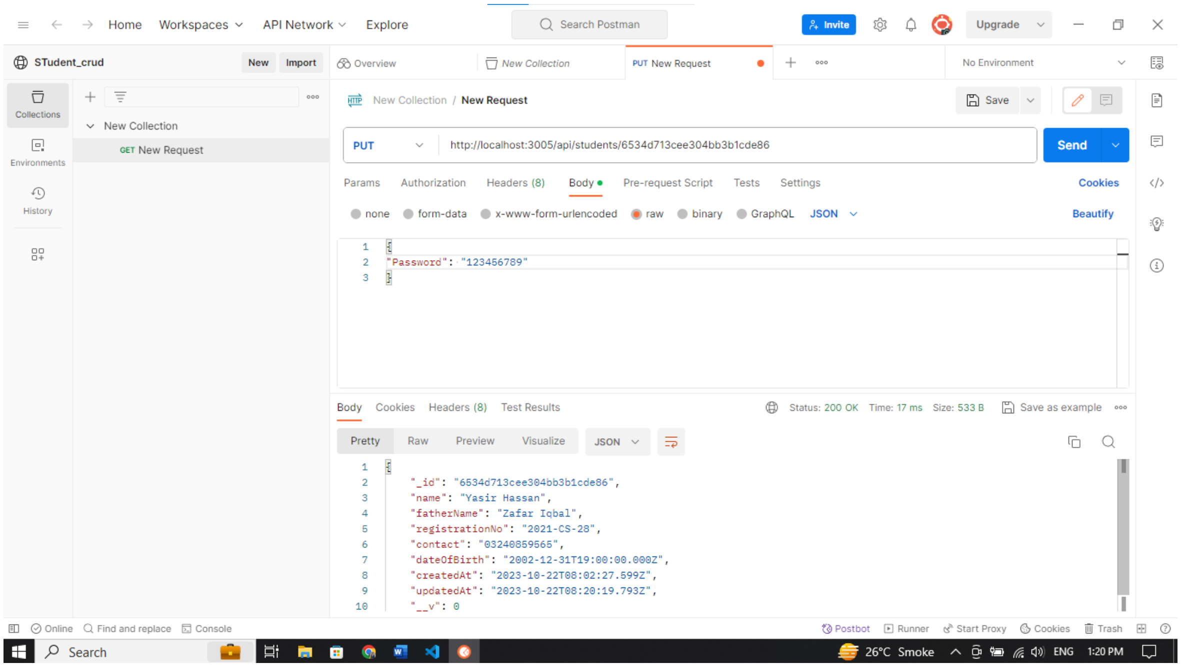
POST http://localhost:3005/api/students

```
{
"name": "Yasir Hassan",
"fatherName":"Zafar Iqbal",
"registrationNo":"2021-CS-28",
"contact":"03240859565",
"dateOfBirth":"2003-1-1"
}
```

Status: 201 Created Time: 47 ms Size: 538 B

```
{
    "name": "Yasir Hassan",
    "fatherName": "Zafar Iqbal",
    "registrationNo": "2021-CS-28",
    "contact": "03240859565",
    "dateOfBirth": "2002-12-31T19:00:00.000Z",
    "_id": "6534d713cee304bb3b1cde86",
    "createdAt": "2023-10-22T08:02:27.599Z",
    "updatedAt": "2023-10-22T08:02:27.599Z",
    "__v": 0
```

6.3 Put API



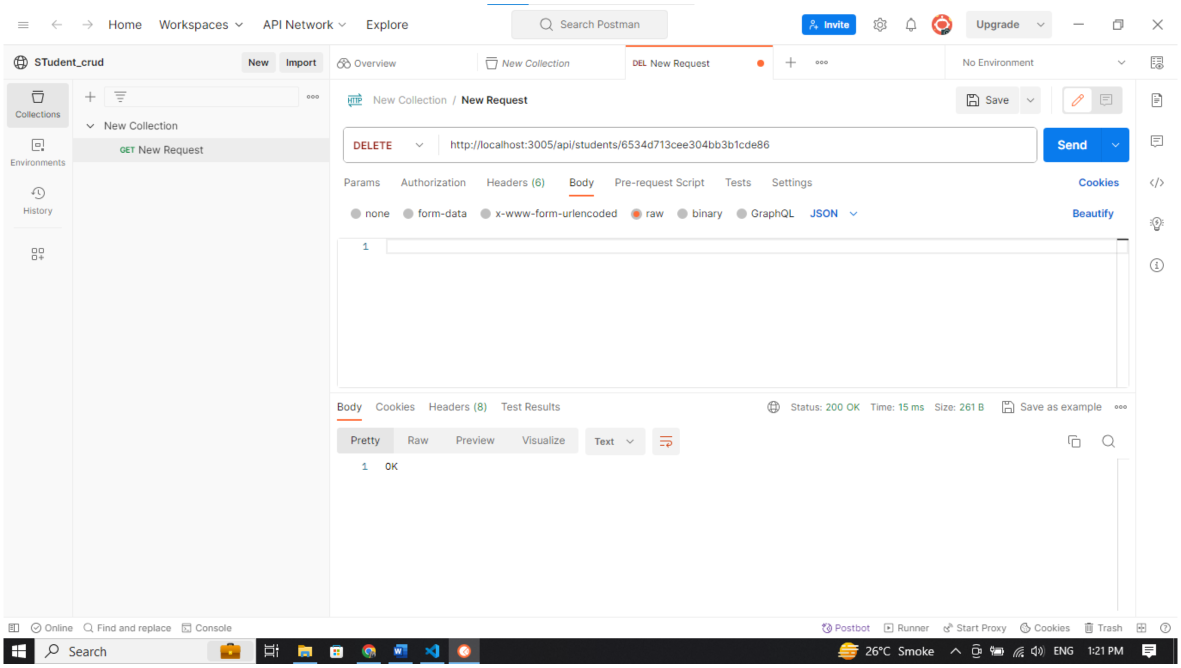
PUT http://localhost:3005/api/students/6534d713cee304bb3b1cde86

```
{
"Password": "123456789"
}
```

Status: 200 OK Time: 17 ms Size: 533 B

```
{
    "_id": "6534d713cee304bb3b1cde86",
    "name": "Yasir Hassan",
    "fatherName": "Zafar Iqbal",
    "registrationNo": "2021-CS-28",
    "contact": "03240859565",
    "dateOfBirth": "2002-12-31T19:00:00.000Z",
    "createdAt": "2023-10-22T08:02:27.599Z",
    "updatedAt": "2023-10-22T08:20:19.793Z",
    "__v": 0
```

6.4 Delete API



DELETE http://localhost:3005/api/students/6534d713cee304bb3b1cde86

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

Status: 200 OK Time: 15 ms Size: 261 B

```
OK
```