

Lab Report SE

CS-165L



Submitted To:

Sir Laeeq Khan Niazi

Submitted By:

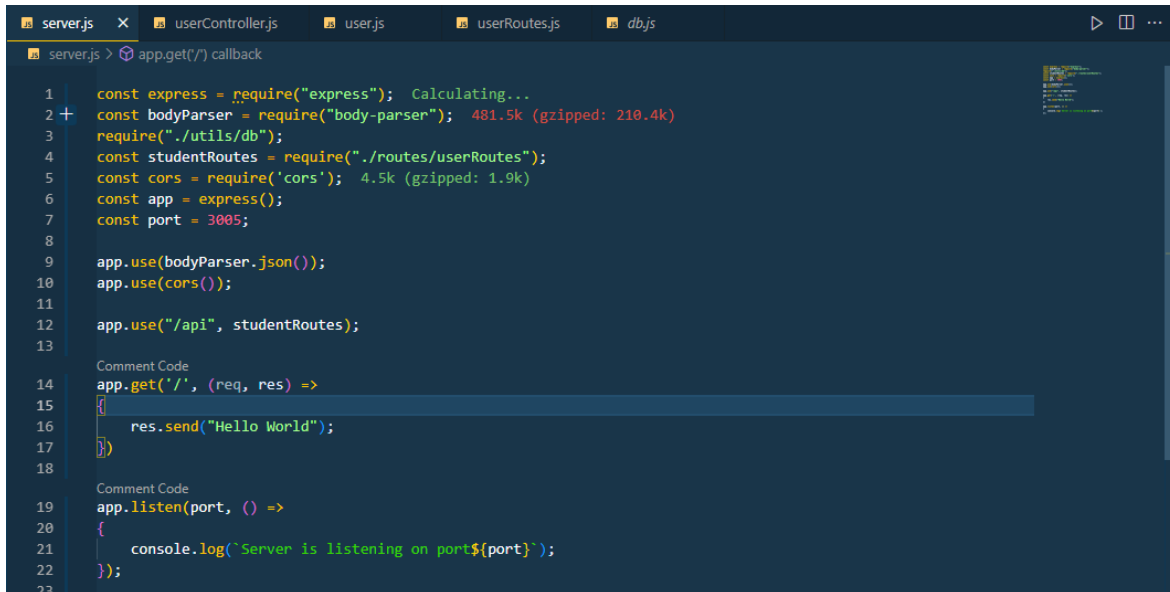
Muhammad Moazam 2021-CS-53

University of Engineering Technology

Lahore

1. Complete Code

1.1 Server

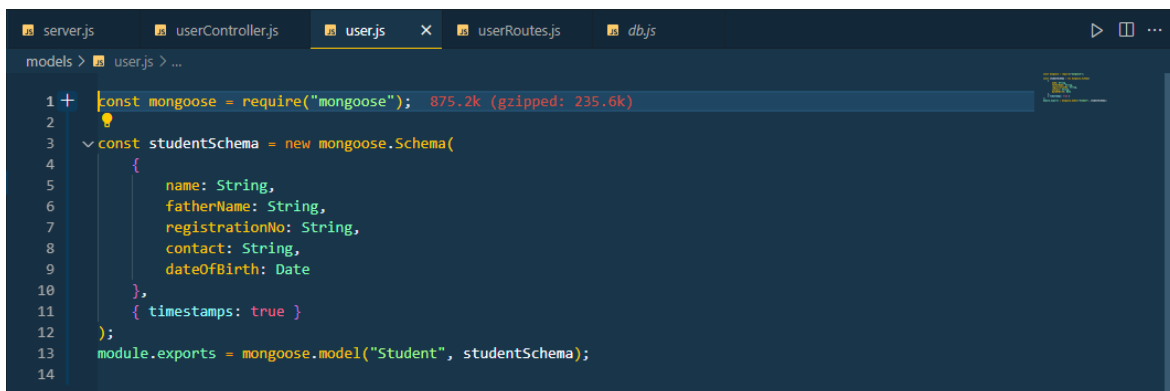


```
server.js x userController.js user.js userRoutes.js db.js
server.js > app.get(/) callback

1  const express = require("express"); Calculating...
2  const bodyParser = require("body-parser"); 481.5k (gzipped: 210.4k)
3  require("../utils/db");
4  const studentRoutes = require("../routes/userRoutes");
5  const cors = require('cors'); 4.5k (gzipped: 1.9k)
6  const app = express();
7  const port = 3005;
8
9  app.use(bodyParser.json());
10 app.use(cors());
11
12 app.use("/api", studentRoutes);
13
14 Comment Code
15 app.get('/', (req, res) =>
16 {
17   res.send("Hello World");
18 });
19
20 Comment Code
21 app.listen(port, () =>
22 {
23   console.log(`Server is listening on port${port}`);
24 });
```

2. Models

2.1 User Model



```
server.js userController.js user.js x userRoutes.js db.js
models > user.js > ...

1  const mongoose = require("mongoose"); 875.2k (gzipped: 235.6k)
2
3  const studentSchema = new mongoose.Schema(
4  {
5    name: String,
6    fatherName: String,
7    registrationNo: String,
8    contact: String,
9    dateOfBirth: Date
10 },
11 { timestamps: true }
12 );
13 module.exports = mongoose.model("Student", studentSchema);
14
```

3. Controllers

3.1 UserController

```
server.js userController.js user.js userRoutes.js app.js
controllers > userController.js > getStudents
1 const Student = require("../models/user");
2
3 // get student
4 comment Code
5 async function addStudent(req, res) {
6   try {
7     const student = await Student.create(req.body);
8     res.status(200).json(student);
9   } catch (err) {
10     res.status(500).json({ error: err.message });
11   }
12 }
13 +
14 // get all students
15 comment Code
16 async function getAllStudent(req, res) {
17   try {
18     const students = await Student.find();
19     res.status(200).json({ student: students, message: "Successfully find" });
20   } catch (err) {
21     res.status(500).json({ error: err.message, message: "Internal server error" });
22   }
23 }
24 // update student by id
25 comment Code
26 async function updateStudent(req, res) {
27   try {
28     const { id } = req.params;
29     const updateStudent = await Student.findByIdAndUpdate(id, req.body, {
30       new: true,
31     });
32     res.json(updateStudent);
33   } catch (err) {
34     res.status(500).json({ error: err.message });
35   }
36 }
37 // delete a student by id
38 comment Code
39 async function deleteStudent(req, res) {
40   try {
41     const { id } = req.params;
42     console.log(id);
43     await Student.findByIdAndDelete(id);
44     res.status(200).json({ error: err.message });
45   } catch (err) {
46     res.status(500).json({ error: err.message });
47   }
48 }
49 module.exports = {
50   addStudent,
51   updateStudent,
52   deleteStudent,
53   getAllStudent
54 }
```

3.2 Login Controller

```
server.js userController.js LoginController.js user.js userRoutes.js
controllers > LoginController.js > LoginUser
1 const userModel = require("../models/user");
2
3 Comment Code
4 async function LoginUser(req, res) {
5   try {
6     const { Username, Password } = req.body;
7     const loginuser = await userModel.findOne({ Username, Password });
8     res.status(200).json({ loginuser });
9   } catch (error) {
10     console.error(error);
11     res.status(500).json({ error: error.message });
12   }
13 +
14 }
15 module.exports = {
16   LoginUser,
17 };
18 }
```

4. Routes

4.1 User Routes

```
server.js  userController.js  LoginController.js  user.js  userRoutes.js X
routes > userRoutes.js > ...

1  const express = require("express");
2  const router = express.Router();
3  const userController = require("../controllers/userController");
4  const loginController = require("../controllers/LoginController");
5
6  router.post("/students", userController.addStudent);
7  router.get("/students", userController.getAllStudent);
8  // update and delete students on the base of their Id
9  router.put("/students/:id", userController.updateStudent);
10 router.delete("/students/:id", userController.deleteStudent);
11 router.post("/login", loginController.LoginUser);
12
13 module.exports = router;
14 +
```

5. Utils

5.1 DB

```
server.js  userController.js  LoginController.js  user.js  userRoutes.js  db.js X
utils > db.js > ...

1  const mongoose = require("mongoose");  875.2k (gzipped: 235.6k)
2  mongoose.set("strictQuery", true);
3
4  mongoose.connect("mongodb://127.0.0.1:27017/student-crud", {
5    useNewUrlParser: true,
6    useUnifiedTopology: true,
7  });
8  const db = mongoose.connection;
9  Comment Code
10 db.on("error", (err) =>
11 {
12   console.log("Failed to connect with database");
13 });
14 Comment Code
15 db.once("open", () =>
16 {
17   console.log("Connected with database");
18 });
19 +
```

6. Testing

6.1 Post API

The screenshot displays the Postman application interface. On the left sidebar, the 'Collections' tab is active, showing a collection named 'Student_crud' with a sub-collection 'New Collection' containing a 'GET New Request'. The main workspace shows the details of this request. The method is 'GET' and the URL is 'http://localhost:3005/api/students'. The 'Body' tab is selected, showing a JSON response. The response status is '200 OK' with a time of '5.18 s' and a size of '1.06 KB'. The response body is a JSON array containing one student object.

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

Params Authorization Headers (6) Body Pre-request Script Tests Settings

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

1

Body Cookies Headers (8) Test Results Status: 200 OK Time: 5.18 s Size: 1.06 KB Save as example

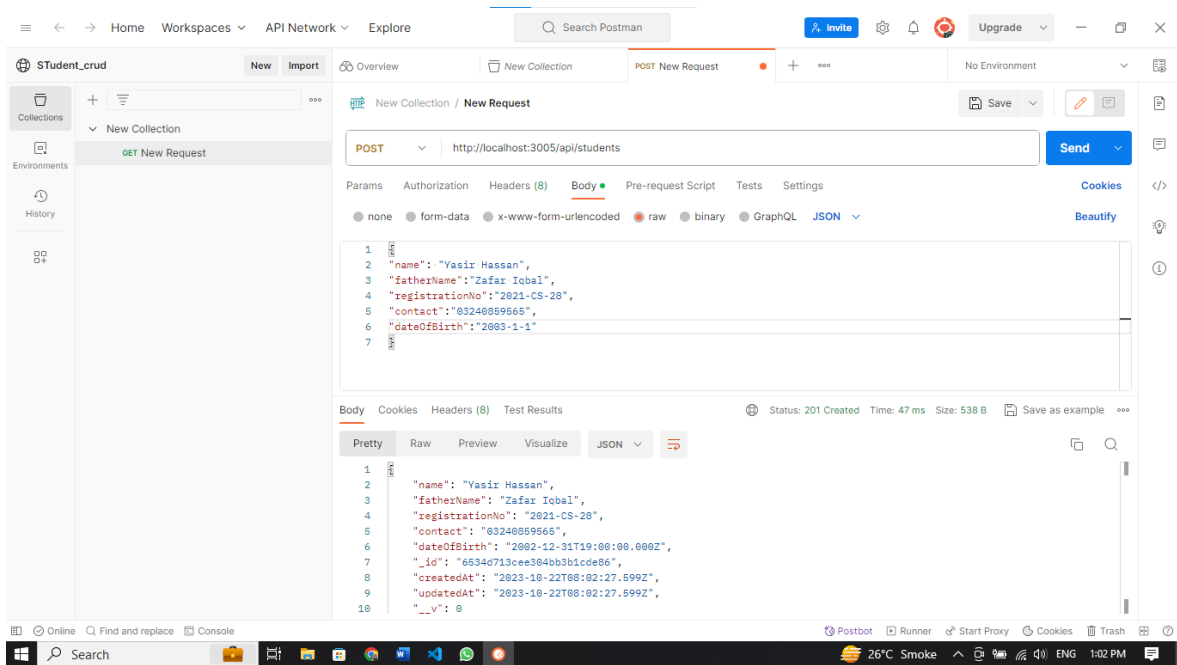
Pretty Raw Preview Visualize JSON

```
1 {
2   "student": [
3     {
4       "_id": "6533c2e8b60c6f66417a8f61",
5       "name": "M Moazan",
6       "fatherName": "M Aslam",
7       "registrationNo": "2021-CS-53",
8       "contact": "0321-9341847",
9       "dateOfBirth": "2023-10-21T08:08.000Z",
10      "createdAt": "2023-10-21T12:24:08.777Z",
11    }
12  ]
13 }
```

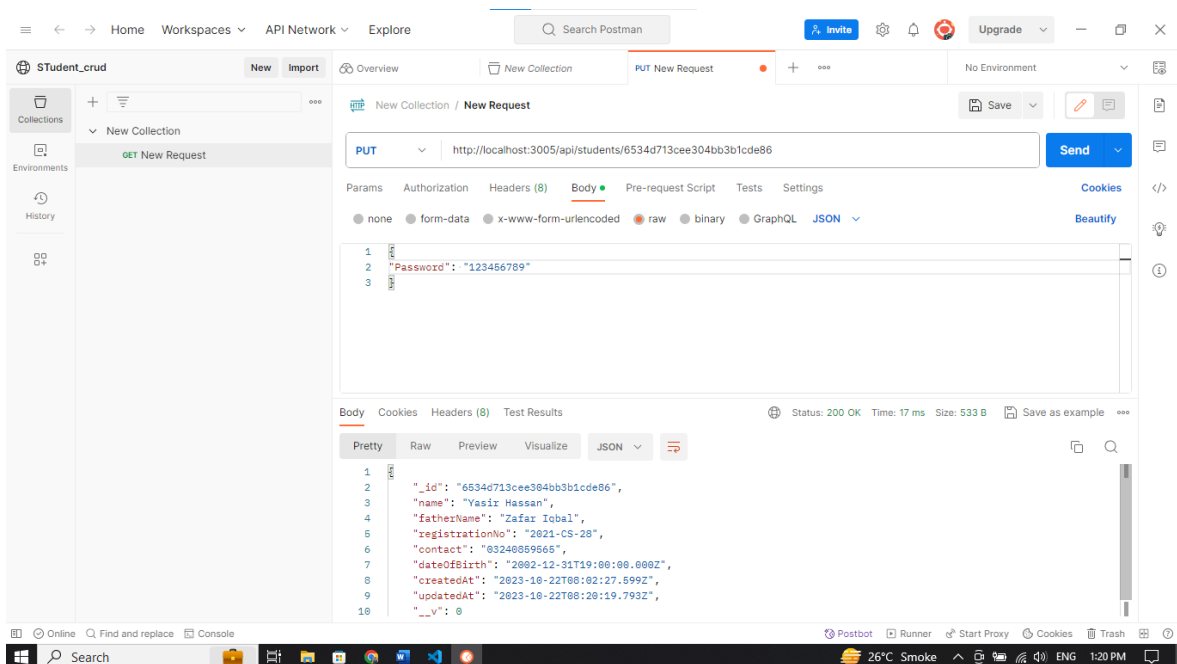
Postbot Runner Start Proxy Cookies Trash

26°C Smoke ENG 1:00 PM

6.2 Get API



6.3 Put API



6.4 Delete API

