Mod 6 task 2

```
|>controllers
|           =>student.control.js
|>routes
|           =>student.js
|app.js

|db.json
|
```

db.json

```
[

  {
    "name": "alan",
    "age": 22,
    "dept": "cse",
    "active": false
  },
  {
    "name": "amal",
    "age": 22,
    "dept": "cse",
    "active": false
  },
  {
    "name": "john",
    "age": 23,
    "dept": "cse",
    "active": true
  }
]
```

App.js

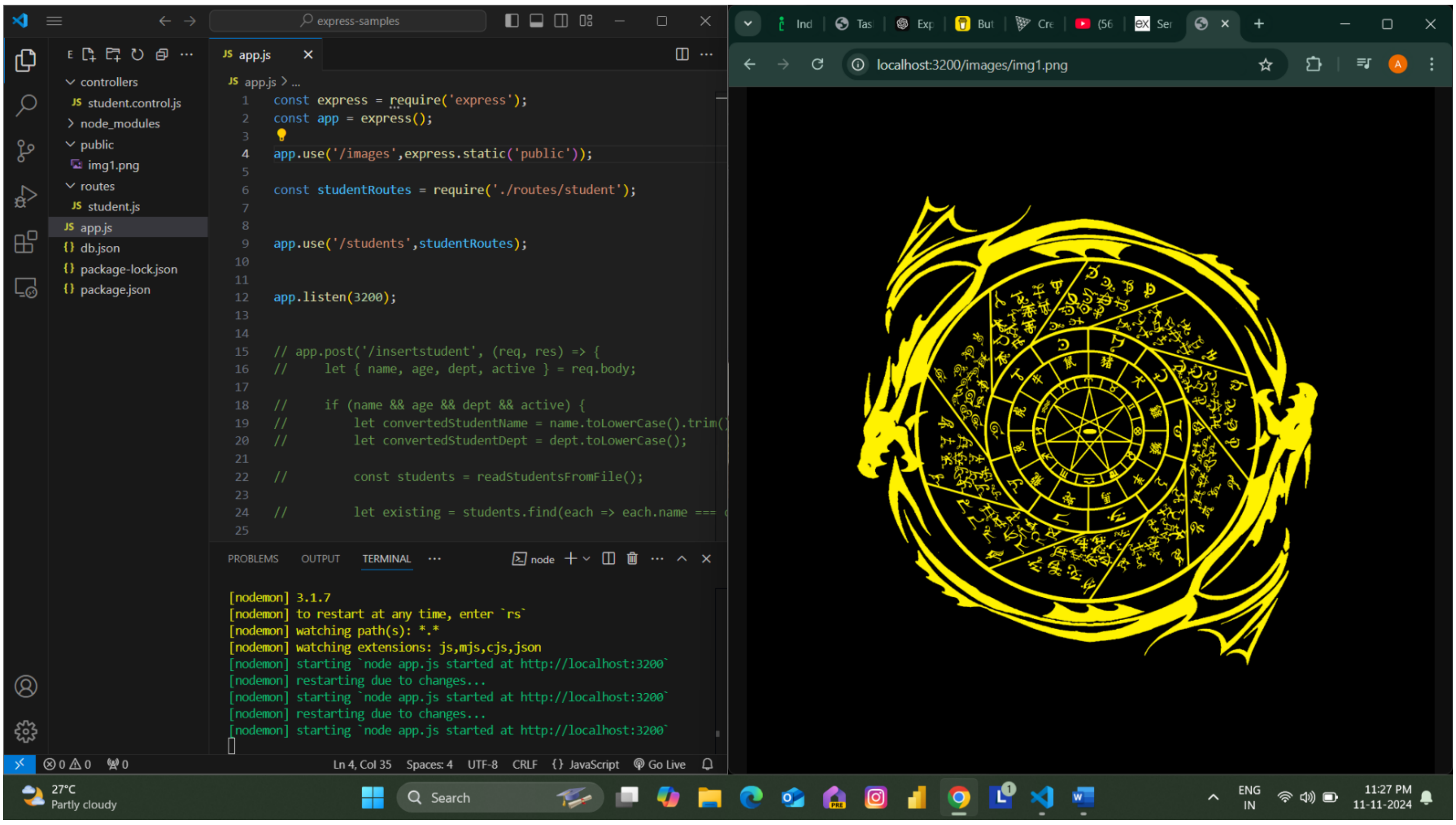
```
const express = require('express');

const app = express();


const studentRoutes = require('./routes/student');


app.use('/students',studentRoutes);


app.listen(3200);
```

student.js

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

const router = express.Router();
const studentControl = require('./../controllers/student.control');



router.route('/getStudents').get(studentControl.getStudents);

router.route('/insertstudent').post(studentControl.insertstudent);


router.route('/deletestudent').delete(studentControl.deletestudent);


router.route('/updatestudent').put(studentControl.updatestudent);


module.exports = router;
```

student.controls.js

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

const dbPath = 'db.json';



function readStudentsFromFile() {
    try {
        const data = fs.readFileSync(dbPath, 'utf-8');
        console.log('file readed.');
        return JSON.parse(data);
    } catch (err) {
        console.error('Error reading db.json:', err);
        return [];
    }
}

function writeStudentsToFile(students) {
    try {
        fs.writeFileSync(dbPath, JSON.stringify(students, null, 2));
        console.log('file writed.');
    } catch (err) {
        console.error('Error writing to db.json:', err);
    }
}

function getStudents(req, res){
        const students = readStudentsFromFile();
        res.json(students);
        console.log('Read file db.json');
}

function insertstudent(req, res){
    let { name, age, dept, active } = req.body;

    if (name && age && dept && active) {
        let convertedStudentName = name.toLowerCase().trim();
        let convertedStudentDept = dept.toLowerCase();

        const students = readStudentsFromFile();

        let existing = students.find(each => each.name ===
convertedStudentName);

        if (existing) {
            res.send(`Student ${name} already exists.`);
        } else {
            let newStudent = { name: convertedStudentName, age, dept:
convertedStudentDept, active };
            students.push(newStudent);
            writeStudentsToFile(students);
            res.send(`Inserted student ${name} successfully.`);
        }

    } else {
        res.send('Please provide all necessary fields(name, age, dept,
active).');
    }
}


function deletestudent(req, res) {
    let studentName = req.query.name;

    if (studentName) {
        let convertedStudentName = studentName.toLowerCase().trim();

        const students = readStudentsFromFile();
        const index = students.findIndex(each => each.name.toLowerCase() ===
convertedStudentName);

        if (index >= 0) {
            students.splice(index, 1);
            writeStudentsToFile(students);
            res.send(`Student ${studentName} has been deleted successfully.`);
        } else {
            res.send(`Student ${studentName} not found.`);
        }

    } else {
        res.send('Please enter a valid name to delete.');
    }
}

function updatestudent(req, res) {
    let name = req.query.name;
    let newData = req.body;

    if (name) {
        const students = readStudentsFromFile();
        const index = students.findIndex(each => each.name.toLowerCase() ===
name.toLowerCase());

        if (index >= 0) {
            students[index] = {
                name: newData.name || students[index].name,
                age: newData.age || students[index].age,
                dept: newData.dept || students[index].dept,
                active: newData.active !== undefined ? newData.active :
students[index].active
            };
            writeStudentsToFile(students);
            res.json({
                message: `Updated ${name} successfully.`,
                updatedStudent: students[index]
            });
        } else {
            res.json({ message: `Student ${name} not found.` });
        }
    } else {
        res.send('Please provide a name to update.');
    }
}

module.exports = {
    getStudents,insertstudent,deletestudent,updatestudent
}
```



```
const express = require('express');
const app = express();

app.use('/images',express.static('public'));

const studentRoutes = require('./routes/student');


app.use('/students',studentRoutes);


app.listen(3200);


// app.post('/insertstudent', (req, res) => {
//     let { name, age, dept, active } = req.body;

//     if (name && age && dept && active) {
//         let convertedStudentName = name.toLowerCase().trim(
//         let convertedStudentDept = dept.toLowerCase();

//         const students = readStudentsFromFile();

//         let existing = students.find(each => each.name ===
```

```
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js started at http://localhost:3200`
[nodemon] restarting due to changes...
[nodemon] starting `node app.js started at http://localhost:3200`
[nodemon] restarting due to changes...
[nodemon] starting `node app.js started at http://localhost:3200`
```

```javascript
const express = require('express');
const app = express();
const studentRoutes = require('./routes/student');

app.use('/images',express.static('public'));
app.use((req,res,next)=>{
    let time = new Date();
    console.log(`(${time.toDateString()}) ${req.method}/ ${req.url}`);
    next();
});
app.use('/students',studentRoutes);


app.listen(3200);
```

[nodemon] restarting due to changes...

[nodemon] starting `node app.js started at http://localhost:3200`

(Mon Nov 11 2024) GET/

[nodemon] restarting due to changes...

[nodemon] starting `node app.js started at http://localhost:3200`

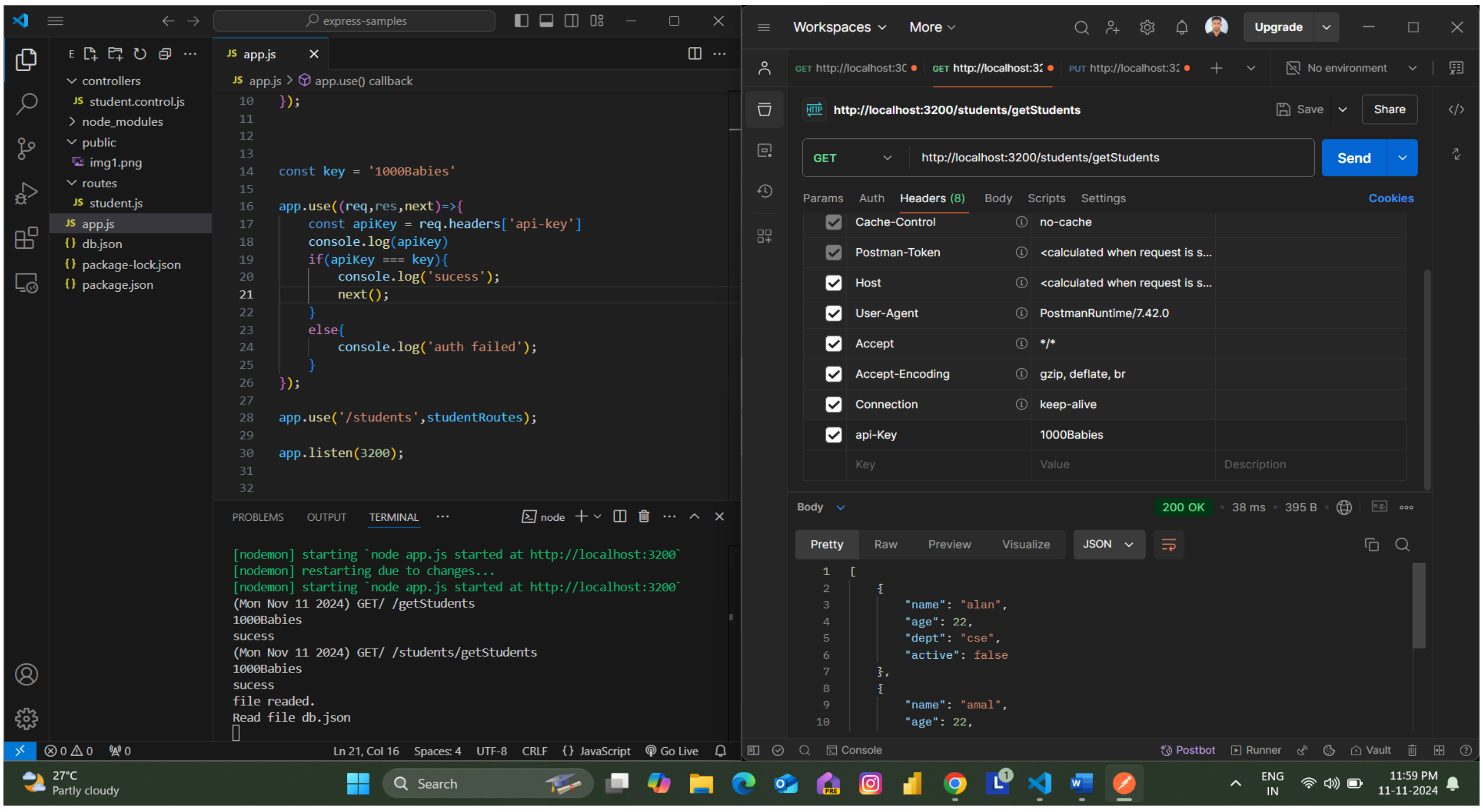
(Mon Nov 11 2024) GET/ /

(Mon Nov 11 2024) GET/ /images/

(Mon Nov 11 2024) GET/ /students/getStudents

file readed.

Read file db.json



```javascript
});



const key = '1000Babies'

app.use((req,res,next)=>{
    const apiKey = req.headers['api-key']
    console.log(apiKey)
    if(apiKey === key){
        console.log('sucess');
        next();
    }
    else{
        console.log('auth failed');
    }
});

app.use('/students',studentRoutes);

app.listen(3200);
```

```
[nodemon] starting `node app.js started at http://localhost:3200`
[nodemon] restarting due to changes...
[nodemon] starting `node app.js started at http://localhost:3200`
(Mon Nov 11 2024) GET/ /getStudents
1000Babies
sucess
(Mon Nov 11 2024) GET/ /students/getStudents
1000Babies
sucess
file readed.
Read file db.json
```

GET http://localhost:3200/students/getStudents

	Key	Value
✓	Cache-Control	no-cache
✓	Postman-Token	<calculated when request is s...
✓	Host	<calculated when request is s...
✓	User-Agent	PostmanRuntime/7.42.0
✓	Accept	*/*
✓	Accept-Encoding	gzip, deflate, br
✓	Connection	keep-alive
✓	api-Key	1000Babies

200 OK · 38 ms · 395 B

```json
[
    {
        "name": "alan",
        "age": 22,
        "dept": "cse",
        "active": false
    },
    {
        "name": "amal",
        "age": 22,
```