

RESTFUL OPERATION(devops)

M.Nishitha

2303A51540

bt no 19

simple Restfull operation

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

const app = express();

app.use(express.json());

app.get('/students', (req, res) => {

res.json([ { id: 1, name: "Ravi" } ]);

});

app.post('/students', (req, res) => {

const student = req.body;

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

});

app.listen(3000, () => {

console.log("Server running on port 3000");

});
```

restfull operation with HTML

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

const app = express();

app.use(express.urlencoded({ extended: true }));

app.use(express.json());

let students = [];

app.get('/', (req, res) => {

res.send(`

<h2>Add Student</h2>

<form method="POST" action="/students">

<input type="text" name="name" placeholder="Enter name" />

<button type="submit">Add</button>
```

```
</form>
```

```
`);
```

```
});
```

```
app.post('/students', (req, res) => {
```

```
  const name = req.body.name;
```

```
  const newStudent = {
```

```
    id: students.length + 1,
```

```
    name: name
```

```
  };
```

```
  students.push(newStudent);
```

```
  res.send("Student Added Successfully!");
```

```
});
```

```
app.get('/students', (req, res) => {
```

```
  res.json(students);
```

```
});
```

```
app.listen(3000, () => {
```

```
  console.log("Server running on port 3000");
```

```
});
```

```
#### resfull operation with POSTMAN desktop app(https://www.postman.com/downloads/)
```

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

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

```
app.use(express.json());
```

```
let students = [
```

```
  { id: 1, name: "Ravi" },
```

```
  { id: 2, name: "Sita" }
```

```
];
```

```
// 1 GET - View all students
```

```
app.get('/students', (req, res) => {
```

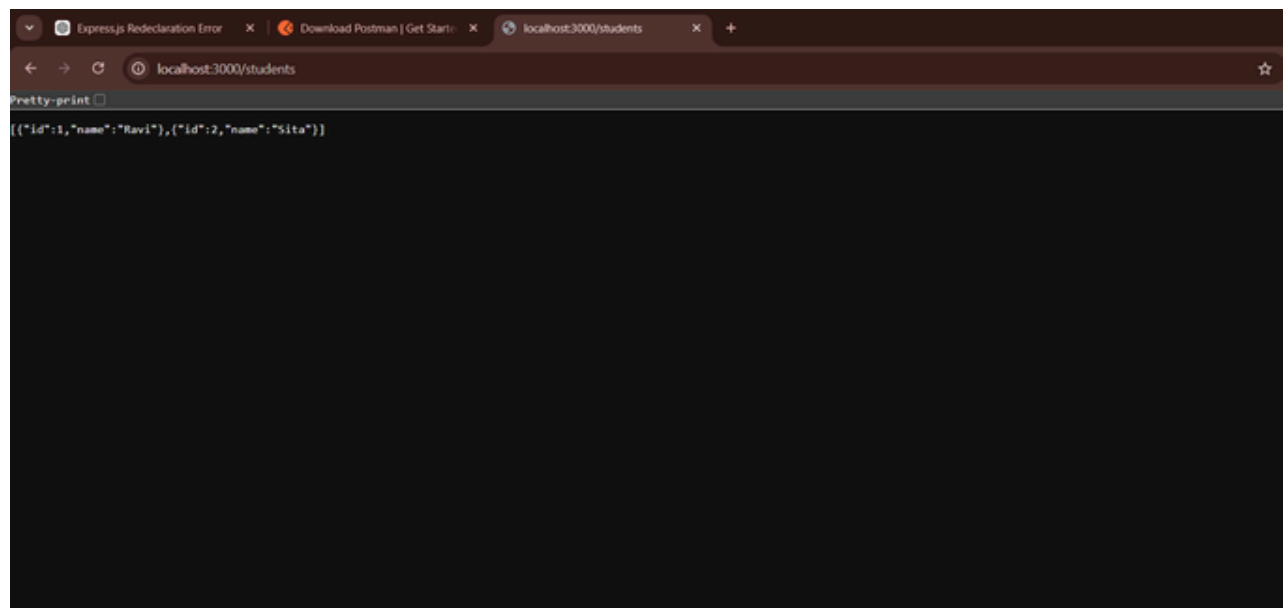
```
  res.json(students);
```

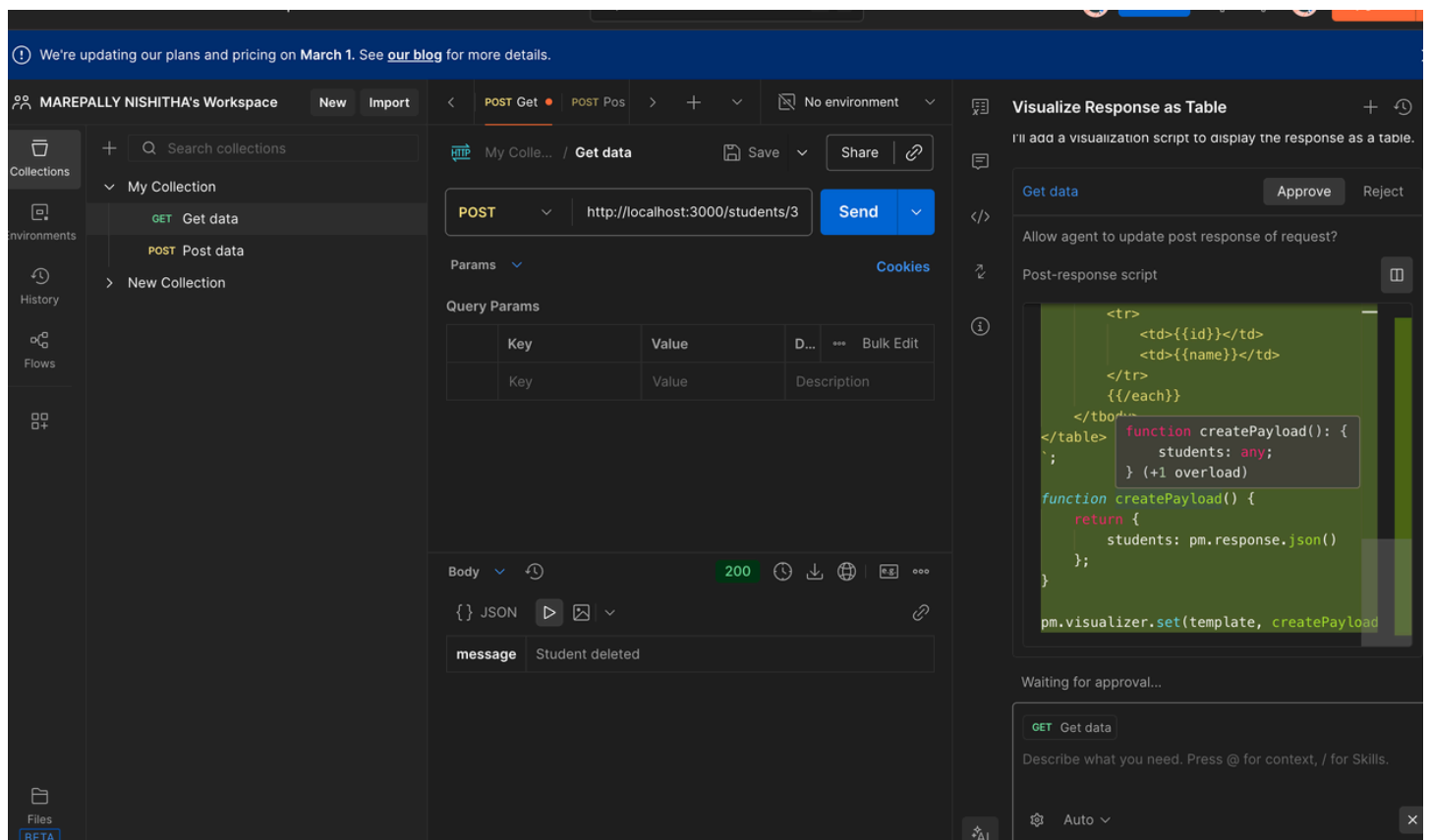
```
}); GET OPERATION:
```

/ 1 GET - View all students

```
app.get('/students', (req, res) => {  
  
  res.json(students);  
  
});OUTPUT:
```

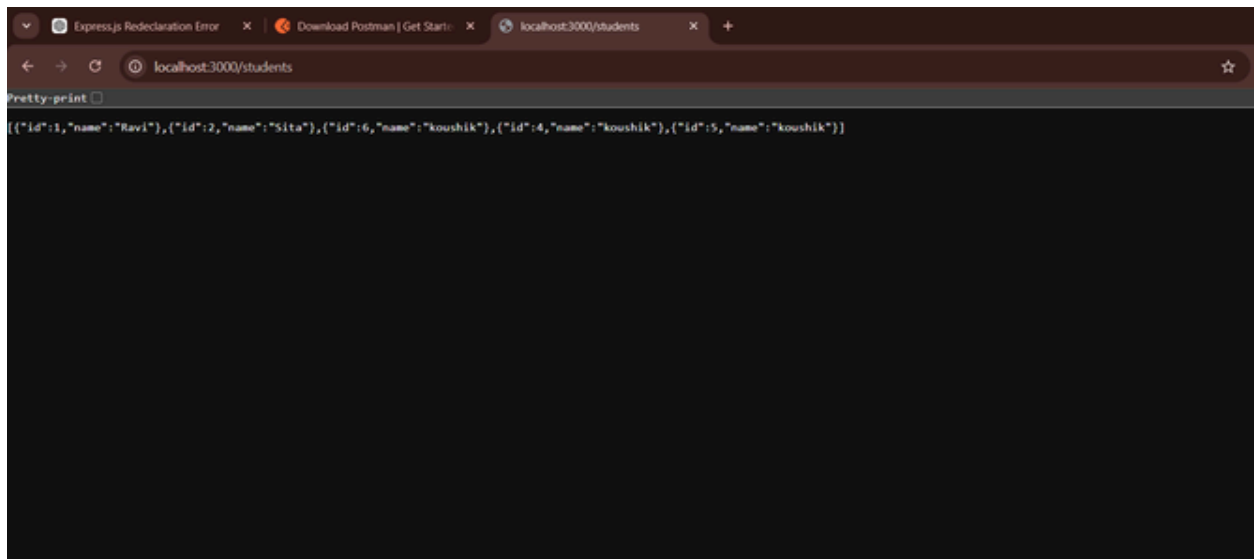
```
dev2.html      package-lock.json      student2.js  
dev01.js       package.json  
node_modules   result.js  
nishithamarepally@nishithas-MacBook-Air devopps % node result.js  
Server running on port 3000  
SyntaxError: Bad control character in string literal in JSON at position 20 (line 2 column 19)  
    at JSON.parse (<anonymous>)  
    at parse (/Users/nishithamarepally/Desktop/devopps/node_modules/body-parser/lib/types/json.js:72:19)  
    at /Users/nishithamarepally/Desktop/devopps/node_modules/body-parser/lib/read.js:162:18  
    at AsyncResource.runInAsyncScope (node:async_hooks:214:14)  
    at invokeCallback (/Users/nishithamarepally/Desktop/devopps/node_modules/raw-body/index.js:238:16)  
    at done (/Users/nishithamarepally/Desktop/devopps/node_modules/raw-body/index.js:227:7)  
    at IncomingMessage.onEnd (/Users/nishithamarepally/Desktop/devopps/node_modules/raw-body/index.js:287:7)  
    at IncomingMessage.emit (node:events:508:28)  
    at endReadableNT (node:internal/streams/readable:1701:12)  
    at process.processTicksAndRejections (node:internal/process/task_queues:89:21)
```



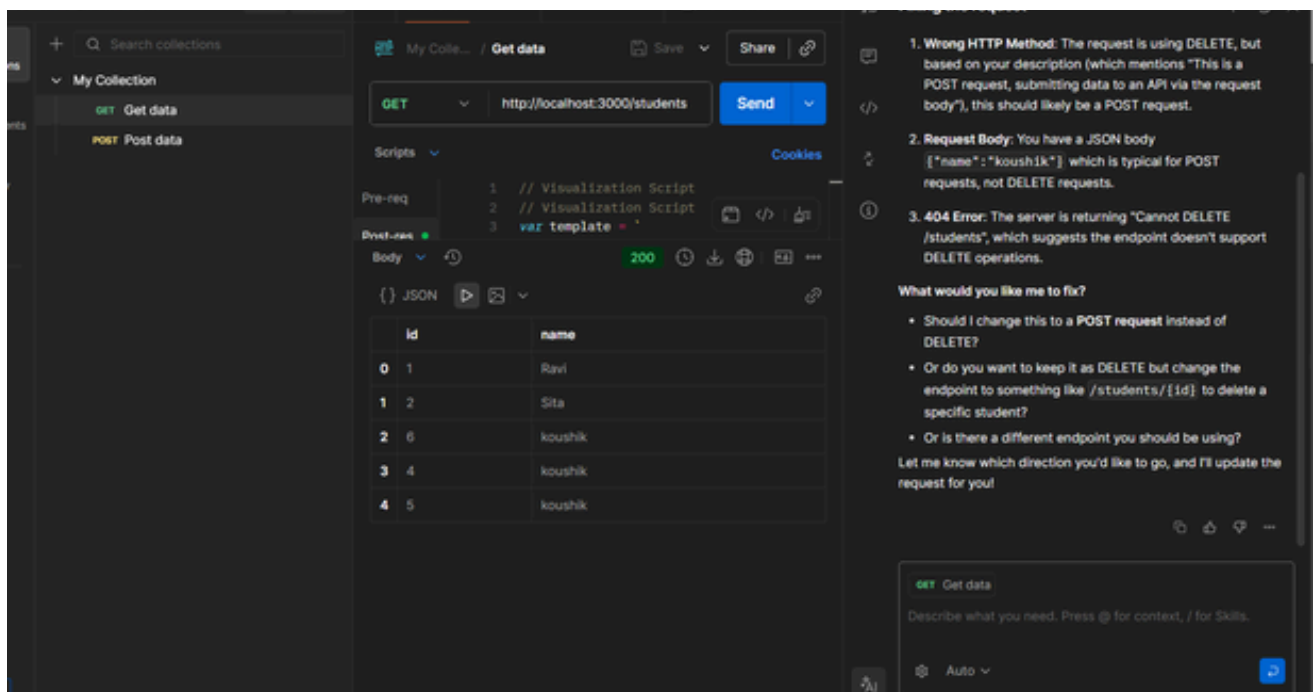


// 2 GET - View single student

```
app.get('/students/:id', (req, res) => {  
  const id = parseInt(req.params.id);  
  const student = students.find(s => s.id === id);  
  if (!student) {  
    return res.status(404).json({ message: "Student not found" });  
  }  
  res.json(student);  
});
```



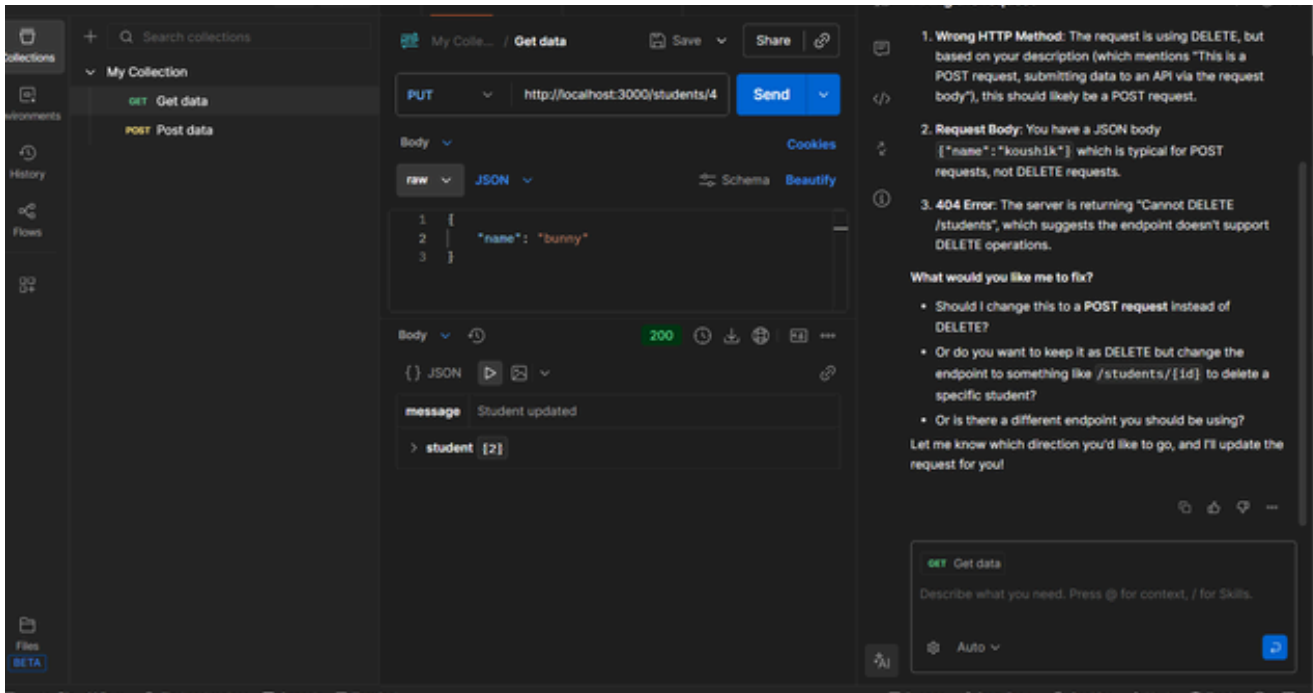
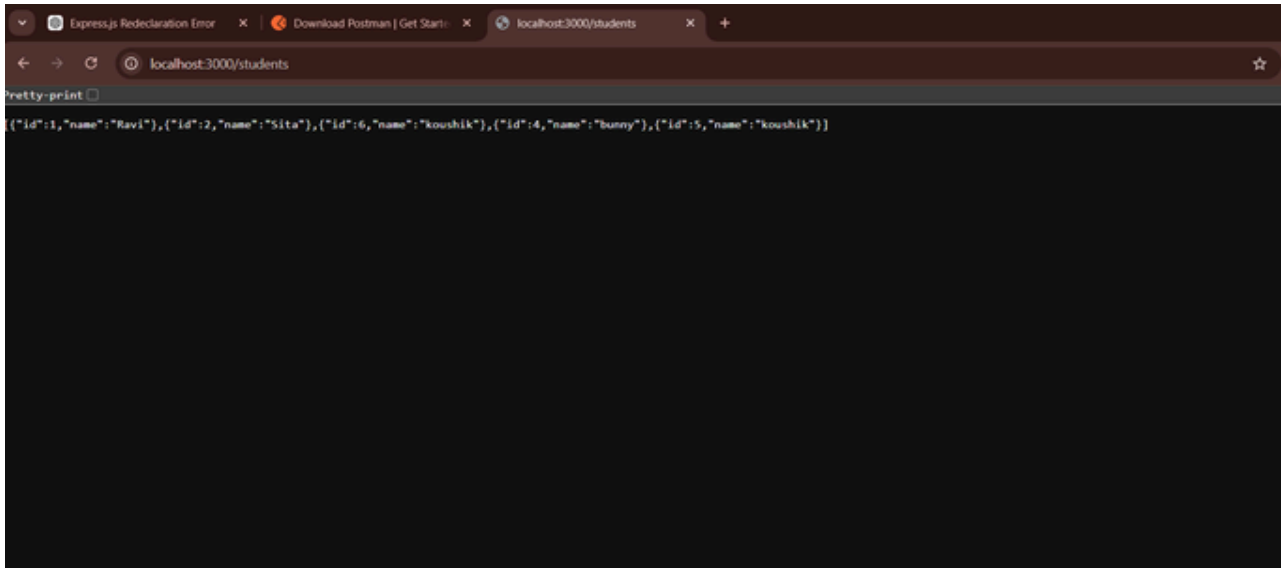
OUTPUT:



/ 4 PUT - Update student

```
app.put('/students/:id', (req, res) => {  
  const id = parseInt(req.params.id);  
  const { name } = req.body;  
  const student = students.find(s => s.id === id);  
  if (!student) {  
    return res.status(404).json({ message: "Student not found" });  
  }  
  student.name = name;  
  res.json({ message: "Student updated", student });  
});
```

}); OUTPUT:



// 5 DELETE - Remove student

```
app.delete('/students/:id', (req, res) => {  
  const id = parseInt(req.params.id);  
  students = students.filter(s => s.id !== id);  
  res.json({ message: "Student deleted" });  
});  
app.listen(3000, () => {  
  console.log("Server running on port 3000");  
});
```

}); OUTPUT:

