

## Experiment No: 7: Validating RESTful APIs using Postman.

### Code:

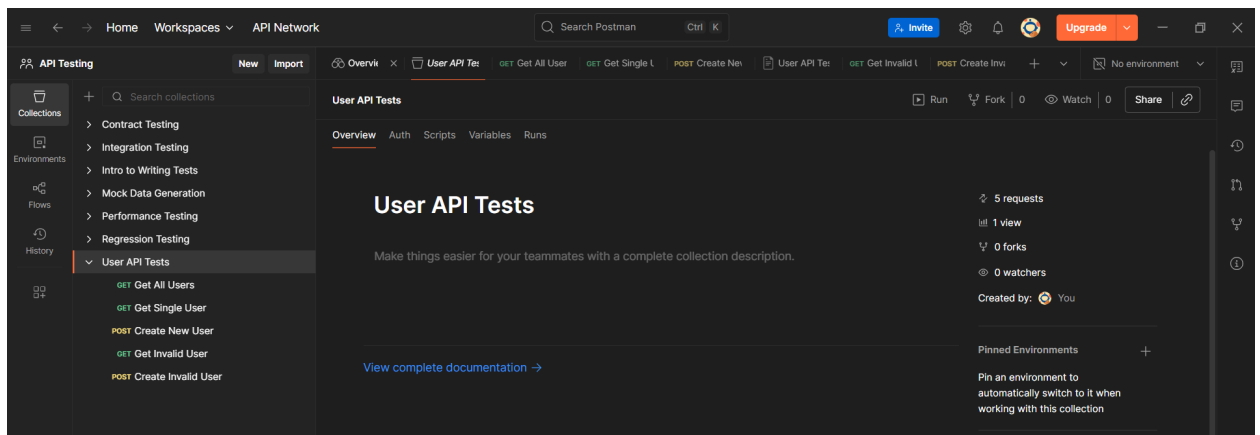
```
JS api.js > ...
CodeMate
1 | const express = require("express");
CodeMate
2 | const app = express();
3 |
4 | app.use(express.json());
5 |
6 | // Sample data
CodeMate
7 | let users = [
8 |   { id: 1, name: "John Doe", email: "john@example.com" },
9 |   { id: 2, name: "Jane Smith", email: "jane@example.com" },
10 | ];
11 |
12 | // GET all users
Tabnine | Edit | Test | Explain | Document | CodeMate
13 | app.get("/api/users", (req, res) => {
14 |   res.json(users);
15 | });
16 |
17 | // GET user by id
Tabnine | Edit | Test | Explain | Document | CodeMate
18 | app.get("/api/users/:id", (req, res) => {
19 |   const user = users.find((u) => u.id === parseInt(req.params.id));
20 |   if (!user) return res.status(404).json({ message: "User not found" });
21 |   res.json(user);
22 | });
23 |
24 | // POST new user
Tabnine | Edit | Test | Explain | Document | CodeMate
25 | app.post("/api/users", (req, res) => {
26 |   const user = {
27 |     id: users.length + 1,
28 |     name: req.body.name,
29 |     email: req.body.email,
30 |   };
31 |   users.push(user);
32 |   res.status(201).json(user);
33 | });
34 |
35 | Tabnine | Edit | Test | Explain | Document | CodeMate
36 | app.listen(3000, () => {
37 |   console.log("Server running on port 3000");
38 | });
```

## Server running:

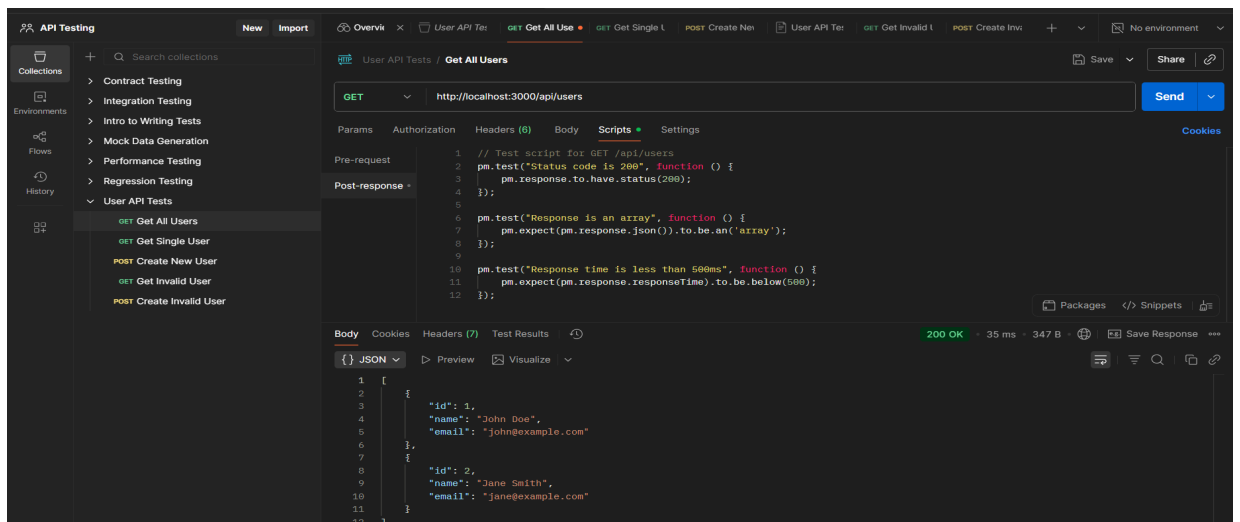
```
Node.js v22.14.0  
E:\VESIT 5th Sem\FSD\EXP 7>node api.js  
Server running on port 3000
```

## Output:-

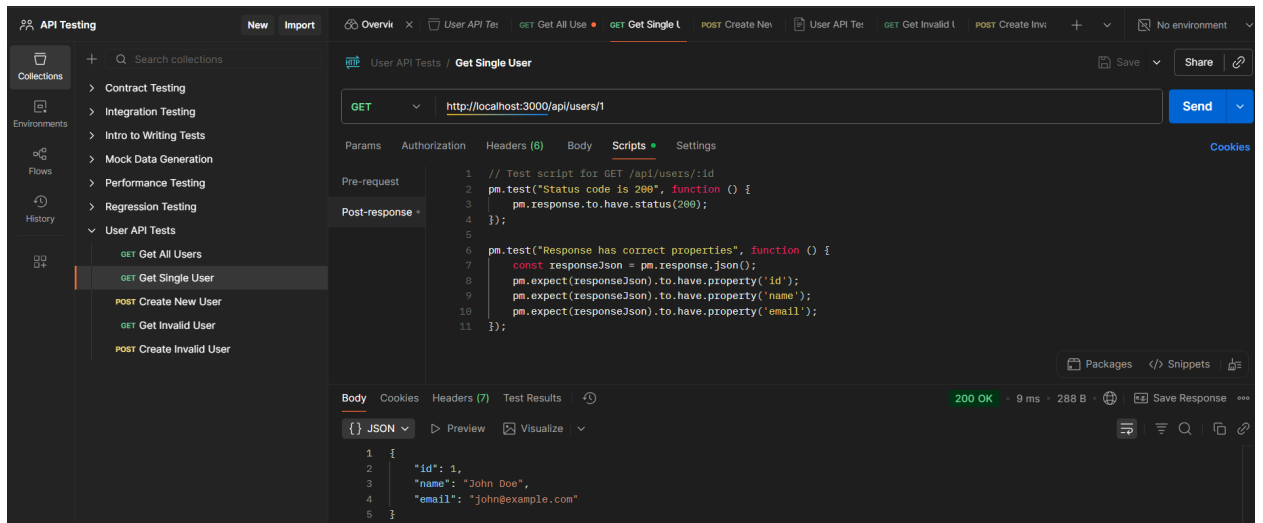
1.



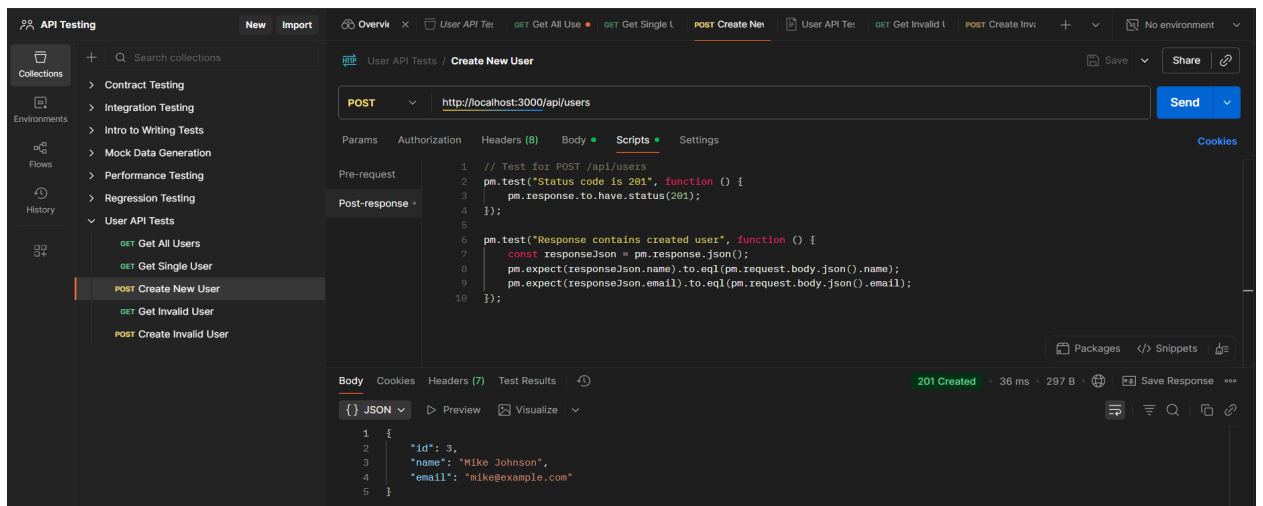
2.



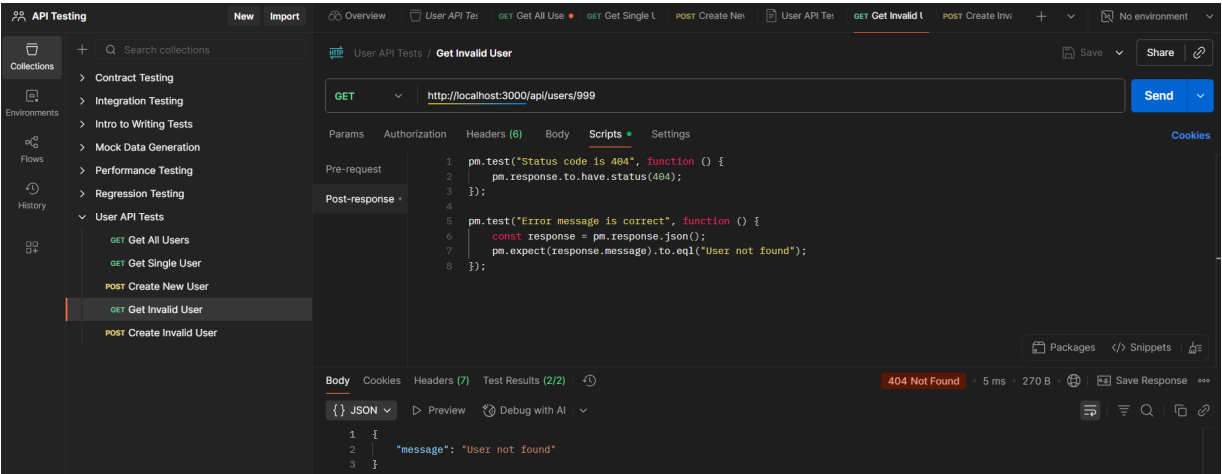
3.



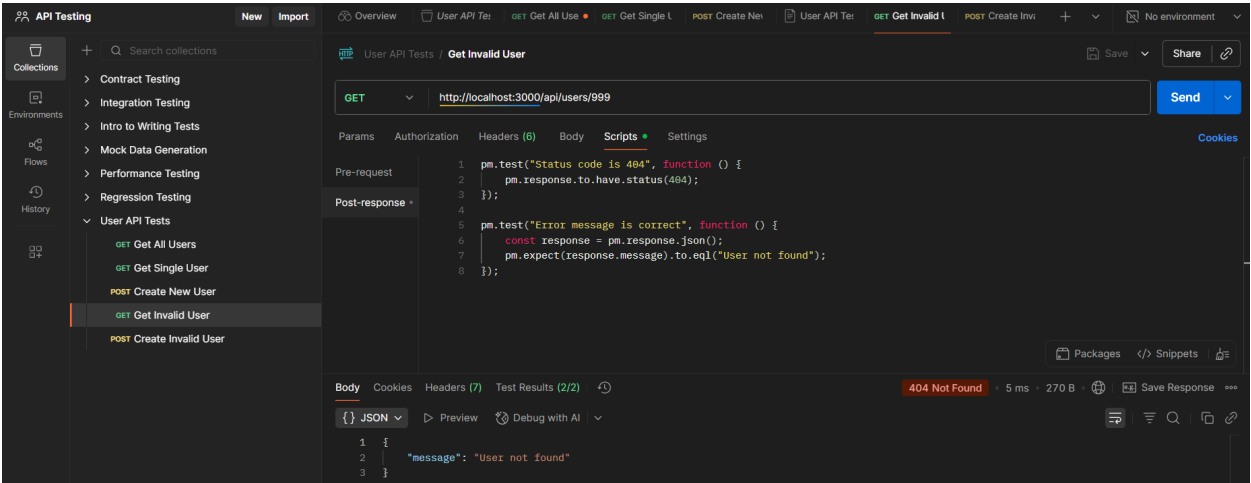
4.



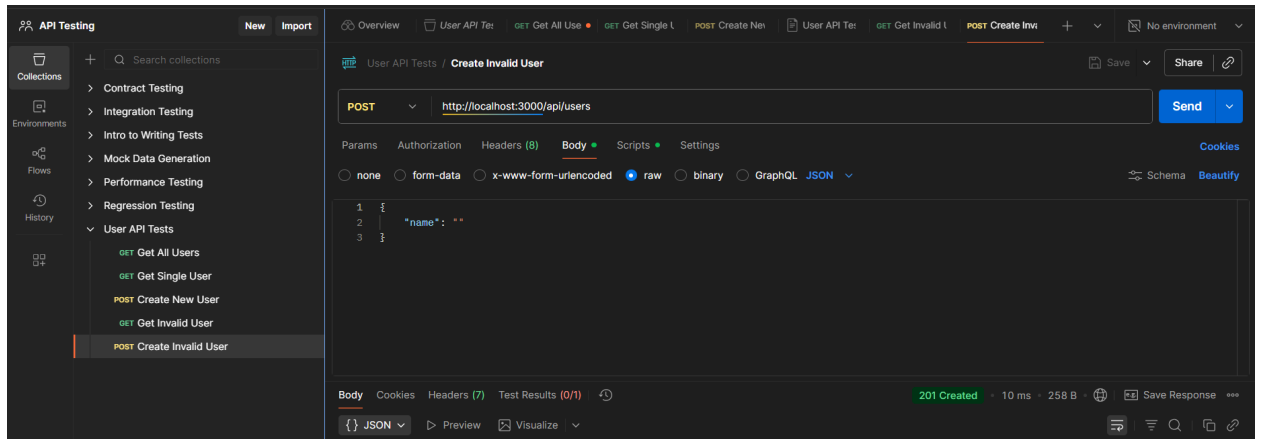
5.



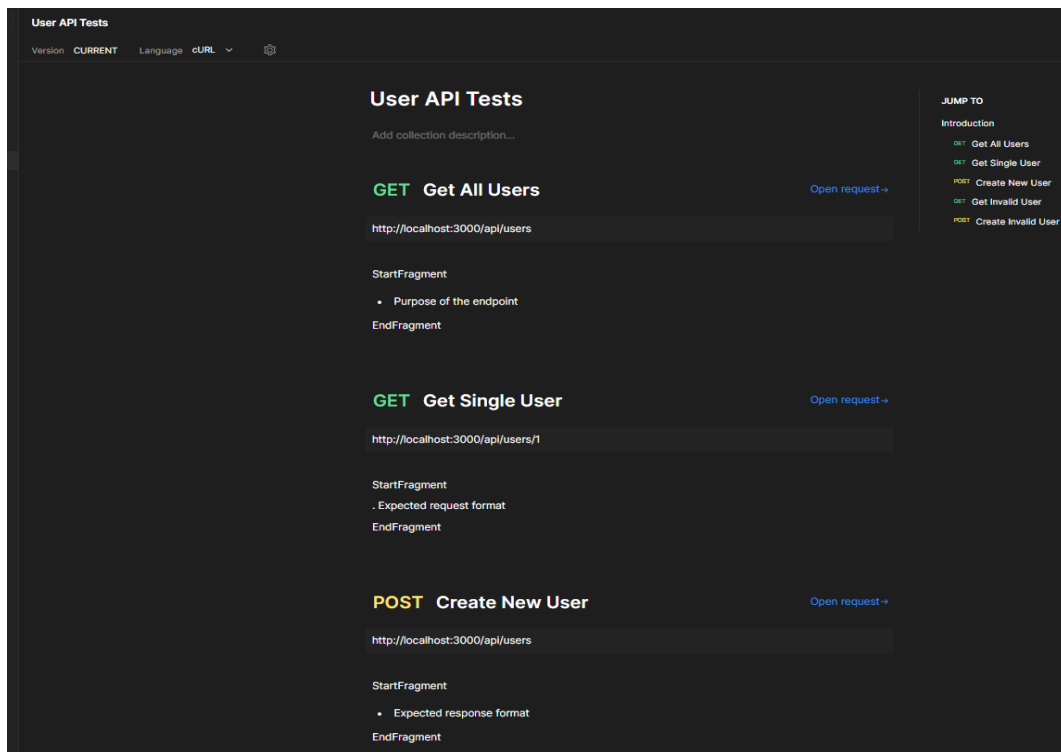
6.



7.



8.



9.

**GET** Get Invalid User [Open request →](#)

`http://localhost:3000/api/users/999`

Add request description...

**POST** Create Invalid User [Open request →](#)

`http://localhost:3000/api/users`

Add request description...

**Body** raw (json)

json

```
{  
  "name": ""  
}
```