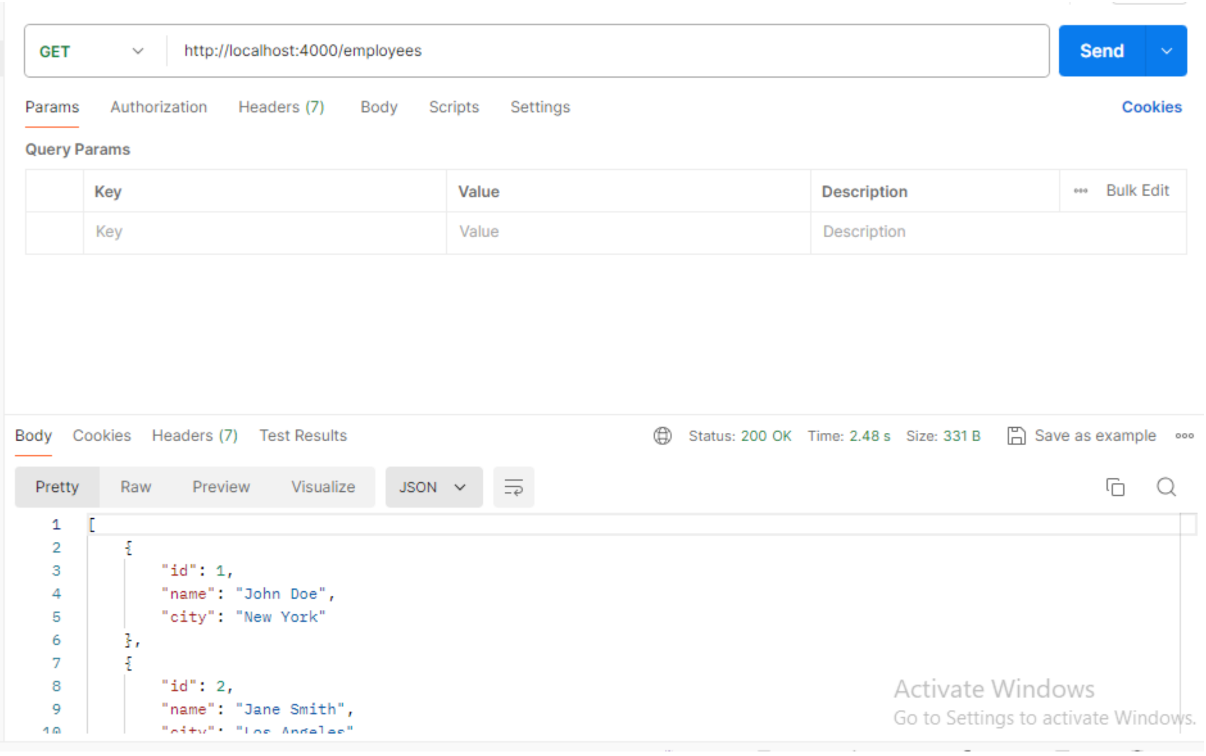
Day-24 Assignment (API)

1. Employees

(a) - get all employees



GET http://localhost:4000/employees

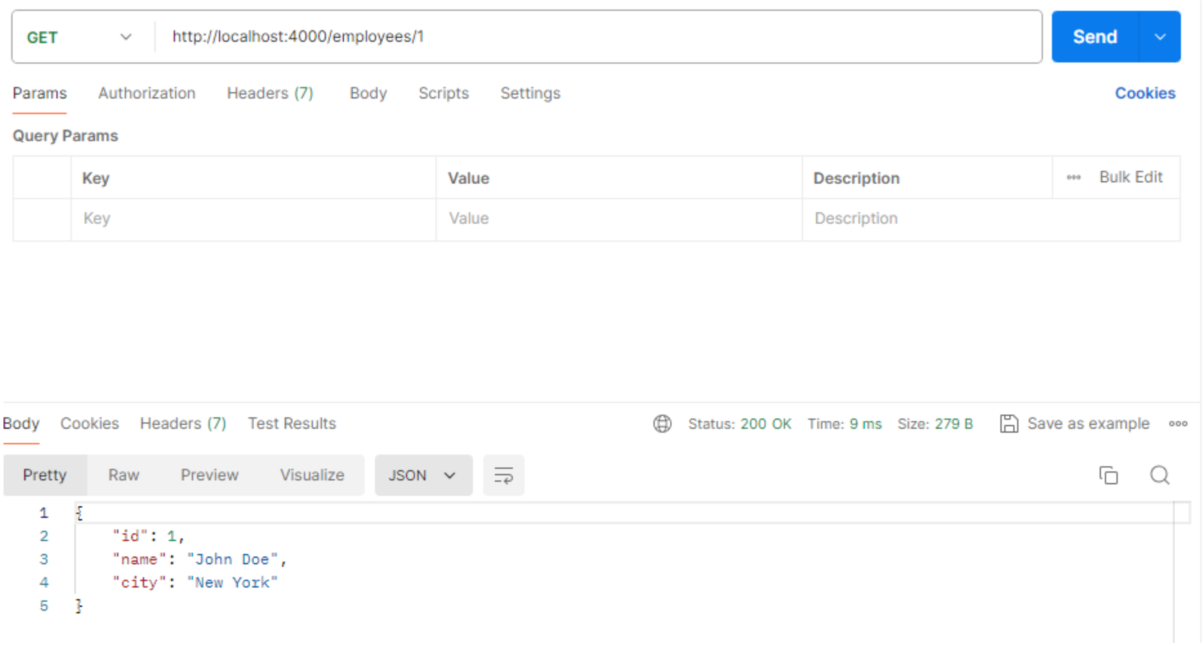
Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Status: 200 OK Time: 2.48 s Size: 331 B

```
[
    {
        "id": 1,
        "name": "John Doe",
        "city": "New York"
    },
    {
        "id": 2,
        "name": "Jane Smith",
        "city": "Los Angeles"
```

(b) Get specific employee using id



GET http://localhost:4000/employees/1

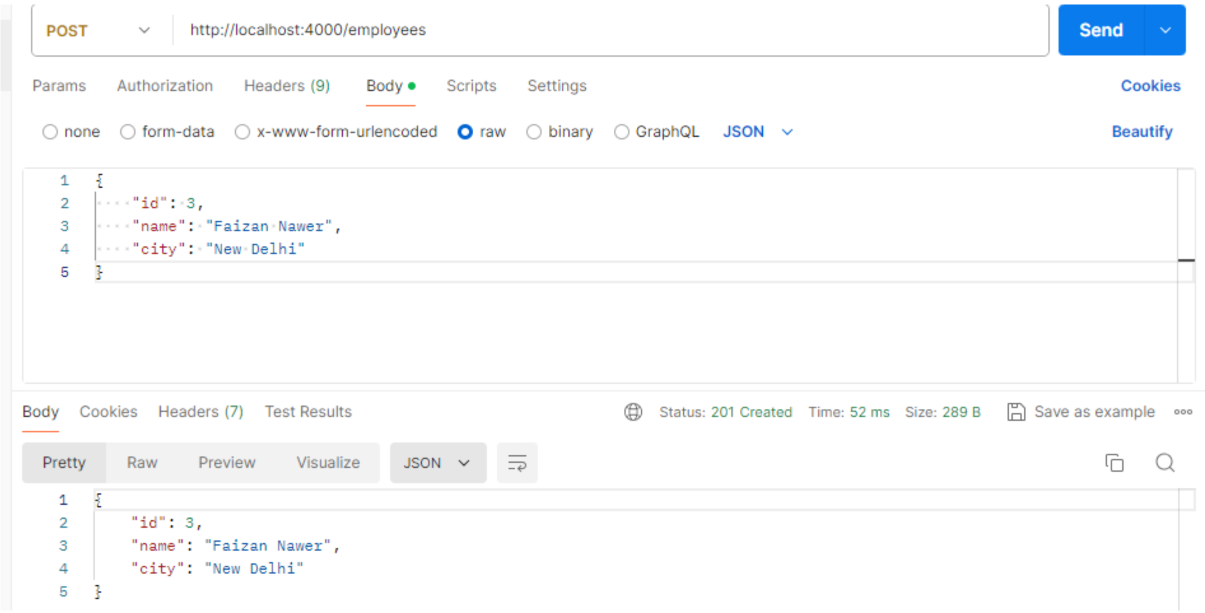
Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Status: 200 OK Time: 9 ms Size: 279 B

```
{
    "id": 1,
    "name": "John Doe",
    "city": "New York"
}
```

(c) Creating an employee



POST http://localhost:4000/employees

● raw JSON

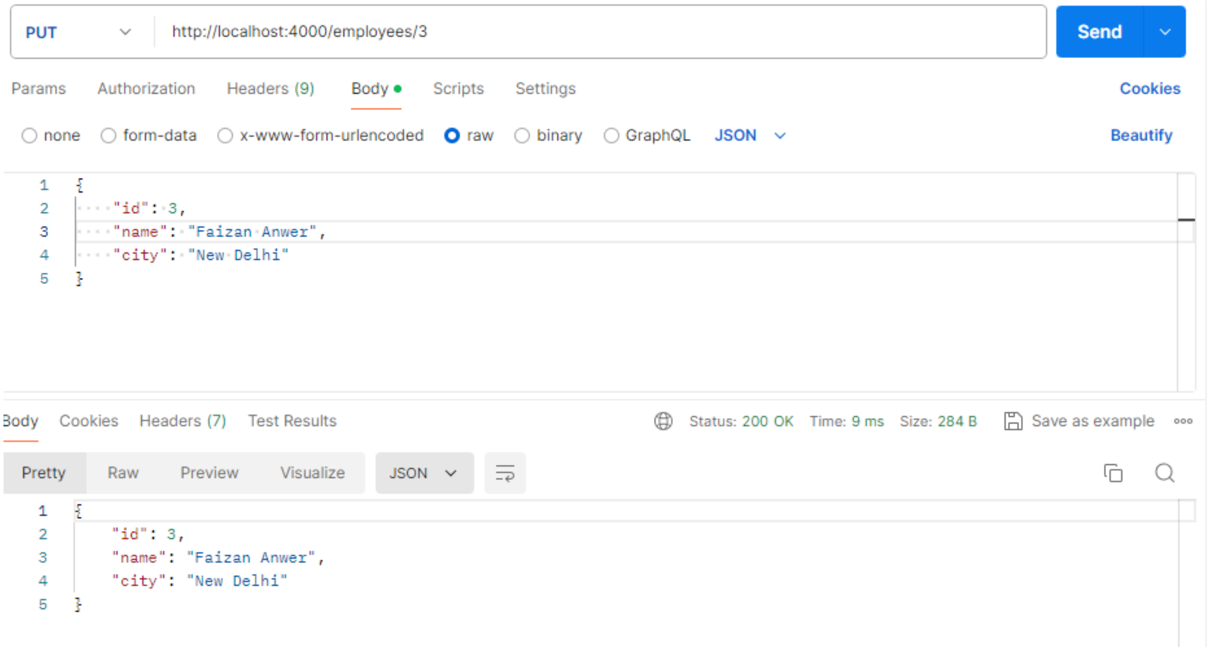
```
{
    "id": 3,
    "name": "Faizan Nawer",
    "city": "New Delhi"
}
```

Status: 201 Created Time: 52 ms Size: 289 B

JSON

```
{
    "id": 3,
    "name": "Faizan Nawer",
    "city": "New Delhi"
}
```

(d) Update employee using id



PUT http://localhost:4000/employees/3

● raw JSON

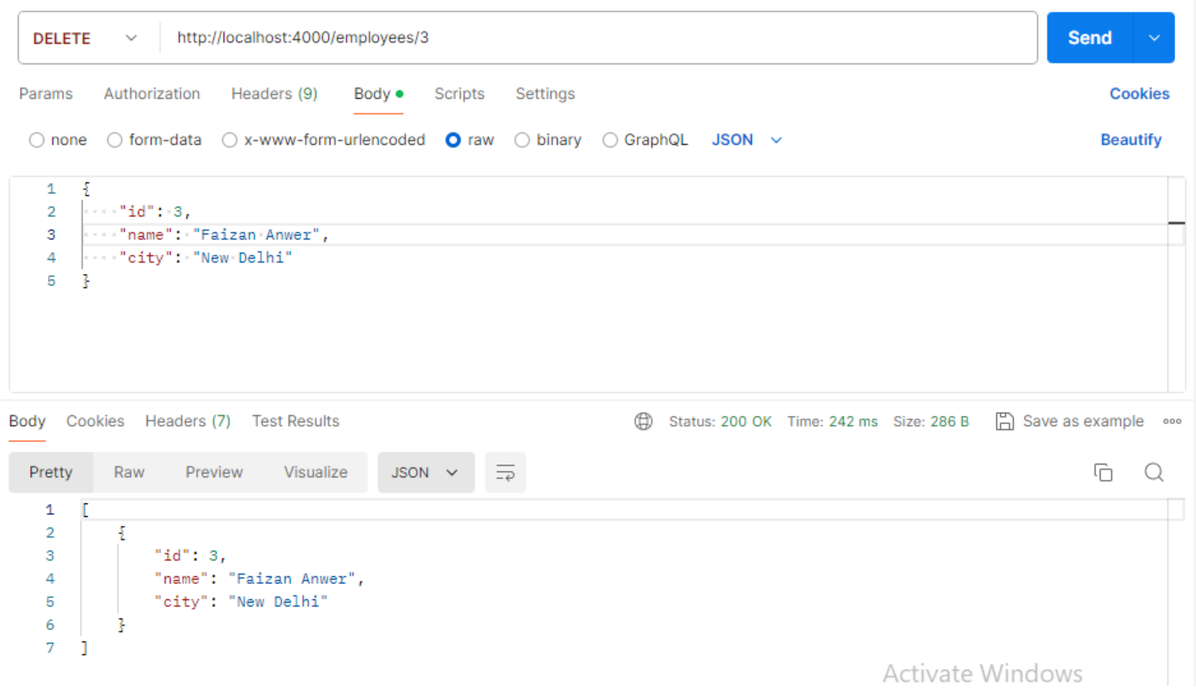
```
{
    "id": 3,
    "name": "Faizan Anwer",
    "city": "New Delhi"
}
```

Status: 200 OK Time: 9 ms Size: 284 B

JSON

```
{
    "id": 3,
    "name": "Faizan Anwer",
    "city": "New Delhi"
}
```

(e) Deleting employee using id

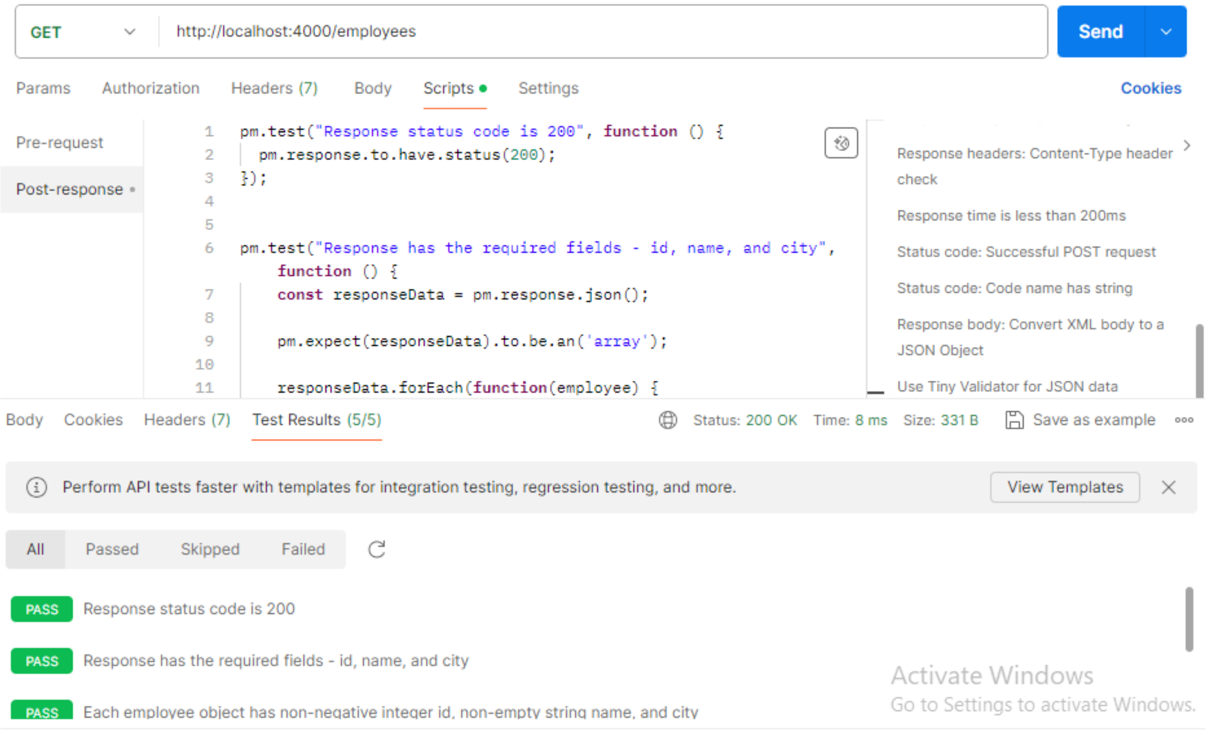


```
{
    "id": 3,
    "name": "Faizan Anwer",
    "city": "New Delhi"
}
```

Status: 200 OK Time: 242 ms Size: 286 B

```
[
    {
        "id": 3,
        "name": "Faizan Anwer",
        "city": "New Delhi"
    }
]
```

(f) Verification


GET
http://localhost:4000/employees
Send
Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies
Pre-request
Post-response

```
pm.test("Response status code is 200", function () {
  pm.response.to.have.status(200);
});


pm.test("Response has the required fields - id, name, and city",
    function () {
    const responseData = pm.response.json();

    pm.expect(responseData).to.be.an('array');

    responseData.forEach(function(employee) {
```

Response headers: Content-Type header check
Response time is less than 200ms
Status code: Successful POST request
Status code: Code name has string
Response body: Convert XML body to a JSON Object
Use Tiny Validator for JSON data
Body
Cookies
Headers (7)
Test Results (5/5)
Save as example

Status: 200 OK Time: 8 ms Size: 331 B

Perform API tests faster with templates for integration testing, regression testing, and more. View Templates

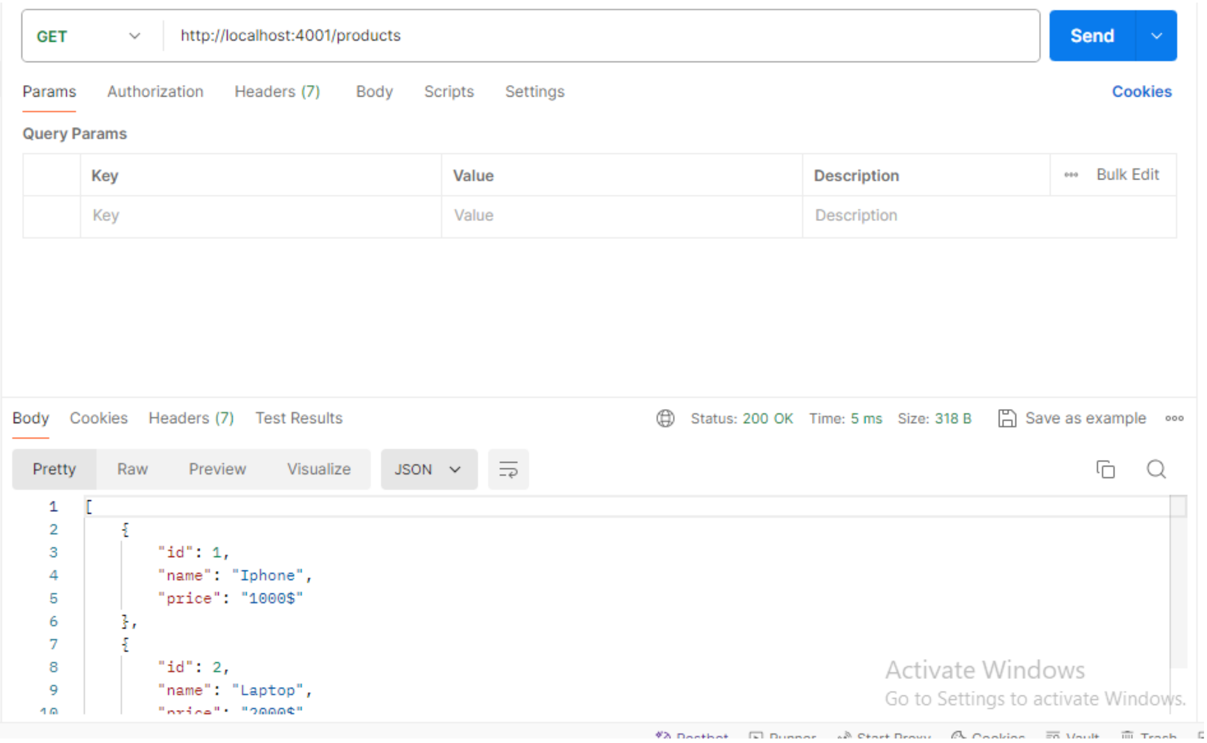
PASS Response status code is 200

PASS Response has the required fields - id, name, and city

PASS Each employee object has non-negative integer id, non-empty string name, and city

2. Products

(a) Get all products

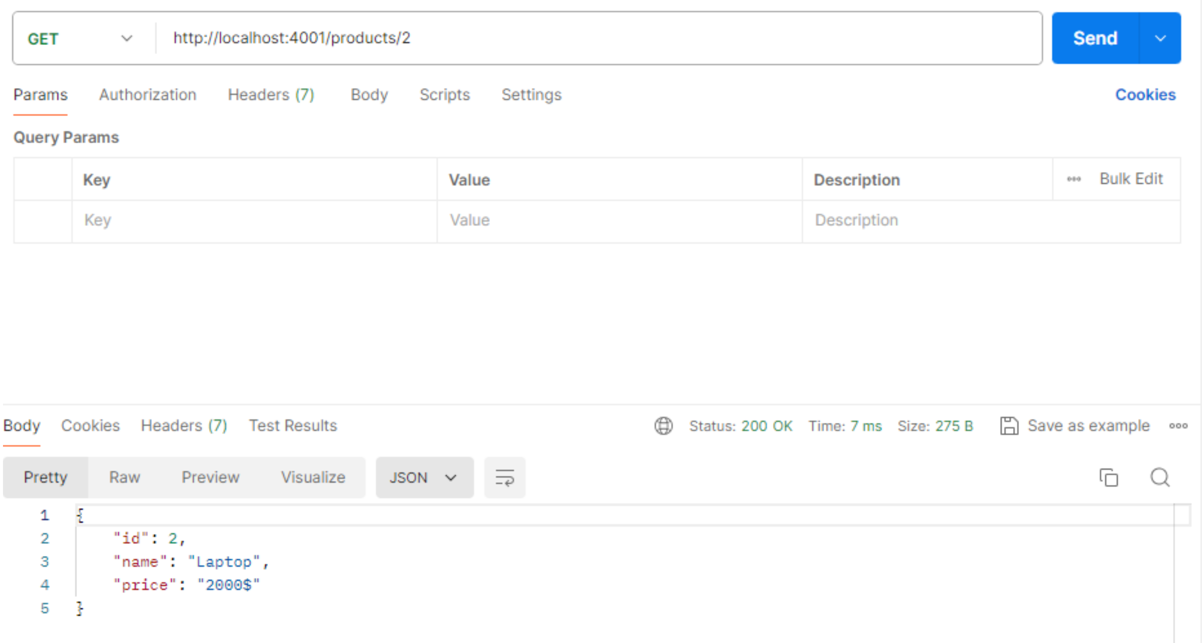


GET http://localhost:4001/products

Status: 200 OK Time: 5 ms Size: 318 B

```
[
    {
        "id": 1,
        "name": "Iphone",
        "price": "1000$"
    },
    {
        "id": 2,
        "name": "Laptop",
        "price": "2000$"
```

(b) Get a specific product

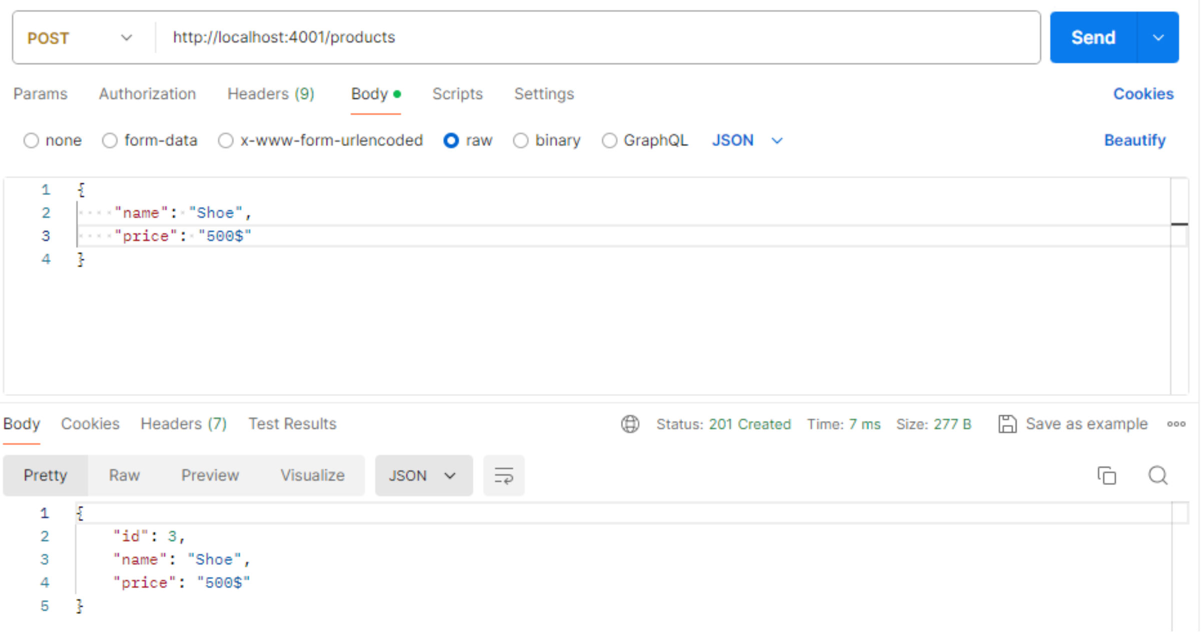


GET http://localhost:4001/products/2

Status: 200 OK Time: 7 ms Size: 275 B

```
{
    "id": 2,
    "name": "Laptop",
    "price": "2000$"
}
```

(c) Create a new product



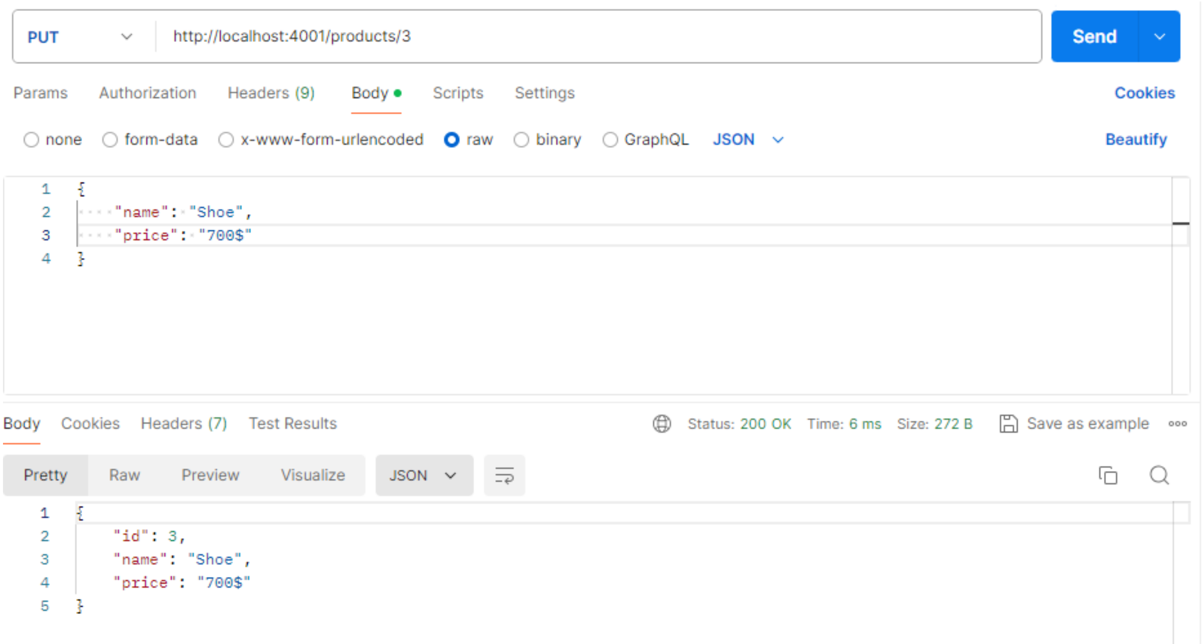
POST http://localhost:4001/products

```
{
    "name": "Shoe",
    "price": "500$"
}
```

Status: 201 Created Time: 7 ms Size: 277 B

```
{
    "id": 3,
    "name": "Shoe",
    "price": "500$"
}
```

(d) Update a product using id



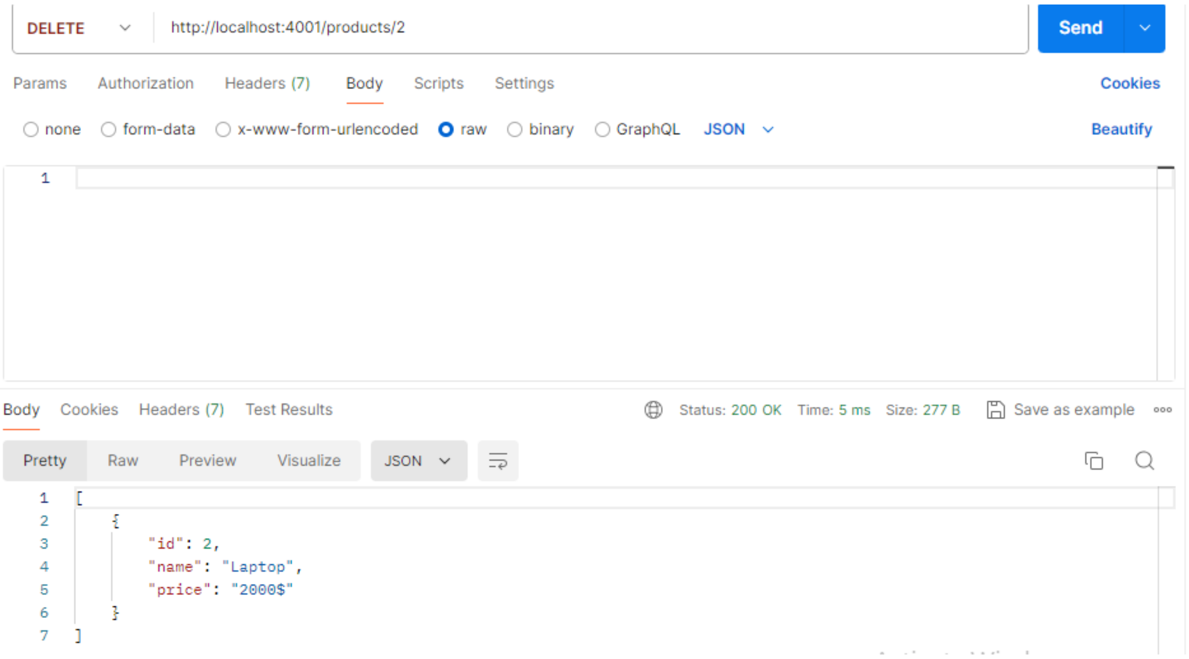
PUT http://localhost:4001/products/3

```
{
    "name": "Shoe",
    "price": "700$"
}
```

Status: 200 OK Time: 6 ms Size: 272 B

```
{
    "id": 3,
    "name": "Shoe",
    "price": "700$"
}
```

(e) Deleting a specific product

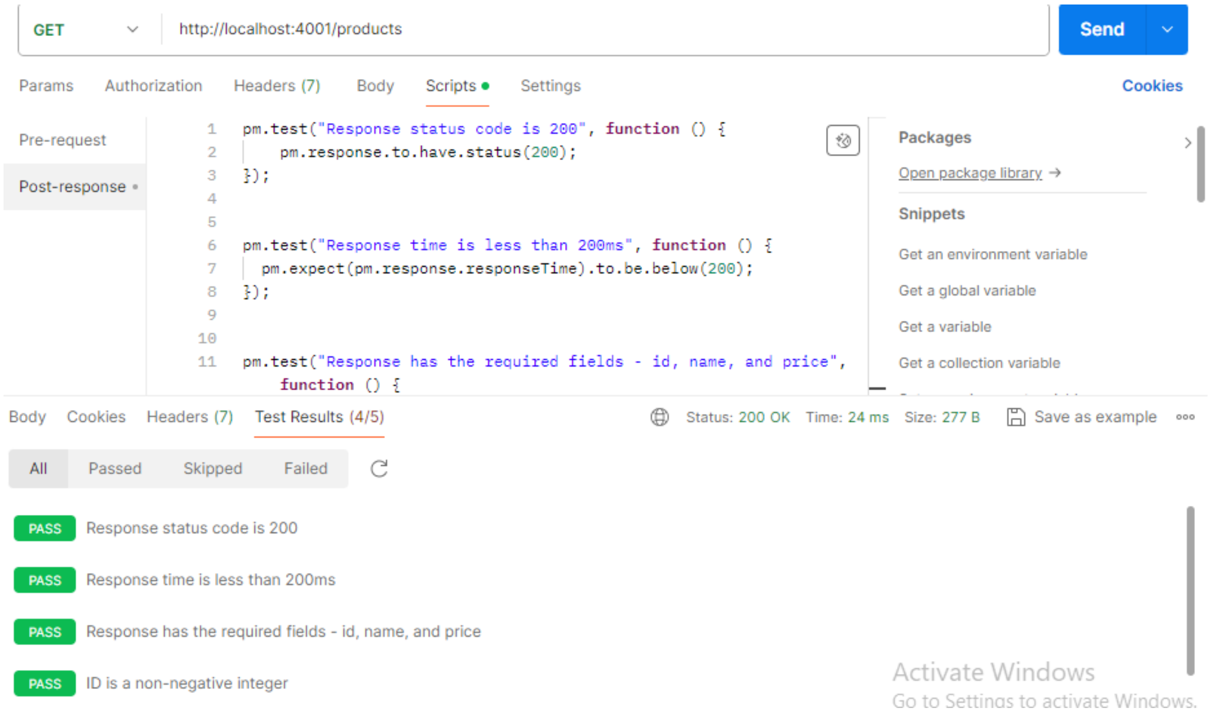


DELETE http://localhost:4001/products/2

Status: 200 OK Time: 5 ms Size: 277 B

```
[
    {
        "id": 2,
        "name": "Laptop",
        "price": "2000$"
    }
]
```

(f) Verification



GET http://localhost:4001/products

```
pm.test("Response status code is 200", function () {
    pm.response.to.have.status(200);
});


pm.test("Response time is less than 200ms", function () {
  pm.expect(pm.response.responseTime).to.be.below(200);
});


pm.test("Response has the required fields - id, name, and price",
    function () {
```

Status: 200 OK Time: 24 ms Size: 277 B

PASS Response status code is 200

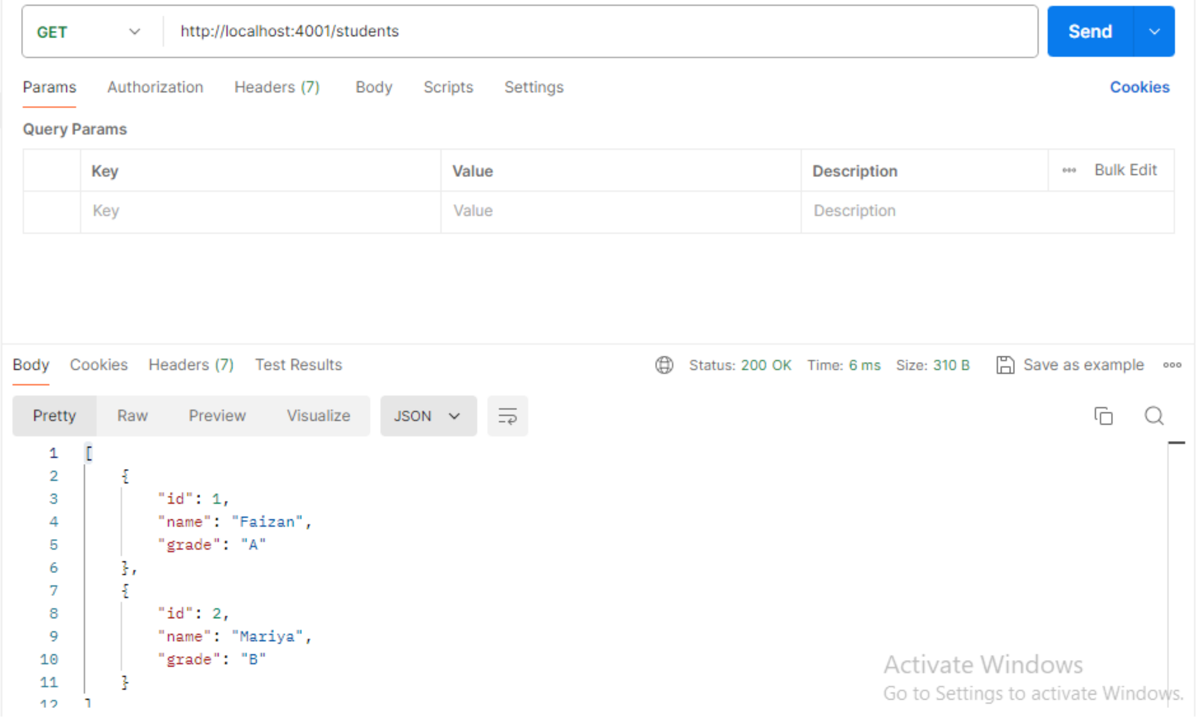
PASS Response time is less than 200ms

PASS Response has the required fields - id, name, and price

PASS ID is a non-negative integer

3. Students

(a) Get all students

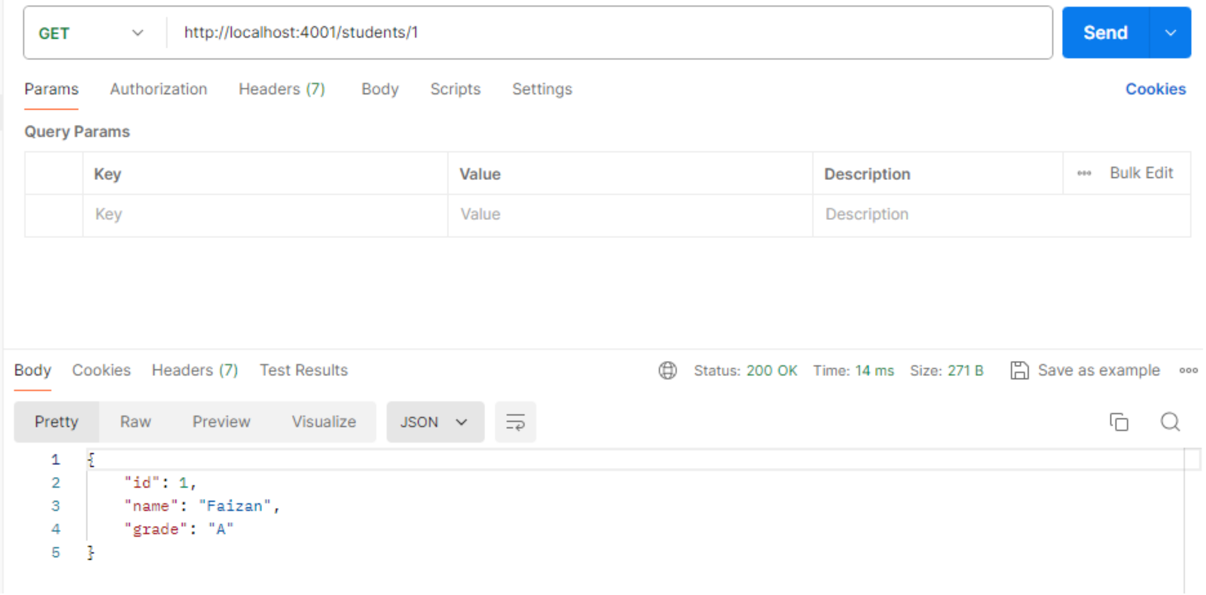


GET http://localhost:4001/students

Status: 200 OK Time: 6 ms Size: 310 B

```
[
    {
        "id": 1,
        "name": "Faizan",
        "grade": "A"
    },
    {
        "id": 2,
        "name": "Mariya",
        "grade": "B"
    }
]
```

(b) Get a specific student

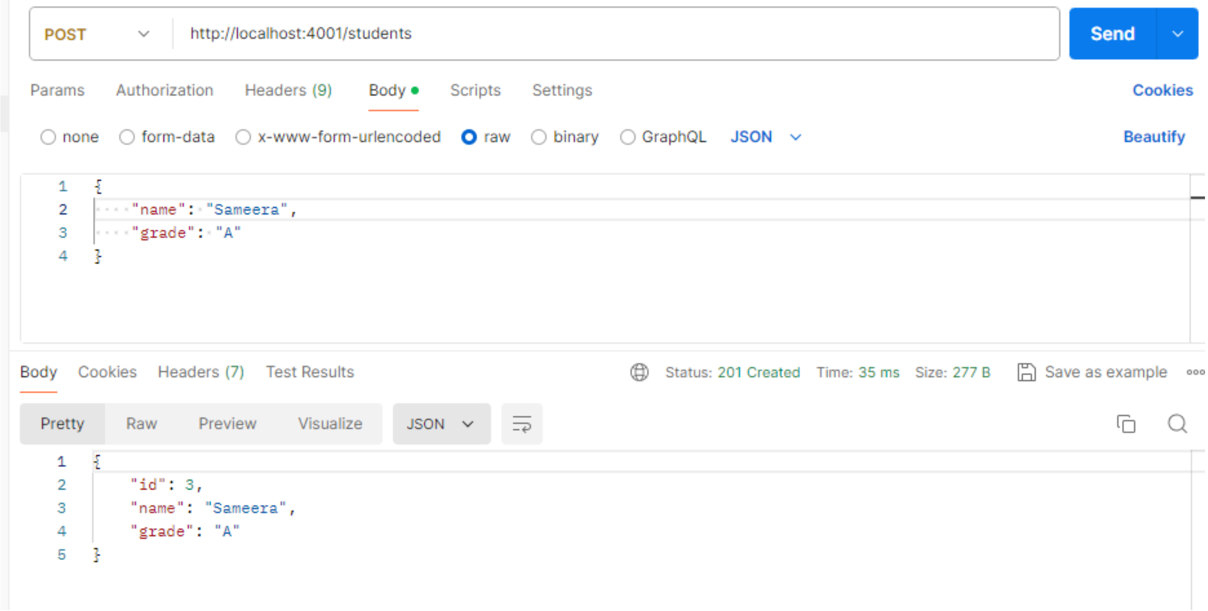


GET http://localhost:4001/students/1

Status: 200 OK Time: 14 ms Size: 271 B

```
{
    "id": 1,
    "name": "Faizan",
    "grade": "A"
}
```

(c) Creating a new student



POST http://localhost:4001/students

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

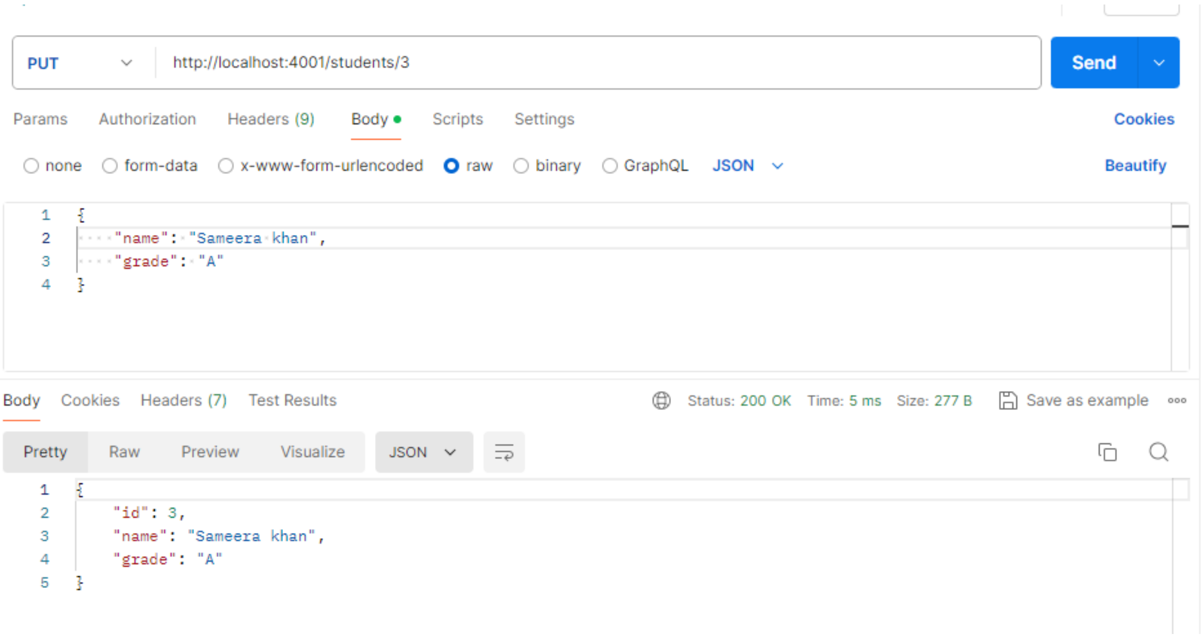
```
{
    "name": "Sameera",
    "grade": "A"
}
```

Status: 201 Created Time: 35 ms Size: 277 B

JSON

```
{
    "id": 3,
    "name": "Sameera",
    "grade": "A"
}
```

(d) Update a student using id



PUT http://localhost:4001/students/3

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

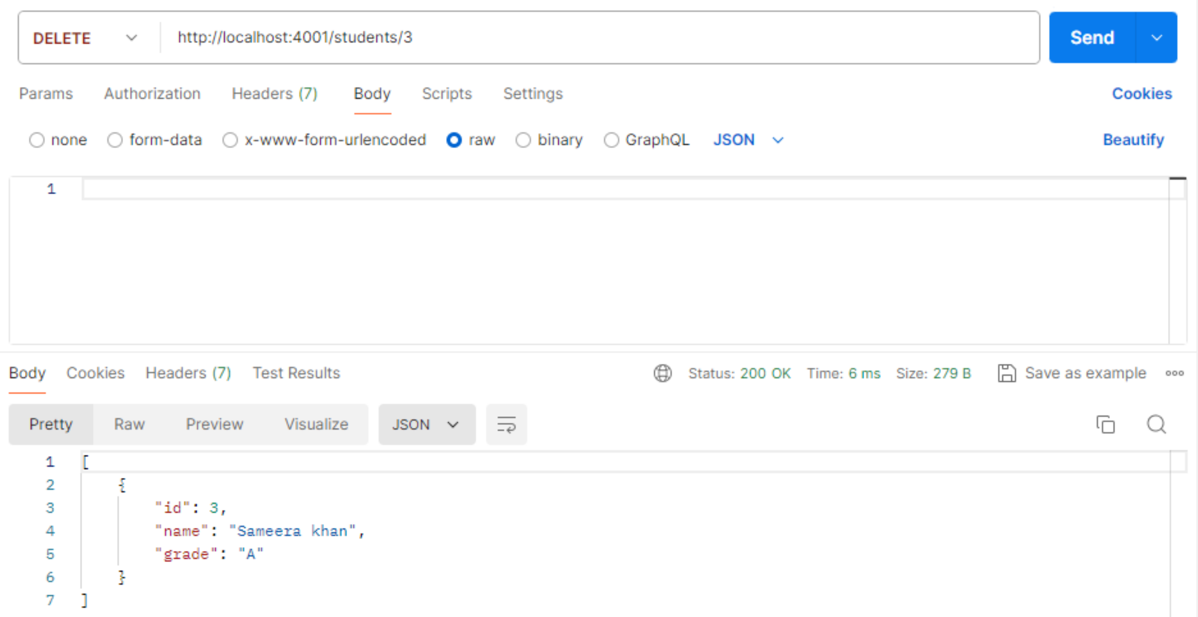
```
{
    "name": "Sameera khan",
    "grade": "A"
}
```

Status: 200 OK Time: 5 ms Size: 277 B

JSON

```
{
    "id": 3,
    "name": "Sameera khan",
    "grade": "A"
}
```

(e) Deleting a student using id

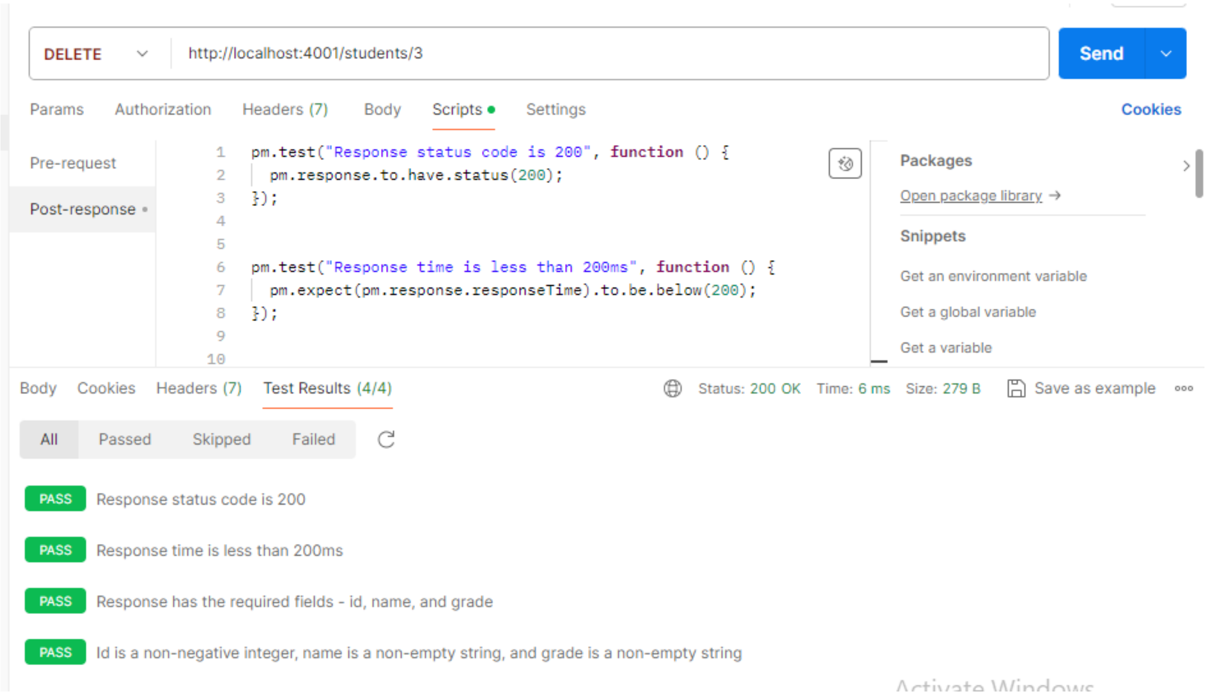


DELETE http://localhost:4001/students/3

Status: 200 OK Time: 6 ms Size: 279 B

```
[
    {
        "id": 3,
        "name": "Sameera khan",
        "grade": "A"
    }
]
```

(f) Verification



DELETE http://localhost:4001/students/3

```
pm.test("Response status code is 200", function () {
  pm.response.to.have.status(200);
});


pm.test("Response time is less than 200ms", function () {
  pm.expect(pm.response.responseTime).to.be.below(200);
});
```

Status: 200 OK Time: 6 ms Size: 279 B

PASS Response status code is 200

PASS Response time is less than 200ms

PASS Response has the required fields - id, name, and grade

PASS Id is a non-negative integer, name is a non-empty string, and grade is a non-empty string