

Capstone Project: Telecom Domain Project Requirements

Project Overview

Project Name: API Testing for Contact List Application

Domain: Telecom

Project Description: This project aims to perform comprehensive testing of the APIs provided by the <https://thinking-tester-contact-list.herokuapp.com/> application.

The testing will cover various endpoints to ensure the functionality and reliability of the APIs. **Objectives:**

- Verify the correctness and consistency of API responses.
- Validate authentication and authorization mechanisms (if applicable).
- Document test cases and results for future reference.

Instructions: Follow the API document and test all the test cases in Postman apply the chai library assertion and prepare the script in RestAssured & generate the report.

Testing Flow: Add User → Get user profile → Update user → Login user → Add Contact → Get contact list → Get contact → Update full contact → Update partial contact → Logout User

PostMan :

//AddNewUser 1 :

Post : <https://thinking-tester-contact-list.herokuapp.com/users/me>

Headers : Key : Content -Type Value -application /json

Body :

```
{  
  "firstName": "Nagaraj",  
  "lastName": "M Naik",  
  "email": "nagaraj.unique48@example.com",  
  "password": "NagarajNaik@2025"  
}
```

Scripts

/ Test for status code should be 201

```
pm.test("Test for status code should be 201", () => {  
    pm.expect(pm.response.code).to.eql(201);  
});
```

// Get JSON Response

```
let jsonData = pm.response.json();  
console.log("Response Body:", jsonData);
```

// Test for first name

```
pm.test("Test for first name should be Nagaraj", () => {  
    pm.expect(jsonData.user.firstName).to.eql("Nagaraj");  
});
```

// Test for last name

```
pm.test("Test for last name should be M Naik", () => {  
    pm.expect(jsonData.user.lastName).to.eql("M Naik");  
});
```

// Test for email

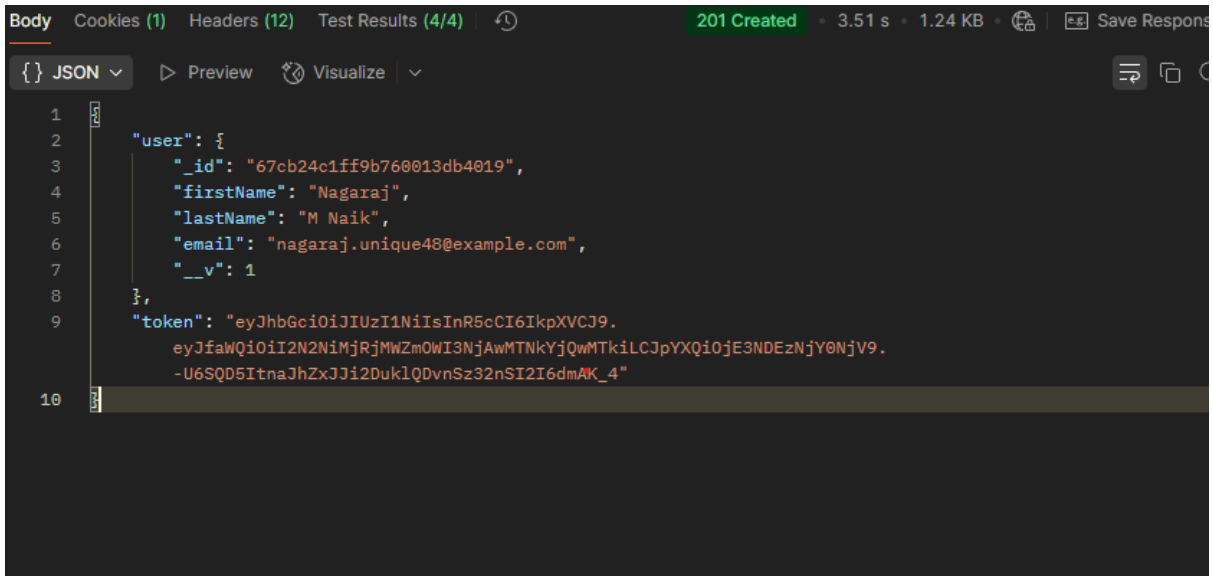
```
pm.test("Test for email should be nagaraj.unique48@example.com", () => {  
    pm.expect(jsonData.user.email).to.eql("nagaraj.unique48@example.com");  
});
```

// Set environment variables if user_id and token exist

```
if (jsonData.user && jsonData.user._id && jsonData.token) {  
    pm.environment.set("user_id", jsonData.user._id);  
    pm.environment.set("btoken", jsonData.token);  
    console.log("user_id & token set successfully");  
}
```

```
} else {  
    console.log("Error: user_id or token is missing in response");  
}
```

Response :



2> GetNewUser :

Get : <https://thinking-tester-contact-list.herokuapp.com/users/me>

Headers : Key : Authorization Value: Bearer{{btoken}}

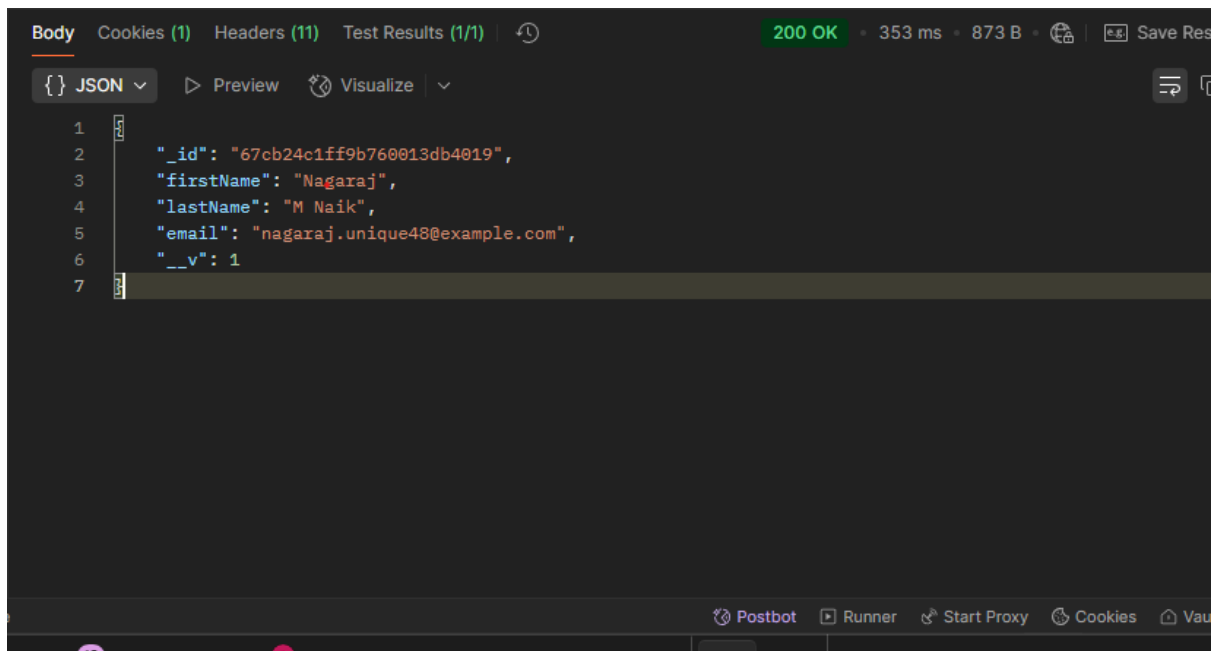
Scripts:

```
pm.test("Test for status code should be 200", () => {
    pm.expect(pm.response.code).toEqual(200);
});
```

```
let response = pm.response.json();
```

```
pm.environment.set("user_id", response._id);
```

Response :



3> PutNewUser :

PATCH: <https://thinking-tester-contact-list.herokuapp.com/users/me>

Headers :

Key: Authorization -Value -Bearer{{btoken}}

Key - Content-Type Value -application/json

Body : {

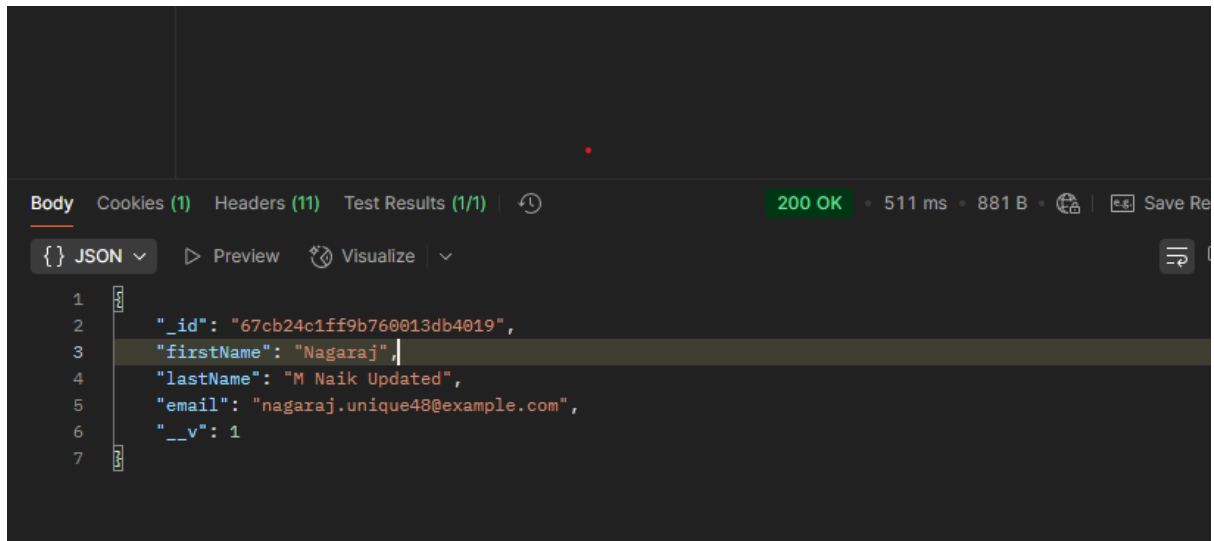
```
"firstName": "Nagaraj",  
"lastName": "M Naik Updated",  
"email": "nagaraj.unique2026@example.com",  
"password": "NagarajNaik@2025"
```

}

Scripts:

```
pm.test("Test for status code should be 200", () => {  
  pm.expect(pm.response.code).to.eql(200);  
});
```

Response :



4>LoginUser

Post : <https://thinking-tester-contact-list.herokuapp.com/users/login>

Headers : Key : Content-Type Value : application/json

Body :

```
pm.test("Status Code is 200", () => {  
  pm.response.to.have.status(200);  
});
```

```
let jsonData = pm.response.json();
```

```
pm.test("Token is present in response", () => {  
  pm.expect(jsonData).to.have.property("token");  
});
```

```
console.log("Auth Token:", jsonData.token);
```

```
pm.environment.set("Ltoken", jsonData.token);
```

Response:

```
{ } JSON ▾ ▶ Preview 🔗 Visualize ▾
1  {
2    "user": {
3      "_id": "67cb24c1ff9b760013db4019",
4      "firstName": "Nagaraj",
5      "lastName": "M Naik Updated",
6      "email": "nagaraj.unique48@example.com",
7      "__v": 2
8    },
9    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2N2NiMjRjMWZmOWI3NjAwMTNkYjQwMTkiLCJpYXQiOiE3NDEzNjY0ODB9.pS_dNWvXKt4nudBERa1LuB0y2P8nSuVxHPWh39fdb4I"
10 }
```

5> Add -Contact

Post : <https://thinking-tester-contact-list.herokuapp.com/contacts>

Headers :

Key : Authorization : Value : Bearer{{Ltoken}}

Key : Content-Type Value : Value : application/json

Body :

```
{
  "firstName": "Pavi",
  "lastName": "M Naik",
  "birthdate": "1997-01-01",
  "email": "pavi.unique@example.com",
  "phone": "9408868899",
  "street1": "2nd main road",
  "street2": "Bangalore",
  "city": "Jpnagar",
```

```
"stateProvince": "KA",  
"postalCode": "560001",  
"country": "India"  
}
```

Scripts :

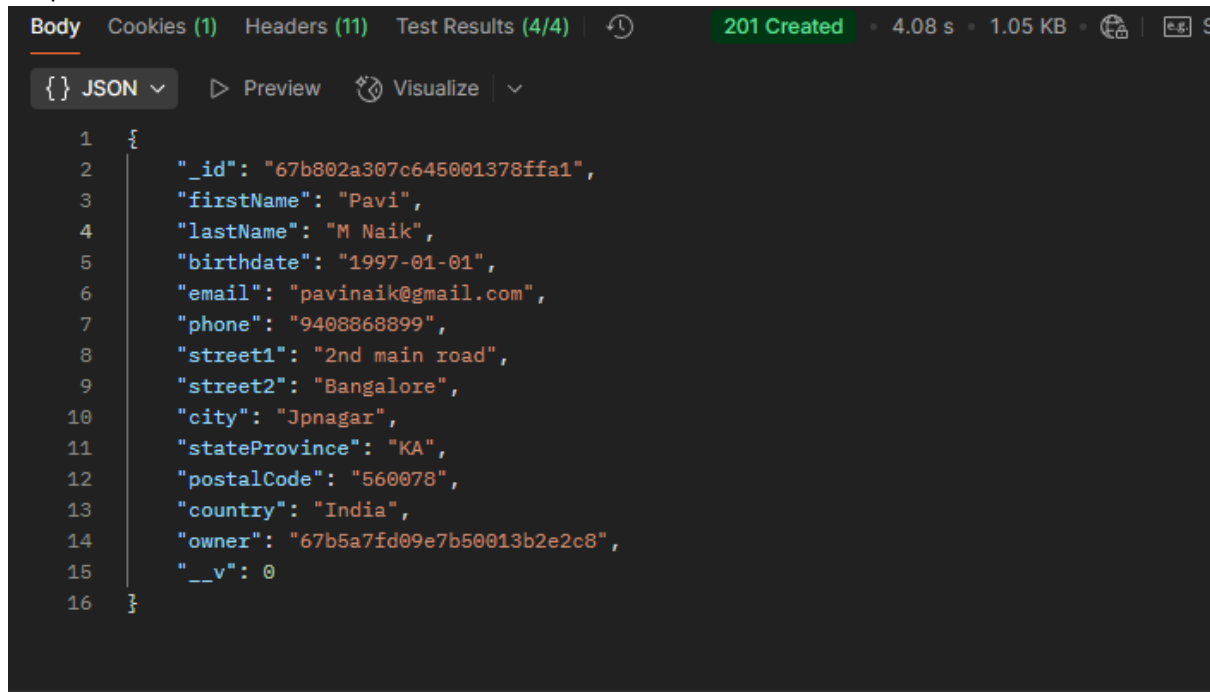
```
pm.test("Status Code is 201", function () {  
    pm.response.to.have.status(201);  
});
```

```
const res = pm.response.json();  
pm.test("firstName should be Pavi", () => {  
    pm.expect(res.firstName).to.eql("Pavi");  
});
```

```
pm.test("lastName should be M Naik", () => {  
    pm.expect(res.lastName).to.eql("M Naik");  
});
```

```
pm.test("Country Name should India", () => {  
    pm.expect(res.country).to.eql("India");  
});
```

Response:



6>Get ContactList

Get : <https://thinking-tester-contact-list.herokuapp.com/contacts>

Headers :

Key: Authorization Value: Bearer{{Ltoken}}

Key: Content-Type Value: application/json

Scripts:

```
pm.test("Status code is 200", () => {  
    pm.response.to.have.status(200);  
});
```

```
// Log response
```

```
let res = pm.response.json();
```

```
console.log("All contacts:", res);
```

```
if (Array.isArray(res) && res.length > 0) {
```

```
    pm.environment.set("contactId", res[0]._id);
```

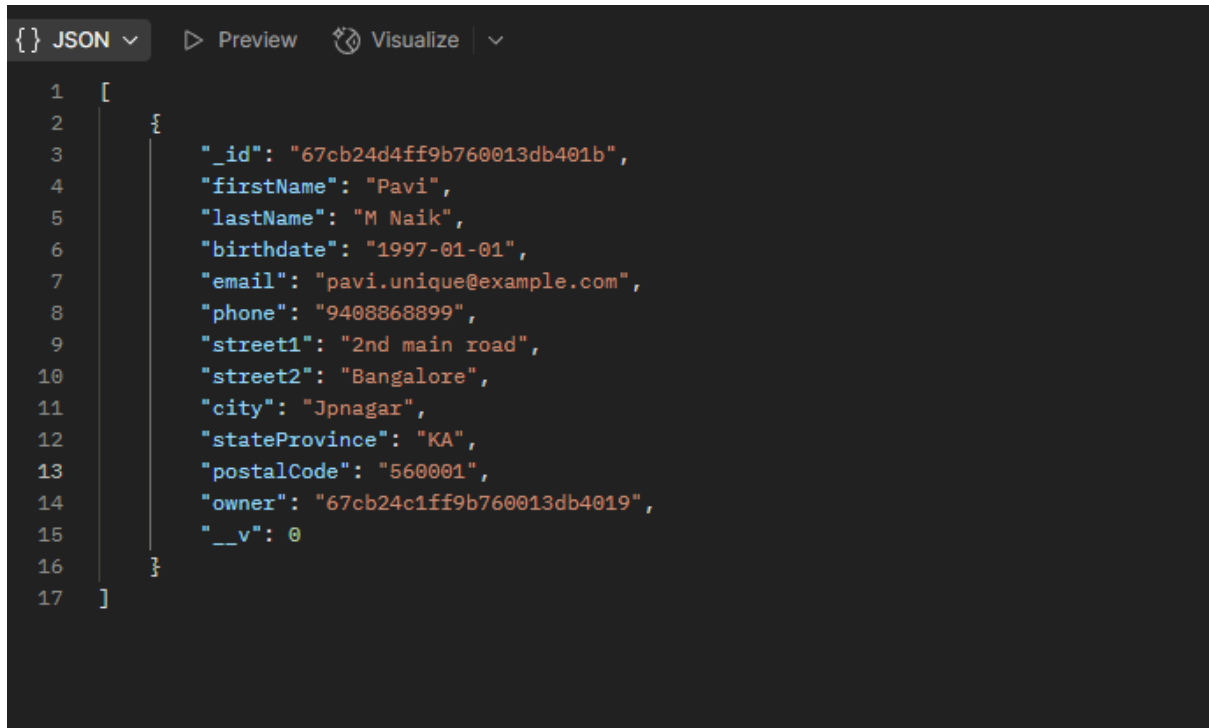


```

    console.log("contactId set to", res[0]._id);
  } else {
    console.log("No contacts found.");
  }
}

```

Response :



```

1  [
2    {
3      "_id": "67cb24d4ff9b760013db401b",
4      "firstName": "Pavi",
5      "lastName": "M Naik",
6      "birthdate": "1997-01-01",
7      "email": "pavi.unique@example.com",
8      "phone": "9408868899",
9      "street1": "2nd main road",
10     "street2": "Bangalore",
11     "city": "Jpnagar",
12     "stateProvince": "KA",
13     "postalCode": "560001",
14     "owner": "67cb24c1ff9b760013db4019",
15     "__v": 0
16   }
17 ]

```

07>Get Contact : Get : <https://thinking-tester-contact-list.herokuapp.com/contacts/{{contactId}}>

Haeders :

Key: Authorization : Value : Bearer{{Ltoken}}

Scripts :

```

pm.test("Status code is 200", () => {
  pm.response.to.have.status(200);
});

```

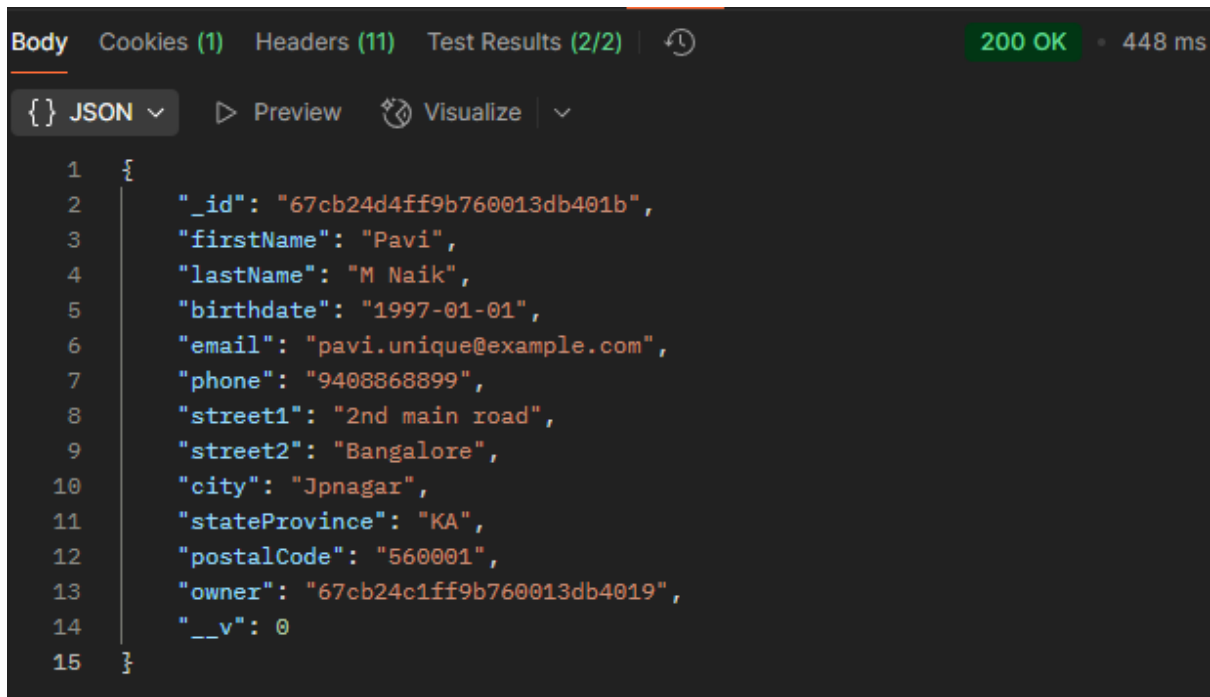
// Parse response and validate

```
let res = pm.response.json();
```

```
console.log("Single contact:", res);
```

```
pm.test("firstName should not be empty", () => {  
    pm.expect(res.firstName).to.exist;  
});
```

Response :



08>Update Contact:

Put : https://thinking-tester-contact-list.herokuapp.com/contacts/{{user_id2}}

Headers

Key: Authorization : Value : Bearer{{Ltoken}}

Key : Content-Type : Value : application/json

Body:

```
{  
    "firstName": "Pavi Updated",  
    "lastName": "M Naik",  
    "email": "pavi.unique@example.com",  
    "phone": "9998887770"  
}
```

Scripts :

```
pm.test("Status code is 200", () => {  
    pm.response.to.have.status(200);  
});
```

```
// Parse and log response
```

```
let res = pm.response.json();
```

```
console.log("Updated contact:", res);
```

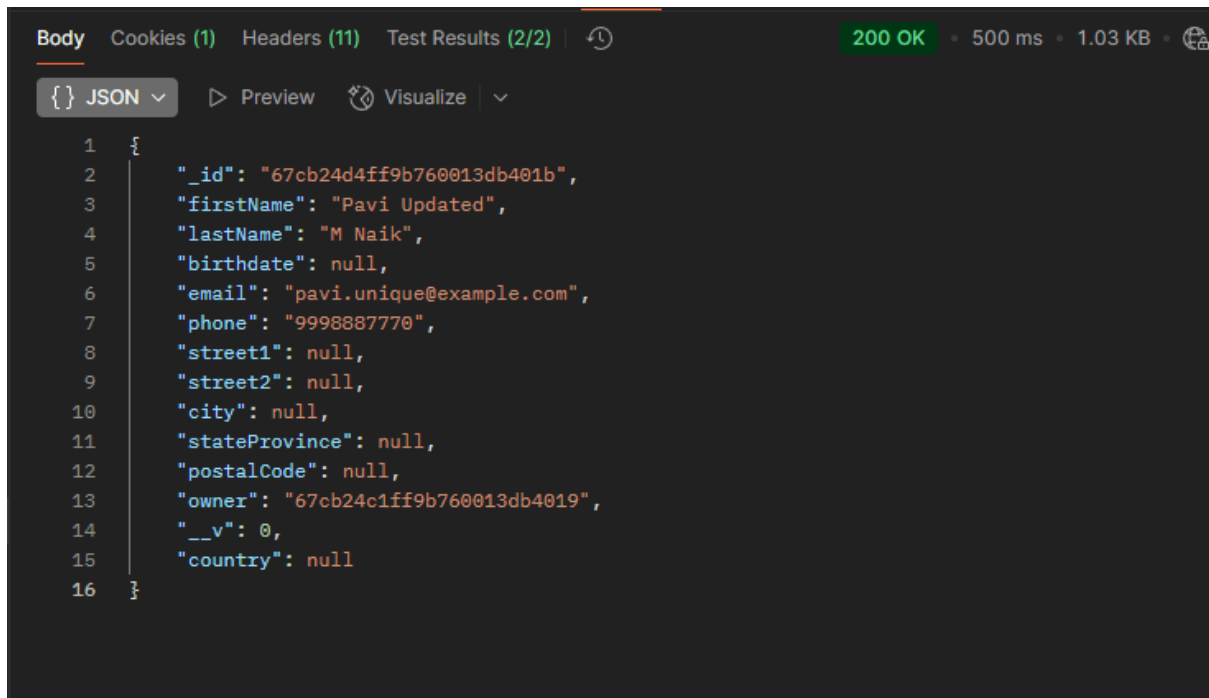
```
// Validate updated firstName
```

```
pm.test("Contact firstName is updated", () => {
```

```
    pm.expect(res.firstName).to.eql("Pavi Updated");
```

```
});
```

Response:



09>Update Contact :

Patch : https://thinking-tester-contact-list.herokuapp.com/contacts/{{user_id2}}

Headers :

Key: Authorization Value: Bearer{{Ltoken}}

Key : Authorization : Value : application/json

Scripts :

// Check status code is 200

```
pm.test("Status code is 200", () => {  
  pm.response.to.have.status(200);  
});
```

// Parse and log response

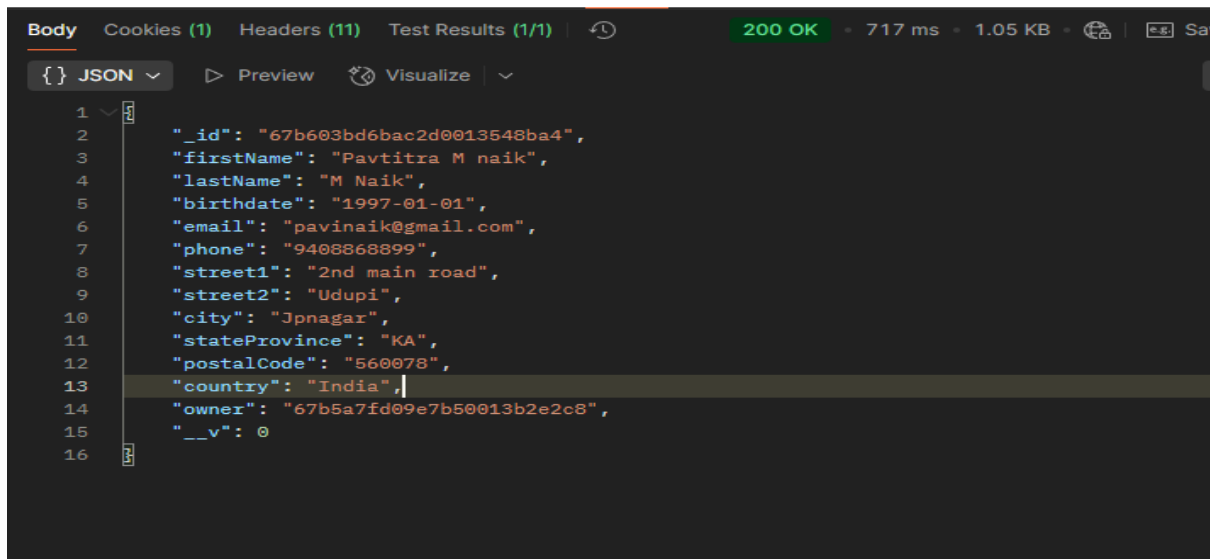
```
let res = pm.response.json();  
console.log("Patched contact:", res);
```

// Validate partial update of firstName

```
pm.test("Contact firstName is updated with PATCH", () => {
```

```
pm.expect(res.firstName).to.eql("Pavi Patched");
});
```

Response :



10>Logout :

Post : <https://thinking-tester-contact-list.herokuapp.com/users/logout>

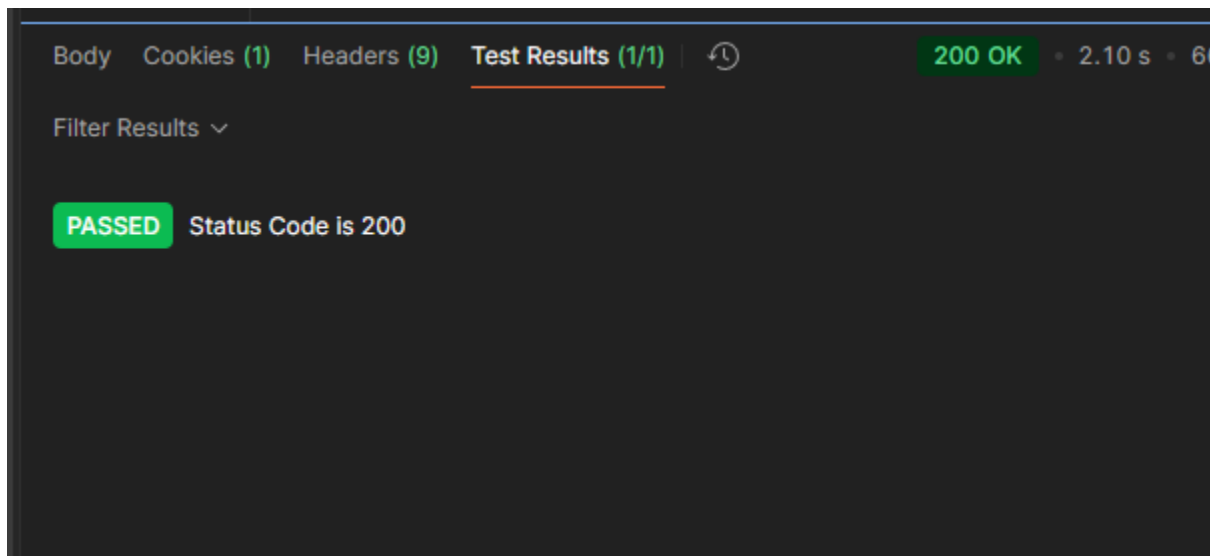
Headers :

Key: Authorization Value : Bearer{{Ltoken}}

Scripts :

```
pm.test("Status code is 200", () => {
  pm.response.to.have.status(200);
});
```

```
pm.environment.unset("Ltoken");
pm.environment.unset("contactId");
console.log("Logged out, environment variables cleared.");
```



Running Command :

```
newman run "Telecom Domain_2025.postman_collection.json" -e  
"Capstone_project.postman_environment.json" -r html --reporter-html-export  
"Telecom_Domain_2025_Report.html"
```