Batch: D-2	**Roll No.:** 16010122151
Experiment / assignment / tutorial No.____	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

Title: Create a RESTful API server in Express and Node.js. Implementation + Testing application u postman/Thurderclient

AIM: Create a RESTful API server in Express and Node.js. Implementation + Testing application using postman/Thurderclient

Problem Definition:

Resources used:

Expected OUTCOME of Experiment:

CO 3: Test the concepts and components of various front-end, back-end web app development technologies & frameworks using web development tools.

Books/ Journals/ Websites referred:

1. Shelly Powers Learning Node O' Reilly 2 nd Edition, 2016.

Pre Lab/ Prior Concepts:

Write details about the following content

- Testing in POSTMAN

Methodology:

1. Setup Development Environment:

   Install Node.js and initialize a new project using npm init.
   Install Express.js and nodemon as a development dependency.
   Optionally install Mongoose if a database is involved.

2. Create RESTful API:

   Build a simple REST API that supports CRUD operations.
   Create routes for each operation using appropriate HTTP methods (GET, POST, PUT, DELETE).

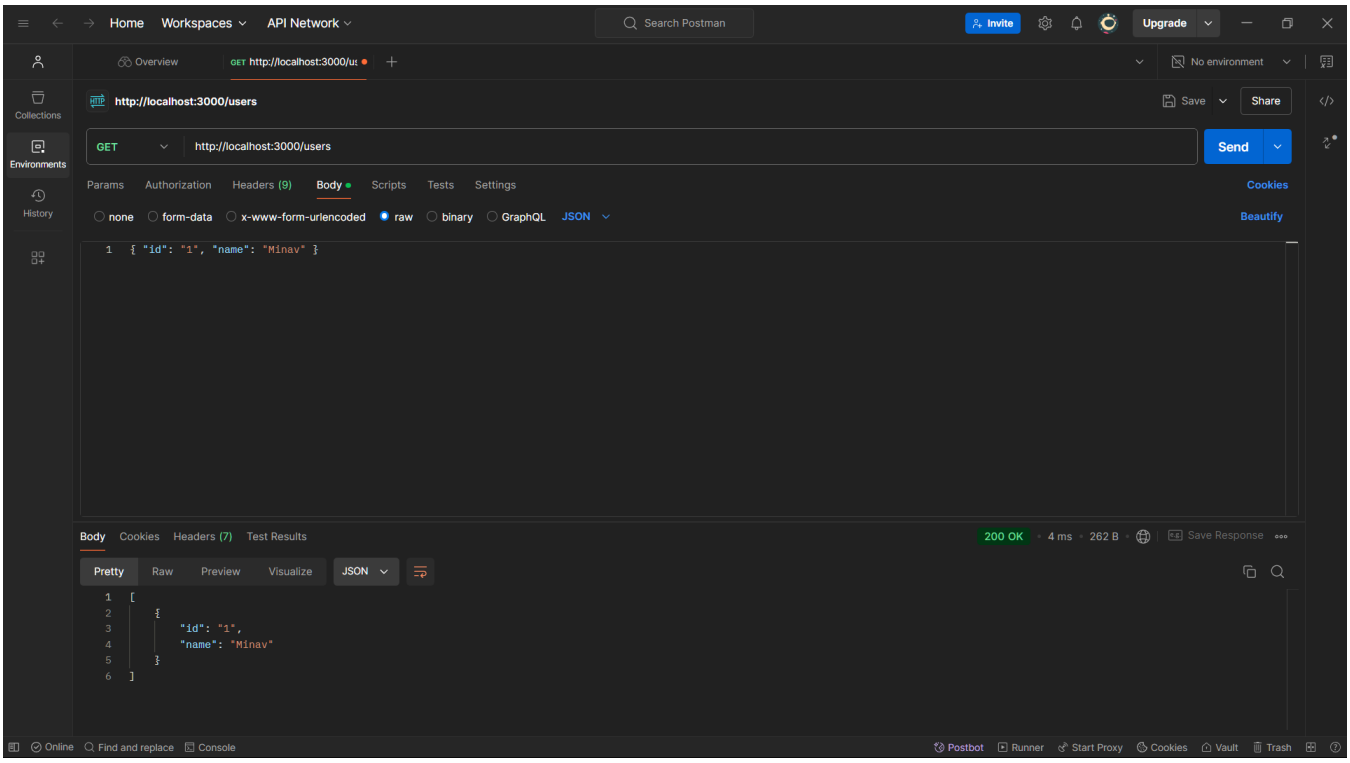
3. Test API Endpoints:

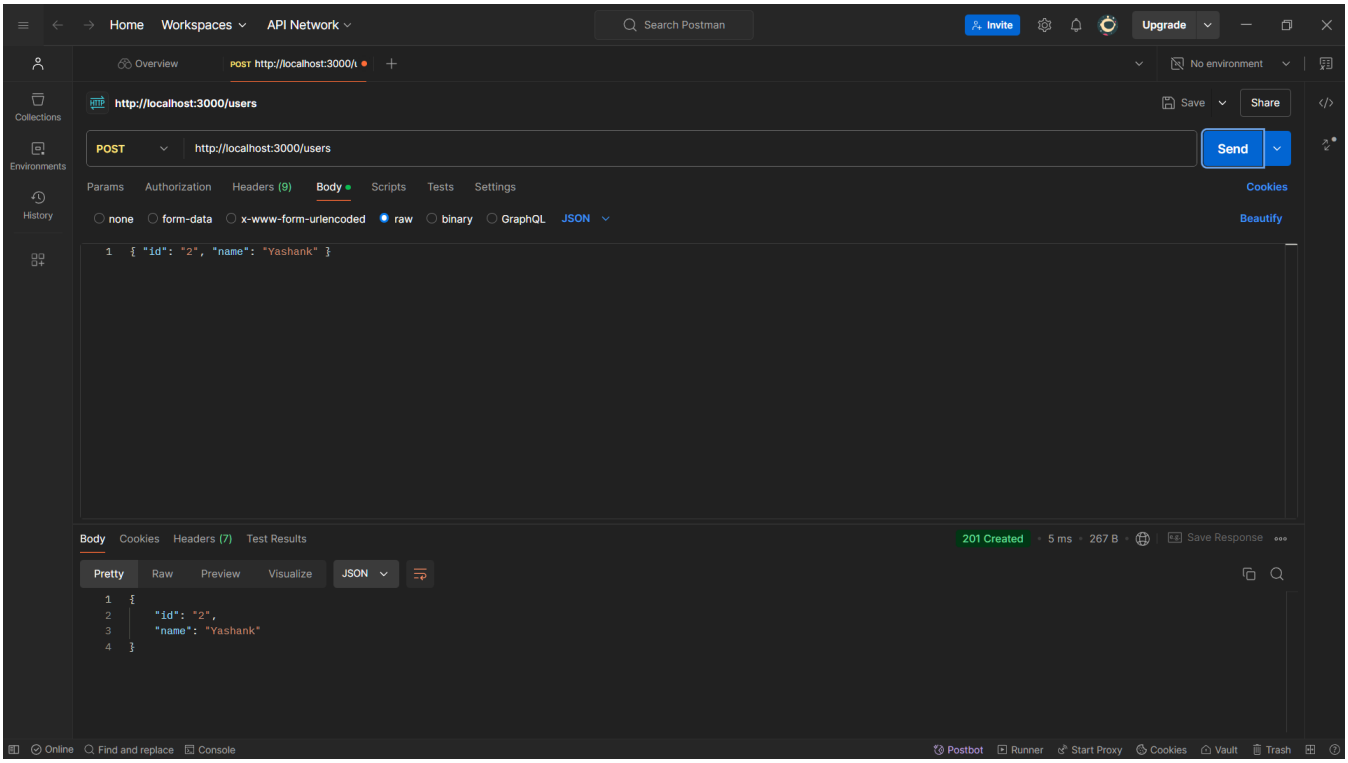
   Use Postman or Thunderclient to test API endpoints.
   Check for the correct response for each HTTP request and validate error handling.

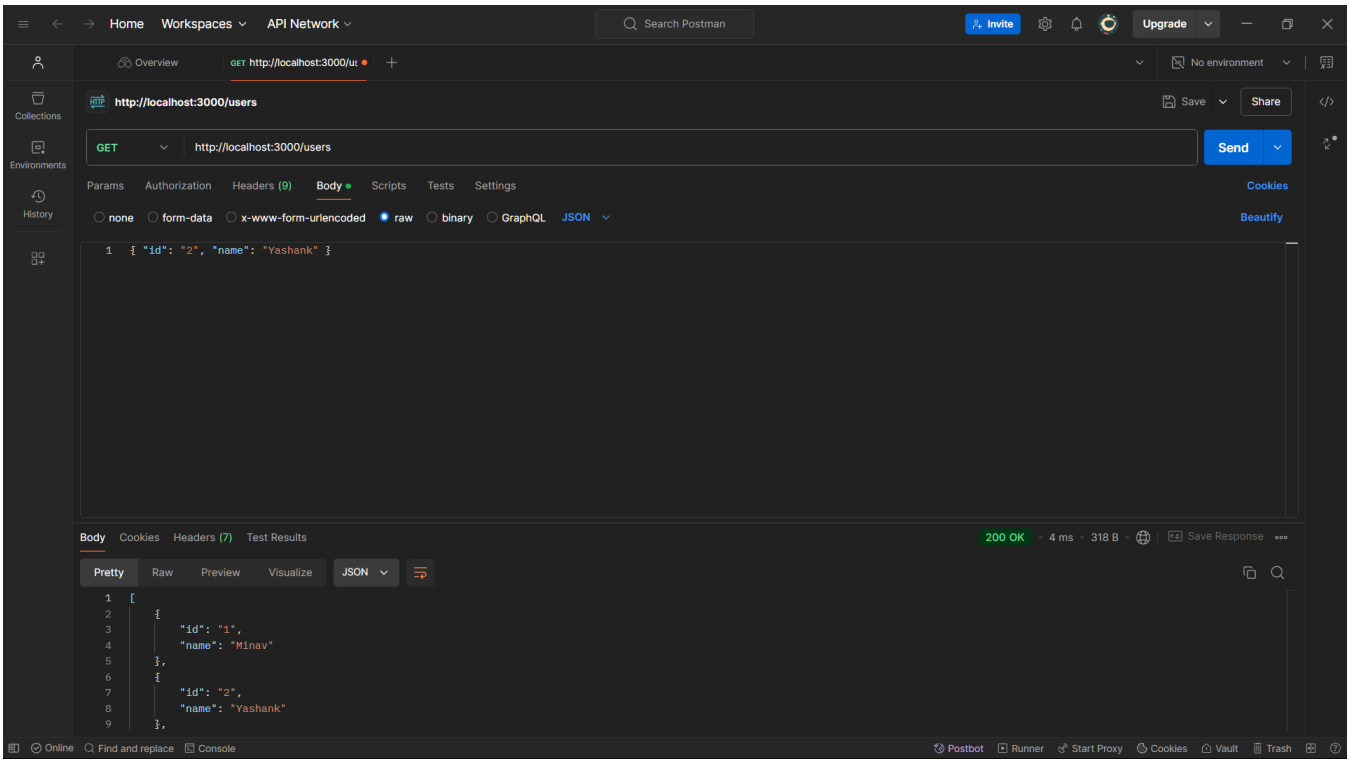
Implementation Details:

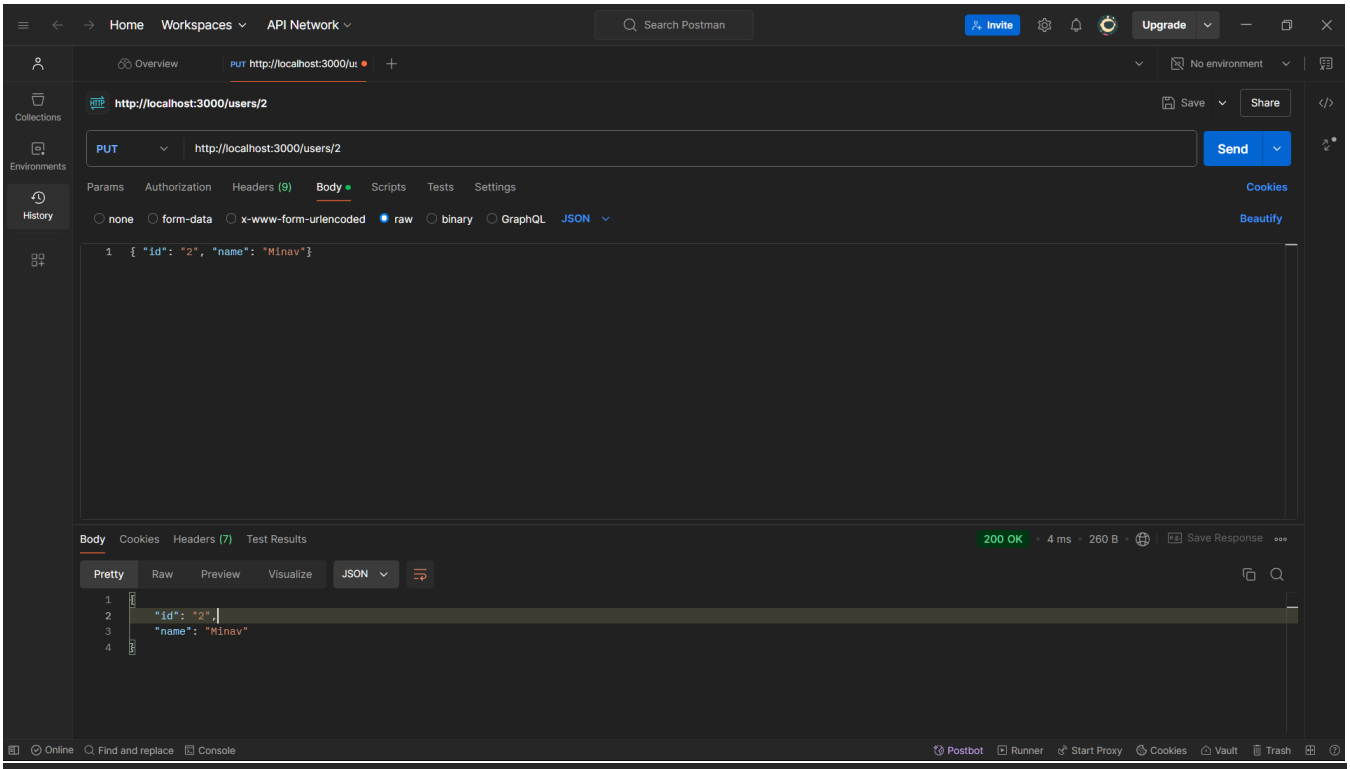
```
const express = require('express');
const app = express();
app.use(express.json());

let users = [];
app.get('/users', (req, res) => {
    res.status(200).json(users);
});
app.post('/users', (req, res) => {
    const newUser = req.body;
    users.push(newUser);
    res.status(201).json(newUser);
});
app.put('/users/:id', (req, res) => {
    const { id } = req.params;
    const updatedUser = req.body;
    users = users.map(user => user.id === id ? updatedUser : user);
    res.status(200).json(updatedUser);
});
app.delete('/users/:id', (req, res) => {
    const { id } = req.params;
    users = users.filter(user => user.id !== id);
    res.status(204).send();
});
app.listen(3000, () => {
    console.log('Server is running on port 3000');
});
```

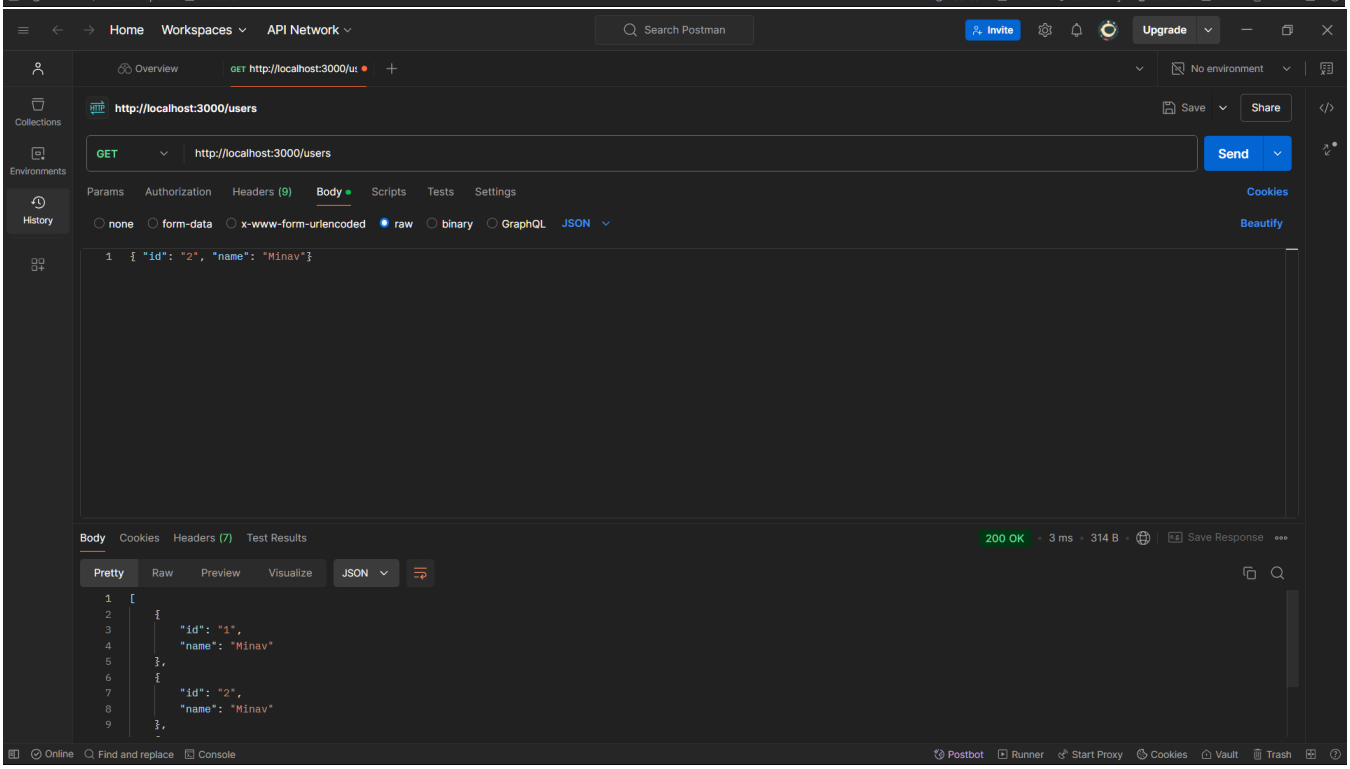
Output:

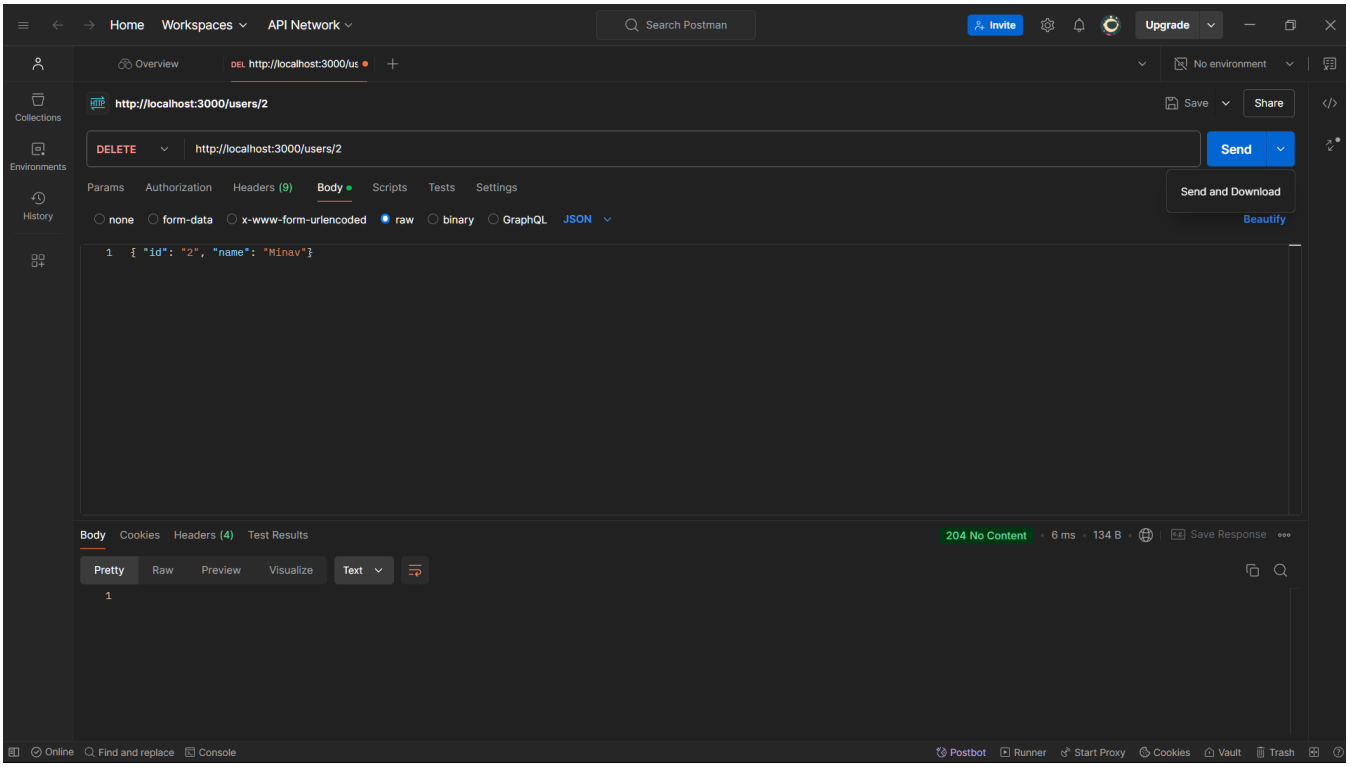

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
GET http://localhost:3000/us
No environment
Collections
Environments
History
http://localhost:3000/users
Save
Share
GET
http://localhost:3000/users
Send
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{ "id": "1", "name": "Minav" }
Body
Cookies
Headers (7)
Test Results
200 OK
4 ms
262 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": "1",
"name": "Minav"
}
]
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

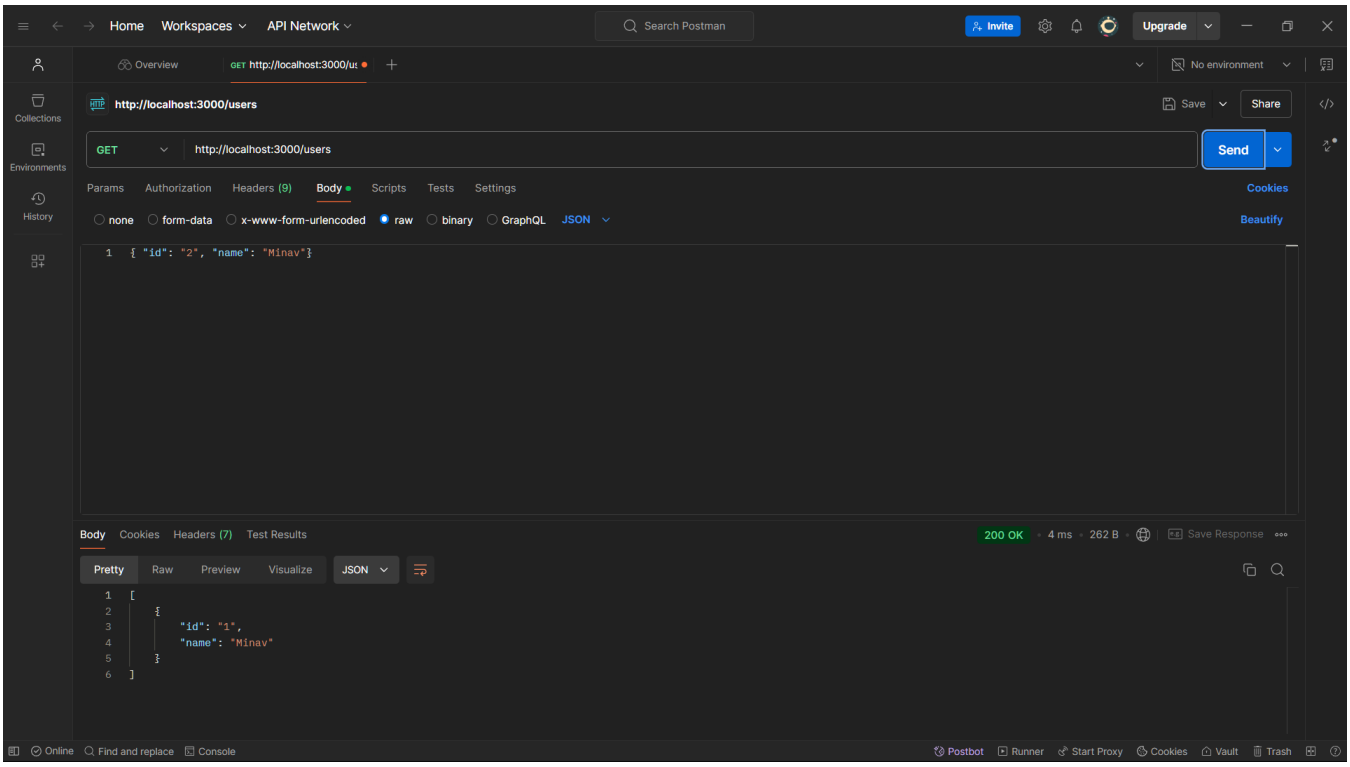

Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
POST http://localhost:3000/
No environment
http://localhost:3000/users
Save
Share
POST
http://localhost:3000/users
Send
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{ "id": "2", "name": "Yashank" }
Body
Cookies
Headers (7)
Test Results
201 Created
5 ms
267 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"id": "2",
"name": "Yashank"
}
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
GET http://localhost:3000/u
No environment
http://localhost:3000/users
Save
Share
GET
http://localhost:3000/users
Send
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{ "id": "2", "name": "Yashank" }
Body
Cookies
Headers (7)
Test Results
200 OK
4 ms
318 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": "1",
"name": "Minav"
},
{
"id": "2",
"name": "Yashank"
},
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
PUT http://localhost:3000/u
No environment
Collections
Environments
History
http://localhost:3000/users/2
Save
Share
PUT
http://localhost:3000/users/2
Send
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{ "id": "2", "name": "Minav"}
Body
Cookies
Headers (7)
Test Results
200 OK
4 ms
260 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"id": "2",
"name": "Minav"
}
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
GET http://localhost:3000/u
No environment
Collections
Environments
History
http://localhost:3000/users
Save
Share
GET
http://localhost:3000/users
Send
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{ "id": "2", "name": "Minav"}
Body
Cookies
Headers (7)
Test Results
200 OK
3 ms
314 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": "1",
"name": "Minav"
},
{
"id": "2",
"name": "Minav"
},
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
DEL http://localhost:3000/us
No environment
http://localhost:3000/users/2
Save
Share
DELETE
http://localhost:3000/users/2
Send
Send and Download
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{ "id": "2", "name": "Minav"}
Body
Cookies
Headers (4)
Test Results
204 No Content
6 ms
134 B
Save Response
Pretty
Raw
Preview
Visualize
Text
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Overview
GET http://localhost:3000/us
No environment
http://localhost:3000/users
Save
Share
GET
http://localhost:3000/users
Send
Params
Authorization
Headers (9)
Body
Scripts
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{ "id": "2", "name": "Minav"}
Body
Cookies
Headers (7)
Test Results
200 OK
4 ms
262 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": "1",
"name": "Minav"
}
]
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

Steps for execution:

- Install dependencies using `npm install`.
- Run the server using `npm start`.
- Open Postman or Thunderclient and test the API routes using HTTP methods like GET, POST, PUT, DELETE.
- Check for expected responses and status codes for each operation.

Conclusion:

In this experiment, we successfully created a RESTful API server using Express.js and Node.js. We implemented CRUD operations and tested the API endpoints using Postman/Thunderclient. This experiment provided insights into how to build and test REST APIs, ensuring that the server correctly handles various HTTP requests and returns appropriate responses.