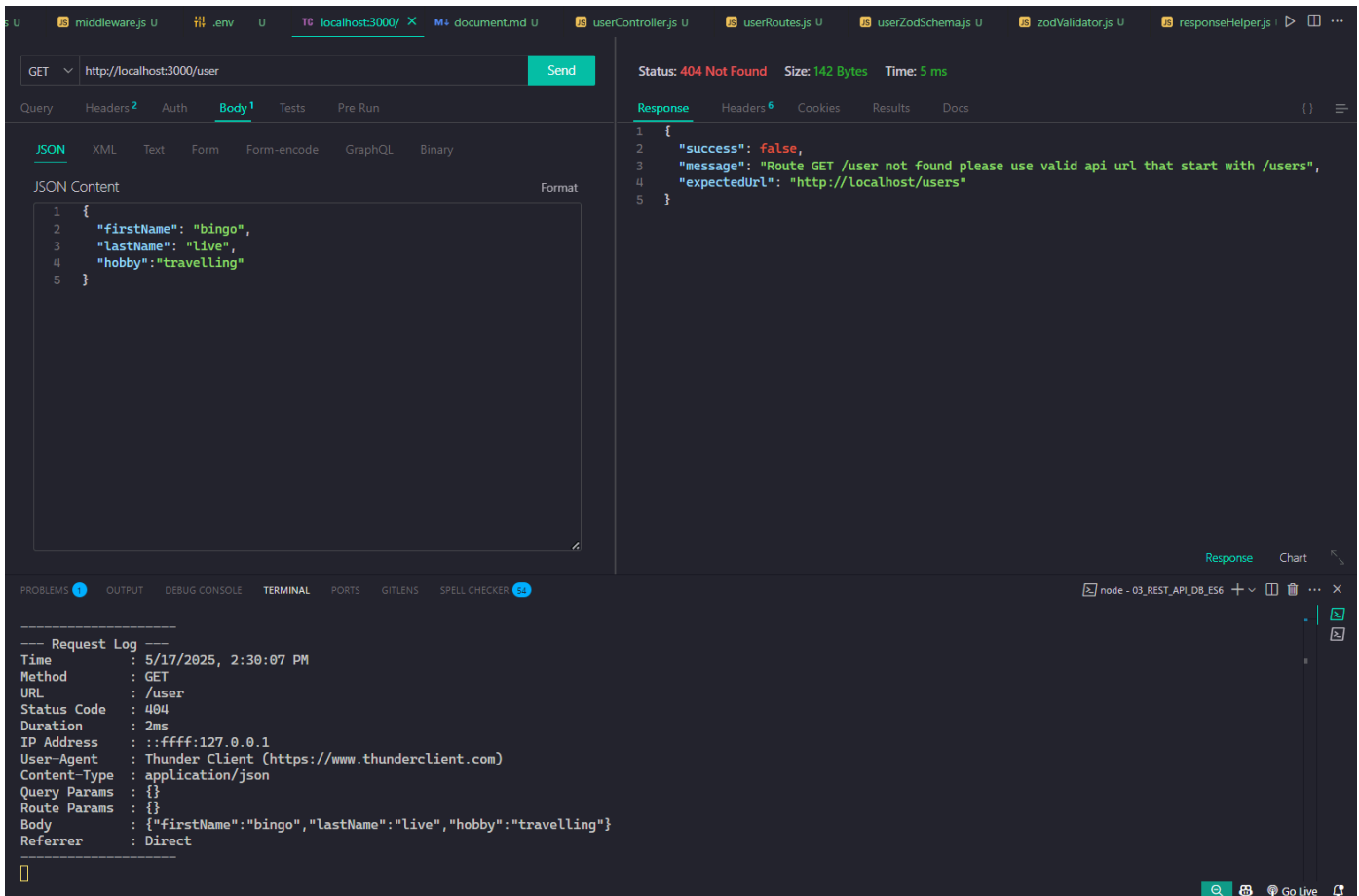


API Test Results Document

GET Request: Fetching All Users

(1) When API URL is `http://localhost:port/user`

- **Scenario:** API URL does **not** start with `/users`.
- **Expected:** Server returns an error due to invalid endpoint.
- **Actual:** Error thrown.
- **HTTP Status:** 404 Not Found



(2) When API URL is `http://localhost:port/users`

- **Scenario:** API URL is valid and starts with `/users`.
- **Expected:** Returns the list of all users.
- **Actual:** Successful response with user data.

- HTTP Status: 200 OK

is U X middleware.js U .env U TC localhost:3000/ X userController.js U userRoutes.js U userZodSchema.js U zodValidator.js U responseHelper.js U userService.js L

GET http://localhost:3000/users Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "firstName": "bingo",
3   "lastName": "lust",
4   "hobby": "x"
5 }
```

Status: 200 OK Size: 436 Bytes Time: 166 ms

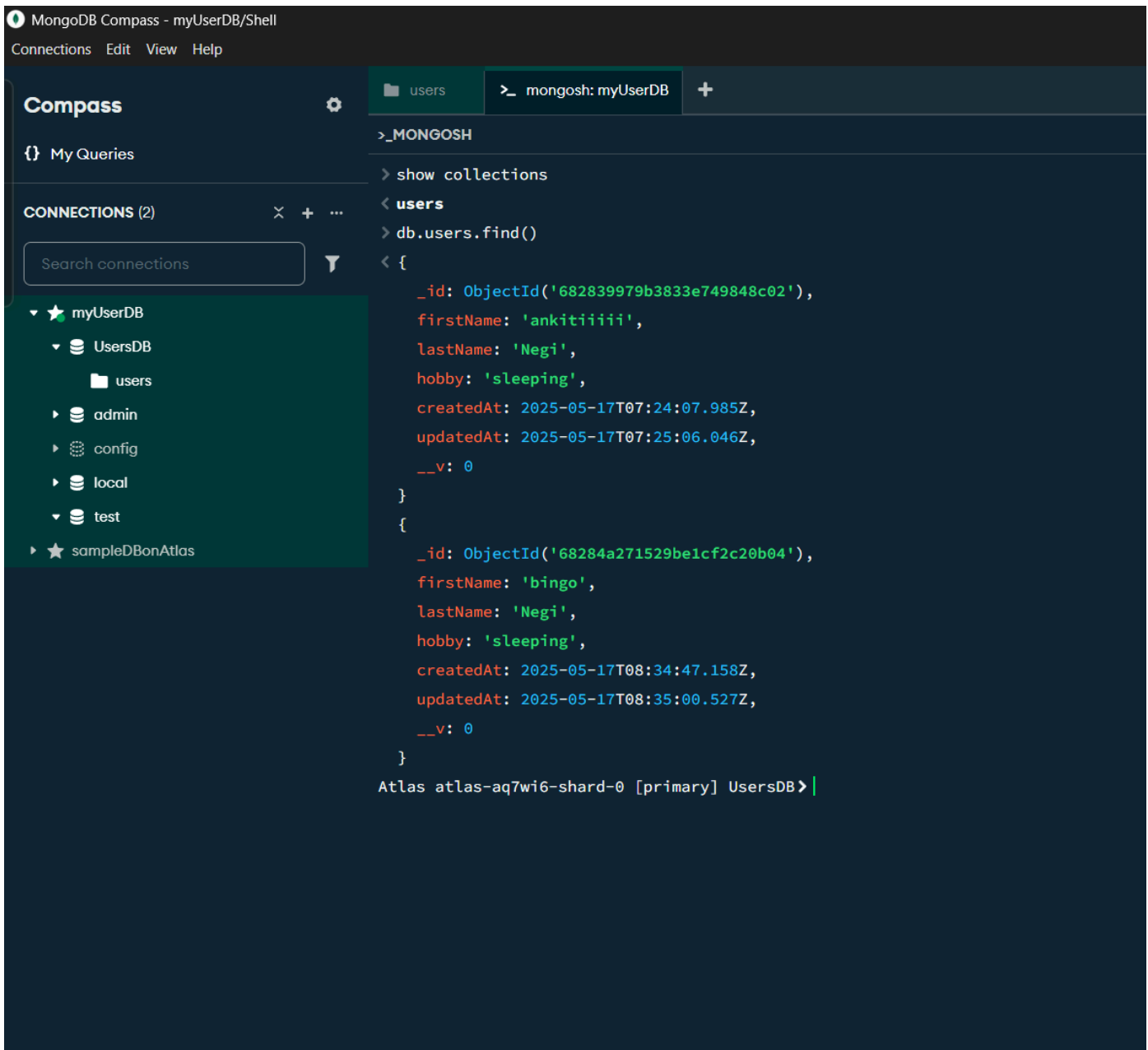
Response Headers 6 Cookies Results Docs

```
1 {
2   "success": true,
3   "data": [
4     {
5       "_id": "682839979b3833e749848c02",
6       "firstName": "ankitiiiiii",
7       "lastName": "Negi",
8       "hobby": "sleeping",
9       "createdAt": "5/17/2025, 12:54:07 PM",
10      "updatedAt": "5/17/2025, 12:55:06 PM",
11      "_v": 0
12     },
13     {
14       "_id": "68284a271529be1cf2c20b04",
15       "firstName": "bingo",
16       "lastName": "Negi",
17       "hobby": "sleeping",
18       "createdAt": "5/17/2025, 2:04:47 PM",
19       "updatedAt": "5/17/2025, 2:05:00 PM",
20       "_v": 0
21     }
22   ],
23   "message": "congratulations all users fetched successfully"
24 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER 25

node - 03_REST_API_DB_ES6

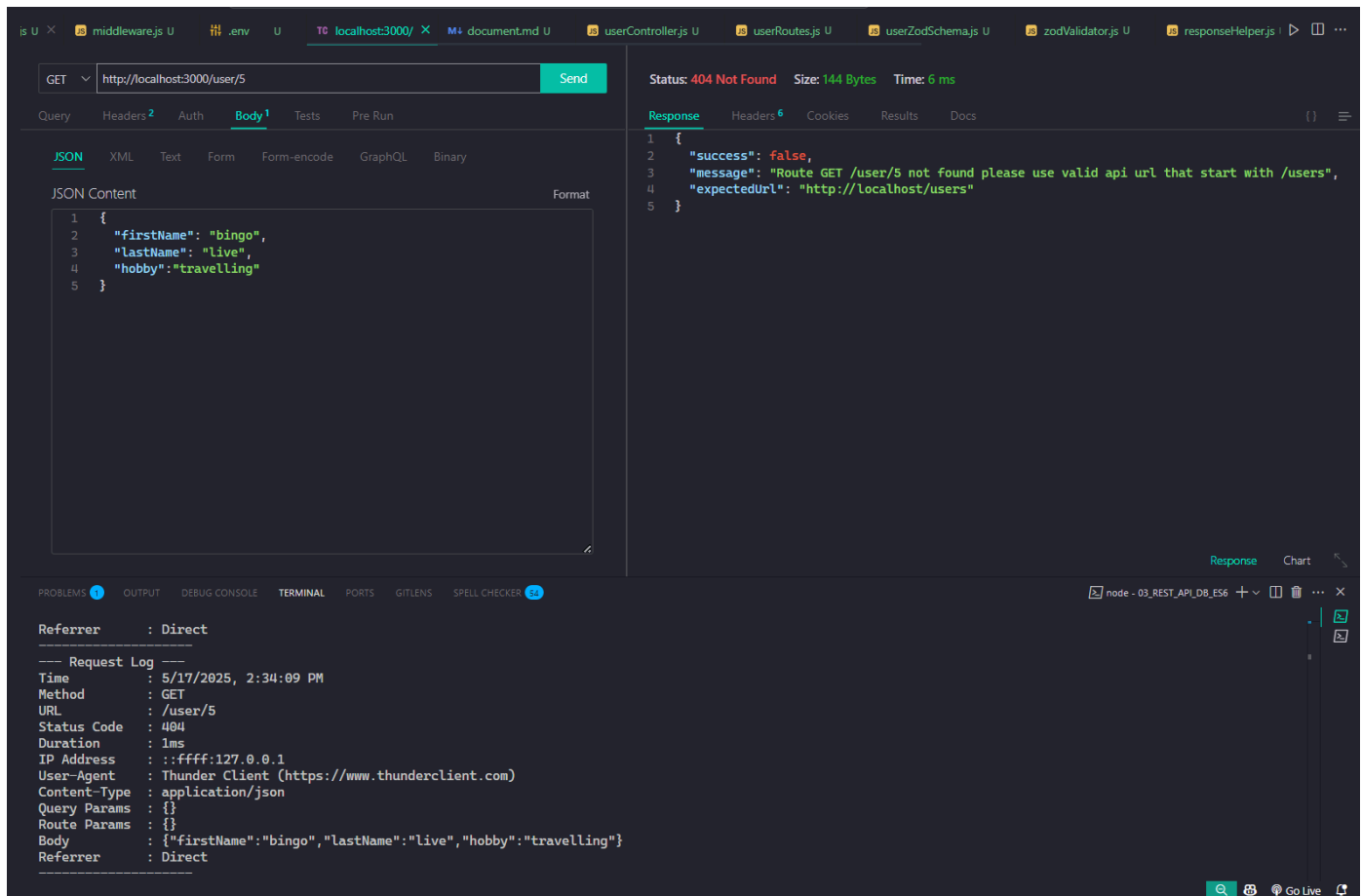
```
]
--- Request Log ---
Time       : 5/17/2025, 2:25:31 PM
Method     : GET
URL        : /users
Status Code : 200
Duration   : 140ms
IP Address  : ::ffff:127.0.0.1
User-Agent  : Thunder Client (https://www.thunderclient.com)
Content-Type : application/json
Query Params : {}
Route Params : {}
Body       : {"firstName":"bingo","lastName":"lust","hobby":"x"}
Referrer    : Direct
```



GET Request: Fetching Single User Data

(1) When API URL is `http://localhost:port/users/invalidId`

- **Scenario:** The user ID in the URL is invalid.
- **Expected:** Server returns an error indicating invalid ID.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request / 404 Not Found (based on implementation)



(2) When API URL is `http://localhost:port/users/validId`

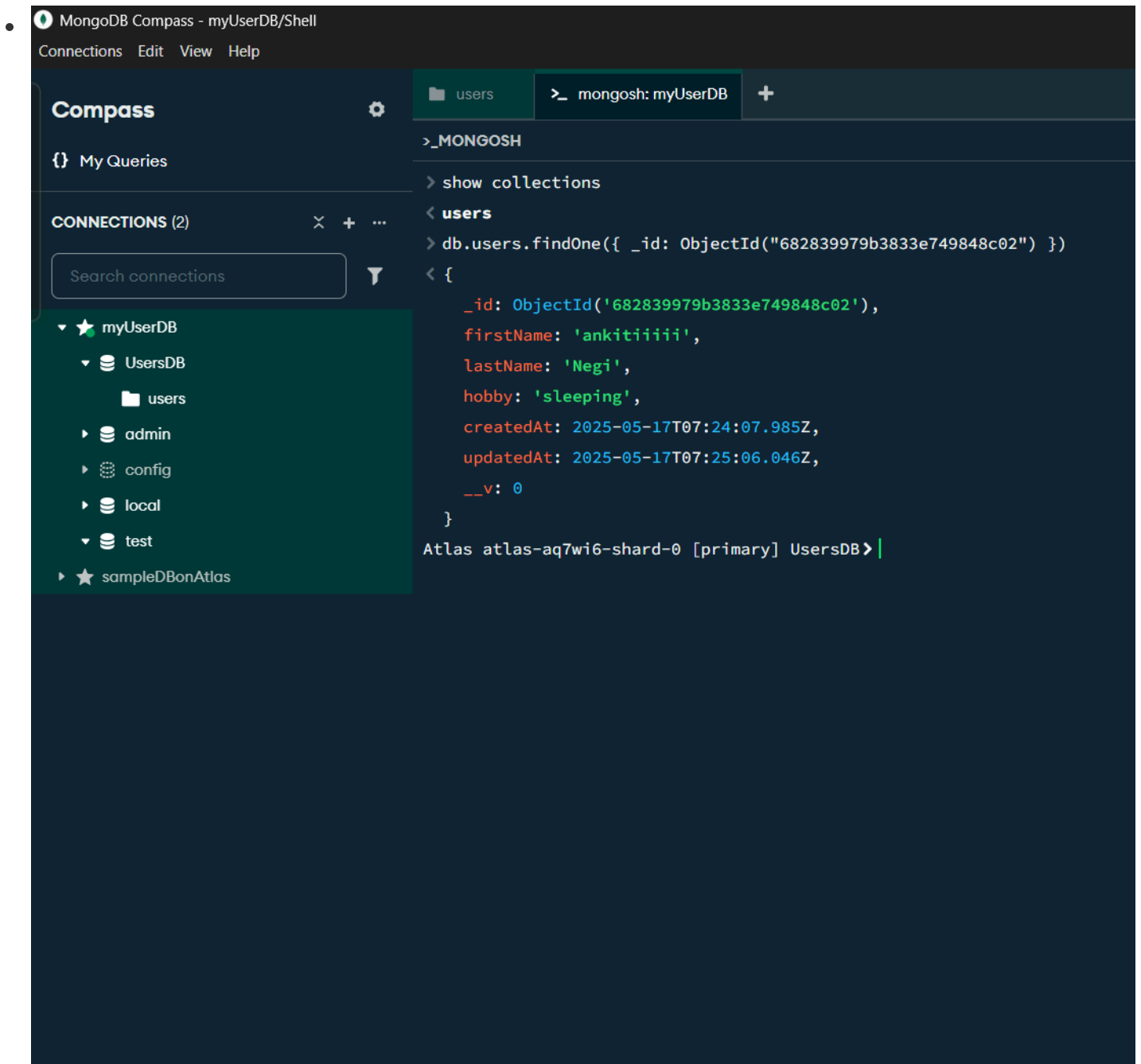
- **Scenario:** The user ID is valid.
- **Expected:** Returns the user data corresponding to the valid ID.
- **Actual:** Successful response with user data.
- **HTTP Status:** 200 OK

The screenshot shows the VS Code interface with a REST client request and response. The request is a GET request to `http://localhost:3000/users/682839979b3833e749848c02`. The response is a JSON object with the following structure:

```
{
  "success": true,
  "data": {
    "_id": "682839979b3833e749848c02",
    "firstName": "ankitiiiiii",
    "lastName": "Negi",
    "hobby": "sleeping",
    "createdAt": "5/17/2025, 12:54:07 PM",
    "updatedAt": "5/17/2025, 12:55:06 PM",
    "__v": 0
  },
  "message": "congratulations user fetched successfully"
}
```

The bottom panel shows the "PROBLEMS" view with a "Request Log" entry:

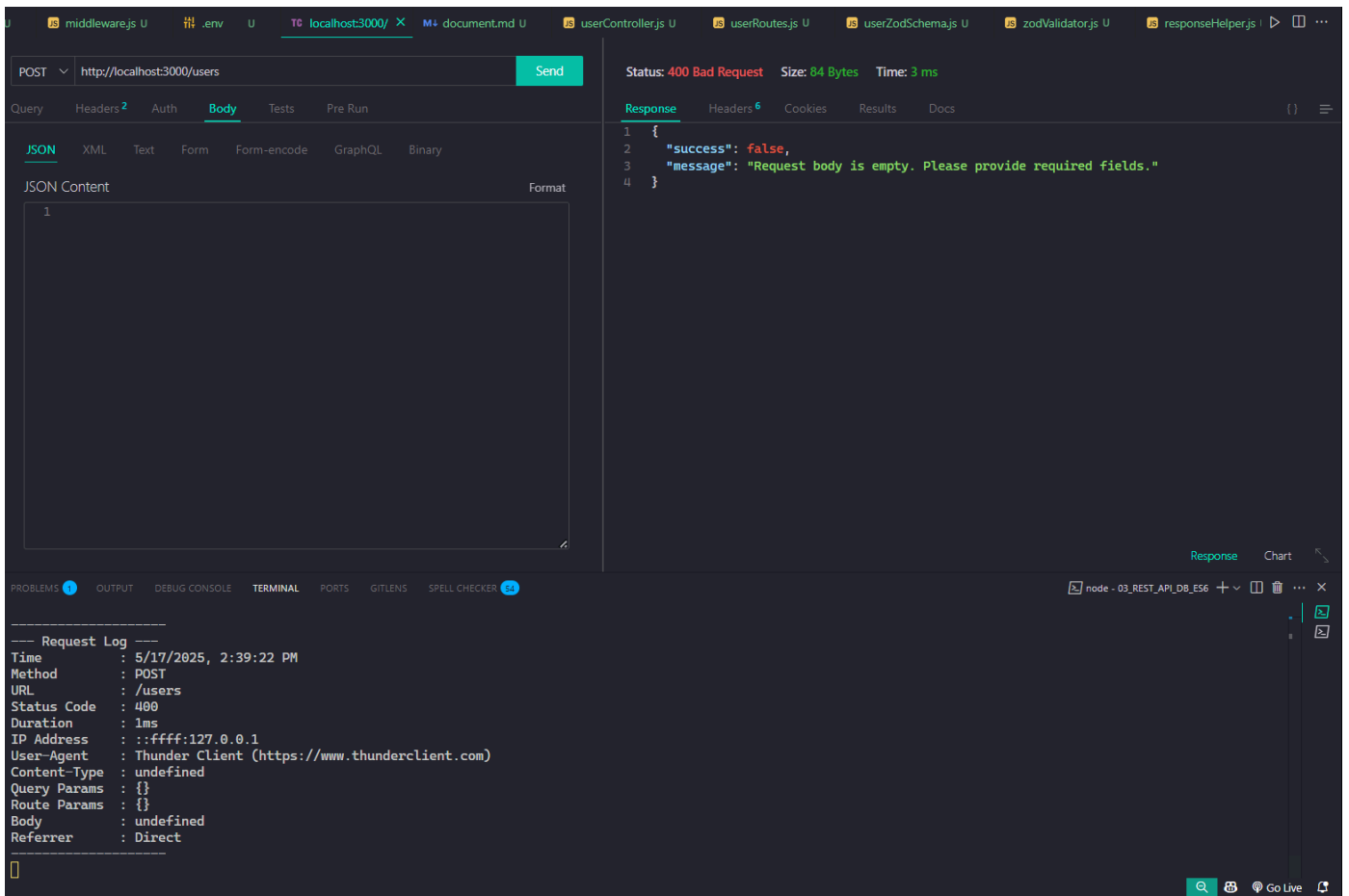
```
--- Request Log ---
Time      : 5/17/2025, 2:35:25 PM
Method    : GET
URL       : /users/682839979b3833e749848c02
Status Code : 200
Duration  : 96ms
IP Address : ::ffff:127.0.0.1
User-Agent : Thunder Client (https://www.thunderclient.com)
Content-Type : application/json
Query Params : {}
Route Params : {"id":"682839979b3833e749848c02"}
Body        : {"firstName":"bingo","lastName":"Live","hobby":"travelling"}
Referrer    : Direct
```



POST Request: Creating a User

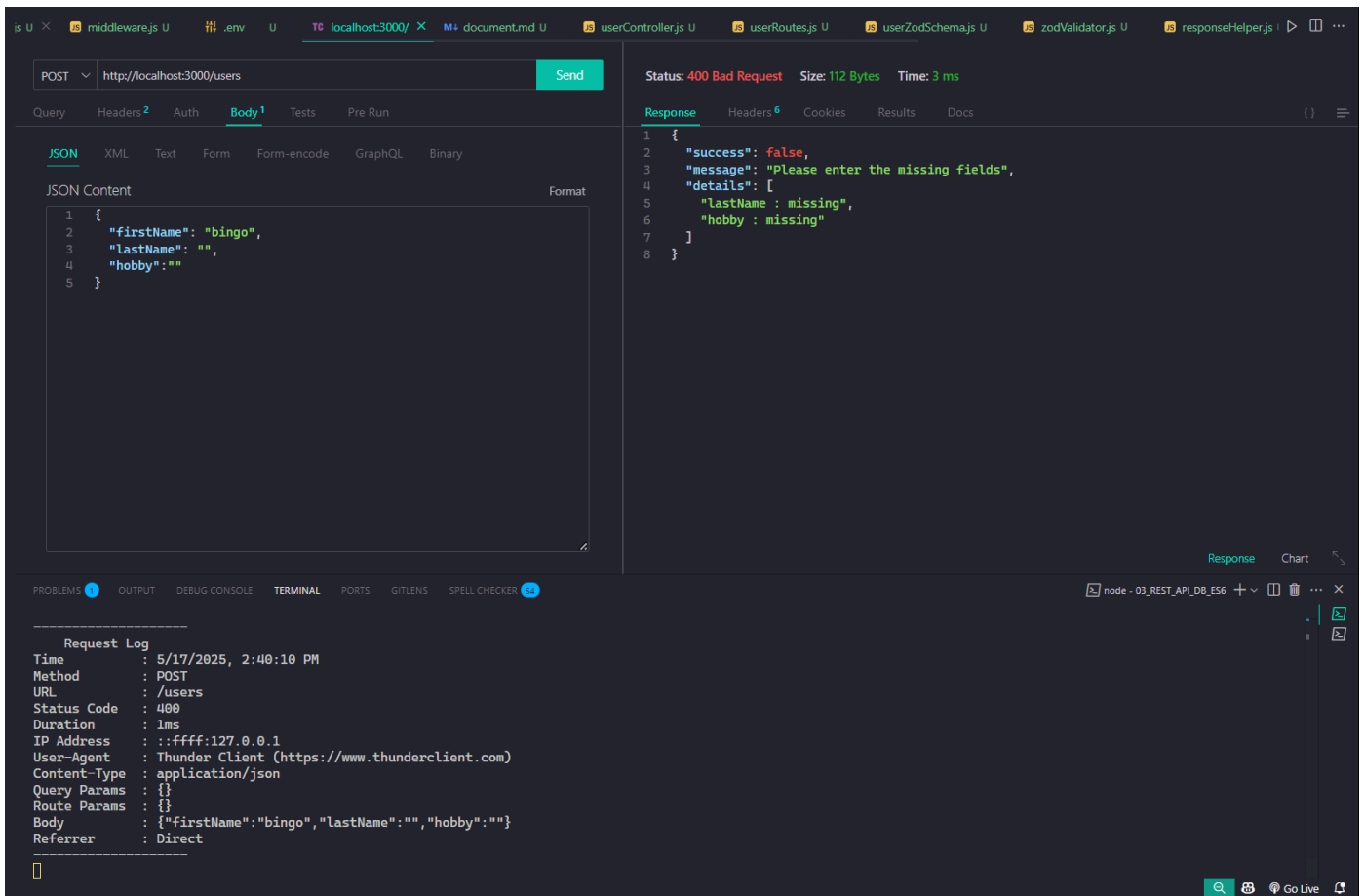
(1) When API URL is `http://localhost:port/users` — No request body provided

- **Scenario:** No request body sent.
- **Expected:** Server throws an error indicating missing request body.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



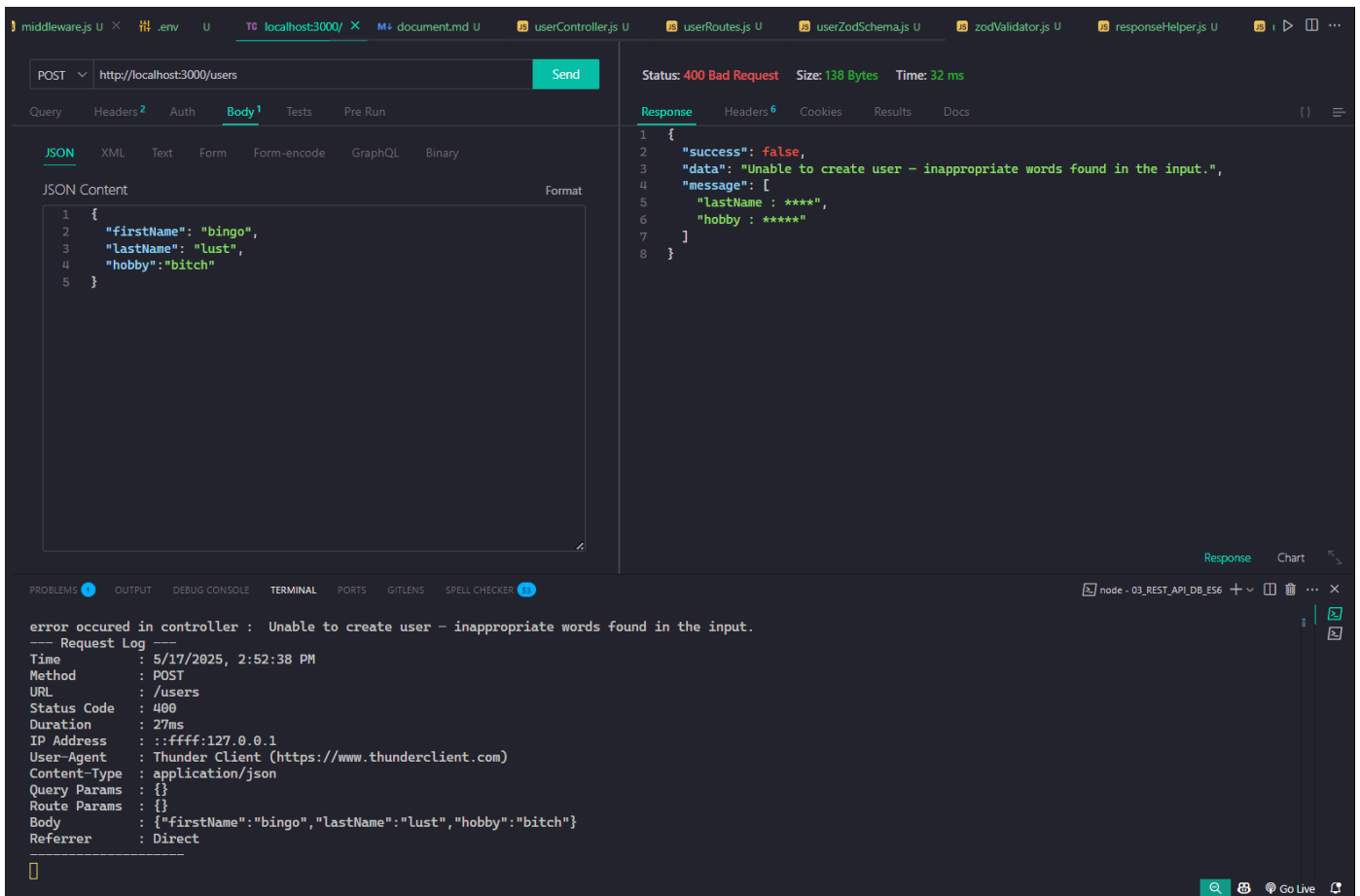
(2) When API URL is `http://localhost:port/users` — Some fields in the request body are empty

- **Scenario:** Request body is provided but some required fields are empty.
- **Expected:** Server returns an error indicating validation failure.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



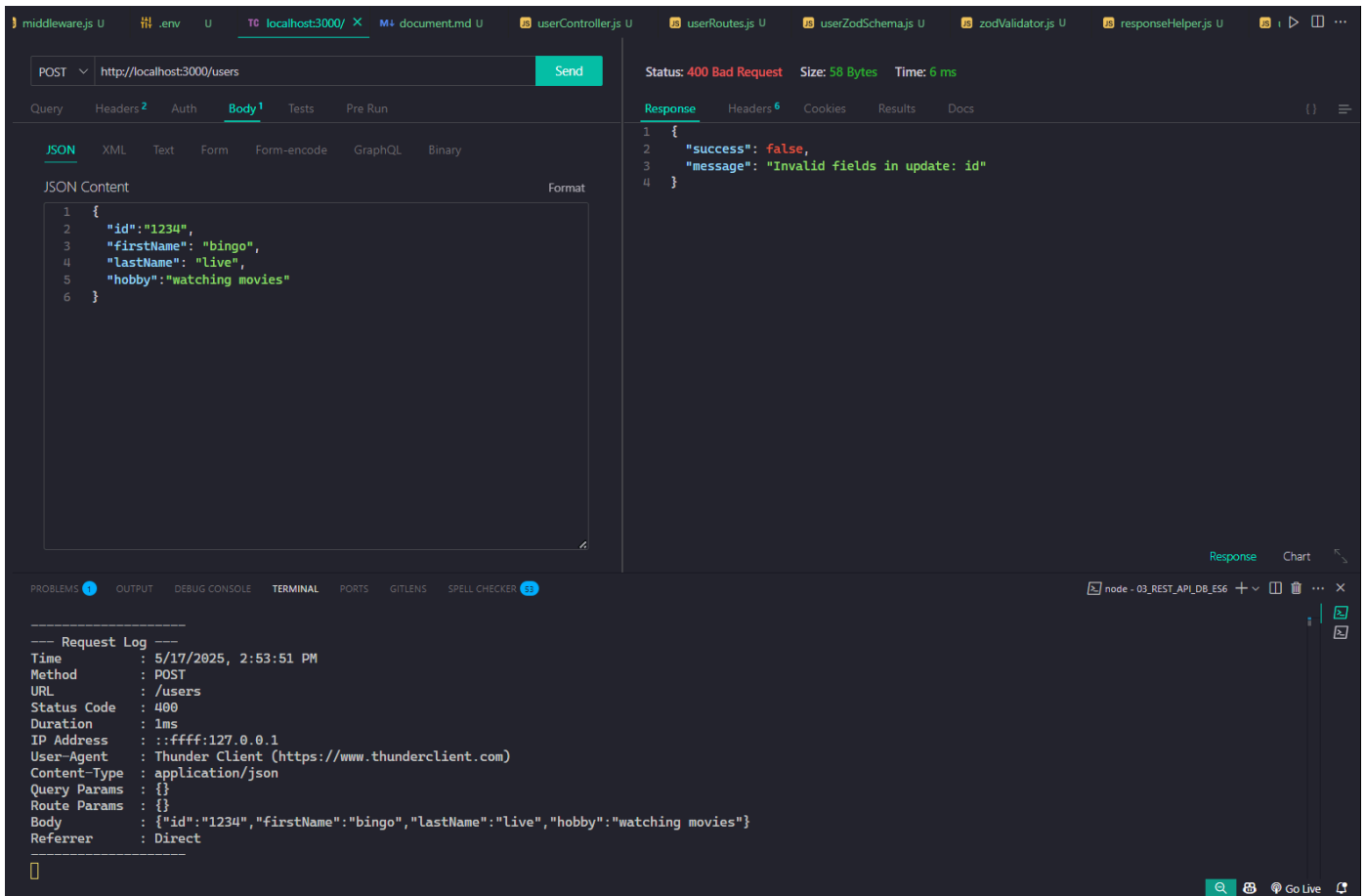
(3) When API URL is `http://localhost:port/users` — Some bad words are passed in request body

- **Scenario:** Request body contains inappropriate or bad words.
- **Expected:** Server rejects input with an error related to content policy.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



(4) When API URL is `http://localhost:port/users` — Invalid fields passed in request body

- **Scenario:** Request body contains fields that are not allowed, e.g., user tries to set the ID manually.
- **Expected:** Server ignores or throws error for invalid fields.
- **Actual:** Error thrown or invalid fields ignored.
- **HTTP Status:** 400 Bad Request



(5) When API URL is `http://localhost:port/users` — Correct fields passed

- **Scenario:** Request body contains all required fields correctly.
- **Expected:** User is created successfully.
- **Actual:** Success response.
- **HTTP Status:** 201 Created

middleware.js U .env U T6 localhost:3000/ X M+ document.md U userController.js U userRoutes.js U userZodSchemas U zodValidator.js U responseHelper.js U

POST http://localhost:3000/users Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "firstName": "anjana",
3   "lastName": "Rawat",
4   "hobby": "watching movies"
5 }
```

Status: 201 Created Size: 259 Bytes Time: 84 ms

Response Headers 6 Cookies Results Docs

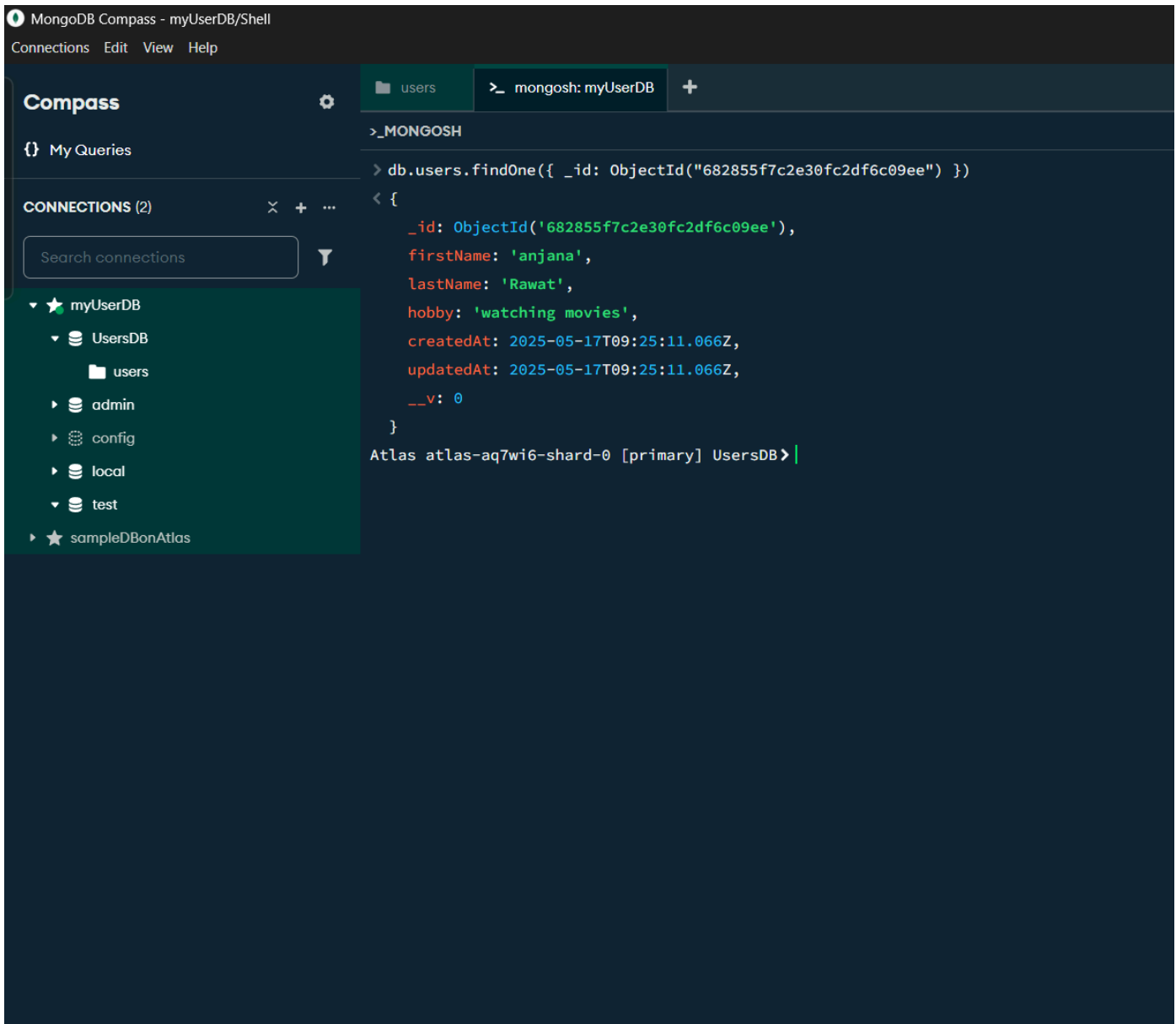
```
1 {
2   "success": true,
3   "data": {
4     "firstName": "anjana",
5     "lastName": "Rawat",
6     "hobby": "watching movies",
7     "_id": "682855f7c2e30fc2df6c09ee",
8     "createdAt": "5/17/2025, 2:55:11 PM",
9     "updatedAt": "5/17/2025, 2:55:11 PM",
10    "__v": 0
11  },
12  "message": "congratulations user created successfully"
13 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS GIT LENS SPELL CHECKER 5

node - 03_REST_API_08_ES6

```
request.body is : { firstName: 'anjana', lastName: 'Rawat', hobby: 'watching movies' }
--- Request Log ---
Time      : 5/17/2025, 2:55:11 PM
Method    : POST
URL       : /users
Status Code : 201
Duration  : 82ms
IP Address  : ::ffff:127.0.0.1
User-Agent : Thunder Client (https://www.thunderclient.com)
Content-Type : application/json
Query Params : {}
Route Params : {}
Body       : {"firstName":"anjana","lastName":"Rawat","hobby":"watching movies"}
Referrer   : Direct
```

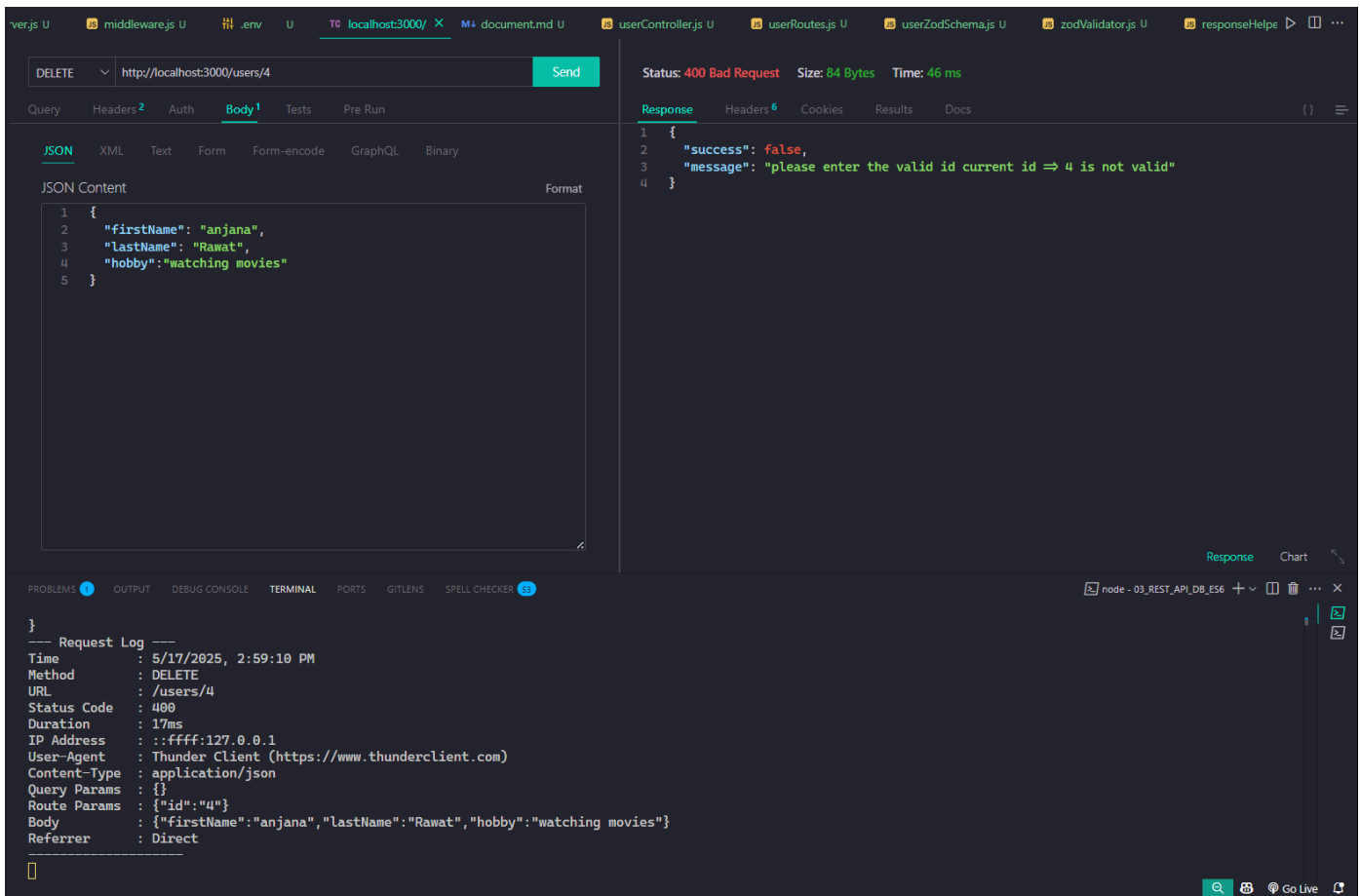
Go Live



DELETE Request: Deleting a User

(1) When API URL is `http://localhost:port/users/id` — Invalid ID passed

- **Scenario:** ID provided does not exist or is invalid.
- **Expected:** Server returns error for invalid ID.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad request or 404.



(2) When API URL is `http://localhost:port/users/id` — Valid ID passed

- **Scenario:** Valid user ID provided.
- **Expected:** User is deleted successfully.
- **Actual:** Success response.
- **HTTP Status:** 200 OK

ver.js U middleware.js U .env U localhost:3000/ X M+ document.md U userController.js U userRoutes.js U userZodSchema.js U zodValidator.js U responseHelper.js U

DELETE http://localhost:3000/users/682855f7c2e30fc2df6c09ee Send

Query Headers Auth Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "firstName": "anjana",
3   "lastName": "Rawat",
4   "hobby": "watching movies"
5 }
```

Status: 200 OK Size: 133 Bytes Time: 87 ms

Response Headers Cookies Results Docs

```
1 {
2   "success": true,
3   "data": {
4     "success": true,
5     "message": "User deleted successfully"
6   },
7   "message": "congratulations user deleted successfully"
8 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SPELL CHECKER

node - 03_REST_API_DB_ES6

Request Log

Time : 5/17/2025, 3:02:07 PM

Method : DELETE

URL : /users/682855f7c2e30fc2df6c09ee

Status Code : 200

Duration : 84ms

IP Address : ::ffff:127.0.0.1

User-Agent : Thunder Client (https://www.thunderclient.com)

Content-Type : application/json

Query Params : {}

Route Params : {"id":"682855f7c2e30fc2df6c09ee"}

Body : {"firstName":"anjana","lastName":"Rawat","hobby":"watching movies"}

Referrer : Direct

MongoDB Compass - myUserDB/Shell

Connections Edit View Help

Compass

users mongosh: myUserDB +

>_MONGOSH

```
> db.users.findOne({ _id: ObjectId("682855f7c2e30fc2df6c09ee") })
< {
  _id: ObjectId('682855f7c2e30fc2df6c09ee'),
  firstName: 'anjana',
  lastName: 'Rawat',
  hobby: 'watching movies',
  createdAt: 2025-05-17T09:25:11.066Z,
  updatedAt: 2025-05-17T09:25:11.066Z,
  __v: 0
}
> db.users.findOne({ _id: ObjectId("682855f7c2e30fc2df6c09ee") })
< null
Atlas atlas-aq7wi6-shard-0 [primary] UsersDB>
```

My Queries

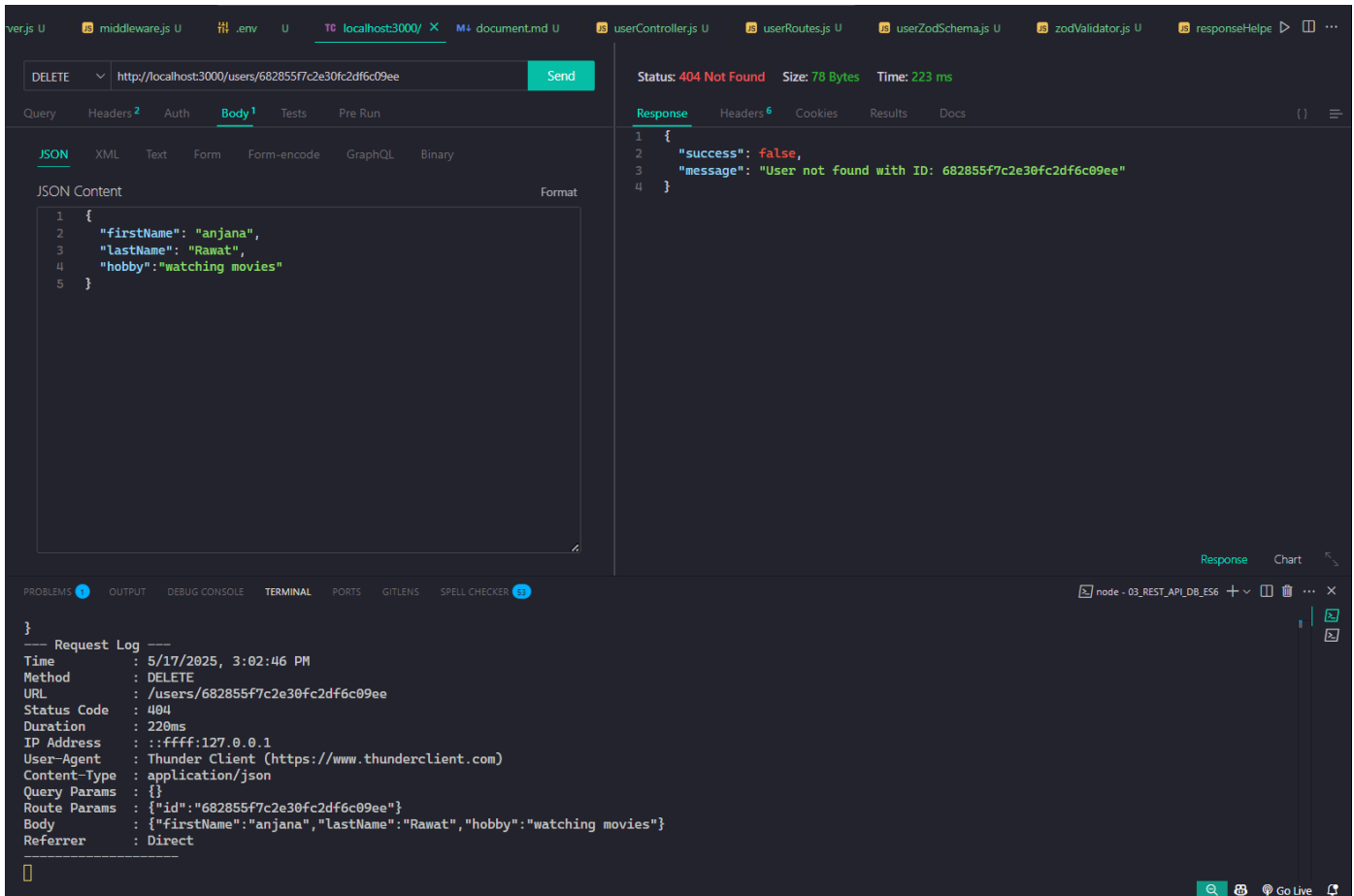
CONNECTIONS (2)

Search connections

- myUserDB
 - UsersDB
 - users
 - admin
 - config
 - local
 - test
 - sampleDBonAtlas

(3) When API URL is `http://localhost:port/users/id` — Deleting a user already deleted

- **Scenario:** Attempting to delete a user that was already deleted.
- **Expected:** Server returns error indicating user not found.
- **Actual:** Error thrown.
- **HTTP Status:** 404 Not Found

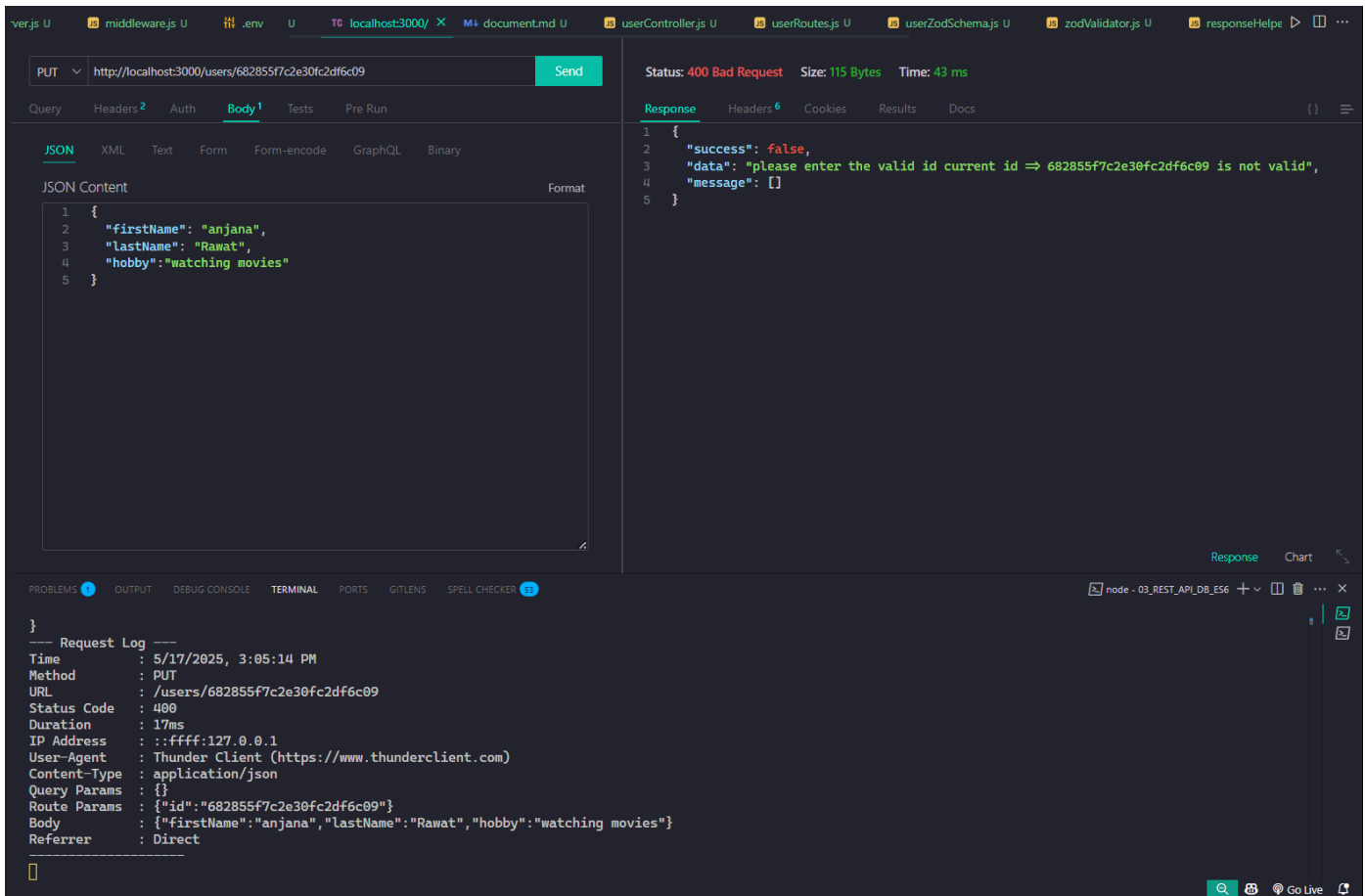


PUT Request: Updating Existing User

(1) When API URL is `http://localhost:port/users/id` — Invalid ID passed by user

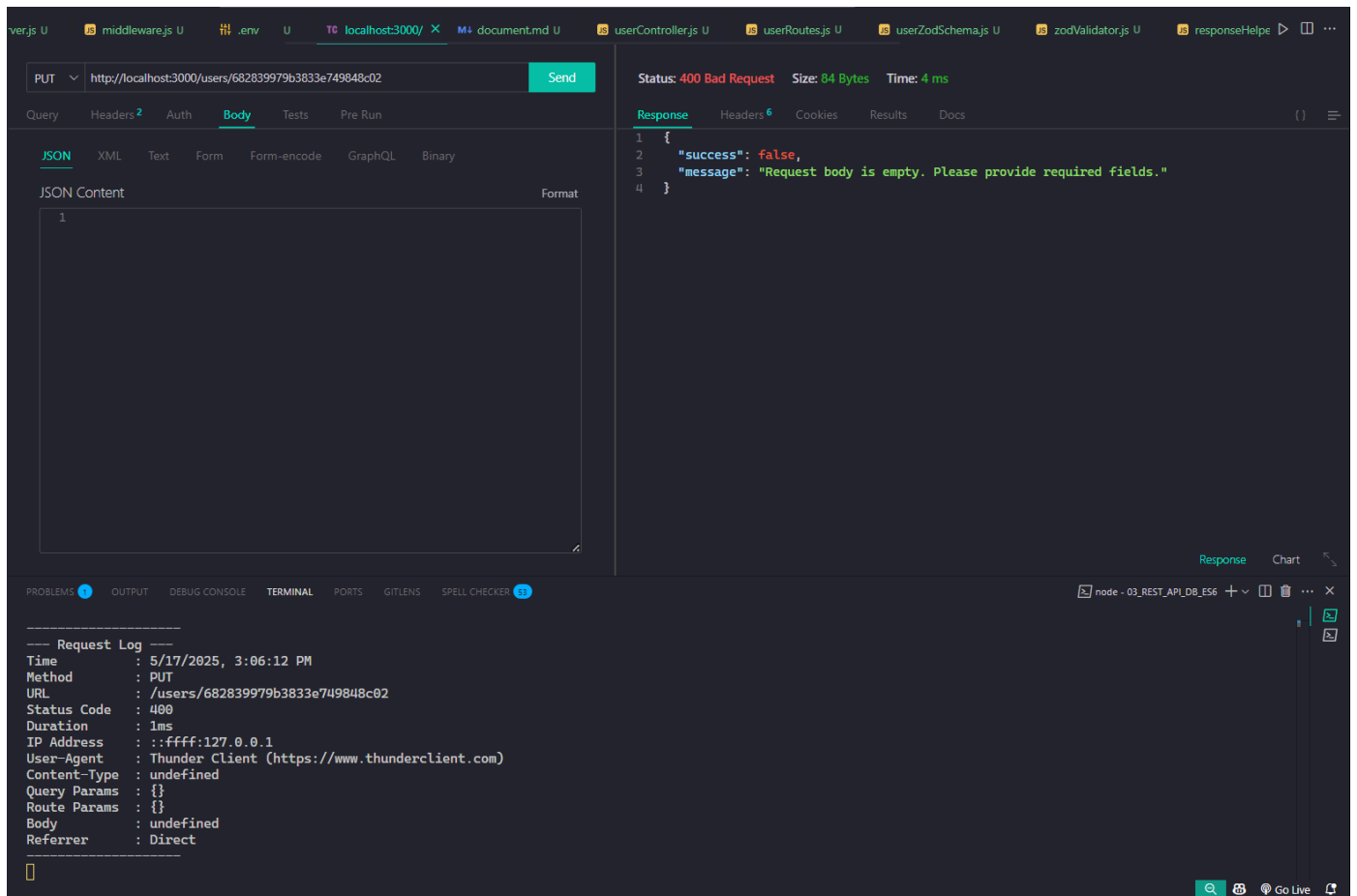
- **Scenario:** User ID is invalid.
- **Expected:** Server returns error indicating invalid ID.
- **Actual:** Error thrown.

- **HTTP Status: 400 Bad Request / 404 Not Found**



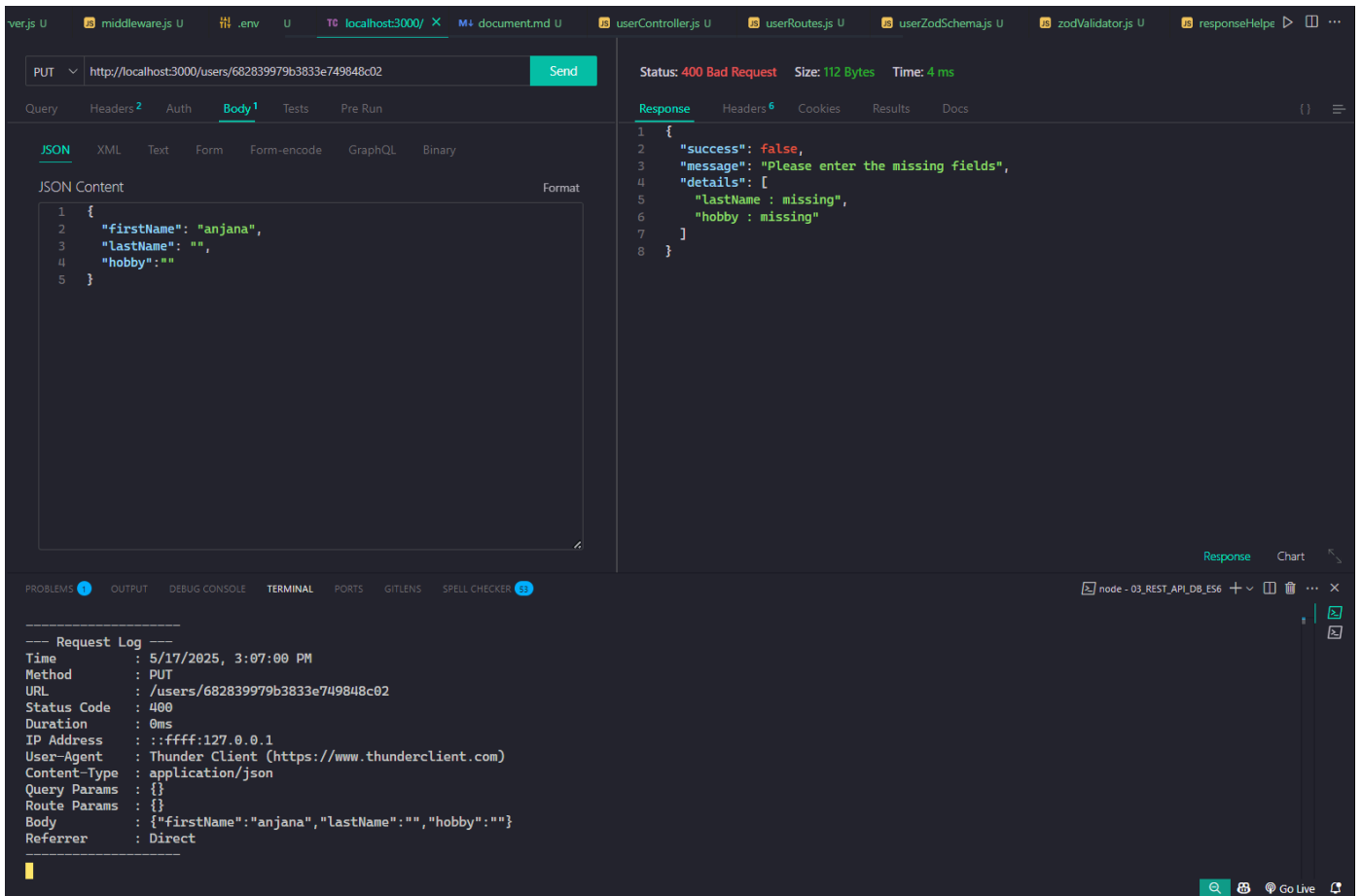
(2) When API URL is http://localhost:port/users/id — Valid ID but no request body passed

- **Scenario:** No request body sent with the update.
- **Expected:** Server returns error for missing update data.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



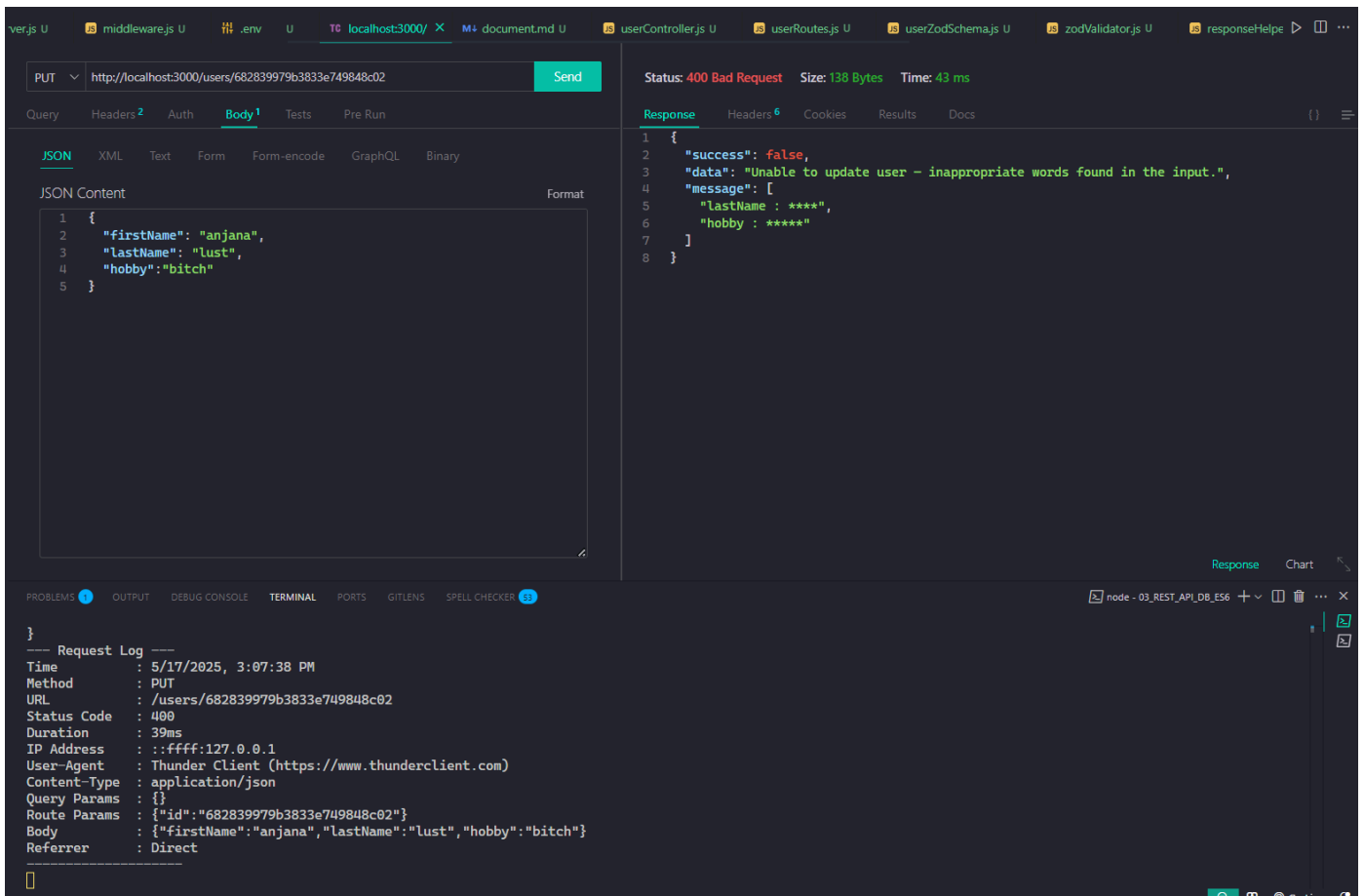
(3) When API URL is `http://localhost:port/users/id` — Valid ID but some fields are empty

- **Scenario:** Request body has some empty fields.
- **Expected:** Server returns validation error.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



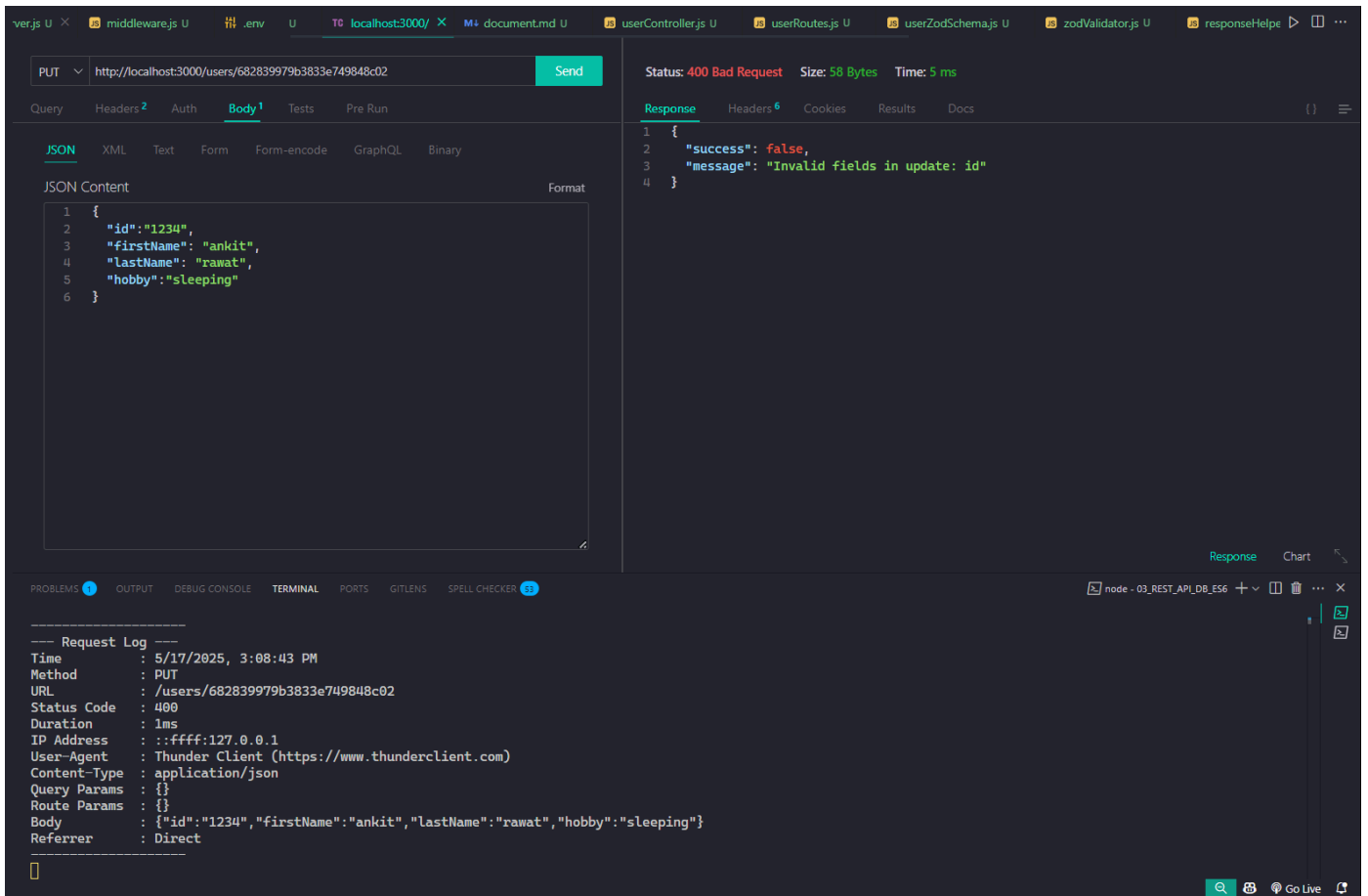
(4) When API URL is `http://localhost:port/users/id` — Valid ID but some fields contain bad words

- **Scenario:** Request body contains inappropriate content.
- **Expected:** Server rejects update due to bad content.
- **Actual:** Error thrown.
- **HTTP Status:** 400 Bad Request



(5) When API URL is `http://localhost:port/users/id` — Valid ID but extra invalid fields included

- **Scenario:** Request body contains fields not allowed to be updated.
- **Expected:** Server ignores or throws error for invalid fields.
- **Actual:** Error thrown or ignored.
- **HTTP Status:** 400 Bad Request



(6) When API URL is `http://localhost:port/users/id` — Valid ID and valid request body

- **Scenario:** All fields are valid and allowed.
- **Expected:** User updated successfully.
- **Actual:** Success response.
- **HTTP Status:** 200 OK

Compass

{ } My Queries

CONNECTIONS (2)

Search connections

- ★ myUserDB
 - UsersDB
 - users
 - admin
 - config
 - local
 - test
- ★ sampleDBonAtlas

users

> mongosh: myUserDB

+

> _MONGOSH

> db.users.findOne({ _id: ObjectId("682855f7c2e30fc2df6c09ee") })

< {

```
  _id: ObjectId('682855f7c2e30fc2df6c09ee'),
  firstName: 'anjana',
  lastName: 'Rawat',
  hobby: 'watching movies',
  createdAt: 2025-05-17T09:25:11.066Z,
  updatedAt: 2025-05-17T09:25:11.066Z,
  __v: 0
}
```

> db.users.findOne({ _id: ObjectId("682855f7c2e30fc2df6c09ee") })

< null

> db.users.findOne({ _id: ObjectId("682839979b3833e749848c02") })

< {

```
  _id: ObjectId('682839979b3833e749848c02'),
  firstName: 'ankit',
  lastName: 'rawat',
  hobby: 'sleeping',
  createdAt: 2025-05-17T07:24:07.985Z,
  updatedAt: 2025-05-17T09:39:35.028Z,
  __v: 0
}
```

Atlas atlas-aq7wi6-shard-0 [primary] UsersDB>