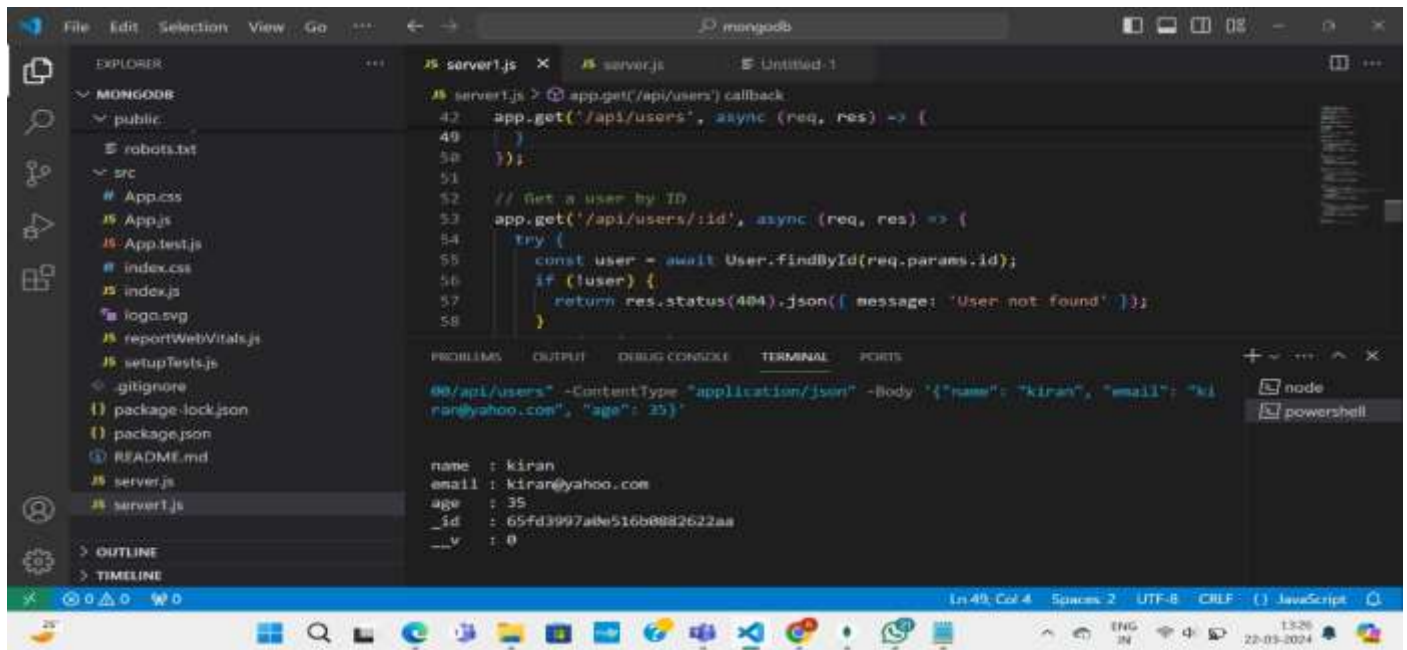


Output:



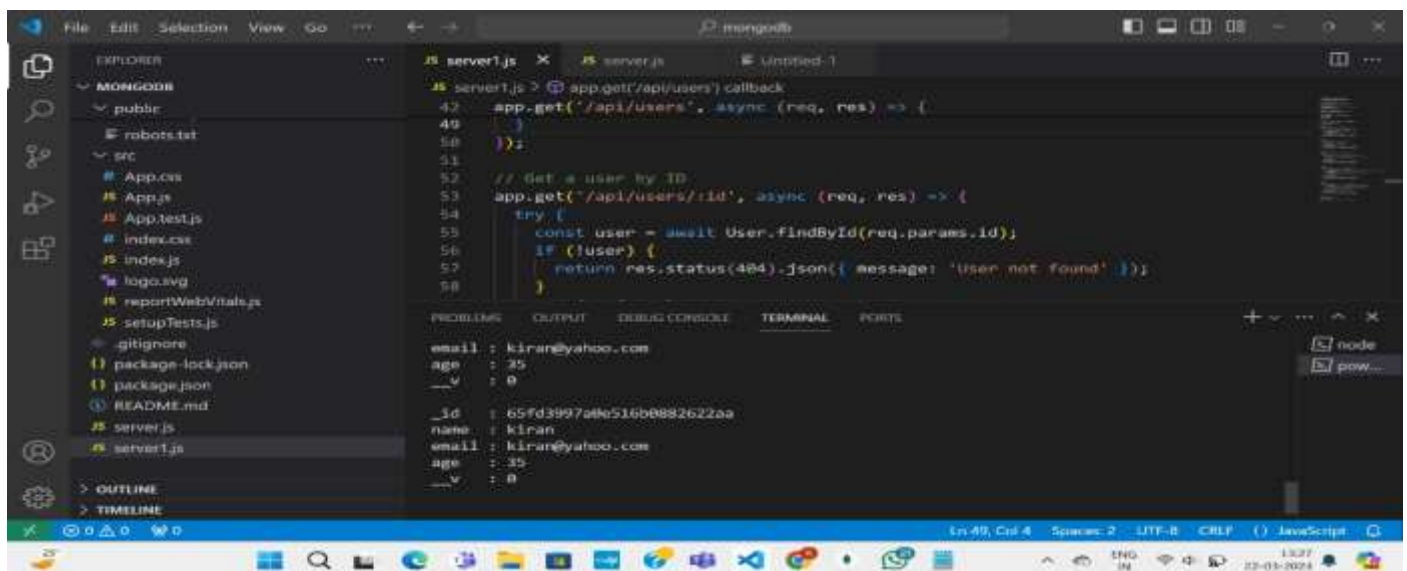
The screenshot shows a VS Code editor with a REST client request and its response. The request is a GET request to `/api/users` with a body of `{ "name": "kiran", "email": "kiran@yahoo.com", "age": 35 }`. The response is a 200 status code with a body of `{ "name": "kiran", "email": "kiran@yahoo.com", "age": 35, "_id": "65fd3997a8e516b882622aa", "__v": 0 }`.

```
server.js > app.get('/api/users') callback
42 app.get('/api/users', async (req, res) => {
49   }
50 });
51
52 // Get a user by ID
53 app.get('/api/users/:id', async (req, res) => {
54   try {
55     const user = await User.findById(req.params.id);
56     if (!user) {
57       return res.status(404).json({ message: 'User not found' });
58     }
59   }
60 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

00/api/users" -ContentType "application/json" -Body '{"name": "kiran", "email": "kiran@yahoo.com", "age": 35}'

name : kiran
email : kiran@yahoo.com
age : 35
_id : 65fd3997a8e516b882622aa
__v : 0

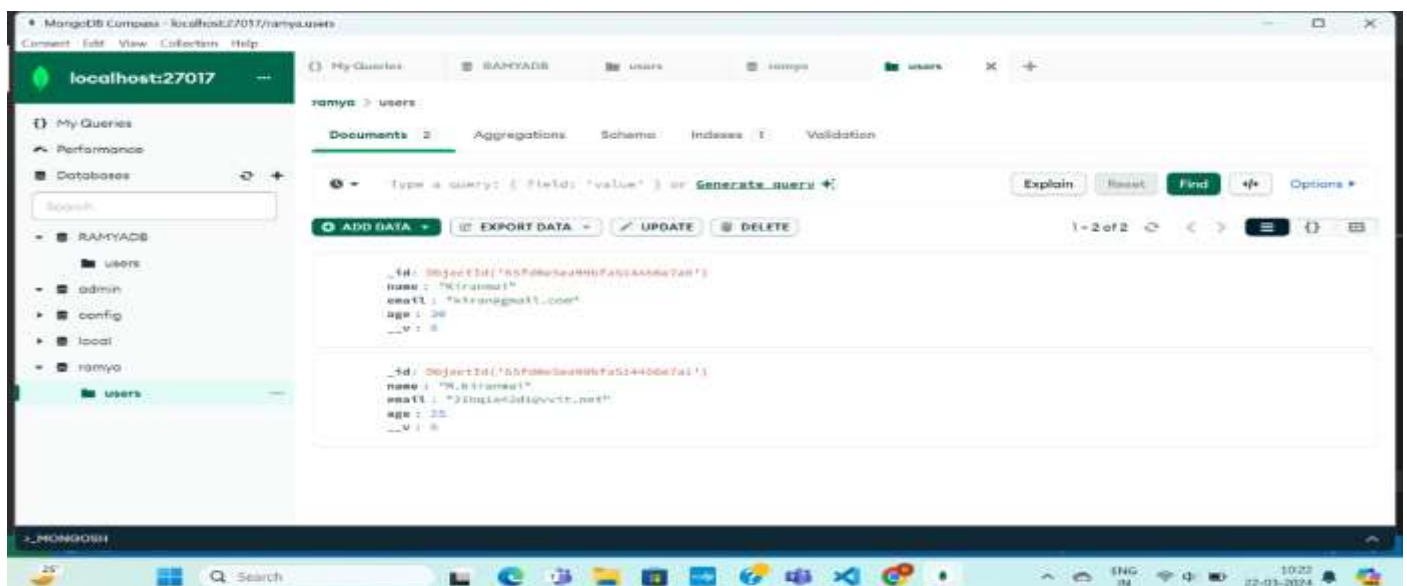


The screenshot shows a VS Code editor with a REST client request and its response. The request is a GET request to `/api/users` with a body of `{ "name": "kiran", "email": "kiran@yahoo.com", "age": 35 }`. The response is a 200 status code with a body of `{ "name": "kiran", "email": "kiran@yahoo.com", "age": 35, "_id": "65fd3997a8e516b882622aa", "__v": 0 }`.

```
server.js > app.get('/api/users') callback
42 app.get('/api/users', async (req, res) => {
49   }
50 });
51
52 // Get a user by ID
53 app.get('/api/users/:id', async (req, res) => {
54   try {
55     const user = await User.findById(req.params.id);
56     if (!user) {
57       return res.status(404).json({ message: 'User not found' });
58     }
59   }
60 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

email : kiran@yahoo.com
age : 35
__v : 0
_id : 65fd3997a8e516b882622aa
name : kiran
email : kiran@yahoo.com
age : 35
__v : 0



The screenshot shows the MongoDB Compass interface. The left sidebar shows the database structure with the `users` collection selected. The main area shows the `users` collection with two documents. The first document has `_id`: `ObjectId('65fd3997a8e516b882622aa')`, `name`: `"kiranmai"`, `email`: `"kiranmai@gmail.com"`, `age`: `30`, and `__v`: `0`. The second document has `_id`: `ObjectId('65fd3997a8e516b882622aa')`, `name`: `"kiranmai"`, `email`: `"kiranmai@gmail.com"`, `age`: `35`, and `__v`: `0`.

localhost:27017

My Queries Performance Databases Search

RAMYACB

- admin
- config
- local
- ramya
- users

ramya > users

Documents 2 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or Generate query

EXPLAIN RESET FIND OPTIONS

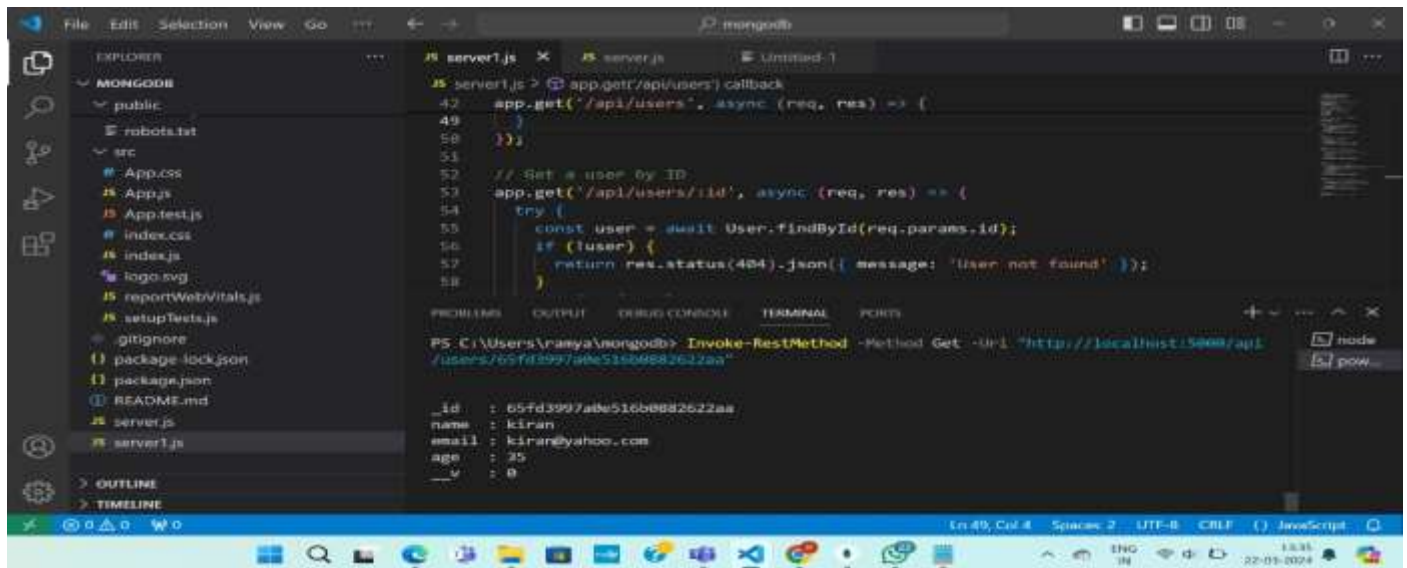
ADD DATA EXPORT DATA UPDATE DELETE

1 - 2 of 2

_id: ObjectId('65fd3997a8e516b882622aa')
name: "kiranmai"
email: "kiranmai@gmail.com"
age: 30
__v: 0

_id: ObjectId('65fd3997a8e516b882622aa')
name: "kiranmai"
email: "kiranmai@gmail.com"
age: 35
__v: 0

_MONGODB

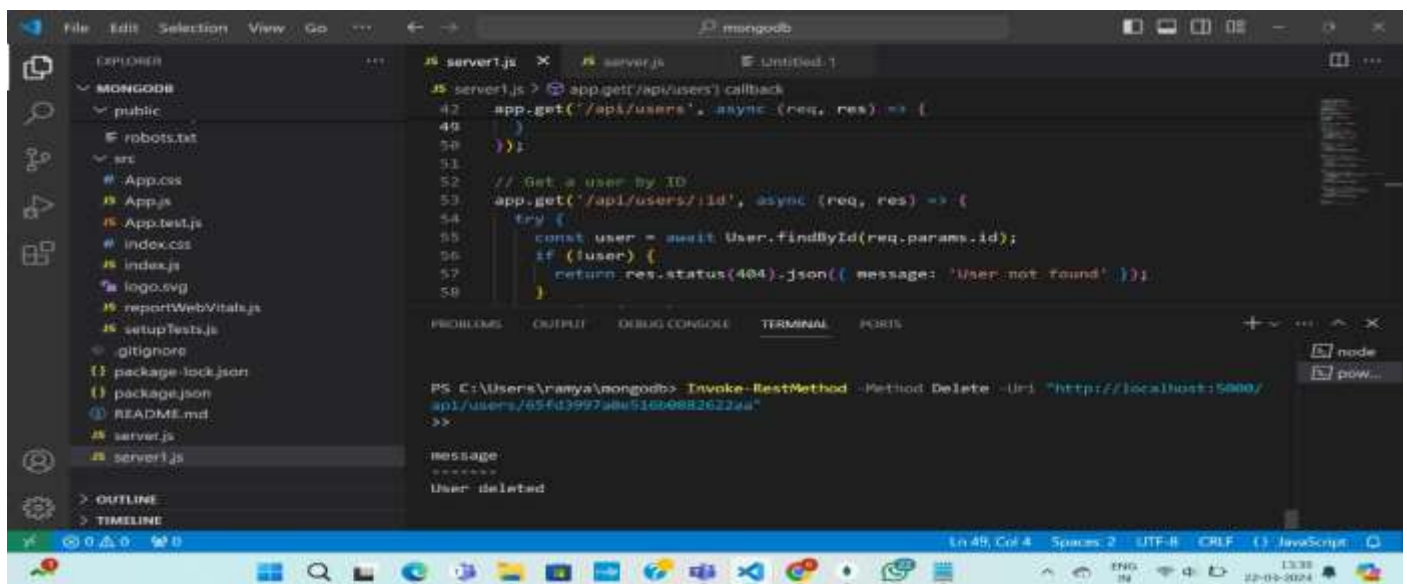


The screenshot shows a VS Code editor with a file explorer on the left containing a project named 'MONGODB'. The main editor displays a JavaScript file 'server.js' with the following code:

```
42 app.get('/api/users') callback
43
44 app.get('/api/users', async (req, res) => {
45
46 })
47
48 // Set a user by ID
49 app.get('/api/users/:id', async (req, res) => {
50   try {
51     const user = await User.findById(req.params.id);
52     if (!user) {
53       return res.status(404).json({ message: 'User not found' });
54     }
55   }
56 })
```

The terminal at the bottom shows a REST client request being executed:

```
PS C:\Users\ranya\nongodb> Invoke-RestMethod -Method Get -Uri "http://localhost:5000/api/users/65fd3997a8e516b0882622aa"
{
  "_id": "65fd3997a8e516b0882622aa",
  "name": "kiran",
  "email": "kiran@yahoo.com",
  "age": 25,
  "__v": 0
}
```



The screenshot shows the same VS Code editor with the 'server.js' file. The terminal now shows a REST client delete request being executed:

```
PS C:\Users\ranya\nongodb> Invoke-RestMethod -Method Delete -Uri "http://localhost:5000/api/users/65fd3997a8e516b0882622aa"
{
  "message": "User deleted"
}
```

