📚 Database Schema

1. `books`

Column Name	Data Type	Description
`id`	Integer	Book ID (Primary Key)
`title`	String	Book title
`author_id`	Integer	Foreign key referencing `authors.id`
`genre_id`	Integer	Foreign key referencing `genres.id`

---

2. `authors`

Column Name	Data Type	Description
`id`	Integer	Author ID (Primary Key)
`name`	String	Author's name

---

3. `genres`

Column Name	Data Type	Description
`id`	Integer	Genre ID (Primary Key)
`name`	String	Genre name

---

4. `users`

Column Name	Data Type	Description
`id`	Integer	User ID (Primary Key)
`name`	String	User name

`email`	String	User email

🔌 Assignment Tasks

🔧 API Functionality (Required)

Implement REST API endpoints for each table with these operations:

Table	Endpoints
`books`	`GET`, `POST`, `PUT`, `DELETE`
`authors`	`GET`, `POST`, `PUT`, `DELETE`
`genres`	`GET`, `POST`, `PUT`, `DELETE`
`users`	`GET`, `POST`, `PUT`, `DELETE`

You Should Be Able To

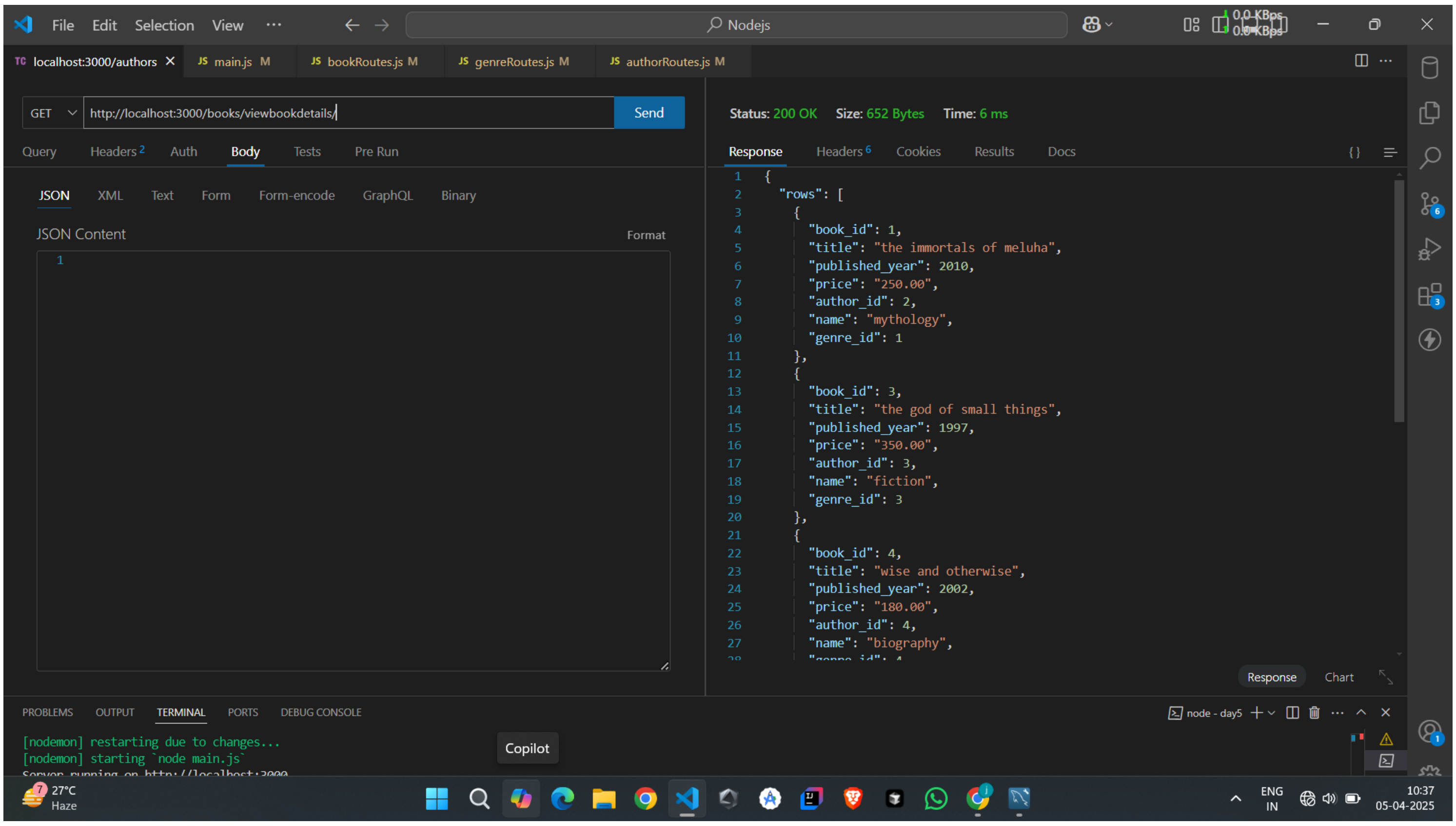
- List all books **with author and genre names**.
- Create new authors, genres, books, and users.
- Update and delete records by ID.

Jafar Beldar

[Github Link]

OUTPUTS

List all books by author and genre names:



GET http://localhost:3000/books/viewbookdetails/

Status: 200 OK Size: 652 Bytes Time: 6 ms

```
{
  "rows": [
    {
      "book_id": 1,
      "title": "the immortals of meluha",
      "published_year": 2010,
      "price": "250.00",
      "author_id": 2,
      "name": "mythology",
      "genre_id": 1
    },
    {
      "book_id": 3,
      "title": "the god of small things",
      "published_year": 1997,
      "price": "350.00",
      "author_id": 3,
      "name": "fiction",
      "genre_id": 3
    },
    {
      "book_id": 4,
      "title": "wise and otherwise",
      "published_year": 2002,
      "price": "180.00",
      "author_id": 4,
      "name": "biography",
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
```

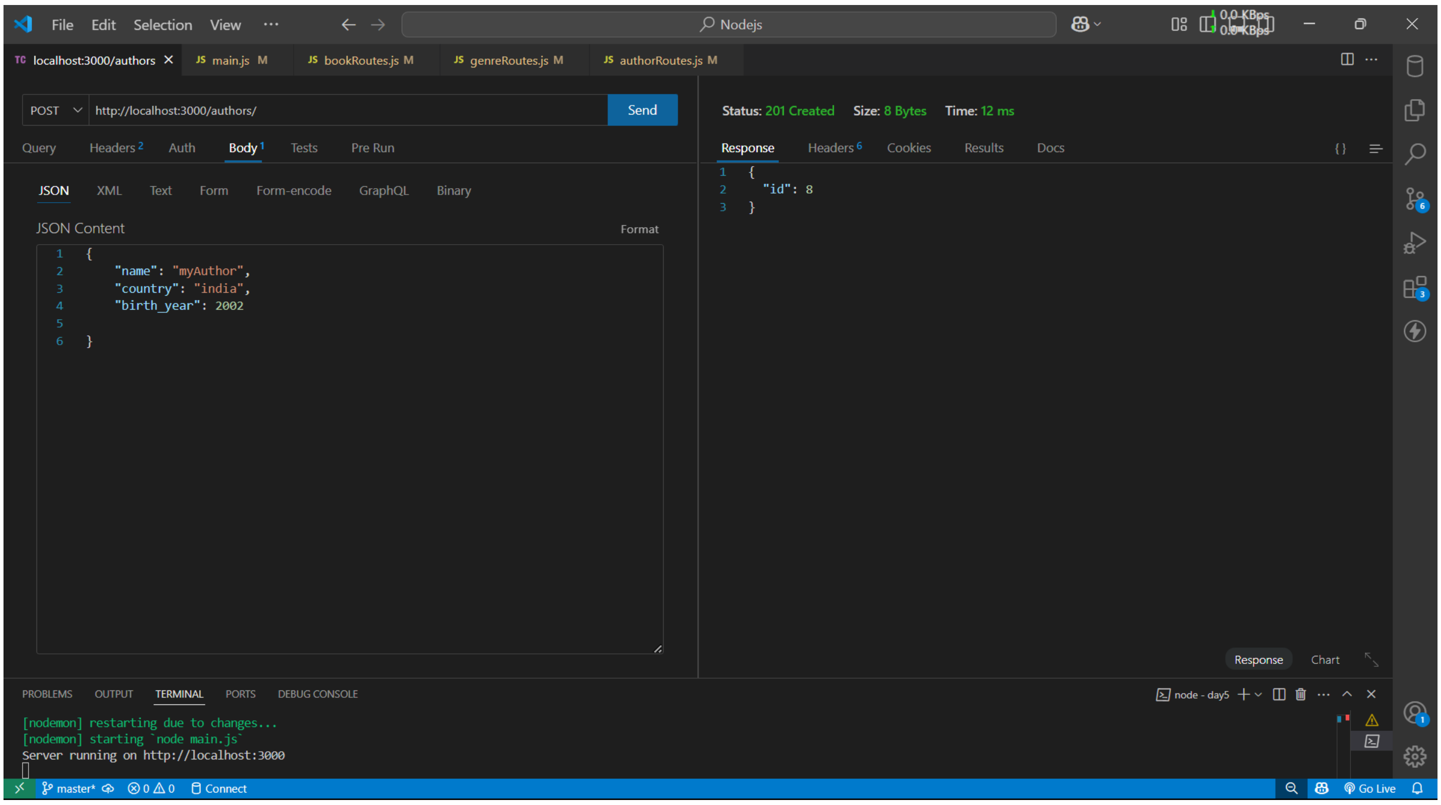
Sql query:

select book_id,title,published_year,price,b.author_id,a.name,g.genre_id,g.name from books b

join authors a on b.author_id=a.author_id

join genres g on b.genre_id=g.genre_id;

Create new authors, genres, books, and users.



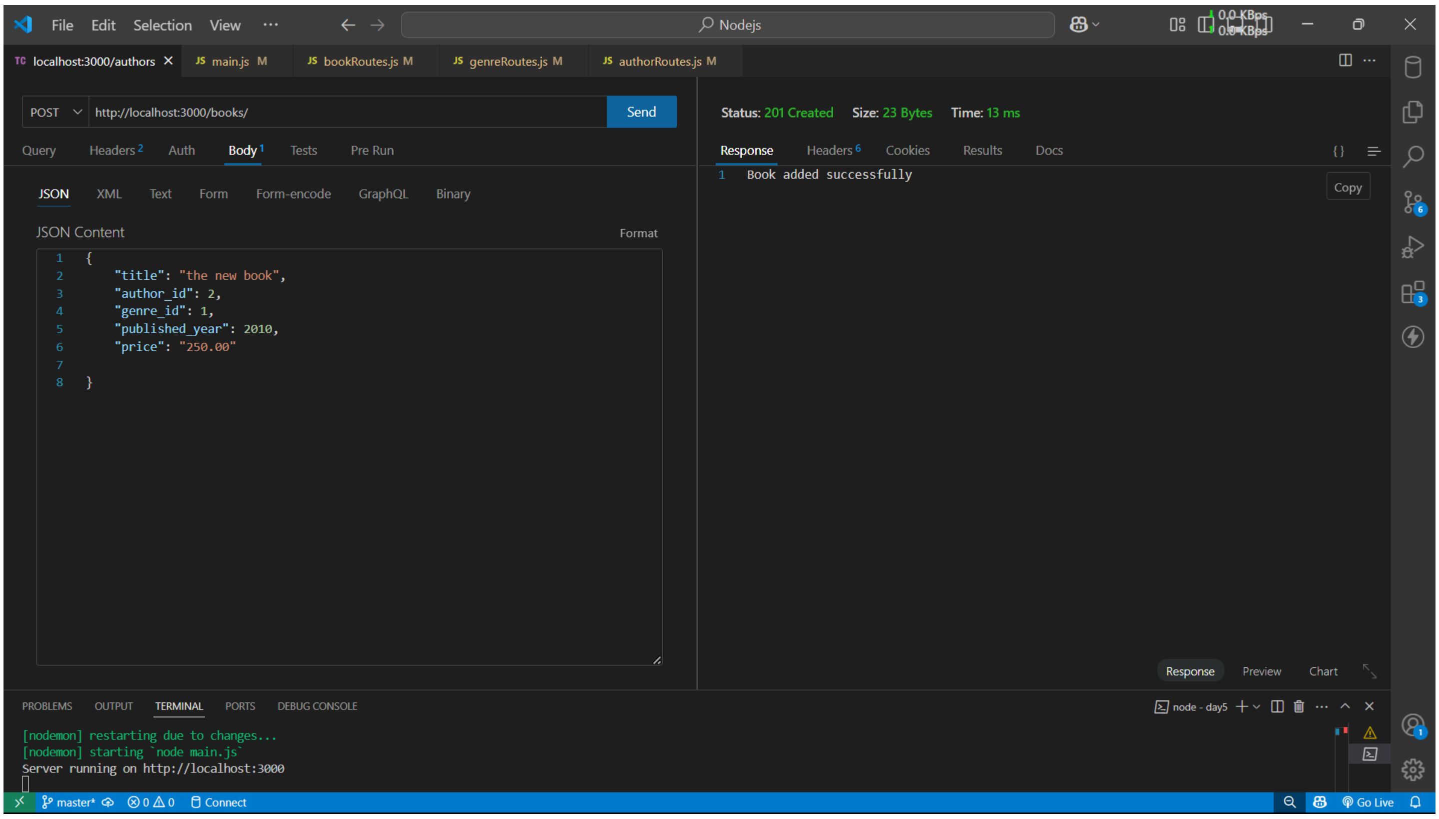
POST http://localhost:3000/authors/

```
{
    "name": "myAuthor",
    "country": "india",
    "birth_year": 2002

}
```

Status: 201 Created Size: 8 Bytes Time: 12 ms

```
{
  "id": 8
}
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
Server running on http://localhost:3000
```



POST http://localhost:3000/books/

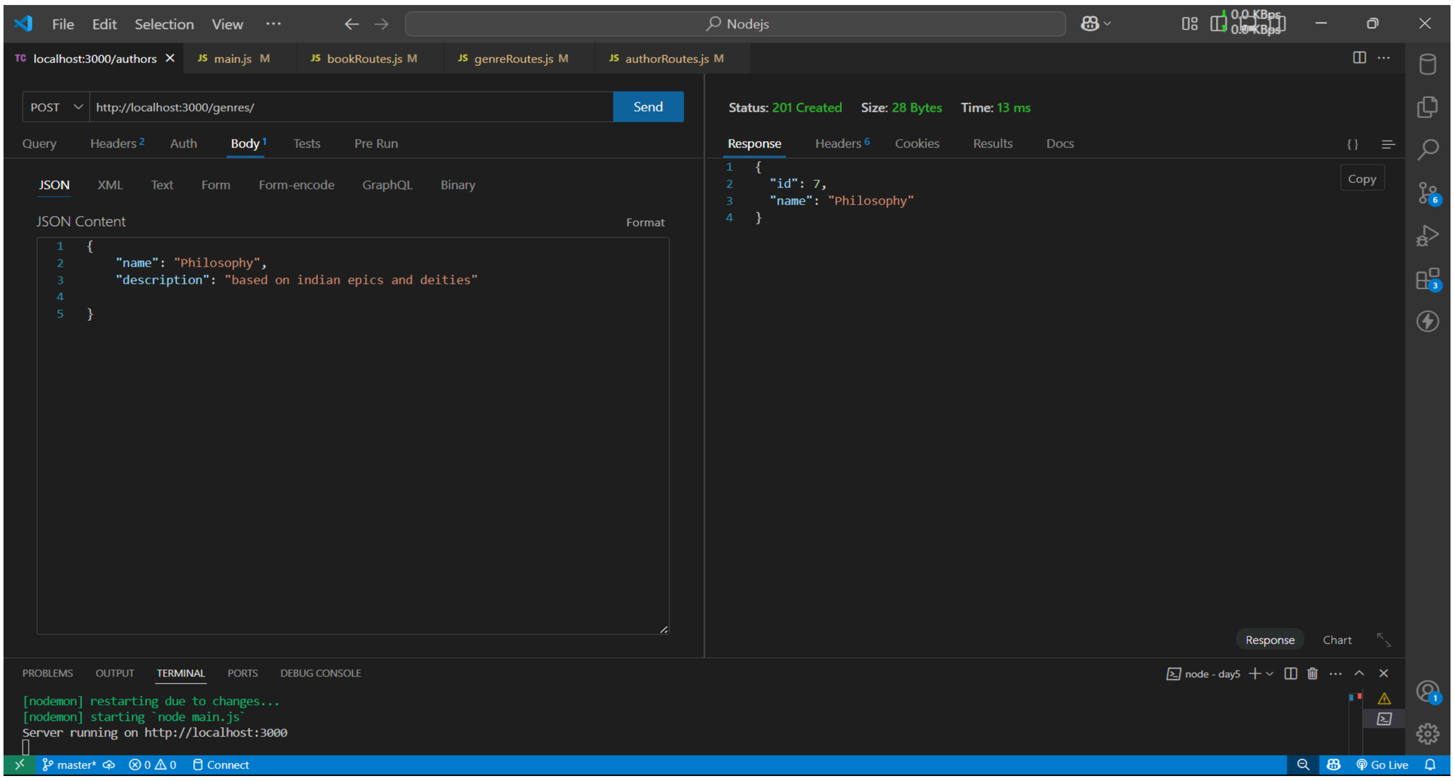
```
{
    "title": "the new book",
    "author_id": 2,
    "genre_id": 1,
    "published_year": 2010,
    "price": "250.00"

}
```

Status: 201 Created Size: 23 Bytes Time: 13 ms

```
Book added successfully
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
Server running on http://localhost:3000
```

Jafar Beldar



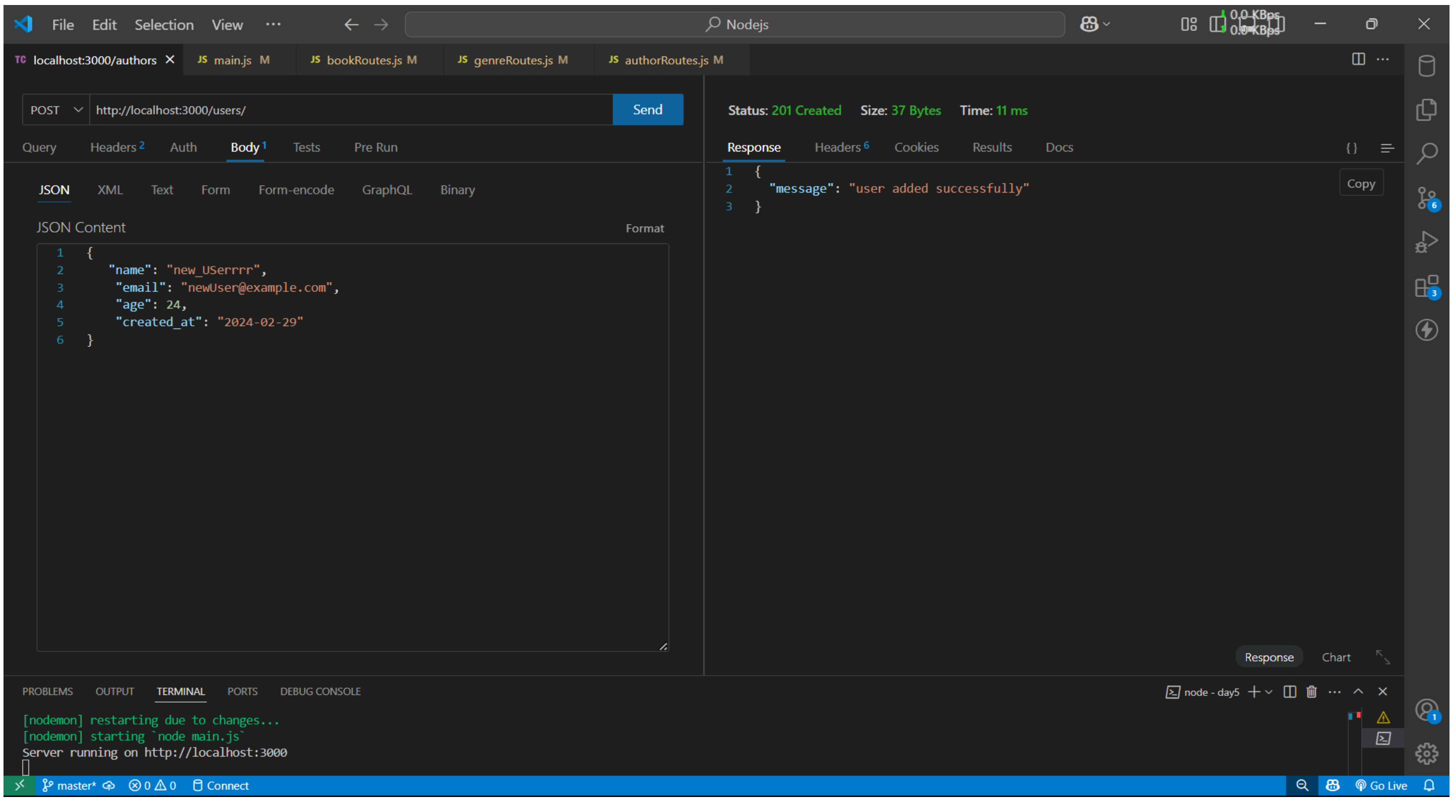
POST http://localhost:3000/genres/

```
{
    "name": "Philosophy",
    "description": "based on indian epics and deities"

}
```

Status: 201 Created Size: 28 Bytes Time: 13 ms

```
{
  "id": 7,
  "name": "Philosophy"
}
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
Server running on http://localhost:3000
```



POST http://localhost:3000/users/

```
{
   "name": "new_USerrrr",
    "email": "newUser@example.com",
    "age": 24,
    "created_at": "2024-02-29"
}
```

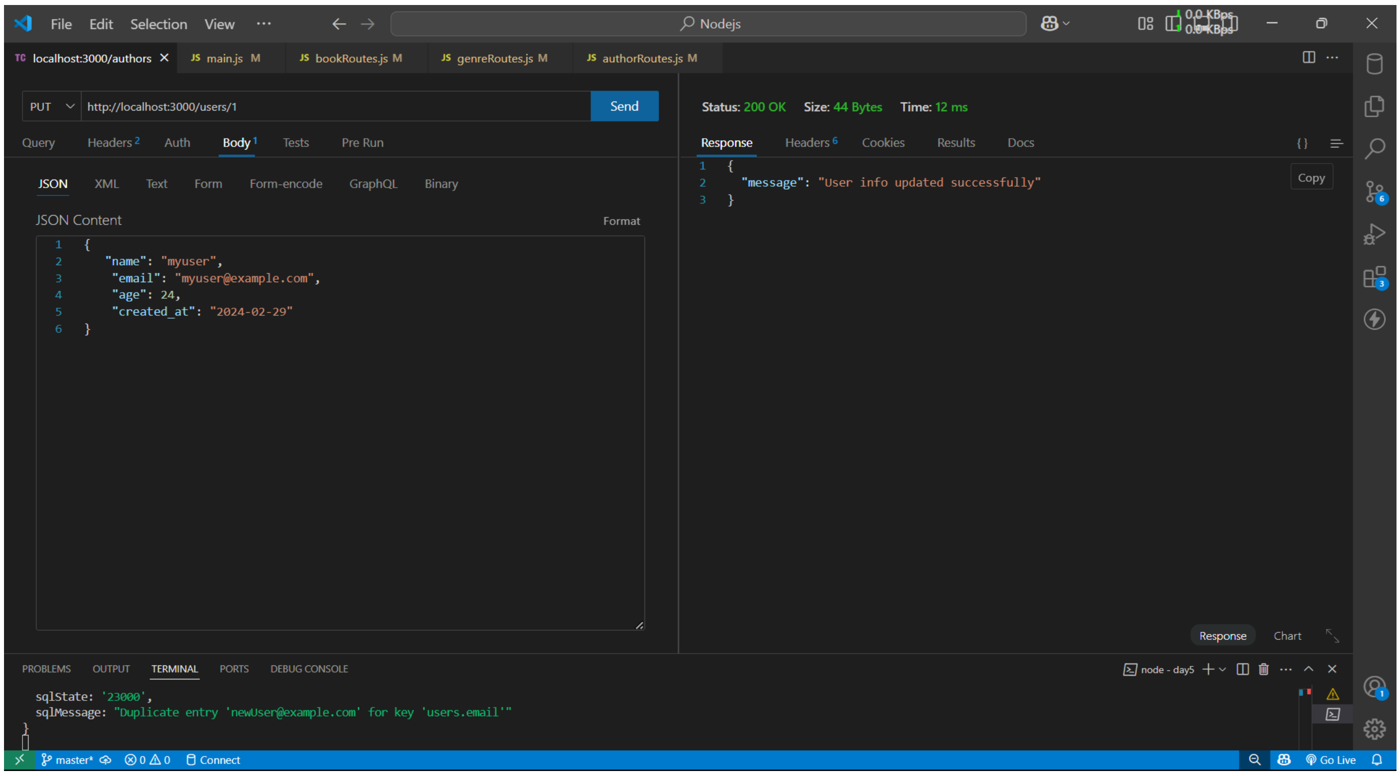
Status: 201 Created Size: 37 Bytes Time: 11 ms

```
{
  "message": "user added successfully"
}
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
Server running on http://localhost:3000
```

Jafar Beldar

Update and delete records by ID.



File Edit Selection View
Nodejs
localhost:3000/authors
main.js M
bookRoutes.js M
genreRoutes.js M
authorRoutes.js M
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
Response Headers 6 Cookies Results Docs
Copy
Response Chart
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
node - day5
master*
Connect
Go Live

PUT http://localhost:3000/users/1 Send

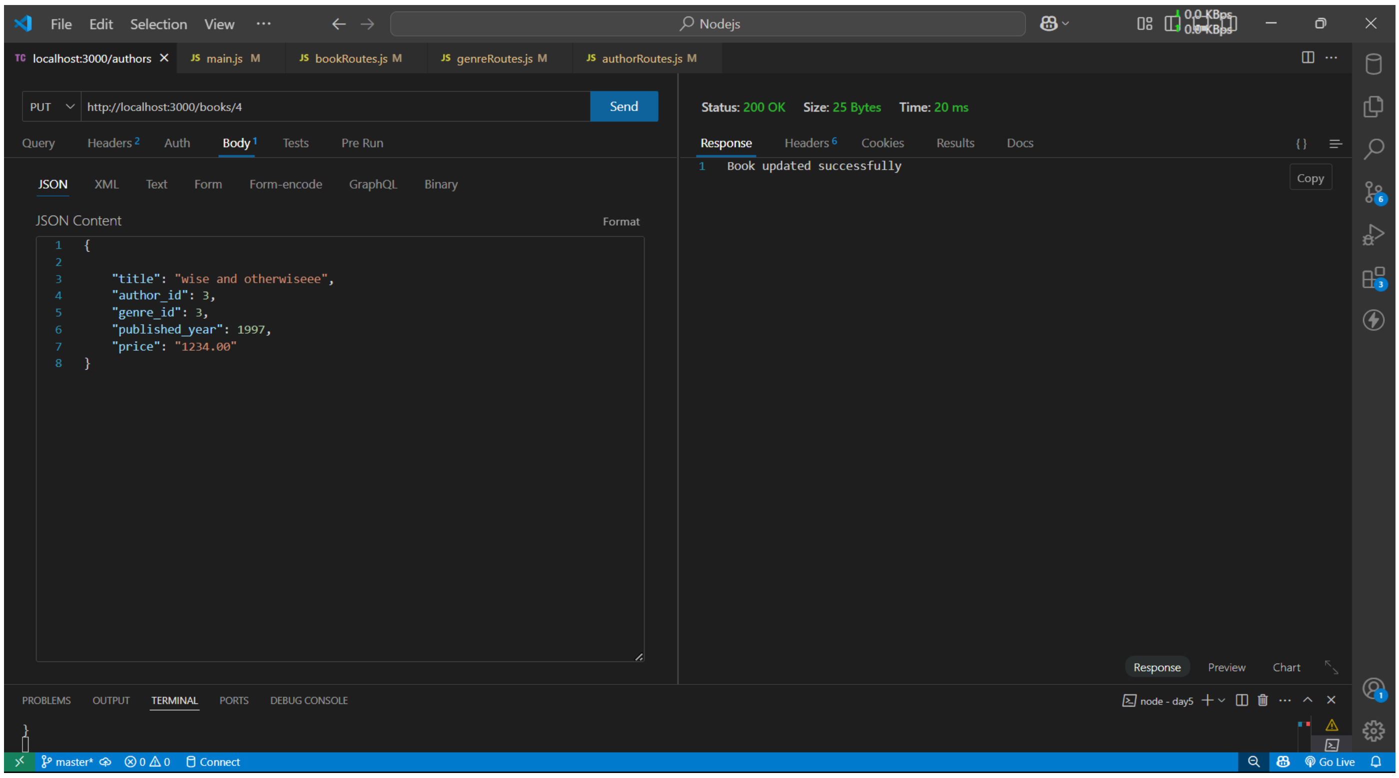
JSON Content Format

```
{
  "name": "myuser",
   "email": "myuser@example.com",
   "age": 24,
   "created_at": "2024-02-29"
}
```

Status: 200 OK Size: 44 Bytes Time: 12 ms

```
{
  "message": "User info updated successfully"
}
```

```
  sqlState: '23000',
  sqlMessage: "Duplicate entry 'newUser@example.com' for key 'users.email'"
}
```



File Edit Selection View
Nodejs
localhost:3000/authors
main.js M
bookRoutes.js M
genreRoutes.js M
authorRoutes.js M
Query Headers 2 Auth Body 1 Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
Response Headers 6 Cookies Results Docs
Copy
Response Preview Chart
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
node - day5
master*
Connect
Go Live

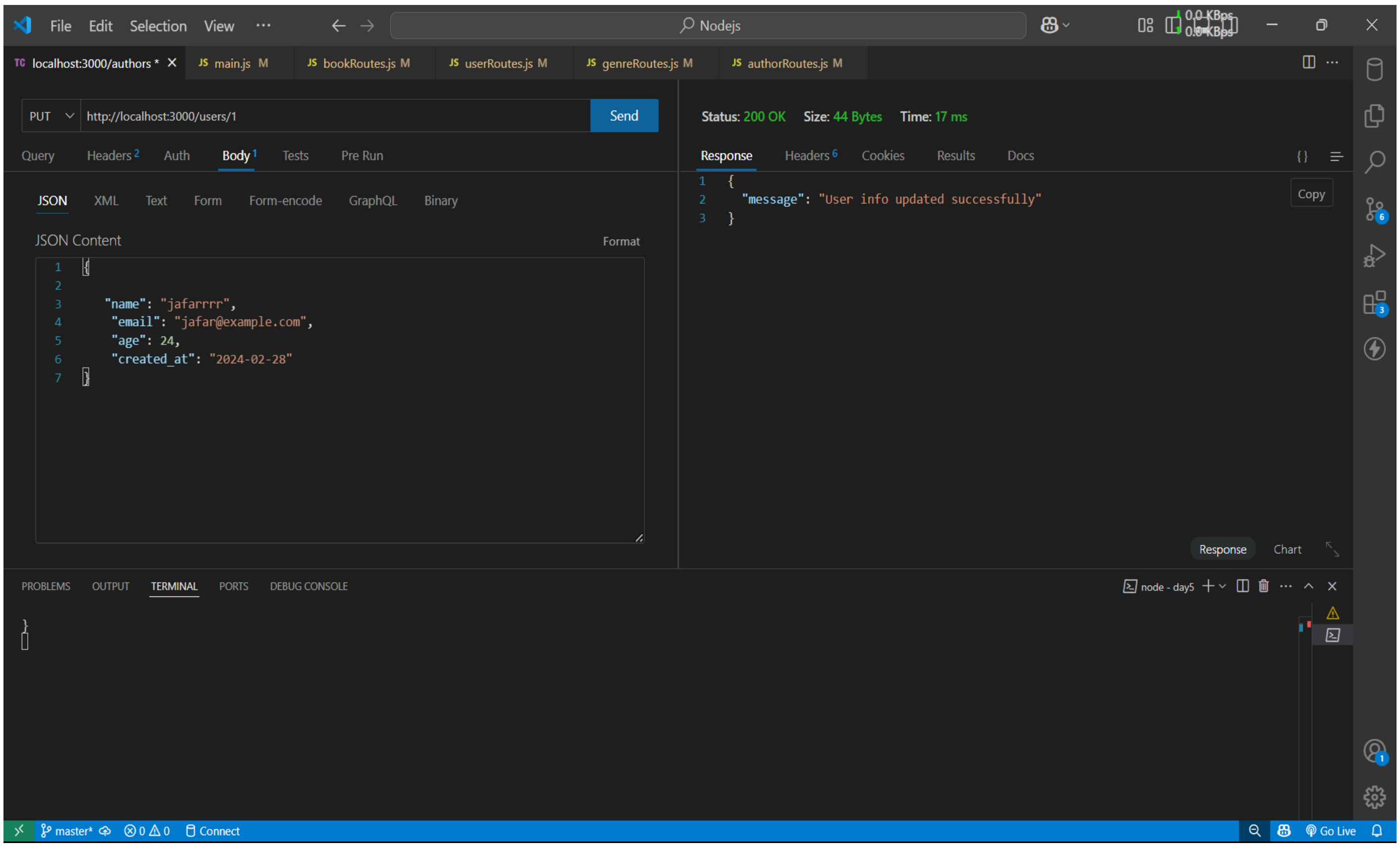
PUT http://localhost:3000/books/4 Send

JSON Content Format

```
{

  "title": "wise and otherwiseee",
  "author_id": 3,
  "genre_id": 3,
  "published_year": 1997,
  "price": "1234.00"
}
```

Status: 200 OK Size: 25 Bytes Time: 20 ms

```
Book updated successfully
```

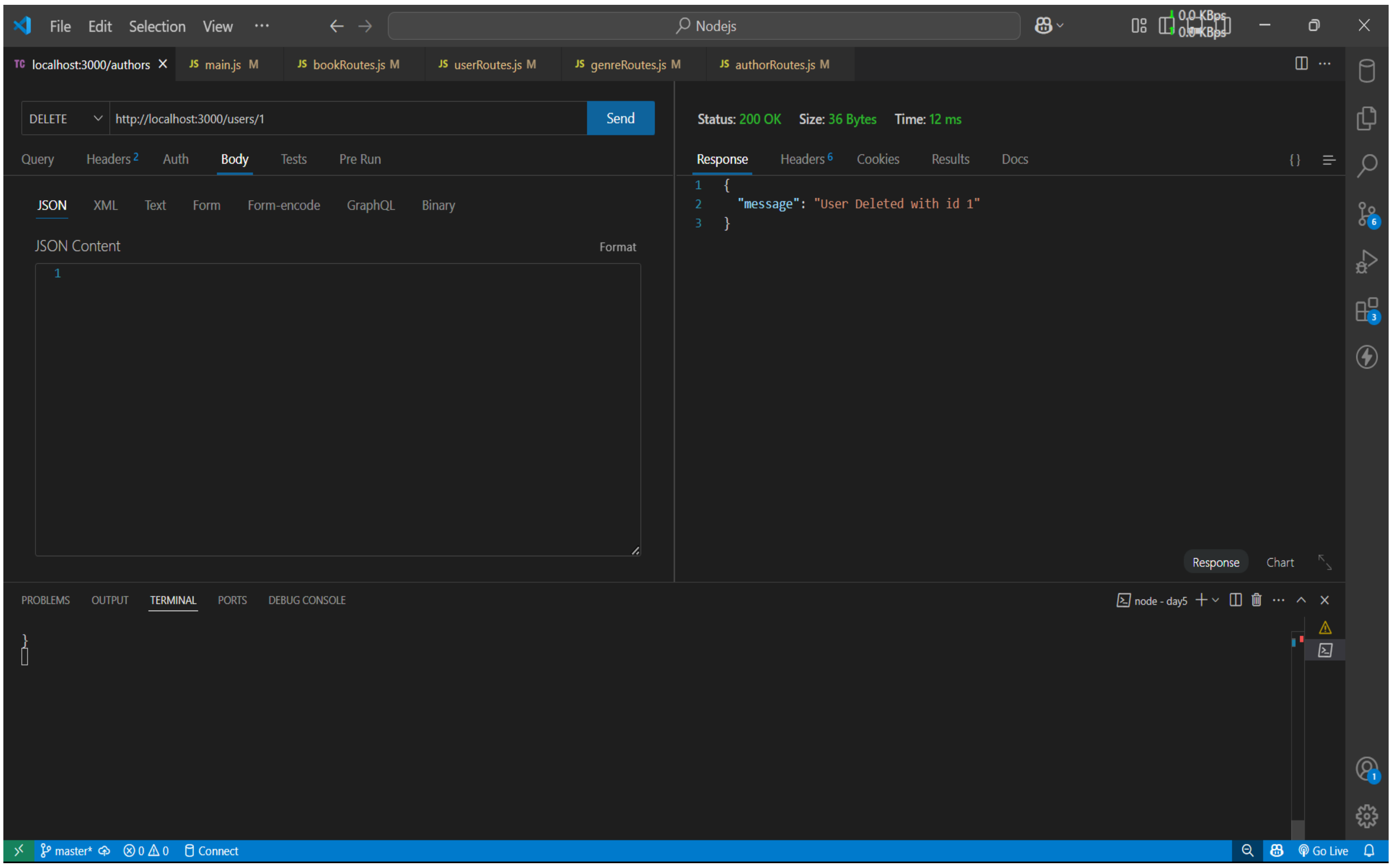


```
PUT  http://localhost:3000/users/1
```

```
{

    "name": "jafarrrr",
     "email": "jafar@example.com",
     "age": 24,
     "created_at": "2024-02-28"
}
```

Status: 200 OK   Size: 44 Bytes   Time: 17 ms

```
{
  "message": "User info updated successfully"
}
```


File Edit Selection View
Nodejs
localhost:3000/authors
main.js M
bookRoutes.js M
userRoutes.js M
genreRoutes.js M
authorRoutes.js M
Send
Query Headers 2 Auth Body Tests Pre Run
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
Response Headers 6 Cookies Results Docs
Response Chart
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
node - day5
master* 0 0 Connect Go Live

DELETE http://localhost:3000/users/1

Status: 200 OK Size: 36 Bytes Time: 12 ms

```
{
  "message": "User Deleted with id 1"
}
```