Task 4

Được làm bởi TEAM B gổm:  
Nguyễn Cảnh Toàn-Nhóm Trưởng

# Mục lục:

Table of Contents

[1. Mục lục: 1](#_Toc157156750)

[2. Bật, tắt thông báo cuộc trò chuyện: 1](#_Toc157156751)

[3. Bật, tắt trạng thái gửi tin nhắn: 1](#_Toc157156752)

[4. Ẩn cuộc trò chuyện bằng mã pin: 1](#_Toc157156753)

[5. Tham gia nhóm bằng link mời: 1](#_Toc157156754)

[6. Quản lí nhóm: 1](#_Toc157156755)

[7. Ghim cuộc trò chuyện: 1](#_Toc157156756)

[8. Tạo ghi chú: 1](#_Toc157156757)

[9. Tạo bình chọn: 1](#_Toc157156758)

[10. Đăng nhập: 1](#_Toc157156759)

[11. Đăng kí: 1](#_Toc157156760)

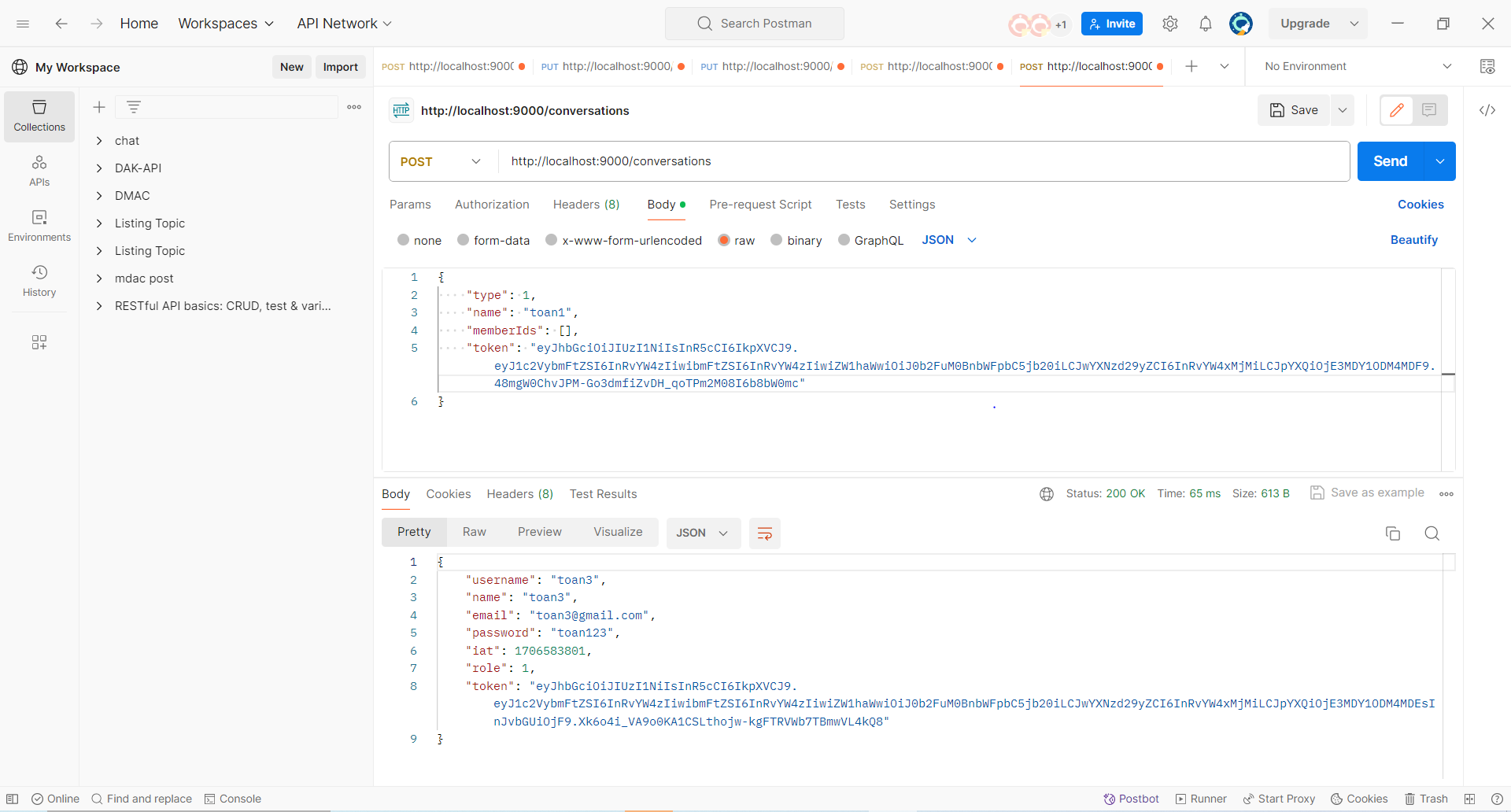
# Tạo group chat:

Trong bài task 4 này, em sử dụng nodejs làm backend và em sử dụng Visual Studio Code làm IDE cho bài của em, còn phần database thì em sử dụng Xamqq để làm control panel và Navicat để lưu dữ liệu.

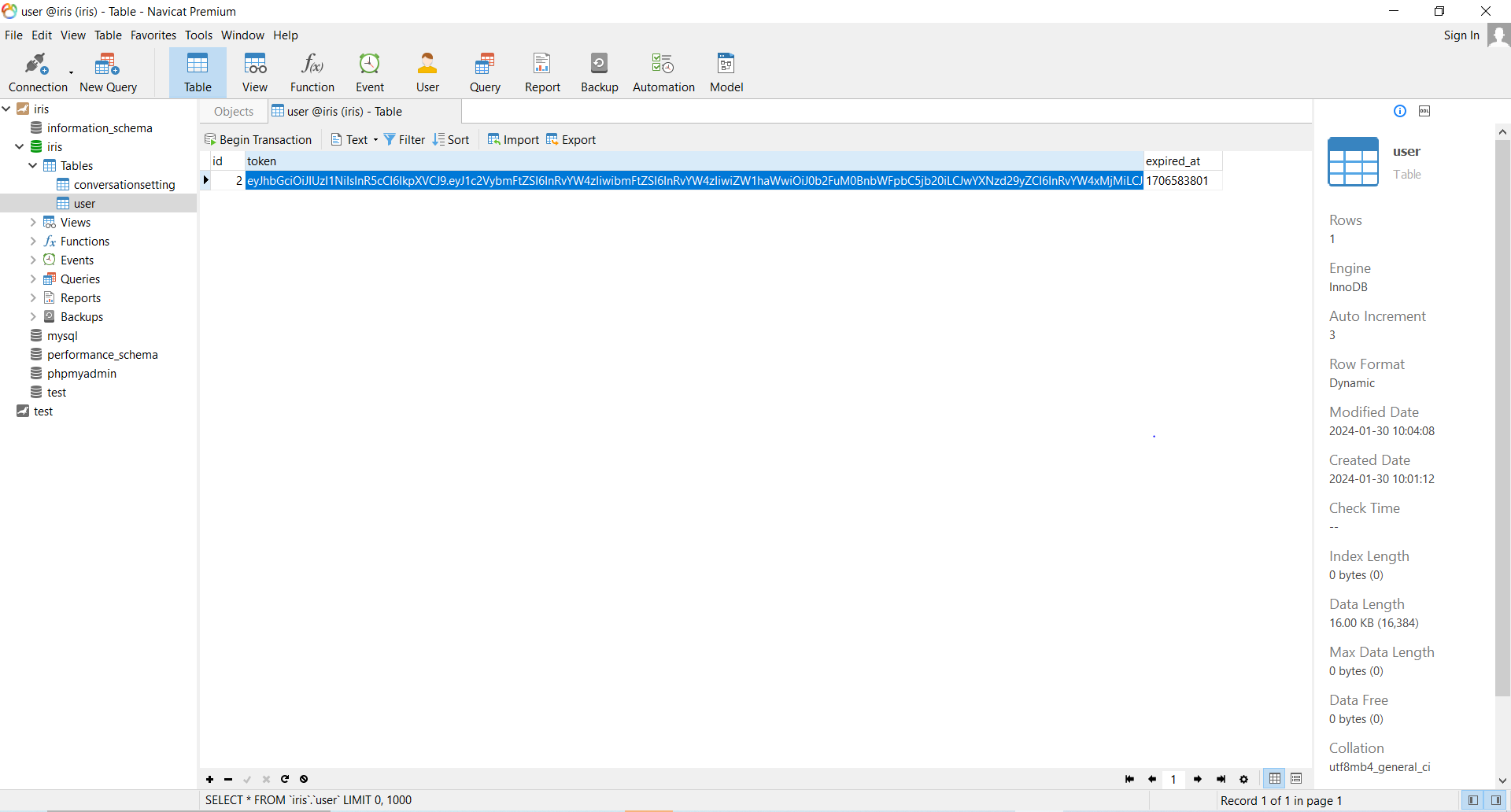
Đầu tiên để làm được các chức năng thì người dùng cần phải đăng nhập và sau đó tạo group chat. Phần đăng kí đăng nhập em sẽ mô tả ở phía dưới còn phần này em sẽ nói về cách làm sao để tạo group chat.

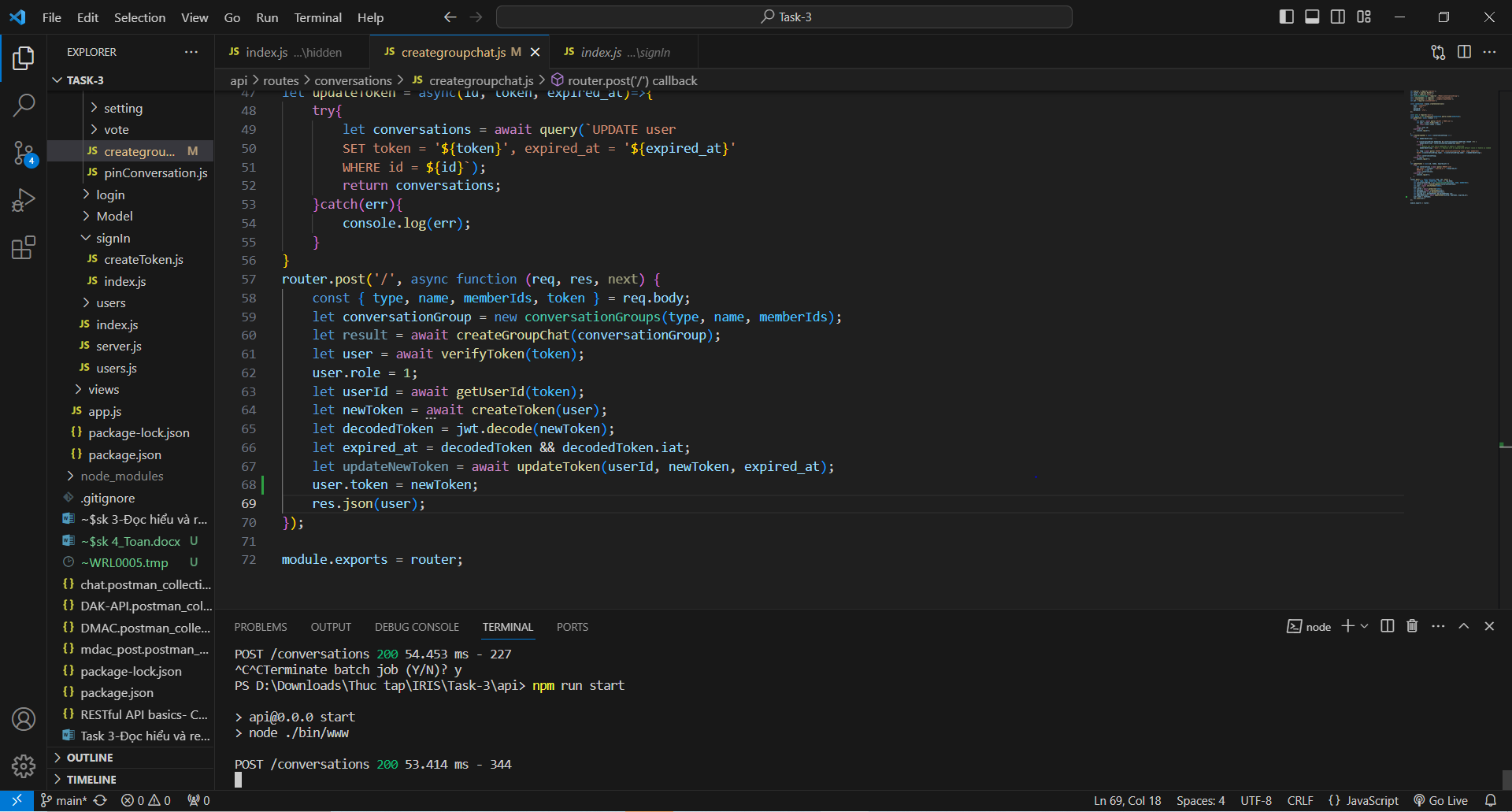
Để tạo được group chat thì người dùng cần phải nhập vào 4 thứ là type, tên group, memberIds là tên những người tham gia, token của người tạo. Và người tạo sẽ mặc định trở thành owner và có thể phân quyền cho những người khác.

Trong code nodejs của em thì đầu tiên em tạo group chat trước và sau đó xác nhận người dùng và gán cho user đó role = 1 (Owner) và sau đó em cập nhật lại cái token mới trên database. Khi thành công thì postman sẽ hiện ra user và token mới:



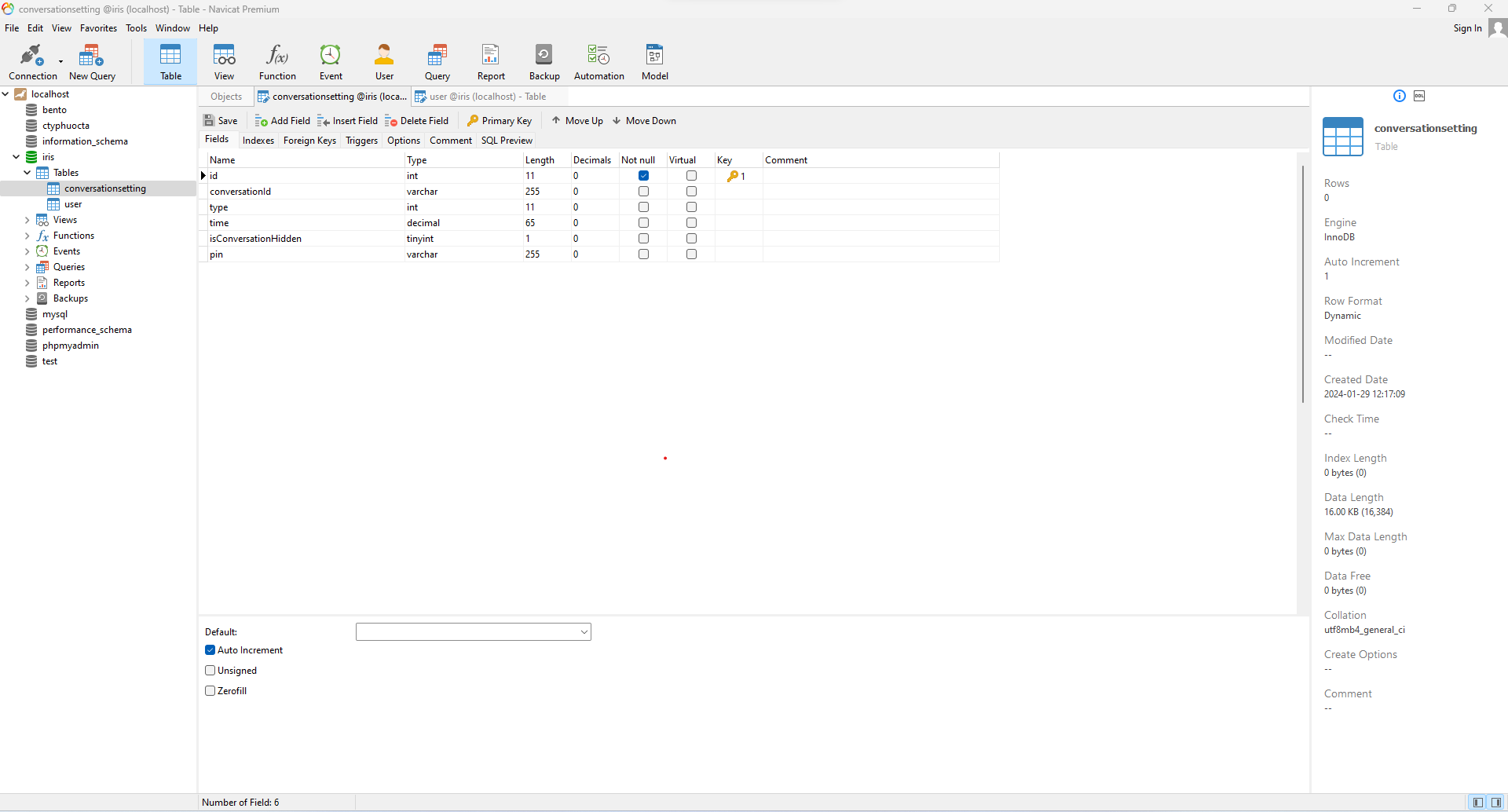
Còn về navicat database username sẽ cập nhật token và expired at. Và đây là code nodejs và database:



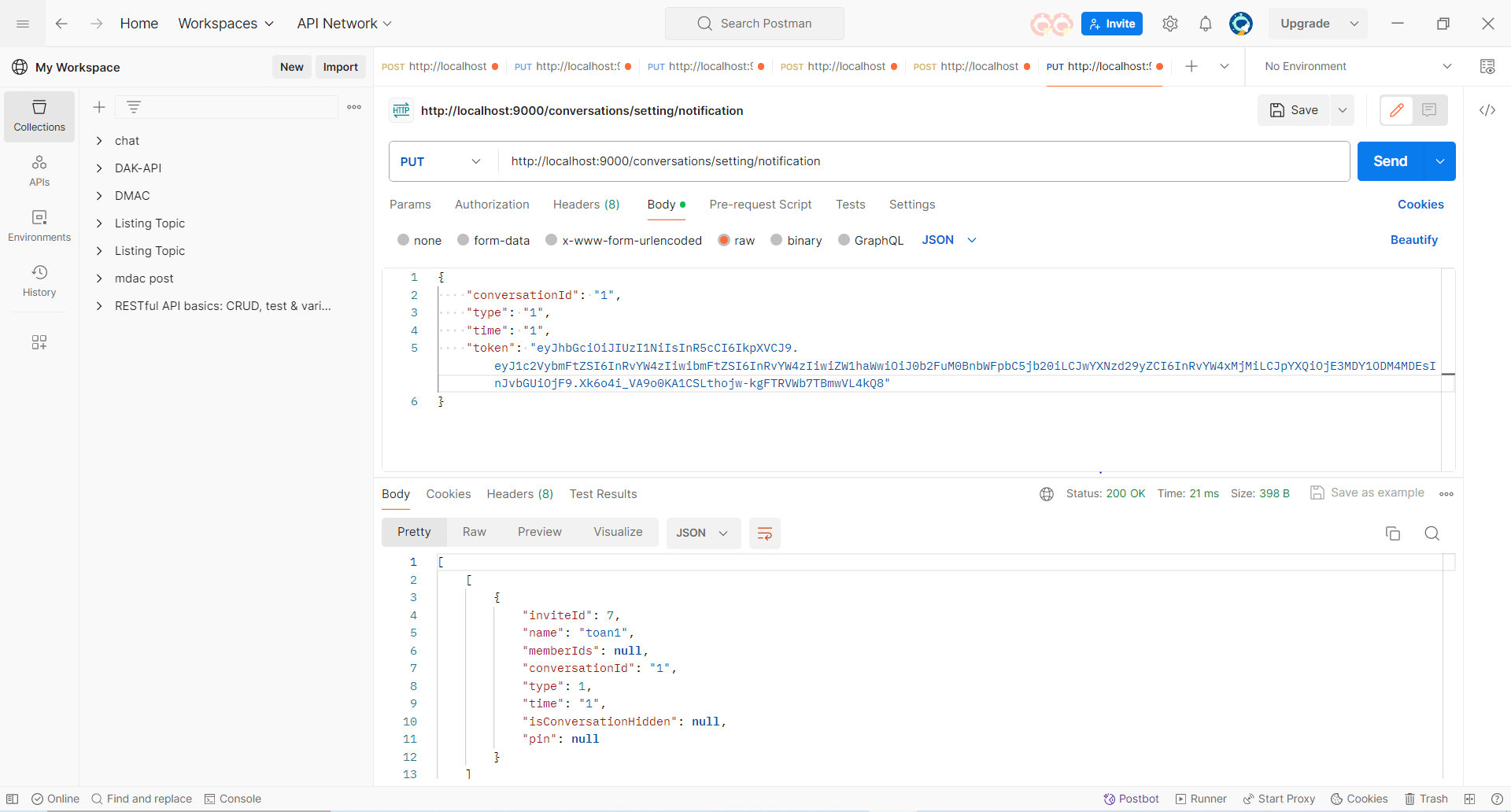


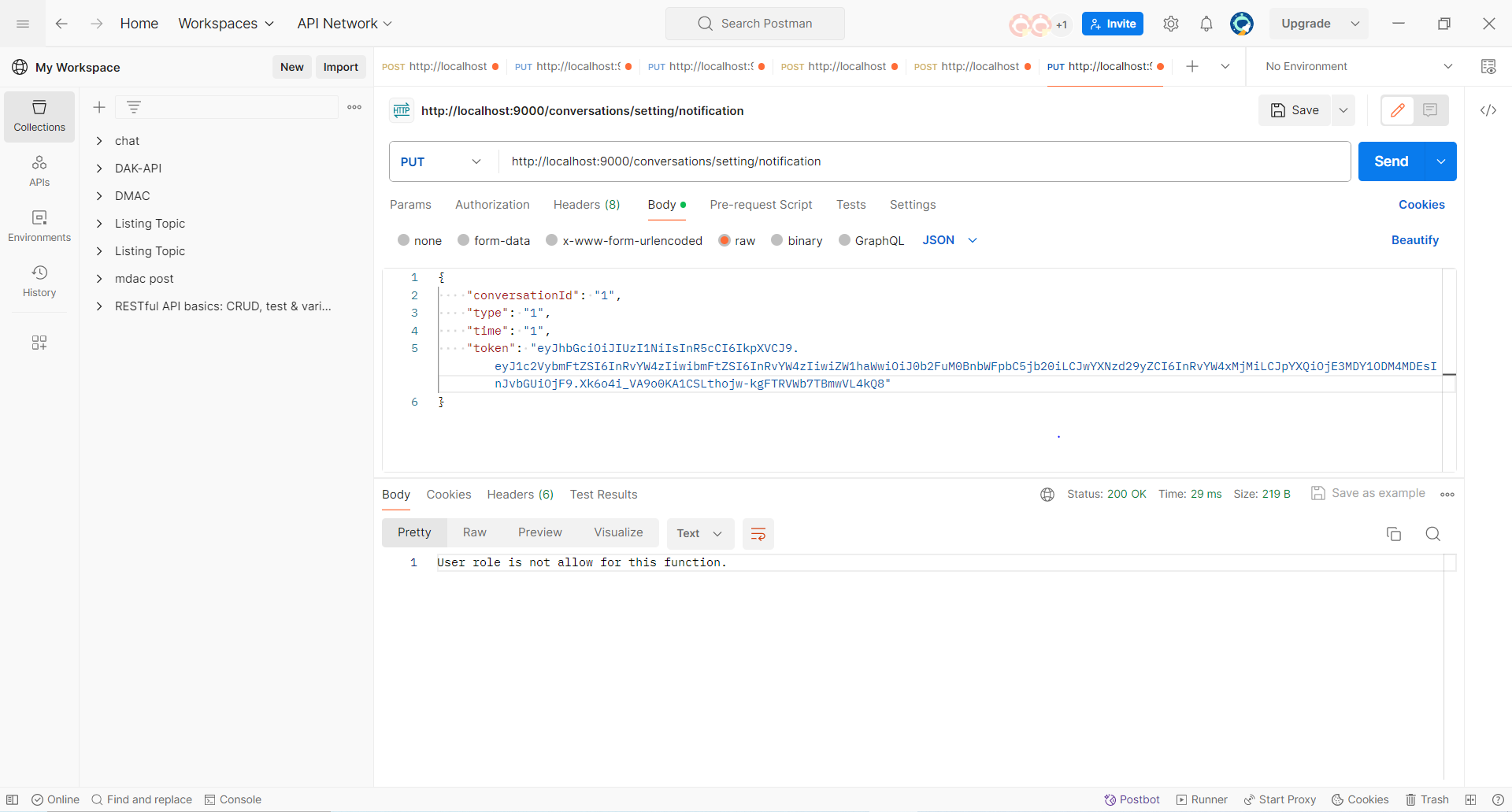
# Bật, tắt thông báo cuộc trò chuyện:

Về phần bật tắt thông báo thì đầu tiên cần phải tạo database có tên là conversationSetting và trong conversationSetting tạo trường id, conversationId, type, time, isConversationHidden, pin. Sau khi hoàn thành thì database sẽ hiển thị như sau:

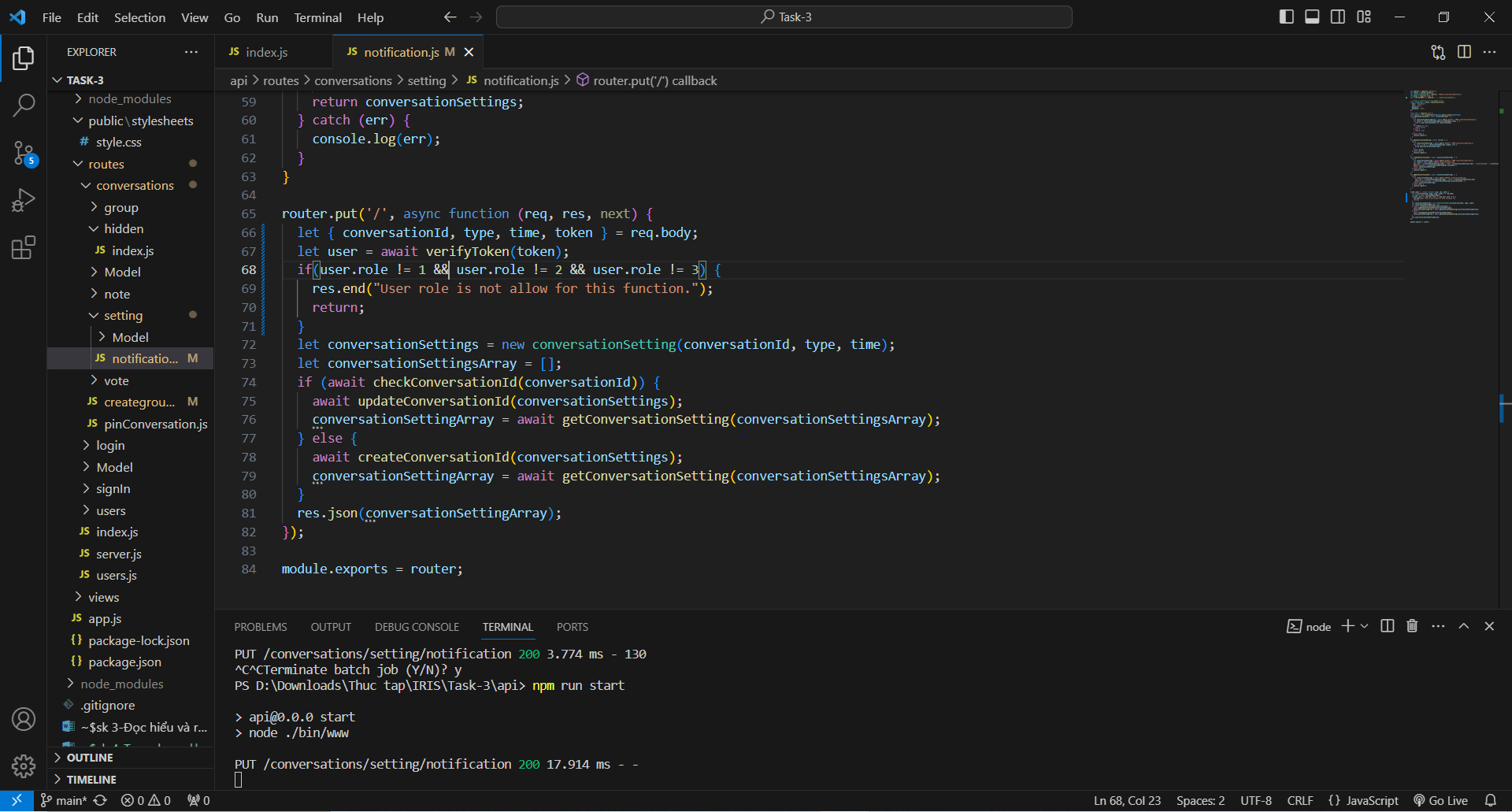


Sau đó trong postman phần body bạn gửi raw body json object có 3 địa chỉ là conversationId, type, time, token. Trong đó token thì bạn phải sử dụng token có role = 1, 2, 3 thì mới đăng nhập được. Sau khi thành công thì postman sẽ hiện ra như sau:





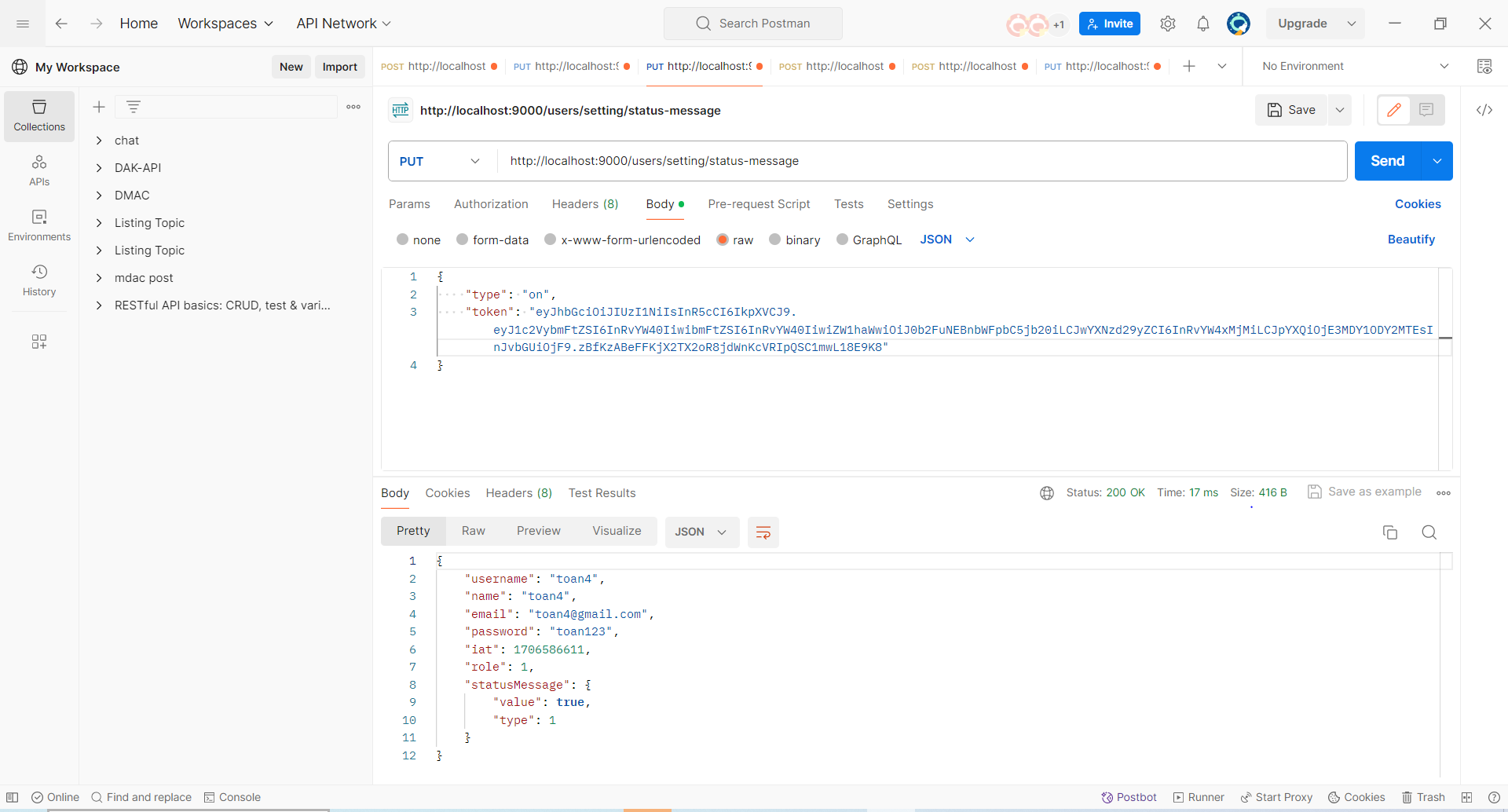
Trong phần code BE thì nodejs làm 3 việc là check thử coi có conversationId đó ko, nếu có thì update lại cái conversations có conversationId cũ còn nếu không thì tạo ra 1 biến mới trong conversationSetting. Đây là code của em:



# Bật, tắt trạng thái gửi tin nhắn:

Đầu tiên để bật tắt trạng thái gửi tin nhắn thì người dùng cần phải nhập vào token có user role là 1, 2, 3 và type bật tắt.

Về phần token của người dùng thì bạn cần phải copy token khi vừa mới đăng kí còn type on để bật, off để tắt là được. Khi thành công thì postman sẽ hiện ra như sau:



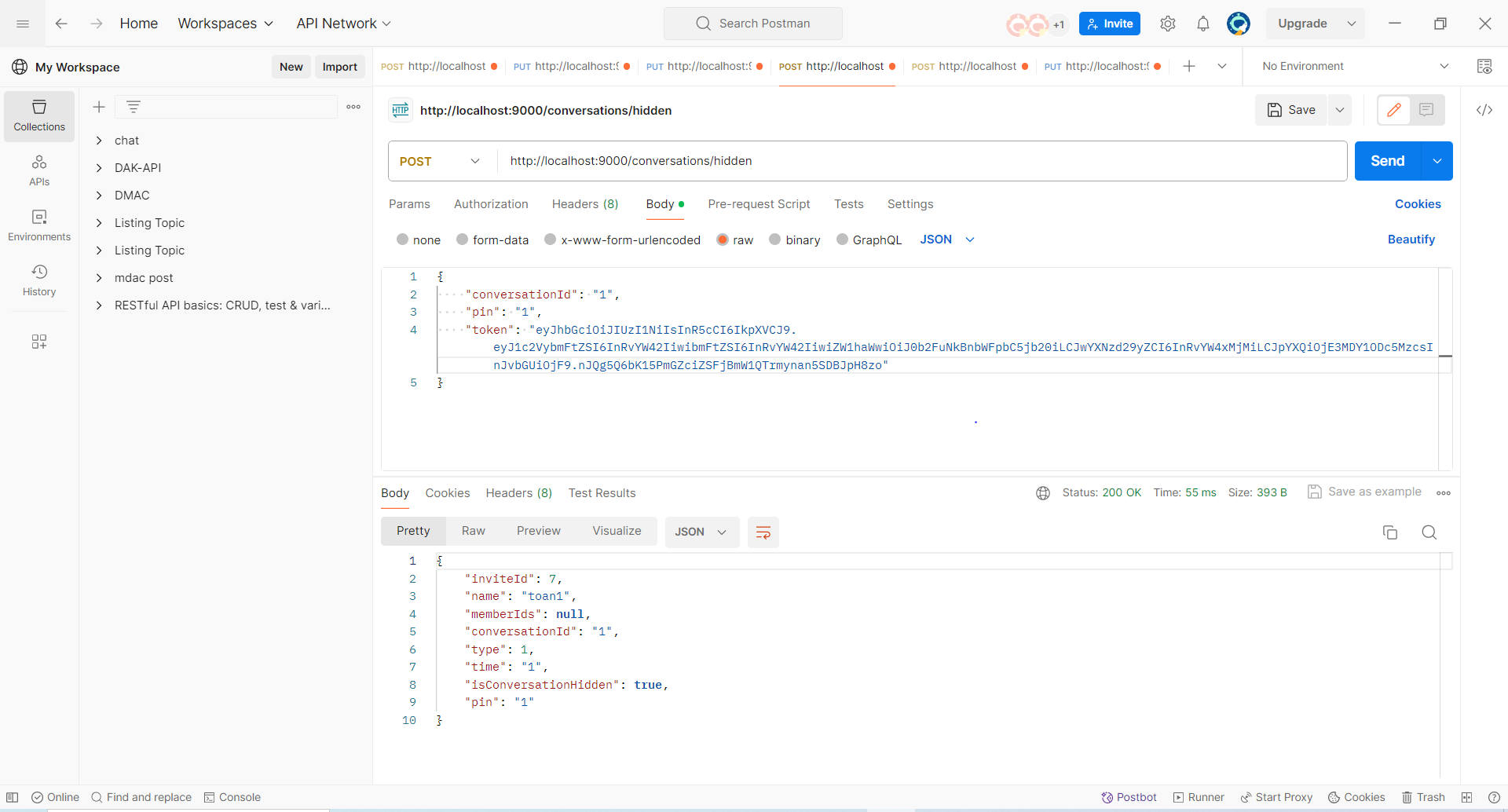
Về phần BE thì em sử dụng jwttoken của phần đăng nhập để lấy được thông tin người đăng nhập token. Sau đó em cho type và value vào thông tin người đăng nhập và lấy id của của cái token đó để cập nhật vào id của cái token đó 1 cái token mới tạo và ngày hết hạn mới. Cụ thể thì code post của em như sau:



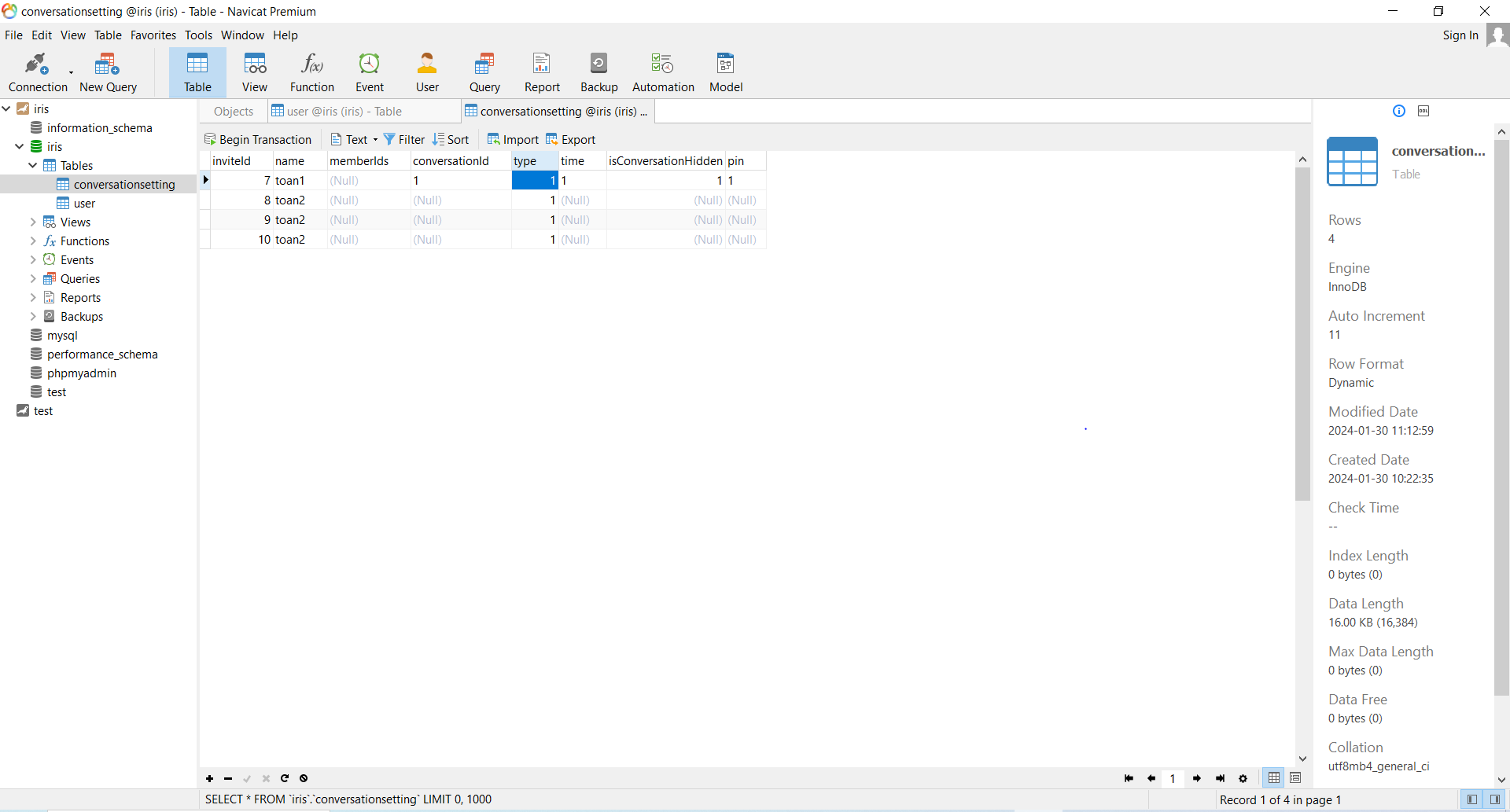
# Ẩn cuộc trò chuyện bằng mã pin:

Đầu tiên để ẩn cuộc trò chuyện bằng mã pin thì người dùng cần phải nhập vào postman conversationId, pin.

Về phần postman thì cần phải nhập conversationId, pin vào body => raw và nhập vào json object như ở dưới. Sau khi thành công thì postman sẽ hiện ra như sau:

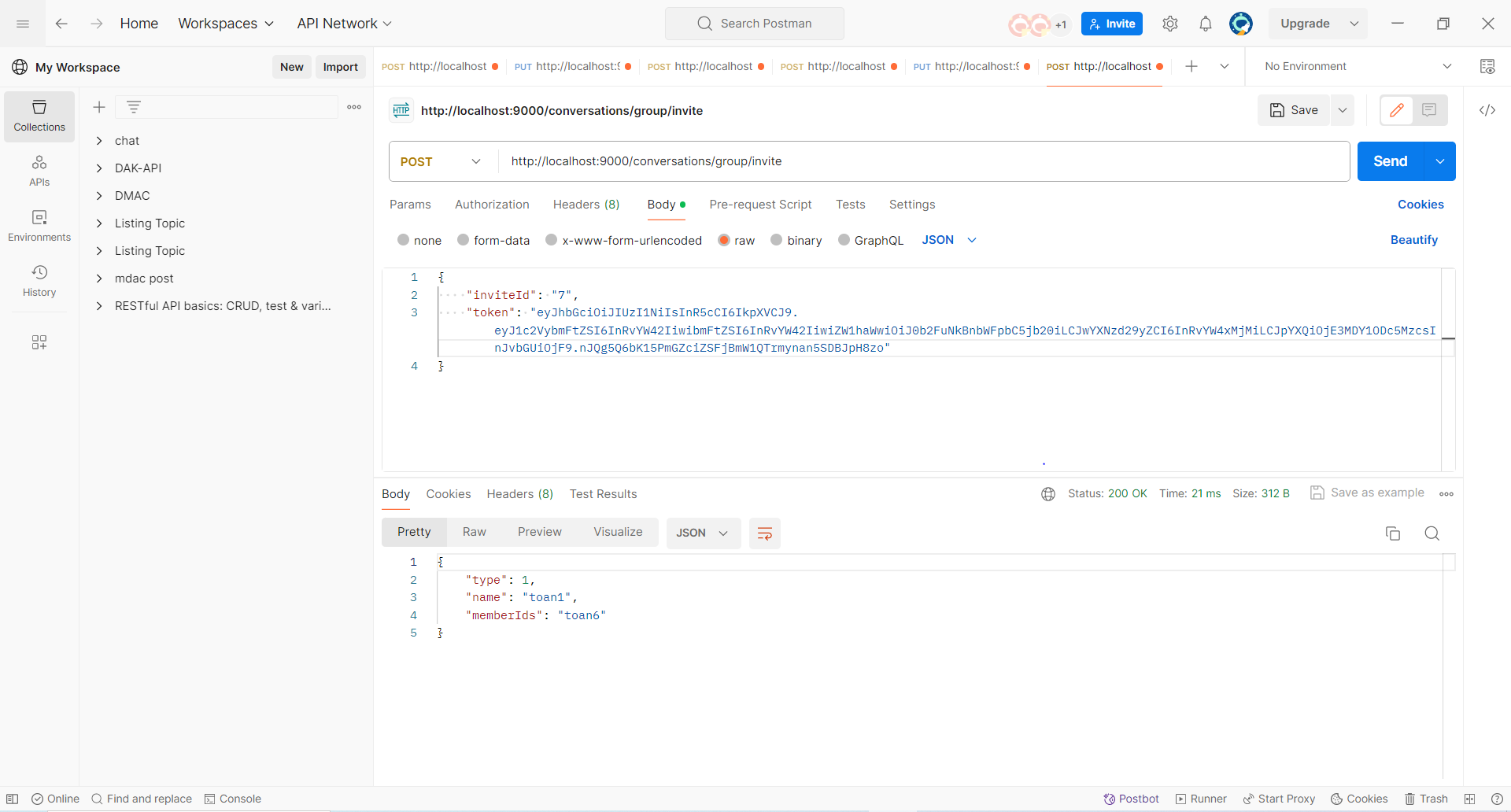


Về phần database thì khi nhận được giá trị thì trường pin sẽ chuyển thành pin mới và isConversationHidden sẽ chuyển từ null sang true hoặc false. Khi thành công thì navicat sẽ hiện ra như sau:

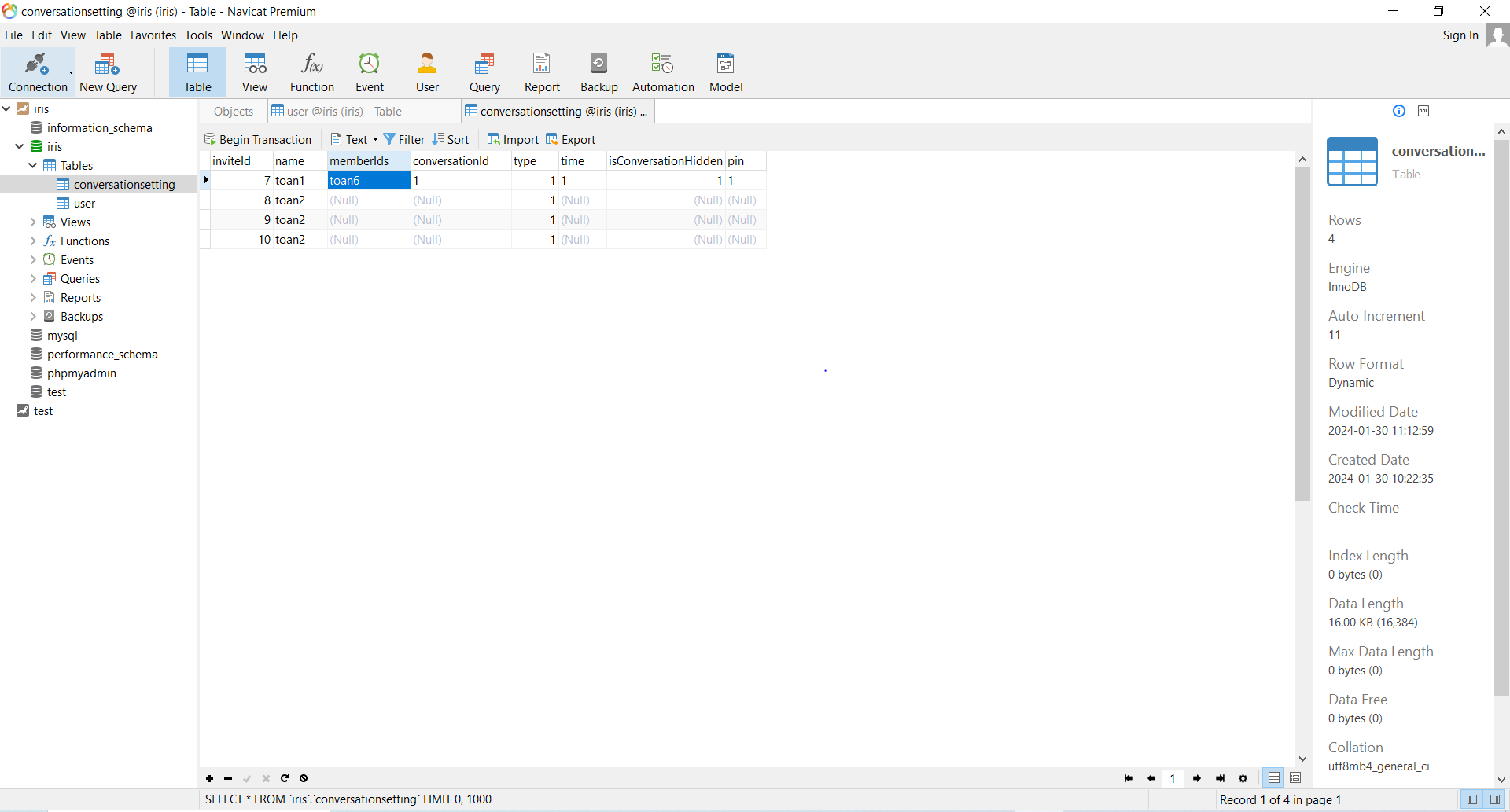


# Tham gia nhóm bằng link mời:

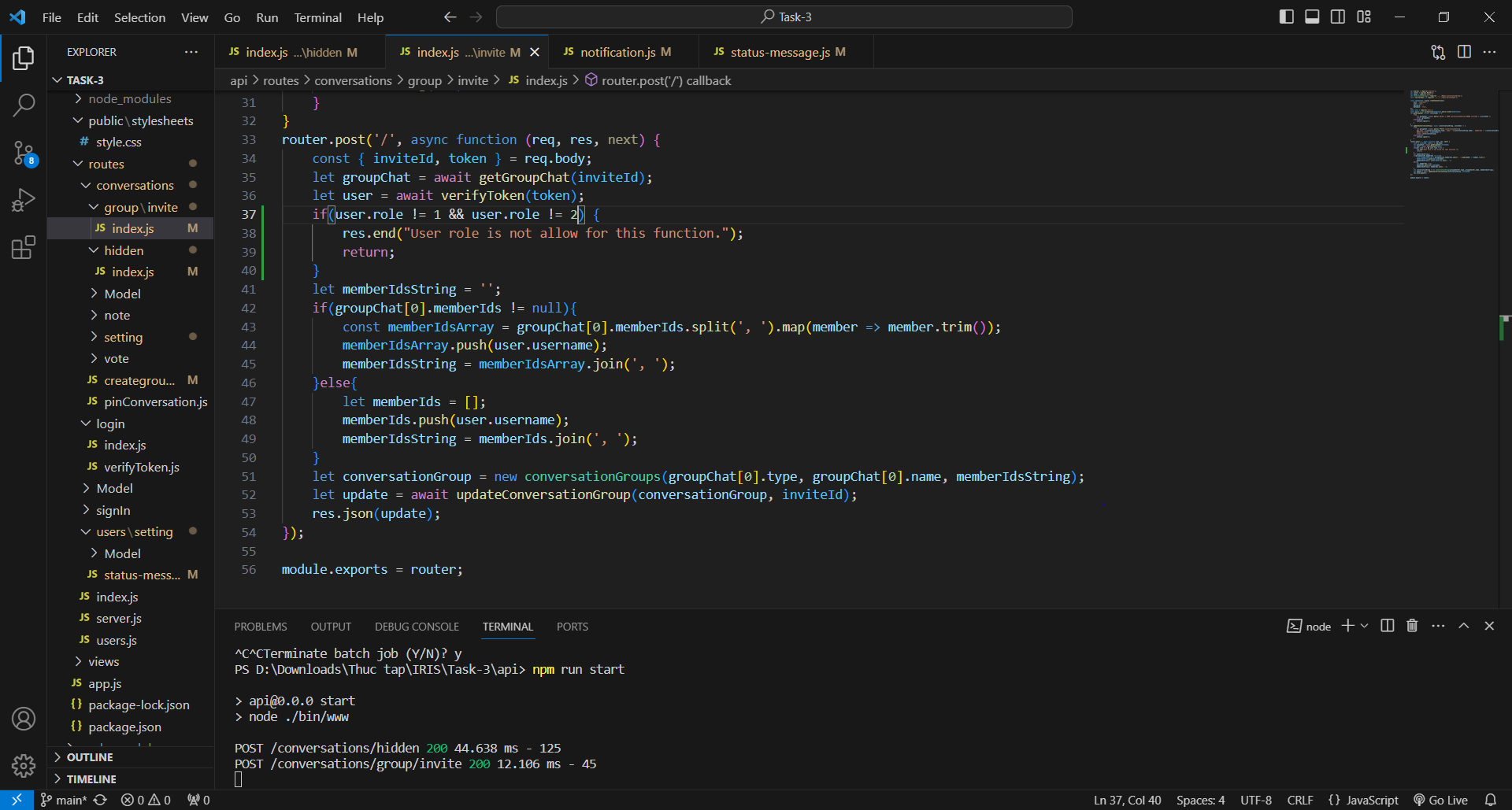
Để tham gia nhóm bằng link mời thì đầu tiên người dùng cần phải nhập vào token của user có role là 1. Sau đó nhập vào inviteIds muốn vô và sau khi thành công thì memberIds sẽ thêm username của bạn vào nhóm. Khi thành công postman sẽ gửi lại như thế này:



Và trong database navicat sẽ hiện ra như thế này:



Còn về phần trong backend thì em sử lý dữ liệu như thế này, đầu tiên em check thử xem token đó role có phải là owner hay admin ko. Nếu sao thì nó sẽ trả về user role còn nếu đúng thì nodejs sẽ lấy cái username của tài khoản và lấy các memberIds cũ ra và thêm username của bạn vào memberIds và lưu vào database. Đây là code của em:

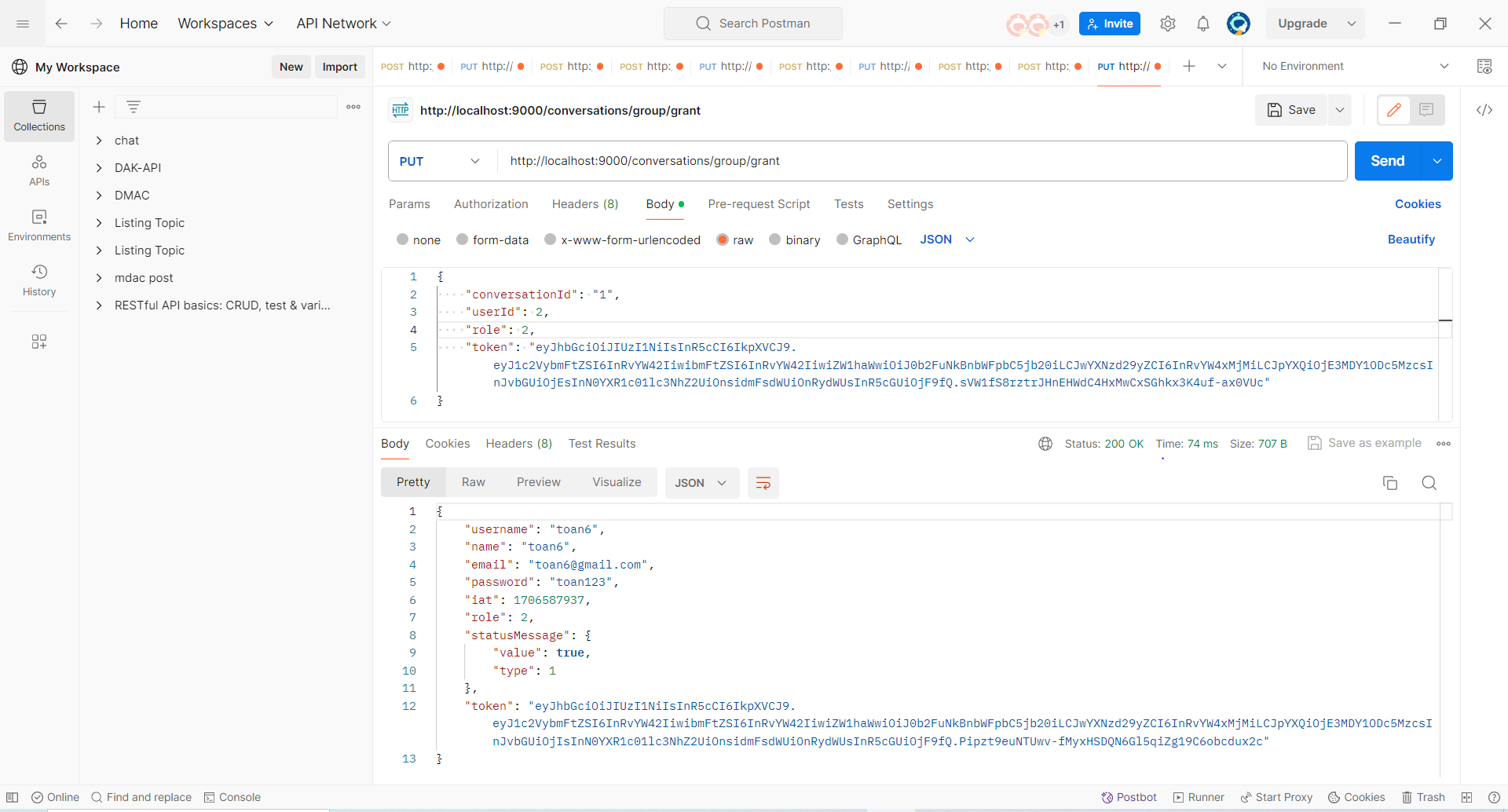


# Quản lí nhóm:

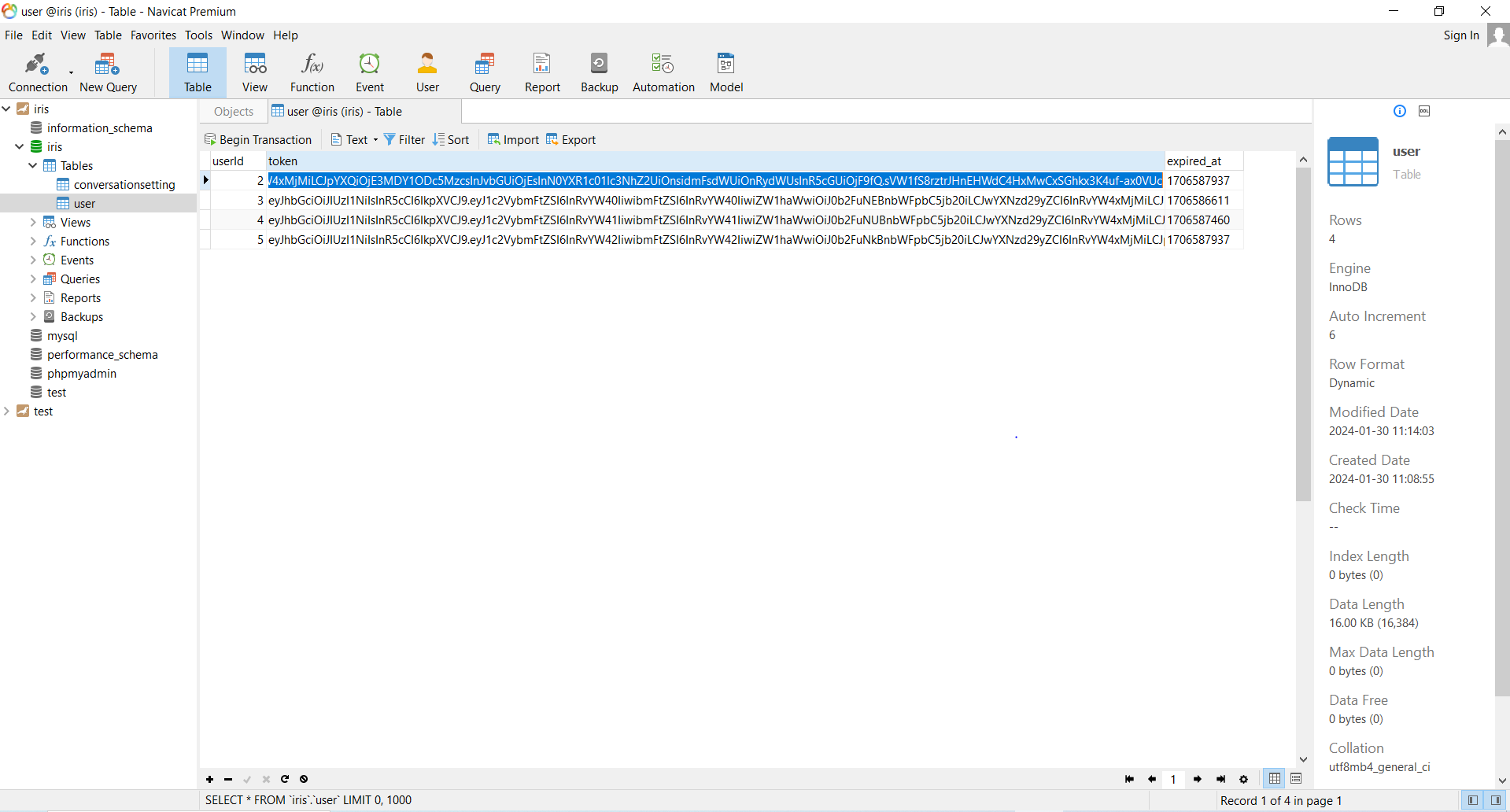
Trong phần quản lí nhóm có nhiều phần khác nhỏ hơn nên em chia các phần như sau:

## Phân quyền cho thành viên trong nhóm:

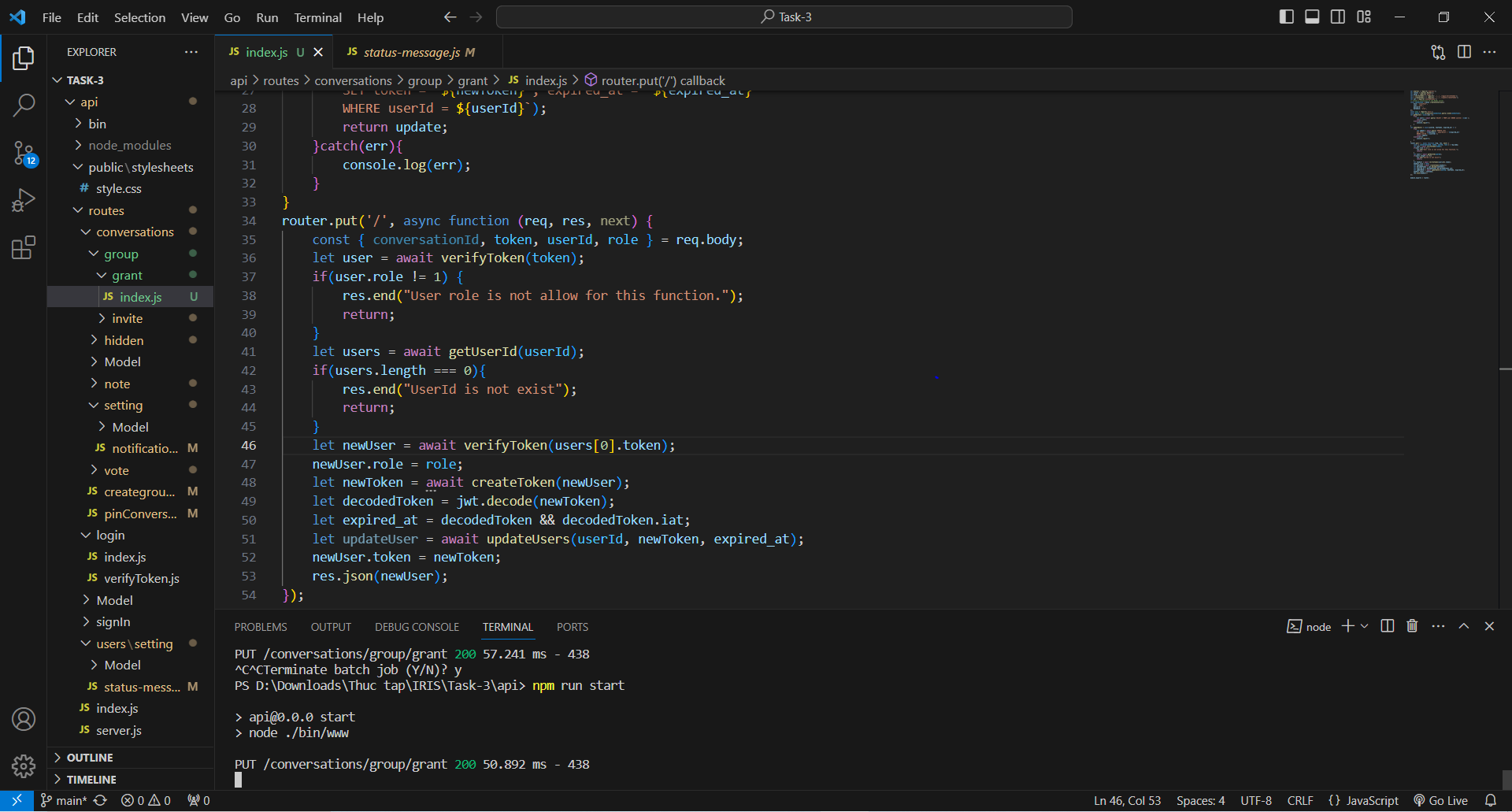
Đầu tiên để phân quyền cho các thành viên trong nhóm thì token của bạn phải có role là 1. Sau đó bạn phải nhập userId, role và conversationId và userId là id của user, role là role mà bạn muốn thay đổi. Sau khi thành công postman sẽ gửi status 200 như sau:



Về phần database navicat thì nó sẽ cập nhật token và expired\_at:

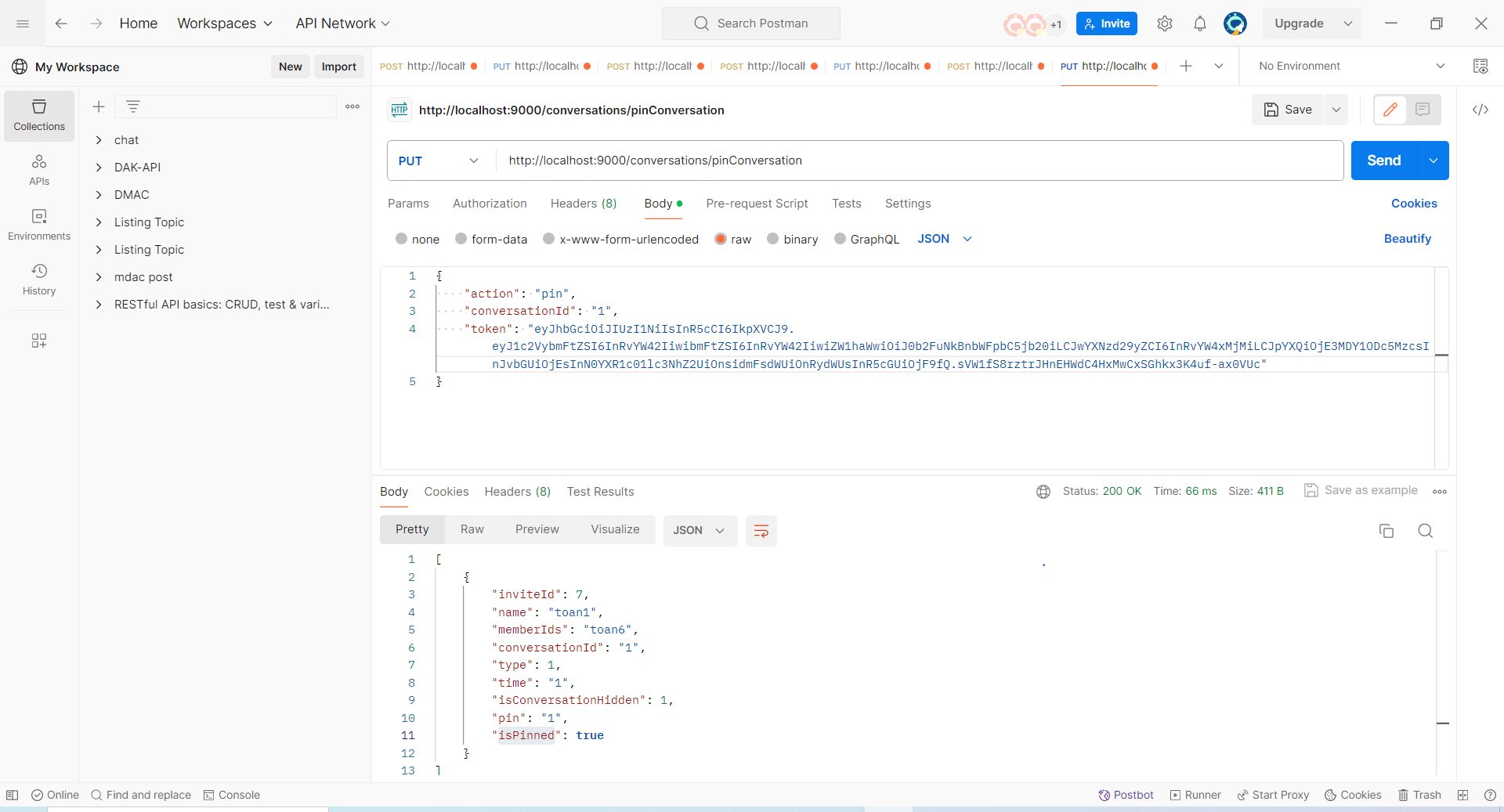


Về phần backend thì em xài nodejs để kiểm tra xem token mới cho có role bằng 1 không nếu có thì lấy user dựa theo userId đã cho và sau đó giải mã token và gắn cho token mới role bằng giá trị mới cho và cập nhật lại token cũ. Đây là code của em:

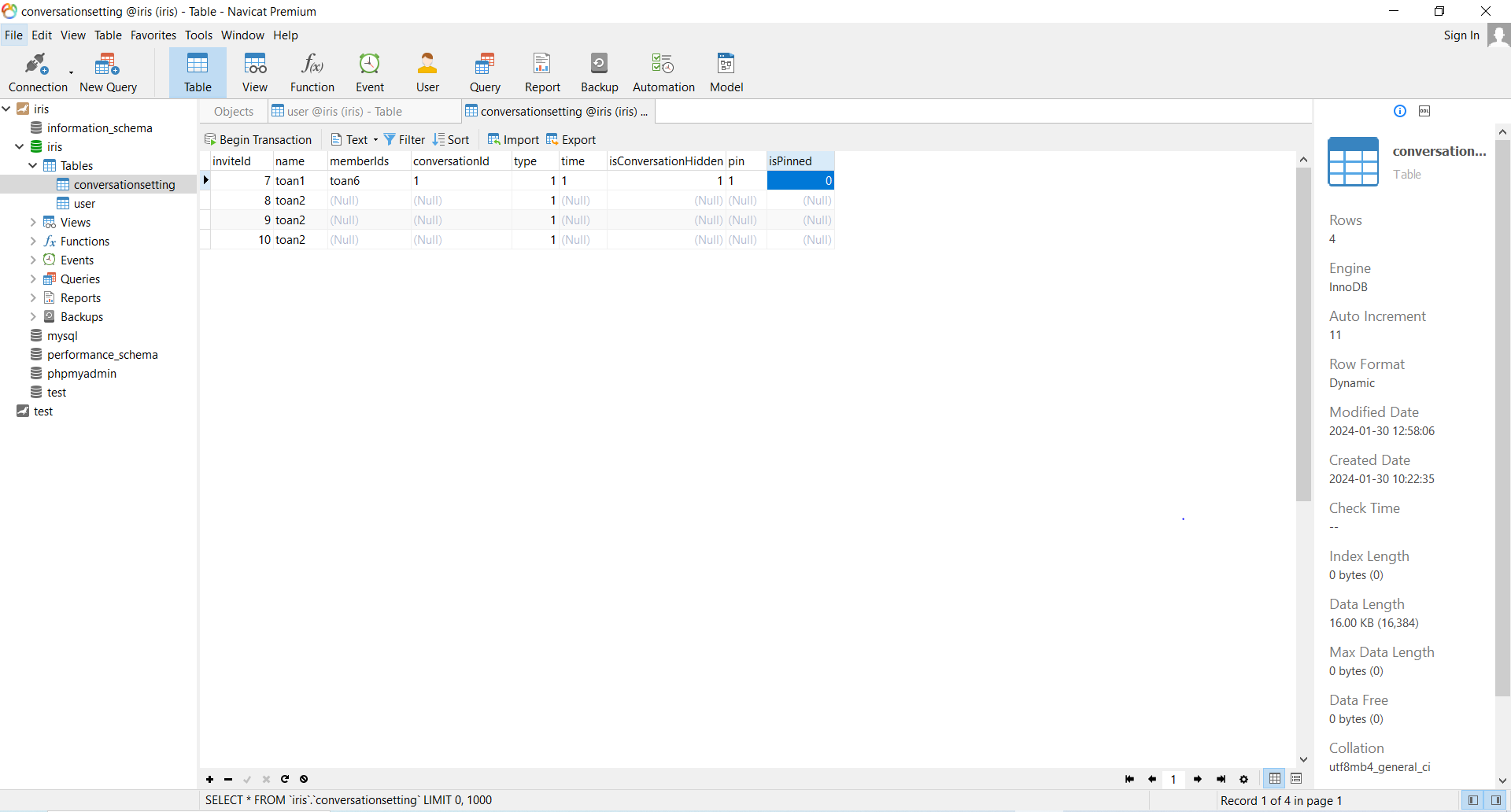


# Ghim cuộc trò chuyện:

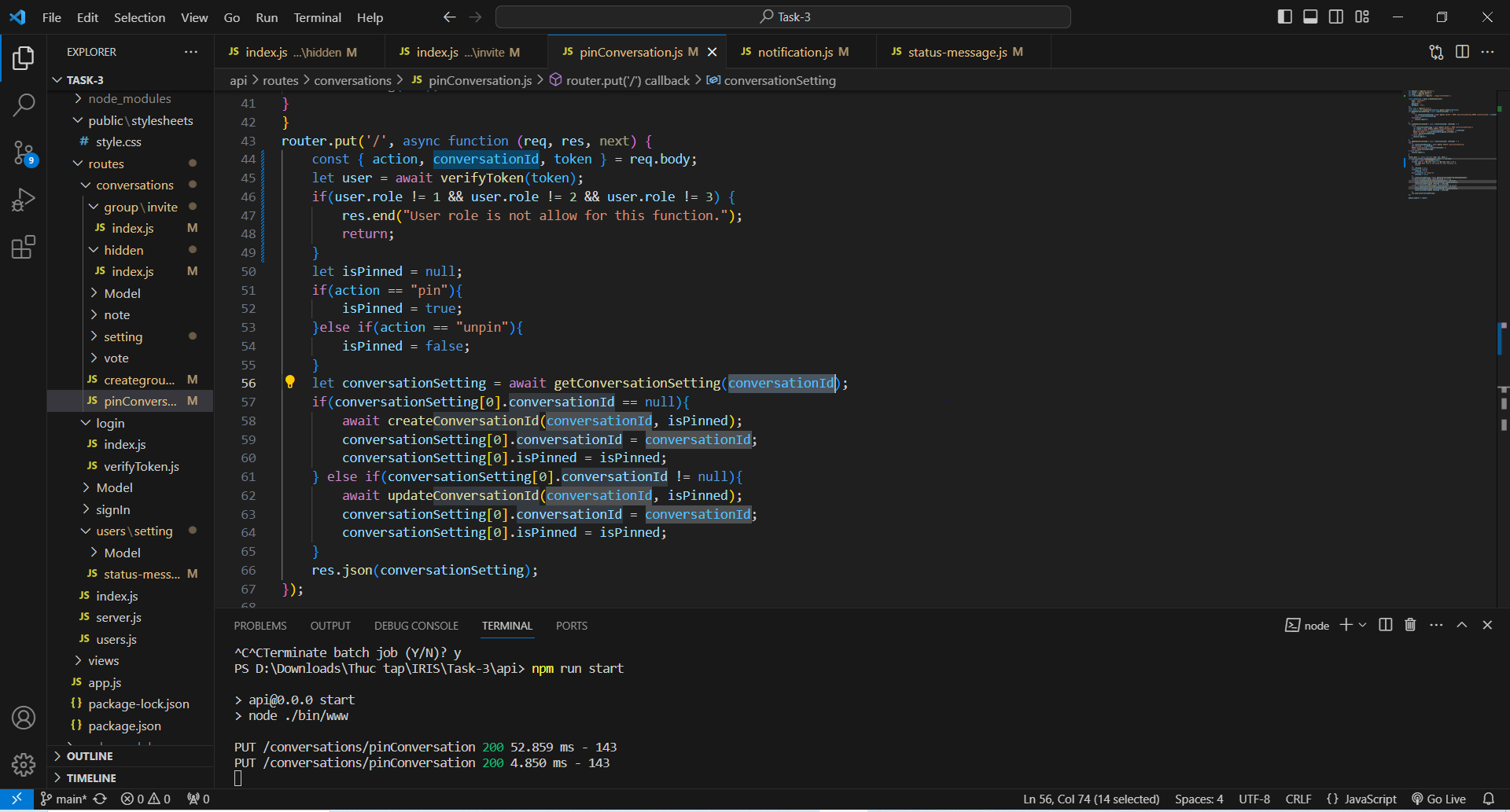
Đầu tiên để ghim cuộc trò chuyện thì người dùng cần phải nhập đúng token có role là 1, 2, 3. Sau đó người dùng nhập là ghim hay ko ghim vào action và nhập cái conversationId của group chat. Sau khi cập nhật xong postman sẽ hiện ra như thế này:



Và cái database của navicat sẽ hiện 0 như thế này:

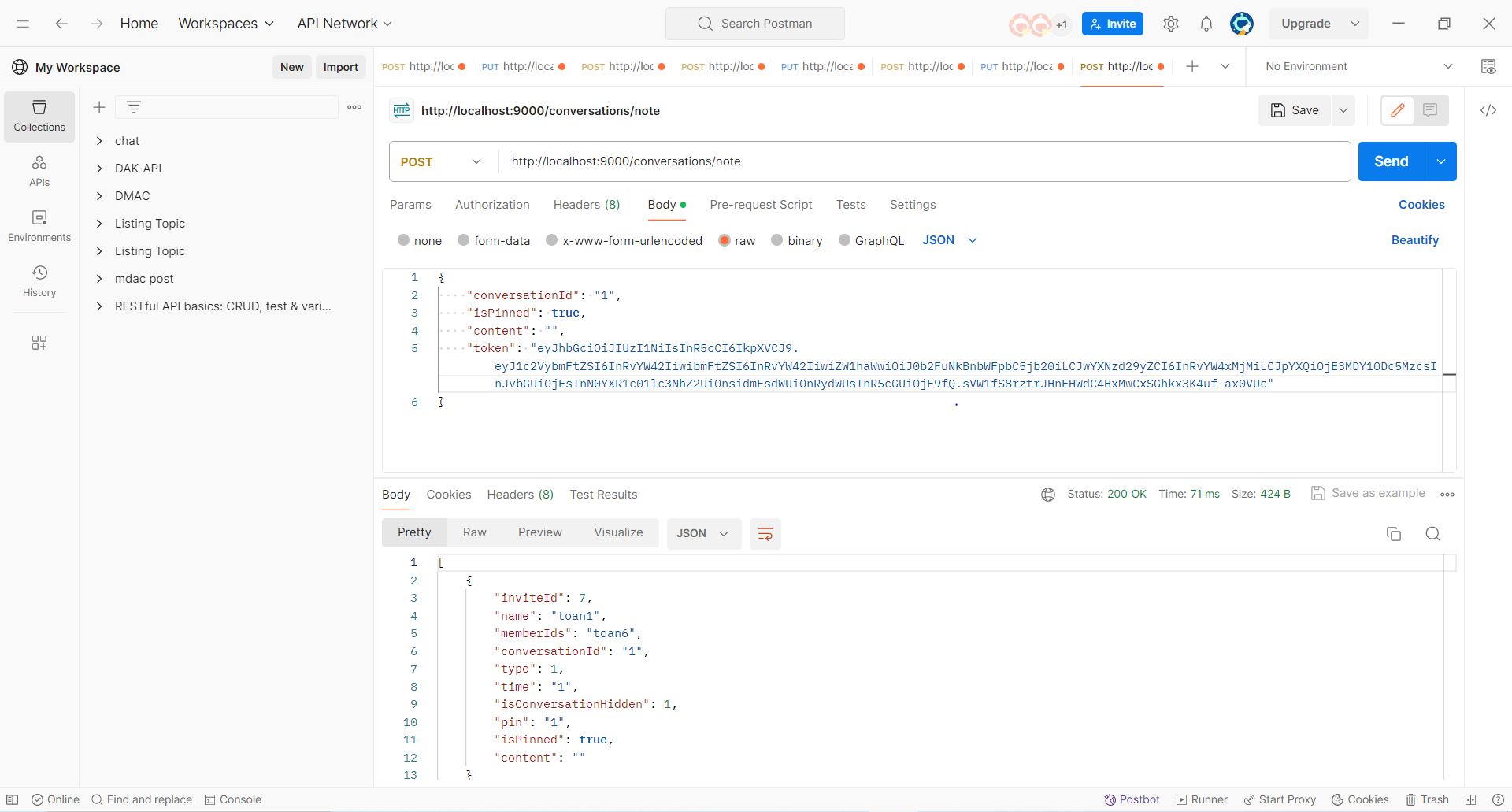


Còn về backend thì em sử dụng nodejs để lấy conversationSetting có conversationIds đó. Sau đó em thử kiểm tra conversationId có trong database hay ko nếu không có thì em cập nhật theo inviteId còn nếu có thì em cập nhật vào conversationIds. Code em thì như thế này:

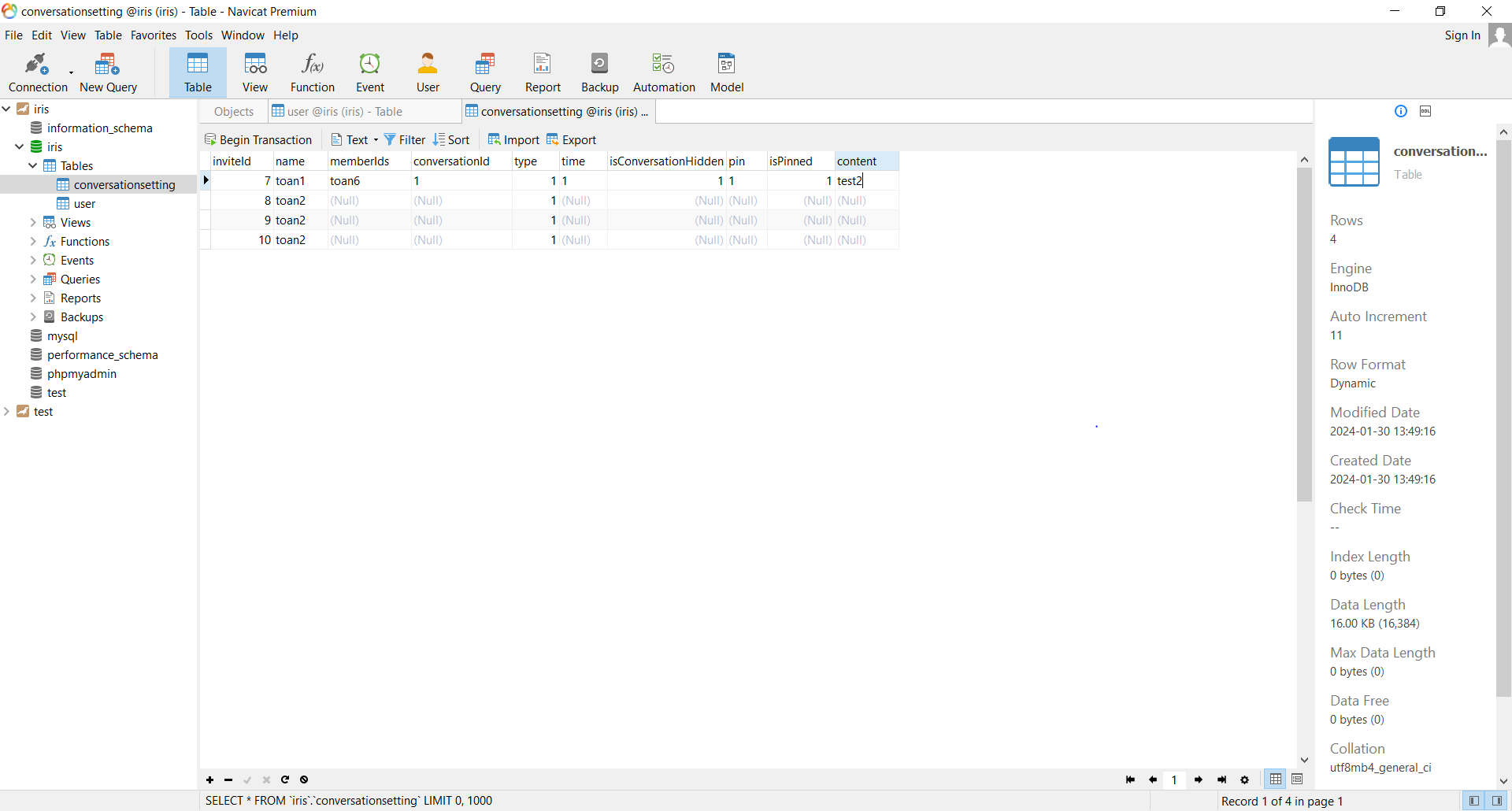


# Tạo ghi chú:

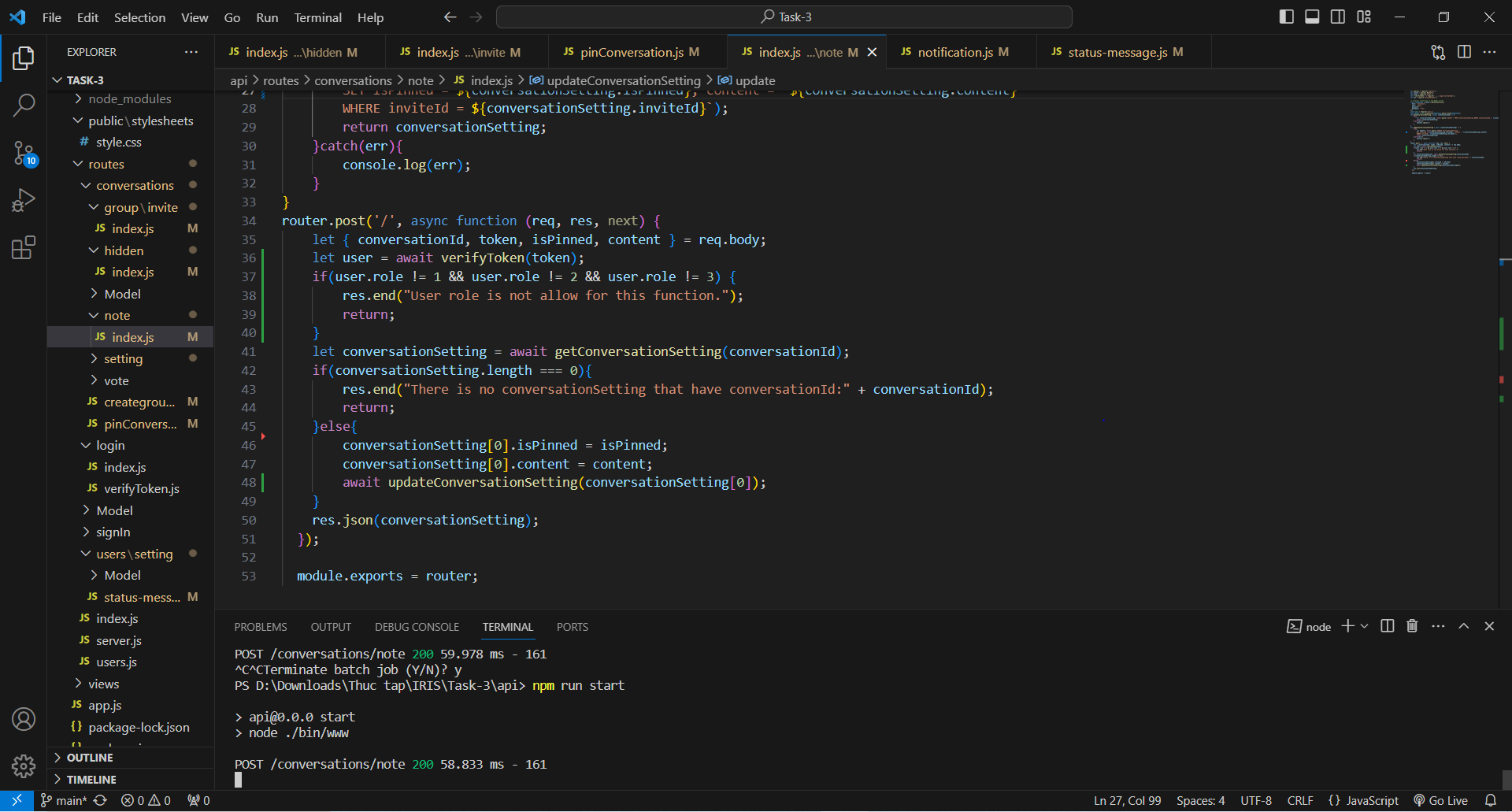
Đầu tiên để tạo được ghi chú thì người dùng cần phải nhập vào token của user có role bằng 1, 2, 3. Sau đó bạn cần phải nhập vào conversationId, isPinned, content. Sau khi thành công thì postman sẽ gửi ra user như sau:



Và database navicat sẽ hiện ra như thế này:

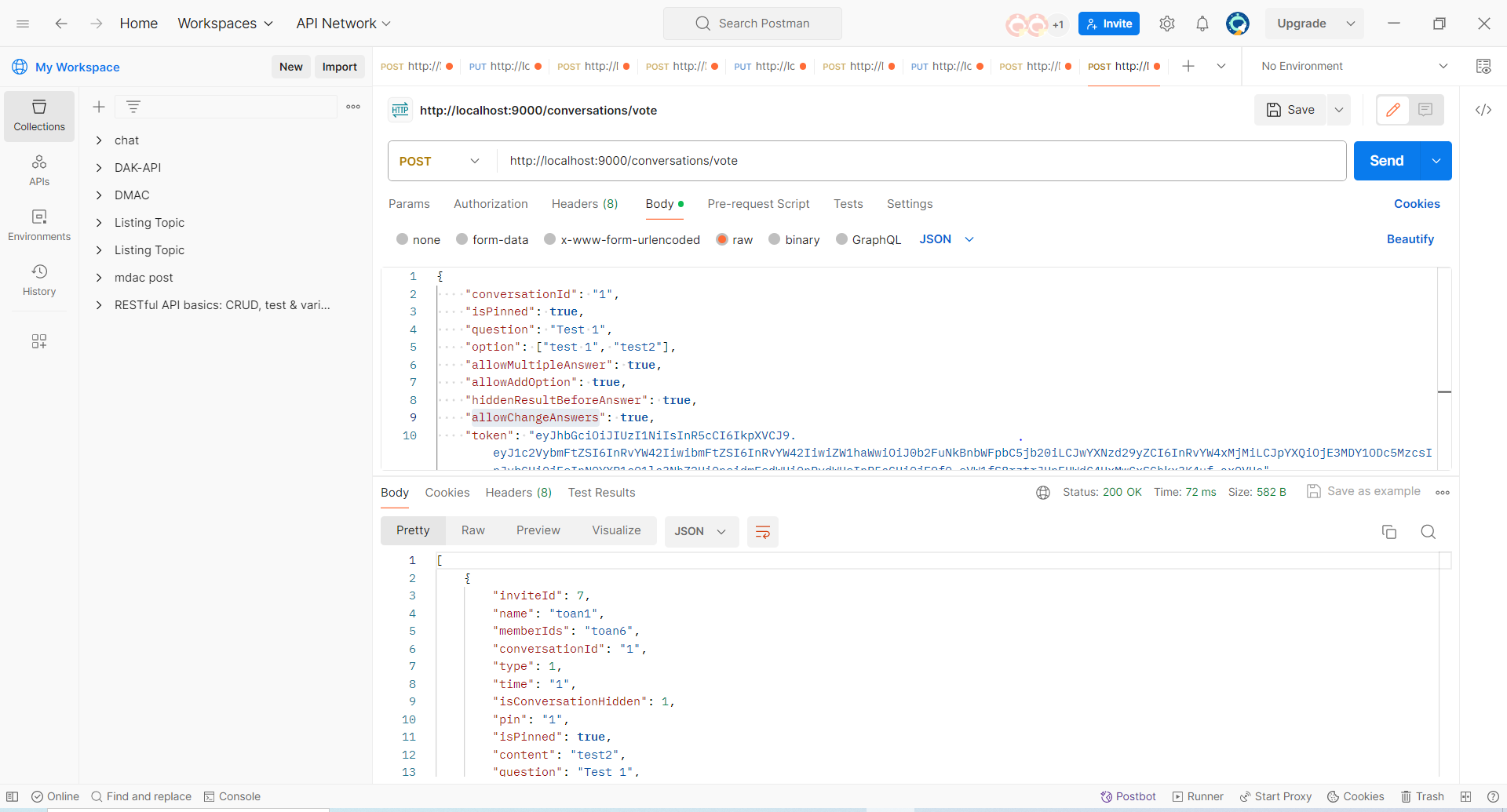


Còn về phần backend thì em sử dụng nodejs để xác nhận role của user có bằng 1,2,3 ko. Sau đó em lấy conversationSetting của conversationId đó coi có hay ko nếu có thì em update cái content của cái note và ghim của cái note. Code của em như sau:

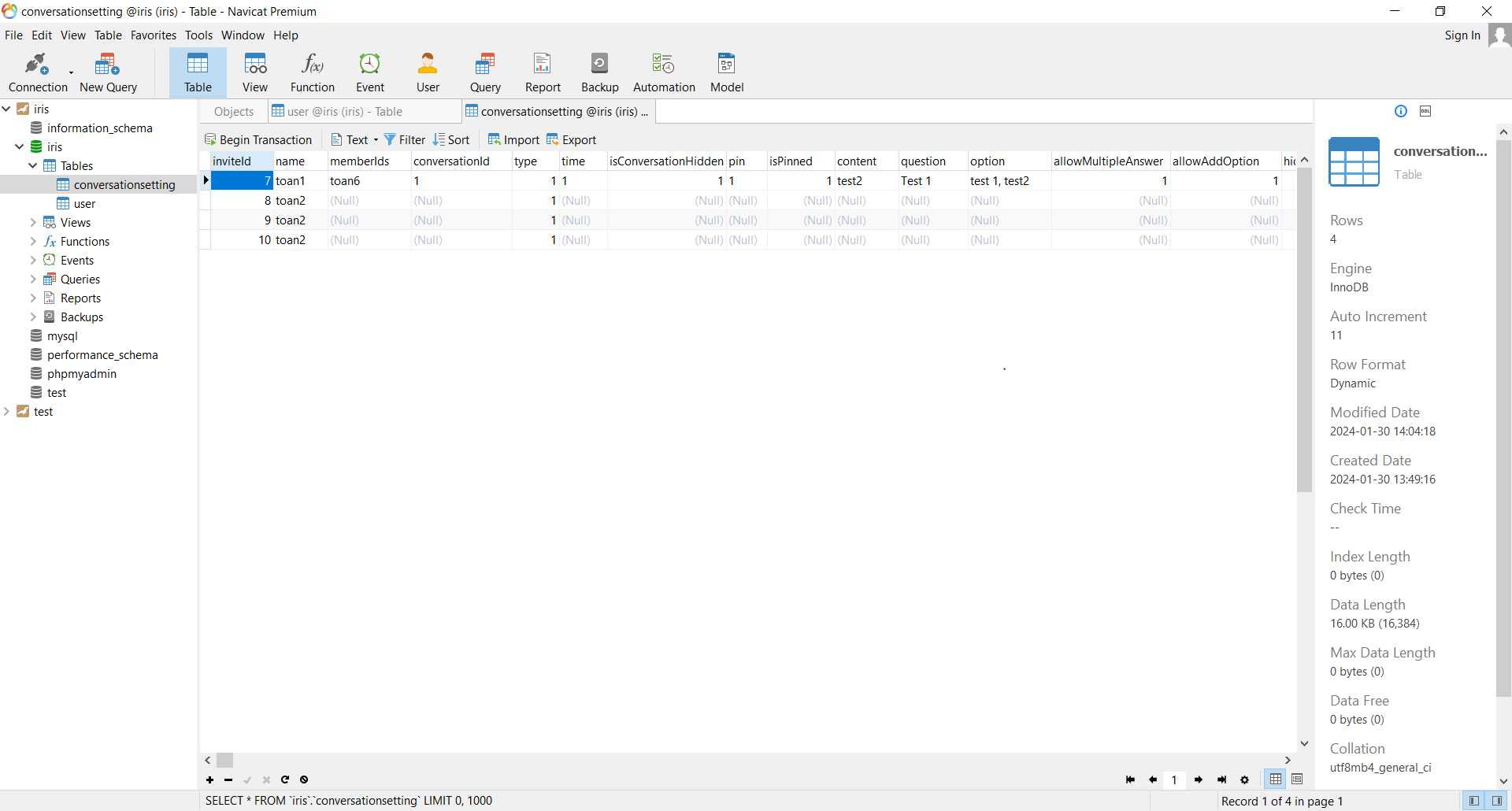


# Tạo bình chọn:

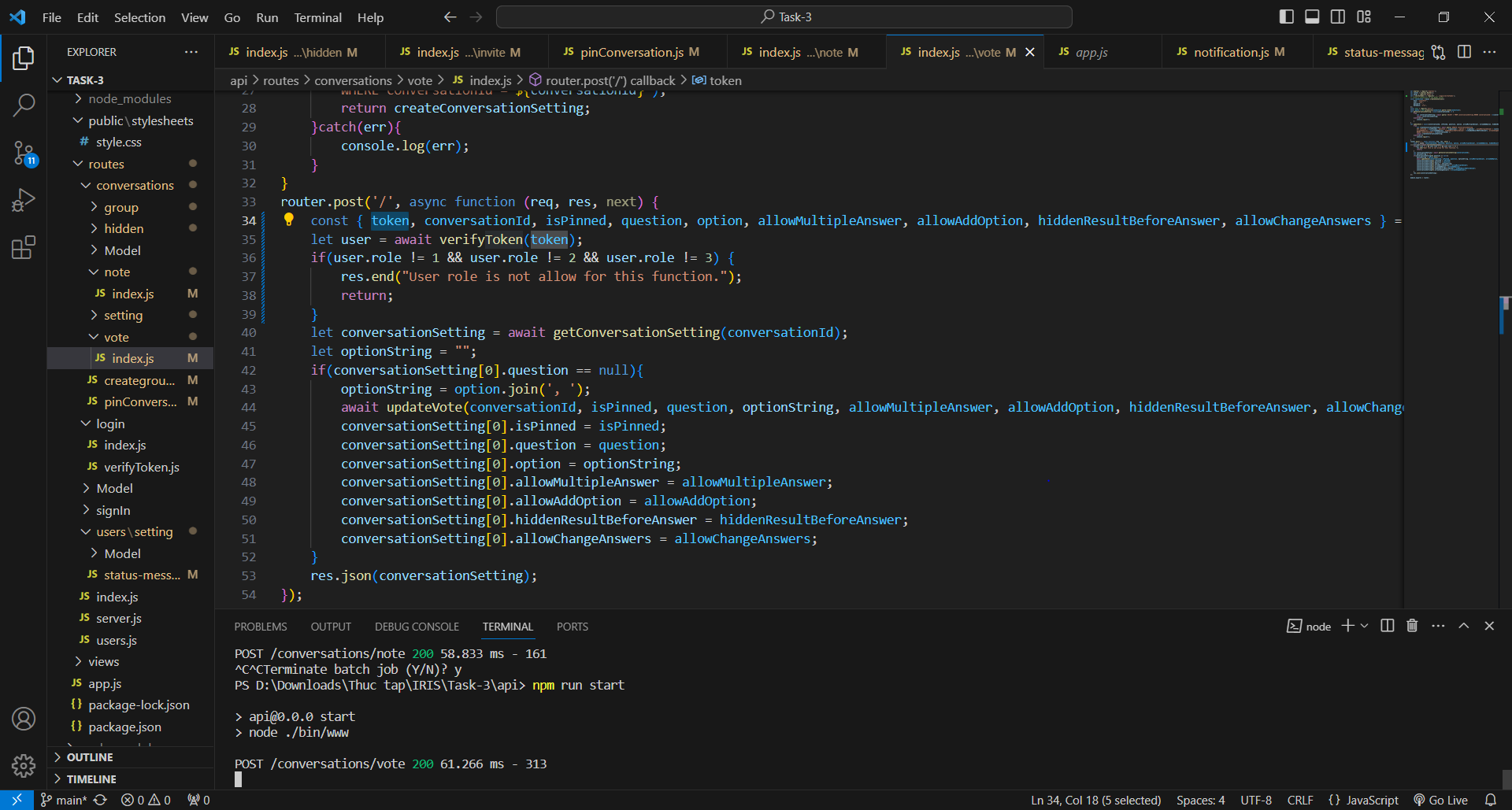
Đầu tiên để tạo được bình chọn thì người dùng cần phải nhập vào token của user có role là 1, 2, 3. Sau đó người dùng nhập vào conversationIds là id của cái group chat, isPinned là có ghim bình chọn hay không, question là câu hỏi bình chọn, option là lựa chọn của cuộc bình chọn, allowMultipleAnswer là có cho phép nhiều lựa chọn hay không, allowAddOption là cho phép tạo thêm lựa chọn hay không, hiddenResultBeforeAnswer là có che kết quả trước khi chọn xong hay không, allowChangeAnswers là có cho phép thay đổi lựa chọn hay không. Sau khi người dùng nhập xong hết rồi thì postman sẽ hiện ra như sau:



Và database navicat sẽ hiện ra như sau:



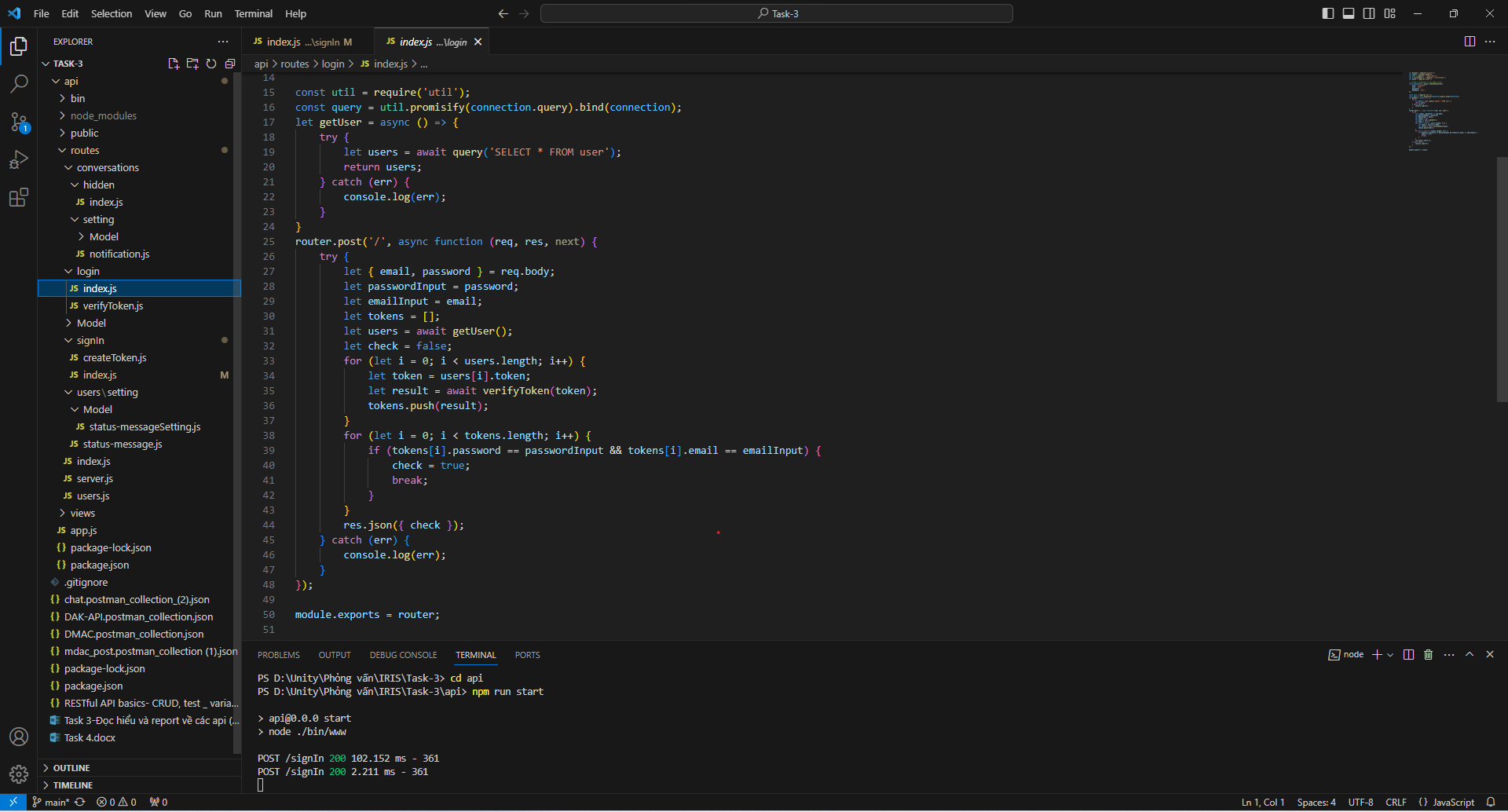
Về phần backend thì đầu tiên em sử dụng token để kiểm tra coi user role có bằng 1, 2, 3 không. Sau đó em lấy conversationSetting và cho tất cả memberIds cũ vào 1 chuỗi và sau đó gắn memberId mới vào optionString và cập nhật lại tất cả cái vote. Code em như sau:



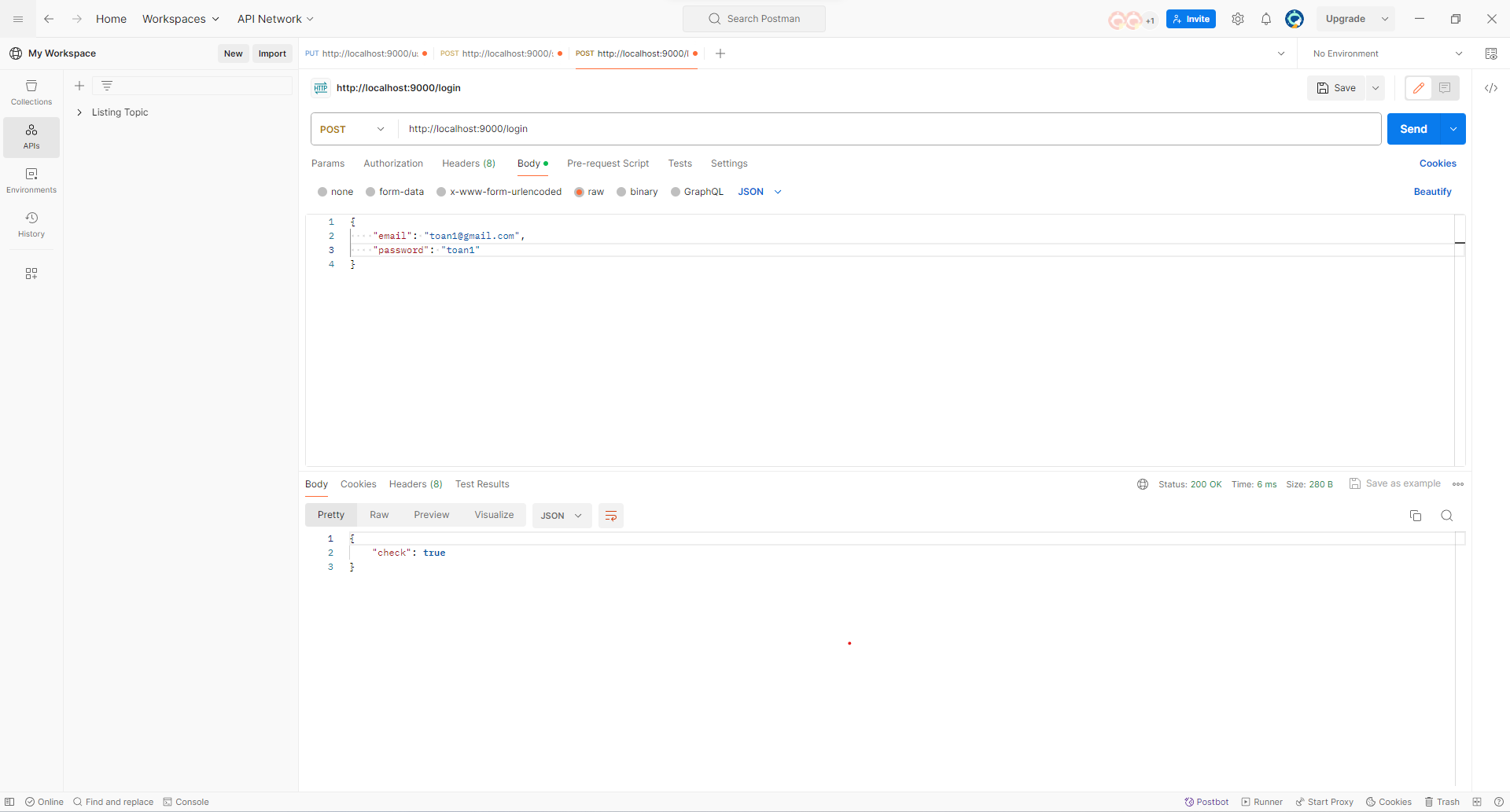
# Đăng nhập:

Đầu tiên để đăng nhập được thì người dùng cần phải đăng kí trên postman trước và sau đó lấy email và password khi đăng nhập để nhập vào postman.

Về phần đăng nhập thì trong code backend, em chỉ cần lấy tất cả user và giải mã tất cả user và cho từng user vào 1 cái mảng rổng để kiểm tra xem có cái nào trùng email với password với cái vừa đăng nhập ko nếu có thì postman sẽ xuất ra là yes. Code em sẽ như sau:

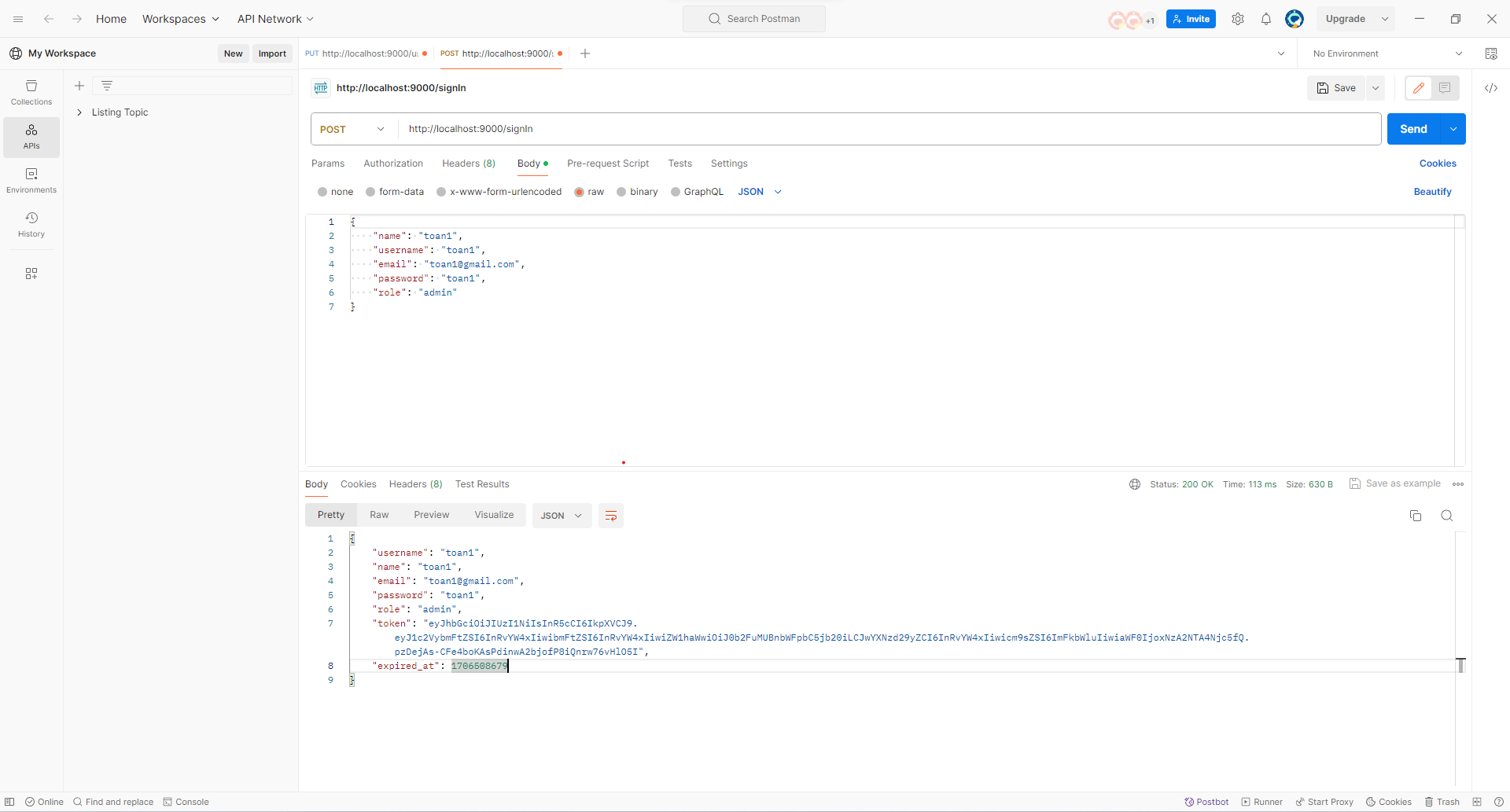


Trong postman thì em cần phải nhập email và password vào phần body và khi thành công postman sẽ hiện ra như sau:

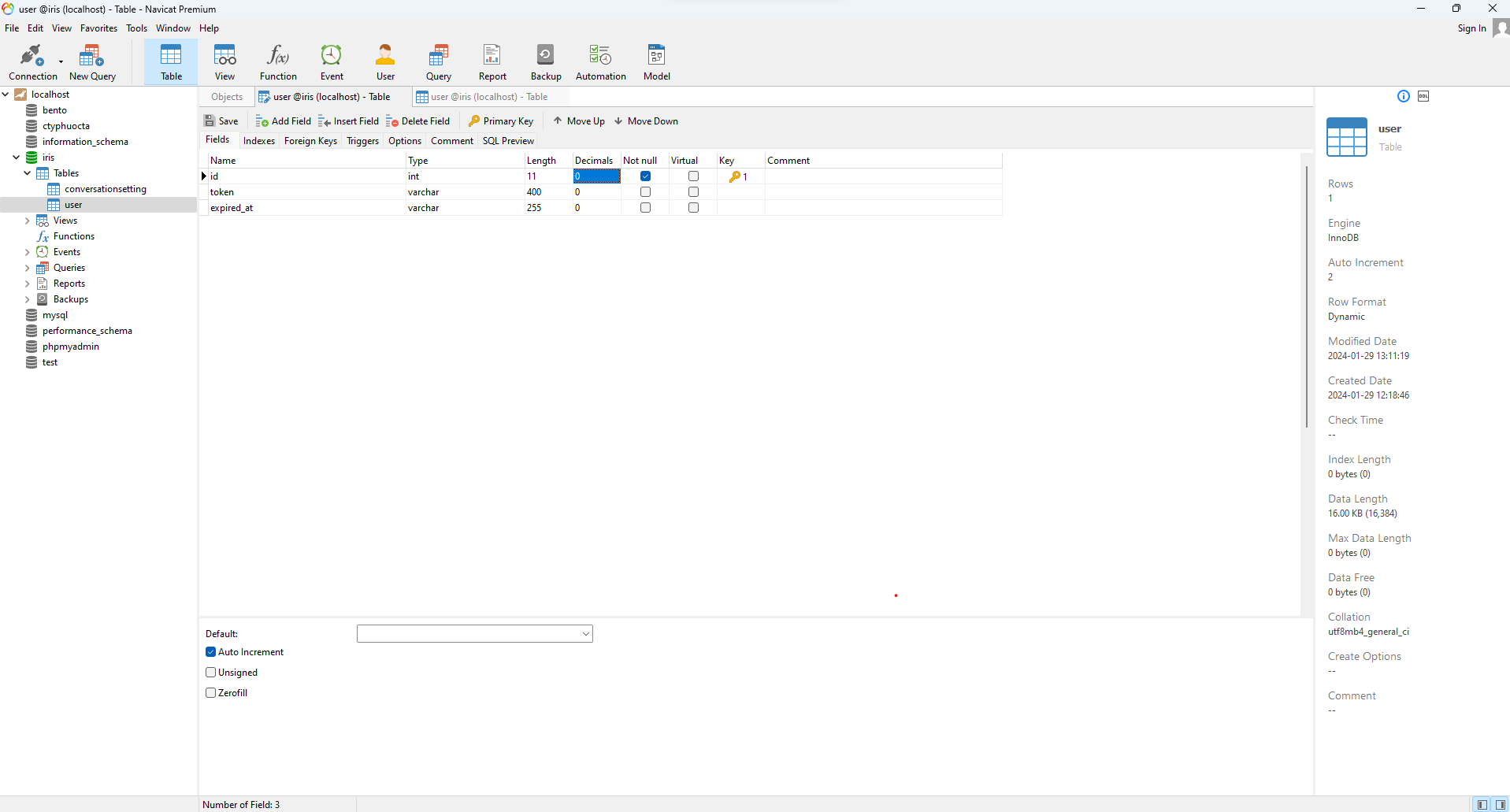


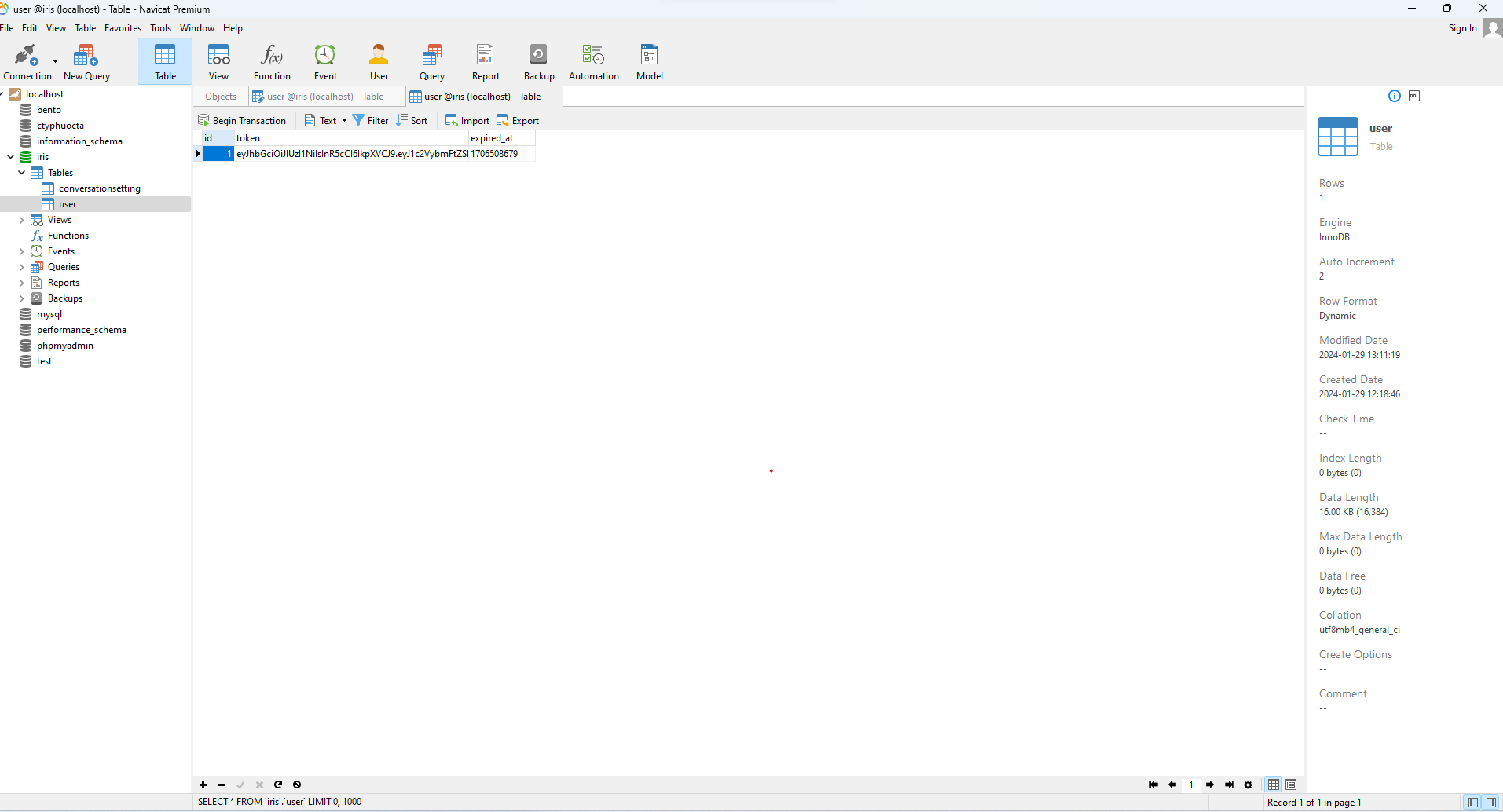
# Đăng kí:

Đầu tiên để đăng kí được thì trong postman em bắt user nhập name, username, email, password, role vào địa chỉ json object và sử dụng method post và sau khi thành công thì postman sẽ hiện ra như hình sau:

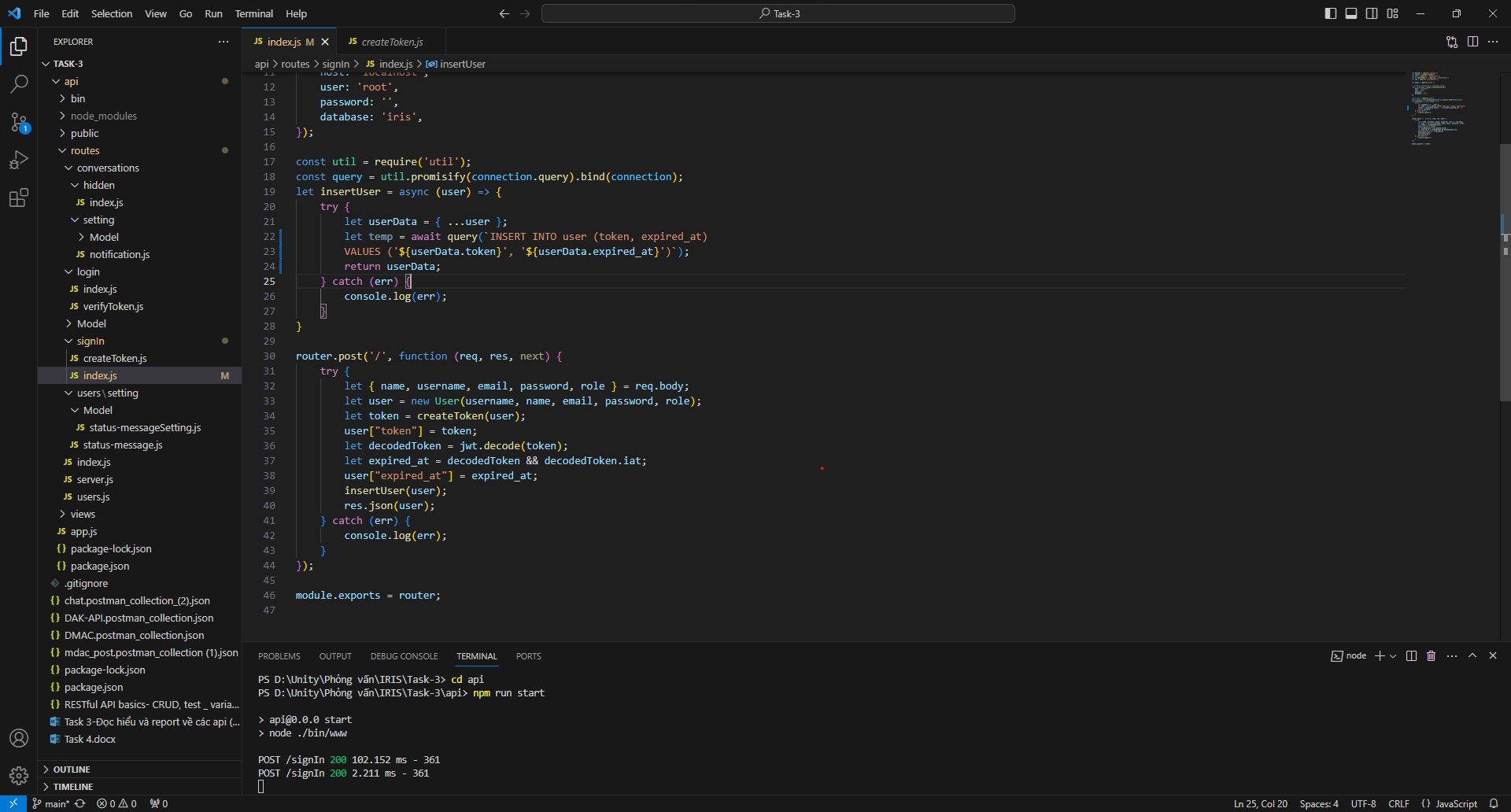


Trong phần database navicat thì em tạo ra database mới là user có 3 trường là id, token, expired\_at. Token sẽ lưu hết tất cả thông tin về user khi postman gửi về và sau khi postman gửi về thành công thì database sẽ hiện ra như sau:





Trong phần backend, em sử dụng jwtoken với payload là dữ liệu postman nhập vô, secret là Canhtoan111, em sử dụng mã HS256 và hết hạn sau 1h để tạo token. Sau đó em lấy cái token đó giải mã ra để lấy expired\_at để lưu vô database. Code em như hình ở dưới đây:



# Link Github:

https://github.com/canhtoan122/Task-3