

# Backend for blog

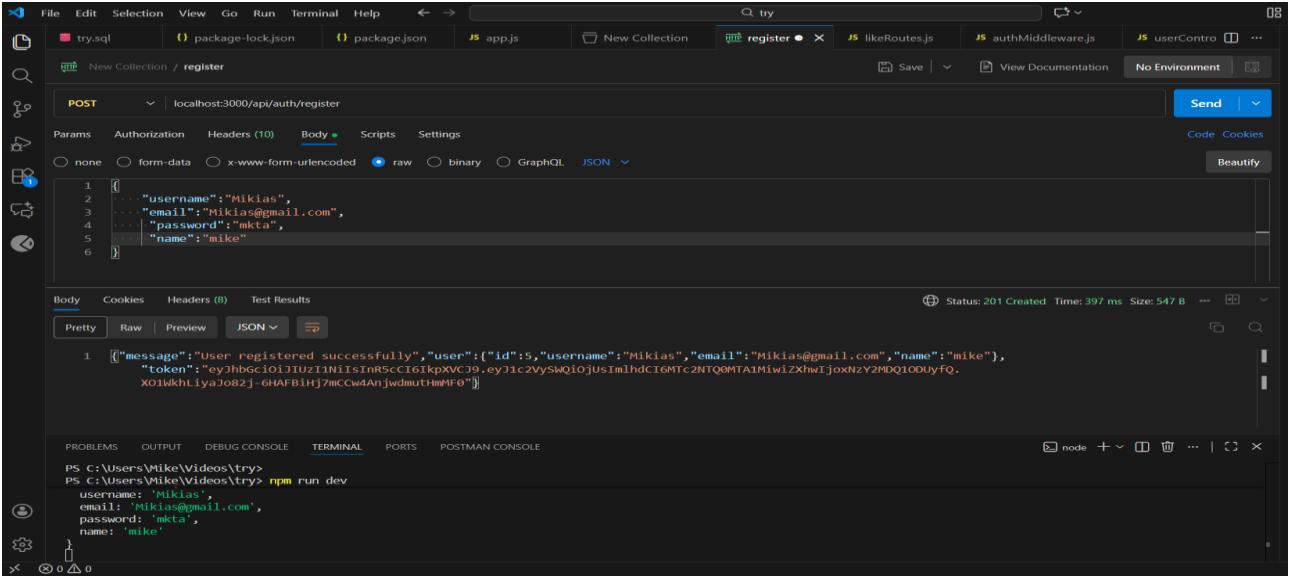
## Libraries and tools used

- Node
- Express
- PostgreSQL(and pgadmin)
- CORS
- JWT
- Bcrypt

## Functionalities and Features

- User Registration and Login with JWT authentication
- Follow and Unfollow
- Create read update and delete blog
- Rate blog
- Create edit and delete comment on blogs
- like unlike blogs
- Search blog by title or by user

All the mentioned functionalities and features have been implemented and tested using postman extesion on vs code and proved to be working, however from the required features I had trouble including profile pictures and did not include it, every other thing is done and here are few SELECTED screenshots.



The screenshot shows the Postman interface with a successful API call to 'localhost:3000/api/auth/register'. The request method is POST, and the body contains a JSON object with fields: username ('Mikias'), email ('Mikias@gmail.com'), password ('mikta'), and name ('mike'). The response status is 201 Created, and the response body is a JSON object indicating successful registration with an id (5), username ('Mikias'), email ('Mikias@gmail.com'), name ('mike'), and a token ('eyJhbGciOiJIUzI1NiBhX2lkXVCF9.eyJpc3MiOiJsb2dpbiJ9.eyJhdC16IjC2NTQ0MTA1MiwiZXhwIjoxNzY2MDQ1ODQyfQ.xO1wkhL1yaJ082j-6HAFB1Hj/mCCw4AnJwdmuthmP0'). Below the main interface, the terminal shows the command 'npm run dev' being run in the directory 'C:\Users\MikeVideos\try', and the output shows the same JSON object as the response.

register

```
1 SELECT * FROM public.users
2 ORDER BY userid ASC
```

	userid [PK] integer	username character varying (50)	email character varying (100)	password character varying (255)	name character varying (100)	role character varying (20)
1	1	testuser	test@email.com	plain_for_now	Test User	User
2	3	danil	danike@gmail.com	\$2a\$10\$eyoNJBhyILvExVbxeuOtWXKnvQghfm4d5ra0Bgy4s/FZTuhwLS	kebededani	User
3	4	abel	abela@gmail.com	\$2a\$10\$06xytN1x17pPn.e9WfI7q.uBbHHOnD1Wp0g8XDKePwJtyNH05ao...	belit	User
4	5	Mikias	Mikias@gmail.com	\$2a\$10\$LH6ZlsNwEk7.UjwO0310s08GBRa9VAcGFWoESRD/Hath7Sfaat...	mike	User

Total rows: 4 Query complete 00:00:00.175 CRLF

The screenshot shows the Microsoft Visual Studio Code interface with the Postman extension installed. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The left sidebar has icons for .env, try.sql, package-lock.json, app.js, and collections named 'create blog' and 'register'. The main workspace shows a POST request to 'localhost:3000/api/blogs'. The 'Body' tab is selected, displaying the following JSON payload:

```
1 {  
2   ... "userId":5,  
3   ... "title": "Building road",  
4   ... "content": "we need road "  
5   ...  
6 }
```

The 'Test Results' tab shows a successful response with status 201 Created, time 51 ms, and size 483 B. The response body is:

```
1 [ {"message": "Blog created successfully", "blog": {"blogid": 2, "userid": 5, "title": "Building road", "content": "we need road ", "tags": null, "createdat": "2025-12-11T10:29:24.012Z", "updatedat": "2025-12-11T10:29:24.012Z"}]
```

The bottom navigation bar includes PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and POSTMAN CONSOLE. The bottom right corner shows the date and time as 12/11/2025 2:29 AM.

pgAdmin 4

File Object Tools Edit View Window Help

explorer    No limit    Scratch Pad

public.blogs/try/postgres@PostgreSQL 18

Query History

```
1 SELECT * FROM public.blogs
2 ORDER BY blogid ASC
```

Data Output Messages Notifications

Showing rows: 1 to 2 Page No: 1 of 1

blogid [PK] integer	userid integer	title character varying (255)	content text	tags text[]	createdat timestamp without time zone	updatedat timestamp without time zone
1	1	4 Trial	once upon	[null]	2025-12-08 02:47:48.828254	2025-12-08 02:47:48.828254
2	2	5 Building road	we need roa...	[null]	2025-12-11 02:29:24.012363	2025-12-11 02:29:24.012363

Servers > PostgreSQL 18 > Databases > try > Schemas > public > Tables > blogs 09

CRLF Ln 1, Col 1

Type here to search

then follow /unfollow

File Edit Selection View Go Run Terminal Help

EXPLORER ... age.json JS app.js New Collection http://localhost:3000 register Login userRoutes.js http://localhost:3000 Like.js

TRY controllers

POST http://localhost:3000/api/users/follow/6

Params Authorization Headers (9) Body Scripts Settings

Headers @ 8 hidden

Key	Value
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjUsInlvbGUiOiVcVyl...055Z
Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

Status: 200 OK Time: 64 ms Size: 397 B

```
1 [{"message": "Followed successfully", "follow": {"followid": 1, "followerid": 5, "followingid": 6, "createdat": "2025-12-11T09:21:37.055Z"}}]
```

OUTLINE TIMELINE

Type here to search

The screenshot shows the pgAdmin 4 interface. On the left, the sidebar displays the schema structure under the 'public' schema, including tables like 'follows', 'users', and 'blogs'. The main area shows a query window with the following SQL code:

```
1 SELECT * FROM public.follows
2 ORDER BY followid ASC
```

The results pane displays the data from the 'follows' table:

	followid [PK] integer	followerid integer	followingid integer	createdat timestamp without time zone
1	1	5	6	2025-12-11 01:21:37.055786

Total rows: 1 Query complete 00:00:00.227 CRLF Ln 1, Col 1

The screenshot shows the Postman application interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The main workspace shows a DELETE request to the URL <http://localhost:3000/api/users/unfollow/6>. The Headers tab is selected, showing the following configuration:

Key	Value
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjUsInlvbGUiOiJVc2VylwiWF0joxNzY1...
Key	Value

The Body tab shows a JSON response:

```
1 {"message": "Unfollowed successfully"}
```

The bottom navigation bar includes PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and POSTMAN CONSOLE. The status bar indicates Status: 200 OK, Time: 53 ms, Size: 304 B.

Ready to explain and answer questions.