

Uniconn: My Implementation of a Simple Web Forum Application

Kavish Sathia
kavish@comp.nus.edu.sg

23rd January 2024

1 Features

These are the features that I have included in Uniconn. To try out these features, go to: Uniconn Web (You can click on "Lurk" to skip the authentication process):

- Account-based authentication with JWT stored in a HTTP Cookie.
- Write and edit threads using markdown and include a thumbnail picture for more engagement, deletion of threads
- Real-time notifications when the user's thread or comment is liked or commented on.
- Nested comments (up to 5 levels).
- Mobile responsiveness
- Progressive web app (allowing users to install the web app onto their devices).
- Searching for forums based on title, description, tags and author username.
- A point system (Aura) that will increase when a user's thread or comment gets liked. This will incentivize participation.
- Content moderation using AI and community feedback.
- AI-generated descriptions for threads
- Fact checking for threads

These are features that I plan to include or is currently in progress:

- Communities feature with each community having its own theme and color scheme
- I'm also planning to write unit tests for the backend and end-to-end tests for the frontend using cypress.io.

2 Tech Stack, User Interface and API

I have covered the tech stack and user interface in more detail in my mid-submission, and there have been not many changes since then. I will be including a user manual at the end of this document so, I will not be covering this here. As for the Developer Manual, you can find it at <https://github.com/kavishsathia/gossip?tab=readme-ov-file#developer-manual> and you can find the API Documentation at <https://gossip.s6wyfaw6z9q0r.ap-southeast-1.cs.amazonlightsail.com/swagger/index.html>

3 Data Model

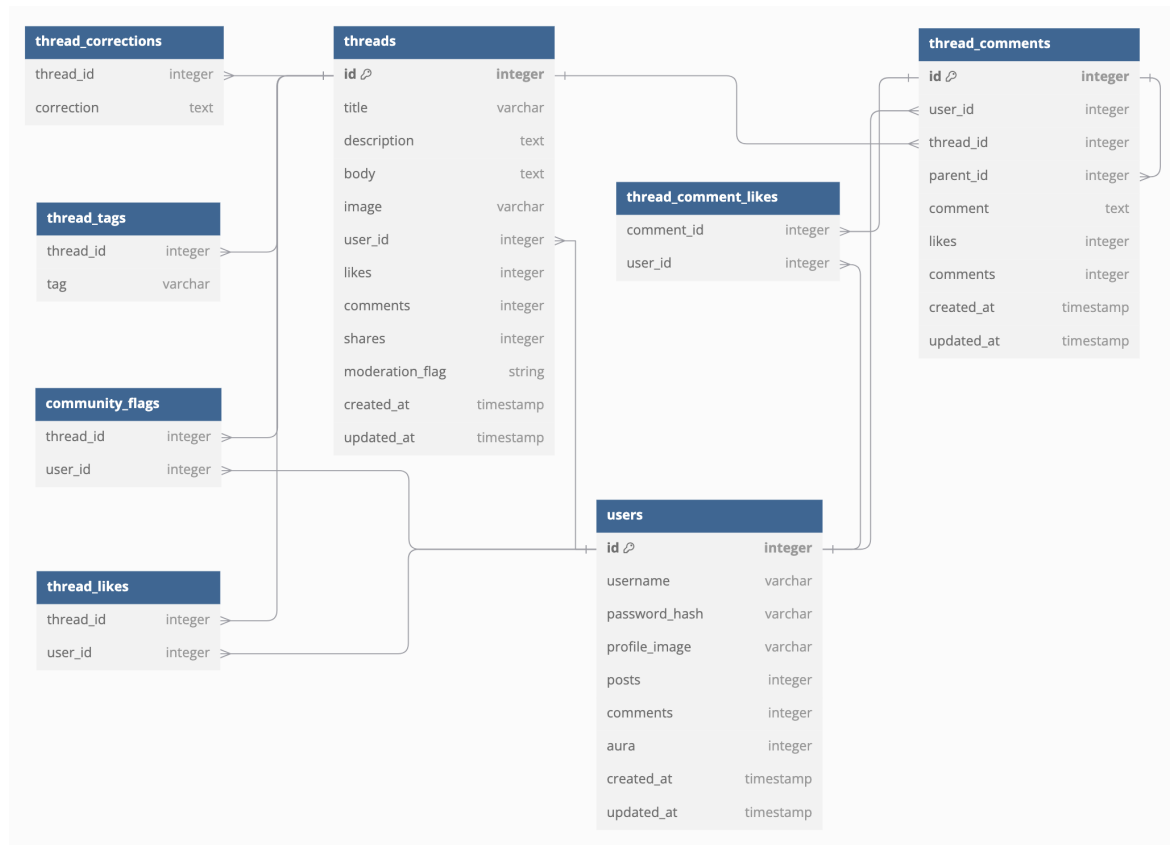


Figure 1: The ERD diagram for Unicorn's relational database

4 Architecture

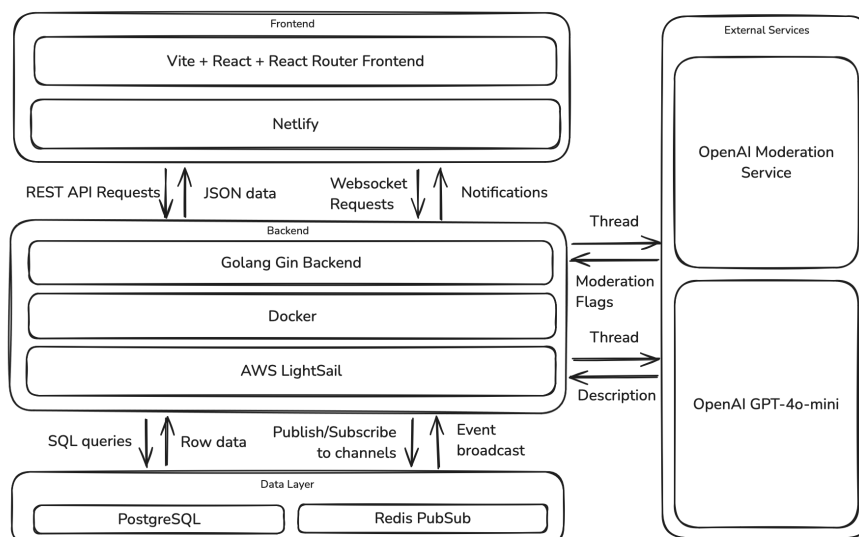


Figure 2: The tech architecture for Unicorn

The architecture consists of 4 segments: the frontend which is hosted on Netlify, the backend which is hosted on AWS Lightsail as a Docker container, the data layer which has two databases, PostgreSQL

and Redis as well as the external services layer which includes the AI services. The implementation of the real-time notification system was done using the PubSub feature of Redis and is covered in detail in the mid-submission write up. As for the moderation feature, I used the free service from OpenAI to moderate posts based on a few categories. The description and the fact checking is done using the GPT-4o-mini model from OpenAI.

5 Accomplishments

This is my first (almost) complete open-source project on GitHub, and it has been a game-changer for me as a programmer. Through this project, I learned so much that I genuinely feel like I've leveled up. One of the biggest challenges was setting up the markdown editor and deploying the application as a container on AWS Lightsail (tasks that initially took me around 5 hours and a full day respectively). More recently though, when I had to do the same for Hack&Roll 2025 (a hackathon), I broke my personal best, configuring the Markdown editor in just 10 minutes and deploying to AWS Lightsail in 15 minutes. These improvements contributed vastly to my team winning at the hackathon, and I owe a huge thanks to the CVWO team for helping me hone my skills along the way.

6 User Manual

1. How to create a new account? Click on sign up on the login page.
2. Can I view posts without signing up? Yes, you can! Just click on "Lurk" on the login page, however if you try to like or comment, you will be redirected to the login page.
3. How to search for posts? Use the search bar on the left sidebar.
4. How to search by tags and author username? Add "#" in front of tags and "@" in front of usernames when searching in the search bar. For example, to find all posts written by me with the "nus" tag and contains the word "CS", you can search "@kavish #nus cs".
5. Is there a quicker way to search by tags and username? Yes, you can click on the tags in the posts and click on the author username in the posts (then you will be shown a modal in which "View their Threads" is an option")
6. How to view a thread? Click on any of the threads in the left sidebar.
7. How to view comments? Scroll down and you will see comments if there are any. You can view nested comments by clicking on "Show replies", and you will also be presented with text input to reply to that comment.
8. How to write comments? Simply fill in the text input at the bottom of the thread and Enter to post the comment.
9. How to like threads and comments? Click on the heart button, if you are being redirected to the login page, it means you are lurking.
10. How to create my own thread? Click on the "Plus" button at the bottom right of the page. An editor page will open up. You can submit a URL for the thumbnail picture. You can change the title by editing the "Untitled Thread" text. You can also add more tags by clicking on "Add tag" or the "X" button next to the tags. The body of the post can be edited within the markdown editor. After you're done, click on the tick button on the bottom right corner.
11. How do I edit my thread or comment? For your own threads and comments, you will see a pencil icon on the top right corner. Click on that to edit. For threads, the familiar markdown editor will open up, and after you're done, click on the tick button. For comments, an "Edit" button will replace the pencil icon on the top right corner, and the comment text will be replaced with a textarea. Simply edit the text in the textarea and click on the "Edit" button.
12. How to delete comments and threads? Click on the trash icon next to the pencil icon.
13. How to report threads? Click on the "Report" button if it appears on the top of the thread. It will only appear if the thread has been flagged by the moderation service.