liteForum – a Modern Forum App

Introduction:

LiteForum is a lightweight forum app designed with simplicity in mind. It is currently online at xsy.science. Developed in under three weeks, it achieved what I originally planned. Powered by React in the frontend and Ruby on Rails in the backend, it features modern web development concepts MVC and RESTful API. The frontend is deployed on GitHub Pages, while the backend is containerized using Docker and deployed on AWS EC2.

Use case

A Lightweight forum is always in demand. While I am still adding features to the forum to make it less “toyish", I hope it can become a dedicated forum for CS students to discuss Ph.D. applications. Information about Ph.D. applications is scattered across the web; I believe a dedicated forum will be welcomed. Users of the forum will be able to view past applicants’ cases, find useful resources and chat with their peers in the forum.

User manual

The forum app is easy to navigate. Upon visiting the site, users are directed to the post-list where they can view details of specific posts by tapping on them. The detailed page for a post includes comments and tags. In order to leave a comment, users must be logged in. The tag list displays all available tags, and tapping on one directs users to the tag show page, where they can view a list of all posts with that specific tag. Additionally, clicking a tag on the post-show page also directs users to the tag-show page. Once logged in, users have the ability to create their own posts and view their personal profile page. On their profile page, they can edit their original posts and delete past comments. The app offers many additional features for users to explore.

Implementation details

The frontend of the forum was built using the React JavaScript framework. This allowed for the creation of reusable UI components and efficient rendering of the user interface. The project was set up using the create-react-app tool, which provided a basic structure and configuration for the React application. Bootstrap library was used to create simple UI components. Axios was implemented to simplify sending HTTP requests and parsing JSON. I also included Redux in the front end, which was primarily used to keep track of user authentication information.

For the backend, I used the Ruby on Rails framework. While Ruby on Rails is a full-stack framework, it is also a great practice to use it as a backend API. Its advanced Active Record system allows me to build associations between tables and set up databases with ease. In the database for the LiteForum, there is a joined table between tags and posts, allowing for many-to-many relations and an advanced tagging system. Gem is used to simplify JWT workflow.

For the deployment part, the frontend is deployed on GitHub pages with some compromises. Brower Routing is not supported on GitHub pages, so I have to switch it to Hash Routing. For the backend, I set up an NGINX server that listens at port 443 and forwards the request to localhost:3000, where the docker container listens at. Host port 3000 is then mapped to container port 3000. NGINX enables reverse routing and simplifies the process of enabling HTTPS in the backend. (I actually tried to enable HTTPS with the rails built-in puma server, it was more complicated.)

Limitations and further plans

The forum app, although functional, still has several limitations that need to be addressed. Some of these limitations include: 1) The absence of image upload support, which would enhance the user experience. 2) The lack of spam protection measures, which could lead to unwanted content on the platform. This is primarily due to the absence of industry-standard tools such as AWS S3 and reCAPTCHA. 3) The app also lacks the ability to update personal information, which would be beneficial to users. 4) It also doesn't support page flipping, which is an important aspect of the user interface. In future iterations, I aim to continue working on this project and make it a more robust and feature-rich app.