ForoCity

A Web Forum

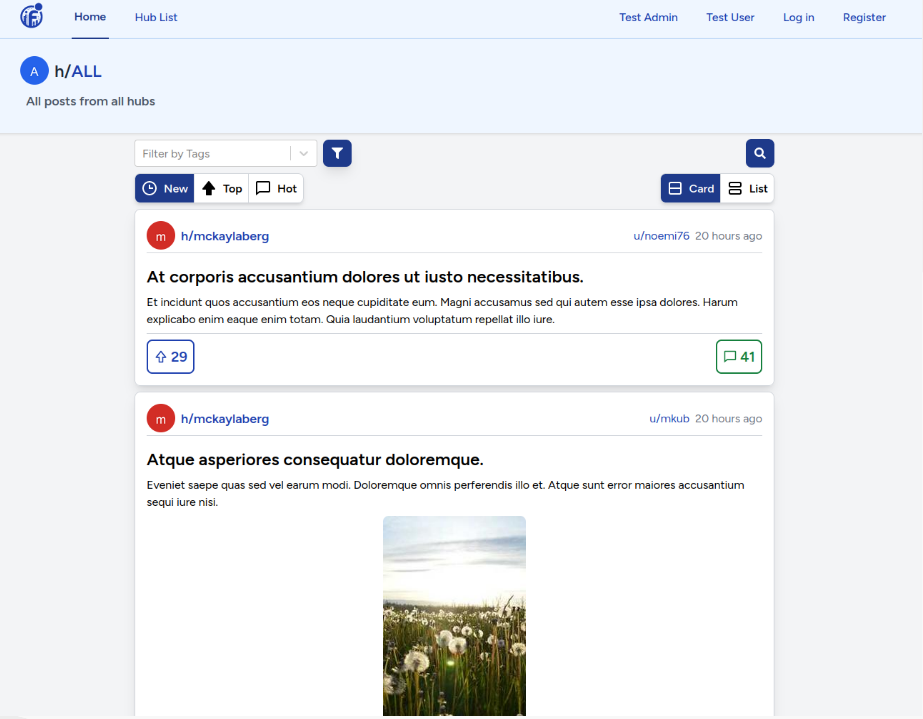
-Gurjant Singh

Index

- Introduction
- Key Features
- Important Pages, Components
- How it is implemented
- Challenges
- Improvements
- External Libraries

ForoCity

ForoCity is a web forum that allows users to create and participate in discussion topics of their city.
Filter, sort and search posts by tags, votes, title or content.
It has also direct messages between users.



At corporis accusantium dolores ut iusto necessitatibus.

Et incidunt quos accusantium eos neque cupiditate eum. Magni accusamus sed qui autem esse ipsa dolores. Harum explicabo enim eaque enim totam. Quia laudantium voluptatum repellat illo iure.

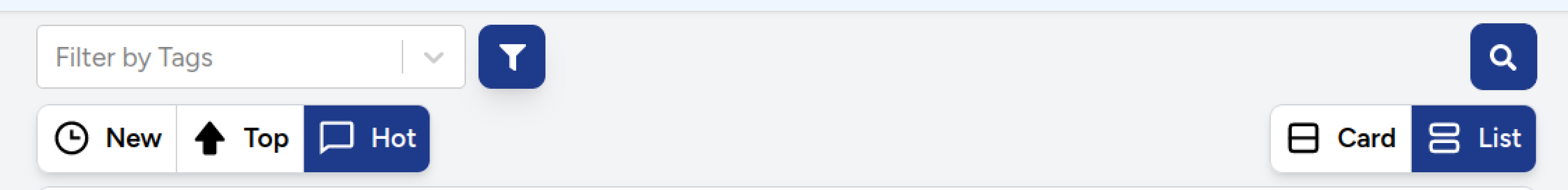
Atque asperiores consequatur doloremque.

Eveniet saepe quas sed vel earum modi. Doloremque omnis perferendis illo et. Atque sunt error maiores accusantium sequi iure nisi.

Key Features

It has many features like:

- Filter by Tags
- Search by title or content
- Sort by New, Most Voted, Most Commented.
- Two View types: Card and List

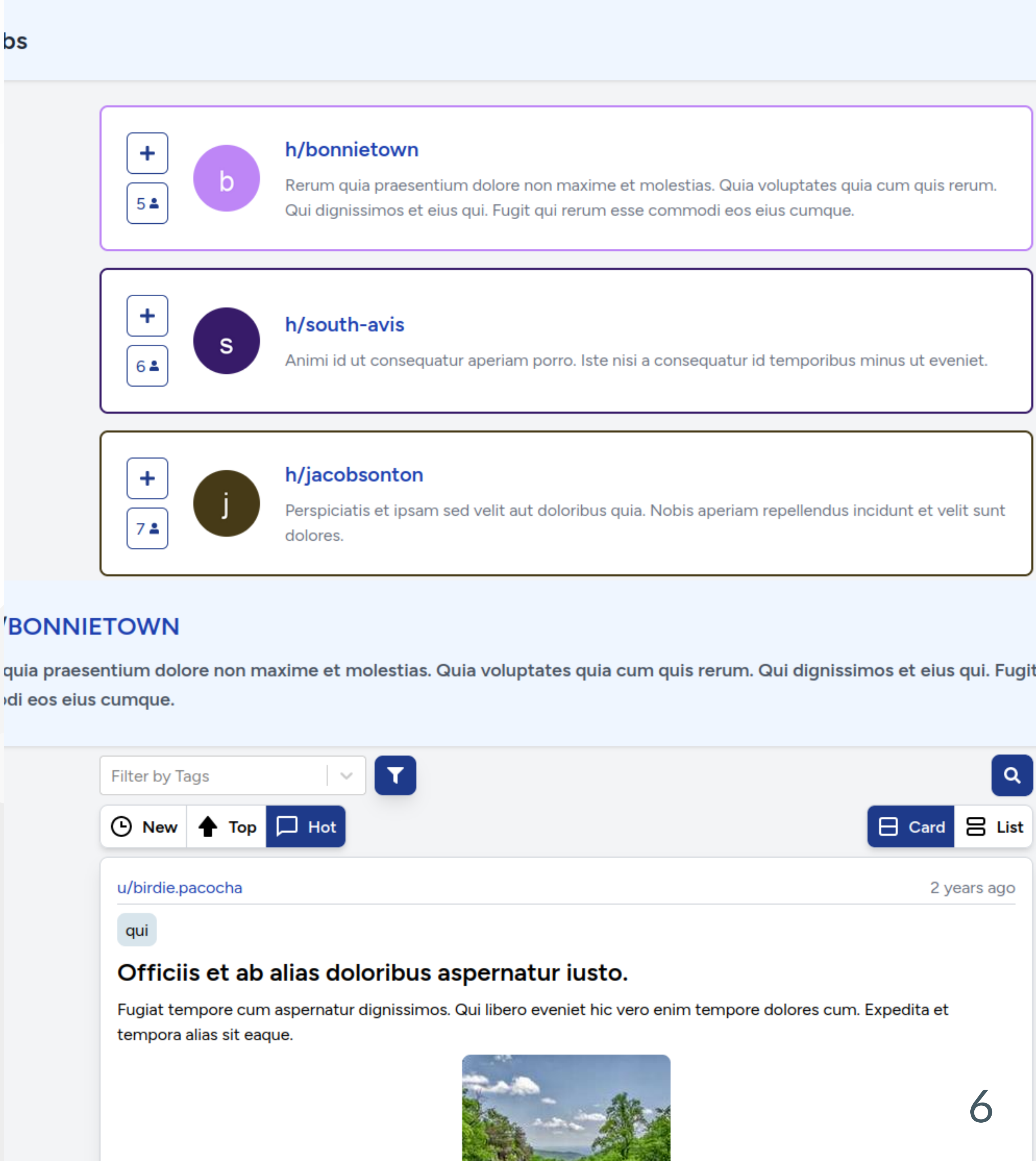


Key Features

- Chat with other users
- Create New Hubs, Join Hubs
- Create Posts, Vote on them
- Create Responses, Vote on them
- Direct Messages with other users
- Edit and Delete Posts, Responses, Messages
- Admin can only delete things not edit them

Hubs

Hubs are groups for a city. User can create a hub for their city and join other hubs. and post in them. Any user can join any hub. But only the creator of the hub and admin can delete the hub.



bs

h/bonnietown

Rerum quia praesentium dolore non maxime et molestias. Quia voluptates quia cum quis rerum. Qui dignissimos et eius qui. Fugit qui rerum esse commodi eos eius cumque.

h/south-avis

Animi id ut consequatur aperiam porro. Iste nisi a consequatur id temporibus minus ut eveniet.

h/jacobsonton

Perspiciatis et ipsam sed velit aut doloribus quia. Nobis aperiam repellendus incidunt et velit sunt dolores.

'BONNIETOWN

quia praesentium dolore non maxime et molestias. Quia voluptates quia cum quis rerum. Qui dignissimos et eius qui. Fugit
ɔdi eos eius cumque.

u/birdie.pacocha 2 years ago

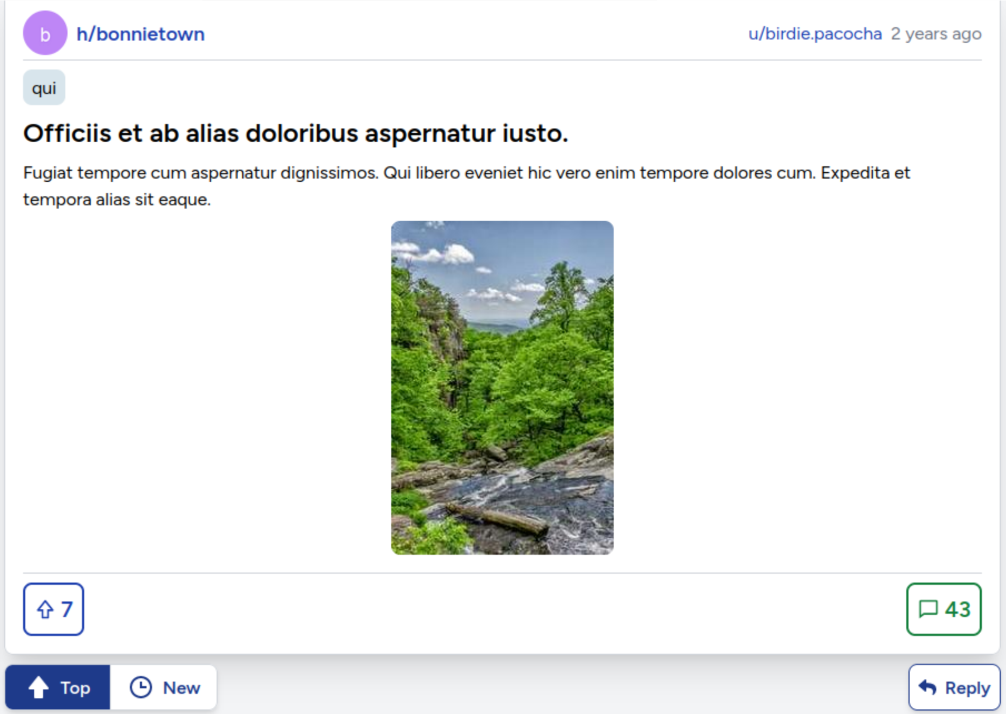
qui

Officiis et ab alias doloribus aspernatur iusto.

Fugiat tempore cum aspernatur dignissimos. Qui libero eveniet hic vero enim tempore dolores cum. Expedita et tempora alias sit eaque.

Posts

User can create posts in any hub. They can also edit and delete their posts. They can also vote on posts. Posts can be sorted by New, Most Voted, Most Commented. They can also be filtered by tags.



h/bonnietown

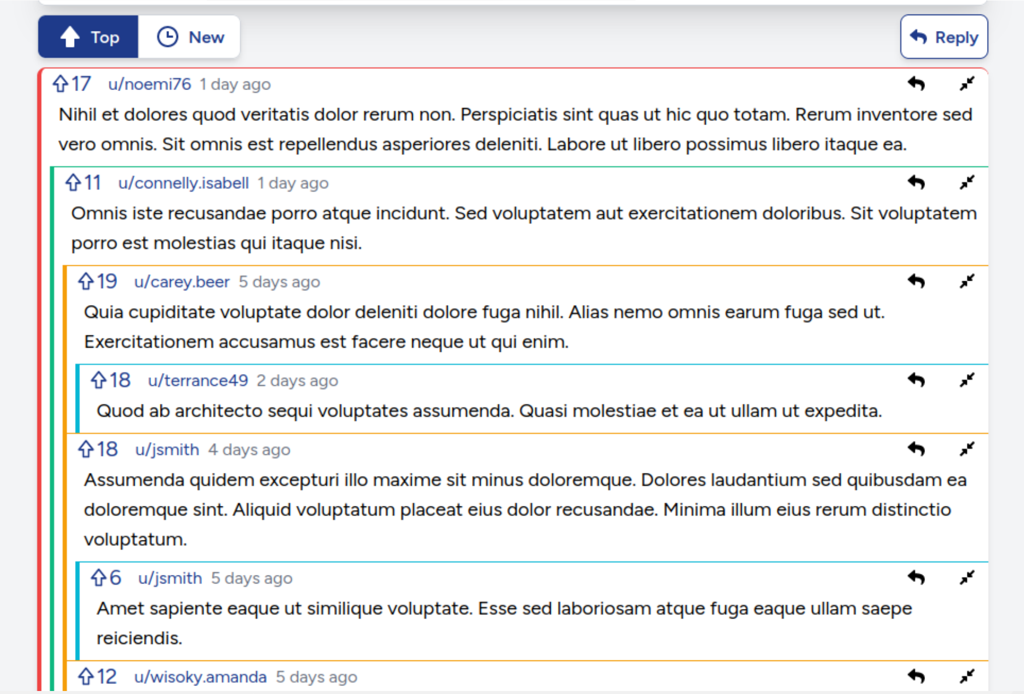
u/birdie.pacocha 2 years ago

qui

Officiis et ab alias doloribus aspernatur iusto.

Fugiat tempore cum aspernatur dignissimos. Qui libero eveniet hic vero enim tempore dolores cum. Expedita et tempora alias sit eaque.

Responses



17 u/noemi76 1 day ago

Nihil et dolores quod veritatis dolor rerum non. Perspiciatis sint quas ut hic quo totam. Rerum inventore sed vero omnis. Sit omnis est repellendus asperiores deleniti. Labore ut libero possimus libero itaque ea.

11 u/connelly.isabell 1 day ago

Omnis iste recusandae porro atque incidunt. Sed voluptatem aut exercitationem doloribus. Sit voluptatem porro est molestias qui itaque nisi.

19 u/carey.beer 5 days ago

Quia cupiditate voluptate dolor deleniti dolore fuga nihil. Alias nemo omnis earum fuga sed ut. Exercitationem accusamus est facere neque ut qui enim.

18 u/terrance49 2 days ago

Quod ab architecto sequi voluptates assumenda. Quasi molestiae et ea ut ullam ut expedita.

18 u/jsmith 4 days ago

Assumenda quidem excepturi illo maxime sit minus doloremque. Dolores laudantium sed quibusdam ea doloremque sint. Aliquid voluptatum placeat eius dolor recusandae. Minima illum eius rerum distinctio voluptatum.

6 u/jsmith 5 days ago

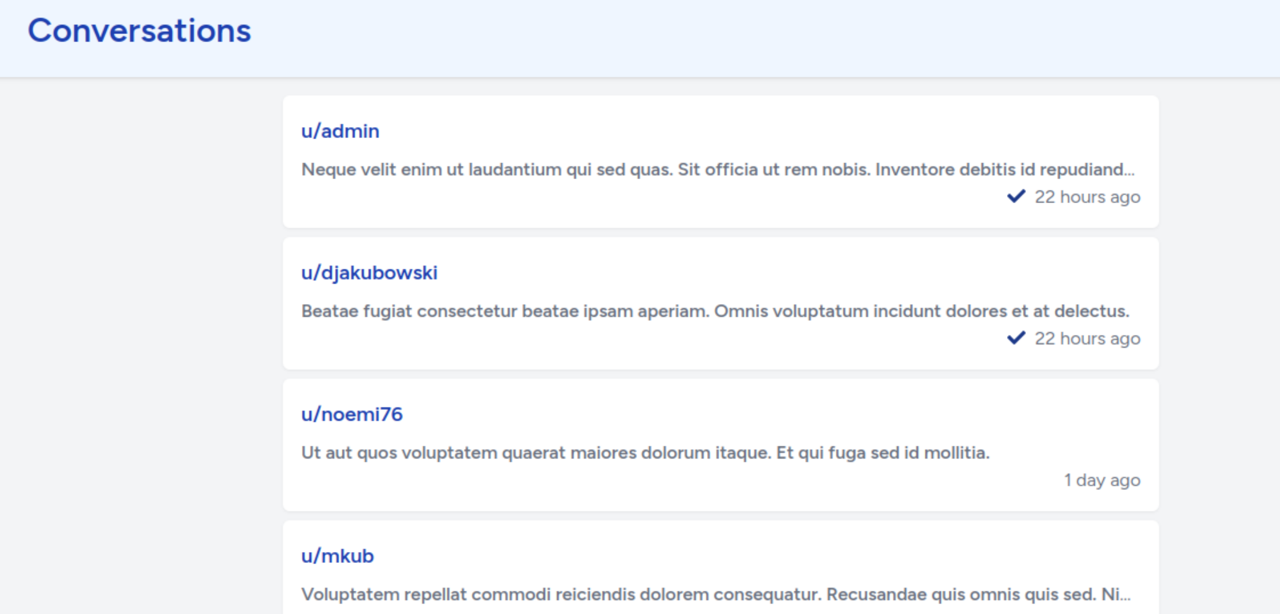
Amet sapiente eaque ut similique voluptate. Esse sed laboriosam atque fuga eaque ullam saepe reiciendis.

12 u/wisoky.amanda 5 days ago

User can create responses to posts. They can also edit and delete their responses. They can also vote on responses. Responses can be sorted by New, Most Voted. They are displyed in a tree like structure. They can be sort by New, or Most voted.

Conversations

User can send direct messages to other users. They are shown in style of a chat. They can also click on the user to see their profile.



Conversations

u/admin

Neque velit enim ut laudantium qui sed quas. Sit officia ut rem nobis. Inventore debitis id repudiand...

✓ 22 hours ago

u/djakubowski

Beatae fugiat consectetur beatae ipsam aperiam. Omnis voluptatum incidunt dolores et at delectus.

✓ 22 hours ago

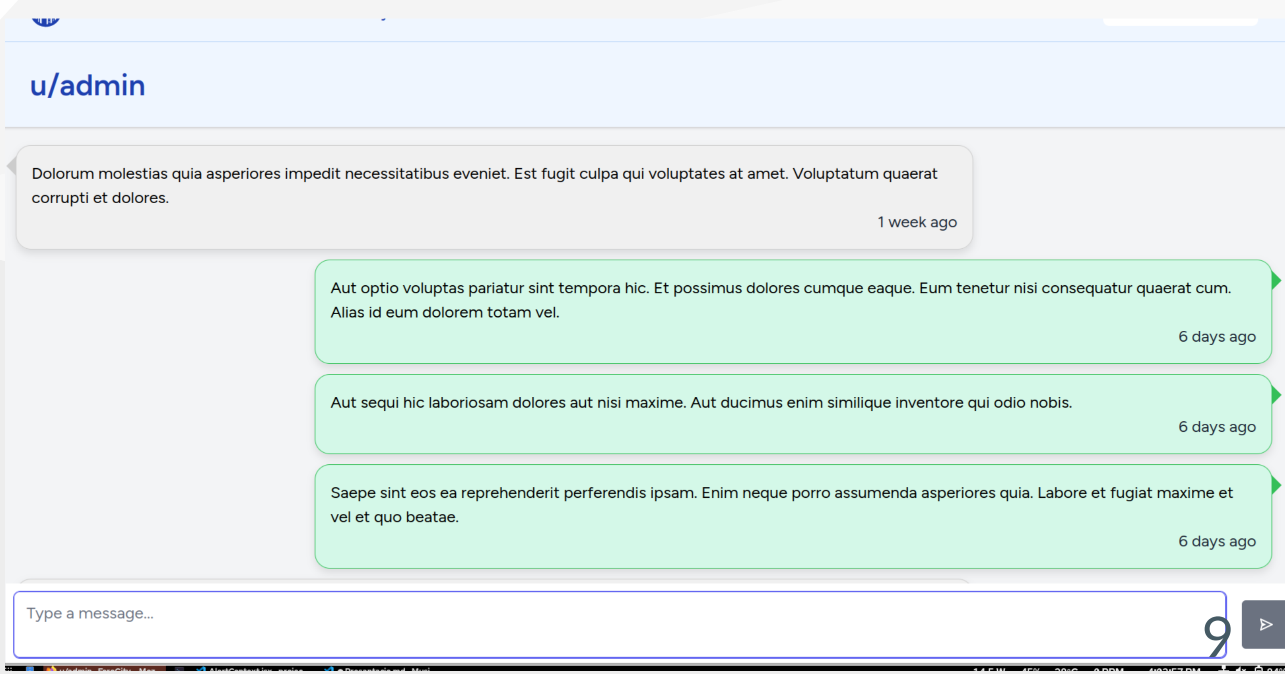
u/noemi76

Ut aut quos voluptatem quaerat maiores dolorum itaque. Et qui fuga sed id mollitia.

1 day ago

u/mkub

Voluptatem repellat commodi reiciendis dolorem consequatur. Recusandae quis omnis quis sed. Ni...



u/admin

Dolorum molestias quia asperiores impedit necessitatibus eveniet. Est fugit culpa qui voluptates at amet. Voluptatum quaerat corrupti et dolores.

1 week ago

Aut optio voluptas pariatur sint tempora hic. Et possimus dolores cumque eaque. Eum tenetur nisi consequatur quaerat cum. Alias id eum dolorem totam vel.

6 days ago

Aut sequi hic laboriosam dolores aut nisi maxime. Aut ducimus enim similique inventore qui odio nobis.

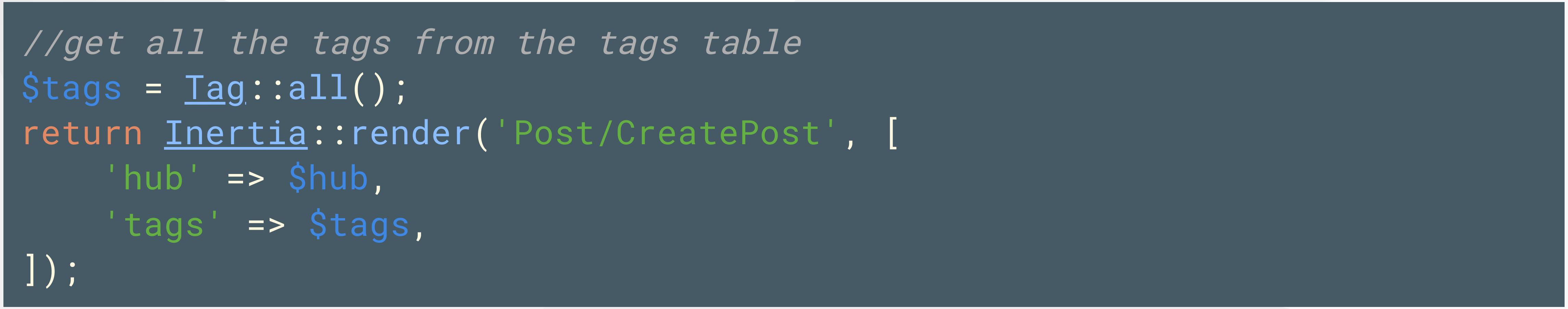
6 days ago

Saepe sint eos ea reprehenderit perferendis ipsam. Enim neque porro assumenda asperiores quia. Labore et fugiat maxime et vel et quo beatae.

6 days ago

How it is implemented - Passing Data

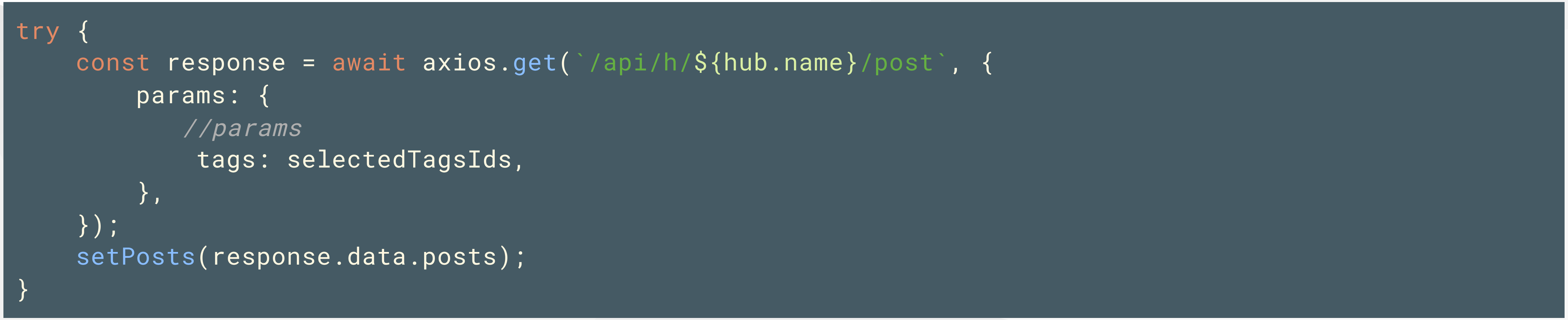
I have used Laravel and React with Inertia.js . The vew where i dont care if the pages are reloaded or not, data is directly passed from controller to view via props.



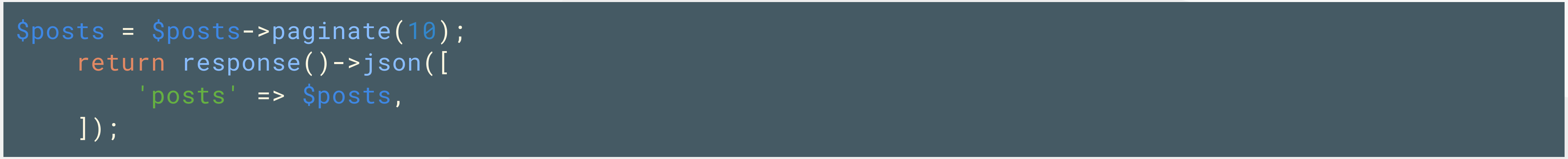
```
//get all the tags from the tags table
$tags = Tag::all();
return Inertia::render('Post/CreatePost', [
    'hub' => $hub,
    'tags' => $tags,
]);
```

How it is implemented- API

The pages where I dont want pages to be reloaded, I have used axios t get or pass the data.



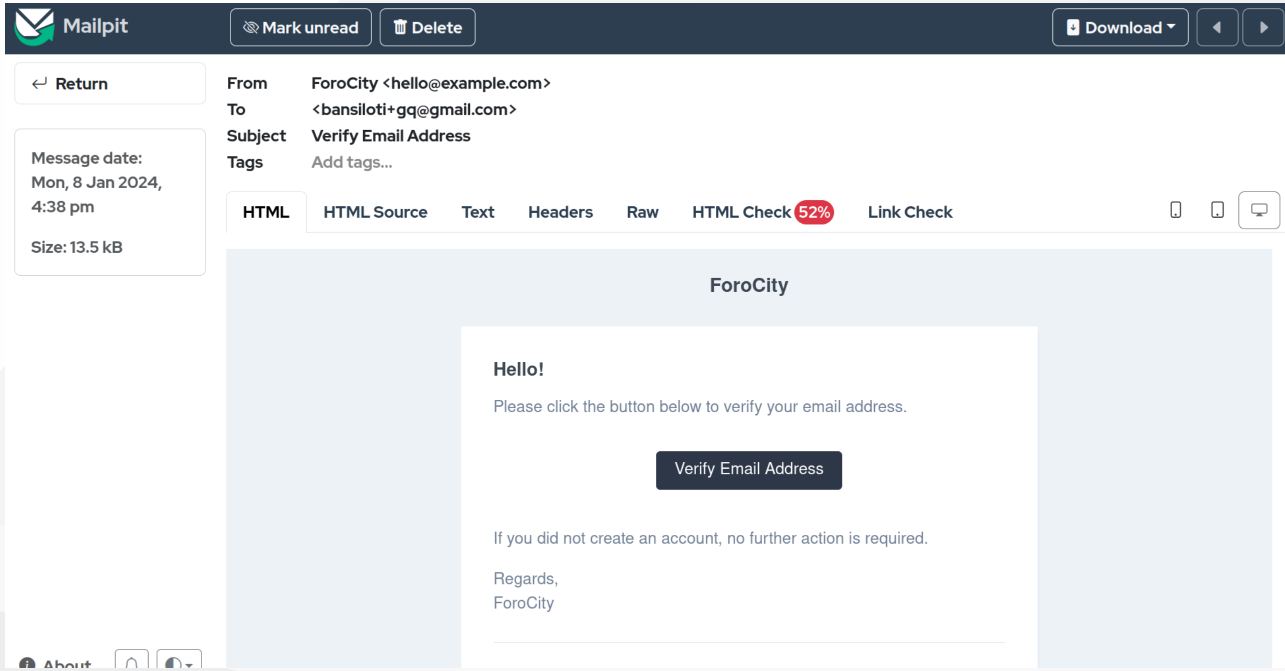
```
try {
    const response = await axios.get(`/api/h/${hub.name}/post`, {
        params: {
            //params
            tags: selectedTagsIds,
        },
    });
    setPosts(response.data.posts);
}
```



```
$posts = $posts->paginate(10);
    return response()->json([
        'posts' => $posts,
    ]);
```

User Authentication

I have used Breeze for authentication. It provides a simple way to authenticate users with email verification. To simulate the email verification, I have used a mailpit. It allows me to simulate emails to any email address.



Message date: Mon, 8 Jan 2024, 4:38 pm

Size: 13.5 kB

From ForoCity <hello@example.com>
To <bansiloti+gq@gmail.com>
Subject Verify Email Address
Tags Add tags...

ForoCity

Hello!

Please click the button below to verify your email address.

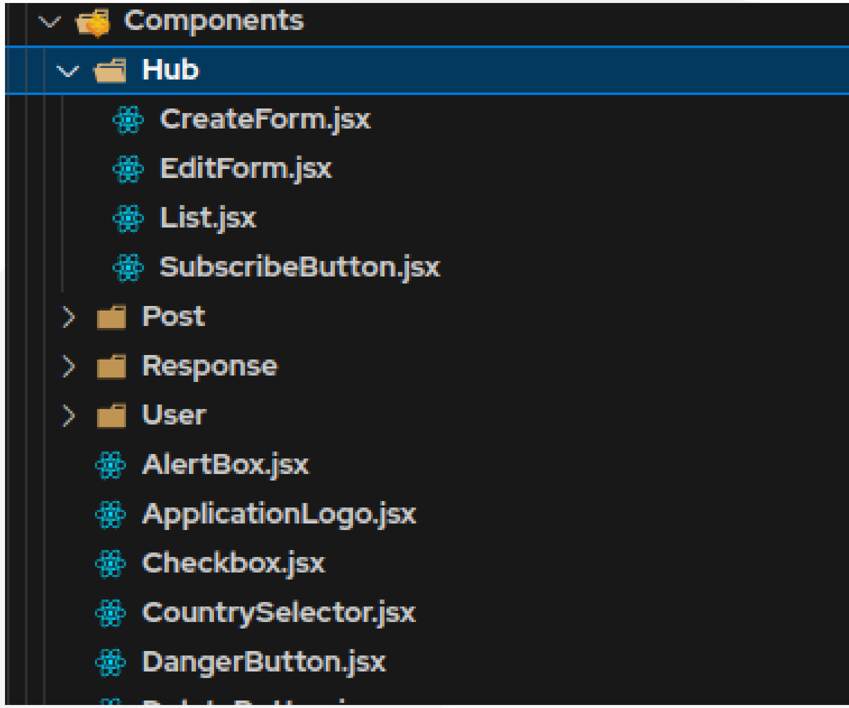
Verify Email Address

If you did not create an account, no further action is required.

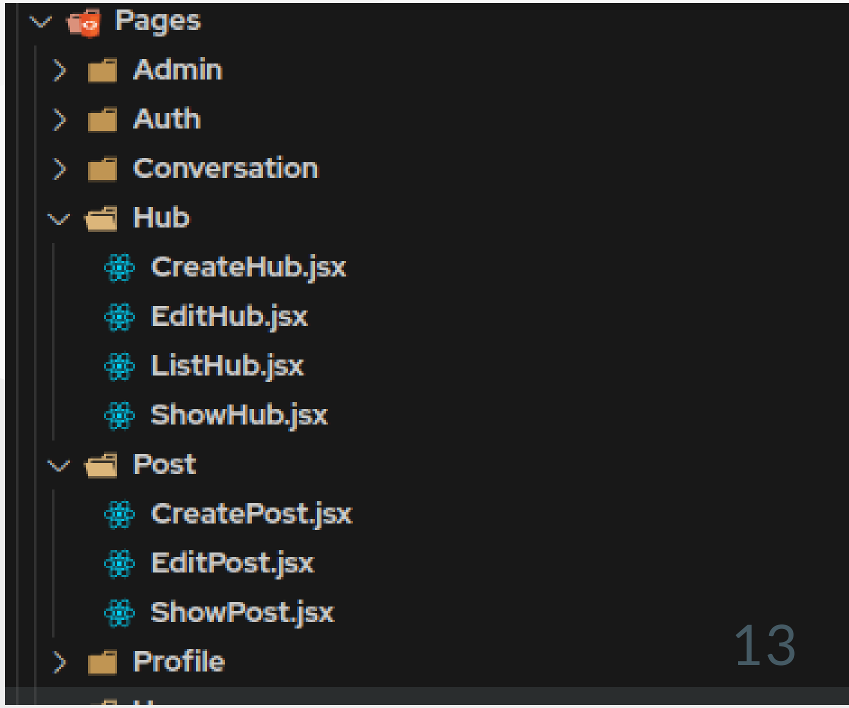
Regards,
ForoCity

Dividing the project

In pges I have tried to write least amount of code of possible and instead seperate the code into components. Components are intended to be reusable. I have also created a context called AlertContext. It is used to show alerts to the user.



- Components
  - Hub
    - CreateForm.jsx
    - EditForm.jsx
    - List.jsx
    - SubscribeButton.jsx
  - Post
  - Response
  - User
  - AlertBox.jsx
  - ApplicationLogo.jsx
  - Checkbox.jsx
  - CountrySelector.jsx
  - DangerButton.jsx



- Pages
  - Admin
  - Auth
  - Conversation
  - Hub
    - CreateHub.jsx
    - EditHub.jsx
    - ListHub.jsx
    - ShowHub.jsx
  - Post
    - CreatePost.jsx
    - EditPost.jsx
    - ShowPost.jsx
  - Profile

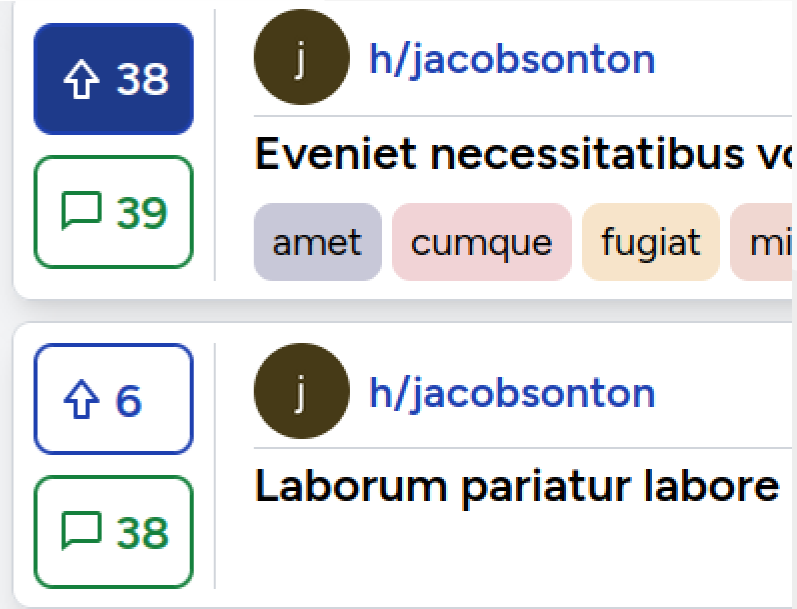
Challenges - Pagination with Api

As pagination of posts is done with api, I had to create a custom pagination component. It was a bit challenging to get the pagination links from the api response and then pass it to the component.

```
const fetchPosts = async (pageNumber = 1, query = "") => {
        try {
            const response = await axios.get(`/api/h/${hub.name}/post`, {
                params: {
                    page: pageNumber,
                },
            });
        }
```

Challenges - Reactiveness of the UI

When voting on a post or response, I had to update the UI without waiting for the response, So that it was instant and not wait 1-2 seconds. This is called optimistic UI. If the request fails, then just revert the UI back to the previous state.



Challenges - Sorting and Filtering

I have pagination while loading posts. So when I sort or filter the posts, I had to do them in backend. Becuse before I was doing it in frontend, so when I was changing the page, the posts were aonly sorted and filteres for only 10 posts.

Improvements that can be made

- **Converstaion**: It would be better if the conversation page was not reload when a new message is sent or received. Or receiving end automatically gets the new message, without reloading the page.
- **Get the total vount code of the users**: I had implemented this but it was ver slow, as first It has to load all the posts or responses of the user and then count the votes. So I had to remove it. Could have used a column in the user table to store the total votes.

External Libraries used - Laravel

- **spatie/laravel-medialibrary** - To upload images
- **inertiajs/inertia-laravel** - To use inertia.js with laravel
- **Mailpit** - To simulate email verification
- **Brezee** - To use authentication, and some starter views
- **fakerphp/faker** - To generate fake data for testing
- **barryvdh/laravel-debugbar** - To have a debug bar in the browser that shows the queries and other information

External Libraries used - React

- **react-icons** - To use icons, like voting, delete, edit, etc
- **react-select** - To use select box for tags and country, customisable.
- **react-select-country-list** - To get the list of countries for the select box.