

COMP2011

Saif Elzegheiby

201687022

<https://seifzogheibi.pythonanywhere.com/>

email: seifzogheibi@gmail.com

password: seif

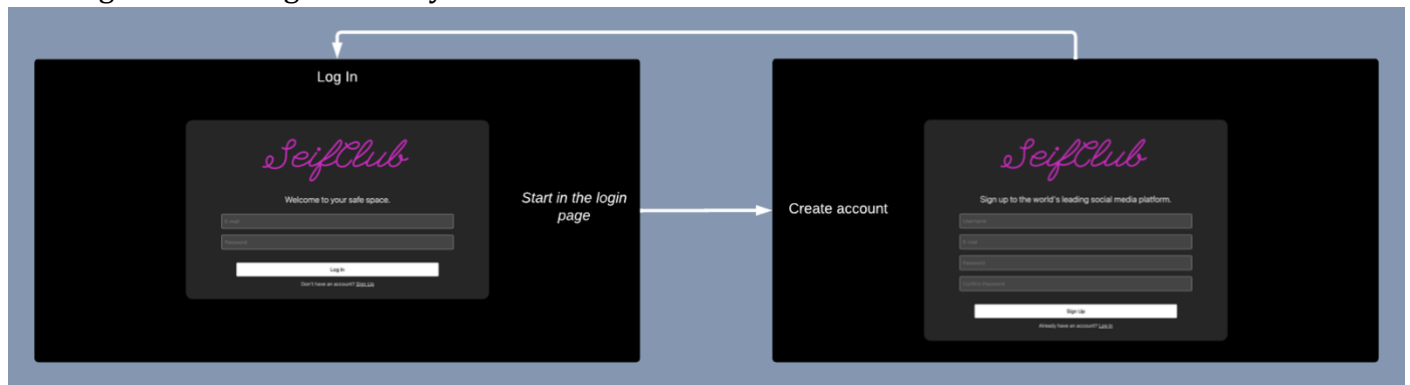
(or create account)

My Website

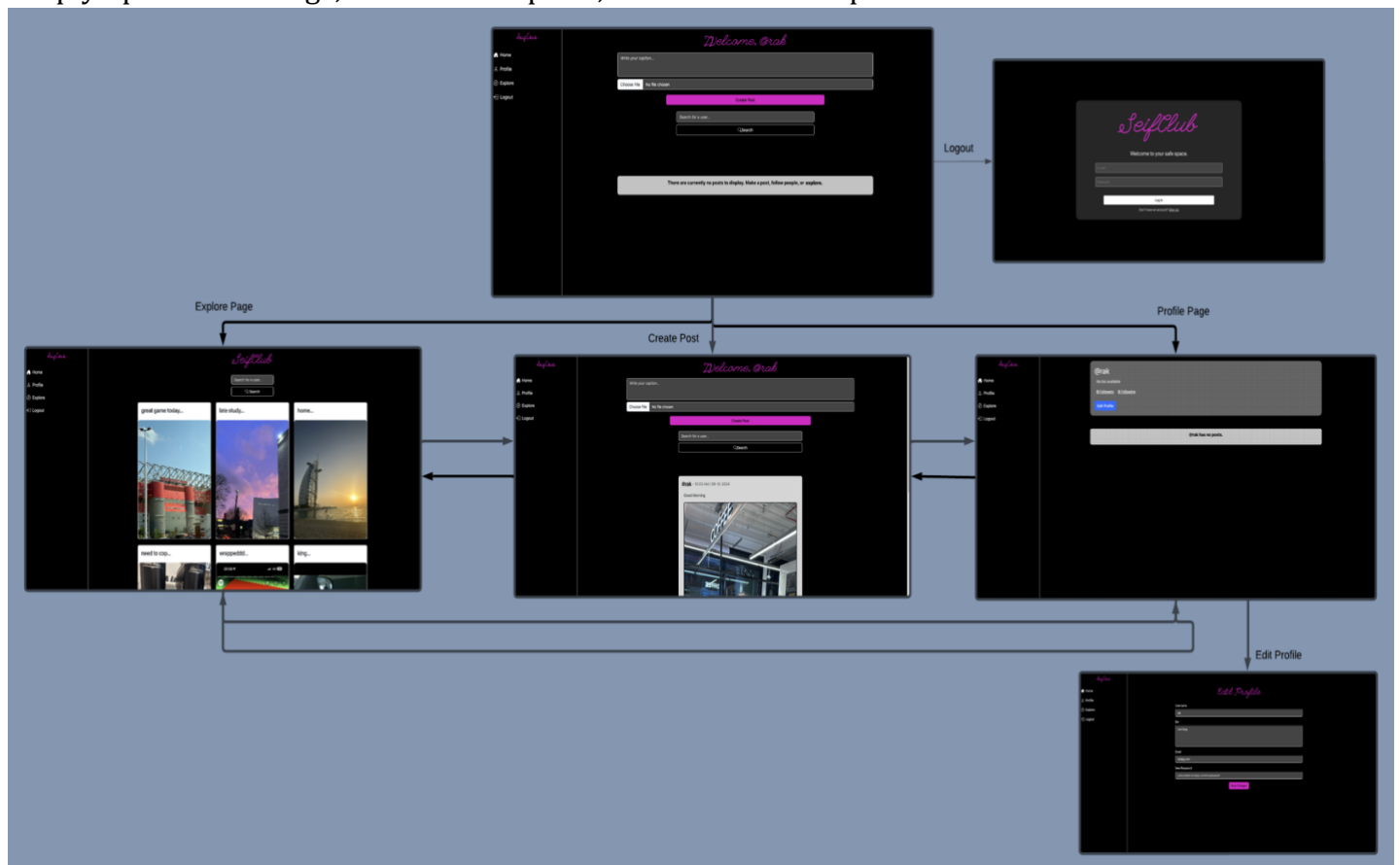
SeifClub is a social media web application platform, which allows users to create, view, and interact with each other's posts. Users can also follow each other and keep track of their most recent posts via the Home Page (dashboard). SeifClub also allows users to edit their profile by updating their personal information.

User Path:

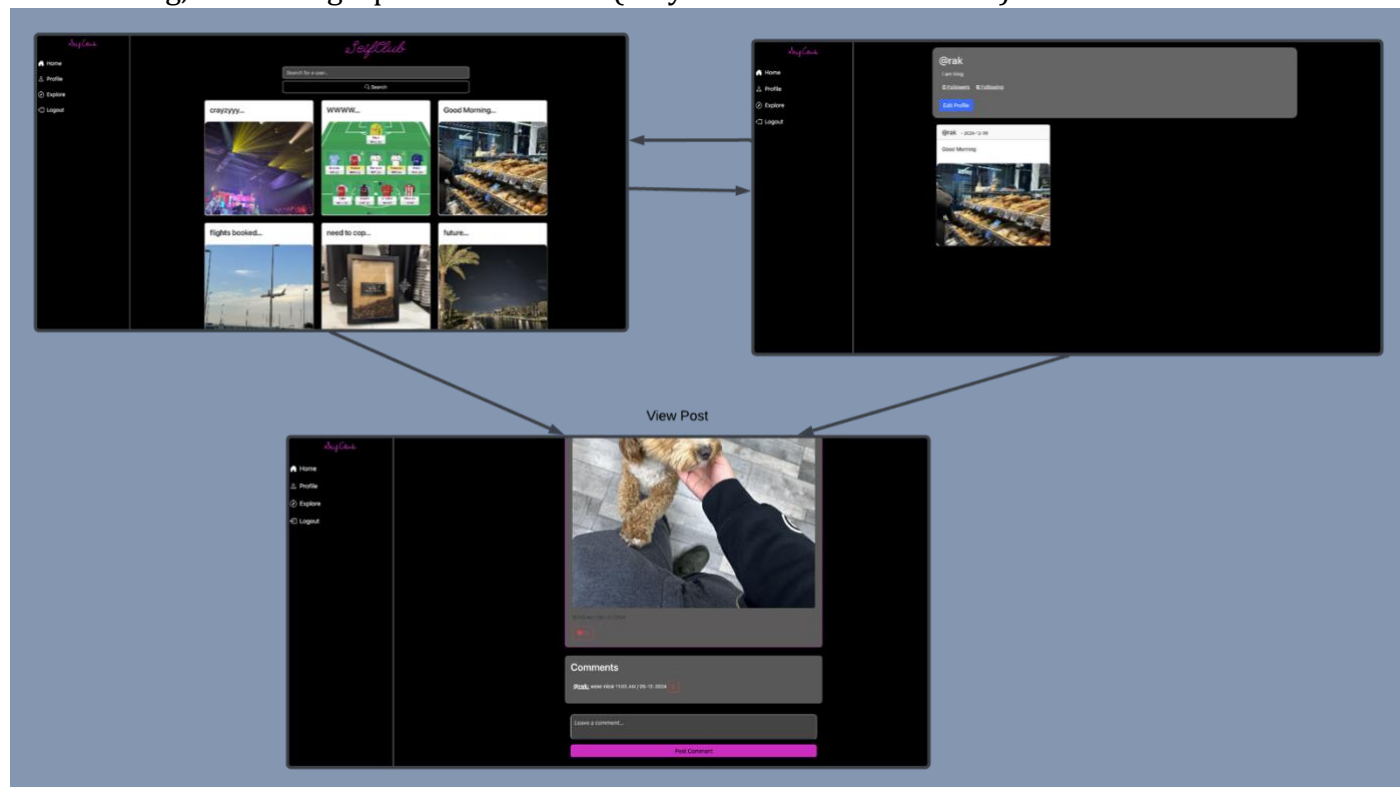
The user is greeted with a subtle form that requires them to enter their credentials before accessing the website. New users will have to create an account by clicking on "Sign Up", while existing users can log in directly.



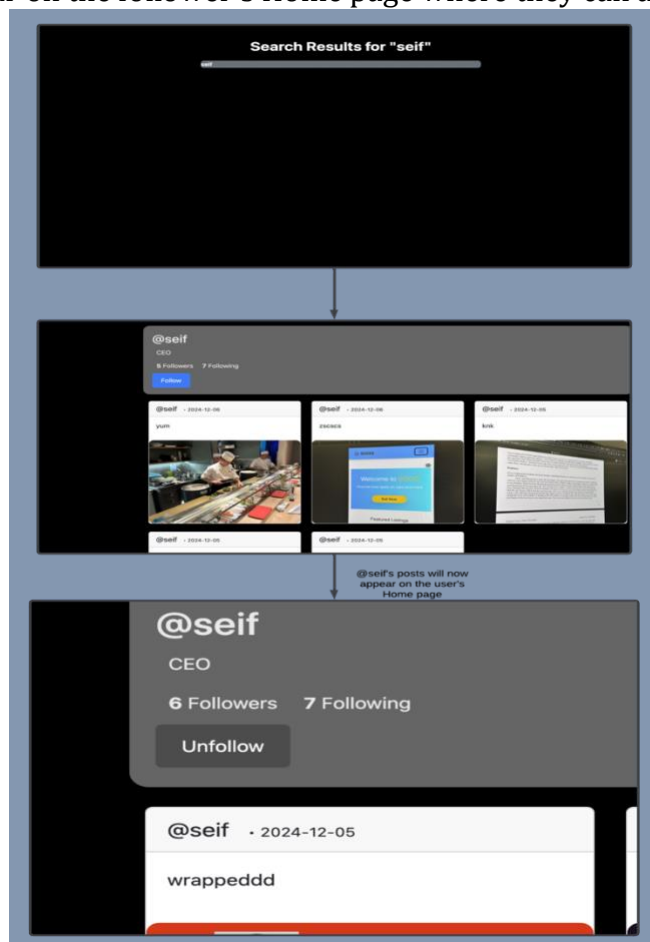
Once the user is logged in, they are redirected to the Home Page (Dashboard) where several features and paths are available. These features include creating posts, which is the core function of the application, or searching for users, or browsing through an uncapped number of random posts by other users on the Explore Page. The user can also go to their profile where they can make changes to their account, such as adding a bio, or changing their personal information. To create a post, the user simply uploads an image, and adds a caption, then clicks on the post button.



Besides creating a post, SeifClub includes other forms of engagement such as liking, commenting, or deleting a post or comment (only available to the author).



The app also features a following system where users can find each other through the search bar available on both the Home and Explore Pages. When a user follows another user, the followed person's posts will appear on the follower's Home page where they can also interact with them.



Design:

The application uses a simple, but elegant layout with logical navigation, and follows the favoured standards for readability and accessibility. The color palette focuses mainly on two colors: pink and black. The choice of these colors was inspired by a family member's artwork for a local gallery. Moreover, the cursive font on the logo and welcome page, add a sense of elegance to the app, especially when combined with the bright pink color.



SeifClub also includes JavaScript functions that allow some features to behave dynamically. For example the like button updates the count without reloading the page. Same thing with the delete buttons that don't refresh the page after executing. The JavaScript snippet below allows the like button to update its like count, without reloading the page. It also includes some Bootstrap features for added quality. On the right is an example of one of a few media queries that were used to improve responsiveness across the website.

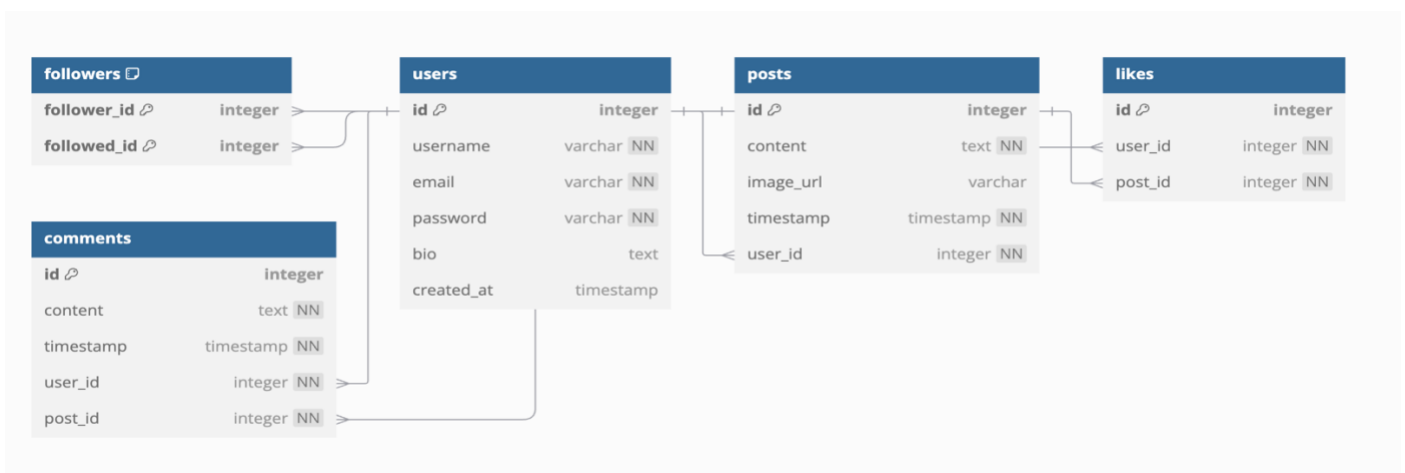
```
fetch(`/like/${postId}`, {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
})
.then((response) => {
  if (!response.ok) {
    throw new Error("Failed to like/unlike the post");
  }
  return response.json();
})
.then((data) => {
  const icon = button.querySelector("i");
  if (data.status === "liked") {
    icon.className = "bi bi-heart-fill";
    button.innerHTML = `<i class="${icon.className}"></i>${data.like_count}`;
  } else if (data.status === "unliked") {
    icon.className = "bi bi-heart";
    button.innerHTML = `<i class="${icon.className}"></i>${data.like_count}`;
  }
})
```

```
@media (max-width: 767px) {
  .edit-profile-container {
    margin-left: 20px !important;
  }

  .edit-profile-container h5 {
    font-size: .6em !important;
    margin-right: 50px;
  }
}
```

Database:

The database holds many tables that all store specific information, such as Users, Posts, Comments, Likes, and Followers. The relations are in one-to-many, and many-to-many. For the many-to-many relation, it is designed for many users being able to follow other users, allowing them to track their posts on the home page.



Accessibility

SeifClub uses contrasting colours at every point on the page to be accessible for visually impaired users. In addition, the app is built on HTML tags that help optimize screen reader compatibility. In addition, all posts have alt tags that informs screen readers which posts it is looking at.

```

```

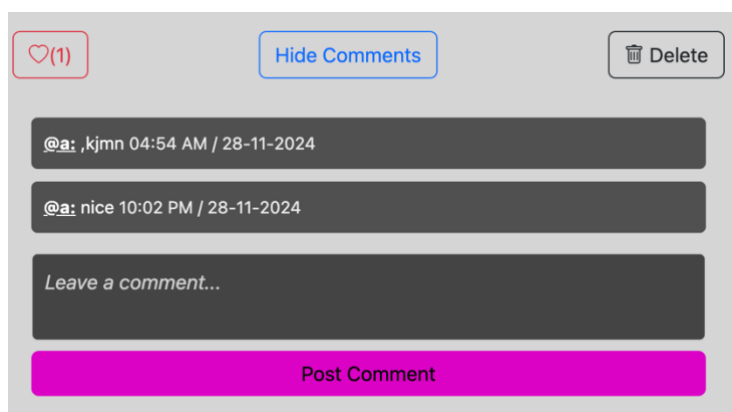
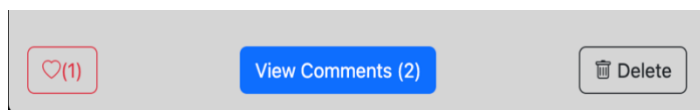
Advanced Feature

The application uses AJAX to dynamically update the like count without reloading the page. The snippet below demonstrates how this is done.

```
fetch(`/like/${postId}`, {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
})
.then((response) => {
  if (!response.ok) {
    throw new Error("Failed to like/unlike the post");
  }
  return response.json();
})
.then((data) => {
  const icon = button.querySelector("i");
  if (data.status === "liked") {
    icon.className = "bi bi-heart-fill";
    button.innerHTML = `<i class="${icon.className}"></i>${data.like_count}`;
  } else if (data.status === "unliked") {
    icon.className = "bi bi-heart";
    button.innerHTML = `<i class="${icon.className}"></i>${data.like_count}`;
  }
})
```



Another special feature is the toggle button on the comments that allows the user to view and hide the comments under the post. This allows for a less cluttered dashboard and maintains a consistent layout and post size across the app.



```
if (button) {
  const commentsContainer = button.closest(".dashboard-posts-cards").querySelector(".comments-container");

  if (commentsContainer) {
    commentsContainer.classList.toggle("hidden");

    const commentsCount = commentsContainer.querySelectorAll(".comment").length;

    button.textContent = commentsContainer.classList.contains("hidden")
      ? `View Comments (${commentsCount})`
      : `Hide Comments`;
  }
};
```

Testing & Critical Analysis

Testing

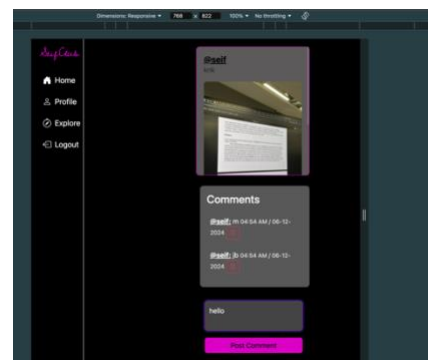
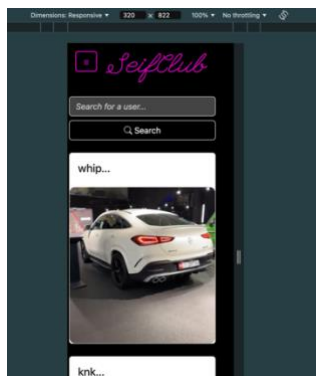
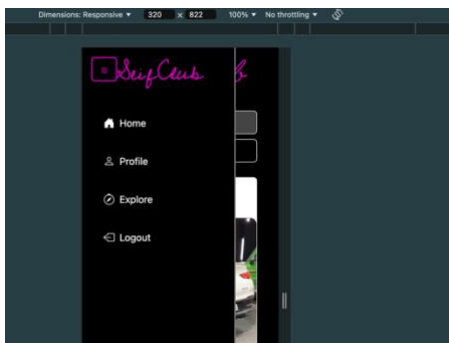
Functionality

To ensure better functionality, multiple debugging statements were placed and displayed in the console logs and terminal. Below is the terminal, report everything that is happening on the backend side, for example. when a button is clicked or display all the referenced pages.

```
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/uploads/IMG_3397.jpeg HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/css/profile.css HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/uploads/IMG_2902.jpeg HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/css/edit_profile.css HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/js/password.js HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/js/toggle.js HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/uploads/IMG_3404.png HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/js/like.js HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/js/delete_comment.js HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/uploads/IMG_2792.png HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:33:23] "GET /static/uploads/IMG_3439.png HTTP/1.1" 304 -
127.0.0.1 - - [06/Dec/2024 13:43:29] "POST /like/31 HTTP/1.1" 200 -
127.0.0.1 - - [06/Dec/2024 13:43:30] "POST /like/31 HTTP/1.1" 200 -
127.0.0.1 - - [06/Dec/2024 13:44:12] "POST /like/3 HTTP/1.1" 200 -
```

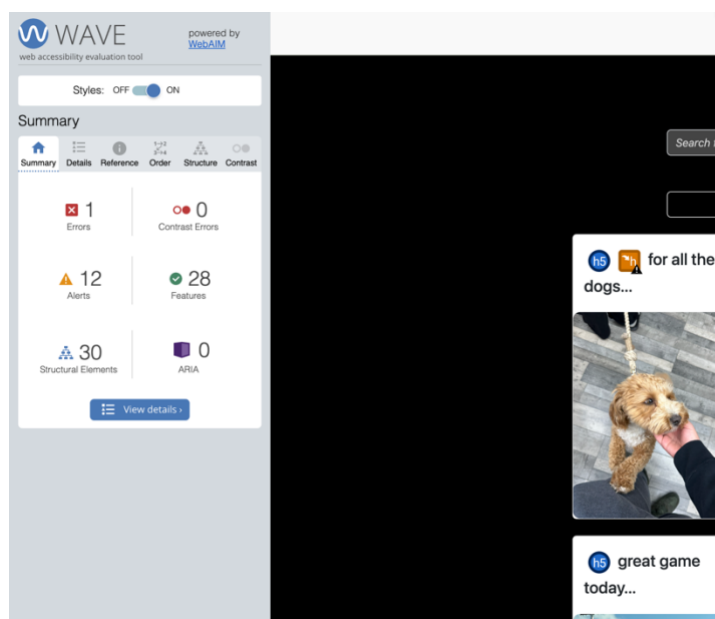
Layout & responsiveness

Using the Inspect feature built in the Google Chrome browser, the pages were tested against different dimensions to ensure responsiveness and consistent layouts.



Accessibility

Each page was checked for Web Content Accessibility Guideline (WCAG) errors by Web Accessibility Evaluation Tools (WAVE) to achieve standard accessibility requirements. Furthermore, manual keyboard commands like (CMD+Up) were tested to ensure smooth keyboard navigation.



Analysis

What Went Well

Implementing the special features (dynamic content) was an enjoyable task, as the use of AJAX aligns with my previous experience in JavaScript. The addition of this feature was also necessary to deliver a smooth and user-friendly application.

The final successful launch of the website was a great and rewarding feeling. The overall experience from the planning, to building the database, to then implementing the sketched designs and making it all come together to deliver this project made it a memorable moment in my academic career.

What Went Badly

One of the challenges that were overcome in the development stage was making sure all pages were responsive and clean across all devices. The reason for this was that it only occurred halfway through the project, and by that point the HTML layout and design was pure CSS which caused issues like overlapping content. I was then advised by my peers to develop the website using Bootstrap 5. However, in doing this, it caused a clash between Bootstrap functions and my CSS classes. The code was then cleaned up, but it is safe to say that this was the most time-consuming and difficult challenge to overcome.

Improvements

It would be interesting to see a version of this website with different colour palettes, or an edit post button that would give users more control over their posts. Something else I would consider is working on an offline workspace like VSCode instead of GitHub, this is because there were many times I tried to work but there was no internet available. Another reason is that I ran out of working hours because the university account has a specific number of hours available each month, and when I reached this limit I had to pay to upgrade my account.

In terms of features, I really admire the impact that dynamic content can have on a website. For this reason, I would definitely hope to include more of these features like making the Explore Page sort the posts by preferences based on the user's likes. Additionally, I would add a "People You May Know" box in the home page that recommends a few users to the primary user based on linking the their followed accounts' followers and following.