# Lab 2 - Dynamic webpages

Start Assignment

**Due** 11 Feb by 23:59 **Points** 1 **Submitting** a file upload
**Available** until 11 Feb at 23:59

## Overview

Develop a comprehensive web application showcasing posts, user profiles, comments, and a navigation system. The application should dynamically fetch data, incorporate a responsive flexbox design, and provide an engaging user experience.

**Features and Requirements**

1. **Posts Display**
    - Dynamically load and display posts containing an ID, title, body, userId, tags, and reaction count.
    - Implement pagination or infinite scrolling for loading posts.
2. **Navigation Bar**
    - Develop a responsive navigation bar with links to various sections (e.g., Home, Posts) that sits on the top of all pages.
3. **Home Page**
    - Design a Home page describing the application, purpose, and user instructions.
4. **Posts Page**
    - A list of posts
    - Each post should have
        - Title
        - Body
        - Username (How can you get username from id?)
    - each post should have its associated comments attached to it.

1. **Error Handling and Feedback**
    - Manage network request errors.
    - Provide messages for empty states or errors.

### Technical Requirements

1. **HTML**

- Structured markup for posts, user profiles, comments, and navigation elements as well as a contact form
  - The contact form should have field for "Name", "Email", "Send" button as well as a checkmark for "confirm to send"
- Make sure to use semantic HTML tags throughout your code.

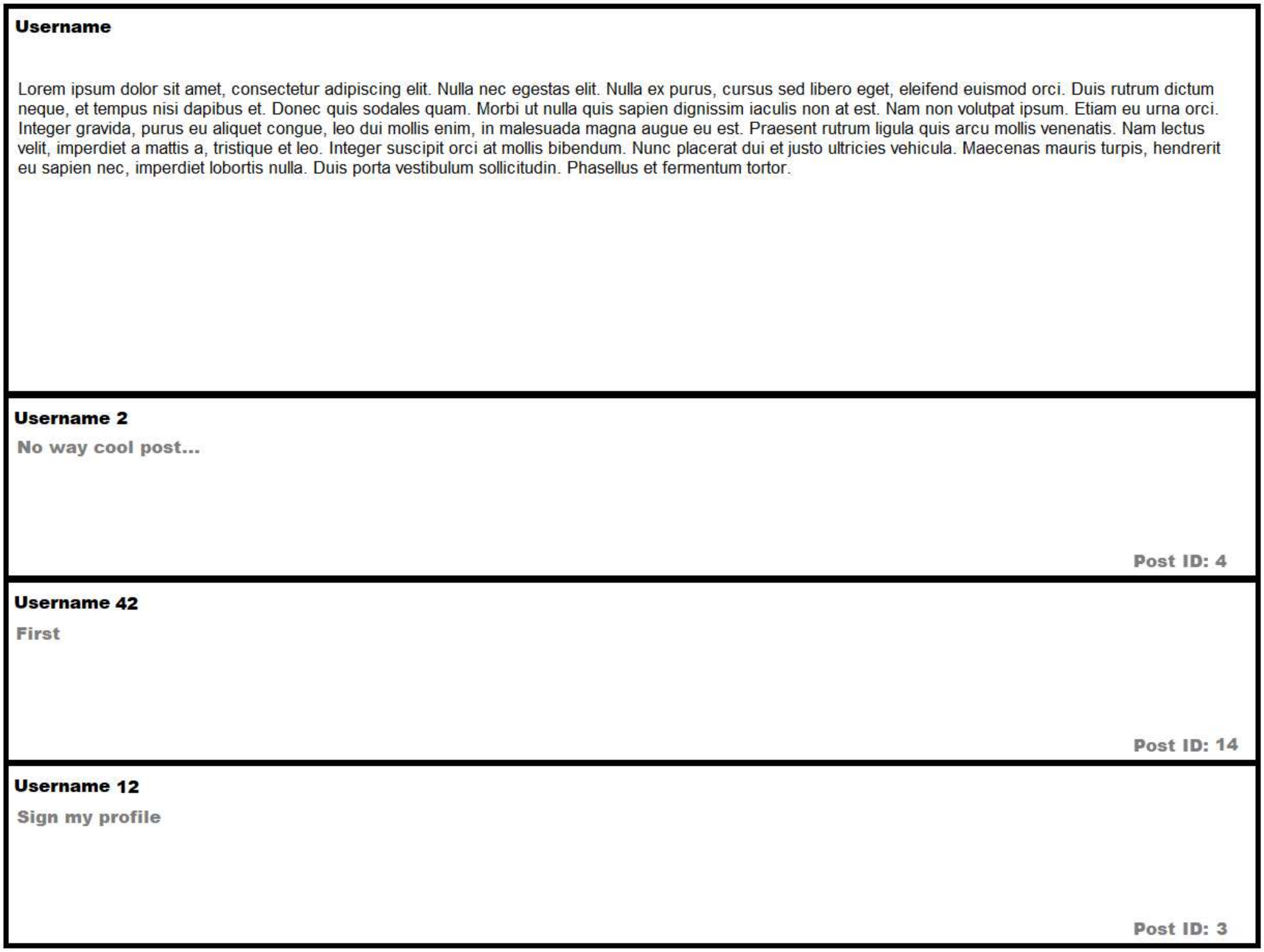
2. **CSS**

- Use a flexbox or grid for responsive layouts.
- Apply styling to make the application visually appealing and ensure usability across various devices.

3. **JavaScript**

- **Fetch Post Data :** Use `fetch` to get posts from from `https://dummyjson.com/posts (https://dummyjson.com/posts)`
- **Fetching Comments**: Use `fetch` to get comments from `https://dummyjson.com/comments (https://dummyjson.com/comments)`
- **Displaying Comments**: Create a functionality that appends comments to the respective post.
- **Fetching User Data**: Utilize the `fetch` to retrieve user profiles from `https://dummyjson.com/users (https://dummyjson.com/users)`
- **Linking Posts to User Profiles**: Add functionality where clicking on a username in a post opens a model with user's profile information from the "users" endpoint.
- **User objects:** Each user should have either an object or a class (ie after the users are read in from the API they should be re-organized to a javascript object or class).
- **Validate the contact form:** Add functionality where the contact form is validated at submission
  - Name should not contain integers
  - Email has to contain "@" and "."
  - Checkmark has to checked for form button to be clickable.

Post with comment structure could look like this (made in paint):



**Username**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla nec egestas elit. Nulla ex purus, cursus sed libero eget, eleifend euismod orci. Duis rutrum dictum neque, et tempus nisi dapibus et. Donec quis sodales quam. Morbi ut nulla quis sapien dignissim iaculis non at est. Nam non volutpat ipsum. Etiam eu urna orci. Integer gravida, purus eu aliquet congue, leo dui mollis enim, in malesuada magna augue eu est. Praesent rutrum ligula quis arcu mollis venenatis. Nam lectus velit, imperdiet a mattis a, tristique et leo. Integer suscipit orci at mollis bibendum. Nunc placerat dui et justo ultricies vehicula. Maecenas mauris turpis, hendrerit eu sapien nec, imperdiet lobortis nulla. Duis porta vestibulum sollicitudin. Phasellus et fermentum tortor.

**Username 2**

**No way cool post...**

**Post ID: 4**

**Username 42**

**First**

**Post ID: 14**

**Username 12**

**Sign my profile**

**Post ID: 3**