

Let's build an app that will act as a frontend client for a social media platform(have any better ideas...?).

General guidelines

1. Make sure to properly organize your code.
2. Make sure to not repeat code if it can be avoided.
3. Remember ***decoupling***.
4. Make sure you understand your own solutions and the decisions you made in order to build it.
5. Make sure to properly document and format your code.

Reminders

1. For CSS *selectors*, we use `#` to denote the `id` attribute, and the `.` sign to denote the `class` attribute. Wire your documents accordingly.
2. [CSS validation](#)
3. Feel free to use any layouting tools you may need.
Examples - tables, ordered list.
You can also position the elements based on their natural behaviour, and/or using css rules.
- 4.

Specification and requirements

The application should be able to:

1. Display an endless feed/stream of items/posts. We will call it *the main feed*.
2. Users should be able to sort the items/posts.
3. Allow easy and circular navigation between different parts of the application.
4. Users should be able to register.
5. Users should be able to send new items/posts.
6. Users should be able to comment.
7. Each post should have comments.
8. Each item/post/comment should be *'likeable'*.
9. **All** input must be validated on the frontend in a meaningful way(we do not have to use javascript for that).
10. Have a dedicated page for an endless feed of *video and audio* media. You can also suggest a way to implement directly under *the main feed*.
11. The frontend must be visually appealing.

Bonus

Since we still do not have a server to send our requests to, here is a way to invoke an action when submitting a form so that we can have a visual representation of the action:

```
<form onsubmit="alert('submit!');return false">
```

```
  First: Enter in the input field <input type="text" value="text"><br>
```

```
  Second: Click "submit": <input type="submit" value="Submit">
```

```
</form>
```

Using the above logic:

1. Print the content of the form to the screen(can you think of more than one way to do it?)
2. Create a pretty element to display the confirmation that the form was sent.
 - a. What would be useful to display in such situations?
 - b. Are there any other situations that we might need to take into account?
 - c. How would we deal with such situations?(if, in fact, they exists)
3. We can also redirect the user to the same page, or some other page. On a different form, try to send the user to another meaningful page upon submission.

Extra bonus - store useful information in a cookie. Use that information to make the user feel more welcome.