

Fakestagram

Submission Details

- Check the AnimoSpace submission page for instructions

Site to Recreate

- Recreate a simple Social Media Profile app
- The required HTML and CSS is already provided
- Recreate the site as close as possible to the reference image in Figure 2
- Some comments are present in the files to guide you

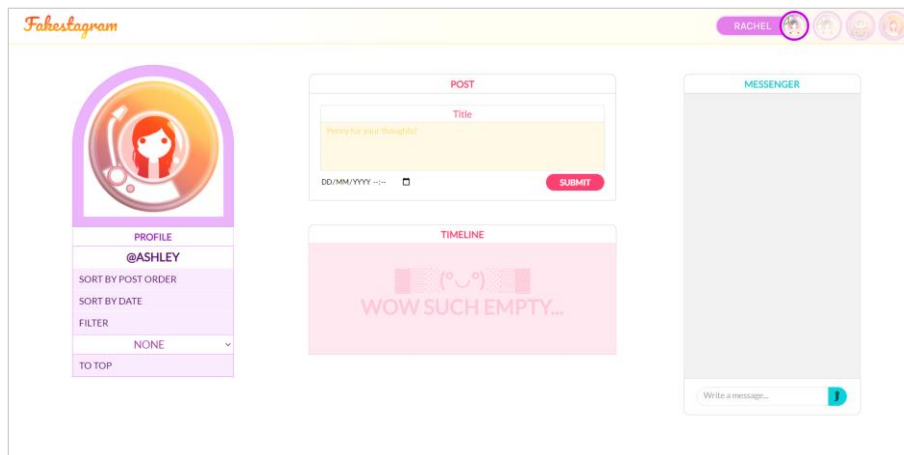


Figure 1. Initial Appearance

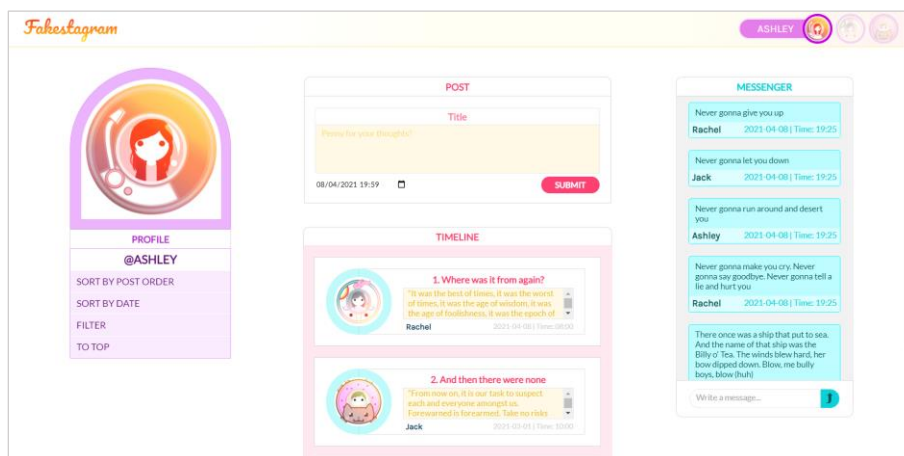


Figure 2. Correct Appearance

The user icon selected on the top bar will be the **current user**. By default, it is set to Rachel. To **change users** simply click a different user icon. The code for changing users is already provided in the starter JS code.

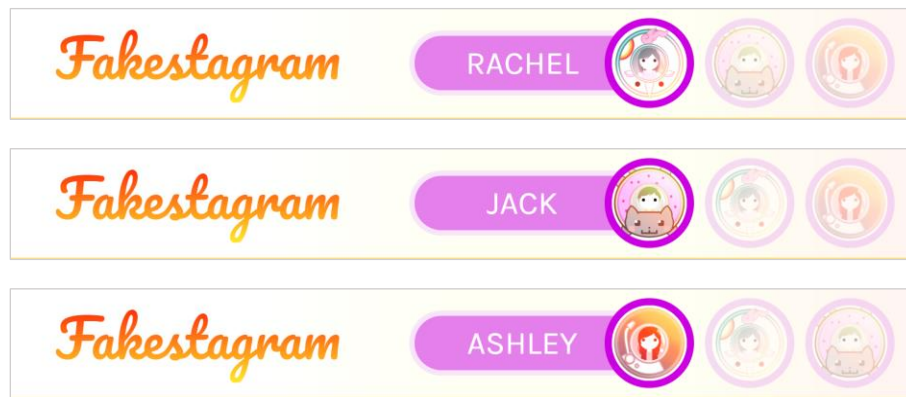


Figure 3. (Top) Current User is Rachel, (Middle) Current User is Jack, (Bottom) Current User is Ashley

The user is required to **create a post** and set the title, content, and date & time. As a default, initialize the post's Date & Time to the current date and time when the page is opened or reloaded. **The code for creating and formatting the Date & Time is already provided in the starter JS code.** It is placed in a function called `formatDate(objDate);`, where `objDate` is a Date object. The function returns the Date & Time in this format: `yyyy-MM-ddThh:mm`. Note that this is an acceptable format for a date input type value. **HINT!** For easy object creation, you can declare an Object constructor ([w3schools link](#)).

The image shows a form titled 'POST' in red. Inside the form, there is a section for 'Title' with a text input field containing the placeholder text 'Penny for your thoughts?'. Below the title field is a larger text area for the post content. At the bottom left of the form, there is a date and time input field showing '07/04/2021 22:03' and a calendar icon. At the bottom right, there is a red 'SUBMIT' button.

Figure 4. Create Post

Submitting a post is only **valid** if its Title and Content are filled. If the submitted post is **invalid**, the user is shown an error message (see Figure 5 below). **Only one error message can be shown at a time.** The error message for **Title** takes precedence over the error message for Content. The code for showing an error message is already given in the starter JS Code. However, you are still tasked to properly call it upon post submission. Call `showError(message);` to make the error message container appear. The parameter 'message' can either be the `errorTitle` variable or the `errorContent` variable given in the JS starter code. **DO NOT** edit the contents of these variables.

When an error message is visible, clicking on either the Title input box or the Content textarea should hide it. Call the `hideError();` function to do this. The function declaration is provided in the starter code, but it is up to you to properly call it.

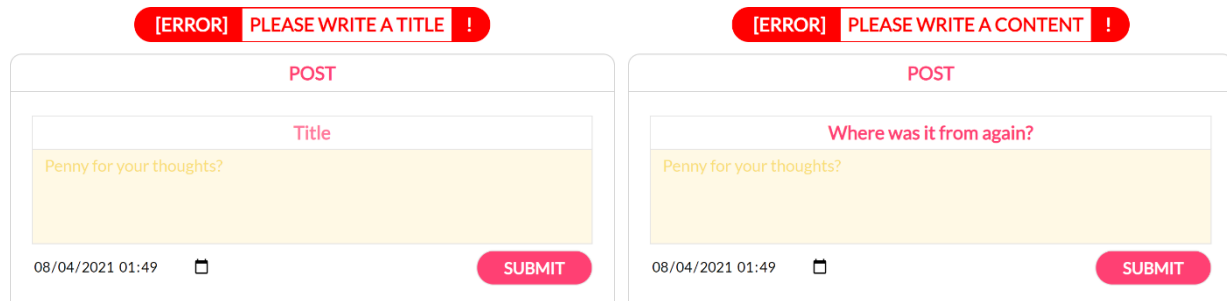




Figure 5. (Left) Error Message for Title, (Right) Error Message for Content
Figure 6.

If a submitted post is **valid**, proceed in adding a new post to the Timeline (or post-container). On initialization, the Timeline should contain the text “ (°◡°)  Wow such empty...” as shown in the left image of Figure 6 below. When filled, the Timeline should contain the submitted Single Post. When new posts are added, the Timeline should expand and add new posts at the very end of it. Please refer to the right image for an example of this. The Create Post fields should also **reset** once a valid post is submitted. Its Title and Content should be empty and its Date & Time should be set to the Date & Time right after the post was submitted. There is no limit as to how many posts can be added to the Timeline. Remember that the poster’s name is dependent on the **current user**.

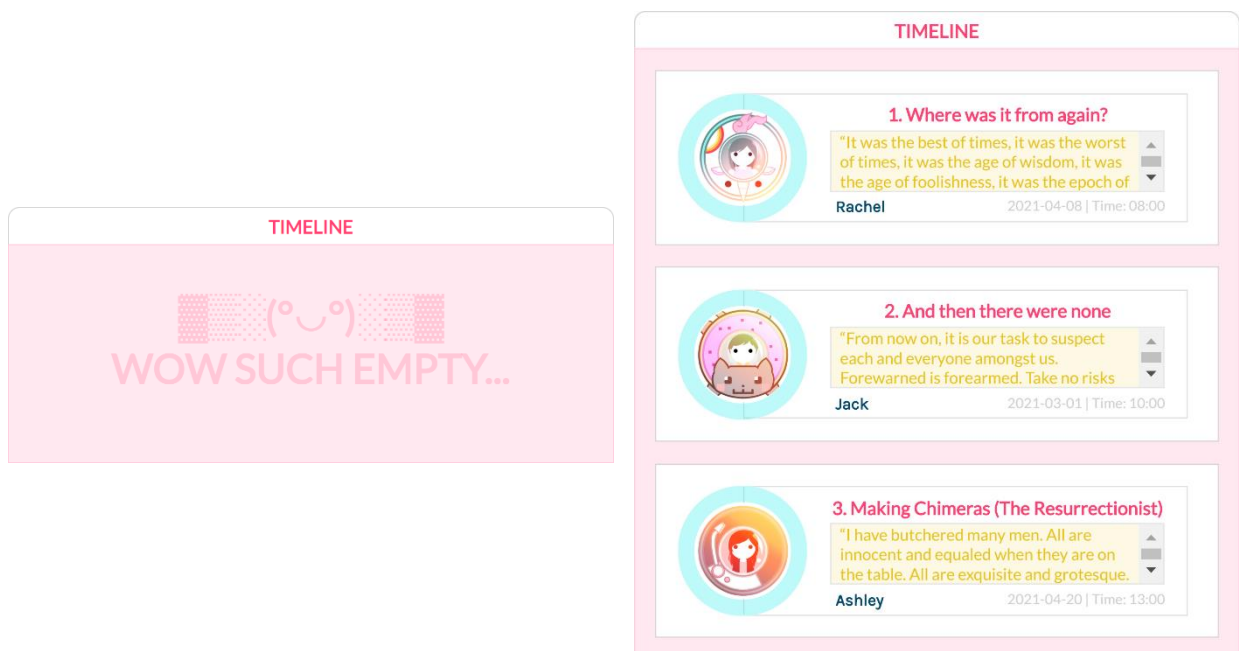


Figure 7. (Left) Empty Timeline, (Right) Timeline containing 3 Posts

A Single Post should look like Figure 7. The styles for this are already given in the `style-post.css` file. The description of the classes are show in Table 1 below.



Figure 8. Single Post

Class Name	Description	Class Name	Description
<code>.single-post-main</code>	outer-most parent container, contains the outer white border	<code>.sp-title</code>	contains styles for the post title
<code>.single-post</code>	<code>.single-post-main</code> 's child, parent container of all post elements	<code>.sp-body</code>	contains styles for the post body
<code>.sp-left</code>	first child of <code>.single-post</code> , contains all the left elements of the post	<code>.sp-right-bottom</code>	parent container for the footer elements in <code>.sp-right-content</code>
<code>.sp-right</code>	second child of <code>.single-post</code> , contains all the right elements of the post	<code>.sp-name</code>	contains the styles for the poster's name
<code>.sp-picture</code>	<code>.sp-left</code> 's child, contains styles for the user's picture	<code>.sp-date</code>	contains the style for the post date in the format: yyyy-MM-dd Time: hh:mm
<code>.sp-right-content</code>	<code>.sp-right</code> 's child, for margin help		

Table 1. Class Names and their Description

Note that when writing the date in Timeline posts, the proper format is: yyyy-MM-dd | Time: hh:mm. So if the date was December 25, 2021, 01:30 PM, it should be written as: 2021-12-25 | Time: 13:30. **HINT!** The `formatDate(objDate);` function might be able to help you with date formatting.

When showing the post Title in the Timeline, it will also contain the post Number. So the Title format is: <number>. <title>. If a post with a title of "And then there were none" was posted, and it is the second post, the Title will show "2. And then there were none". There is a space after the period and before the Title text.

The left side of the website contains the Profile Pane shown in Figure 8. The picture and profile name (@Ashley) are **static** (do not change their contents). The buttons in the Profile Control Pane (underneath the @Ashley profile name) are actually DIVs. Their IDs are as follows: sort-by-order, sort-by-date, filter, to-top. The hover styles are already given. Please write their functionalities as described in the following paragraphs.

The Sort by Post Order button sorts the Timeline posts by their post Number. This is actually the same as the default Post Order. This button is mainly used to return the Timeline's Post Order whenever Sort by Date is clicked.

The Sort by Date button sorts the Timeline posts by their Date & Time. Since the user can change their post Date & Time during Create Post, their post Number isn't necessarily in the same order as their post Date & Time. **NOTE!** If you're having trouble with this feature, implement it last. It can be considered optional since I still have a way of testing your Sort by Post Order button in my end.

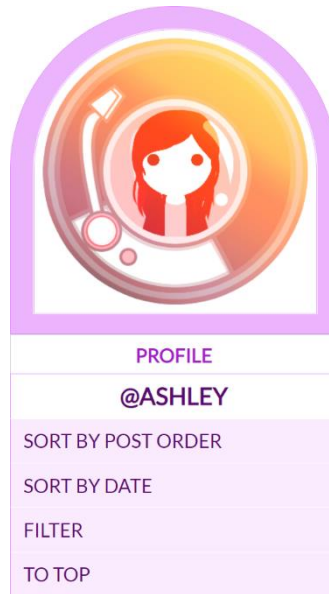


Figure 9. Profile Pane

The Filter button enables and disables filtering. When the Filter button is clicked, it shows the Filter drop-down box (sometimes called select, due to the select tag). The enabled and disabled Filter drop-down is shown in Figure 9 below.

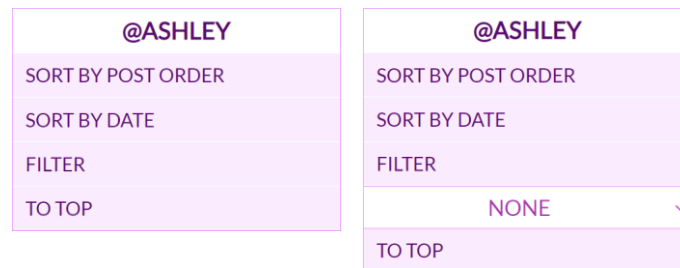


Figure 10. (Left) Profile Control Pane with Filter Off, (Right) Profile Control Pane with Filter On

The Filter drop-down has 4 options, namely: None, Rachel, Jack, Ashley. Selecting None removes the Filter. Selecting Rachel filters the Timeline to only shows posts by Rachel. Selecting Jack filters the Timeline to only shows posts by Jack. Selecting Ashley filters the Timeline to only show posts by Ashley. If the filtered Timeline results in an empty Timeline, the Timeline should contain the “👤(°◡°)👤 Wow such empty...” text once more.

Hiding the Filter drop-down (Filter Off) should disable the current filter, but the value of the Filter drop-down after re-showing it (Filter On) should still be the last selected drop-down option. It should also apply the selected Filter whenever Filter is On.

The To Top button returns the view to the top of the page. Since the Timeline post-container expands with the number of posts, the user has to scroll down to view posts at the end of the Timeline. The To Top button should return the view to the top of the page so that the user doesn't have to manually do so. Note that the Profile Pane is fixed to the left of the webpage, so it scrolls with the page scroll bar.

When you're done, please enter the post samples shown in Posts.txt. Please mind the **current user** as well as the **Date & Time** values (some dates are also in 2020 not 2021). Your final Timeline should look similar to Figure 11 below. The image is one long image cut into two, with the left part as the top and the right part as the bottom.

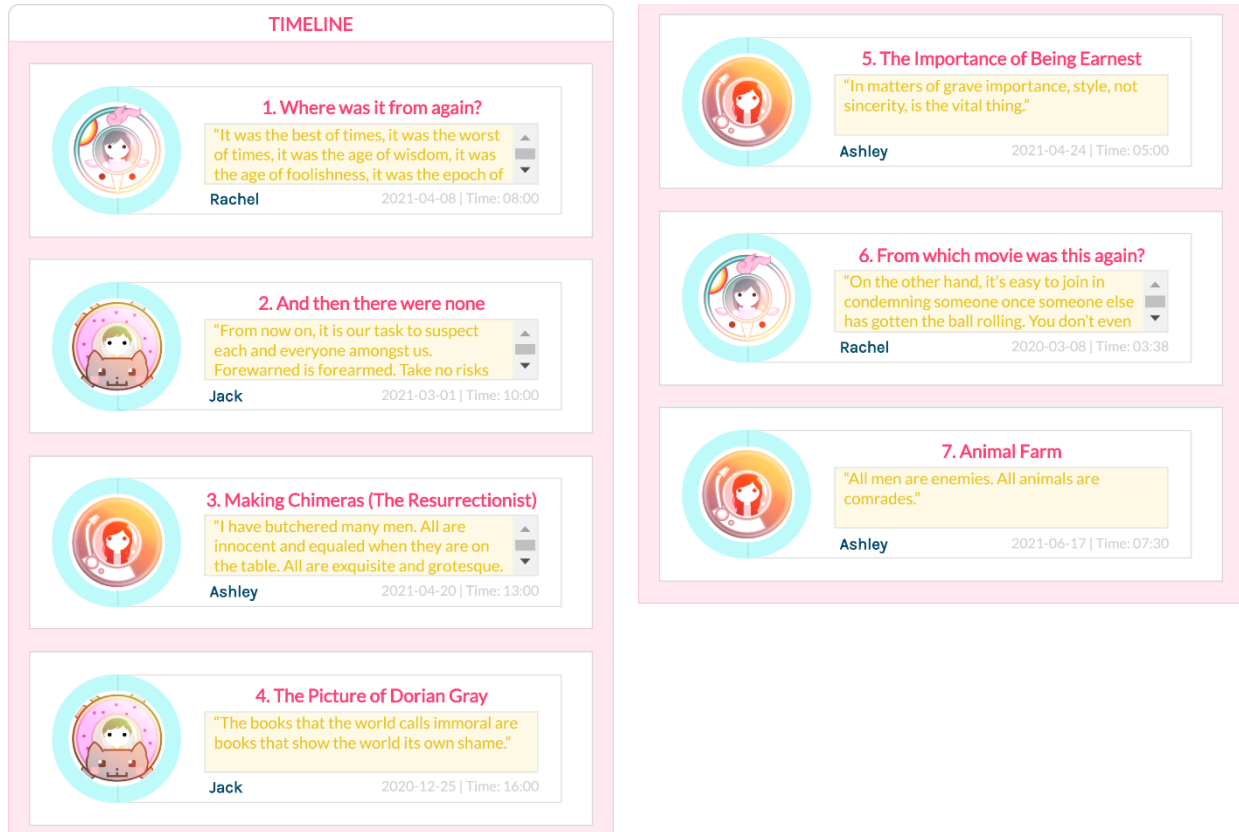


Figure 11. Full Timeline with Sample Posts

Breakdown of Grades

This practice activity is graded, but the grade received does not count towards your final grade.

NOTE! The score for Sort by Post Date is written **above** the Total score of 45.

Content	Score
Working Create Post	5
Create Post validation	5
Create Post reset	5
Correct Timeline display	10
Working Sort by Post Order	5
Working Sort by Post Date	5
Working post Filter	10
Working To Top button	5
TOTAL	45

Bonus Items

You may only get bonus items once you're done with the required specs (sans Sort by Post date).

Implement the Messenger Feature found on the right side of the page (shown in Figure 12). The middle section is the Message Body and the bottom section is the Message Controls. A user should be able to type a message in the Message Controls textarea. To send a message, the user should click on the Message Controls send button.

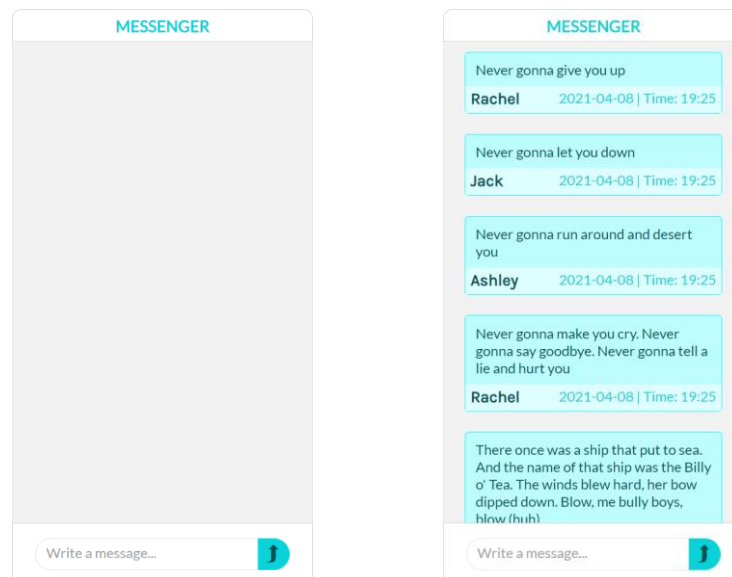


Figure 12. (Left) Empty Messenger, (Right) Messenger with Sample Messages

There is no error message if the user clicks the Send Message button when the Message Text Area is empty. However, **no message** should be sent to the Message Body.

The Message Body will contain all posted messages. If the posted messages are past the Message Body height, it should scroll. The scrolling feature is already given. It should be hidden but functional. If Message Post styling is applied properly, it should scroll when you place your cursor on the Message Body pane and use the mouse wheel (given that enough messages are present to be more than the Message Body height).

A Message Post should look like Figure 13 below. In the example, the message was sent when the **current user** was Ashley. It was sent on 08/04/2021 at 19:25.

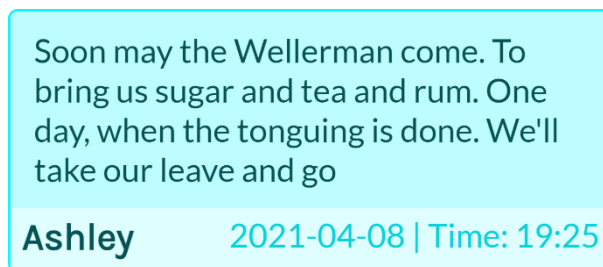


Figure 13. Sample Message

The Message Controls and Message styles are given in the style-messenger.css file. The base index.html should already be calling it. For this bonus, you won't be given the DIV container and class + description table. Try to implement this feature by studying the index file and messenger styles. When you're done, populate your Messenger with the entries in Messages.txt.