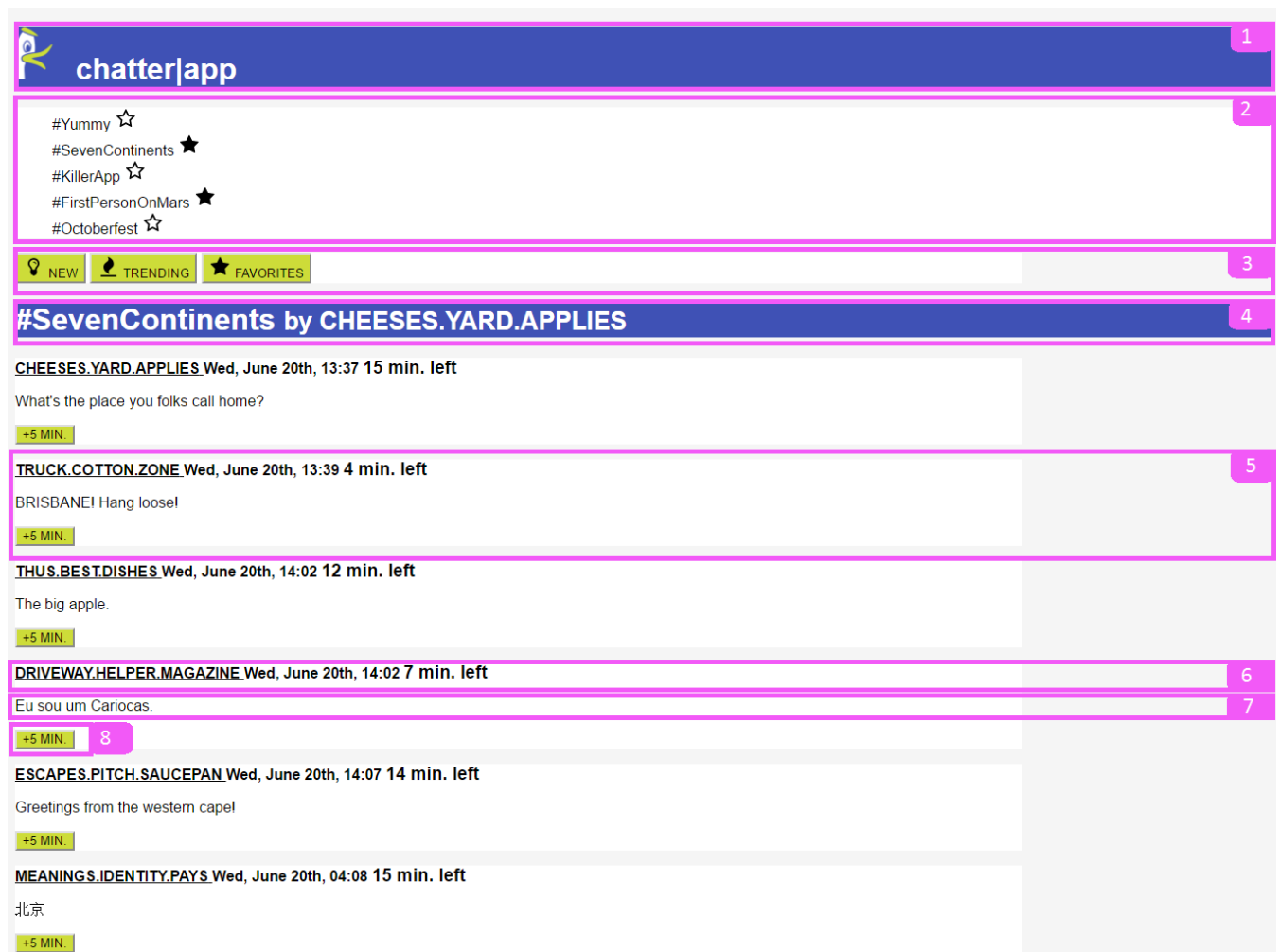# Procedure

Please follow the instructions below and use the provided Github template for this week's assignment. You will need to upload your code after challenge 4.

You can download the instructions as pdf here.

Challenge Goal

After this assignment your app will look similar to the picture below. You will use your newly acquired knowledge in this section to change the previous code and embed media elements. Nesting content in block or inline elements like `<div></div>` or `<span></span>` helps to structure your code and makes it easier to add specific CSS styles to those elements later.

1. Left app bar
2. Channel list
3. Tab bar
4. Right app bar
5. Message container
6. Metadata
7. Message
8. "+5 min." button

Graded Criteria

You will be graded on the following criteria for this challenge:

`#channels` `#moocus` : Channel area is comparable to solution.

- 1 Pt. moocus.png is displayed inside app bar.
- 1 Pt. The channel list has a white background and no bullet points.
- 1 Pt. Each channel has a 'star' or a 'star-o'.

`#tab-bar` : The tab bar is similar to sample solution.

- 1 Pt. Tab bar buttons are wrapped in one container element.
- 1 Pt. Background color of container element is white.
- 1 Pt. An image for the tab button is referenced in the HMTL code prior to the corresponding button text.

`#chat` : Messages (width, link color) look and work like those in the sample solution.

- 1 Pt. Message width is comparable to sample solution.
- 1 Pt. All locations link to w3w website.
- 1 Pt. w3w locations have primary text color.

Your `#syntax` will be graded automatically (1 pt). Overall, 10 Points can be received.

Instructions

Improve the `#channels` view

- `#moocus` , a very talkative duck, will be our app logo. To add moocus' picture to your app you could use a local file. However, when your colleagues will review your code online, references to local files will not work anymore. For this reason we provide an image pack on our server that is accessible from everywhere. You can reference it as follows: `https://ip.lfe.mw.tum.de/sections/anyImage.png` . Just replace `anyImage.png` with the respective file name.

- Insert Moocus (`moocus.png`) into the app bar above the channel area. Do not forget to set a fallback text, if the image cannot be displayed. Resize the image to a height of 56px using inline CSS.

- So far, your channels are simple headlines or paragraphs. You have learned better ways to structure menus or lists than using `<h#></h#>` tags. Rework them to be a `#list` ! Adapt your CSS to keep the list's background white.

- Our list is rather a navigation to different channels than a list. So it should not have bullets. Remove the `#bullets` by using CSS.

- Each channel can be marked as `#favorite` with a little star-button. From our image pack, insert `star.png` for channels you really like and `star-o.png` for those you do not care about so much.

Work on the `#tab-bar`

- At the bottom of the chatter|app will be a tab bar. Wrap the three buttons in a `#container` at the bottom and let it have a white background.

- Insert `#images` from the image pack before each button's text. Use `lightbulb-o.png` for 'new', `fire.png` for 'trending', and `star.png` for 'favorites'.

Improving the `#chat` view

- Group each chat message in a `#container` .

- Make these containers use 80% of the browser's `#width` .
  Tip: You learned about a property, which sets the width of an element, when formatting images.

- Connect all the messages' creator `#locations` (e.g., cheeses.yard.applies) with the w3w service. The map with the corresponding location should be opened in a new tab.

- Change the preset `#link` color to primary text color.

Cleanup

- Clean up your code! Make useful sections and comment your code.

- All links should be working.

- All images (e.g. icons) should have a fallback text.

- Check code & syntax. Using [W3C Validator](#) helps.

Done?

Do not forget to save your code! You will build upon your work in the next implementation challenge.