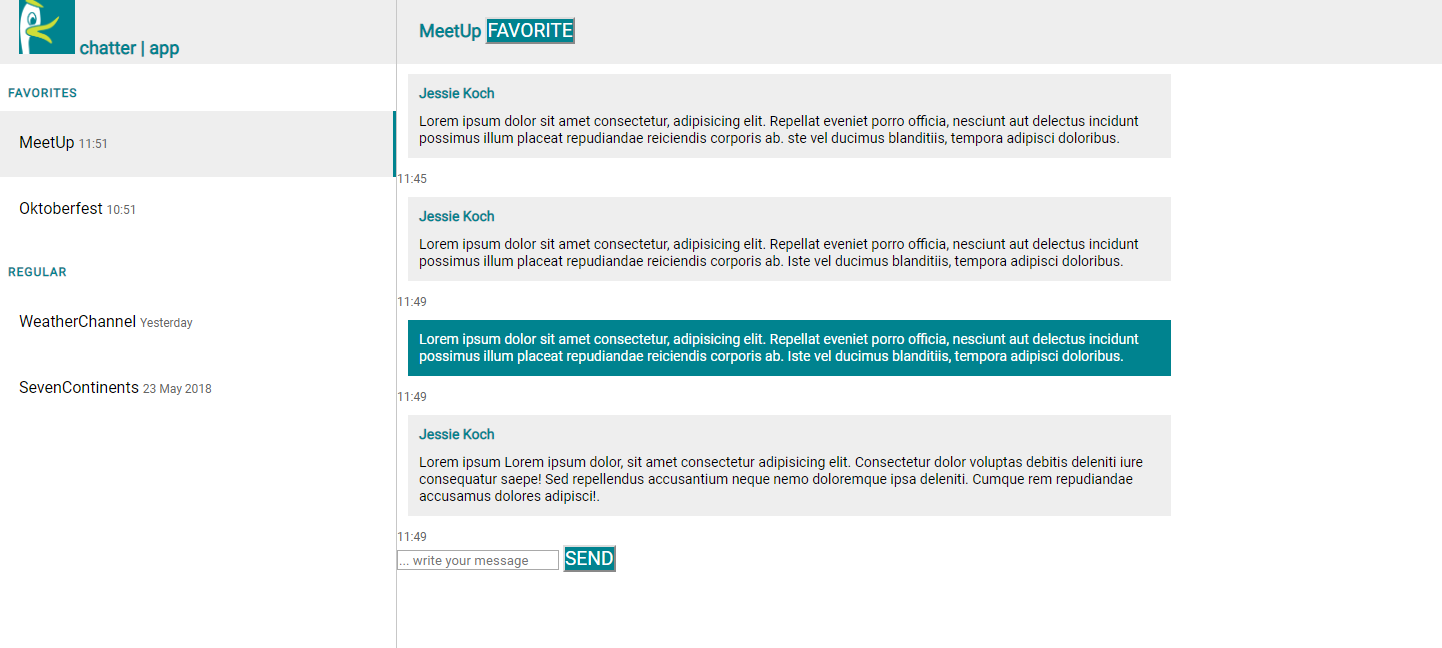
Challenge 2

### 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 the **challenge**.

### Challenge Goal

After this challenge your app will look pretty good already:



### Graded Criteria

You will be graded on the following criteria for this challenge: **21 points**

**#channels**: 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. The channel categories “FAVORITES” and “REGULAR” have the correct appearance.
* 2 Pt. Current channel is highlighted.
* 4 Pt. Channel list is comparable to solution. Compare tile height, font-size, all tiles should be same height.

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

* 3 Pt. Message width is comparable to sample solution.
* 2 Pt. Vertical distance between message containers.
* 2 Pt. Styling of ‘own’ messages.
* 1 Pt. Styling of ‘outgoing’ messages.
* 2 Pt. Space between chat text and borders is set.
* 1 Pt. Input field has a proper placeholder and the text type.

Your **#syntax** will be graded automatically. Overall, 21 Points (Pt.) can be achieved.

### Instructions

#### 1. Outsource your CSS

* Create an empty external style.css file in a CSS subfolder and link to it in your index.html. Divide your CSS file in five different “components” or “sections” by eye-catching CSS comments:
  + Global
  + App bar
  + Channels list
  + Messages
* Remove ugly spacing between all components (document, headlines, …). To do so, select **all elements** and reset their margins and paddings to 0. How do you select all elements? It’s normal to forget, everyone does. Ask Google: ‘select all elements css’ and check out already recommended sources.
* Link another stylesheet to your application to make your fonts look better. Therfore simply use this <link href="https://fonts.googleapis.com/css?family=Roboto" rel="stylesheet">.

#### 2. Improve the channels view

* There is only one channels view in your app so it makes sense to give an **id** to the corresponding <section> tag that surrounds the whole 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/imagetitle.png. Just replace imagetitle.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.
* Group the elements of your headbar in one <div> with the class “headbar”.

***Tip:****Remember that whenever you want to style some parts of you app there are several ways to do it. You can assign****classes****,****ids****or adress them with the corresponding tags etc.*

* So far, your channels are simple headlines or paragraphs. You have learned better ways to structure menus or lists than using <h></h> or <strong></strong> tags.

***Tip:****Remember there is something like <ul> and <li>.*

* Assign the class “channel-list” to your <ul> elements. They will help with styling your app.
* Within your <li> elements we want the channel name and the timestamp to be displayed in one line. Therefore create two <spans> and give assign them class names.
* Rework your channel 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.

#### 3. Improve the chat view

* Assign an id to the chat section.
* Group the headbar in a container. It makes sense to assign the same class as for the channel section, because they are meant to look similar.
* Surround all messages in a container and assign the id “chat-area” and with container we mean <div> elements.
* Group each chat message in a container. It might make sense to use several <div> elements. One can hold the whole message element and the other can hold the message content like username and text. Make sure to assign class names to the containers you use.

***Tip:****Remember that it is possible to assign several classes to one element. Maybe it makes sense to create one message class for all messages and two specific ones for incoming and outgoing messages. As always if you don’t know how to assign several classes to an element use Google.*

* Do the same thing to the input area on the bottom of the chat area. Group it in one div.
* Modify the input field. Set the type of input field to “text”, give it a proper id and a meaningful placeholder.

#### 4. Improve your global styles

* Make sure your body takes 100% of the screens width and height.
* Style the <section> with a position attribute “fixed”.
* Align it to the top and give it the height of 100vh.

***Tip:****Maybe you want to look up what 100vh means.*

* Now we want to style the two sections channel area and chat area.
* Give the channel area a width of 399px and a height of 100vh.
* Insert a thin vertical line to separate between channel area and chat area.
* Assign a left attribute to the message area that makes sure that the section position is 400px from the left edge of the screen.
* For the width you can use the calc function. You have calculate the substraction of 400px from 100% of the width.

***Tip:****Again you might need google to figure out how to do that.*

#### 5. Style your headbar

* Now it’s time to get the app bar look like the sample solution. Adress the background-color, color, height, font-size, padding and line-height to make your headbar look similar to the sample solution. **No worries** it is not important to use the exact values for everything.

***Tip:****You might need to adress several classes, ids or elements to create this appearance.*

#### 6. Beautify the channel area

* We get to the channel area. Start with styling the Favorites and Regular headings.
* Style the channel list <li> elements. Assign the following properties to them:
  + box-sizing: border-box;
  + padding: 22px;
  + height: 66px;
* We want to highlight the currently selected channel. Therfore assign the class “selected” to one of the channels and style it with a proper border to its right and a light grey background.

#### 7. Beautify the chat area

* First of all we have to set the display property of the <h1> element in the message area to **inline** to get the “FAVORITE” button in the same line as the channel name.
* Make sure that one message uses 75% of the width.
* Choose proper margins, paddings and font-sizes for the different message contents.
* Create a class for incoming messages and one for outgoing messages and assign them to the dummy messages that are already displayed.
* Make sure to style the different types of messages to look like the sample solution.

#### 8. Clean up

* 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](https://validator.w3.org/) helps.

#### 9. Save your code

Do not forget to **save** your work, **push** it to Github, and paste the URL in the submission mask. If you have questions concerning Github, consider the discussion board or look it up in the earlier sections.