Assessment 1 - Meme Generator Report

by: Etienne Deneault

Meme Generator - do you meme?

Features

- Meme is dynamilly generated image with the use of the `<canvas>` element.
- Image can be uploaded or used with image URL.
- Generated *meme* can be downloaded or added to a *meme* gallery.
- Images can be deleted from the gallery.
- Submit form is included as per assessment requirements (addBtn), from is cleared when submitted.
- *localStorage* is used to store and track image-data.
- Proxy server is used to allow crossOrigin canvas with url-images.
- Mobile *pseudo-responsive* with the use of CSS *flexbox*.
- Dark-nmode enabled.

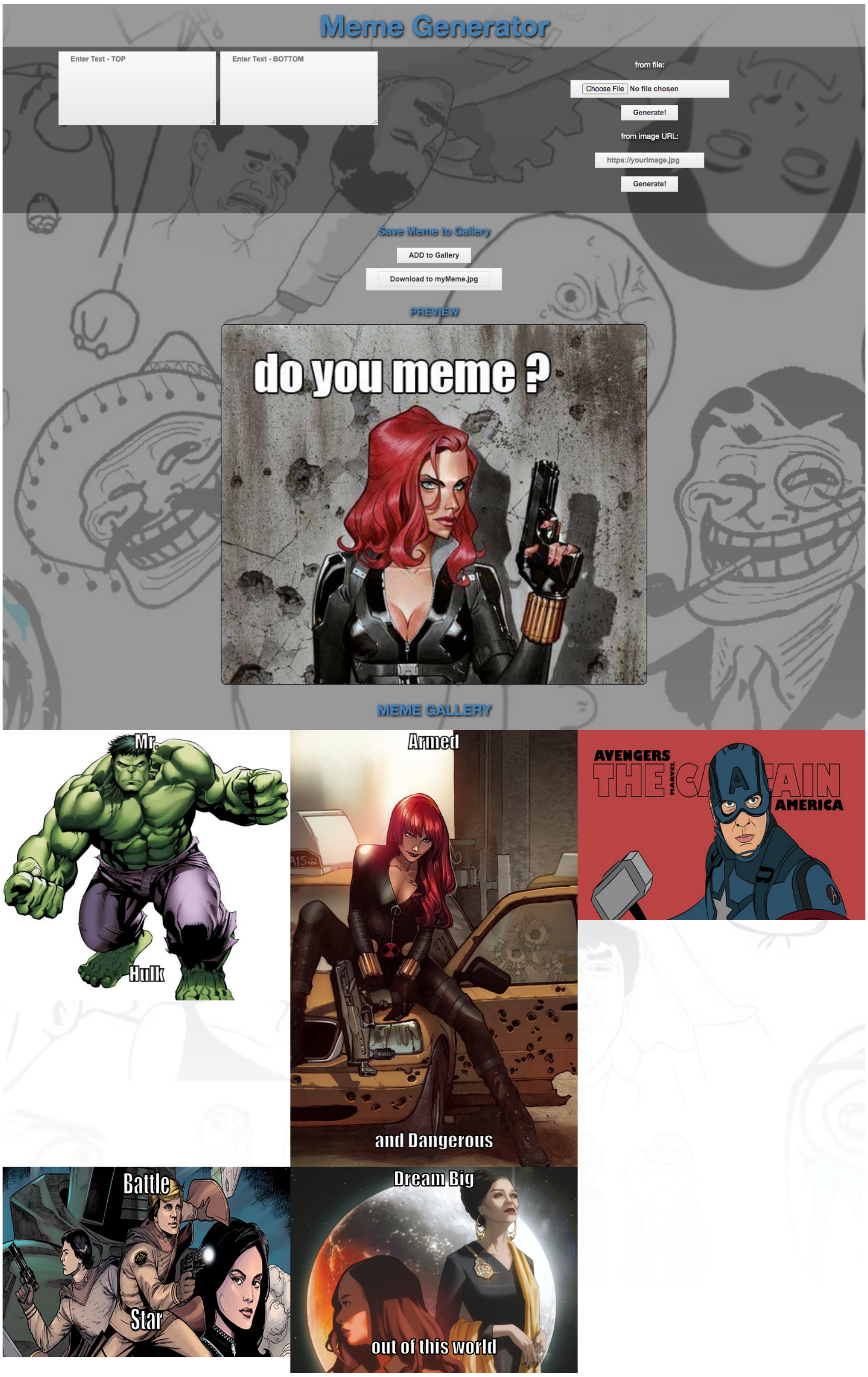
Issues

- The use of localStorage to save image data is not ideal and limits size/number of memes stored in the gallery. It does simulate the request-response cycle with remote servers.
- Untested.
- Responsive behaviour with CSS flexbox is unstable.

Process/Learnings

- Learned a few things in CSS flexbox, how to use `<canvas>` element, and a few things about proxy-servers.
- It took me longer than expected, the logic part of the assignment was quick. The mark-up and css took me much longer than if I had used bootstrap4. My knowledge of HTML and CSS is limited.

Page Rendering



References

- Stackoverlow, MDN, W3 School
- GeekLaunch - Meme Generator Youtube Tutorial (did this tutorial a few months ago as I was teaching myself javascript, still had the code base files I had written at the time, I used it as a code base and modified)
- Wanago.io - how to draw a simple drawing app with canvas + saving and loading images.
- https://cors-anywhere.herokuapp.com (learned a little on how to pass url through a proxy server to bypass crossOrigin issues...not sure how secure this is or if it matters in this specific case use.