**Lernjournal:**

**Reused assets:**

We reused the template from our last project together as that would mean we'd still have a good basic structure and the menu. There may still be some old unnecessary files laying around that don’t belong in this project, in case we forget to remove them, but it’s not a problem since it doesn’t affect the overall product.

**GD Library:**

Since I had already used GD Library for resizing the Images for thumbnails at this point, I only needed to copy the code for recognising the file type and use imagecopy(), to paste the WMS logo on a blank picture I took. For the names I saw a way to implement actual text using GD Library, but as it was very late at the time and the description was vague, I decided to add a picture with the text, which looks just as good as with actual text. (For the record, I would’ve implemented it if it weren’t for the time)

**Gallery and resizing pictures:**

First I created a class called ImageManagement that would get 3 functions: CompressImages(), ShowLinks(), CreateFullPage(). The compressImages() function was built around a basic code I found online, that transformed images into a jpeg. I then adapted that code so it would take all the images in a certain directory, loop that process through every one of them and finally set the quality for it as well, so that could be influenced as well.

After that I worked on the the showLinks() function, which was supposed to generate all the links to the existing images. At that point I realised I wouldn't need a 3rd function, as linking the image itself does a pretty good job of it already. I also put the previous function in the beginning of this one, so that it's insured that there is a thumbnail for every image. At first I just put a link to the image with the thumbnail as an Imagelink and called it a day. But then I decided to implement a carousel James used on another page just to make it prettier. It works and it's functional, but when I last saw it, it had a little issue with the first image being stuck. James mostly worked with Bootstrap so we thought it would be the best if he handled that, while I'm writing the journal.

**Notifications:**

At first there was a little confusion about this one... We weren't sure what was being requested in terms of the list function, but in the end I decided to make 4 function within a class: createFile(), updateFile(), deleteFile() and listFiles(). The first 3 were very simple to make and just required some research for syntax and how the file management in PHP works. Things like openFile() and so on... The last one, I created using scandir(). It scans a specified directory for all it's files and returns an array containing their names. The only real problem I had was one where the 2 first values were always . and .., I even researched it, but found no conclusive answer. But I solved it by skipping the first 2 values and only returning the others. This way I can list all the files and the user knows the names of those files.

The actual page itself consists of 4 parts: The top has a list of all the text files already in there. This allows the user to know which files exist and which don't. Then there are 3 forms: 1 for creating a file in which the user enters the name he wants and what should be written in it. Then there's the updating form, where the user can overwrite what was previously in a file. And there's the deleting form where the user just enters an existing file's name and it gets deleted from existence.

There was no mention, that the text in the files would need to be displayed, so I didn't add that, but that could easily be arranged considering I had a version where something similar was done.

**UPDATE:** After rethinking this last subject I went in and added the functionality of having the file’s text shown at the bottom of the page. I added the showText() function, which does the same regular directory check, like the other functions, but then I looped through the text files, opened them with reading permissions, used fread() to display the text and checked if it was empty. If so, then it will notify the user. Otherwise it will display the text normally. In the end the file is closed to avoid any unnecessary resources to be wasted.

**Webcam:**

For the webcam I searched around until I found a webcam that livestreamed an image I would deem good. However I accidentally stumbled upon a double camera, that just so happened to have a really good example and a really bad example, so I chose that one. They were both set up the same way: 90° of rotation, no user-control, when accessed only a 20s video feed of the camera's last moments is shown until it's updated again. However one of them was far away and mainly captured a hotel and some scenery, while the other flatout invades privacy, as it is very close to a populated street, people's faces are quite visible and it even films a café's open area. I was easily able to see a couple arguing and their faces were clearly visible. I could tell that it was prerecorded tho, as when I clicked on it again it showed me the same clip twice. But after 1 min or so the clip updates to real time.

Due to these arguments I put a link to these cameras on our index page and commented a more in depth explanation in the code.

**WARNING**: We just realised, that these webcams have wroking hours, so depending on what time you’re trying to access them, they might be out of order. We already tested it on a Tuesday at around 10:00 and it also worked on a Saturday at 9:00. We realised it’s out of order on a Saturday at 23:14, so if you’re having issues viewing it, you may need to try later.

**Bootstrap:**

The Bootstrap was very simple to implement. All it required was referencing a few CSS and JS files. Seeing as this was required for every page I decided to put it in the Header.php and Footer.php respectively.

**Gif:**

James went to work on Opentoonz and created a "questionable" GIF that WAS supposed to be some unknown guy snapping his fingers, but eventually evolved into a \***cough\*** \***cough\*** darker theme... Anyway the GIF works and James strapped it on nicely in the index page. Later when I made the gallery, I decided to use his GIF for it as well, because not only did it have a transparent background, it was also animated, so I could test whether the conversion was fine or not. It turned out great and in the thumbnail the background just becomes lime green, which is fitting for a meme.

**Video:**

James took care of the video and found a lovely tutorial from miss Duc, the nicest teacher in the world! He safely strapped it into our index page and with that our page became that much friendlier :). Youtube provides very simple support, all that I had to do was copy and paste the code Youtube generated.

**Audio:**

According to James implementing the audio was easy and I believe it, since it took him around 10 min. However he accidentally put in, the wrong audio and we only realised that much, much later. At the moment of writing this, he's currently charged with changing it back or just adding the right audio. Adding audio is the simplest thing. An interesting aspect though is adding lots of different formats, so the Browser has backups incase one is unsupported.

**Creativity:**

Yeah... we already used all of it up in our other projects :). But in all seriousness: We have a background, that consists of stripes with our names, we have a carousel for our pictures, our testing pictures are memes and although it’s not the prettiest thing in the world, it’s what we built using our knowledge and we had fun doing it, which is what matters. (At least I hope James had fun doing it :) )