

Web Programmation Final Project - Korean journey

Emmanuel Vouillon - 23040092

June 2023



1 Introduction

The goal of this project is to create a cooperative gallery which allows users to create their own account and to share their Korean experience by uploading photographs they have taken, alongside metadata to add context.

This project has been developed with two key ideas in mind. On the one hand it was created with the intention of creating a fully functioning and almost professional looking website. On the other hand, the objective was to show, as much as possible, what we have learned during this semester.

2 Development process

2.1 Using wire frames

As specified in the guidelines, I have started this project by creating wire frames of the pages of my website.



Figure 1: Examples of the wire frames created for the website.

2.2 Version control

As specified in the project guidelines, I employed Git alongside GitHub for my project for version control and remote repository hosting. Therefore, all my code is available on my Github repository which can be found here: <https://www.github.com/emmanuelvln/sejong-webprog-final> where we can easily track the progress of my project thanks to Git's commits system.

3 A closer look

3.1 User authentication

In order to let user post their own photographs, a user authentication was implemented, using mainly the PHP language.

3.1.1 Session management

Since this website's objective is to be a public gallery, there are no restrictions on accessing the home page and appreciating the pictures which have been uploaded.

However, if a user is not registered on the site, other pages used to upload or manage images, are locked off using PHP's session system¹. If a user tries to load theses pages, he is automatically redirected to the log-in page. Besides, once a user has logged-in, he can freely wander from page to page, not needing to register again.

3.1.2 Signing-in

The first thing a new user needs to do in order to get access to all the features this website offers is to create a brand new account². Once on the sign-up page, an email address, a nickname and a password are required.

However, before sending anything to the server, a first set of checks are done on the inputs to make sure theses are valid using JavaScript³. Field which were not correctly completed are colored in red to indicate invalid information.

Once the user's inputs are valid, PHP is used to check the uniqueness of the email address and the nickname, in order to avoid duplicate accounts. Finally if everything is in order, the new user is registered in the database and is redirected to the log-in page.

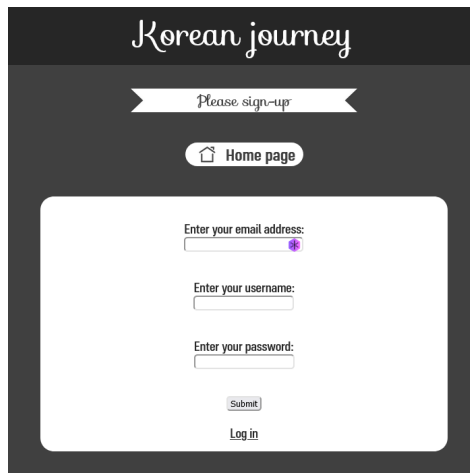


Figure 2: A screenshot of the sign-up page.

¹The function `check_login($con)` which can be found in the `functions.php` file helps automating this process.

²All the code about signing-in can be found in the `signup.php` file.

³Checking for correct email address, non-empty fields and non-numerical nickname is done in the `signup.js` file.

3.1.3 Logging-in

A user who already has an account can log-in using the appropriate page by entering his nickname and his password. PHP is used to check whether these information are correct and if so, the user is redirected to the home page and has now access to all the features of the website.

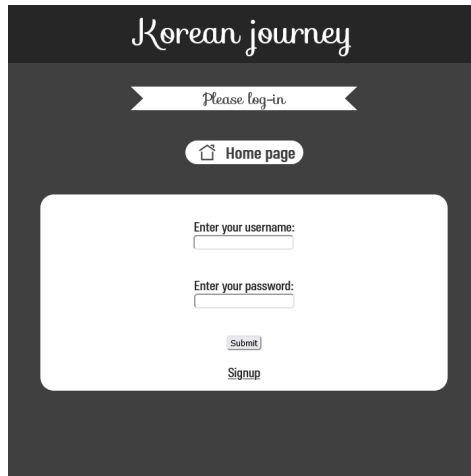


Figure 3: A screenshot of the log-in page.

3.2 Photographs management

3.2.1 Uploading images

In order to upload their photographs, logged-in users must use the upload page which can be accessed from the home page. On this page, the users can select, from their computer, the image they want to upload to the gallery. Besides, the users can enter information about their image such as a title and a location.

When a user submits his image, some checks are done to ensure that the file is valid⁴. If it is, the picture is then copied to the website files and the image is registered in the database using PHP⁵. Besides, if no title or no location is given, the image is automatically marked as untitled or the location is automatically marked as unknown.

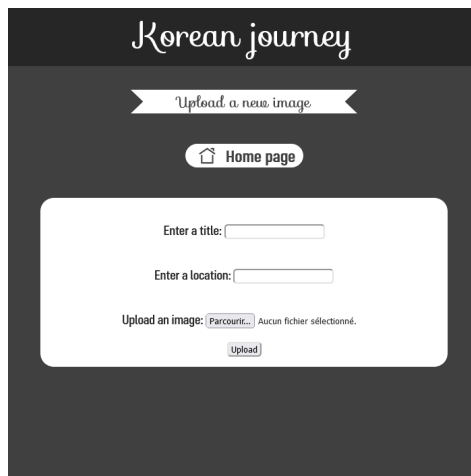


Figure 4: A screenshot of the upload page.

⁴Checks regarding the format and the size of the file are done and can be found in the **transfer.php** file.

⁵This process can be found in the **transfer.php** file.

3.2.2 Managing images

Any images that a user has uploaded can be found in the profile page which is can be accessed from the home page. On this page, which is only accessible by logged-in users, they can see a list of all the pictures they have uploaded and can delete them if needed.

The deletion of an image is done instantly using PHP⁶, it is removed from the database and the copy is deleted from the sources files. Since PHP does not allow responsive design, users on the home page must still reload the page to see the difference.

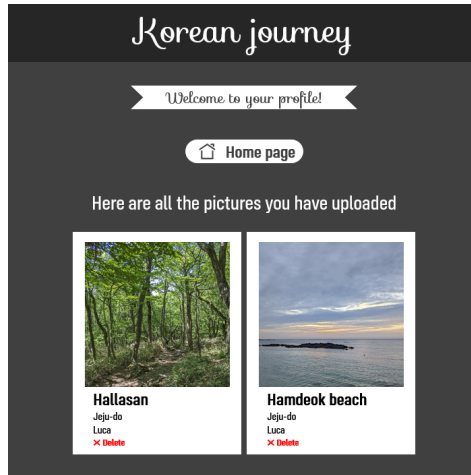


Figure 5: A screenshot of the profile page.

3.2.3 Displaying and sorting images

Initially, the images on the home page are displayed chronologically, from the most recent to the oldest, using PHP to fetch the list of all the recorded images and loading them onto the web page. However, some buttons allow the users to also sort the images chronologically from the oldest to the most recent, alphabetically by title and alphabetically by location⁷.

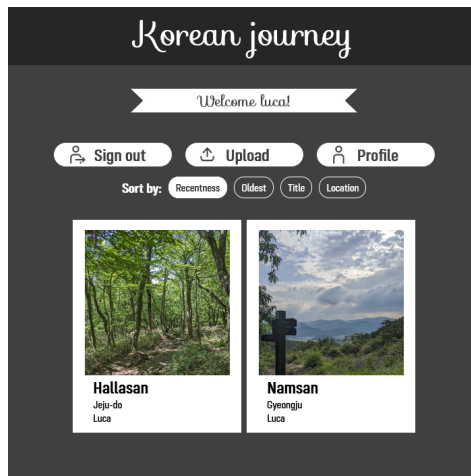


Figure 6: A screenshot of the home page.

⁶The removal of the image is done in the **profile.php** file.

⁷The displaying and sorting system can be found in the **index.php** file.