Web

The Camagru Project

Pedago pedago@42.fr

Summary: The goal of this project is to build a web application a little more complex than the previous ones with a little less means.

Contents

Chapter I

Forewords

History if communication is as ancient as the history of Humanity and Man has managed to evolve along centuries throughout incredible revolutions.

In 1794, Claude Chappe tackled the issue of long distance communication which was actually limited to horse speed at that time. He set up an ingenious communication system of air telegraph during the French Revolution. Chappe's "tours" were covered by a mobile mast that coud be seen with binoculars from the nearest neighbor tour situated at approximately 10 to 15 km.

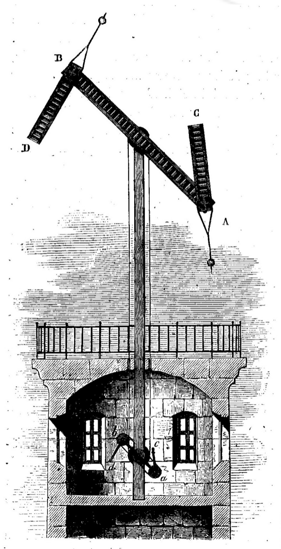


Fig. 19. — Télégraphe de Chappe.

The line Paris-Lille was operational since 1794 and allowed for instance to transmit messages between the two cities with, for instance, a letter being transmitted in 9 minutes throughout 15 tours. Transmission time depended of its length.

In 1844, 534 tours were dispersed on the french territory relying more than 5000km between the most important cities.

The big inconvenience of the system was that it could not be functional during the night due to bad visibility and that it required a lot of operators (around 2 every 15 km).

Fortunately, we are in the 21st century.

Chapter II

Introduction

This web project is challenging you to create a small web application allowing to make basic video editing using your webcam and some predefined images.

`Of course, these images will need to have an alpha channel, otherwise your superposition would not have the expected effect !`

We will, for instance, picture the fatal moment of a intergalactical chat launch, here's the evidence:

A user of your web app will need to be able to select an image in a list of superposable images (for instance a picture frame, or other "we don't wanna know what you are using this for" objects), take a picture with his/her webcam and admire the result that should be mixing both pictures.

All captured images should be public, likeables and commentable.

Chapter III

General instructions

When you say top site you say top language. You will be forced to use PHP to create this project.

You are not authorized to use any framework, micro-framework, librairies or anything from the outside world (except for fonts), neither for the client not for the server side. So this means no Bootstrap, no jQuery, no Symfony, etc... Only the PHP installed extensions (GD and SGBD drivers, among others) and the native Javascript APIs of your browser are allowed.

You will have to use the abstraction interface PDO[1] to accessyour database and define the error mode[2] on PDO::ERRMODE_EXCEPTION.

You can use the web server of your choice, either Apache, Nginx or even built-in web server[3].

Your web application should be at list be compatible with Firefox (>= 41) and Chrome (>= 46).

Your website should have a decent page layout (meaning at least a header, a main section and a footer), able to display on mobile devices and have an adapted layout on small resolutions.

All your forms should have correct validations and the whole site should be secured. This point is COMPULSORY and shall be verified when the project is evaluated. To have an idea, here are some stuff that is NOT considered as SECURE:

- Store plain (unencrypted) passwords in the database.
- Offer the ability to inject HTML ou “user” JavaScript in badly protected variables.
- Offer the ability to upload undesired content on the server.
- Offer the possibility of altering an SQL query.

[1] http://php.net/manual/en/book.pdo.php

[2] http://php.net/manual/en/pdo.error-handling.php

[3] http://php.net/manual/en/features.commandline.webserver.php

Chapter IV

Requirements

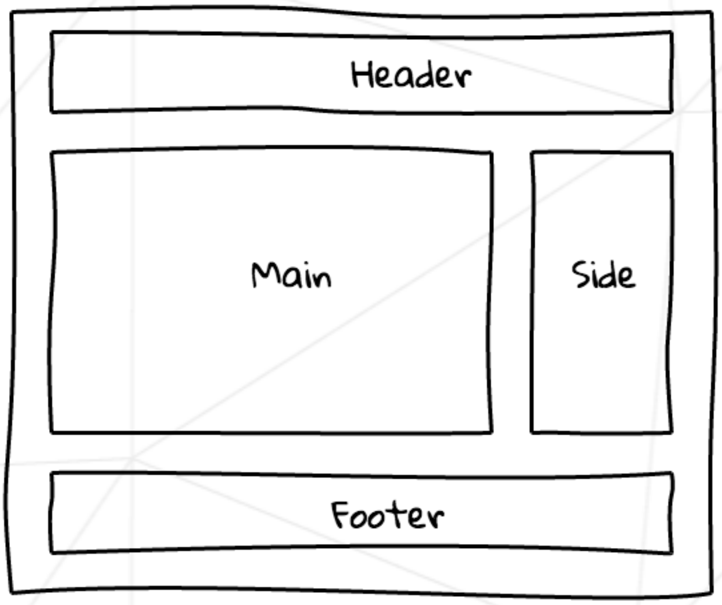
You will develop a web application. Even if this is not required, try to structure your application (in MVC for instance). Your application should have the following features:

IV.1 User features

- The application should allow a user to sign up by asking at least for an email, a password with at least a minimum level of complexity. At the end of the sign-up process a confirmation email should ve sent to the user in order to validate the sign-up process.
- The user should then be able to connect using his username and his password. This user also should be able to receive an email for resetting his password in case he forgot it.
- The user should be able to disconnect in one click at any time on any page.

IV.2 Editing features

Figure IV.1: Just an idea of layout for the editing page



This part should be accessible only to users that are authentified/connected and gently reject all other users that attempt to access it without being successfully logged in.

This page should contain 2 sections:

- A main section containing the preview of the user's webcam, the list of superposable images and a button allowing to capture a picture.
- A side section displaying thumbnails of all previous pictures taken.

Your page layout should normally look like in Figure IV.1.

- Superposable images must be selectable and the button allowing to take the picture should be inactive (not clickable) as long as no superposable image has been selected.
- The creation of the final image (so among others the superposing of the two images) must be done on the server side, in PHP.
- Because not everyone has a webcam, you should allow the upload of a user image instead of capturing one with the webcam.

- The user should be able to delete his edited images, but only his, not other users' creations.

IV.3 Gallery features

- This part is to be public and must display all the images edited by all the users, ordered by date of creation. It should also allow (only) a connected user to like them and/or comment them.

- When an image receives a new comment, the author of the image should be notified by email.

- The list of images must be presented in successive pages (i.e. X images by page).

Chapter V

Bonus part

If the required part is done entirely and perfectly, you can add any bonus you wish; They will be evaluated by your reviewers. You should however still respect the requirements in the bonus parts (i.e. image processing should be done on server side).

If you lack inspiration, here are some leads:

- "AJAXify" exchanges with the server.
- Propose a live preview of the edited result, directly on the webcam preview. We should note that this is much easier than it looks.
- Do an infinite pagination of the gallery part of the site.
- Offer the possibility to a user to share his images on social networks.
- Render an animated GIF.

Chapter VI

Assessment and peer-review

The following requirements are evaluated in the scale. Be very attentive and consider them carefully as they will be sanctioned by a 0 (zero) if not respected.

VI.1 Required files

You project should contain imperatively:

- A `index.php` file, containing the entering point of your site and located at the root of your file hierarchy.

- A `config/database.php` file, containing your database configuration, that will be instanced via `PDO` in the following format:

```
<?php
    $DB_DSN = ...;
    $DB_USER = ...;
    $DB_PASSWORD = ...;
?>
```

  `DSN` (Data Source Name) contains required information needed to connect to the database, for instance `'mysql:dbname=testdb;host=127.0.0.1'`.

  Generally, a DSN is composed of the PDO driver name, followed by a specific syntax for that driver. For more details take a look at the PDO doc of each driver[1].

- A `config/setup.php` file, capable of creating or re-creating the database schema, by using the info contained in the file `config/database.php`.

[1] For more info see the documentation of the PDO constructor

VI.2 Disqualification issues

- Your code should not produce any error, warning or notice neither on the server nor on the client side in the browser console (due to absence of HTTPS, errorss related to getUserMedia() on the web console will be tolerated).

- Use of a framework, a micro-framework, a librairy, etc... is strictly forbidden, and you must use `PDO`.

- By the way, anything that is not specifically allowed means it is forbidden.

- To conclude, the smallest security flaw will mean you get a 0. You must at least handle security issues specified in the general requirements, specifically not store unencrypted passwords, be protected against SQL injections, and have a validation of all input and upload forms.

You can ask your questions on the intra forum, Slack, etc...

To your coding !