

Fiche de Suivi
Installation de services réseau
(SAÉ S2.03)

public
BUT 1 - INFO

FICHE DE SUIVI

Cette fiche est à rendre (à inclure dans votre rapport de projet) à la fin de la séance de projet encadrée. Elle doit être remplie au fur et à mesure des séances encadrées pour refléter la composition et l'état de présence des membres de votre équipe. Une version .doc est donnée sur Moodle afin de faciliter son édition électronique

Consignes importantes

Voici quelques consignes à lire attentivement :

1. Le travail de cette SAÉ sera évalué sur la base de votre rapport de projet à rendre (sur Moodle lors de la toute dernière séance de projet (qu'elle soit encadrée ou en autonomie). La date limite est donc soumise à l'emploi du temps et ne pourra être reportée. Pour l'équité entre groupes, le volume de travail prise en considération est les heures de TP et Projet. Ceci n'empêche pas de travailler en dehors de ces séances mais ce n'est pas obligatoire.
2. La remise de votre rapport de projet (un seul fichier PDF comportant tout votre travail de projet y compris la recherche bibliographique, les configurations, captures d'écrans, et tests techniques, etc.) s'effectue à la dernière séance réglementaire dédié à votre SAÉ.
3. Habitez-vous à bien prendre note d'une manière progressive non seulement pour les intégrer dans votre rapport de projet mais aussi pour reconfigurer votre machine virtuelle de A à Z si vous devez refaire la configuration depuis une machine virtuelle vierge.
4. Le rapport du projet (fichier PDF) doit inclure la présente fiche de suivi fournie en début du projet qui doit être remplie (état de présence des membres au fil des séances, encadrés "tâches effectuées de chaque séance" et "tâches prévisionnelles pour la séances suivante"
5. Aucun autre mode de remise ne sera accepté, aucun retard ne sera toléré. La date de l'upload/téléversement de votre fichier fait foi.
6. Assurez-vous de bien uploader/téléverser votre compte rendu dans la section de votre groupe TP (et non pas dans une section qui concerne un autre groupe)
7. Assurez-vous de noter les noms de familles de chaque membre de chaque groupe d'étudiants dans le nom du fichier téléchargé et dans la page de garde de votre rapport de projet. Attention, tous les membres sont responsables pour s'assurer que le groupe a uploadé son rapport une seule fois sur Moodle.
8. Les rapports de projets (contenu et forme) seront soumis à un traitement anti-plagiat en utilisant les outils automatisés de l'université
9. Le non-respect de ces consignes peut donner lieu à des pénalisations lors de votre évaluation de SAÉ.

Déroulement et timing

Le timing accordé pour cette mission est de 2h de TD, 2h de TP, 7h de projet. Ces séances sont encadrées par un enseignant mais vous avez également 8h de projet en autonomie pour mener à bien la réalisation du projet.

La première séance de TD sera consacrée à

1. L'explication du cahier de charge avec quelques notions techniques
2. La mise en place de votre propre organisation pour mener à bien votre projet.

La première séance de TP sera consacrée à la familiarisation de votre environnement pratique (avec la machine virtuelle consacrée à la SAÉ) et le test des notions du projet expliquées en TD. Les séances projets suivantes seront consacrées à la mise en place du cahier de charge jusqu'à la fin.

Rentabilisez donc bien votre temps car ça va passer très vite ! N'hésitez pas à inclure dans votre rapport de projet un diagramme de Gantt pour gérer votre projet et visualiser dans le temps les diverses tâches associées et leurs répartitions entre les membres du projet. L'idéal est d'inclure 2 versions : une version du diagramme prévisionnel (au début du projet), et une version finale (affinée par le vrai déroulement du projet). N'oubliez pas d'ordonner vos tâches par priorité et bien répartir le travail entre membres. La rédaction du rapport du projet en fait partie et ne doit pas être négligée.

Composition de votre groupe

Nom et prénom -> groupe 2	Numéro affecté	Demi Groupe TP
Berg Doryan	1	A2
Brouard Ashley	2	A2
Jégo-Dufau Manech	3	A2

Nom du Chef du projet : Doryan Berg

Nom du Chef-adjoint : Ashley Brouard

État de Présence :

#Etudiant	Séance 1	Séance 2	Séance 3	Séance 4	Séance 5
1	Oui / Non				
2	Oui / Non				
3	Oui / Non				

#Etudiant	Séance 6	Séance 7	Séance 8	Séance 9	...
1	Oui / Non				
2	Oui / Non				
3	Oui / Non				

Résumés des avancées de chaque séance TD, TP ou Projet

Séance 1

Résumé des tâches effectuées

Ashley: Diagramme UML, début de BDD

Doryan:Lecture des TP/TD, configuration de la base du site web

Manech: Diagramme de Gantt

Résumé des tâches identifiées pour la séance suivante

création du site

fin installation apache

Séance 2

Résumé des tâches effectuées

fin installation apache

début de création du site (nav)

Résumé des tâches identifiées pour la séance suivante

Continuer HTML/CSS

Continuer TP, réglé probleme apache2

Séance 3

Résumé des tâches effectuées

Perte de temps dû à une modification de la VM du pc utilisé par un utilisateur précédent

Solution trouvé pour créer un dossier private

commencement module PHP

Continuation page de garde, création d'une table pour la recherche de quizz et une autre pour les filtres de recherche en HTML CSS

Résumé des tâches identifiées pour la séance suivante

Installation du serveur : Continuation du module PHP

HTML CSS : Finir les tableau et les redimensionner, faire un footer, et si il y a le temps page 404

Séance 4

Résumé des tâches effectuées

Finalisation du module apache (sans problèmes contrairement aux fois précédentes) et commencement du module php

Création de la page de quizz, adaptations CSS concernant cette dernière.

Ajout de variables CSS

Modification de la base de données

Résumé des tâches identifiées pour la séance suivante

Séance 5

Résumé des tâches effectuées

Impossibilité de travailler, session corrompue. En attente du CRI.

Résumé des tâches identifiées pour la séance suivante

Séance 6

Résumé des tâches effectuées

Session récupérée, avancée sur les questions demandées du PHP.

Page 404 et login

Résumé des tâches identifiées pour la séance suivante

Finir le TP PHP

Séance 7

Résumé des tâches effectuées

VMware ne fonctionne pas donc changement de salle,
VMware ne se lance pas, changement de session.

Nous avons fini la partie PHP du TP, maintenant on peut inclure du PHP dans le code HTML.

Résumé des tâches identifiées pour la séance suivante

Passer à la partie PHP + MySQL

Séance 8

Résumé des tâches effectuées

Paramétrage du serveur SQL et de différentes fonctionnalités par PHP.

Résumé des tâches identifiées pour la séance suivante

Finalisé le compte rendu et le Gantt.

Séance 9

Résumé des tâches effectuées

Ajout de fonctionnalité PHP depuis le site web

Commande utile :

- dpkg -i <package> -> installer un package
- systemctl status apache2 -> vérifie le statut d'exécution du serveur
- apache2 -l -> liste les modules d'apache2
- apache2 -M -> liste les modules aussi dynamique d'apache2

Diagramme de Gantt prévisionnel:

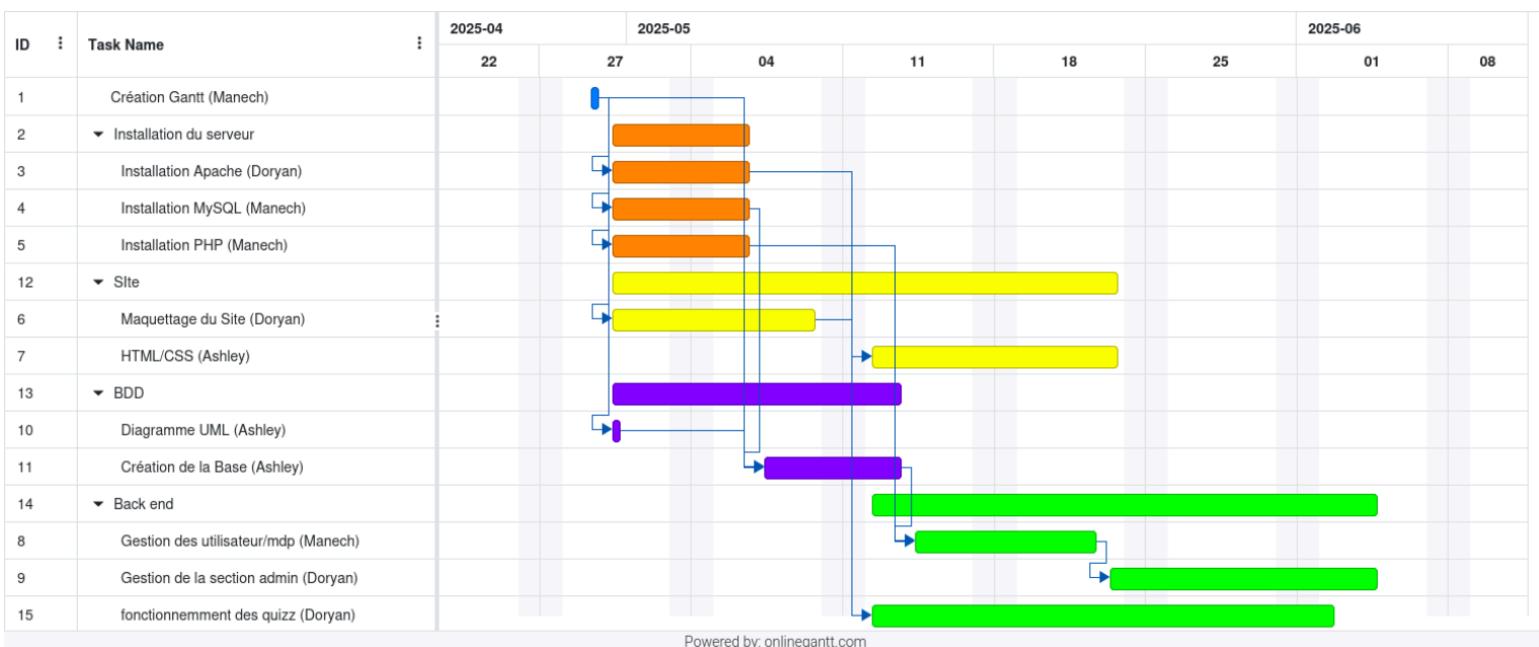
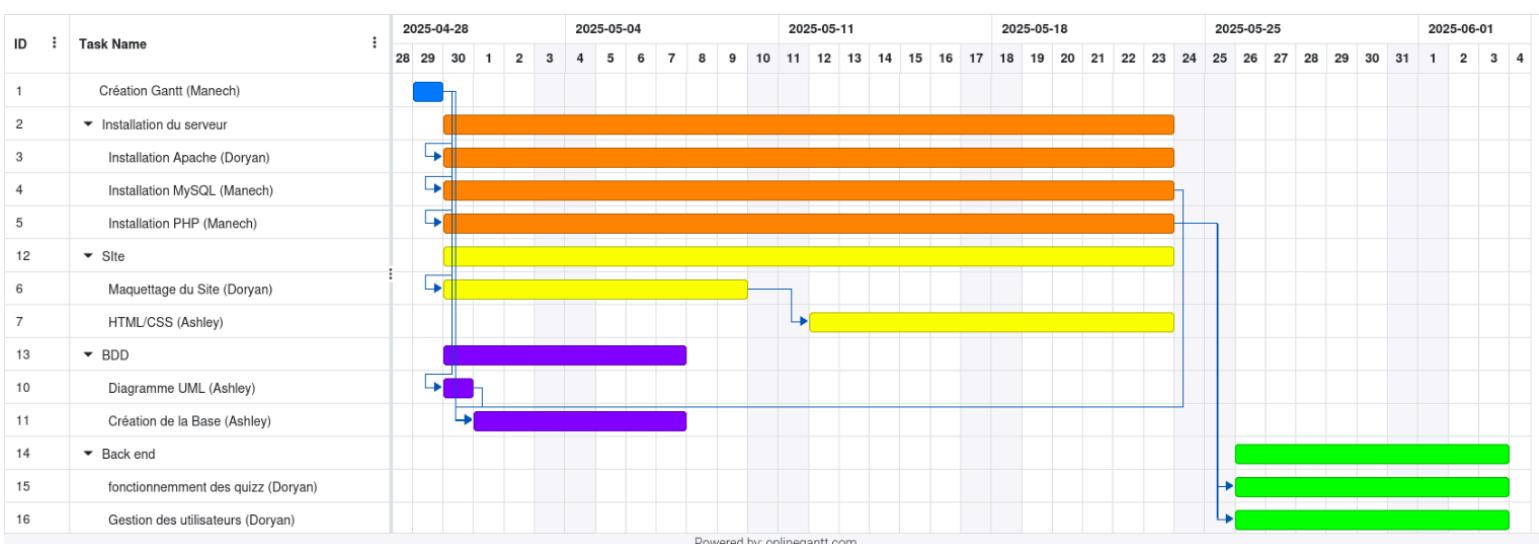


Diagramme de Gantt final:



Module Apache2

2) Après l'installation d'Apache2 sur la Machine Virtuelle, on peut voir tous les détails importants du système Apache2 à l'aide de : "systemctl status apache2"

```
root@SAE:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-04-30 11:15:43 CEST; 4min 21s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 518 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 634 (apache2)
      Tasks: 6 (limit: 2285)
        Memory: 21.4M
          CPU: 103ms
        CGroup: /system.slice/apache2.service
                  ├─634 /usr/sbin/apache2 -k start
                  ├─640 /usr/sbin/apache2 -k start
                  ├─641 /usr/sbin/apache2 -k start
                  ├─642 /usr/sbin/apache2 -k start
                  ├─643 /usr/sbin/apache2 -k start
                  ├─644 /usr/sbin/apache2 -k start

avril 30 11:15:43 SAE systemd[1]: Starting The Apache HTTP Server...
avril 30 11:15:43 SAE systemd[1]: Started The Apache HTTP Server.
```

3) Le dossier root/ représente la directive "DocumentRoot", on peut le découvrir grâce à la commande "pwd DocumentRoot". DocumentRoot est désigné par le serveur Web pour rechercher du contenu web.

```
root@SAE:~# pwd DocumentRoot
/root
```

4) Grâce à la commande "type -a apache2", on trouve le chemin absolu des exécutables d'Apache2.

```
root@SAE:~# type -a apache2
apache2 est /usr/sbin/apache2
apache2 est /sbin/apache2
```

5) La version exacte d'Apache2 est 2.4.53. Connaître la version permet de s'assurer qu'on utilise une version sécurisée, mais aussi que certains modules sont compatibles qu'avec certaines versions, donc cela assure aussi la compatibilité des modules.

```
root@SAE:~# apache2 -v
Server version: Apache/2.4.53 (Debian)
Server built:   2022-03-14T16:28:35
```

6) Voici les modules installés et compilés du noyau :

```
root@SAE:~# apache2 -l
Compiled in modules:
  core.c
  mod_so.c
  mod_watchdog.c
  http_core.c
  mod_log_config.c
  mod_logio.c
  mod_version.c
  mod_unixd.c
```

7) Je pense que le module "mod_log_config.c" sert à la configuration des fichiers de Logs, permettant la personnalisation des résultats dans les fichiers Log.

8) La commande "apache2 -M" permet lister absolument tous les modules :

```
root@SAE:/root# apache2 -M
Loaded Modules:
  core_module (static)
  so_module (static)
  watchdog_module (static)
  http_module (static)
  log_config_module (static)
  logio_module (static)
  version_module (static)
  unixd_module (static)
  access_compat_module (shared)
  alias_module (shared)
  auth_basic_module (shared)
  authn_core_module (shared)
```

```
authn_file_module (shared)
authz_core_module (shared)
authz_host_module (shared)
authz_user_module (shared)
autoindex_module (shared)
deflate_module (shared)
dir_module (shared)
env_module (shared)
filter_module (shared)
mime_module (shared)
mpm_prefork_module (shared)
negotiation_module (shared)
php7_module (shared)
reqtimeout_module (shared)
setenvif_module (shared)
status_module (shared)
```

9) La différence entre "apache2 -v" et "apache2 --version" est que le 2eme envoie un message d'erreur "core" ..

```
root@SAE:~# apache2 --version
[Wed Apr 30 11:48:44.591643 2025] [core:warn] [pid 1530] AH00111: Config variable ${APACHE_LOCK_DIR} is not defined
[Wed Apr 30 11:48:44.591711 2025] [core:warn] [pid 1530] AH00111: Config variable ${APACHE_RUN_DIR} is not defined
apache2: Syntax error on line 86 of /etc/apache2/apache2.conf: DefaultRuntimeDir must be a valid directory, absolute or relative to ServerRoot
```

Alors avec la commande ci-dessous, on exécute des fichiers de configuration dans l'environnement actuel. Le "apache2 -version" n'apporte plus de problème.

```
root@SAE:/# source /etc/apache2/envvars
root@SAE:/# apache2 --version
httpd (pid 665) already running
```

On peut aussi tomber sur un "friendly warning" qu'on règle en définissant dans le apache2.conf en ajoutant une ligne "ServerName localhost" :

```
root@SAE:/# apache2 --version
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
```

10) On crée un fichier de mot de passe dans le dossier etc/apache2. Et grâce à la commande "htpasswd -c /etc/apache2/mdp user" (le -c servant à la création), on définit un nom d'utilisateur et un mot de passe lié qui sont stockés dans le fichier de mot de passe.

note : admin / mdp = lannion

```
root@SAE:~# htpasswd -c /etc/apache2/mdp admin
New password:
Re-type new password:
Adding password for user admin
```

On veut aussi cadenasser des dossiers privés par des comptes. Pour cela, on crée un fichier de blocage dans le dossier privé ".htaccess", avec comme contenu :

```
AuthType Basic
AuthName "Accès restreint"
AuthUserFile /etc/apache2/mdp
Require user admin
```

Et on ajoute au fichier de configuration "apache2.conf" :

```
<Directory /var/www/>
    DirectoryIndex Accueil.html
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

<Directory /var/www/private>
    DirectoryIndex admin.html
    Options Indexes FollowSymLinks
    AllowOverride All
</Directory>
```

Module PHP

1) Il faut créer un fichier php pour la gestion interne entre les pages html et la base de données. On crée donc ce fichier appelé `phpinfo.php`.

2) Il est recommandé de créer un dossier secret pour éviter que les utilisateurs lambda puissent accéder au fichier PHP.

3) L'url utilisé pour accéder au fichier php est "`http://localhost/.secret/phpinfo.php`".

4) L'url amène sur les informations relatives au php du serveur. On peut donc voir que php est activé sur le serveur.

PHP Version 7.4.28	
System	Linux SAE 5.10.0-14- amd64 #1 SMP Debian 5.10.113-1 (2022-04-29) x86_64
Build Date	Feb 17 2022 16:17:19
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-finfo.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API20190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no

5) Dans les dossiers "mods-available" et "mods-enabled" d'apache2, on peut voir que les fichiers de configuration et de chargement de php sont installés.

```
root@SAE:~# cd /etc/apache2/mods-available/
root@SAE:/etc/apache2/mods-available# ls | grep php
php7.4.conf
php7.4.load
```

```
root@SAE:~# cd /etc/apache2/mods-enabled/
root@SAE:/etc/apache2/mods-enabled# ls | grep php
php7.4.conf
php7.4.load
```

Et dans le dossier "mods-enabled", avec la commande "ls -l", on peut voir que les fichiers sont des liens vers les fichiers se trouvant dans le dossiers "mods-available".

```
root@SAE:/etc/apache2/mods-enabled# ls -l php7.4.load
lrwxrwxrwx 1 root root 29  4 mai  2022 php7.4.load -> ../mods-available/php7.4.load
root@SAE:/etc/apache2/mods-enabled# ls -l php7.4.conf
lrwxrwxrwx 1 root root 29  4 mai  2022 php7.4.conf -> ../mods-available/php7.4.conf
```

```

root@SAE:/etc/apache2/mods-enabled# cat php7.4.conf
<FilesMatch ".+\.ph(ar|p|tml)$">
    SetHandler application/x-httdp-php
</FilesMatch>
<FilesMatch ".+\.phps$">
    SetHandler application/x-httdp-php-source
    # Deny access to raw php sources by default
    # To re-enable it's recommended to enable access to the files
    # only in specific virtual host or directory
    Require all denied
</FilesMatch>
# Deny access to files without filename (e.g. '.php')
<FilesMatch "^\.ph(ar|p|ps|tml)$">
    Require all denied
</FilesMatch>

# Running PHP scripts in user directories is disabled by default
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
<IfModule mod_userdir.c>
    <Directory /home/*/*public_html>
        php_admin_flag engine Off
    </Directory>
</IfModule>

```

6)

La version exacte du module PHP utilisé par notre serveur Web est 7.4.28.
Le dossier de configuration du module PHP est : "/etc/php/7.4/apache2".
Le fichier de configuration de php est : "/etc/php/7.4/apache2/php.ini".

- b) La valeur affichée à l'étiquette "short_open_tag" est Off pour la valeur Local et Master.
- c) On peut modifier la valeur de cette étiquette dans le fichier de configuration php.ini, avec son chemin absolue (/etc/php/7.4/apache2/php.ini)

```

-- 
182-; used regardless of this directive.
183-; Default Value: On
184-; Development Value: Off
185-; Production Value: Off
186-; http://php.net/short-open-tag
187:short_open_tag = Off
188-

```

- d) Oui, il faudrait relancer le serveur Web pour que la nouvelle valeur "short_open_tag" soit prise en compte car le fichier est lu à chaque démarrage du serveur.

7) Le chemin du module php est "/usr/bin/php":

On peut voir avec "-v" la version 7.4.28.

```

root@SAE:~# whereis -b php
php: /usr/bin/php7.4 /usr/bin/php
che
root@SAE:~# /usr/bin/php -v
PHP 7.4.28 (cli) (built: Feb 17 20
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c)
with Zend OPcache v7.4.28, Cop

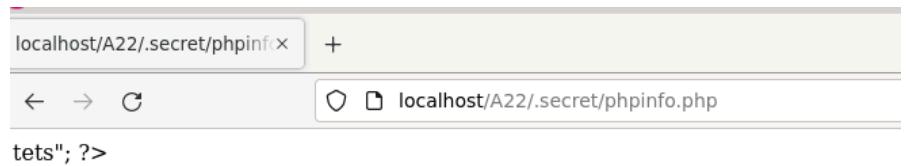
```

8) Grâce à la commande “a2dismod”, on désactive le module php7.4 :

```
root@SAE:~# a2dismod php7.4
Module php7.4 disabled.
To activate the new configuration, you need to run:
systemctl restart apache2
```

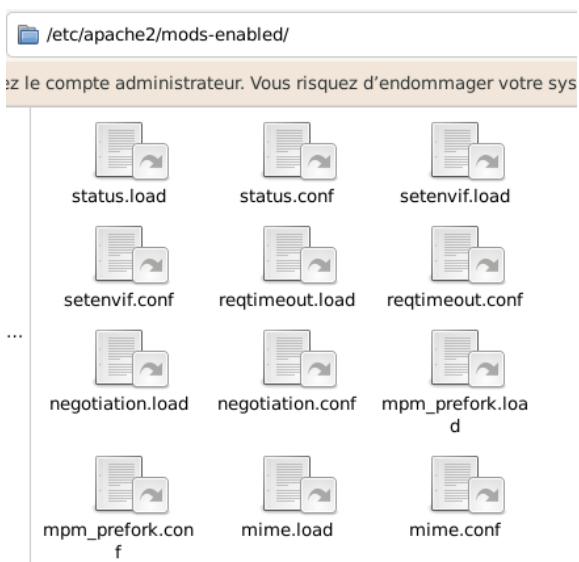
9) En effet, il faut restart Apache2 pour que les modifications soient prises en compte, comme demandé.

10) En désactivant php7.4, on a toujours accès au fichier en eux-mêmes, mais ne fonctionne plus comme des fichier php normaux.



Par exemple avec phpinfo.php, il n'affiche plus les informations (sauf ce morceau de texte ci-dessus qui n'était qu'un ajout personnel pour tester le fichier).

On peut aussi voir dans le dossier des modules activés que “php7.4” n'est pas présent.



11) Même manipulation mais avec “a2enmod” pour activer “php7.4”.

Pour sa réactivation, il est aussi important de relancer apache2.

```
root@SAE:~# a2enmod php7.4
Considering dependency mpm_prefork for php7.4:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php7.4:
Enabling module php7.4.
To activate the new configuration, you need to run:
systemctl restart apache2
root@SAE:~# !systemctl
systemctl restart apache2
```

Le fichier se retrouve dans le dossier “mod-enabled” et le fichier phpinfo.php affiche bien les informations.



otre système.

reqtimeout. reqtimeout. php7.4.load php7.4.conf negotiation .load
load conf .load

deflate.conf autoindex.load autoindex.conf authz_user.load authz_host.load

PHP Version 7.4.28

System	Linux SAE 5.10.0-14-amd64 #1 SMP Debian 5.10.113-1 (2022-04-29) x86_64
Build Date	Feb 17 2022 16:17:19
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-xml.ini, /etc/php/7.4/apache2/conf.d/20-fp-ini.ini, /etc/php/7.4/apache2/conf.d/20-fileno.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-mbstring.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo-mysql.ini, /etc/php/7.4/apache2/conf.d/20-pdo-pgsql.ini, /etc/php/7.4/apache2/conf.d/20-pdo_sqlite.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-zip.ini

12) En ajoutant des lignes PHP, comme un "echo" par exemple, le message s'affiche bien sur la page WEB.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
    <head>
        <meta charset="utf-8" />
        <title>Accueil</title>
        <meta name="description" content="" />
        <meta name="keywords" content="" />
        <link rel="stylesheet" type="text/css" href="css.css" />
    </head>
    <body>
        <?php echo "coucou, je suis un code php dans une page HTML !" ?>
        <a href="/private">Administration</a>
    </body>
</html>
```

13) Il n'y a d'ailleurs pas besoin de restart Apache2 pour que la modification soit acceptée.

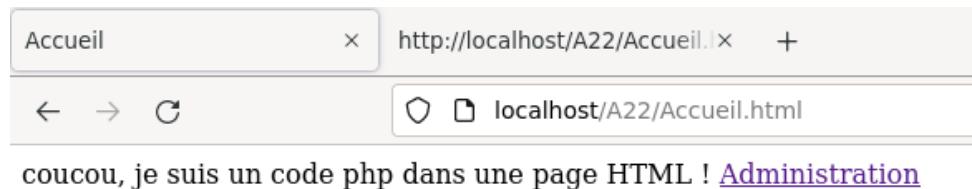
14) On peut donc voir que la commande "echo" ne s'affiche pas.

A screenshot of a web browser window. The address bar at the top shows the URL "http://localhost/A22/Accueil.html". The main content area of the browser displays the text "Administration".

15) Dans le fichier "php7.4.conf", on ajoute à la 1ere ligne dans la commande regex l'élément html pour que php puisse interpréter les fichier d'extension html

```
<FilesMatch "\.(php|par|tml)|html$">
    SetHandler application/x-httpd-php
</FilesMatch>
```

Maintenant le message avec la commande "echo" s'affiche bien.



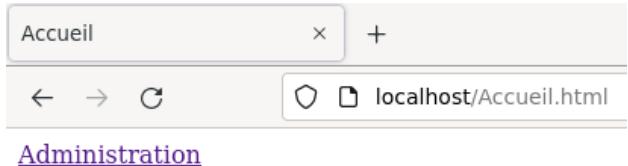
16) En dupliquant un fichier appelé "phpinfo", on peut voir que le fichier est vide. Cela est dû à la modification des extensions lu par Apache2 défini dans la question précédente. L'extension .php est lue mais sans l'extension, le fichier n'affiche rien.



17) Pour qu'Apache2 puisse lire les fichiers sans extension avec du php, on reconfigure le fichier "php7.4.conf" :

```
<FilesMatch ".+(\.|(\.ph(ar|p|tml)|html)$)">
  SetHandler application/x-httpd-php
</FilesMatch>
```

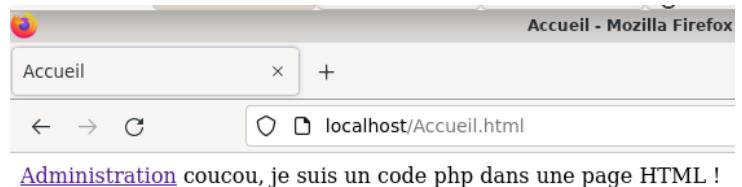
18) En changeant l'insertion de php dans le html avec la syntaxe : <?...?>, L'affichage ne se fait pas :



On modifie le "short_open_tag" en On :

```
, 'production_value. 0',
; http://php.net/short-open-tag
short_open_tag = On
. The number of significant digits
```

Et le message s'affiche bien :



MySQL & PHP

1) Lancement du déploiement de la sécurité pour MySQL, le mot de passe étant celui du root : "lannion", on ne le change pas.

```
root@SAE:/var/www/html# mysql_secure_installation
Securing the MySQL server deployment.

Enter password for user root:
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.
Using existing password for root.

Estimated strength of the password: 25
Change the password for root ? ((Press y|Y for Yes, any other key for No) : N
```

2) On peut voir que le service mysql est actif et tourne continuellement. La date de son premier lancement est notée : "Jeudi 22/05/2025 à 10:35".

```
root@SAE:/etc/apache2# systemctl status mysql.service
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2025-05-22 10:35:59 CEST; 10min ago
    Docs: man:mysqld(8)
          http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 537 ExecStartPre=/usr/share/mysql-8.0/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 667 (mysqld)
     Status: "Server is operational"
        Tasks: 37 (limit: 2285)
       Memory: 462.2M
          CPU: 1.834s
        CGroup: /system.slice/mysql.service
                 └─667 /usr/sbin/mysqld

mai 22 10:35:55 SAE systemd[1]: Starting MySQL Community Server...
mai 22 10:35:59 SAE systemd[1]: Started MySQL Community Server.
```

3) On trouve la version de MySQL est 8.0.29 avec la commande ci-dessous :

```
root@SAE:/etc/apache2# mysql -V
mysql Ver 8.0.29 for Linux on x86_64 (MySQL Community Server - GPL)
```

4) On accède au serveur MySQL avec le login configuré pour le compte root :

```
root@SAE:/etc/apache2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.29 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

- 5) On peut observer les bases de données actuellement sur ce serveur SQL,

```
mysql> SHOW DATABASES ;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sacha          |
| sys            |
+-----+
5 rows in set (0,00 sec)
```

- 6) On crée un fichier PHP qui permet d'afficher le même résultat. Mais un message d'erreur s'affiche présentant que la fonction mysqli_connect() n'existe pas.

```
[Thu May 22 11:00:22.909988 2025] [php7:error] [pid 1819] [client 127.0.0.1:40650] PHP Fatal error: Uncaught Error: Call to undefined function mysqli_connect() in /var/www/html/.secret/phpinfo:3\nStack trace:\n#0 {main}\n    thrown in /var/www/html/.secret/phpinfo on line 3
```

On installe donc le nouveau package "deb11u1_amd64.deb" pour importer la fonction :

```
root@SAE:/usr/local/src# dpkg -i ./php7.4-mysql_7.4.28-1+deb11u1_amd64.deb
Sélection du paquet php7.4-mysql précédemment désélectionné.
(Lecture de la base de données... 110640 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../php7.4-mysql_7.4.28-1+deb11u1_amd64.deb ...
Dépaquetage de php7.4-mysql (7.4.28-1+deb11u1) ...
Paramétrage de php7.4-mysql (7.4.28-1+deb11u1) ...

Creating config file /etc/php/7.4/mods-available/mysqlnd.ini with new version
Creating config file /etc/php/7.4/mods-available/mysqli.ini with new version
Creating config file /etc/php/7.4/mods-available/pdo_mysql.ini with new version
Traitement des actions différées (« triggers ») pour libapache2-mod-php7.4 (7.4.28-1+deb11u1) ...
Traitement des actions différées (« triggers ») pour php7.4-cli (7.4.28-1+deb11u1) ...
root@SAE:/usr/local/src# dpkg -i ./php-mysql_7.4+76_all.deb
Sélection du paquet php-mysql précédemment désélectionné.
(Lecture de la base de données... 110653 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de ./php-mysql_7.4+76_all.deb ...
Dépaquetage de php-mysql (2:7.4+76) ...
Paramétrage de php-mysql (2:7.4+76) ...
```

- 7) On peut créer une nouvelle base de données avec la commande "CREATE DATABASE doryan;", puis vérifier sa création, avec la commande d'affichage par exemple :

- 8) On crée une nouvelle table dans cette base de données avec "CREATE TABLE" :

```
mysql> CREATE DATABASE doryan
      -> ;
Query OK, 1 row affected (0,01 sec)

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| doryan        |
| information_schema |
| mysql          |
| performance_schema |
| sacha          |
| sys            |
+-----+
6 rows in set (0,00 sec)
```

```
mysql> CREATE TABLE etudiant ( id INT PRIMARY KEY NOT NULL,
-> nom VARCHAR(10),
-> dateNaissance DATE,
-> classement INT);
Query OK, 0 rows affected (0,01 sec)
```

```
mysql> SHOW TABLES FROM doryan;
+-----+
| Tables_in_doryan |
+-----+
| etudiant |
+-----+
1 row in set (0,00 sec)
```

On peule cette table avec différents étudiants :

```
mysql> INSERT INTO etudiant VALUES ( 01, 'manche' , '2006-04-01', 20);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO etudiant VALUES ( 02, 'doryan' , '2006-08-04', 19);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO etudiant VALUES ( 03, 'ashley' , '2006-06-24', 1);
Query OK, 1 row affected (0,00 sec)
```

Et on les affichent, cela fonctionne bien :

```
mysql> SELECT * FROM etudiant;
+----+-----+-----+-----+
| id | nom | dateNaissance | classement |
+----+-----+-----+-----+
| 1 | manche | 2006-04-01 | 20 |
| 2 | doryan | 2006-08-04 | 19 |
| 3 | ashley | 2006-06-24 | 1 |
+----+-----+-----+-----+
3 rows in set (0,00 sec)
```

```
mysql> DELETE FROM etudiant WHERE (nom = 'manche');
Query OK, 1 row affected (0,00 sec)

mysql> SELECT * FROM etudiant;
+----+-----+-----+-----+
| id | nom | dateNaissance | classement |
+----+-----+-----+-----+
| 2 | doryan | 2006-08-04 | 19 |
| 3 | ashley | 2006-06-24 | 1 |
+----+-----+-----+-----+
2 rows in set (0,00 sec)
```

9) On supprime les lignes à l'aide de la fonction ci-contre, en précisant une id existante :

```
UPDATE etudiant
SET dateNaissance = '1990-01-01'
WHERE id = (SELECT MIN(id) FROM etudiant GROUP BY id);
```

11)

Voici ci-contre le code PHP pour modifier la date de naissance de la plus petite ID.

```
<?php
// Informations de connexion à la base de données
$servername = "localhost";
$username = "admin"; // Votre nom d'utilisateur MySQL
$password = "lannion"; // Votre mot de passe MySQL
$dbname = "doryan"; // Nom de votre base

// Création de la connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérification de la connexion
if ($conn->connect_error) {
    die("Échec de la connexion : " . $conn->connect_error);
}

// Modification de la date de naissance de la plus petite ID
$sql = "UPDATE etudiant
        SET dateNaissance = '1990-01-01'
        WHERE id = (SELECT MIN(id) FROM etudiant GROUP BY id);";
$result = $conn->query($sql);
?>
```

PHP Pour le site :

Pour l'interaction et l'échange de données entre la partie visible de l'utilisateur, le site web, et la base de données, on utilise du PHP.

Pour commencer, sur la page principal doit apparaître tous les quizz créer avec quelques caractéristiques. On ajoute donc un script PHP qui connecte, puis qui récupère les données:

```
<?php
// Informations de connexion à la base de données
$servername = "localhost";
$username = "root"; // nom d'utilisateur MySQL
$password = "lannion"; // mot de passe MySQL
$dbname = "doryan"; // Nom base

// Création de la connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérification de la connexion
if ($conn->connect_error) {
    die("Échec de la connexion : " . $conn->connect_error);
}

// récup all quizz
$sql = "SELECT id,type,name,creation FROM quizz";
$result = $conn->query($sql);
```

On ajoute les éléments récupérés dans le tableau :

```
<tbody id="resultats">

<?php if ($result->num_rows > 0): ?>
    <?php while ($row = $result->fetch_assoc()): ?>
        <tr>
            <td class="t-cge">Culture Générale</td>
            <td><span><a href="/quiz.html" class="trnc"><?= $row['name'] ?></a></span></td>
            <td><?= $row['creation'] ?></td>
        </tr>
    <?php endwhile; ?>
    <?php else: ?>
        <tr>
            <td colspan="3">Aucun quiz trouvé</td>
        </tr>
    <?php endif; ?>
    <?php $conn->close(); ?>
</tbody>
able>
```

Pour la gestion des utilisateurs, on peut créer un nouveau compte :

```

// Insertion de nouveau utilisateur
function add_user() {
    $sql = "SELECT COUNT(*) FROM usr";
    $nb_user = $conn->query($sql);

    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $username = htmlspecialchars($_POST['username']);
        $password = htmlspecialchars($_POST['password']);

        // Préparation et liaison
        $stmt = $conn->prepare("INSERT INTO usr(id,name,isAdmin,cue) values (?, ?, ?, ?, ?)");
        $nb_user = $nb_user+1;
        $stmt->bind_param("sssss", $nb_user, $username, false, $password);

        // Exécution de la requête
        if ($stmt->execute()) {
            echo "Nouvel utilisateur créé avec succès";
        } else {
            echo "Erreur : " . $stmt->error;
        }

        $stmt->close();
    }
}

```

Et voici le formulaire pour obtenir les informations pour le nouveau compte :

```

<div id="belowNavCtn" class="containerMargin">
    <form class="login-form">
        <label for="username">Nom d'utilisateur</label>
        <input type="text" id="username" name="username" required>

        <label for="password">Mot de passe</label>
        <input type="password" id="password" name="password" required>

        <button onclick="add_user()" type="submit">Login</button>
    </form>
</div>

```

En cliquant sur le bouton “Login”, cela lance la fonction “add_user()”.

Compte rendu - partie web/création du site SAÉ 203

Sommaire:

- 1/ Charte Graphique
- 2/ HTML
 - 2.X - Chaque page individuelle
- 3/ CSS
- 4/ SQL

1/ Charte Graphique

Nous avons besoin d'un front-end pour le site. J'ai donc décidé d'une charte graphique:

Polices utilisées:

Titre: Source Sans 3

Texte: Instrument Sans

Couleurs:

```
/*colours/
--spectralon: □#FFF;
--cream: □#ffffe7;
--vantablack: ■#000;
--primaryc: □#00FF8C;
--secondaryc: □#ff81bc;
--cerulean: □#42c6e7;
--hiLightColour: □#7a86f3;
--shadow: □#0005;

--wrongAnswer: ■#be0000;
--rightAnswer: □#43ff43;

--t-spo: □#ff2d7d;
--t-cge: □#ffbf00;
--t-cmo: □#82ffba;
--t-his: ■#ff0000;
```

Par bloc:

Couleurs principales,

Bonne / mauvaise réponse

Couleurs par thème de quiz
(sport, culture générale, culture moderne,
histoire, géographie, langue, faune et flore,
science, autres)

2/ HTML

Nous avons besoin de ces pages web:

- index.html : la page d'accueil et de navigation de quiz
- quiz.html : la page où les questions sont posées pendant le quiz
- results.html : la page de résultats après le quiz, ainsi que le classement "leaderboard" des différents utilisateurs.ice.s
- login.html : la page de login pour se connecter à un compte existant
- 404.html : la page d'erreur 404 pour indiquer qu'un contenu n'est pas/plus accessible

Tous les textes vus dans les captures d'écran sont des placeholders, des exemples.

Je ne parlerai des règles CSS que dans la partie 3/.

2.1/ index

En premier, j'ai créé un nav, qui sera utilisé sur chaque page du site :

```
<body id="index">
  <nav id="topNav">
    <a class="title shadow" id="logoNav" href=".//">ULTRAQUIZZ</a>
    <div id="searchCtn">
      <input class="searchbar" type="text" placeholder="Rechercher un quiz..."/>
      <button>Go</button>
    </div>
    <a class="login" href=".//login.html"><button>Connection / Créer un Compte</button></a>
  </nav>
```

Ce nav contient un logo, texte "ULTRAQUIZZ" avec ombre colorée, une barre de recherche avec un texte placeholder "Rechercher un quiz..", un bouton Go pour exécuter une recherche, et un bouton pour aller à la page de connexion vers un compte.

En dessous du nav, nous avons notre header qui est dans un container de tout ce qui est en dessous du nav (belowNavCtn, below Nav Container), sauf le footer, qui est gardé individuellement en bas, pour des raisons expliquées dans la partie 3.

```

</nav>
<div id="belowNavCtn" class="containerMargin">
  <header>
    <p>Bienvenue sur Ultraquizz, le site de
      quiz #1 de Lannion Trégor Communauté
      (et ses magnifiques alentours) ! Nous
      avons plein de quiz disponibles pour
      tous, sur une grande variété de thèmes.
      Essayez de conquérir le #1 sur le
      leaderboard des quiz (si vous en êtes
      capables.)<br>
    </p>
  </header>
  <main>
    <div id="searchfilterHolder">
      <div id="searchHolder">
        ...
      </div>
    </div>
  </main>
</div>

```

Nous avons une table de base, nommée “queryResults”, avec les résultats de “recherche”. Par défaut, les résultats affichés sont récents.

Le nom de la recherche est remplaçable facilement par le #searchQuery.

Chaque résultat (quiz) est structuré de façon simple: Un thème/catégorie, un nom, et un nombre de parties totales.

```

<div id="searchHolder">
  <table id="queryResults" class="table-m-set">
    <thead>
      <tr>
        <th colspan="3" class="bigfont">Résultats :
          <span id="searchQuery" class="ignore-m-set bigfont">Récents</span>
        </th>
      </tr>
      <tr>
        <th>Catégorie</th>
        <th>Quiz</th>
        <th>Tentatives</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td class="t-cge">Culture Générale</td>
        <td><span><a href="/quiz.html" class="trnc">text something something greeting</a></span></td>
        <td>1.4K</td>
      </tr>
      <tr>
        <td class="t-lan">Langues</td>
        <td><span><a href="/quiz.html" class="trnc">Devine la lanque</a></span></td>
        ...
      </tr>
    </tbody>
  </table>
</div>

```

Les filtres sont dans cette page:

Nous pouvons choisir 0..* types avec des checkbox pour la recherche. Quand 0 types sont sélectionnés, tous les types de quizzes sont affichés, car une recherche excluant tous les quizzes n'aurait aucun sens.

```

<div id="filterHolder">
    <p class="bigfont no-m-b">Filtres:</p>
    <form action="/filtres.php">
        <div id="filterByTypeHolder">
            <input type="checkbox" id="type1" name="type1" value="Sport">
            <label for="type1"> Sport</label><br>
            <input type="checkbox" id="type2" name="type2" value="Culture Moderne">
            <label for="type2"> Culture Moderne</label><br>
            <input type="checkbox" id="type3" name="type3" value="Culture Générale">
            <label for="type3"> Culture Générale</label><br>
            <input type="checkbox" id="type4" name="type4" value="Langues">
            <label for="type4"> Langues</label><br>
            <input type="checkbox" id="type5" name="type5" value="Géographie">
            <label for="type5"> Géographie</label><br>
            <input type="checkbox" id="type6" name="type6" value="Histoire">
            <label for="type6"> Histoire</label><br>
            <input type="checkbox" id="type7" name="type7" value="FauneetFlore">
            <label for="type7"> Faune & Flore</label><br>
            <input type="checkbox" id="type8" name="type8" value="Sciences">
            <label for="type8"> Sciences</label><br>
            <input type="checkbox" id="type9" name="type9" value="Autres">
            <label for="type9"> Autres</label><br>
        </div>
    </form>

```

Nous avons aussi mis des boutons radio pour indiquer le tri de manière simple:

le choix est soit du plus récent, soit du plus populaire, soit par ordre alphabétique.

Il y a aussi un bouton à la fin pour valider nos choix de filtres et de tri.

```

</div>
<p class="bigfont no-m-b">Trier par:</p>
<div id="radioFilterHolder">
    <input type="radio" id="age1" name="age" value="30" checked="checked">
    <label for="age1">Récents</label><br>
    <input type="radio" id="age2" name="age" value="60">
    <label for="age2">Popularité</label><br>
    <input type="radio" id="age3" name="age" value="100">
    <label for="age3">Ordre Alphabétique</label><br><br>
</div>

<input class="pretend-button" type="submit" value="Filtrer">
</form>

```

Rendu final :

Bienvenue sur Ultraquizz, le site de quiz #1 de Lannion Trégor Communauté (et ses magnifiques alentours) ! Nous avons plein de quiz disponibles pour tous, sur une grande variété de thèmes. Essayez de conquérir le #1 sur le leaderboard des quiz (si vous en êtes capables.)

Résultats : Récents

Catégorie	Quiz	Tentatives
Culture Générale	test of all time	1.4K
Langues	Devine la langue	4.5K
Culture Moderne	Connais-tu le lorc de Denis Vaillant?	932583T
Autres	Top des rappeurs qui se sont fait djoufara	1M
Sciences	Nomme l'élément chimique	1.22M
Sport	Les footballeurs les plus guez il me faut un titre long bonjour bonsoir comment allez vous bien et vous o...	339K
Géographie	Capitale de pays	23K

Filtres:

- Sport
- Culture
- Moderne
- Culture Générale
- Langues
- Géographie
- Histoire
- Faune & Flore
- Sciences
- Autres

Trier par:

- Récents
- Popularité
- Ordre
- Alphabétique

Filtrer

2.2/ quiz

Dans la page de quiz nous avons le même début que la page principale avec le nav. Puis en dessous, nous avons un header, contenant “Quiz: (nom du quiz)”, nom du quiz simplement modifiable dans le back-end avec l’ID #quiz-name.

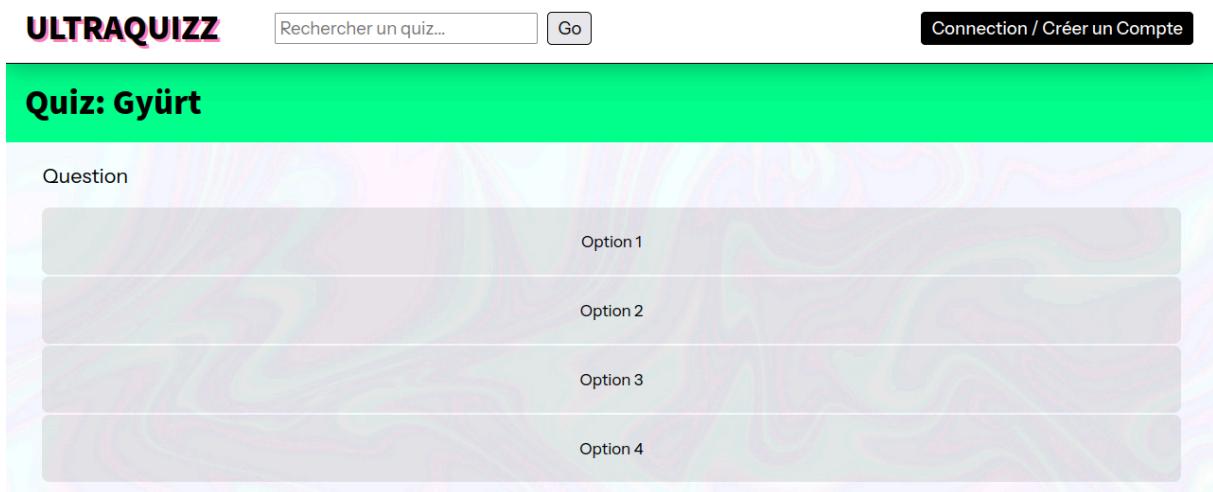
Nous avons ensuite 4 boutons radio, avec option 1, 2, 3, et 4, qui permettent donc de faire des choix dans le quiz. Les choix seront envoyés dès que l’utilisateur.ice choisira un des boutons, il n’y a pas besoin de confirmer sa sélection. Ça permet d’avoir une expérience de quiz plus rapide.

```

<header>
  <h3 id="qznm" class="title gigafont no-m-t">
    Quiz: <span id="quiz-name" class="title gigafont no-m-t">Gyürt</span>
  </h3>
</header>
<div id="belowNavCtn" class="containerMargin">
  <p id="question-name" class="bigfont">Question</p>
  <div class="radio-group">
    <label class="radio-block">
      <input type="radio" name="option" value="1" hidden>
      <div class="custom-box">Option 1</div>
    </label>
    <label class="radio-block">
      <input type="radio" name="option" value="2" hidden>
      <div class="custom-box">Option 2</div>
    </label>
    <label class="radio-block">
      <input type="radio" name="option" value="3" hidden>
      <div class="custom-box">Option 3</div>
    </label>
    <label class="radio-block">
      <input type="radio" name="option" value="4" hidden>
      <div class="custom-box">Option 4</div>
    </label>
  </div>
</div>

```

Rendu final de la page:



The screenshot shows the UltraQuizz quiz interface. At the top, there is a navigation bar with the text "ULTRAQUIZZ", a search bar with the placeholder "Rechercher un quiz...", and a "Go" button. To the right of the search bar is a "Connection / Créer un Compte" button. The main content area has a green header bar with the text "Quiz: Gyürt". Below the header, the word "Question" is displayed. There are four rectangular buttons, each containing the text "Option 1", "Option 2", "Option 3", and "Option 4" respectively, representing the four options for the quiz question.

2.3/ résultats

Dans la page de résultats nous avons le même header et nav que le quiz. Nous ne nous y attarderons pas.

Nous avons un grand affichage pour les résultats de l'utilisateur.ice, comme ci-dessous, avec des id spécifiques pour les valeurs qui pourraient être amenées à être modifiées dans le backend du site.

```
<div id="belowNavCtn" class="containerMargin">
  <p id="question-name" class="bigfont">Résultats :</p>
  <div class="resultsHolder">
    <div id="time" class="bigResultHolder">
      <p class="titleAboveResult">Temps pris:</p>
      <p id="timeTaken" class="bigresult">4:32</p>
    </div>
    <div id="placement" class="bigResultHolder">
      <p class="titleAboveResult">Placement:</p>
      <p id="placementTaken" class="bigresult">23</p>
      <p class="subtitleResult">sur <span id="placementMax">56</span> participant.e.s</p>
    </div>
    <div id="rightsHolder" class="talc rwhl">
      <p>Réponses Correctes:</p>
      <p id="rights">10</p>
    </div>
    <div id="wrongsHolder" class="talc rwhl">
      <p>Réponses Fausses:</p>
      <p id="wrongs">8</p>
    </div>
  </div>
</div>
```

Nous avons ensuite une table “queryResults” comme pour l'accueil, mais avec des informations à afficher différentes.

```
<table id="queryResults" class="table-m-set leaderResults individualize">
  <thead><tr>
    <th>Placement</th>
    <th>Joueur</th>
    <th>Temps</th>
    <th>Bonnes réponses</th>
  </tr></thead>
  <tbody><tr>
    <td>1</td>
    <td><span class="trnc">DoryanBerg</span></td>
    <td>04:05</td>
    <td>40/40</td>
  </tr><tr>
    <td>2</td>
    <td><span class="trnc">AshleyBrouard</span></td>
    <td>04:02</td>
    <td>39/40</td>
  </tr><tr>
    <td>3</td>
    <td><span class="trnc">sdffkjbsdkjfskdfsknfldlsjdlfjsdlkfjsldkfjlsdkflskdflskd1</span></td>
    <td>04:10</td>
    <td>39/40</td>
  </tr>
  </tbody>
</table>
```

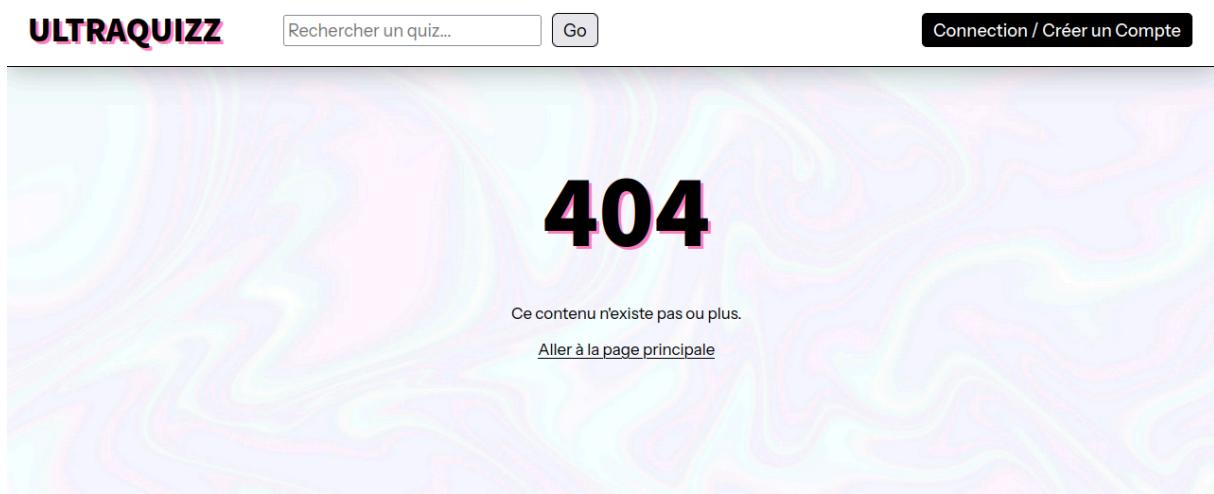
2.4/ Page d'erreur 404

Nous avons une page simple affichant un 404 en large et un message explicatif pour comprendre la signification de l'erreur.

Nous avons aussi ajouté un lien vers la page principale du site pour que l'utilisateur.ice puisse éventuellement retrouver la page voulue, ou naviguer le site.

```
~/.vscode
<body id="error404">
  <nav id="topNav">
    <a class="title shadow" id="logoNav" href="./">ULTRAQUIZZ</a>
    <div id="searchCtn">
      <input class="searchbar" type="text" placeholder="Rechercher un quiz..."/>
      <button>Go</button>
    </div>
    <a class="login" href=".//login.html"><button id="topRightBtnTxt">Connection / Cr  er un Compte</button></a>
  </nav>
  <div id="belowNavCtn" class="containerMargin">
    <h1 class="no-m title shadow megalovafont">
      404
    </h1>
    <p>
      Ce contenu n'existe pas ou plus.
    </p>
    <a href="/index.html" class="ul">
      Aller   la page principale
    </a>
    
  </div>
</body>
tml>
```

Voici l'apparence de la page:



2.5/ page de login

Le nav est le même que les autres pages,
Nous avons un champ pour entrer le login et le mot de passe, ainsi qu'un bouton de confirmation. L'input du mot de passe est mis en mode 'password' pour qu'il cache les caractères entrés pour protéger la sécurité des utilisateur.ices.

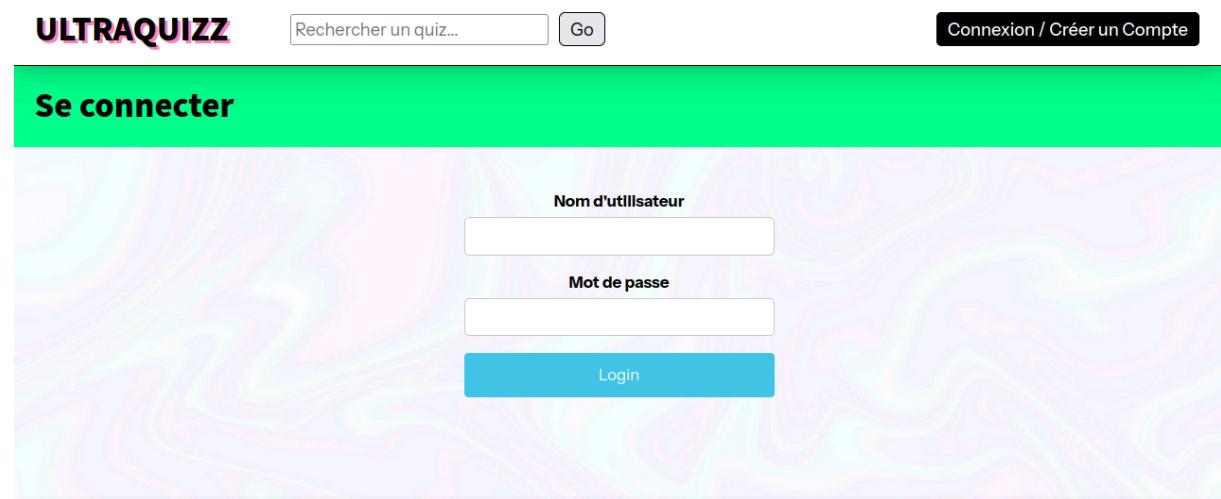
```
<header>
  <h3 id="qznm" class="title gigafont no-m-t">
    Se connecter
  </h3>
</header>
<div id="belowNavCtn" class="containerMargin">
  <form class="login-form">
    <label for="username">Nom d'utilisateur</label>
    <input type="text" id="username" name="username" required>

    <label for="password">Mot de passe</label>
    <input type="password" id="password" name="password" required>

    <button type="submit">Login</button>
  </form>
  <div>

  </div>
</div>
```

Voici un aperçu de la page:



The screenshot shows the Ultraquizz login page. At the top, there is a green header bar with the text "ULTRAQUIZZ" in white. Below the header is a search bar with the placeholder "Rechercher un quiz..." and a "Go" button. To the right of the search bar is a "Connexion / Créer un Compte" button. The main content area has a light blue background with a subtle wavy pattern. It features a green "Se connecter" button. Below the button is a form with two input fields: "Nom d'utilisateur" and "Mot de passe", each with a corresponding label and an empty input box. At the bottom of the form is a blue "Login" button.

3/ Le CSS

Nous commençons le CSS avec les imports de polices et les créations de constantes à réutiliser plus tard dans le code.

```
1  @font-face {
2      font-family: "Title Font";
3      src: url(font/SourceSans3-VariableFont_wght.ttf);
4  }
5  @font-face {
6      font-family: "Regular Font";
7      src: url(font/InstrumentSans-VariableFont_wdth\,wght.ttf);
8  }
9  :root {
10     /*sizes*/
11     --basefont: 1.1rem;
12     --bigfont: 1.4rem;
13     --gigafont: 2.4rem;
14     --btnCornerRadius: 5px;
15
16     /*colours*/
17     --spectralon: #FFF;
18     --cream: #ffffe7;
19     --vantablack: #000;
20     --primaryc: #00FF8C;
21     --secondaryc: #ff81bc;
22     --cerulean: #42c6e7;
23     --hiLightColour: #7a86f3;
24     --shadow: #0005;
25
26     --wrongAnswer: #be0000;
27     --rightAnswer: #43ff43;
28
29     --t-spo: #ff2d7d;
30     --t-cae: #ffhf00;
```

```

    --t-cmo: #82ffba;
--t-his: #ff0000;
--t-geo: #2020de;
--t-lan: #aa40aa;
--t-fef: #54ef16;
--t-sci: #aabbff;
--t-aut: #919191;

--hMarginQuiz: calc((100vh - 23rem) / 8);
}

html {
  font-size: clamp(14px, 1.2vw, 18px);
}

html, body {
  height: 100%;
}

* {
  font-family: "Regular Font";
  font-weight: 400;
  color: var(--vantablack);
  font-size: var(--basefont);
}

```

les t-xxx sont les couleurs des catégories de quiz, qui ont les mêmes noms dans la Base de données. Les autres couleurs sont les couleurs utilisées.

Nous avons une indication d'une taille de police clampée (qui s'adapte selon la taille de l'écran) pour que le site soit plus accessible à tous.tes. Nous utiliserons des tailles de polices en 'rem', une unité se basant sur la taille de police de la racine du fichier, ce qui permettra d'avoir moins de problèmes d'adaptabilité avec diverses tailles d'écran.

Nous faisons ensuite en sorte que la police par défaut de tout élément utilise la police principale, qu'il soit en noir, et que la taille soit universelle.

Nous faisons un set up de base pour les liens et les boutons:
 Les liens ne sont plus soulignés, mais quand on hover par dessus, ils doivent devenir bleus. (en utilisant la constante définie var(--cerulean))
 Nous avons aussi des règles spéciales pour le bouton de login, vu qu'il est de couleur différente.

```

a {
  text-decoration: none;
}

a:not(nav > a):hover {
  color: var(--cerulean);
}

button, .pretend-button {
  border-radius: var(--btnCornerRadius);
  padding: var(--btnCornerRadius);
  padding-left: calc(var(--btnCornerRadius) * 2);
  padding-right: calc(var(--btnCornerRadius) * 2);
  border: 1px solid var(--vantablack);
}
button:hover, .pretend-button:hover {
  background-color: #ccc;
  cursor: pointer;
}

.login > button {
  background-color: var(--vantablack);
  color: var(--spectralon);
  font-weight: 500;
}
.login > button:hover {
  background-color: #555;
}

```

Nous avons ensuite des classes de base, utilisées pour ne pas avoir à redéfinir constamment des règles CSS et à seulement avoir à utiliser des classes pour des règles utilisées souvent, comme des tailles de police, ou une ombre particulière.

Nous faisons aussi en sorte que le body soit un flex vertical, pour permettre un footer qui ne parte pas loin au-dessus sur un écran suffisamment grand.

```

.bigfont {
  font-size: var(--bigfont);
}
.gigafont {
  font-size: var(--gigafont);
}
.megalovafont {
  font-size: calc(3 * var(--gigafont));
}

body {
  background-image: url(img/background.webp);
  background-size: cover;
  margin: 0px;
  display: flex;
  flex-direction: column;
}

.title {
  font-family: "Title Font";
  font-weight: 800;
}

.shadow {
  text-shadow: 3px 3px 0px var(--secondaryc);
}

```

Voici des règles pour le nav du haut: il est en sticky pour qu'il soit toujours affiché, et il y a une bordure en bas ainsi qu'une ombre en dessous.

Le sélecteur CSS “ *+* ”, souvent surnommé [Lobotomised Owl \(Chouette Lobotomisée\)](#), est très pratique, il est très court et mémorable et sélectionne tous les éléments sauf le premier.

Nous faisons en sorte que le login soit affiché à la fin, tout à droite du nav.

```

#topNav {
  position: sticky;
  top: 0;
  background-color: white;
  z-index: 100;
  display: flex;
  border-bottom: 1px solid black;
  box-shadow: 0px 0px 30px var(--shadow);
  align-items: center;
}

#topNav > *+* {
  font-size: calc(var(--bigfont) - 3px);
}

#topNav .login {
  margin-left: auto;
  justify-content: flex-end;
  margin-right: 25px;
  &>button{
    font-size: inherit;
  }
}

```

Nous faisons en sorte que le titre du site soit en gros et qu'il soit bien espacé.

```

#topNav .title {
  font-size: var(--gigafont);
  padding-left: 20px;
}

#topNav .searchbar {
  font-size: calc(var(--bigfont) - 3px);
  padding: 3px;
}

#topNav #searchCtn button {
  font-size: inherit;
  margin-left: 7px;
}

#topNav #logoNav {
  margin-right: 60px;
  padding-top: 10px;
  padding-bottom: 10px;
}

```

À partir de maintenant, je vais ignorer certaines classes et règles que j'ai créées, car elles n'ajoutent souvent que des marges, de légers changements de positions et ne valent pas le coup d'être détaillées. Elles sont juste là pour de légères simplifications.

Nous avons la partie de recherche:

Nous nous assurons qu'elle change de taille progressivement selon la taille de l'écran avec l'utilisation d'un flex et d'un flex-grow.

Le sélecteur `#queryResults thead th[colspan="3"]` permet de sélectionner seulement les th d'une largeur de 3 cases dans un query result dans un header de table (thead). Il permet donc de ne pas influencer les autres th dans le thead.

```
#searchfilterHolder {
  display: flex;
  flex-direction: row;
}

#searchHolder {
  flex-grow: 9;
}

#filterHolder {
  flex-grow: 1;
  padding-left: 2rem;
}

#queryResults thead th {
  border: 1px solid var(--vantablack);
  font-weight: 100;
}
#queryResults thead th[colspan="3"] {
  font-weight: 700;
}
#queryResults>* {
  border: 3px solid var(--vantablack);
}
#queryResults tbody tr {
  border: 1px solid var(--vantablack);
}
```

Nous faisons aussi en sorte que la deuxième colonne se redimensionne plus que les autres, vu qu'elle contient le nom des quizzes, et c'est en général l'information la plus importante, et que 1 ligne sur 2 ait un fond un peu plus foncé pour augmenter la lisibilité du tableau.

```

217  #queryResults tr>*:nth-child(2){
218  |   width: clamp(140px, 70%, 1000px);
219 }
220
221 #queryResults tbody tr:nth-child(odd) {
222 |   background: #0001;
223 }
224
225 .table-m-set tr *:not(.ignore-m-set) {
226 |   padding: 0.6rem;
227 }
~~~
```

Nous avons des règles simplifiées de CSS pour chaque catégorie : elles changent simplement la background color et éventuellement, si besoin pour des raisons de contraste et de lisibilité, rendent le texte blanc.

```

.t-spo {
|   background-color: var(--t-spo);
}
.t-cmo {
|   background-color: var(--t-cmo);
}
.t-cge {
|   background-color: var(--t-cge);
}
.t-lan {
|   background-color: var(--t-lan);
|   color: var(--spectralon);
}
.t-geo {
|   background-color: var(--t-geo);
|   color: var(--spectralon);
}
.t-his {
|   background-color: var(--t-his);
}
.t-fef {
|   background-color: var(--t-fef);
}
.t-sci {
|   background-color: var(--t-sci);
}
.t-aut {
|   background-color: var(--t-aut);
}
```

Avec cette règle, nous nous assurons aussi que si un nom de quiz est trop long, il sera rogné et que une ellipse sera placée à la fin (text-overflow:ellipsis):

Exemple -

“Quiz sur l’histoire des systèmes de gestion de base de données (SGBD) ainsi que les diagrammes de contraintes d’intégrité référentielles”

deviendrait quelque chose comme

“Quiz sur l’histoire des systèmes de gestion de base de données (SGBD)...”

```
.trnc {
  display: inline-block;
  max-width: clamp(200px, 95%, 1000px);
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
}
```

Règle pour le titre de la page (de base c’était le titre du quiz, mais elle est aussi utilisé dans la page résultat et login par exemple)

```
#qznm {
  padding: 20px;
  padding-bottom: 15px;
  padding-top: 15px;
  background-color: var(--primaryc);
}
```

Nous avons aussi les règles pour la page de quiz, utilisant des boutons radio re-stylés pour afficher les réponses et pouvoir interagir avec.

```

.radio-group {
  display: flex;
  flex-direction: column;
  gap: 2px;
  flex-wrap: nowrap;
  margin-top: 0px;
}

.radio-block {
  cursor: pointer;
  display: inline-block;
}

.custom-box {
  padding: var(--hMarginQuiz) 20px;
  border: 2px solid transparent;
  border-radius: 8px;
  text-align: center;
  transition: 0.1s;
  min-width: 100px;
  background-color: #00000015;
}

.custom-box:hover {
  background-color: var(--hiLightColour)
}

.radio-block input:checked + .custom-box {
  background-color: var(--wrongAnswer);
  color: white;
}

.radio-block input:checked + .custom-box {
  background-color: var(--wrongAnswer);
  color: white;
  border-color: var(--wrongAnswer);
}

#question-name {
  margin-top: -15px;
}

```

Par défaut, le hover est en rouge, car 3 réponses sur 4 seront fausses. Le back-end permet de modifier ceci pour la bonne réponse et utilise la variable var(--rightAnswer)

Maintenant, la page résultat:

Nous avons plusieurs textes à afficher pour indiquer différentes informations sur les résultats, et ils doivent être indiqués de façon claire et concise.

Nous avons donc créé un grid, avec une fonction que j'aime particulièrement dans le CSS: le grid-template-areas.

Il permet de créer une sorte de matrice/tableau de lettres, et chaque lettre est assignée à une zone d'un écran. C'est très pratique pour indiquer de façon très simple un layout quelque peu complexe autrement.

```
.bigresult {  
    font-size: calc(var(--gigafont) * 1.5);  
    font-weight: 900;  
    margin-top: 0px;  
    margin-bottom: 0px;  
}  
.titleAboveResult {  
    margin-bottom: 0px;  
}  
.bigResultHolder * {  
    text-align: center;  
}  
.resultsHolder {  
    display: grid;  
    grid-template-areas:  
        "a a a c c b b b"  
        "a a a c c b b b"  
        "a a a d d b b b"  
        "a a a d d b b b";  
    grid-template-columns: repeat(8, 1fr);  
}  
.sideInfoResult {  
    position: relative;  
    transform: translateX(5rem);  
}
```

Nous pouvons ensuite assigner une lettre de zone à chaque identifiant (ou classe ou élément) voulu. Le temps A est donc indiqué à gauche, le placement B à droite, les mauvaises réponses D en bas, et les bonnes réponses C en haut.

```

1 #time {
|   grid-area:a;
}
2 #placement {
|   grid-area:b;
}
3 #wrongHolder {
|   grid-area:d;
}
4 #rightsHolder {
|   grid-area:c;
}

```

Nous assignons aussi une couleur correcte et mauvaise pour le texte indiquant le nombre de réponses correctes/mauvaises, ainsi qu'une taille de police.

```

1 #rights, #wrongs {
2   font-size: calc(var(--gigafont) * 0.77);
3   font-weight: 600;
4 }
5 #rights {
6   color: var(--rightAnswer);
7 }
8 #wrongs {
9   color: var(--wrongAnswer);
10 }

```

Nous avons aussi créé quelques règles pour le leaderboard pour l'afficher d'une façon claire et satisfaisante, comme des bordures à chaque case du tableau (.individualize), ce qui augmente la lisibilité de ce tableau, ayant plus de colonnes que le tableau de résultats de recherche.

```

1 .leaderResults {
2   margin-top: calc(var(--gigafont) * 1.1);
3 }
4 .leaderResults tr *:nth-child(1) {
5   width: 7rem;
6 }

7 .individualize tr>*+* {
8   border-left: 1px solid var(--vantablack);
9 }
10 #queryResults tr span{
11   max-width: 50vw;
12   width: 100%;
13 }

```

Nous avons aussi des règles pour la page de login:

Le formulaire de login a une largeur de 300px, et est placé au centre (avec le left:50% et translateX(-50%). Le left se base sur le point le plus à gauche de l'élément et le déplace à droite, ce qui fait que la gauche de l'élément est au centre de la page (absolute), d'où le translate -50%, qui se base sur la taille même de l'élément: il se déplace à gauche de 50% de sa propre taille.

```
.login-form {  
    padding: 10px;  
    border-radius: 6px;  
    width: 300px;  
    position: absolute;  
    left: 50%;  
    transform: translateX(-50%);  
}  
  
.login-form label {  
    display: block;  
    margin-bottom: 6px;  
    font-weight: bold;  
    text-align:center;  
}  
  
.login-form input {  
    width: 100%;  
    padding: 8px;  
    margin-bottom: 16px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
    box-sizing: border-box;  
}
```

```

.login-form button {
  width: 100%;
  padding: 10px;
  background: var(--cerulean);
  color: white;
  border: 2px solid transparent;
  border-radius: 4px;
  cursor: pointer;
}

.login-form button:hover {
  background: var(--spectralon);
  color: var(--cerulean);
  border: 2px solid var(--cerulean);
}

```

Nous avons aussi des règles CSS très simples pour la page 404, qui font en sorte que le 404 soit bien affiché en gros au bon endroit sur la page.

```

#error404 #belowNavCtn {
  position: absolute;
  top: 50%;
  left: calc(50% - var(--hMarginQuiz));
  transform: translate(-50%, -50%);
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}

.ul {
  text-decoration: underline;
}

```

Comme sur beaucoup de pages 404 de sites “fun”, nous avons aussi créé un petit “easter egg” sur la page 404, affichant un webp animé (afin de réduire la taille des fichiers à charger) si on maintient la souris sur le chiffre 404 pendant au moins 6 secondes. Cela utilise des keyframes et une pseudo classe :hover.

```

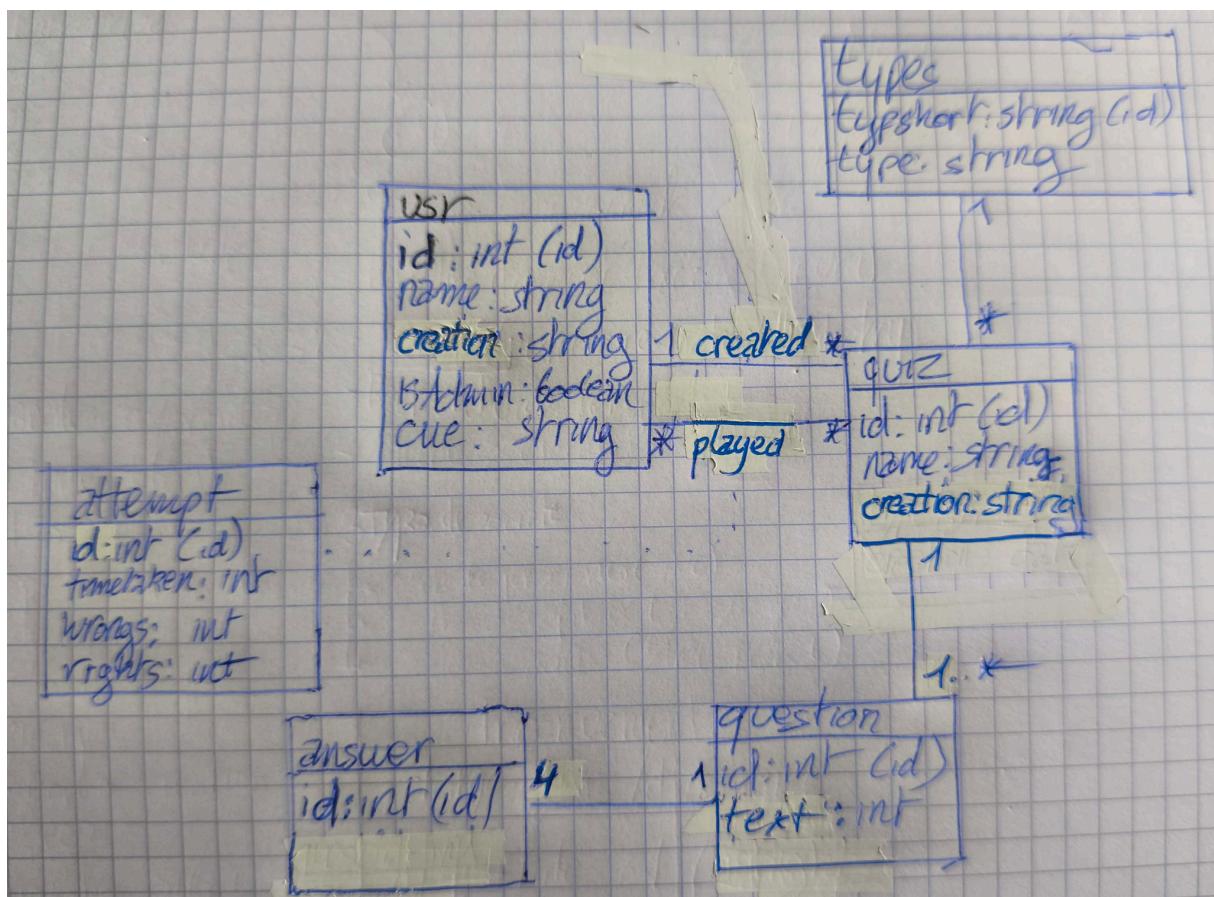
@keyframes legiftruc {
  0% {
    opacity: 0;
  }
  60% {
    opacity: 0;
  }
  60.001% {
    opacity: 1;
  }
  100% {
    opacity: 1;
  }
}
#ouais {
  opacity: 0;
  pointer-events: none;
  position: absolute;
  left: -15vw;
  bottom: -10vw;
}
#belowNavCtn:hover #ouais {
  animation: legiftruc 10s forwards;
}

```

4/ La base de données en SQL.

Nous avons aussi créé une base de données en MySQL pour garder toutes informations.

Nous avons commencé par créer un diagramme UML :



Un utilisateur (usr) est défini par un ID, un nom, une date de création, si il est admin, et la valeur d'encryption de mot de passe.

Un utilisateur admin peut créer autant de quiz qu'il veut. Un quiz est forcément créé par une seule et unique personne.

Un utilisateur lambda peut jouer à autant de quiz qu'il veut, et un quiz peut être joué par autant d'utilisateurs que possible.

Chaque tentative est stockée selon son id, le temps pris, le nombre de bonnes réponses, le nombre de mauvaises réponses. Il peut y avoir plein de tentatives pour un utilisateur. Finalement, seulement la meilleure sera affichée sur le site, mais les autres sont bel et bien gardées.

Un quiz a un seul type (catégorie), et un type peut avoir tout nombre de quiz.

Un quiz a au moins une question, et une question ne peut être assignée qu'à un quiz.

La question a un identifiant, et un texte posant la question.

Une question a quatre réponses possibles, et une réponse ne peut être assignée qu'à une seule question.

Une réponse a un id, et un texte (**oublié dans le diagramme UML, mais bel et bien présent**).

Ce diagramme UML de base de donnée relationnelle permet donc d'avoir un système de quiz simple, ainsi qu'une gestion des utilisateurs ainsi que leurs scores.

Nous avons ensuite traduit ce diagramme en MySQL. Il n'y a pas eu de difficultés particulières dans la création des tables, car elles ne consistaient que de créer les attributs des classes UML et des contraintes pour les lier comme des clés primaires et étrangères.

Nous avions aussi inséré des valeurs de test :

```
‣ INSERT INTO usr(id,name,creation,isAdmin,cue) values
  | (1,'ashley','2025-02-01',true,'tsdjfnfs'),
  | (2,'doryan','2025-04-29',false,'sqdfsdfds')
  ;
‣ INSERT INTO quiz(id,name,creation) values
  | (1,'quiz sur le football','2025-03-04'),
  | (2,'quiz internet des années 2010','2025-04-29')
  ;
‣ INSERT INTO author(quizid,usrid) values
  | (1,1),
  | (2,1)
  ;
‣ INSERT INTO attempt(quizid,usrid,attemptid,timetaken,wrong,rights) values
  | (1,1,1,504,1,13),
  | (1,2,1,572,6,8),
  | (1,1,2,492,0,14)
  ;
```

(cas fictionnels et incomplets car ils n'ont pas de questions créées, mais ils suffisent pour tester les leaderboards.)

Et nous avons donc pu tester nos valeurs sur des cas fictionnels de tests.

Voici trois sélecteurs SQL : un sélectionnant la liste de quizs disponibles, un sélectionnant tous les utilisateurs et utilisatrices de Ultraquizz, et un autre créant une vue leaderboard et le trie par temps pris.

```
SELECT * FROM quiz;
SELECT * FROM usr;

CREATE VIEW ldb AS
SELECT usr.name, qz.name, att.attemptid, att.rights, att.wrong, att.timetaken
FROM usr
INNER JOIN attempt att
ON att.usrid = usr.usrid
INNER JOIN quiz qz
ON lb.lbid = att.lbid
ORDER BY att.timetaken
;
```

À partir de cette vue, on peut sélectionner un quizz en particulier, et donc obtenir les données du leaderboard de tout quizz voulu, puis ensuite de les afficher en utilisant le PHP/back-end du site.

(une sélection à partir de ce leaderboard prendrait cette forme:

```
SELECT * FROM ldb WHERE <condition voulue>;
```