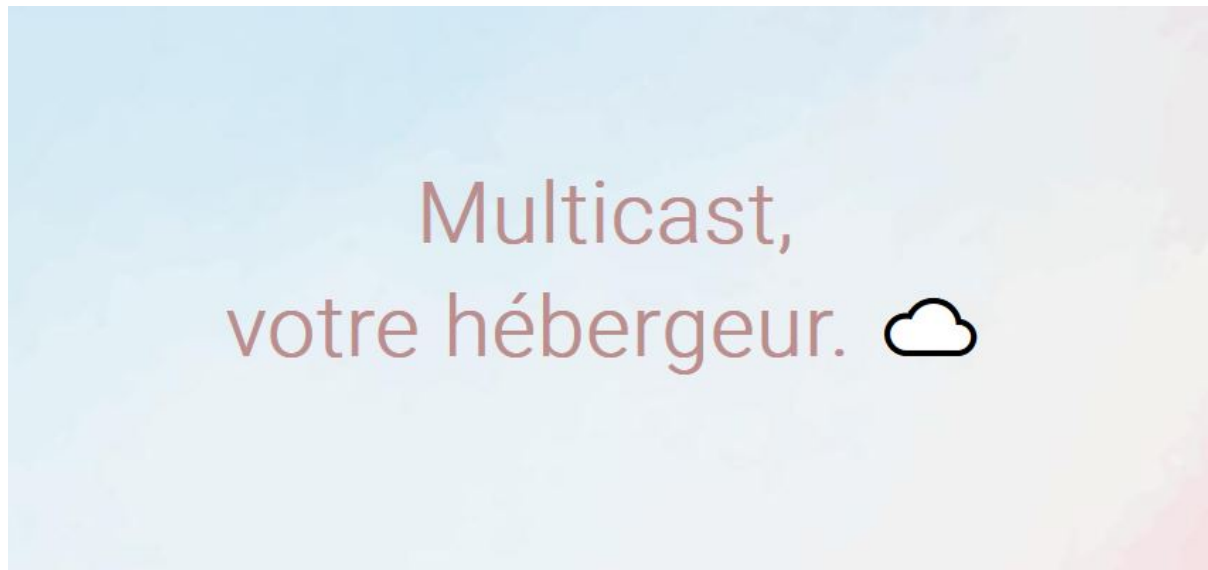


Présentation de notre projet '**Multicast**'



Introduction :

Notre site "**Multicast**" est une plateforme d'hébergement de serveurs en ligne. Il loue contre une rémunération, différents types de serveurs qui seraient adaptés au besoin du client. Cela comprend les serveurs de stockage, de jeu et web.

Présentation des fonctionnalités (Comptes):

L'inscription :

Chaque visiteur du site a la possibilité de se créer un compte via un formulaire créer pour l'occasion.

Inscription

Bienvenue sur notre plateforme
d'hébergement 😊

Nom :

Prénom :

E-mail :

Mot de passe :

Vérifiez votre mot de passe :

~ *En back-end* ~

Une requête SQL va automatiquement se faire en PHP pour insérer le nouvel utilisateur dans la base de données. Le mot de passe est automatiquement hashé en SHA-1 pour la sécurité.

La connexion :

Une fois le compte créé, l'utilisateur peut naturellement se connecter sur le site via la page dédiée.

Connexion

E-mail

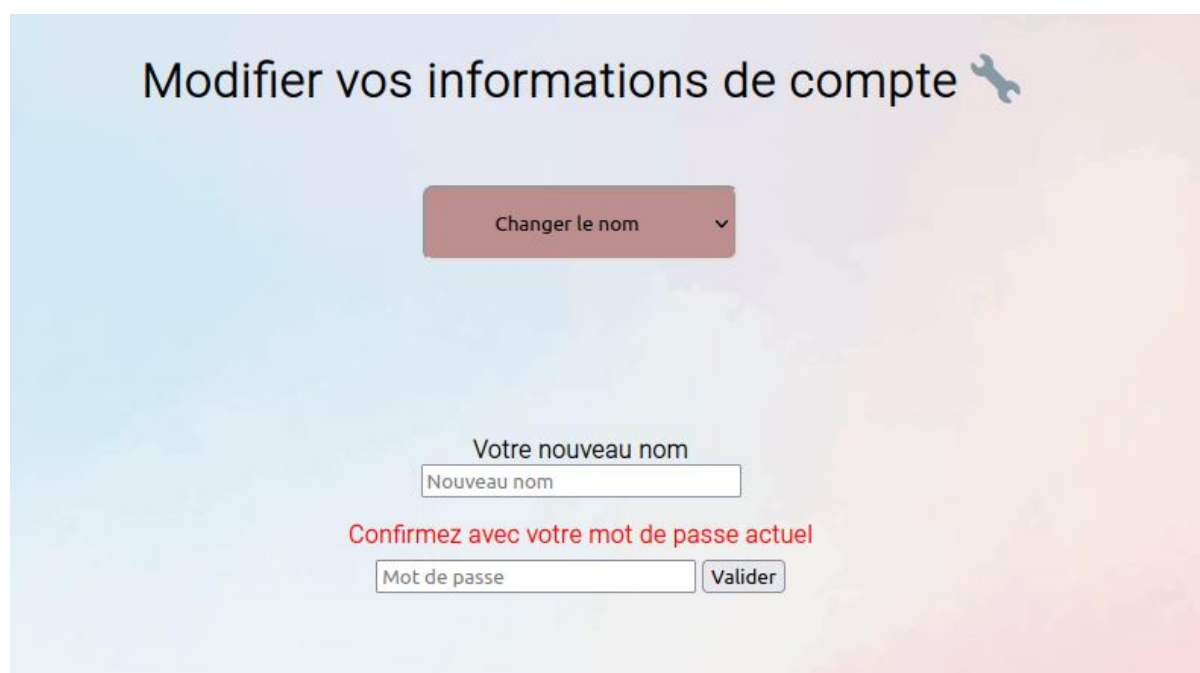
Mot de passe

~ *En back-end* ~

Une requête SQL va automatiquement se faire en PHP pour vérifier les données par rapport à la base de données. Le formulaire prend en compte les erreurs et les retournes pour guider l'utilisateur (Mauvais e-mail... ainsi de suite)

Modification des informations de compte :

Si l'utilisateur souhaite modifier des informations de son compte comme son adresse mail, son mot de passe ou encore son pseudo, il peut le faire via une rubrique de gestion de compte.



The screenshot shows a web form titled "Modifier vos informations de compte" with a wrench icon. Below the title is a dropdown menu labeled "Changer le nom" with a downward arrow. Underneath is a text input field labeled "Votre nouveau nom" with the placeholder text "Nouveau nom". Below the input field is a red text prompt "Confirmez avec votre mot de passe actuel". At the bottom are two input fields: "Mot de passe" and a "Valider" button.

~ *En back-end* ~

Cela modifie automatiquement les données associées à chaque utilisateur. Les saisies sont filtrées pour éviter diverses injections SQL et ainsi les failles de sécurité.

Remarques : Tout ce qui est supprimé, que ce soit les utilisateurs, les serveurs... restent dans la base de données avec une valeur nulle à l'élément 'Actif' pour pouvoir garder un historique de chaque action et une certaine harmonie de la base de données.

Présentation des fonctionnalités (Location) :

La location de serveur :

Il est possible au visiteur une fois inscrit, d'accéder à un espace de location où il peut chercher en temps réel, par rapport à la disponibilité des serveurs et à ses besoins, la liste des serveurs louables en fonction de leurs caractéristiques.

Espace de location de serveurs

Type de serveur ▼ Puissance du CPU ▼ Quantité de RAM ▼ Je veux !

Type : WEB / Nombre minimum de coeurs : 32 / Quantité minimal de RAM : 256 Go

IdServeur —	Type —	CPU —	RAM —	STOCKAGE —	Action
59	WEB	32	256	250	Louer

Il peut alors choisir le serveur qui lui plait, puis d'un clic, le louer. Il lui sera alors attribué et ne pourra plus être loué par d'autres personnes.

~ *En back-end* ~

Lorsque l'utilisateur clique sur le bouton "Louer", le serveur choisi est automatiquement relié à l'utilisateur via un IDutilisateur. C'est comme ça qu'il se démarque des autres serveurs libres (avec un IDutilisateur nul).

La gestion des serveurs loués :

Une fois que l'utilisateur a loué ses différents serveurs, il a accès à une nouvelle page où il peut voir le récapitulatif des différents serveurs qu'il a pris, également la date de location, les spécificités mais également les rendre (ce qui rendra à nouveau possible la possibilité de louer ce dit serveur).

Espace de gestion de serveurs loués

Le serveur à bien été loué.

☐ Afficher seulement les locations en cours

Début —	Fin —	IdServeur —	Type —	CPU —	RAM —	STOCKAGE —	Action
2022-06-09 17:38:28		23	WEB	32	128	2000	Rendre
2022-06-09 17:38:28		20	STOCKAGE	8	128	250	Rendre
2022-06-09 17:38:28		1	JEU	32	256	2000	Rendre
2022-05-20 10:00:00	2022-06-09 17:38:28	13	JEU	8	32	500	Rendre
2022-06-09 18:10:13		59	WEB	32	256	250	Rendre

Il est possible de trier par ordre dé/croissant chaque options ainsi que d'activer la fonction qui permet d'afficher uniquement les locations en cours. Les autres serveurs sont présents comme d'un historique de ses anciennes transactions.

~ *En back-end* ~

Une requête SQL va automatiquement se faire en fonction de l'ID de l'utilisateur connecté en regardant son ID de connexion. Il est donc possible d'afficher toutes les informations reliées au compte, notamment via son IDutilisateur qui lié à chaque serveur qu'il a loué. Le bouton 'Rendre' sert à remettre à zéro l'IDutilisateur lié au serveur.

Présentation des fonctionnalités d'administration :

La gestion des serveurs loués :

Lorsqu'on se connecte avec un compte spécialement prévu pour l'administration (admin en l'occurrence), la page débloquent un nouvel onglet accessible uniquement pour cet utilisateur et qui lui permet de gérer toutes les informations lié aux serveurs ainsi que les utilisateurs.

Pannel de gestion et d'administration des serveurs et des clients

Rechercher des :

☐ Afficher seulement les locations en cours

ID	IDclient	IDserveur	DebutLoc	FinLoc	Action
1	2	23	2022-06-09 17:38:28		Terminer
2	2	20	2022-06-09 17:38:28		Terminer
3	2	1	2022-06-09 17:38:28		Terminer
4	3	8	2022-06-09 17:38:28		Terminer
5	3	30	2022-06-09 17:38:28		Terminer
6	3	5	2022-06-09 17:38:28		Terminer
7	2	13	2022-05-20 10:00:00	2022-06-09 17:38:28	Terminer
8	2	59	2022-06-09 18:10:13		Terminer

L'administrateur peut gérer sur cette page :

- ~ Les serveurs : Cela implique d'afficher l'entièreté des serveurs disponibles ou non (Il y a également un historique des serveurs supprimés) mais également la présence d'un bouton 'Supprimer' qui sert à retirer de la base de donnée (Tout en gardant une trace) un serveur pour qu'il ne soit plus disponible à la location.

Aperçu :

Rechercher des :							
<input type="button" value="Serveurs"/>		<input type="button" value="Clients"/>		<input type="button" value="Locations"/>		<input type="checkbox"/> Afficher seulement les serveurs en services	
ID	Type	Cpu	Ram	Stockage	idLocation	EnService	Action
1	JEU	32	256	2000	3	1	Supprimer
2	JEU	16	128	500		1	Supprimer
3	JEU	8	32	250		1	Supprimer
4	JEU	8	128	500		1	Supprimer
5	WEB	8	256	500		1	Supprimer
6	STOCKAGE	32	128	250		0	Supprimer
7	JEU	8	32	2000		0	Supprimer
8	STOCKAGE	32	256	250	4	1	Supprimer

- ~ Les clients : Cet onglet affiche tous les utilisateurs inscrits sur le site ainsi que leurs informations principales comme leur adresse e-mail, leur ID, nom/prénom, mais il y a également un bouton 'Supprimer' qui sert simplement à retirer un utilisateur de la base de données pour qu'il ne puisse plus se connecter. Cela a pour effet de remettre à zéro tous les serveurs auquel il était lié.

Aperçu :

Rechercher des :

Serveurs

Clients

Locations

☐ Afficher seulement les clients actifs

ID	Email	Nom	Prenom	Actif	Action
2	toto@protonmail.com	toto	toto	1	Supprimer
3	tata@gmail.com	tata	tata	1	Supprimer

- ~ Les locations : Finalement dans cette section se trouve tous les serveurs qui sont ou ont été loués. C'est pratique pour avoir une vue d'ensemble de l'activité des serveurs. Il y a un bouton 'Terminer' qui permet lui d'arrêter la location d'un serveur.

Aperçu :

Rechercher des :

Serveurs

Clients

Locations

☐ Afficher seulement les locations en cours

ID	IDclient	IDserveur	DebutLoc	FinLoc	Action
1	2	23	2022-06-09 17:38:28		Terminer
2	2	20	2022-06-09 17:38:28		Terminer
3	2	1	2022-06-09 17:38:28		Terminer
4	3	8	2022-06-09 17:38:28		Terminer
5	3	30	2022-06-09 17:38:28		Terminer
6	3	5	2022-06-09 17:38:28	2022-06-09 19:03:10	Terminer
7	2	13	2022-05-20 10:00:00	2022-06-09 17:38:28	Terminer
8	2	59	2022-06-09 18:10:13		Terminer

Présentation des facilités de développement:

Le même environnement pour tous

Afin de faciliter le processus de développement nous avons mis en place une infrastructure avec des conteneurs gérés sous docker. Tout les services ont la mêmes versions que sur le serveur r207.borelly.net .

```
[...] Running 4/4
:: Network sae23_fullop_fiore_moretti_sae23net Created 0.0s
:: Container sae23_fullop_fiore_moretti-http-1 Created 0.0s
:: Container sae23_fullop_fiore_moretti-sql-1 Created 0.0s
:: Container sae23_fullop_fiore_moretti-pma-1 Created 0.0s
Attaching to sae23_fullop_fiore_moretti-http-1, sae23_fullop_fiore_moretti-pma-1, sae23_fullop_fiore_moretti-sql-1
sae23_fullop_fiore_moretti-http-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.1
68.0.3. Set the 'ServerName' directive globally to suppress this message
sae23_fullop_fiore_moretti-http-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.1
68.0.3. Set the 'ServerName' directive globally to suppress this message
sae23_fullop_fiore_moretti-http-1 | [Fri Jun 10 14:10:54.519515 2022] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.25 (Debian) PHP/5
.6.40 configured -- resuming normal operations
sae23_fullop_fiore_moretti-http-1 | [Fri Jun 10 14:10:54.519580 2022] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUN
D'
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] mysqld (mysqld 10.0.38-MariaDB-1-xenial) starting as process 1 ...
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: Using mutexes to ref count buffer pool pages
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: The InnoDB memory heap is disabled
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: Compressed tables use zlib 1.2.8
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: Using linux native AIO
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: Using CPU crc32 instructions
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: Initializing buffer pool, size = 256.0M
sae23_fullop_fiore_moretti-sql-1 | 220610 16:10:54 [Note] InnoDB: Completed initialization of buffer pool
```

Une facilité de déploiement

Le site peut aussi être déployé plus facilement grâce au fichier docker-compose.yml.

```
docker-compose.yml
1 version: "3"
2 services:
3   http:
4     build: ./httpserv/
5     ports:
6       - "8080:80"
7     networks:
8       - sae23net
9     volumes:
10      - ./site:/var/www/html
11   pma:
12     image: phpmyadmin
13     ports:
14       - "8081:80"
15     networks:
16       - sae23net
17     links:
18       - sql
19     environment:
20       PMA_HOST: sql
21   sql:
22     image: mariadb:10.0.38
23     command: --character-set-server=utf8
24     networks:
25       - sae23net
26     environment:
27       - MYSQL_ROOT_PASSWORD=root
28       - TZ=Europe/Paris
29     volumes:
30       - ./db:/var/lib/mysql
31   networks:
32     sae23net:
33       driver: bridge
```

Une base de donnée configurable rapidement

Le script FillDb.py permet de générer des requêtes SQL afin de remplir la base de fausses données, les fichier user.sql et db.sql permettent de mettre en place le squelette de la base de donnée.


```
'ram' int NOT NULL,
'stockage' int NOT NULL,
'idLocation' int NULL DEFAULT NULL,
'enService' boolean NOT NULL DEFAULT 1,
PRIMARY KEY ('id')
) ENGINE = MyISAM;

CREATE TABLE 'multicast'.'clients' (
'id' int NOT NULL AUTO_INCREMENT,
'email' varbinary(50) NOT NULL,
'mdp' varbinary(50) NOT NULL,
'nom' varbinary(50) NOT NULL,
'prenom' varbinary(50) NOT NULL,
'actif' boolean NOT NULL DEFAULT 1,
PRIMARY KEY ('id')
) ENGINE = MyISAM;

CREATE TABLE 'multicast'.'locations' (
'id' int NOT NULL AUTO_INCREMENT,
'idClient' int NOT NULL,
'idServeur' int NOT NULL,
'debutLoc' DATETIME NOT NULL,
'finLoc' DATETIME NULL DEFAULT NULL,
PRIMARY KEY ('id')
) ENGINE = MyISAM;

CREATE VIEW 'multicast'.'ServeursReserve' (idLocation, Debut, Fin, IdServeur, Type, CPU, RAM, STOCKAGE, idClient, mail, nom, prenom)
AS SELECT locations.id, locations.debutLoc, locations.finLoc, locations.idServeur, locations.type, locations.cpu, locations.ram, locations.stockage, locations.idClient, locations.mail, locations.nom, locations.prenom
FROM locations;

CREATE VIEW 'multicast'.'VueClient' (ID, Debut, Fin, IdServeur, Type, CPU, RAM, STOCKAGE, enService, mail) AS SELECT locations.id, locations.debutLoc, locations.finLoc, locations.idServeur, locations.type, locations.cpu, locations.ram, locations.stockage, locations.enService, locations.mail
FROM locations;

CREATE VIEW 'multicast'.'ServeursDispo' (IdServeur, Type, CPU, RAM, STOCKAGE) AS SELECT serveurs.id, serveurs.type, serveurs.cpu, serveurs.ram, serveurs.stockage
FROM serveurs;
```

▲ **Projet Multicast ~ Elouan FIORE, Timothée FULOP, Luigi MORETTI** ▲