

Documentation Technique

Valres

CARCENAC Mathis - GHAMZI Amin

TABLE DES MATIÈRES

1. Installation

2. Connexion au site

3. Langages utilisés

4. Architecture

5. Base de données

- Installation de la base de données (Déploiement)
- Exportation de la base de données

6. Commandes Git

- Envoyer sur un repository à distance
- Retirer depuis un repository à distance (Déploiement)

7. Sécurités & Collecte des données

- Comment les données sont collectées

8. Différents scripts utiles

Installation

Le site web “Valres” est hébergé sur un serveur local grâce à la plateforme de développement Web de type WAMP nommé WampServer, elle permet de faire fonctionner localement les scripts PHP et est aussi utilisé pour la base de données avec PHPMyAdmin qui est intégré à WampServer. Et l’IDE utilisée est Visual Studio Code. Pour le côté versionning et sauvegarde nous avons utilisé l’outil Github, et enfin pour le backlog, nous avons utilisé l’outil Tuleap Campus



Connexion au site

Pour se connecter au site via les différents types d'accès, voici les emails et mots de passes pour se connecter :

Administrateur :

- PICARD Emilie : emilie@gmail.com ; emilie9982

Secrétariat :

- LEROY Nicolas: nicolas@gmail.com ; nicolas456

Responsable :

- MARTIN Laura : laura@gmail.com ; laura789

Utilisateur :

- BANDILELLA Clement : clement@gmail.com ; clement12
- BIACQUEL Veronique : vero@gmail.com ; veronique12321
- SILBERT GILLES : gilles@gmail.com ; gilles111
- TORTEMANN Pierre : pierre@gmail.com ; pierre1932

Langages utilisés

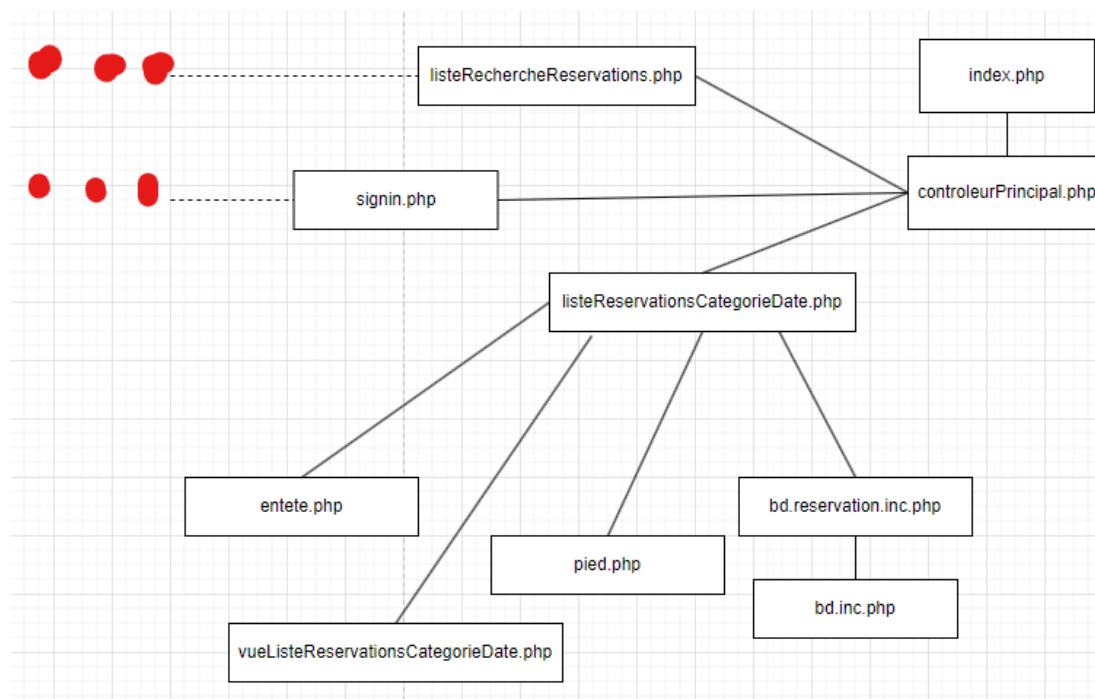
Les langages utilisés sur le site Valres sont HTML pour le placement du contenu, le CSS pour s'occuper de tout le design du site web et enfin on utilise PHP pour tous les scripts du site web, les scripts de connexions à la base de données, l'inclusions des différentes parties du site au contrôleur de la page. Et du Javascript pour toutes les fonctions qui sont présentes dans le site et qui vont faire les requêtes et appeler la base de données pour récupérer différentes informations.



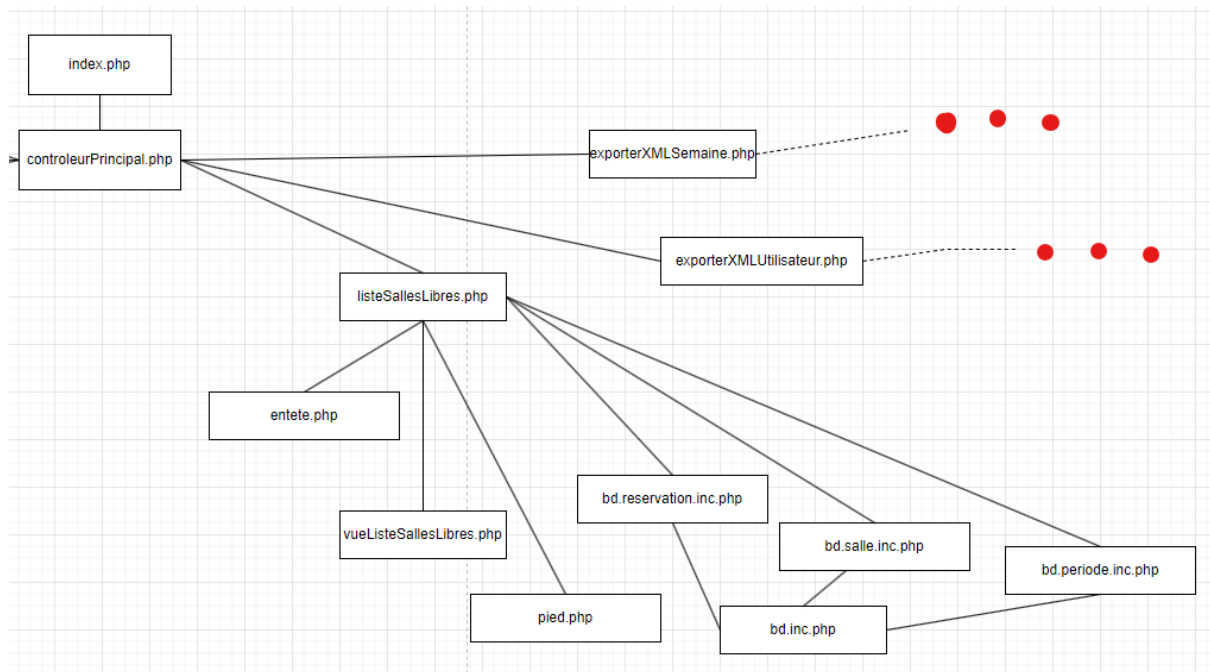
Architecture

Concernant l'architecture du site, il est fait de sorte à respecter le modèle MVC (Modèle Vue Contrôleur), de façon à ce que le site soit structuré en 3 grands fichiers portant ces noms. Le contrôleur principal va appeler les autres contrôleur de façon à ce que chaque page ait une partie "Vue" donc ce que va voir et va être affiché à l'écran, une partie "contrôleur" qui va contrôler l'action de la page avec l'inclusion de la tête et du pied de page puis de son contenu et enfin le modèle qui va permettre de récupérer des données avec la base de données

Exemple pour la page "listeReservationsCategorieDate.php" :



Exemple pour la page listeSallesLibres.php :



Base de données

La base de données comprend 8 tables, la table categorie_salle, etatreservation, periode, reservation, salle, structure, type_acces, utilisateur.

```
DROP TABLE IF EXISTS `categorie_salle`;
CREATE TABLE IF NOT EXISTS `categorie_salle` (
  `idCategorieSalle` int NOT NULL,
  `libelle` varchar(32) NOT NULL,
  PRIMARY KEY (`idCategorieSalle`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `categorie_salle` (`idCategorieSalle`, `libelle`) VALUES
(1, 'Salle de réunion'),
(2, 'Salle avec équipements'),
(3, 'Amphithéâtre');

DROP TABLE IF EXISTS `etatreservation`;
CREATE TABLE IF NOT EXISTS `etatreservation` (
  `idEtat` int NOT NULL AUTO_INCREMENT,
  `libelle` varchar(32) NOT NULL,
  PRIMARY KEY (`idEtat`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `etatreservation` (`idEtat`, `libelle`) VALUES
(1, 'Provisoire'),
(2, 'Confirmé'),
(3, 'Annulé');

DROP TABLE IF EXISTS `periode`;
CREATE TABLE IF NOT EXISTS `periode` (
  `idPeriode` int NOT NULL AUTO_INCREMENT,
  `libelle` varchar(50) NOT NULL,
  PRIMARY KEY (`idPeriode`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `periode` (`idPeriode`, `libelle`) VALUES
```

```

(1, 'Matinée'),
(2, 'Midi'),
(3, 'Après-midi'),
(4, 'Soirée');

DROP TABLE IF EXISTS `reservation`;
CREATE TABLE IF NOT EXISTS `reservation` (
  `idReservation` int NOT NULL AUTO_INCREMENT,
  `datee` date NOT NULL,
  `idPeriode` int NOT NULL,
  `idUtilisateur` int NOT NULL,
  `idEtat` int NOT NULL,
  `idSalle` int NOT NULL,
  `idCreateur` int NOT NULL,
  PRIMARY KEY (`idReservation`),
  KEY `idPeriode` (`idPeriode`),
  KEY `idUtilisateur` (`idUtilisateur`),
  KEY `idEtat` (`idEtat`),
  KEY `reservation_ibfk_3` (`idSalle`),
  KEY `idCreateur` (`idCreateur`)
) ENGINE=InnoDB AUTO_INCREMENT=88 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `reservation` (`idReservation`, `datee`, `idPeriode`,
`idUtilisateur`, `idEtat`, `idSalle`, `idCreateur`)
VALUES
(1, '2024-01-10', 1, 1, 1, 1, 9), -- Provisoire: Creator ID 9
(2, '2024-01-11', 2, 2, 1, 1, 9), -- Provisoire: Creator ID 9
(3, '2024-01-12', 3, 3, 1, 1, 9), -- Provisoire: Creator ID 9
(4, '2024-01-15', 4, 4, 2, 1, 8), -- Confirmé: Creator ID 8
(5, '2024-01-16', 2, 5, 2, 1, 8), -- Confirmé: Creator ID 8
(6, '2024-01-14', 1, 1, 1, 1, 9), -- Provisoire: Creator ID 9
(7, '2024-01-14', 2, 2, 1, 2, 9), -- Provisoire: Creator ID 9
(8, '2024-01-14', 3, 3, 1, 3, 9), -- Provisoire: Creator ID 9
(9, '2024-01-14', 4, 4, 2, 4, 8), -- Confirmé: Creator ID 8
(10, '2024-01-14', 2, 5, 2, 5, 8), -- Confirmé: Creator ID 8
(11, '2024-01-14', 3, 6, 2, 6, 9), -- Provisoire: Creator ID 9
(12, '2024-01-09', 1, 1, 1, 1, 9), -- Provisoire: Creator ID 9
(13, '2024-01-10', 2, 2, 1, 2, 9), -- Provisoire: Creator ID 9
(14, '2024-01-11', 3, 3, 1, 3, 9), -- Provisoire: Creator ID 9
(15, '2024-01-09', 1, 1, 1, 1, 9), -- Provisoire: Creator ID 9
(16, '2024-01-10', 2, 2, 1, 2, 9), -- Provisoire: Creator ID 9
(17, '2024-01-11', 3, 3, 1, 3, 9), -- Provisoire: Creator ID 9
(18, '2024-01-15', 1, 4, 2, 4, 8), -- Confirmé: Creator ID 8
(19, '2024-01-16', 2, 5, 1, 5, 9), -- Provisoire: Creator ID 9
(20, '2024-01-17', 3, 6, 1, 6, 9), -- Provisoire: Creator ID 9

```



```

(21, '2024-01-17', 3, 7, 1, 5, 9), -- Provisoire: Creator ID 9
(22, '2024-01-10', 3, 2, 1, 6, 9), -- Provisoire: Creator ID 9
(23, '2024-01-08', 3, 7, 1, 1, 9), -- Provisoire: Creator ID 9
(24, '2024-01-14', 3, 1, 2, 4, 9), -- Provisoire: Creator ID 9
(25, '2024-01-06', 3, 10, 3, 7, 8),
(26, '2024-01-30', 4, 8, 2, 13, 8),
(27, '2024-01-02', 3, 8, 2, 10, 8),
(28, '2024-01-19', 2, 4, 1, 5, 9),
(29, '2024-01-02', 1, 9, 2, 1, 8),
(30, '2024-01-13', 1, 1, 3, 9, 8),
(31, '2024-01-17', 3, 3, 3, 13, 9),
(32, '2024-01-07', 3, 9, 3, 10, 8),
(33, '2024-01-10', 3, 4, 1, 4, 9),
(34, '2024-12-09', 4, 6, 3, 5, 8),
(35, '2024-01-27', 3, 8, 2, 9, 8),
(36, '2024-01-03', 1, 1, 3, 5, 9),
(37, '2024-01-05', 3, 3, 3, 5, 9),
(38, '2024-01-07', 1, 5, 1, 8, 8),
(39, '2024-01-05', 3, 7, 1, 1, 8),
(40, '2024-01-13', 3, 2, 3, 11, 9),
(41, '2024-01-14', 1, 7, 2, 5, 8),
(42, '2024-01-17', 1, 5, 2, 3, 8),
(43, '2024-01-22', 4, 3, 1, 1, 9),
(44, '2024-01-06', 4, 10, 3, 14, 8);

```

```

DROP TABLE IF EXISTS `salle`;

```

```

CREATE TABLE IF NOT EXISTS `salle` (
  `idSalle` int NOT NULL,
  `salle_nom` varchar(32) NOT NULL,
  `capacite` int NOT NULL,
  `idCategorieSalle` int NOT NULL,
  PRIMARY KEY (`idSalle`),
  KEY `salle_ibfk_1` (`idCategorieSalle`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

```

INSERT INTO `salle` (`idSalle`, `salle_nom`, `capacite`, `idCategorieSalle`)
VALUES

```

```

(1, 'Daum', 15, 1),
(2, 'Corbin', 20, 1),
(3, 'Baccarat', 30, 1),
(4, 'Longwy', 12, 1),
(5, 'Multimédia', 50, 2),
(6, 'Amphithéâtre', 200, 3),
(7, 'Lamour', 50, 1),
(8, 'Grüber', 40, 1),
(9, 'Majorelle', 50, 1),

```

```

(10, 'Salle de restauration',50, 2),
(11, 'Galerie', 25, 1),
(12, 'Salle informatique', 20, 2),
(13, 'Hall d\'accueil', 50, 2),
(14, 'Gall  ',40, 1);

DROP TABLE IF EXISTS `structure`;
CREATE TABLE IF NOT EXISTS `structure` (
  `idStructure` int NOT NULL,
  `structure_nom` varchar(80) DEFAULT NULL,
  `structure_adresse` varchar(80) NOT NULL,
  PRIMARY KEY (`idStructure`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `structure` (`idStructure`, `structure_nom`, `structure_adresse`)
VALUES
(1, 'Ligue d\'escrime Lorraine', '5, rue des trois   pis 54600 Villers l  s
Nancy'),
(2, 'Fives Nordon', '5 Pl. Aim   Morot 54000 Nancy'),
(3, 'FFT- COMITE DEPARTEMENTAL DE TENNIS DE MOSELLE', '42, rue de la
commanderie 54840 Sexey les bois'),
(4, 'Ligue Volley Ball Lorraine', '30, rue Widric 1er 54600 Villers l  s
Nancy'),
(5, 'Sporting Club Ennery', '48 Rue Marcel Decker, 57365 Ennery'),
(6, 'Lyc  e public Frederic Chopin', '39 rue du Sergent Blandan 54000 Nancy'),
(7, 'Association Sportive Nancy Lorraine (ASNL)', '30, rue Widric 1er 54600
Villers l  s Nancy'),
(8, 'Maison des ligues', '5 Rue Alb  ric 57000 Metz');

DROP TABLE IF EXISTS `type_accees`;
CREATE TABLE IF NOT EXISTS `type_accees` (
  `idAccees` int NOT NULL,
  `libelle` varchar(50) NOT NULL,
  PRIMARY KEY (`idAccees`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

INSERT INTO `type_accees` (`idAccees`, `libelle`) VALUES
(1, 'Administrateur'),
(2, 'Secr  tariat'),
(3, 'Responsable'),
(4, 'Utilisateur');

DROP TABLE IF EXISTS `utilisateur`;
CREATE TABLE IF NOT EXISTS `utilisateur` (
  `idUtilisateur` int NOT NULL AUTO_INCREMENT,

```

```

`nom` varchar(32) NOT NULL,
`prenom` varchar(50) NOT NULL,
`mail` varchar(32) NOT NULL,
`motDePasse` varchar(50) NOT NULL,
`idStructure` int NOT NULL,
`idAccees` int NOT NULL,
PRIMARY KEY (`idUtilisateur`),
KEY `idStructure` (`idStructure`),
KEY `idAccees` (`idAccees`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

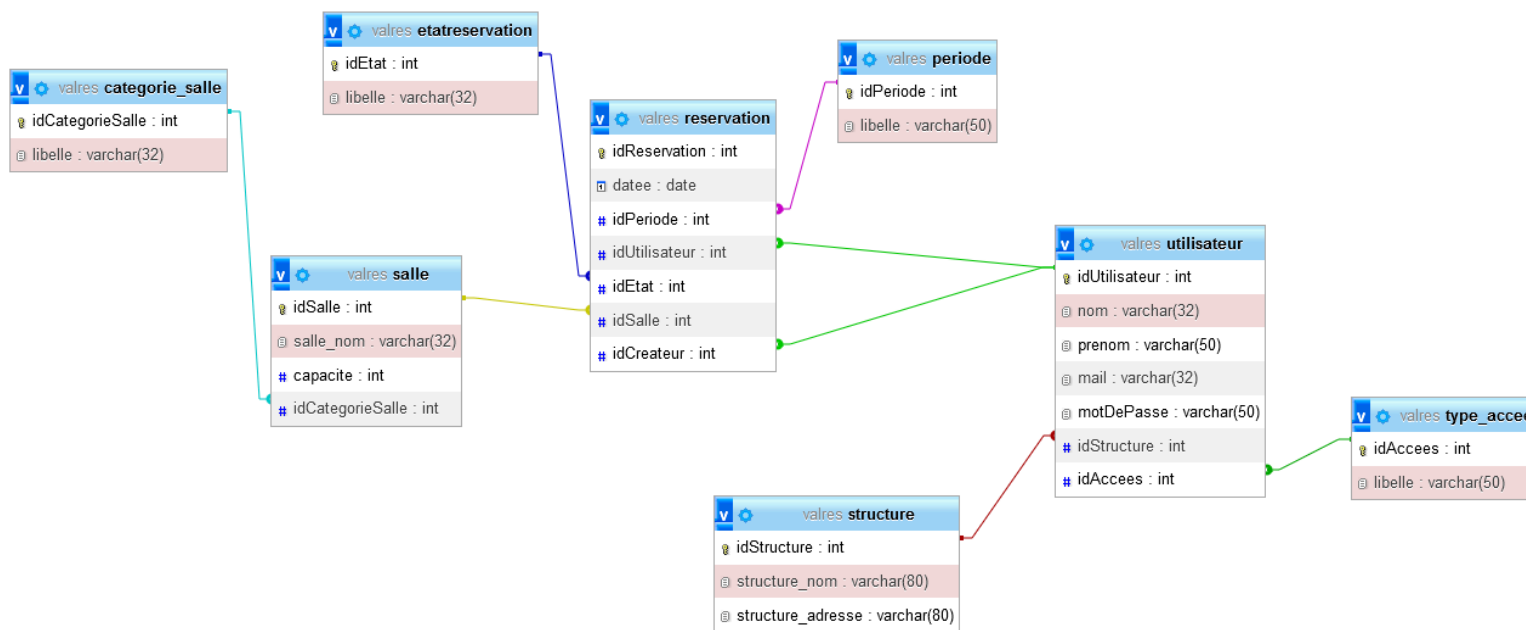
INSERT INTO `utilisateur` (`idUtilisateur`, `nom`, `prenom`, `mail`,
`motDePasse`, `idStructure`, `idAccees`) VALUES
(1, 'BANDILELLA', 'CLEMENT', 'clement@gmail.com', 'clement12', 4, 4),
(2, 'BIACQUEL', 'VERONIQUE', 'vero@gmail.com', 'veronique12321', 2, 4),
(3, 'SILBERT', 'GILLES', 'gilles@gmail.com', 'gilles111', 5, 4),
(4, 'TORTEMANN', 'PIERRE', 'pierre@gmail.com', 'pierre1932', 7, 4),
(5, 'PERNOT', 'LEA', 'lea@gmail.com', 'lea1408', 6, 4),
(6, 'ZUEL', 'STEPHANIE', 'stephanie@gmail.com', 'stephanie131231', 2, 4),
(7, 'LIEVIN', 'NATHAN', 'nathan@gmail.com', 'nathan1111', 3, 4),
(8, 'LEROY', 'NICOLAS', 'nicolas@gmail.com', 'nicolas456', 7, 2),
(9, 'MARTIN', 'LAURA', 'laura@gmail.com', 'laura789', 7, 3),
(10, 'PICARD', 'Emilie', 'emilie@gmail.com', 'emilie9982', 7, 1);

ALTER TABLE `reservation`
    ADD CONSTRAINT `reservation_ibfk_1` FOREIGN KEY (`idUtilisateur`)
REFERENCES `utilisateur` (`idUtilisateur`),
    ADD CONSTRAINT `reservation_ibfk_2` FOREIGN KEY (`idEtat`) REFERENCES
`etatreservation` (`idEtat`),
    ADD CONSTRAINT `reservation_ibfk_3` FOREIGN KEY (`idSalle`) REFERENCES
`salle` (`idSalle`),
    ADD CONSTRAINT `reservation_ibfk_4` FOREIGN KEY (`idPeriode`) REFERENCES
`periode` (`idPeriode`),
    ADD CONSTRAINT `reservation_ibfk_5` FOREIGN KEY (`idCreateur`) REFERENCES
`utilisateur` (`idUtilisateur`);

ALTER TABLE `salle`
    ADD CONSTRAINT `salle_ibfk_1` FOREIGN KEY (`idCategorieSalle`) REFERENCES
`categorie_salle` (`idCategorieSalle`);

ALTER TABLE `utilisateur`
    ADD CONSTRAINT `utilisateur_ibfk_1` FOREIGN KEY (`idStructure`) REFERENCES
`structure` (`idStructure`),
    ADD CONSTRAINT `utilisateur_ibfk_3` FOREIGN KEY (`idAccees`) REFERENCES
`type_accees` (`idAccees`);

```



Installation Base de données

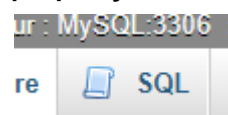
Pour installer la base de données vous aurez besoin de phpMyAdmin ou son équivalent, mais c'est cette interface qu'on utilise via WampServer.

Pour que MyCoach soit connecté à la BDD l'identifiant doit être "root" et ne pas contenir de mot de passe car c'est écrit comme cela dans le code . Le nom de la bdd est "valres"

On va donc créer une base de données

The screenshot shows the 'Création d'une base de données' (Create a new database) dialog in phpMyAdmin. The database name 'valres' is entered in the first field, and the character set 'utf8mb4_0900_ai_ci' is selected in the second field. A 'Créer' (Create) button is visible on the right.

Une fois cela fait on va aller dans la catégorie "SQL" de phpMyAdmin



Et on Copie/Colle la requête SQL pour insérer les données et créer les tables qui se situent dans le fichier Valres:

Sélectionner MySQL 8.0.0 » Base de données : valres

StructureSQLRechercherRequêteExporterImporterOpérationsPrivilègesP

Exécuter une ou des requêtes SQL sur la base de données « valres »:

197

198 ALTER TABLE `reservation`

199 ADD CONSTRAINT `reservation_ibfk_1` FOREIGN KEY (`idUtilisateur`) REFERENCES `utilisateur` (`idUtilisateur`),

200 ADD CONSTRAINT `reservation_ibfk_2` FOREIGN KEY (`idEtat`) REFERENCES `etatreservation` (`idEtat`),

201 ADD CONSTRAINT `reservation_ibfk_3` FOREIGN KEY (`idSalle`) REFERENCES `salle` (`idSalle`),

202 ADD CONSTRAINT `reservation_ibfk_4` FOREIGN KEY (`idPeriode`) REFERENCES `periode` (`idPeriode`),

203 ADD CONSTRAINT `reservation_ibfk_5` FOREIGN KEY (`idCreateur`) REFERENCES `utilisateur` (`idUtilisateur`);

204

205 ALTER TABLE `salle`

206 ADD CONSTRAINT `salle_ibfk_1` FOREIGN KEY (`idCategorieSalle`) REFERENCES `categorie_salle` (`idCategorieSalle`);

207

208

209 ALTER TABLE `utilisateur`

210 ADD CONSTRAINT `utilisateur_ibfk_1` FOREIGN KEY (`idStructure`) REFERENCES `structure` (`idStructure`),

211 ADD CONSTRAINT `utilisateur_ibfk_3` FOREIGN KEY (`idAcces`) REFERENCES `type_acces` (`idAcces`);

212

Et après avoir copié/collé vous devez obtenir ce résultat :

StructureSQLRechercherRequêteExporterImporterOpérationsPrivilègesProcédures stockéesÉvénementsDéclencheursConcepteur

[Éditer en ligne] [Éditer] [Créer le code source PHP]

✓ 3 lignes insérées. (traitement en 0,0031 seconde(s).)

INSERT INTO `etatreservation` (`idEtat`, `libelle`) VALUES (1, 'Provisoire'), (2, 'Confirmé'), (3, 'Annulé');

[Éditer en ligne] [Éditer] [Créer le code source PHP]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0004 seconde(s).)

DROP TABLE IF EXISTS `periode`;

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Note: #1051 Table 'valres.periode' inconnue

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0439 seconde(s).)

CREATE TABLE IF NOT EXISTS `periode` (`idPeriode` int NOT NULL AUTO_INCREMENT, `libelle` varchar(50) NOT NULL, PRIMARY KEY (`idPeriode`)) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

[Éditer en ligne] [Éditer] [Créer le code source PHP]

✓ 4 lignes insérées. (traitement en 0,0030 seconde(s).)

INSERT INTO `periode` (`idPeriode`, `libelle`) VALUES (1, 'Matinée'), (2, 'Midi'), (3, 'Après-midi'), (4, 'Soirée');

[Éditer en ligne] [Éditer] [Créer le code source PHP]

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0004 seconde(s).)

DROP TABLE IF EXISTS `reservation`;

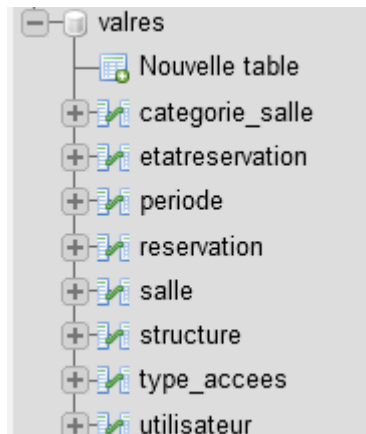
[Éditer en ligne] [Éditer] [Créer le code source PHP]

Et voici votre base de données remplis :

	Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/>	categorie_salle	Parcourir Structure Rechercher Insérer Vidier Supprimer	3	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
<input type="checkbox"/>	etatreservation	Parcourir Structure Rechercher Insérer Vidier Supprimer	3	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
<input type="checkbox"/>	periode	Parcourir Structure Rechercher Insérer Vidier Supprimer	4	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
<input type="checkbox"/>	reservation	Parcourir Structure Rechercher Insérer Vidier Supprimer	44	InnoDB	utf8mb4_0900_ai_ci	96,0 kio	-
<input type="checkbox"/>	salle	Parcourir Structure Rechercher Insérer Vidier Supprimer	14	InnoDB	utf8mb4_0900_ai_ci	32,0 kio	-
<input type="checkbox"/>	structure	Parcourir Structure Rechercher Insérer Vidier Supprimer	8	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
<input type="checkbox"/>	type_acces	Parcourir Structure Rechercher Insérer Vidier Supprimer	4	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
<input type="checkbox"/>	utilisateur	Parcourir Structure Rechercher Insérer Vidier Supprimer	10	InnoDB	utf8mb4_0900_ai_ci	48,0 kio	-
	8 tables	Somme	90	MyISAM	utf8mb4_0900_ai_ci	256,0 kio	0 0

Exportation Base de données

Pour exporter la base de données depuis phpMyAdmin on se place



sur la base de données “Valres”

et on clique sur Exporter



Exporter

Après avoir fait cela on clique sur le bouton “Exporter” en bas

Exportation des tables depuis la base de données « mycoach »

A screenshot of the 'Export' options form in phpMyAdmin. It has a title bar 'Méthode d'exportation :'. Below it, there are two radio buttons: 'Rapide, n'afficher qu'un minimum d'options' (selected) and 'Personnalisée, afficher toutes les options possibles'. Below this is a 'Format :' section with a dropdown menu showing 'SQL'. At the bottom left is an 'Exporter' button.

Et on obtient, en téléchargement le fichier valres.sql



Ce fichier qu'on va par la suite placer dans le dossier valres pour l'envoyer sur Github par la suite.

Commandes GIT

Envoyer sur un Repository à distance

Voici les commandes qui m'ont permis d'insérer les fichiers depuis un fichier local sur un repository Github

Initialiser le Repository local

```
carce@Mathis MINGW64 /c/wamp64/www/Valres
$ git init
Initialized empty Git repository in C:/wamp64/www/Valres/.git/
```

On va créer la branch main sur Github

```
carce@Mathis MINGW64 /c/wamp64/www/Valres (master)
$ git branch -M main
```

On va ajouter tous les éléments du dossier ../www/Valres (le point sert à mettre tous les dossiers)

```
carce@Mathis MINGW64 /c/wamp64/www/Valres (main)
$ git add .
```

On va faire un commit avec un message "Commit 10-01-2024" qui sera affiché sur Github

```
carce@Mathis MINGW64 /c/wamp64/www/Valres (main)
$ git commit -m "Commit 10-01-2024"
[main (root-commit) c690058] Commit 10-01-2024
43 files changed, 2724 insertions(+)
create mode 100644 controleur/accueil.php
create mode 100644 controleur/ajouterReservation.php
create mode 100644 controleur/ajouterUtilisateur.php
create mode 100644 controleur/controleurPrincipal.php
create mode 100644 controleur/exporterXMLSemaine.php
create mode 100644 controleur/exporterXMLUtilisateur.php
create mode 100644 controleur/gererUtilisateur.php
create mode 100644 controleur/listeRechercheReservation.php
create mode 100644 controleur/listeReservationsCategorieDate.php
```

On va créer ajouter l'origine du repository à distance

```
carce@Mathis MINGW64 /c/wamp64/www/Valres (main)
$ git remote add origin https://github.com/MathIzuki/Valres.git
```

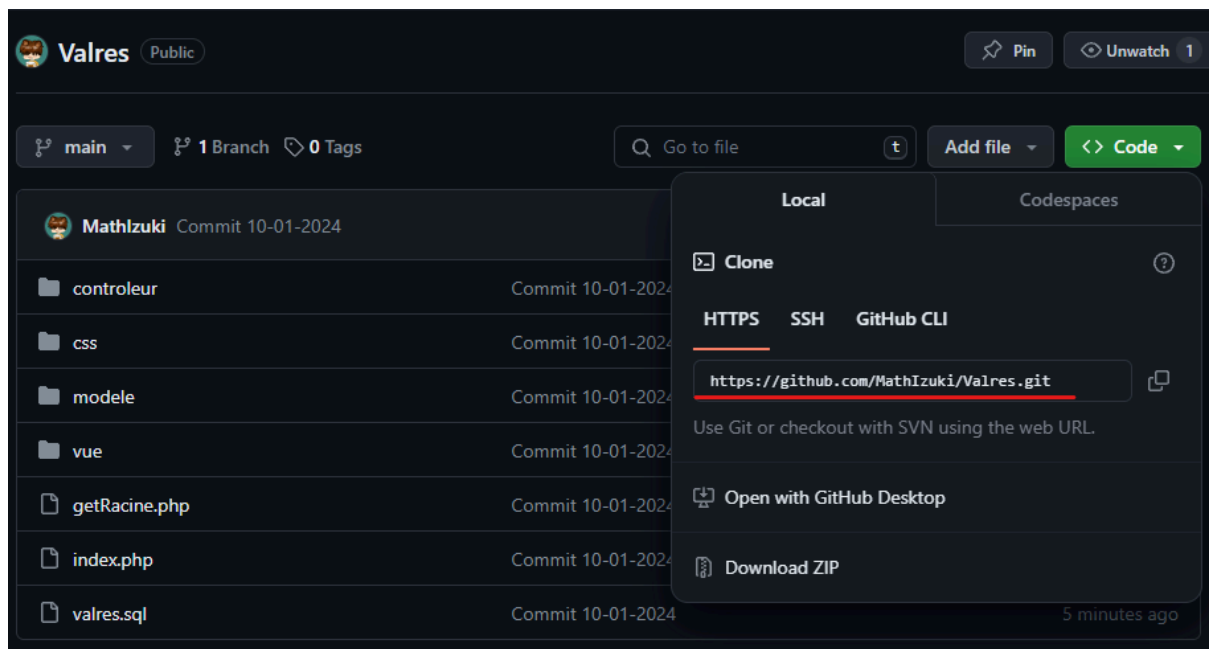
On va ajouter les documents locaux sur Github

```
carce@Mathis MINGW64 /c/wamp64/www/Valres (main)
$ git push -u origin main
Enumerating objects: 49, done.
Counting objects: 100% (49/49), done.
Delta compression using up to 20 threads
Compressing objects: 100% (47/47), done.
Writing objects: 100% (49/49), 25.97 KiB | 8.66 MiB/s, done.
Total 49 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), done.
To https://github.com/MathIzuki/Valres.git
* [new branch]      main -> main
```

Retirer depuis un repository à distance

Commandes pour cloner le repository Github vers un repository local pour pouvoir l'obtenir sur son ordinateur.

Sur Github on va aller dans le repository ou se situe son projet et on va Copier/Coller le lien du Repository.



Par la suite on va ouvrir Git Bash dans le fichier voulu et on fait cette commande qui va importer les données vers le repo local

```
carce@Mathis MINGW64 /c/wamp64/www
$ git clone "https://github.com/MathIzuki/Valres.git"
Cloning into 'Valres'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 49 (delta 7), reused 49 (delta 7), pack-reused 0
Receiving objects: 100% (49/49), 25.97 KiB | 2.00 MiB/s, done.
Resolving deltas: 100% (7/7), done.
```

Une fois cela fait, tous les documents sont présents dans le fichier choisi pour le repository local, et il ne manque plus qu'à les ouvrir avec Visual Studio Code

Collecte des données

Comment les données sont collectées

Données des utilisateurs :

Les données des utilisateurs sont ajoutées par l'Administrateur sur le site, à chaque nouvel utilisateur, il y'aura, son nom, son prénom, son e-mail, son mot de passe (temporaire), la structure à laquelle il appartient, et enfin son type d'accès (généralement il aura un accès utilisateurs, sauf si le nouvel utilisateur est un nouveau secrétaire ou responsable.

Voici l'interface d'ajout des utilisateurs :

Ajouter un nouvel utilisateur

Nom :

Prénom :

Email :

Mot de Passe :

Structure :

Ligue d'escrime Lorraine ▼

Type d'accès :

Administrateur ▼

Ajouter l'utilisateur

Données des réservations :

Concernant les données des réservations, elles peuvent être ajoutées par les responsables et les secrétaires. Lorsqu'ils créeront une nouvelle réservation, il y'aura, la date de cette réservation, la période, la personne qui est à l'origine de cette réservation (le commanditaire) et enfin le nom de la salle qui est réservée.

Ajouter une nouvelle réservation

Date de la réservation:

jj / mm / aaaa

Période:

Matinée

Personne :

BANDILELLA CLEMENT

Nom de la salle :

Daum

Ajouter la réservation

Différents scripts utiles

Fonction "connexionPDO" qui permet de se connecter à la base de données :

```
function connexionPDO() {  
    $login = "root";  
    $mdp = "";  
    $bd = "valres";  
    $serveur = "localhost";  
  
    try {  
        $conn = new PDO("mysql:host=$serveur;dbname=$bd", $login, $mdp, array(PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES \'UTF8\''));  
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
        return $conn;  
    } catch (PDOException $e) {  
        print "Erreur de connexion PDO ";  
        die();  
    }  
}
```

Différentes fonctions qui permettent d'identifier l'état de l'utilisateur en indiquant s'il est connecté, et s'il est connecté son état :

```
function estConnecte(){  
    return isset($_SESSION['utilisateur']);  
}  
  
function estAdmin(){  
    $idAcces = (int)$_SESSION['utilisateur']['idAcces'];  
    return estConnecte() && $idAcces === 1;  
}  
  
function estSecrtaire(){  
    $idAcces = (int)$_SESSION['utilisateur']['idAcces'];  
    return estConnecte() && $idAcces === 2;  
}  
  
function estResponsable(){  
    $idAcces = (int)$_SESSION['utilisateur']['idAcces'];  
    return estConnecte() && $idAcces === 3;  
}  
  
function estUtilisateur(){  
    $idAcces = (int)$_SESSION['utilisateur']['idAcces'];  
    return estConnecte() && $idAcces === 4;  
}
```

Fonction “getReservationsDateCategories” qui prend en paramètre la date et la catégorie qui permet de récupérer toutes les réservations en fonction des paramètres entrées :

```
function getReservationsDateCategories($date, $categorie) {
    $resultat = array();

    try {
        $cnx = connexionPDO();

        $sql = "SELECT p.libelle AS Période, r.datee AS DateReservation, str.structure_nom AS NomStructure, s.salle_nom AS Salle, s.capacite AS Capacite, e.libelle AS EtatReservation
        FROM reservation r
        JOIN periode p ON p.idPeriode = r.idPeriode
        JOIN utilisateur u ON u.idUtilisateur = r.idUtilisateur
        JOIN etatreservation e ON e.idEtat = r.idEtat
        JOIN salle s ON s.idSalle = r.idSalle
        JOIN structure str ON str.idStructure = u.idStructure
        JOIN categorie_salle c ON c.idCategorieSalle = s.idCategorieSalle
        LEFT JOIN utilisateur u_createur ON u_createur.idUtilisateur = r.idCreateur
        WHERE (:date IS NULL OR r.datee = :date) AND (:categorie IS NULL OR s.idCategorieSalle = :categorie)";

        $req = $cnx->prepare($sql);
        $req->bindParam(':date', $date, PDO::PARAM_STR);
        $req->bindParam(':categorie', $categorie, PDO::PARAM_INT);
        $req->execute();

        $resultat = $req->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        // Gérer les erreurs si nécessaire
        // Vous pouvez logger l'erreur, afficher un message, etc.
        echo "Erreur : " . $e->getMessage();
    }

    return $resultat;
}
```

Exemple d'une page qui va utiliser cette fonction
listeReservationsCategorieDate.php :

```
<?php
// Démarrage ou reprise d'une session PHP
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

// Définition de la racine du script pour l'inclusion de fichiers
if ($_SERVER["SCRIPT_FILENAME"] == __FILE__) {
    $racine = ".";
}

// Inclusion du script de gestion de base de données pour les réservations
include_once "$racine/modele/bd.reservation.inc.php";

// Initialisation de la variable pour stocker les réservations
$listeReservations = array();

// Récupération des valeurs soumises par formulaire (date et catégorie)
$date = isset($_POST['date']) ? $_POST['date'] : null;
$categorie = isset($_POST['categorie']) ? $_POST['categorie'] : null;

// Vérification si des filtres de recherche sont appliqués (date ou catégorie)
if (!empty($date) || !empty($categorie)) {
    // Récupération des réservations filtrées selon la date et la catégorie
    $listeReservations = getReservationsDateCategories($date, $categorie);
}

// Titre de la page
$titre = "Les réservations";

// Inclusion de l'en-tête de la page
include "$racine/vue/entete.php";

// Vérification des droits d'utilisateur avant d'afficher la liste des réservations
if(estUtilisateur() or estSecrétaire() or estResponsable() or estAdmin()){
    // Affichage de la vue correspondante aux réservations filtrées
    include "$racine/vue/vueListeReservationsCategorieDate.php";
}
else{
    // Redirection vers la page d'accueil si l'utilisateur n'a pas les droits nécessaires
    header('Location: index.php');
}

// Inclusion du pied de page
include "$racine/vue/pied.php";
?>
```

Et vueListeReservationsCategorieDate.php

```
<div class="page-reservations-liste">
<h1>Liste des réservations</h1>

<form method="post" action="">
  <label for="date">Sélectionner une date :</label>
  <input type="date" id="date" name="date" value="<?= isset($date) ? htmlspecialchars($date) : '' ?>">

  <label for="categorie">Sélectionner une catégorie :</label>
  <select id="categorie" name="categorie">
    <option value="1" <?= ($categorie == "1") ? "selected" : "" ?>>Salle de réunion</option>
    <option value="2" <?= ($categorie == "2") ? "selected" : "" ?>>Salle avec équipements</option>
    <option value="3" <?= ($categorie == "3") ? "selected" : "" ?>>Amphithéâtre</option>
  </select>

  <button type="submit">Filtrer</button>
</form>

<?php if (empty($listeReservations)) { ?>
  <p>Il n'y a aucune réservation à cette date avec cette catégorie.</p>
<?php } else { ?>
  <table border="1">
    <thead>
      <tr>
        <th>Période</th>
        <th>Nom de la structure</th>
        <th>Salle</th>
        <th>Capacité</th>
        <th>État</th>
        <th>Créateur</th>
      </tr>
    </thead>
    <tbody>
      <?php foreach ($listeReservations as $reservation) { ?>
        <tr>
          <td><?= $reservation['Période'] ?></td>
          <td><?= $reservation['NomStructure'] ?></td>
          <td><?= $reservation['Salle'] ?></td>
          <td><?= $reservation['Capacite'] ?> personnes</td>
          <td><?= $reservation['Etat'] ?></td>
          <td><?= $reservation['Createur'] ?></td>
        </tr>
      <?php } ?>
    </tbody>
  </table>
<?php } ?>
</div>
```