



Victor Mongay  
Thomas Lespargot

15 mars 2021

---

# RAPPORT MODAL

## INF472W

---

# 1

## PRÉSENTATION GÉNÉRALE DU PROJET

---

Au cours de ce MODAL Web nous avons développé le site Web du nouveau binet Styl'X, créé l'année dernière et qui a pour but de rassembler les élèves intéressés de près ou de loin par la mode. Les créateurs du binet nous ont demandé que le site soit une sorte de vitrine pour les entreprises qu'ils souhaitent démarcher, et qu'il permette aussi aux élèves de l'école de s'inscrire aux différents événements.

Les différentes fonctionnalités que nous avons ajoutées sur ce site vont donc dans ce sens : une grosse partie du site est interactive et permet aux élèves de visualiser les événements et de s'y inscrire, le tout de façon très visuelle puisque toutes les informations sont présentées sur un calendrier. Une deuxième partie consiste en une présentation des derniers événements avec de nombreuses photos, permettant à tout le monde, même aux personnes extérieures à l'école d'avoir un aperçu de ce qui est proposé par le binet. Comme nous le détaillerons plus tard, le site possède une version admin et une version visiteur/non admin.

Le site est développé en PHP, grâce à Apache et la base de données est codée en MySQL. De nombreuses bibliothèques ont été importées pour le bon fonctionnement du site : fullcalendar, datatables, lighbox, dont on détaillera l'utilité par la suite. Le site repose tout particulièrement sur "JQUERY", qui permet de le rendre dynamique.

## 2

# DÉTAIL DES FONCTIONNALITÉS

---

## 2.1 LE SITE NON ADMIN

---

Commençons par les fonctionnalités communes aux admins et aux visiteurs. Chaque utilisateur a accès à la page "calendrier" (onglet "événements" sur le menu de navigation). Sur ce calendrier sont affichés les événements organisés par le binet : cours de couture, atelier débat, conférence, autres... En cliquant sur un événement, une fenêtre avec les détails s'ouvre, et donne la possibilité à l'utilisateur de s'inscrire en renseignant son mot de passe et son niveau (pour les cours de couture par exemple). Son inscription est alors enregistrée. De plus chaque utilisateur a accès à une page "Mon compte". Cette page lui permet d'avoir accès à la fois à ses inscriptions aux événements, et lui offre la possibilité de modifier son mot de passe, voire de supprimer son compte. Dans ces deux derniers cas, il devra saisir son mot de passe actuel pour effectuer l'opération. Une autre fonctionnalité proposée aux visiteurs est de visualiser les photos des événements déposées sur le site du binet. Il est possible, en cliquant sur une photo de les afficher en diaporama et de quitter ce diaporama dès que l'utilisateur le souhaite. Ces photos sont affichées sur la page à l'aide d'une flexbox. Chaque image porte un identifiant unique, la date de prise de la photo, et une légende qui est affichée pour apporter des informations sur l'événement à l'utilisateur.

## 2.2 LE SITE ADMIN

---

Passons maintenant aux fonctionnalités propres aux administrateurs (qui seront certains membres du binet). Au niveau du calendrier, un administrateur a la possibilité de créer un événement directement en sélectionnant avec la souris la plage horaire de la journée souhaitée. Une fenêtre apparaît alors grâce à JQUERY permettant de définir l'événement (nom, lieu, date, description). En plus de cela, il est possible de sélectionner un événement existant et de modifier nom, horaires, lieu, voire même de le supprimer. Les admins ont aussi accès à un onglet intitulé : "Liste des événements". Il permet d'avoir accès à un tableau récapitulant l'ensemble des événements proposés par le binet. Il voit alors aussi le nombre de personnes inscrites à chaque événement. Il a toujours la possibilité d'ajouter un événement en cliquant sur un bouton qui ouvre grâce à JQUERY le formulaire d'ajout d'événement. Lorsqu'il clique sur une ligne du tableau, il a alors accès à plus d'informations sur l'événement et peut alors le modifier, le supprimer ou accéder à une autre page lui donnant la liste des personnes inscrites à cet événement. Sur cette dernière page, un select lui permet de choisir l'évènement qui l'intéresse, qui met alors à jour le tableau par JQUERY et AJAX, sur lequel nous reviendront plus tard. Sur cette page il a la possibilité d'ajouter un participant à l'événement, pour cela il suffit juste de remplir le formulaire en stipulant le login de l'utilisateur que l'on veut ajouter le niveau de

cet utilisateur, et enfin le mot de passe admin. Une dernière possibilité est offerte à l'admin, le fait d'ajouter et de supprimer des photos sur la page galerie. Pour ajouter des photos un bouton le redirige vers une page avec un formulaire sur lequel il peut uploader des images et leur ajouter une description.

## 3

# BASE DE DONNÉES

---

Le site est basé sur une base de données comportant 4 tables distinctes.

### 3.1 LA CONNEXION

---

La connexion à la base de données est la première étape lors du chargement du site. Elle est effectuée sur chaque page par la page "index.php". La base de données est stockée dans une variable ('\$dbh') qui est envoyé dans les fonctions d'enregistrement.

### 3.2 LA TABLE UTILISATEUR

---

La table utilisateur regroupe l'ensemble des personnes inscrites sur le site. Elle est associée à sa classe php, 'utilisateur', que l'on retrouve dans "script/utilisateur.php". La classe permet de récupérer un utilisateur, de le supprimer, de changer le mot de passe d'un utilisateur. Cette classe est composée de 9 champs : le login, le nom, le prénom, le mot de passe, l'email, la promotion, la date de naissance, Admin, et le style.

**Le champ login** Le login est unique et propre à chaque utilisateur, elle est la clé primaire de la table. C'est le login qui permet de se connecter et de s'identifier sur le site. Le login est d'ailleurs stocké dans la variable de sessions lors de la connexion. C'est un champ varchar de 64 bites.

**Le champ nom, le champ prénom, le champ email, la promotion, la date et la date de naissance** Ce sont des champs simples permettant de stocker des informations sur l'utilisateur. Le nom et le prénom sont stockés dans la variable de SESSION pour éviter de devoir faire appel à la base de données trop régulièrement.

**Le champ mot de passe** Il apporte une sécurité en plus pour se connecter. Il est crypté grâce à la méthode SHA1.

**Le champ Admin** Ce champ n'a que 2 valeurs possibles null ou égal à 1. Il permet de savoir si l'utilisateur est administrateur ou non du site. Il permet de débloquent certaines fonctionnalités du site. Il est stocké dans la variable de SESSION qui est initialisée uniquement si l'utilisateur est admin.

**le champ style** A terme il permettra à chaque utilisateur de personnaliser sa page avec des styles différents. Cette fonctionnalité n'est malheureusement pas encore fonctionnelle.

### 3.3 LA TABLE EVENEMENT

---

La table evenement regroupe l'ensemble des événements proposés par le binet Styl'X. Elle est associée à sa classe php, "Event", codée par le fichier "scripts/utlis.php". Cette classe regroupe les différentes informations nécessaires à un événement, son nom, sa description, sa date de début, sa date de fin, une description et un id qui est la clé primaire. La classe Event est munie de très nombreuses fonctions permettant de modifier un événement d'en rajouter, d'en supprimer. Elle est aussi munie de fonctions récupérant les événements sous des formats intéressants permettant au calendrier par exemple de les lire.

**Le champ id** Ce champ est composé de 12 bites, généré de manière aléatoire et transformé en nombre hexadécimal. Cette méthode de génération du champ id permet de le rendre unique mais aussi de rendre l'id des événements introuvables et donc d'empêcher n'importe qui de pouvoir y accéder.

**Les champs date** Ils sont dans le fuseau horaire français, et interpréter comme telle dans le code.

### 3.4 LA TABLE INSCRIPTION

---

L'inscription est assez simple, elle regroupe 3 champs seulement : le login de l'élève, l'id de l'event, et le niveau de l'élève. Elle regroupe donc l'ensemble des inscriptions à chaque événement. Elle est bien sûr associée à sa classe inscription dans "scripts/inscription.php", qui permet de récupérer les inscriptions, et de les supprimer. La table est utilisée dans les classes Event et Utilisateur, avec des tableaux croisés.

### 3.5 LA TABLE IMAGES

---

Elle regroupe l'ensemble des id des images présentes sur la page galerie. L'id est un entier incrémenté de 1 à chaque fois que l'on rajoute une image. Elle contient aussi un champ légende

et date, donnant des informations sur l'image. Elle est aussi associée à une classe permettant de les supprimer d'en rajouter et de récupérer l'id de la dernière photo.

## 4

# FONCTIONNEMENT GLOBAL DU SITE

---

Tout le site est centré sur la page `index.php` qui se charge d'appeler les autres fichiers pour créer le header et le contenu. Elle se charge aussi de faire les "todo" présents dans l'url, sauf exception particulière pour les formulaires où ils sont gérés par les pages pour pouvoir afficher l'erreur au bon endroit. La grande majorité des pages est basée sur JQUERY. En effet, pour afficher des questionnaires, les remplir automatiquement, on fait appel à des fonctions javascript. Une grande majorité des pages est associée à son fichier javascript : par exemple la page `"upload.php"` est associée au javascript `"upload.js"`. Il permet notamment, de vérifier que la taille des fichier uploadés ne dépasse pas la capacité du site. De manière générale, seuls les enregistrements dans la base de données sont gérés grâce à des "todo".

Le fonctionnement de JQUERY pour l'apparition des formulaires est le même partout. Les questionnaires sont initialement masqués grâce à la classe `"w3-hide"`, le jQUERY, lui retire ou ajoute cette classe en fonction de s'il l'a déjà ou non pour afficher le questionnaire, ce qui évite de devoir recharger la page avec un "todo" dans l'url.

## 4.1 LES PAGES CALENDRIER

---

Les pages calendrier sont générées par fullcalendar. Les calendriers sont remplis par "ajax", qui fait appel à l'url : `"scripts/get-events.php"`. Cette requête ajax fait alors appel à la base de données, et récupère l'ensemble des événements pour les afficher dans le calendrier. L'apparition du formulaire pour s'inscrire ou modifier un événement est géré par JQUERY, qui permet de remplir le questionnaire en fonction de si l'événement existe déjà ou non.

## 4.2 LES PAGES TABLEAUX

---

Les tableaux sont générés par datatables. Les tableaux sont eux aussi remplis par ajax, l'url spécifié varie en fonction de la page sur laquelle on se trouve et des données que l'on souhaite afficher. Sur les pages calendrier admin, un bouton permet l'apparition d'un questionnaire pour ajouter un événement ou un participant, c'est géré par JQUERY. Il en est de même pour l'affichage des éléments supplémentaires lorsque l'on clique sur une ligne du tableau.

## 4.3 LA PAGE GALERIE ET L'UPLOAD DES PHOTOS

---

La page galerie repose essentiellement sur une flexbox, qui organise les photos. Lors de l'upload des photos deux copies de la même photo sont créées, une à la taille originale, et une autre dont la taille a été fixée à 100px de large. L'id de la photo est généré en prenant l'id le plus haut de la base de données et en ajoutant 1. JQUERY permet de s'assurer que la taille des fichiers uploadée ne dépasse pas celle supportée par APACHE. Pour éviter les bugs, l'upload de fichier autre que jpg ou jpeg est bloqué lors de l'upload et au moment de l'enregistrement.

# 5

## ORGANISATION DU PROJET

---

Pour nous organiser nous sommes passés par GitHub pour répartir le travail entre différentes branches et différentes issues.

Pour la bonne compréhension des fichiers, ils ont été répartis entre petits dossiers portant des noms explicites, seul "index.php" reste en dehors de dossier étant la base du projet. Ainsi, le dossier "scripts" contient toutes les classes pour la bonne utilisation de la base de données. Le dossier "routes" contient l'ensemble des routes pour les demandes ajax. Le dossier "js" contient tous les javascripts nécessaires au bon fonctionnement des pages. Le dossier "images" contient toutes les images du site et le dossier "css" tous les fichiers css. Enfin "addremovephoto" contient les formulaires pour les photos et toutes les fonctions nécessaires à l'upload des photos. Il en est de même pour "EventAddRemove" qui contient les fonctions pour l'ajout et la suppression d'événements et "LogInOut" pour les formulaires et les fonctions de connexion ou de déconnexion. Enfin, "content" contient le fichier php permettant de générer les pages.

# 6

## SÉCURITÉ ET ERREUR

---

La première sécurité est que le mot de passe n'est jamais en clair, mais est crypté par SHA1. Cela reste cependant assez faible notamment au moment de la requête POST. Le protocole https devrait venir combler cette lacune. Chaque affichage d'un élément envoyé par l'utilisateur est envoyé grâce à "htmlspecialchars()" pour éviter tout affichage d'élément non voulu. Les injections SQL se font grâce à un tableau. Seuls les pages listées, au début de "utilities/utills.php" sont autorisées. Pour éviter toutes erreurs, lors du rajout dans la base de données, il est vérifié à chaque fois, qu'il n'y a pas de violation de clé primaire.

Enfin toutes les pages ont été vérifiées par "validator.w3.org"

## 7

# PETITES FONCTIONNALITÉS EN PLUS

---

- Le bon enregistrement dans la base de données est montré avec un flag vert à chaque fois. Lorsqu'il y a une erreur c'est un flag rouge qui est affiché.
- Lorsque l'on est connecté, on ne peut pas créer de compte
- Lors du rechargement d'une page avec un todo, le nouvel enregistrement n'est pas pris en compte.

## 8

# CE QUI MANQUE

---

On souhaiterait remplacer chaque todo par une requête AJAX post lorsque c'est possible, pour éviter de devoir recharger les pages lorsque ce n'est pas nécessaire, notamment pour le calendrier et les tableaux.

Le style général du site est à retravailler, on aimerait travailler plus le css pour lui donner un vrai style.

Un utilisateur ne peut pas récupérer son mot de passe lorsqu'il a perdu, on aimerait mettre un système pour combler cette lacune.

On aimerait aussi donner la possibilité à un utilisateur de changer ses informations personnelles autres que le mot de passe.