



# Documentation technique

Stage première année BTS SIO





<b>Objectif.....</b>	<b>3</b>
<b>Technologies utilisées.....</b>	<b>3</b>
<b>Structure du site.....</b>	<b>4</b>
<b>1. Page d'Accueil (index.php).....</b>	<b>4</b>
Description.....	4
Contenu.....	4
Technologies.....	4
<b>2. Planning des Évènements (evenements.php).....</b>	<b>5</b>
Description.....	5
Contenu.....	5
Technologies.....	5
<b>3. Page Jeux (jeux.php).....</b>	<b>5</b>
Description.....	5
Contenu.....	5
Technologies.....	5
<b>4. Page Liens Utiles (liens.php).....</b>	<b>6</b>
Description.....	6
Contenu.....	6
Technologies.....	6
<b>5. Page Histoire (histoire.php).....</b>	<b>6</b>
Description.....	6
Contenu.....	6
Technologies.....	6
<b>6. Page des Pin's (pins.php).....</b>	<b>7</b>
Description.....	7
Contenu.....	7
Technologies.....	7
<b>7. Page des Photos (photos.php).....</b>	<b>8</b>
Description.....	8
Contenu.....	8
Technologies.....	8
<b>8. Page de présentation (presentation.php).....</b>	<b>8</b>
Description.....	8
Contenu.....	8
Technologies.....	9
<b>9. Mode admin (admin.php).....</b>	<b>9</b>
Description.....	9
Contenu.....	10
Technologies.....	10

## Objectif

Créer un site web pour l'association LASEN, mes tâches étant la création des pages d'activités, de jeux, d'événements, d'historique et de liens utiles. Le site sera codé en HTML, CSS, JavaScript et PHP.

## Technologies utilisées

Technologie	Rôle
HTML5	Structure des pages
CSS3	Mise en forme visuelle
JavaScript	Interactions côté client (animations, menus, validations)
PHP	Traitement côté serveur (chargement dynamique, formulaires)
MySQL	Stockage des données (événements, jeux, liens, etc.)



## Structure du site

Le site sera divisé en plusieurs pages accessibles via une barre de navigation.

### 1. Page d'Accueil (**index.php**)

#### Description

C'est la première page visible par les visiteurs. Elle contient une présentation rapide de LASEN, une bannière, et un accès rapide aux autres pages.

#### Contenu

- Logo et nom
- Bannière
- Présentation de l'association
- Accès rapide aux derniers événements
- Liens vers les autres pages

#### Technologies

- HTML pour la structure
- CSS pour la mise en page responsive
- JavaScript pour les animations ou sliders
- PHP pour afficher dynamiquement certaines parties (ex: actualités)

## 2. Planning des Évènements (**evenements.php**)

### Description

Liste des événements organisés par LASEN

### Contenu

- Calendrier puis liste des événements
- Détails des événements (date, lieu, description)

### Technologies

- PHP + MySQL : récupération dynamique des événements depuis une base de données
- JavaScript : pour un calendrier interactif
- HTML/CSS : affichage lisible et moderne

## 3. Page Jeux (**jeux.php**)

### Description

Liste des jeux proposés par LASEN

### Contenu

- Nom du jeu
- Image/affiche
- Description
- Catégorie
- Lien pour en savoir plus (fiche détaillée ou site officiel)

### Technologies

- PHP : chargement dynamique des jeux
- JavaScript : tri ou recherche par nom ou catégorie
- HTML/CSS : mise en page sous forme de cartes

## 4. Page Liens Utiles ([liens.php](#))

### Description

Page avec une liste de liens utiles pour les membres de LASEN

### Contenu

- Nom du lien
- Description courte
- Lien cliquable (s'ouvre dans un nouvel onglet)

### Technologies

- HTML/CSS : structure et style

## 5. Page Histoire ([histoire.php](#))

### Description

Présentation de l'historique de la Faluche depuis sa création ainsi que du folklore faluchard

### Contenu

- Texte de présentation
- Photos d'archives
- Dates importantes

### Technologies

- HTML : structure
- CSS : Structure et style de la page

## 6. Page des Pin's ([pins.php](#))

### Description

Présentation des pin's.

### Contenu

- Infos des pin's :
  - Titre
  - Description
  - Date de création
  - Ville de création
  - Association
  - Image du pin
- Barre de recherche (dans le header)
- Filtres avancés (dans le header)

### Technologies

- Dossiers : Pour les images des pin's
- SQL : Pour obtenir sous forme d'objet => [titre, description, date, ville, asso, image]
- PHP :
  - Composition de la requête SQL (sans filtre, avec filtre ou avec une recherche. Utilisation de \$\_POST)
  - Admin : Ajouter, modifier et supprimer des pin's de la BDD
- CSS
- *(JS - Optionnel : Ajuster l'affichage lors de l'utilisation d'un filtre).*
- *(AOS Lib - Optionnel : Lors du scroll, affiche les éléments de manière fluide et agréable. Très simple à inclure, fonctionne en JS).*

## 7. Page des Photos (**photos.php**)

### Description

Affichage des photos d'événements, triées d'abord par événement puis par année.

### Contenu

- Évènements (Catégories). En balise <h2>
- Années (Sous-catégories). En balise <h3>
- Images avec leur ratio d'origine, affichage en cascade.

### Technologies

- Dossiers :
  - Nom des dossiers = nom des événements
  - Nom des sous-dossiers = les années
- PHP :
  - Afficher les catégories, sous-catégories et photos de manière automatique
  - Admin : Ajouter, supprimer et modifier des dossiers et photos
- CSS
- (*JS - Optionnel : Affichage dynamique, rétrécir / agrandir les groupes de photos*).
- (*AOS Lib - Optionnel*).

## 8. Page de présentation (**presentation.php**)

### Description

Une page de présentation de l'association, avec des liens.

### Contenu

- Présentation du bureau.
- Photos, rôle et nom de chaque membre (+ contact, en cas de consentement).
- Les principaux événements (sous forme de liste) + lien vers la page des événements (*"en savoir plus"*).
- (Autres détails importants)
- (Optionnel : petit bouton qui mène à la page d'accueil)
- Une adresse mail (qu'on peut facilement copier).



## Technologies

- Fichiers .json :
  - 1 personne = 1 objet [nom, role, contact]
  - Textes
- Dossier : Contenant les photos des membres
- SQL : Pour afficher les événements
- PHP :
  - Communication avec le fichier .json (orienté objet, pour les textes)
  - Requête vers la BDD (pour les événements)
  - Affichage des images à partir du dossier
  - Admin : Ajouter, modifier et supprimer textes et photos
- CSS
- (AOS Lib - Optionnel).

## 9. Mode admin (**admin.php**)

### Description

Afin de rendre l'expérience plus fluide, le formulaire de connexion serait inclu dans toutes les pages (en hidden, donc invisible), mais deviendrait visible en appuyant sur un bouton. Une fois les identifiants entrés dans le formulaire, la page se recharge, cette fois en mode admin. Cette méthode évite d'ouvrir un nouvel onglet pour se connecter, ce qui rend les choses plus simples.

Lorsque l'utilisateur est connecté en admin, il peut ajouter, supprimer ou modifier du contenu. Pour ça, il faudra :

- Des fichiers .json : Pour tout ce qui est en objet
- Des fichiers textes : Pour les textes simples
- La BDD : Pour des éléments plus nombreux ou plus sensibles (Admins, pin's, etc.)
- Les dossiers : Surtout pour stocker les images



## Contenu

- Le bouton pour accéder au formulaire
- Le formulaire :
  - identifiant
  - MDP (mot de passe)
  - Un petit bouton pour afficher / cacher le MDP
  - Le bouton de connexion
- Les dispositifs de sécurité :
  - Protection contre SQLi
  - Protection bruteforce
  - Chiffrement des MDP
  - Token CSRF
- Des boutons pour ajouter, modifier et supprimer du contenu (Dépendent de chaque page)

## Technologies

- SQL :
  - Pour identifiants et mots de passe chiffrés
  - *(Optionnel : un ORM sur lequel je travaille, qui permettrait de faire des requêtes beaucoup plus facilement, de manière lisible et sécurisée)*
- PHP :
  - Requête sécurisées vers la BDD
  - Utilisation de \$\_SESSION
  - *(Optionnel : Logs de connexion)*
  - *(Optionnel : Sauvegarde temporaire des éléments supprimés (en cas de suppression involontaire ou mal intentionnée))*
- JS : Afficher et cacher le formulaire lorsqu'on appuie sur le bouton.
- CSS
- *(AOS Lib - Optionnel).*
- *(Font Awesome - Optionnel : Icônes parfaites pour des boutons simple : Édition, suppression, modification et connexion, par exemple)*