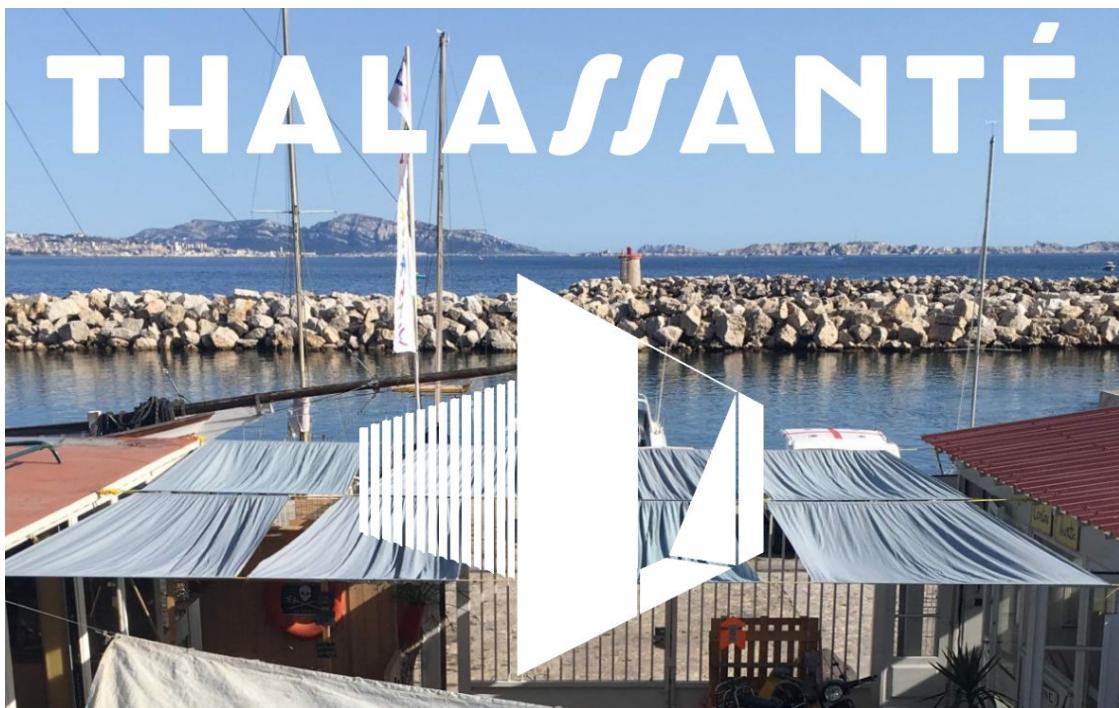


Dossier de Projet Professionnel

Création de site web Thalassanté

Présenté par Mohamed HADJADJI



Sommaire

1. Résumé de projet	3
2. Présentation	3
2.1 Présentation personnelle	3
2.2 Présentation de Thalassanté	3
3. Compétence référentiel couvert par le projet	4
3.1 Connaissance en langages informatiques	4
3.2 Outils de stockage et de partage	5
4. Cahier des charges	5
4.1 Site vitrine	5
4.2 L'Admin panel	
7	
5. Spécifications techniques et sécurité	8
6. Développer la partie front-end	9
6.1 Maquetter la version mobile de site	9
6.2 Réaliser l'interface utilisateur web statique et adaptable	11
6.3 Développer l'interface utilisateur web dynamique	13
7. Développer la partie back-end	15
7.1 Créer la base des données.....	15
7.2 Développer les composants d'accès au données	16
7.3 Développer le back-end de la partie événements	17

1. Résumé de Projet

Ce dossier présente mon projet professionnel effectuée en tant qu'un étudiant au centre de formation La Plateforme. Le projet est de créer le site web de l'association Thalassanté, pour amener de la visibilité à cette association dépourvu de site web, et de leur faciliter la vie.

Le projet est divisé en deux parties, le site vitrine qui est une interface web statique pour les visiteurs et qui présente plusieurs parties bien distinctes sur la même page. Et la deuxième partie l'admin panel qui sera utilisé comme un outils de gestion pour toutes les activités de l'association ainsi que pour centraliser leur point d'émission de leur communication, via des API d'applications et réseaux sociaux (Slack, Facebook, Instagram...).

2. Présentation

2.1 Présentation personnelle

Je m'appelle **Mohamed HADJADJI** j'habite à Marseille je suis ingénieur en économie agroalimentaire. J'ai une expérience professionnelle dans l'informatique de gestion et maintenant je souhaite devenir un développeur web, un métier qui me passionne encore plus. C'est pour ça que j'ai suivi cette formation de développeur web et web mobile à La Plateforme.

2.2 Présentation de Thalassanté

Thalassanté International est un lieu de vie composé de conteneurs maritimes aménagés implanté en bord de mer, à l'Estaque, depuis 20 ans. L'association est un tiers-lieu qui accueille les projets d'autres structures ou particuliers, en mettant des espaces de travail à disposition de divers acteurs agissant dans des domaines tels que l'artisanat, l'art, l'insertion, la culture, la santé, la culture maritime, le "vivre-ensemble"... Les synergies créées par ce laboratoire social s'accompagnent

également d'actions visant à dynamiser la zone Estaque-Riaux, par le biais de l'organisation d'événements. L'association est entièrement auto-financée.

3. Compétence référentiel couvert par le projet

3.1 Connaissance en langages informatiques

[Les langages HTML & CSS](#)

J'ai utilisé HTML pour structurer ma partie Activités et évènements avec la mise en place des balises html en suivant la maquette et les directives de notre web designer. En suite avec le CSS j'ai géré les emplacements des éléments avec la méthode "**flex**", et faire le responsive en utilisant "**media queries**", tout en respectant les couleurs et les tailles exigés.

[Les langages PHP, SQL, AJAX](#)

Pour la mise en place d'un systèmes de création des activités et leur composants pour l'administrateur, j'ai utilisé SQL à l'aide de **Phpmyadmin** pour créer la base de données avec une table "activités". En suite avec PHP j'ai créé une classe "activité" qui contient plusieurs fonctions qui contient des requêtes et des conditions pour insérer, supprimer et sélectionner des activités. Enfin pour faire appel à ces fonctions sans le rechargement de la page, j'ai utilisé AJAX.

[Le framework Bootstrap](#)

J'ai utilisé Bootstrap pour la mise en forme de formulaire d'insertion et le bouton de suppression des activités pour le côté admin, et pour le formulaire de sélection côté utilisateurs.

[L'API Facebook & JAVASCRIPT](#)

Pour récupérer les données des événements à partir de la page facebook de Thalassanté, j'ai utilisé l'API facebook, ensuite avec Javascript j'ai affiché tous ces événements par dates, images, titres et descriptions dans la partie "événements" de site web.

jQuery JAVASCRIPT

j'ai utilisé la bibliothèque jQuery pour créer des événements javascript et ajouté de style et de fluidité au site, comme l'événement “**.Dialog**” qui permet de créer un popup et afficher des formulaires et des affiches, et aussi “**.toggle**” qui permet d'afficher ou masquer les suites des textes.

3.2 Outils de stockage et de partage

Utilisation de Github

Le Github est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

J'ai utilisé le github pour stocker tout mes fichiers de code et pour pouvoir les partager les merger facilement avec mes camarades de groupe.

Utilisation de Trello

Trello est un outil de gestion de projet en ligne, Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

Avec Trello j'ai organisé et géré mes tâches à faire avec mon groupe de projet, en respectant chacun les deadline.

4. Cahier des charges

4.1 Site vitrine

Présentation:

Avec la mis en place d'un background image qui représente l'emplacement de l'association avec le logo officiel Thalassanté. Pour le côté admin on ajoute un formulaire d'insertion et de modification de texte qui présente l'association.

Barre de Navigation:

L'emplacement de la barre de navigation de site doit être juste après le background image de présentation, ensuite quand l'utilisateur se déplace vers le bas la barre reste fixer en haut de la page.

Galerie images:

La mis en place d'un Carrousel image, avec un formulaire admin qui sert à modifier les photos. Ensuite On utilise un API pour récupérer les photos de compte Instagram de Thalassanté, et les affichées à la galerie.

Lieu avec un Plan 3d:

La partie lieu de site doit afficher le plan 3d qui représente la structure de l'association, ce plan est fourni par le web designer. Ensuite à l'aide des boutons l'utilisateur peut afficher les détails de chaque zone ou conteneur.

Il faut aussi un formulaire pour l'admin qui sert à modifier le nom et la fonction de chaque bloc.

Activités:

L'affichage des activités doit être en bloc qui prend en considération l'image, le titre, description, et plusieurs domaine présentés par des couleurs spécifiques. Ensuite à l'aide d'un carrousel l'utilisateur peut faire dérouler les blocs des activités affichés en quatres par quatres, et il peut aussi les sélectionner avec des boutons par rapport aux domaines, et aux années.

L'admin à un formulaire pour l'insertion des activités et un boutons pour les supprimés.

Agenda:

L'agenda mis en place dans le site doit afficher les mêmes événements de l'agenda google de Thalassanté à l'aide d'un API google.

[Evénements:](#)

Les événements affichés sur le site doivent être récupérés de la page Facebook Thalassanté, grâce à l'API Facebook.

L'utilisateur peut faire le tour de ces événements à l'aide d'un carrousel, et les filtrer par année.

[Adhésion](#)

L'utilisateur peut participer et faire un don pour soutenir l'association, en accédant directement au site "**Helloasso**" depuis le site Thalassanté, grâce à un bouton.

[Contact:](#)

Dans la partie contact l'utilisateur peut envoyer des messages aux stuff de l'association, en remplissant un formulaire, et peut aussi accéder à la Newsletter Thalassanté.

4.2 L'admin Panel

[Tchat équipe](#)

[Newsletter](#)

[Partenaires](#)

[Outils de communication](#)

[Réseaux sociaux](#)

[Gestion utilisateur](#)

[Gestion contenu->signalement](#)

[Evénement/Agenda](#)

5. Spécifications techniques et sécurité

Pour sécuriser le coté admin de site et tous les boutons commandes et les formulaires d'insertion et modification, avec le PHP j'ai utilisé la condition suivante:

```
<?php
if (isset($_SESSION['user'])) {
?
<script type = "text/javascript" src="js/script.js"></script>
<?php }
else{
?
<script type = "text/javascript" src="js/script_user.js"><
script>
<?php } ?>
```

Donc j'ai créé deux script js, un pour les visiteurs de site et un autre pour l'administrateur, et ce dernier doit se connecter avec son compte pour avoir l'accès aux commandes.

Pour la création des activités, j'ai ajouté l'élément “**required**” dans les inputs de formulaire html, pour éviter d'envoyer des données vides à la BDD. Ensuite pour charger des images par l'input de type file, avec de php j'ai fait en sorte qu'il prend juste les extensions pour image.

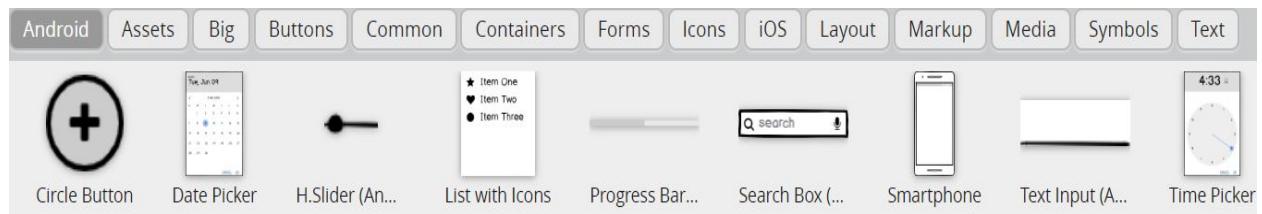
```
$valid_extensions = array('jpeg', 'jpg', 'png', 'gif', 'bmp', 'pdf', 'doc', 'ppt');
// extensions valides
$path = 'uploads/'; // télécharger le répertoire
$pathf = 'uploads_f/';
$img = $_FILES['image']['name']; // stocke le nom de fichier d'origine du client
$tmp = $_FILES['image']['tmp_name']; // stocke le nom du fichier temporaire désigné
$fich = $_FILES['fiche']['name'];
$tmpf = $_FILES['fiche']['tmp_name'];
// récupère l'extension du fichier téléchargé
$ext = strtolower(pathinfo($img, PATHINFO_EXTENSION));
$extf = strtolower(pathinfo($fich, PATHINFO_EXTENSION));
// peut télécharger la même image en utilisant la fonction rand
$final_image = rand(1000,1000000).$img;
$final_fiche = rand(1000,1000000).$fich;
// vérifier format valide
if(in_array($ext, $valid_extensions) && in_array($extf, $valid_extensions))
{
```

6. Développer la partie front-end

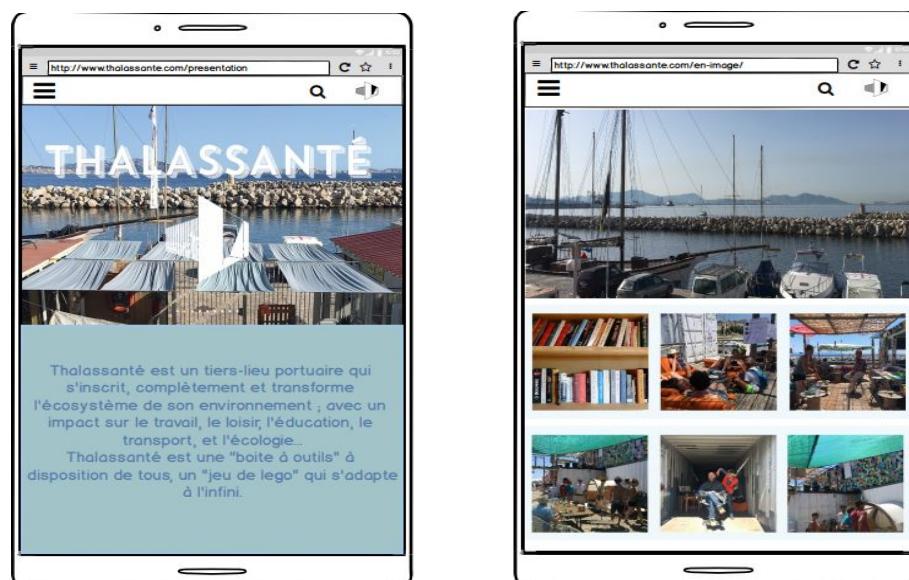
6.1 Maquetter la version mobile de site

J'ai utilisé les éléments communiqués par le graphiste de l'association, dans le cahier des charges. Le travail est donc divisé en deux parties, la première étant le maquettage de la version site web, et la deuxième la version web mobile pour les smartphones et les tablettes. Mon travail est de maquetter la deuxième version.

Pour la réalisation de cette maquette j'ai utilisé **Balsamiq** qui contient beaucoup d'éléments facilitant la mise en place d'une maquette d'un site mobile, avec des conteneurs et des supports version smartphone et tablette.

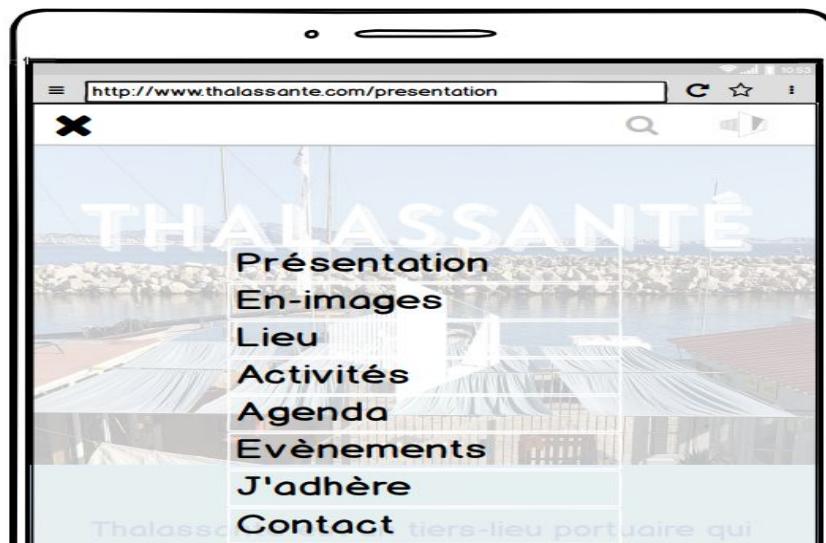
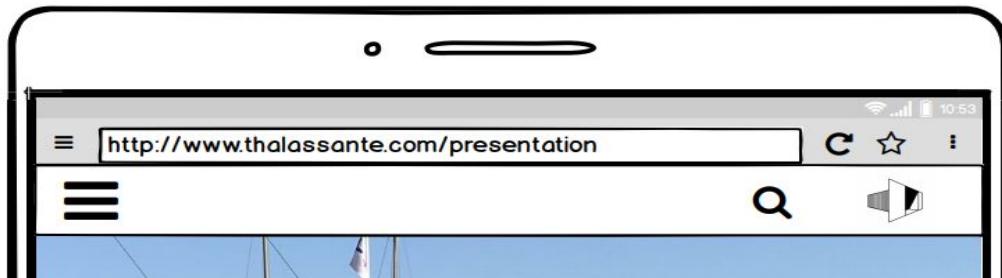


J'ai donc pris le support "Smartphone" dans "Android" et je l'ai inséré en plusieurs fois pour présenter chaque partie de site, avec un affichage responsive et bien clair pour les internautes mobile.





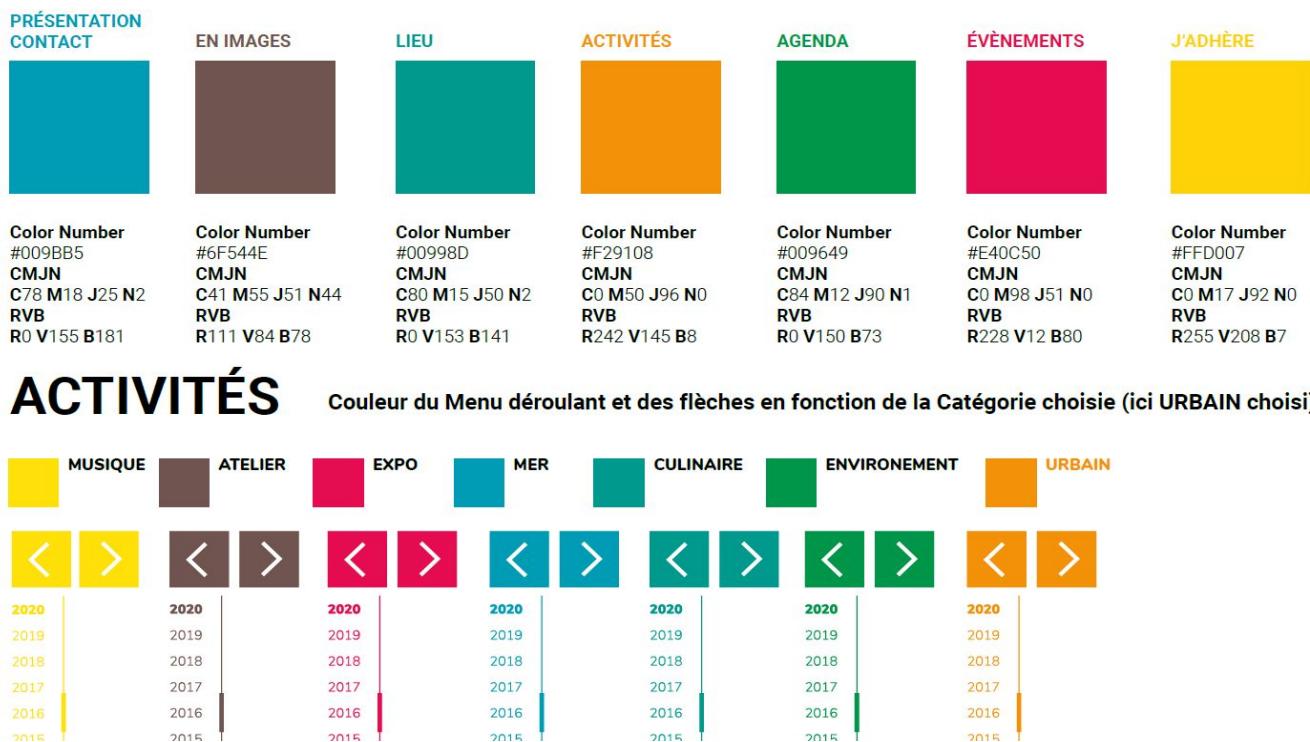
Pour la barre de navigation j'ai utilisé "App Bar" pour android qui contient le menu de site en format "hamburger", et j'ai ajouté une loupe pour "recherche" et le logo officiel de l'association.



Pour insérer les éléments et les photos fournies par le graphiste, j'ai utilisé "image" sur "Media" et "Text Area" de "Text", puis j'ai ajusté les éléments selon l'affichage communiqué par le client pour le support mobile.

6.2 Réaliser l'interface utilisateur web statique et adaptable

Le graphiste de l'association qui est un spécialiste en matière, nous a fourni en détail toute les polices et les tailles des textes à respecter ainsi que les codes couleurs pour chaque partie de site.



Pour réaliser l'interface statique et adaptable de les parties Activités et événements, j'ai utilisé HTML, CSS, JAVASCRIPT.

J'ai commencé par la structure et la mise en place des balises html avec les images, textes et boutons, tout en respectant l'ordre des éléments par rapport au cahier des charges. Ensuite avec le CSS en utilisant de "flex" et "media queries", j'ai respecté l'emplacement des éléments ainsi que la taille qui s'adapte avec l'affichage de l'écran (laptop, tablette, smartphone).

 INCUBATION			
 MARITIME			
 SOCIAL			
 TERRITOIRE			
 ENVIRONNEMENT			
 SPORT			
 ART			
 PEDAGOGIE			
 CONSTRUCTION	En savoir plus 	En savoir plus 	En savoir plus 
 CUISINE			
 RECHERCHE			
Valider			
TOUTS			

INCUBATION

MARITIME

SOCIAL

TERRITOIRE

ENVIRONNEMENT

SPORT

Activité 3

L'inscription sérielle de l'œuvre architecturale de Le Corbusier sur la Liste du patrimoine mondial est intervenue au mois de juillet 2016 lors de la réunion du patrimoine mondial est intervenue au mois de juillet 2016 lors de la réunion du patrimoine mondial est intervenue au mois de juillet 2016

En savoir plus

J'ai utilisé Javascript pour cacher la suite de texte de présentation d'activités, et en cliquant sur la flèche, la suite s'affiche. Et ça grâce aux événements jQuery ".click" et ".toggle".

```
//////fait apparaitre le texte
$('#mh-js-img-icon').click(function(){
  $('#oc-para-act-cacher').toggle(250); //affiche et desaffiche
})
```

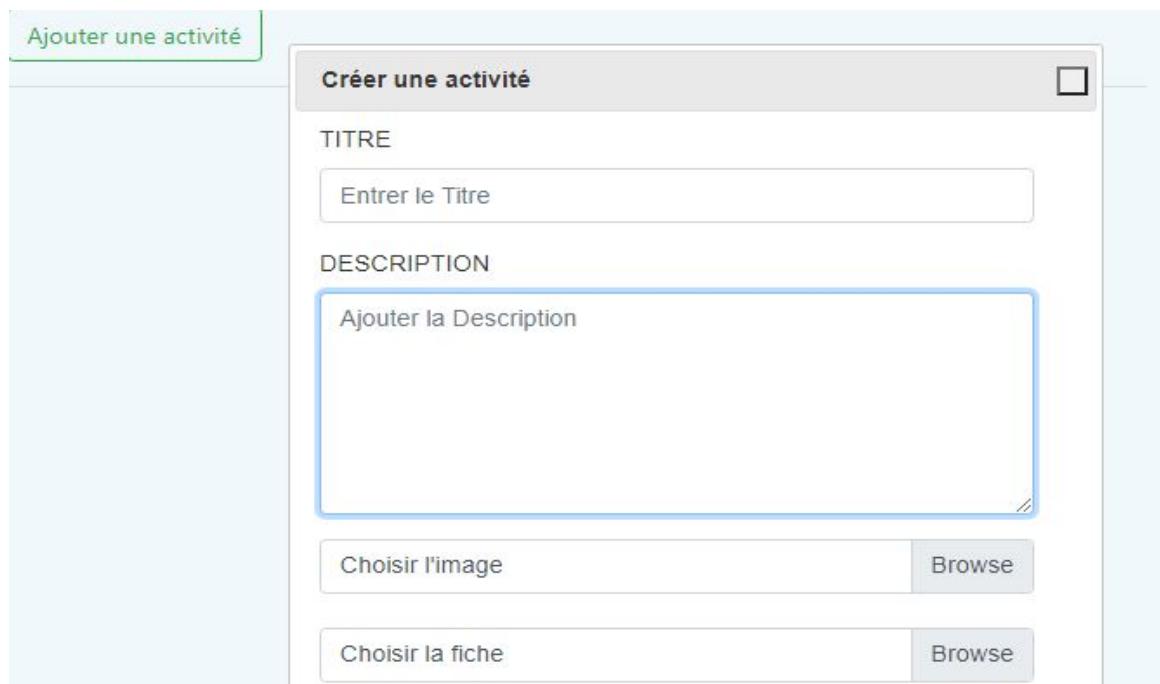
J'ai ajouté aussi des popup pour les affiches activités lorsque l'utilisateur clic sur "En savoir plus", en utilisant l'événement ".dialog".

```
$(function() {
  var fiche = "#fic"+ act[i]['id'];
  `#${fiche}`).dialog({
    autoOpen: false,
    width: 1030
  );
  var bofic = "#affic"+ act[i]['id'];
  $(document).on("click", `#${bofic}`, function() {
    `#${fiche}`).dialog('open');
  });
});
```

6.3 Développer l'interface utilisateur web dynamique

L'administrateur de site quand il se connecte via un module de connexion conçu spécialement pour la partie admin, l'interface de site change, et des boutons de commande ainsi que des formulaires s'ajoutent à la page, pour que l'administrateur peut ajouter, modifier et même supprimer des éléments comme le texte et les blocs d'activités.





Et cela grâce à la condition php "if (isset(\$_SESSION['user']))), comme le montre le code suivant:

```
<?php if (isset($_SESSION['user'])) { ?>
<section class="container">
    <div class="row">
        <div class="col-md-8">
            <button id="add" class="btn btn-outline-success">Ajouter une activité</button>
            <div id="dialog" title="Créer une activité">
                <form id="form" action="classe/fonction.php" method="post" enctype="multipart/form-data">
                    <div class="form-group">
                        <label for="titre">TITRE</label>
                        <input type="text" class="form-control" id="name" name="titre" placeholder="Entrer le Titre" required />
                    </div>
```

7. Développer la partie back-end

7.1 Crédation de la base des données

La base des données de la partie activités est composé d'une seul table "**activités**", et elle contient tous les éléments qui compose les blocs d'activités.

Pour créer cette base des données j'ai donc utilisé "**PhpMyadmin**"

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	id 	int(11)			Non	Aucun(e)		AUTO_INCREMENT
2	titre	varchar(255)	utf8_general_ci		Non	Aucun(e)		
3	description	varchar(600)	utf8_general_ci		Non	Aucun(e)		
4	image	varchar(255)	utf8_general_ci		Non	Aucun(e)		
5	affiche	varchar(255)	utf8_general_ci		Non	Aucun(e)		
6	date	varchar(50)	utf8_general_ci		Non	Aucun(e)		
7	incubation	varchar(255)	utf8_general_ci		Non	Aucun(e)		
8	maritime	varchar(255)	utf8_general_ci		Non	Aucun(e)		
9	social	varchar(255)	utf8_general_ci		Non	Aucun(e)		
10	territoire	varchar(255)	utf8_general_ci		Non	Aucun(e)		
11	environ	varchar(255)	utf8_general_ci		Non	Aucun(e)		
12	sport	varchar(255)	utf8_general_ci		Non	Aucun(e)		
13	art	varchar(255)	utf8_general_ci		Non	Aucun(e)		
14	pedago	varchar(255)	utf8_general_ci		Non	Aucun(e)		
15	construc	varchar(255)	utf8_general_ci		Non	Aucun(e)		
16	cuisine	varchar(255)	utf8_general_ci		Non	Aucun(e)		
17	recherche	varchar(255)	utf8_general_ci		Non	Aucun(e)		

7.2 Développer les composants d'accès aux données

Pour développer les composants d'accès à la base de données en utilisant du PHP, MySQL et AJAX (Javascript), j'ai suivi les étapes suivantes:

- Avant d'insérer les requêtes sql dans le code php, j'ai fait la connexion avec la base de données selon la méthode suivante:

```
$connexion = new PDO("mysql:host=localhost;dbname=thalassante", "root", "");
```

- Ensuite j'ai créé la classe "activité" dans laquelle j'ai introduit plusieurs fonctions avec des requêtes pour chaque bouton de sélection. et l'utilisateur peut faire le filtrage des activités en cliquant sur ces boutons qui représente les domaines existant, ou sur les années. Parmi ces fonctions il ya :

- Get activité pour tous sélectionné sans conditions.

```
public function get_activite()
{
    $connexion = new PDO("mysql:host=localhost;dbname=thalassante", "root", "");
    $connexion->exec("set names utf8");
    $reponseget = $connexion->query("SELECT * FROM activites");
    $reponseget->execute();
    $resultat = $reponseget->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($resultat);
}
```

- Get incub pour sélectionner juste les activités qui ont l'incubation.

```
public function get_incub()
{
    $connexion = new PDO("mysql:host=localhost;dbname=thalassante", "root", "");
    $connexion->exec("set names utf8");
    $reponseinc = $connexion->query("SELECT * FROM activites WHERE incubation = '#AA3614'");
    $reponseinc->execute();
    $resulinc = $reponseinc->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($resulinc);
}
```

- Get year pour sélectionner selon les années.

```
$reponsey = $connexion->query("SELECT * FROM activites WHERE date = '$year'");
$reponsey->execute();
$resuly = $reponsey->fetchAll(PDO::FETCH_ASSOC);
echo json_encode($resuly);
```

- Del activité pour supprimer des activités par l'admin.

```
public function del_activite($idact)
{
    $connexion = new PDO('mysql:host=localhost;dbname=thalassante', 'root', '');
    $connexion->exec("set names utf8");
    $repensesup = $connexion->query("DELETE FROM activites WHERE id = '$idact'");
    $repensesup->execute();
}
```

- Pour la création des activités par l'admin à l'aide d'un formulaire html j'ai créé une fonction avec une requête "INSERT INTO"

```
// inclut le fichier de configuration de la base de données
include_once 'bdd.php';
// insérer des données de formulaire dans la base de données
$insert = $db->query("INSERT activites (titre, description, image, affiche, date, incubation,
    maritime, social, territoire, environ, sport, art, pedago, construc, cuisine, recherche)
    VALUES ('".$titro."','".$descripto."','".$path."','".$pathf."','".$year."','".$incub."'
    , '".$mari."', '".$social."', '".$territ."', '".$enviro."', '".$sport."', '".$art."', '".$pedago."', '".$const."', '".$cuisine."', '".$reche."')");
|
```

- Enfin j'ai utilisé AJAX (jQuery) pour exécuter chacune des ces fonction et pour l'affichage, la sélection et la suppression des activités.

```
$.ajax({
    method: "POST",
    url:"source/fonc_act.php",
    data: { 'function': 'get_activite'},
    datatype: "json",
    success: function(datatype){
        $('.cont').remove();
        var act = JSON.parse(datatype);
        for (var i = 0; i < act.length; i++){
            $('#activ').append('<div class="cont" id="geta' + i +'"><
```



```
function supp_act(value){
    $('#ann' + value + "").click(function(){

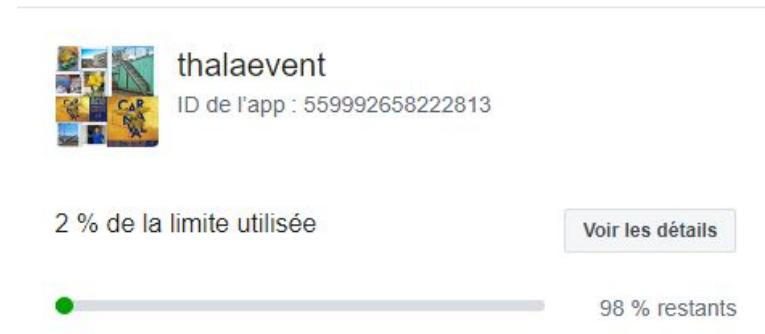
        $.ajax({
            url: "source/fonc_act.php",
            method : "POST",
            data : {'function' : 'del_activite', 'id_act' : value},
            datatype : "json",
        })
        get_act();
```

7.3 Développer le back-end de la partie événements

Pour afficher les événements de la page facebook de Thalassanté sur la partie événements de site, j'ai utilisé l'**API facebook**.

Sur le site: <https://developers.facebook.com/>, j'ai créé une nouvelle application

et je l'ai appelé "**thalaevent**", définie par l'id "**559992658222813**".



Ensuite dans "**Outil**", j'ai accédé au "**Explorateur de l'API Graph**", pour pouvoir générer le "**Token**" et ajouter les autorisations à l'application nécessaires à la récupération des données des événements recherchées (id, cover, name, start_time, description).

Token d'accès

EAAH9T3N4Xt0BAJ7QVZCQJY3iT... copier

Generate Access Token

App Facebook
thalaevent

Utilisateur ou Page
Token utilisateur

Autorisations ⓘ

- ✗ pages_read_engagement
- ✗ pages_manage_metadata
- ✗ pages_read_user_content
- ✗ pages_manage_ads
- ✗ pages_manage_posts
- ✗ pages_manage_engagement
- public_profile

Ajouter une autorisation

Pour afficher ces données des événements récupérer avec l'application, j'ai donc utilisé Javascript, en commençant par introduire le script de connexion avec l'application facebook.

```

window.fbAsyncInit = function() {
  FB.init({
    appId      : '564933664121574',
    cookie     : true,
    xfbml     : true,
    version   : 'v7.0'
  });
  FB.getLoginStatus(function(response) {
    statusChangeCallback(response);
  });
}

```

```

(function(d, s, id){
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) {return;}
  js = d.createElement(s); js.id = id;
  js.src = "https://connect.facebook.net/en_US/sdk.js";
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));

```

Ensuite j'ai inséré la fonction dans laquelle j'ai déclaré les données à afficher comme suit:

```

function testAPI(){
  FB.api(
    '/717455071627605?fields=id,name,events{cover,name,start_time,description,id}',
    function(response){
      if(response && !response.error){
        console.log(response);
        buildProfile(response);
      }
    }
}

```

Enfin j'ai créé la fonction "buildProfile(event)" pour l'affichage des informations pour chaque événements.

```
function buildProfile(even){  
    for (let i in even.events.data){  
        profile += `  
            <div class="evenement">  
                <a href="https://www.facebook.com/events/${even.events.data[i].id}"><img  
                    height="250" width="250" src = "${even.events.data[i].cover.source}"></a>  
                <div id="desc-event">  
                    <h2>${even.events.data[i].start_time}</h2>  
                    <h3>${even.events.data[i].name}</h3>  
                    <p>${even.events.data[i].description}</p>  
                    <a href="https://www.facebook.com/events/${even.events.data[i].id}"><button  
                        type="button" class="bout-fb">Voir sur facebook</button></a>  
                </div>  
            </div>  
        `;  
    }  
  
    document.getElementById('profile').innerHTML = profile;  
}
```

8. K