

Fiche de procédure

Stage en entreprise

Année 2024

Sommaire :

- [1 : Introduction](#)
- [2 : Récupération des Données dans une Table JSON](#)
 - [2.1 : Objectif](#)
 - [2.2 : Étapes](#)
 - Connexion à la Base de Données
 - Extraction des Données
 - Conversion des Données en JSON
- [3 : Intégration des Courbes de Production](#)
 - [3.1 : Objectif](#)
 - [3.2 : Étapes](#)
 - Préparation des Données
 - Création des Graphiques
- [4 : Mise en Place du Plugin PickaDate jQuery](#)
 - [4.1 : Objectif](#)
 - [4.2 : Étapes](#)
 - Installation et Initialisation du Plugin
 - Gestion des Sélections de Dates
- [5 : Ajout d'une Icône de Chargement](#)
 - [5.1 : Objectif](#)
 - [5.2 : Étapes](#)
 - Intégration dans l'interface utilisateur
 - Activation dynamique
- [6 : Accessibilité du Site](#)

1 : Introduction

Disposer d'un système de suivi précis et accessible de la production solaire journalière. L'objectif de ce projet est de développer une section dans une application web existante permettant de visualiser l'historique des données de production solaire, en s'appuyant sur une base de données SQL déjà en place pour le stockage et la gestion des informations.

2 : Récupération des données dans une table JSON

2.1 : Objectif :

Extraire les données de production solaire de la base de données SQL existante et les formater en JSON pour une utilisation dans l'application web.

2.2 : Étapes :

Connexion à la base de données :

Établir une connexion sécurisée à la base de données SQL en utilisant des bibliothèques comme PDO pour PHP.

Extraction des données :

Exécuter des requêtes SQL pour extraire les données de production solaire pour la période sélectionnée.

Conversion des données en JSON :

Parcourir les résultats de la requête SQL et convertir les données en format JSON pour une manipulation côté client.

Vérifier que les données JSON récupérées correspondent aux données stockées dans la base de données. Tester la performance de récupération des données.

3 : Intégration des courbes de production

3.1 : Objectif :

Afficher les courbes de production solaire pour les jours sélectionnés, incluant la production étalée sur la journée, la production totale, la consommation énergétique et le pourcentage des batteries.

3.2 : Étapes :

Préparation des données :

Traiter les données JSON pour les formater correctement et calculer les valeurs nécessaires.

Création des graphiques :

Utiliser des bibliothèques de visualisation de données pour créer des graphiques interactifs affichant les courbes de production.



Vérifier l'exactitude des graphiques et des données affichées. Tester la compatibilité des graphiques sur différents navigateurs et dispositifs.

4 : Mise en place du plugin PickaDate jQuery

4.1 : Objectif :

Intégrer le plugin PickaDate pour permettre aux utilisateurs de sélectionner des dates de manière simple et efficace.

4.2 : Étapes :

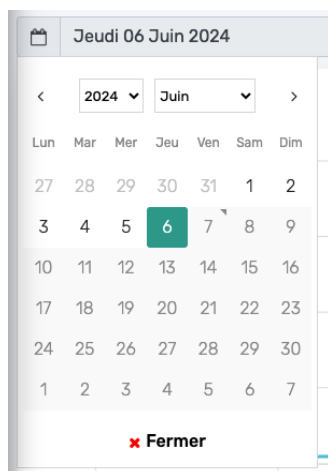
Installation et initialisation du plugin :

Inclure les fichiers JavaScript et CSS du plugin PickaDate, puis initialiser le plugin sur les champs de date du formulaire.

```
{%if histo == true %}
<div class="col-md-12 col-sm-12 col-xs-12 " id="reAffiche" style="margin-bottom: -25px; margin-top: 10px;">
  <div class="input-group">
    <div class="input-group-prepend">
      <span class="input-group-text" >
        <span class="fa fa-calendar-o"></span>
      </span>
    </div>
    <input type='text' class="form-control pickadate-limits"
      data-value="{{ toDay }}"
      id="dateSel"
      name="dateSel"
      placeholder=""/>
    </div>
  </div>
{% endif %}
```

Gestion des sélections de dates :

Ajouter des événements pour gérer les sélections de dates et mettre à jour les graphiques en conséquence.



Tester la fonctionnalité de sélection de dates pour garantir son bon fonctionnement sur différents navigateurs et dispositifs.

5 : Ajout d'une icône de chargement

5.1 : Objectif :

Implémenter une icône de chargement pour indiquer visuellement à l'utilisateur que des données sont en cours de chargement ou de traitement.

5.2 : Étapes :

Intégration dans l'interface utilisateur :

Placer l'icône de chargement à un emplacement visible sur la page, généralement à côté des éléments ou des zones où les données sont chargées ou mises à jour.

Activation dynamique :

Activer l'affichage de l'icône de chargement au début du chargement des données et le masquer une fois le chargement terminé.

```

var url = "{{ path('board') }}" + "?dtSel=" + dateSel + "&histo=" + {{histo}} + "";
showload();
$.ajax({
  type : 'GET',
  url : url,
  //data : data,
  success: function(response) {
    mylog('lien=' + url);
    console.log('lienfinal=' + url);
    $.ajax({
      type : 'GET',
      url : url,
      datatype: 'html',
      success: function(response) {
        $("#affPage").html(response);
        $("#dateSel").data('value', dateSel);
        hideload();
      }
    });
  }
});
  
```

Tester la fonctionnalité de l'icône de chargement dans différentes situations de chargement, y compris des chargements lents. Assurer que l'icône apparaît et disparaît de manière appropriée.

6 : Accessibilité du site

Le site est accessible à l'adresse suivante : <https://home.enlimousin.fr:9090/histo>

Le site est actuellement en phase de développement et d'optimisation, et certaines fonctionnalités peuvent ne pas être complètement terminées ou optimisées.